**PÓS-GRADUAÇÃO EM DESENVOLVIMENTO FULL STACK**

**Vinho Notas: Tornando a Experiência Enológica Acessível e Agradável**

VANDERLEI KLEINSCHMIDT

Orientador: Alexandre Agustini

2024



**SUMÁRIO**

[1. Contextualização da proposta 3](#_Toc166486697)

[1.1. Apresentação do problema 3](#_Toc166486698)

[1.2. Apresentação da solução da proposta 4](#_Toc166486699)

[2. Objetivos da construção da solução 6](#_Toc166486700)

[2.1. Objetivo estratégico 6](#_Toc166486701)

[2.2. Objetivos específicos 6](#_Toc166486702)

[3. Elaboração da jornada do usuário 7](#_Toc166486703)

[3.1. Requisitos funcionais 8](#_Toc166486704)

[3.2. Requisitos não funcionais 9](#_Toc166486705)

[3.3. Diagrama de casos de uso 9](#_Toc166486706)

[3.4. Diagrama de atividades 12](#_Toc166486707)

[3.5. Personas 12](#_Toc166486708)

[3.5.1. Personas: Marina 13](#_Toc166486709)

[3.5.2. Personas: Ricardo 15](#_Toc166486710)

[3.5.1. Personas: André 15](#_Toc166486711)

[3.6. Jornada dos usuários 17](#_Toc166486712)

[3.6.1. Jornada da Marina 17](#_Toc166486713)

[3.6.2. Jornada do Ricardo 18](#_Toc166486714)

[3.6.3. Jornada do André 19](#_Toc166486715)

[4. Apelo mercadológico da solução 21](#_Toc166486716)

[4.1. O modelo de negócios 23](#_Toc166486717)

[4.1.1. Segmentos de cliente 23](#_Toc166486718)

[4.1.2. Proposta de valor 24](#_Toc166486719)

[4.1.3. Canais 24](#_Toc166486720)

[4.1.4. Relacionamento com clientes 24](#_Toc166486721)

[4.1.5. Fontes de receita 24](#_Toc166486722)

[4.1.6. Recursos principais 25](#_Toc166486723)

[4.1.7. Atividades chave 25](#_Toc166486724)

[4.1.8. Parcerias principais 25](#_Toc166486725)

[4.1.9. Estrutura de custos 25](#_Toc166486726)

[5. Ciclo de desenvolvimento da solução 26](#_Toc166486727)

[5.1. Sprint 27](#_Toc166486728)

[5.2. Sprint Planning 27](#_Toc166486729)

[5.3. Daily Scrum 27](#_Toc166486730)

[5.4. Jira Software 28](#_Toc166486731)

[5.4.1. Organização das tarefas 30](#_Toc166486732)

[5.5. Versionamento do código 31](#_Toc166486733)

[5.6. Etapas do desenvolvimento 31](#_Toc166486734)

[6. Mockup da proposta de solução 34](#_Toc166486735)

[6.1. Menu Cadastro 36](#_Toc166486736)

[6.2. Menu Vinho 39](#_Toc166486737)

[6.3. Menu Degustação 42](#_Toc166486738)

[6.4. Menu harmonização 43](#_Toc166486739)

[7. Arquitetura de Software 45](#_Toc166486740)

[7.1. Banco de dados 45](#_Toc166486741)

[7.1.1. Diagrama de esquema para o banco de dados 46](#_Toc166486742)

[7.2. O Back End 52](#_Toc166486743)

[7.3. O front end 55](#_Toc166486744)

[8. Validação da solução 56](#_Toc166486745)

[9. Registros das evidências do projeto 59](#_Toc166486746)

[10. Considerações finais e expectativas 64](#_Toc166486747)

[Referências 66](#_Toc166486748)

**LISTA DE QUADROS**

[Quadro 3.1 – Requisitos funcionais do sistema 8](#_Toc166495004)

[Quadro 3.2 – Requisitos não funcionais do sistema 9](#_Toc166495005)

[Quadro 9.1 – Micro serviços e seus endereços 59](#_Toc166495006)

[Quadro 9.2 - Endpoints da API de cadastro 60](#_Toc166495007)

[Quadro 9.3 - Endpoints da API de vinho 61](#_Toc166495008)

[Quadro 9.4 - Endpoints da API de degustação 62](#_Toc166495009)

[Quadro 9.5 - Endpoints da API de avaliação 62](#_Toc166495010)

[Quadro 9.6 - Endpoints da API de harmonização 63](#_Toc166495011)

**LISTA DE FIGURAS**

[Figura 3.1 – Diagrama de usos de caso UC01 9](#_Toc166486749)

[Figura 3.2 – Diagrama de casos de uso UC02 11](#_Toc166486750)

[Figura 3.3 – Diagrama de casos de uso UC03 11](#_Toc166486751)

[Figura 3.4 – Diagrama de casos de uso UC04 12](#_Toc166486752)

[Figura 3.5 – Diagrama de atividades 13](#_Toc166486753)

[Figura 3.6 – Persona Marina 14](#_Toc166486754)

[Figura 3.7 – Persona Ricardo 16](#_Toc166486755)

[Figura 3.8 – Persona André 17](#_Toc166486756)

[Figura 5.1 – Projeto Vinho Notas 28](#_Toc166486757)

[Figura 5.2 – Linha do temo do projeto Vinho Notas 29](#_Toc166486758)

[Figura 5.3 – Backlog da Sprint 12 29](#_Toc166486759)

[Figura 5.4 – Quadro da Sprint 12 30](#_Toc166486760)

[Figura 5.5 – o Business Model Canvas 32](#_Toc166486761)

[Figura 5.6 – Criação do repositório no GitHub 33](#_Toc166486762)

[Figura 6.1 – Início da jornada do usuário 34](#_Toc166486763)

[Figura 6.2 – Tela de login 35](#_Toc166486764)

[Figura 6.3 – Tela inicial 35](#_Toc166486765)

[Figura 6.4 – Menu Cadastro 36](#_Toc166486766)

[Figura 6.5 – Fluxo de consulta de endereços 37](#_Toc166486767)

[Figura 6.6 – Tela de listagem de endereços 37](#_Toc166486768)

[Figura 6.7 – Tela de listagem de pessoas 38](#_Toc166486769)

[Figura 6.8 – Tela de listagem de usuários 39](#_Toc166486770)

[Figura 6.9 – Tela de vinho 39](#_Toc166486771)

[Figura 6.10 – Fluxo de listar vinhos 40](#_Toc166486772)

[Figura 6.11 – Tela de lista de vinhos 41](#_Toc166486773)

[Figura 6.12 – Tela de avaliações 41](#_Toc166486774)

[Figura 6.13 – Menu fichas de degustação 42](#_Toc166486775)

[Figura 6.14 – Tela lista de degustações 43](#_Toc166486776)

[Figura 6.15 – Fluxo de degustação 43](#_Toc166486777)

[Figura 6.16 – Tela de sugestão de harmonização 44](#_Toc166486778)

[Figura 7.1 – Diagrama de esquema para o banco de dados de cadastro 46](#_Toc166486779)

[Figura 7.2 – Diagrama de esquema para o banco de dados de vinho 47](#_Toc166486780)

[Figura 7.3 – Diagrama de esquema para o banco de dados de avaliação 48](#_Toc166486781)

[Figura 7.4 – Diagrama de esquema para o banco de dados de degustação 49](#_Toc166486782)

[Figura 7.5 - Diagrama Entidade Relacionamento do cadastro 50](#_Toc166486783)

[Figura 7.6 - Diagrama Entidade Relacionamento do vinho e degustação 51](#_Toc166486784)

[Figura 7.7 - Diagrama Entidade Relacionamento para avaliação 52](#_Toc166486785)

[Figura 7.8 – Diagrama do fluxo das requisições 53](#_Toc166486786)

[Figura 7.9 – Desenho da arquitetura em camadas 54](#_Toc166486787)

[Figura 8.1 – TDD 57](#_Toc166486788)

[Figura 8.2 – Relatório do JaCoCo para cobertura de testes unitários do BFF 58](#_Toc166486789)

# Contextualização da proposta

## Apresentação do problema

O vinho, uma bebida milenar apreciada por sua complexa combinação de sabores e aromas, conquista cada vez mais adeptos no Brasil. No entanto, para muitos, esse universo ainda é cercado de mistérios e desafios. Em uma definição fria, pode-se dizer que é uma bebida alcoólica obtida a partir da fermentação de uvas. Mas para quem a aprecia, vai muito além de uma simples definição. Pensando do ponto de vista de um enófilo, ela é uma mistura de sabores, aromas e texturas que permite uma verdadeira viagem em cada taça. Degustar é como atravessar o tempo e o espaço, entre a essência de uma safra e uma experiência sensorial única.

A evolução do mercado brasileiro de vinhos é uma realidade inegável. Segundo Lanari (2020), houve um crescimento considerável na década encerrada em 2020. Cerca de 39 milhões de pessoas bebem vinho regularmente, enquanto 44 milhões o consomem sem a mesma regularidade. Ao todo, estima-se que cerca de 40% da população brasileira consuma essa bebida de alguma forma.

Esses dados são corroborados por Ozbun (2024), que aponta uma demanda de 439 milhões de litros em 2022, com algo em torno de 2 litros por ano per capita desde 2019. O autor também cita o Rio Grande do Sul como o principal produtor e exportador de vinhos no país. Em termos de proporção de consumo, cerca de dois terços do que é comercializado no Brasil é produzido domesticamente, enquanto o restante é importado principalmente do Chile, Argentina e Portugal, cujos principais destinos são Santa Catarina e São Paulo.

Diversos fatores contribuem para o crescimento desse mercado no Brasil. Com as mudanças no padrão de consumo, e considerando que hoje as pessoas têm acesso a vários canais de compra, como o e-commerce, o vinho está mais acessível do que nunca. Além disso, a maturidade da indústria, o crescimento econômico do país, o aumento na renda e o avanço das pessoas na classe média indicam um futuro promissor.

Apesar do crescimento do mercado, ainda existem desafios para o consumidor. De acordo com Croft (2002, p.7), “Por causa de sua fascinante variedade, o vinho pode parecer um tema difícil de dominar”. Degustar um vinho pode parecer uma tarefa difícil, mas com a devida orientação, é possível ter momentos prazerosos entre amigos enquanto se decifra as nuances do líquido de uma taça. O olhar, cheirar, provar e concluir vão muito além de apenas beber o seu conteúdo. Se por um lado essa experiência sensorial pode se tornar única, por outro lado amedronta uma grande quantidade de pessoas que gostariam de se desafiar experimentando coisas novas.

Além de degustar, há ainda a questão de harmonizar corretamente o que se bebe com o que se come. A harmonia que se obtém a partir da combinação de aromas e sabores do vinho com a comida, ainda é um mistério para muitos, que transferem essa tarefa para especialistas. Conforme Melo (2020, p. 11), “A harmonização, potencializa e enriquece as características dos alimentos e das bebidas, podendo alcançar o auge da experiência em que uma simples refeição torna-se um acontecimento memorável”.

Por trás disso tudo ainda existe outro universo, o das taças, serviço, temperatura e guarda. Dito de outra forma, o manuseio do vinho. Entender qual taça usar, ou ainda como abrir e em qual temperatura servir, se deve aerar ou não e por quanto tempo antes de degustar, e o que fazer com a garrafa após aberta, também gera frustração para quem é de fora desse mundo, por não entender o porquê desse ritual.

## Apresentação da solução da proposta

É justamente para ajudar o apreciador de vinho amador ou ainda aquele que deseja entrar nesse mundo que se propõe a criação do Vinho Notas. Em outras palavras, para o enófilo amador que tem dificuldade em compreender os métodos e técnicas de degustação devido às complexidades expostas anteriormente, o Vinho Notas é uma aplicação web progressiva (PWA[[1]](#footnote-1)). Ele facilita a experiência de degustar e, diferentemente dos websites e aplicativos de compras, essa solução melhora a experiência na jornada de descobertas no mundo dos vinhos, ajudando a organizar e registrar as degustações, permitindo recuperar o histórico e sugerindo opções para compras futuras.

O Vinho Notas é um facilitador para enófilos amadores, uma plataforma PWA organizadora de degustações. A ideia é que seja gratuita para esse perfil de público, simples de usar e intuitiva. Por outro lado, não é um e-commerce, nem uma rede social e muito menos um app mobile.

Basicamente, o que o Vinho Notas faz é armazenar as informações dos vinhos adquiridos, criar um rating (avaliação por nota) dos vinhos degustados e classificá-los de acordo com certos critérios definidos pelo usuário, como por exemplo, uva, país, cor do vinho etc. Ele também virtualiza a adega, ajuda na organização e auxilia o usuário no processo de degustação. Apesar de guardar os dados dos diversos usuários da aplicação, ele não faz comparação de avaliações feitas com as de outros usuários, e tão pouco recomenda a compra de um determinado vinho em uma loja específica.

Ao usar essa aplicação, o usuário se sentirá mais confortável para descobrir novos rótulos ou para ingressar no mundo dos vinhos com mais segurança. Outro benefício que se pode obter a partir da sua utilização é evitar que o enófilo incorra em erros de decisão, pois terá um histórico de seus gostos e preferências, permitindo assim fazer escolhas mais assertivas.

Portanto, isso torna a jornada dos novos consumidores de vinho mais fácil e melhora muito a experiência de quem já é experiente nesse ramo.

# Objetivos da construção da solução

## Objetivo estratégico

O objetivo estratégico consiste em melhorar o conhecimento dos processos que envolvem a construção de um software, desde a sua concepção até a entrega final da solução seguindo a metodologia ágil e as melhores práticas.

## Objetivos específicos

Abaixo estão relacionados os objetivos específicos:

1. Construir um projeto no *Jira Software[[2]](#footnote-2)* para organização e controle das tarefas e dos processos;
2. Criar um projeto no GitHub[[3]](#footnote-3) utilizando *branchs* para facilitar o versionamento do código;
3. Construir a solução utilizando micro serviços, dentro de uma arquitetura com separação de camadas desde o banco de dados, *back end* na forma de *API*s consumidas por um *front end*, com a utilização de um BFF[[4]](#footnote-4) como gerenciador de chamadas das requisições;
4. Realizar a configuração de containers Docker deixando a aplicação pronta para produção.

# Elaboração da jornada do usuário

A jornada do usuário é uma peça importante no projeto de software, pois estabelece uma conexão entre quem efetivamente utilizará a solução e quem a está construindo. Segundo Pressman e Maxim (2021, p.237), “A engenharia de requisitos abrange sete tarefas distintas: concepção, levantamento, elaboração, negociação, especificação, validação e gestão”.

As etapas que envolvem a engenharia de requisitos não necessariamente precisam acontecer separadamente. É perfeitamente plausível que algumas delas ocorram de forma concomitante antes e durante a elaboração do projeto.

Em termos de concepção, percebe-se que a entrada de novos consumidores de vinho pode ser um processo complexo e, de certa forma, inibidor. Dessa forma, em linhas gerais, o que se pretende resolver com a construção do Vinho Notas é justamente quebrar essa barreira, garantindo que os futuros enófilos tenham uma forma organizada e segura de iniciar sua trajetória nesse mundo.

Por outro lado, há aqueles que já tem algum conhecimento, consomem os chamados vinhos de mesa, ou seja, vinhos de baixo custo e sem muita complexidade em termos de aromas e sabores. Esse público específico precisa de apoio para mudar de nível, mas muitas vezes não encontra quem lhes proporcione isso. Nesse sentido, a solução proposta pode servir de apoio, atuando em degustações guiadas, fornecendo o caminho necessário para que eles mudem de patamar.

E, por fim, os enófilos mais experientes muitas vezes necessitam organizar melhor as suas degustações, registrando-as de alguma maneira e dessa forma construindo uma base sólida de conhecimento. Essas experiências podem ser compartilhadas posteriormente com outros enófilos ou utilizadas como base para construção de um perfil, ou até mesmo auxiliando em novas descobertas.

Tendo em vista a concepção apresentada, buscou-se as metas de cada grupo de perfil na etapa de levantamento. Assim, um novato no mundo dos vinhos tem por meta principal comprar vinhos bons, que agradem seu paladar e que não lhe custe muito dinheiro. Já os intermediários que querem dar um passo adiante gostariam de experimentar algo novo, mas para isso, querem entender qual o seu perfil de vinho, ou qual a base das suas preferências. Por fim, os enófilos experientes já têm uma ideia do seu perfil, têm um orçamento mais alto para investir em novas descobertas e só querem se organizar e ter uma fonte de consulta rápida e eficiente.

A fase de levantamento, que é o que se pretende descrever neste capítulo, segue com a elaboração de um modelo de levantamento de requisitos ou de casos de uso no próximo tópico.

## Requisitos funcionais

As principais funcionalidades da aplicação estão descritas no quadro 3.1.

Quadro 3.1 – Requisitos funcionais do sistema

|  |  |  |
| --- | --- | --- |
| Requisito | Descrição | Use Case |
| FR01 | O usuário deve acessar o sistema mediante cadastramento prévio de pessoa, usuário e senha | UC01, UC02, UC03, UC04 |
| FR02 | Para se cadastrar no sistema, a pessoa deve informar seu nome completo, CPF, data de nascimento e endereço completo | UC01 |
| FR03 | Para cadastrar o usuário no sistema, deve ser escolhida uma pessoa previamente cadastrada, pode-se selecionar um dos perfis disponíveis (Enófilo, Sommelier e Parceiro), deve informar um e-mail válido e uma senha numérica de 4 dígitos | UC01 |
| FR04 | Para cadastrar o vinho, o usuário deve informar o rótulo, podendo preencher os demais campos conforme sua necessidade ou desejo | UC02 |
| FR05 | Os vinhos cadastrados devem ser listados na tela onde o usuário pode editar ou excluir um vinho | UC02 |
| FR06 | O usuário pode avaliar o vinho, devendo selecionar um dos rótulos já cadastrados na tela de listagem de vinho, podendo preencher os demais campos conforme sua necessidade ou desejo | UC03 |
| FR07 | As avaliações feitas pelos usuários devem ser listadas na tela, onde ele pode editar ou excluir uma avaliação | UC03 |
| FR08 | O usuário pode cadastrar uma degustação devendo informar o vinho que será degustado, a data e o tipo (vertical, horizontal, temática, às cegas, de harmonização ou comparativa) | UC04 |
| FR09 | O usuário pode criar uma ficha de degustação onde ele pode atribuir conceitos à sua percepção visual, olfativa, gustativa e atribuir uma nota geral ao vinho degustado | UC04 |
| FR10 | As degustações registradas devem ser listadas na tela onde o usuário pode editar ou excluir | UC04 |
| Fonte: o autor (2024). | | |

## Requisitos não funcionais

Os requisitos que estão presentes na aplicação, mas que não interferem diretamente na funcionalidade estão descritos no quadro 3.2. Para Pressman e Maxim (2021, p. 248), “Um requisito não funcional (NFR, do *inglês nonfunctional requirement*) pode ser descrito como um atributo de qualidade, de desempenho, de segurança ou como uma restrição geral em um sistema”.

Quadro 3.2 – Requisitos não funcionais do sistema

|  |  |
| --- | --- |
| Requisito | Descrição |
| NFR01 | O sistema deve ser acessado por usuário maior de 18 anos |
| NFR02 | O armazenamento de dados sensíveis deve seguir a Lei Geral de Proteção de Dados |
| NFR03 | Pode ser acessado pela internet mas também *offline*[[5]](#footnote-5) |
| NFR04 | A comunicação entre o frontend e o backend deve se dar através de APIs Restfull em um padrão de BFF (Backend For Frontend) |
| NFR05 | A aplicação deve ser de página única, SPA (Single Page Application) construída na forma de uma PWA (Progressive Web Application) |
| Fonte: o autor (2024). | |

## Diagrama de casos de uso

Figura 3.1 apresenta o diagrama de casos de uso UC1. Esse diagrama é “... uma maneira de descrever as interações entre usuários e um sistema usando um modelo gráfico e um texto estruturado” (Sommerville 2018, p.108).

Figura 3.1 – Diagrama de usos de caso UC01

|  |
| --- |
|  |
| Fonte: o autor (2024). |

Na Figura 3.1, o usuário deve acessar o sistema mediante cadastramento prévio como pessoa, utilizando um usuário e senha. Para isso, ele deve primeiro se cadastrar como pessoa, onde será realizada a validação se ele possui idade igual ou superior a 18 anos. Ao se registrar, a pessoa deve fornecer seu nome completo, CPF, data de nascimento e endereço completo. Após o cadastramento e as validações pertinentes, a próxima etapa consiste em cadastrar o usuário no sistema. Durante o cadastramento, o usuário seleciona um dos perfis disponíveis (Enófilo, Sommelier e Parceiro) e deve fornecer um e-mail válido, além de uma senha numérica de 4 dígitos.

O perfil de usuário Enófilo estará disponível na versão 1.0 do sistema. Esse usuário não pagará mensalidades e terá acesso a todas as funcionalidades do sistema. Em versões posteriores, serão disponibilizados outros dois perfis, o Sommelier que poderá utilizar o Vinho Notas como ferramenta de trabalho mediante parceria, oferecendo cursos de degustação online, o que permitirá a ele gerar uma fonte extra de renda. O terceiro perfil é o Parceiro, que pode ser uma vinícola, um importador de vinhos, uma casa de comércio ou um e-commerce que queira disponibilizar aos seus clientes a possibilidade de utilizar a plataforma para registrar e acompanhar o estoque de vinhos adquiridos e até mesmo fazer sugestões de compra futura.

A figura 3.2 apresenta os casos de uso UC02. Nesse cenário o usuário pretende registrar um vinho na aplicação. Para isso ele deve primeiramente fazer o login no sistema, depois acessar a tela de cadastro de vinho informando o rótulo como campo de preenchimento obrigatório. Os demais campos são opcionais e neles o usuário pode registrar o preço do vinho em R$, o local e a data da compra, o tipo de vinho e sua classificação, a graduação alcoólica e o volume da garrafa. Também pode registrar os dados da uva, o produtor, a temperatura de serviço, a safra, país de origem, tempo de guarda, região produtora do vinho, maturação e de como fazer a harmonização do vinho com comida. Os vinhos cadastrados devem ser listados na tela onde o usuário pode editar ou excluir um vinho, conforme sua necessidade.

O usuário pode fazer uma avaliação do vinho previamente registrado no sistema, como pode ser visto no diagrama de casos de uso UC03, na figura 3.3. Para avaliar o vinho o usuário deve estar logado no sistema, acessar a listagem de vinhos cadastrados, selecionar um dos rótulos e fazer o registro da avaliação. Na avaliação o usuário informa quando fez a degustação daquele vinho, quais os aspectos visuais ele percebeu, quais os aromas e sabores. Ele pode dar uma opinião geral sobre o vinho e atribuir uma avaliação final. Todas as avaliações feitas pelos usuários são listadas na tela e ele pode editar ou excluir uma avaliação.

Figura 3.2 – Diagrama de casos de uso UC02

|  |
| --- |
|  |
| Fonte: o autor (2024) |

Figura 3.3 – Diagrama de casos de uso UC03

|  |
| --- |
|  |
| Fonte: o autor (2024) |

A figura 3.4 apresenta o diagrama de casos de uso UC04, onde o usuário faz o login na aplicação, em seguida seleciona no menu, a opção de cadastrar uma degustação. Para isso ele informa o vinho que será degustado, a data e o tipo (vertical, horizontal, temática, às cegas, de harmonização ou comparativa). O passo seguinte consiste em criar uma ficha onde pode atribuir conceitos à sua percepção visual, olfativa, gustativa e atribuir uma nota geral ao vinho degustado. E finalmente, as degustações registradas são listadas na tela onde o usuário pode editar ou excluir.

Figura 3.4 – Diagrama de casos de uso UC04

|  |
| --- |
|  |
| Fonte: o autor (2024) |

## Diagrama de atividades

Pressman e Maxim (2021, p. 330) descrevem o diagrama de atividades como uma forma de complemento do caso de usos por meio de fluxos de interação. Dessa forma é possível perceber como o sistema reage a eventos internos. A figura 3.5 apresenta o diagrama de atividades, onde é possível ver todas as etapas listadas anteriormente nos casos de uso.

## Personas

A definição das personas dá uma ideia de quem é o usuário do sistema ou de para quem essa solução se destina. Conforme Caroli (2015, p. 62), “Uma persona representa um usuário do sistema, descrevendo não só o seu papel, mas também suas necessidades específicas”. A ideia é criar uma representação de uma pessoa o mais próximo possível da realidade, descrevendo quem é essa pessoa, o que ela pensa, como ela se comporta e como ela utilizaria a solução proposta.

Figura 3.5 – Diagrama de atividades

|  |
| --- |
|  |
| Fonte: o autor (2024) |

### Personas: Marina

A primeira persona é a Marina, uma Designer Gráfica de 28 anos. A figura 3.6 apresenta uma imagem ilustrativa da Marina, criada a partir das descrições passadas para uma IA generativa, o Copilot da Microsoft[[6]](#footnote-6).

Ela tem um perfil de uma pessoa que está começando a explorar o mundo dos vinhos, mas até então não possui um conhecimento aprofundado sobre o tema. Ela vê o vinho como mais uma opção entre outras bebidas alcoólicas. Não tem preferência específica por regiões vinícolas, uvas ou marcas, e geralmente escolhe o vinho com base em recomendações rápidas ou promoções. Considera como uma bebida ocasional e está aberta a experimentar diferentes tipos e estilos, mas ainda não desenvolveu um paladar refinado.

Figura 3.6 – Persona Marina

|  |
| --- |
| Mulher posando para foto em frente a mesa com comida  Descrição gerada automaticamente |
| Fonte: o autor (2024) |

Suas necessidades são de busca por praticidade na escolha do vinho, valorizando sugestões simples e diretas na hora da compra, sem muita complexidade ou informação técnica. Deseja uma abordagem descomplicada para começar a compreender melhor sobre vinhos, sem sentir-se sobrecarregada por detalhes complexos ou excesso de opções. Procura por maneiras mais informais e acessíveis de aprender, como eventos de degustação descontraídos ou guias básicos para iniciantes.

Em termos de comportamento, pode-se dizer que ao comprar vinho, geralmente escolhe opções acessíveis em supermercados ou lojas comuns, sem se ater muito à qualidade ou detalhes específicos. Não investe muito tempo em pesquisas sobre ou em participar de eventos relacionados. Prefere aprender de maneira casual, experimentando novos rótulos quando surge a oportunidade. Não é fiel a uma marca ou tipo específico, e muitas vezes substitui o vinho por outras bebidas, dependendo do contexto ou das preferências do momento.

### Personas: Ricardo

A segunda persona é o Ricardo. Uma imagem gerada pelo Copilot está na figura 3.7. Ele tem 35 anos e é Consultor de Marketing. O seu perfil é de uma pessoa apaixonada por vinhos há alguns anos e que tem se aprofundado no mundo enológico. Aprecia a história por trás de cada vinícola e valoriza a tradição das uvas e marcas renomadas. Possui um conhecimento intermediário e está sempre disposto a aprender mais, frequentando cursos, workshops e degustações. Sua preferência por vinhos cresceu ao ponto de substituir outras bebidas em ocasiões sociais ou até mesmo em momentos relaxantes em casa.

Seus desejos são de acesso a uma ampla variedade de vinhos, tanto os clássicos quanto novas descobertas, procurando lojas que ofereçam uma seleção diversificada e opções exclusivas. Valoriza a qualidade acima do preço, por isso busca orientação para fazer escolhas assertivas, confiando na expertise de especialistas para suas compras. Busca experiências enogastronômicas, como harmonizações e degustações guiadas, para aprimorar sua compreensão sobre vinhos e ampliar seu paladar.

Em termos comportamentais, costuma reservar parte do seu orçamento para investir em vinhos de qualidade, preferindo adquiri-los em lojas especializadas, onde pode receber orientações e recomendações personalizadas dos sommeliers. Passa tempo considerável pesquisando online e lendo avaliações antes de comprar um vinho, priorizando a qualidade, a safra e a reputação da vinícola. Participa ativamente de eventos como festivais, feiras e *tours* por vinícolas, onde busca expandir seu conhecimento e apreciar novos rótulos.

### Personas: André

A terceira persona é o André de 32 anos, cuja imagem gerada pela IA está na figura 3.8. Sua profissão é Consultor Financeiro e ele é um entusiasta do vinho. Consta que descobriu sua paixão por essa bebida há alguns anos e desde então se tornou um verdadeiro aficionado. Ele a vê como uma forma de arte e aprecia a diversidade de sabores e aromas encontrados em diferentes rótulos.

Figura 3.7 – Persona Ricardo

|  |
| --- |
| Homem com garrafa de bebida  Descrição gerada automaticamente |
| Fonte: o autor (2024) |

Está constantemente em busca de novos vinhos para experimentar, querendo ampliar seu conhecimento e paladar ao máximo, deseja provar coisas novas, de diversas regiões e uvas, desde os clássicos até os mais exóticos. Sua paixão o levou a participar ativamente de eventos, degustações em vinícolas, clubes e até mesmo a se juntar a confrarias locais para compartilhar experiências e trocar dicas.

Busca constantemente por novos rótulos e experiências, procurando lojas especializadas, vinícolas ou eventos onde possa descobrir vinhos diferentes e únicos. Valoriza o compartilhamento de conhecimento e a interação com outros entusiastas, buscando comunidades e grupos onde possa trocar informações, dicas e opiniões. Deseja explorar ao máximo esse mundo, planejando visitas a regiões vinícolas, participando de viagens enoturísticas e expandindo seu repertório a cada oportunidade.

Investe considerável tempo e recursos em sua paixão, reservando parte do seu orçamento para a compra de garrafas especiais, frequentemente adquirindo vinhos recomendados por outros entusiastas ou especialistas. Utiliza as redes sociais como plataforma para compartilhar suas experiências enológicas, postando fotos de suas aquisições, comentando sobre o que provou e interagindo com outros amantes da bebida. Além de buscar a diversidade de vinhos, também investe em conhecimento, lendo livros, participando de cursos e workshops para aprimorar sua compreensão sobre esse mundo.

Figura 3.8 – Persona André

|  |
| --- |
| Homem de terno e gravata ao lado de garrafa de bebida na mão  Descrição gerada automaticamente |
| Fonte: o autor (2024) |

## Jornada dos usuários

As três personas definidas anteriormente se depararam com a mesma situação. Foram convidados para o mesmo evento e precisam levar uma garrafa de vinho. A seguir será descrita a jornada de cada um deles dentro desse cenário.

### Jornada da Marina

Marina recebe um convite para um jantar entre amigos, no qual decidem compartilhar uma garrafa de vinho. Ela realiza uma pesquisa em seu Vinho Notas sobre os tipos de vinhos que já degustou, suas características, sugestões de harmonização, preço e onde os adquiriu.

Ao visitar um supermercado local, com base nas informações básicas encontradas no Vinho Notas, escolhe um vinho branco que parece agradável e tem um preço acessível. Marina sai do supermercado e vai para a casa dos amigos.

Durante o jantar, experimenta o vinho que levou e percebe que gostou da experiência. Também degusta o vinho trazido pelos outros amigos e registra suas percepções no Vinho Notas. Ela entende que precisa desenvolver mais seu paladar e sua sensibilidade. Então, começa a comparar a sua escolha com o que os outros amigos trouxeram, usando a avaliação que fez no Vinho Notas. Faz comparações entre vinhos da mesma cor e de cores diferentes. Descobre que o preço pago pelos amigos por suas garrafas difere muito do preço que ela pagou. Tenta perceber a diferença entre os mais caros e os mais baratos.

Ela se sente mais à vontade para explorar novos rótulos e está aberta a recomendações. Aproveita a oportunidade para aprender sobre o processo de degustação. Decide que começará a descobrir novos rótulos, expandindo gradualmente seu paladar.

Compartilha sua experiência nas redes sociais, incluindo sua avaliação dos vinhos degustados, e está ansiosa para continuar sua jornada de descoberta no mundo do vinho.

### Jornada do Ricardo

Ricardo se depara com um convite para um jantar entre amigos, onde decidem compartilhar uma garrafa de vinho. Sabendo que será uma oportunidade de socializar experiências, fica animado para escolher algo especial.

Antes de ir à loja, faz uma rápida reflexão sobre suas preferências pessoais. Ele pensa em uvas e regiões que têm apreciado recentemente. Para essa análise ele recorre ao seu histórico no Vinho Notas. Para garantir uma escolha assertiva, Ricardo lê suas avaliações de vinhos no aplicativo, que se encaixem no perfil do jantar.

Decidido a buscar algo mais específico, Ricardo visita uma loja especializada em vinhos. Ele pede orientação ao *sommelier* sobre opções que se adequem ao estilo do jantar e compara a indicação com suas degustações no Vinho Notas. Ele leva em consideração as preferências dos amigos que estarão no jantar, escolhendo uma garrafa que acredita que agradará a todos os paladares presentes.

Antes de finalizar, Ricardo verifica a harmonização do vinho escolhido com o cardápio do evento, garantindo uma combinação agradável. Após fazer a escolha, Ricardo compra a garrafa e a tem cuidadosamente empacotada adicionando um cartão de vinho explicando sua escolha e registra a compra no Vinho Notas.

Ao chegar ao jantar, ele apresenta a garrafa escolhida com entusiasmo, compartilhando brevemente suas razões e encorajando todos a desfrutarem da experiência. Ricardo aproveita a oportunidade para transmitir algumas informações sobre o seu vinho, destacando detalhes que considera interessantes e registra suas percepções no Vinho Notas.

Ao longo do jantar, engaja-se em conversas com os amigos, trocando experiências e aprendendo mais sobre as preferências de cada um. Ele experimenta o vinho trazido pelos seus colegas e registra a experiência no Vinho Notas. Ele compartilhou tudo nas redes sociais, incluindo sua avaliação dos vinhos que degustou e encoraja a ideia de futuras explorações vinícolas em grupo.

### Jornada do André

André foi convidado para um jantar entre amigos, onde decidem compartilhar uma garrafa de vinho. Ele vê a oportunidade como mais uma chance de explorar novos rótulos. Animado com a perspectiva de experimentar algo novo, André pensa em como essa ocasião pode ser uma oportunidade para expandir ainda mais sua paleta de sabores.

André busca recomendações no Vinho Notas, com base em suas experiências, para descobrir quais rótulos mais lhe agradaram recentemente. Em vez de seguir recomendações específicas, ele decide escolher um vinho que desperte sua curiosidade. Está aberto a experimentar algo novo e diferente.

Ele visita uma loja ou adega especializada, onde pode explorar rótulos menos convencionais. Busca a orientação do pessoal da loja para encontrar algo único e surpreendente. Movido pela curiosidade, opta por um vinho de uma região ou variedade de uva menos comum, escolhendo algo que nunca tenha provado antes e registra a compra no Vinho Notas. Ao fazer a compra, pede um empacotamento especial para dar um toque mais personalizado à sua escolha.

Animado para compartilhar sua descoberta, André chega ao jantar com antecedência. Está ansioso para ver as reações dos amigos ao experimentar algo único. Apresenta o vinho escolhido, compartilhando brevemente a história por trás do rótulo e as características que o tornam especial. Durante o jantar, aproveita o momento para discutir as nuances do seu vinho, anotando suas percepções no Vinho Notas. Experimenta o os seus colegas trouxeram e registra no Vinho Notas.

Ele compartilhou tudo nas redes sociais, incluindo sua avaliação dos vinhos que degustou. Incentiva os amigos a fazer o mesmo e a trocarem impressões sobre diferentes vinhos.

# Apelo mercadológico da solução

O mercado de vinhos no Brasil está passando por uma fase de normalização do consumo, porém possui uma relevância econômica que não pode ser ignorada. A pandemia pelo Corona Vírus alterou profundamente os hábitos de consumo e a busca por coisas novas na internet. Sites de *e-commerce* de vinho ganharam destaque em motores de busca como o Google[[7]](#footnote-7). Mas o pós-pandemia alterou o cenário, trazendo novas oportunidades.

Goresnstein (2024), discutindo as expectativas para o mercado em 2024, aponta para “... alguns fenômenos devem marcar o cenário vinícola no Brasil”. Entre eles o aumento na comercialização do segmento *premium* e *superpremium* que se beneficiam de períodos de crescimento econômico. Vinhos de baixo custo também são destacados em função da recuperação econômica do país.

Entre as novas tendências mundiais, está "... a busca por produtos com menor teor alcoólico ou completamente desalcoolizados...", os vinhos orgânicos, biodinâmicos e naturais e o crescimento de vinhos brancos, rosés e espumantes (GORESNSTEIN, 2024).

Oliveira (2024) aponta algumas tendências para 2024 em nível global, como por exemplo o janeiro sem álcool da França, vinhos *no-low* álcool nos países anglo-saxônicos, escandinavos, França e Alemanha. Destaca também a busca por "... vinhos frescos e leves, com cores mais claras e naturais..." (OLIVEIRA, 2024) e a busca pelos jovens por vinhos sustentáveis, com embalagens com novos materiais que não agridem a natureza.

Uma coisa que chama a atenção no cenário para 2024 é a mudança na faixa etária dos consumidores, como destaca Barelli (2024). Há uma redução na percentagem de pessoas que consomem vinho com regularidade em algumas faixas etárias. Porém a autora destaca que os espumantes estão em alta e devem permanecer assim em 2024, projetando um mercado mundial para essa categoria em algo em torno de US$ 55,4 bilhões em 2028.

Apesar da acomodação que o mercado consumidor está enfrentando em 2024, é sempre bom lembrar os números levantados por Ozbun (2024), mostrando um mercado consumidor girando em torno de 44 milhões de pessoas, ou algo como 40% da população brasileira.

Magalhães (2022), faz uma análise da importância do comércio eletrônico para o varejo. Ela lembra que há 20 anos as páginas amarelas eram a forma como se buscava fornecedores de produtos e serviços. Para a autora, "...o coronavírus se mostrou um dos maiores aceleradores de futuro da história contemporânea, antecipando mudanças que já estavam em curso" (MAGALHÃES, 2022).

É justamente nesse cenário, que o *e-commerce* de vinhos cresceu e se desenvolveu. Tudo começa em 2002 com a Estação de Vinho que foi o primeiro *e-commerce* brasileiro desse segmento. A partir dessa empresa, em 2008, dois sócios criaram a Wine que atualmente é uma das líderes do setor no Brasil. E finalmente em 2013 surge a Evino, com uma estratégia de descontos agressivos, tornando-se outra gigante do comércio eletrônico brasileiro (MAGALHÃES, 2022).

Atualmente existem muitos *players* nesse mercado, que é dominado pela Evino, Wine e Vivino, seguidos de perto por tradicionais vinícolas nacionais como a Famiglia Valduga, Aurora, entre outras. Uma maneira que essas empresas encontraram para fazer suas vendas é através de clubes de assinatura. O funcionamento é bem simples, consiste em um pagamento mensal onde o assinante recebe periodicamente uma ou mais garrafas, conforme o plano contratado.

Mas, como o mundo dos vinhos ainda parece um campo estranho e complexo para boa parte das pessoas, a existência de um comércio *online* não é suficiente para atrair novos consumidores. Existem rituais com certas especificidades que exigem um pouco de estudo e conhecimento. Para suprir essa demanda surgiram as degustações *online* e os cursos de vinho pela *internet*, como uma forma de alcançar um público maior do que as suas versões tradicionais e presenciais são capazes de alcançar.

Há uma quantidade enorme de *sites* que oferecem cursos de degustação e harmonização, fornecendo aos seus alunos *e-books* e apostilas como material de apoio, lojas oficiais para vender artigos personalizados, além de vídeo aulas com acesso por prazo determinado.

Obviamente que em um mundo cada vez mais mobilizado, a existência de aplicativos para celular é imperativo. Nessa linha, os grandes *players* do *e-commerce* possuem suas versões para IOS[[8]](#footnote-8) e Android[[9]](#footnote-9), como por exemplo a Evino[[10]](#footnote-10), Vivino[[11]](#footnote-11) e Wine[[12]](#footnote-12).

Os aplicativos que mais se parecem com o Vinho Notas são o Wine Notes[[13]](#footnote-13) que é um app que permite catalogar a impressão pessoal sobre vinhos, mas não recebe atualização desde 2020 e o Delectable Wine - Scan & Rate[[14]](#footnote-14), que também permite catalogar vinhos. Para uma análise comparativa pode-se recorrer a Caviquioli (2021) que traz mais detalhes sobre estes apps e outros.

## O modelo de negócios

Para descrever o modelo de negócios utilizado na ideação do Vinho Notas, utilizou-se o *Business Model Canvas*, proposto por Osterwalder e Pigneur (2011). Tudo é construído em cima de um quadro composto por 9 elementos: segmentos de clientes, proposta de valor, canais, relacionamento com clientes, fontes de receita, recursos principais, atividades chave, parcerias principais e estrutura de custo.

### Segmentos de cliente

Os segmentos que o Vinho Notas se propõe a atender são quatro. O primeiro deles é o enófilo que pode ser definido como o amante de vinhos, ou aquele que quer aprender mais sobre o tema. O segundo grupo é o *sommelier*, que é um profissional dedicado que pode aproveitar a plataforma para dar cursos *online* e vender seus serviços de consultoria. Os outros dois segmentos são *e-commerce*s e vinícolas que podem se beneficiar dos dados coletados pelo Vinho Notas, criando ofertas de *kits* com descontos promocionais, ou ainda ofertas personalizadas, agregando valor às suas plataformas de venda.

A primeira versão do Vinho Notas será gratuita e dedicada apenas aos enófilos. A ideia é criar uma base sólida de consumidores da aplicação, e quando atingir um certo número de usuários, com a ideia validada, ampliar o público alvo apresentando a massa de usuários como um valor agregado à proposta de utilização da plataforma para os demais segmentos.

### Proposta de valor

A proposta de valor, ou seja, o motivo pelo qual os clientes devem escolher o Vinho Notas é facilitar a experiência de degustar vinhos, favorecendo a expansão e a descoberta de conhecimento de maneira organizada e estruturada. Com isso, pretende-se popularizar o acesso aos vinhos, desmistificando e descomplicando seus rituais.

### Canais

Para o Vinho Notas se comunicar e alcançar seus segmentos de clientes e entregar uma proposta de valor, foi desenvolvida uma Aplicação de Web Progressiva (PWA). Além desse canal, a colaboração com parceiros permitirá atingir um número ainda maior de usuários, com um custo reduzido.

### Relacionamento com clientes

O tipo de relacionamento que o Vinho Notas pretende estabelecer com o segmento de clientes deve se dar, inicialmente, através de comunidades de consumidores de vinho na *internet*, fóruns de discussão temáticos e *landing pages* construídas para atrair os novos usuários, além da colaboração com *digital influencers*.

### Fontes de receita

O segmento de clientes enófilo inicialmente não irá gerar receita, tendo em vista que a aplicação será gratuita para eles. Com o passar do tempo, esse segmento receberá a oferta de uma assinatura *premium*, concedendo-lhes acesso a serviços personalizados e fornecidos por meio de parceiros como *sites* de vinícolas, *e-commerces* e *sommeliers*.

Os demais segmentos ao utilizar a plataforma do Vinho Notas deverão pagar uma certa quantia fixa mensal, além de um percentual, a ser negociado, sobre as vendas que forem originadas a partir da plataforma.

### Recursos principais

Para que o Vinho Notas entre em funcionamento é necessário a construção de uma plataforma web, devendo seguir as condições descritas anteriormente no item 4.1.3. Ela deverá ser hospedada em um servidor de nuvem, para permitir acesso de forma controlada e com um custo menor de manutenção.

### Atividades chave

Para que o modelo de negócio funcione é preciso desenvolver a plataforma e hospedá-la em uma nuvem, como por exemplo a *Amazon Web Services* (AWS)[[15]](#footnote-15).

### Parcerias principais

Por se tratar de um serviço hospedado em nuvem, essa solução conta com a AWS como fornecedor. Ao consolidar uma certa quantidade de usuários, deverá contar também com vinícolas nacionais, sites de venda de vinhos, *sommeliers*, clubes de degustadores, associação de vinhos entre outros.

### Estrutura de custos

Os custos envolvidos na operação estão relacionados ao banco de dados que deverá estar na mesma nuvem onde a aplicação será hospedada. Dessa forma, a nuvem também é parte importante da estrutura de custos. Na fase de construção do MVP, deve-se considerar também o tempo gasto para desenvolver a solução.

# Ciclo de desenvolvimento da solução

Para construir o Vinho Notas optou-se pela metodologia ágil por ser uma abordagem ampla e flexível, com foco em valores e princípios, adotando o Scrum como framework para o desenvolvimento do projeto.

Conforme Verheyen (2019, p.20), “Ágil refere-se à mentalidade, às convicções e às preferências expressas no Manifesto para o Desenvolvimento Ágil de Software”. O manifesto citado pelo autor elenca 12 princípios que estão por traz dos valores propostos pela metodologia[[16]](#footnote-16). É uma abordagem de gestão de projetos dividida em fases em que “As equipes seguem um ciclo de planejamento, execução e avaliação” (ATLASSIAN, 2024).

Optou-se por essa abordagem para que se possa construir a solução de forma incremental, permitindo a revisão constante, fazendo melhorias, eliminando certas funcionalidades ao longo do processo e permitindo que se faça modificações visando uma entrega que agrade mais ao cliente, que deve ser o centro das atenções (cliente centrismo).

Para acelerar a entrega da solução, o framework empregado é o Scrum, que “... é um potencializador para entregar melhores produtos e mais rapidamente” (VERHEYEN, 2019, p. 38). O termo foi criado por Hirotaka Takeuchi e Ikujiro Nonaka nos anos 80 do século passado, e até hoje é muito empregado no meio tecnológico em conjunto com os métodos ágeis, o Lean, entre outros.

Obviamente que todos os papéis envolvidos no Scrum tiveram que ser desempenhados por uma única pessoa, que é o autor desse trabalho. Em uma situação normal, haveria um *Product Owner*, um *Scrum Master* e um *Development Team*. Porém, a maioria dos eventos Scrum foram mantidos, como por exemplo o trabalho em *Sprints* definidas com objetivos específicos (e com isso o *Sprint Planning* e o *Daily Scrum*).

## Sprint

A primeira peça de uma abordagem utilizando o *Scrum* é a definição da *Sprint*. Schwaber e Sutherland (2020, p. 8) dizem que as “*Sprints* são o coração do *Scrum*, onde ideias são transformadas em valor”. Esses eventos podem ter duração fixa de um mês ou menos, com metas específicas visando a entrega de valor.

Para a construção do Vinho Notas optou-se por *Sprints* curtas, de no máximo uma semana, com meta bem definida trabalhando-se com pequenas tarefas que serão detalhadas mais adiante.

## Sprint Planning

Antes de iniciar a *Sprint*, fez-se uma análise do que seria desenvolvido durante a semana, construindo um *backlog* capaz de ser entregue dentro daquele prazo, priorizando alguma geração de valor sempre que possível.

Dessa forma, a quantidade de tarefas inseridas em uma *Sprint* deve ser suficiente, mas não excessiva, evitando a sobrecarga de trabalho e dessa maneira a exaustão do desenvolvedor.

## Daily Scrum

Esse evento faz mais sentido quando se trabalha em uma equipe de desenvolvimento, onde cada membro do time traz para os demais as informações sobre o andamento do seu trabalho. Basicamente cada integrante responde à três perguntas, “o que eu fiz ontem?”, “o que farei hoje?” e se “tenho algum bloqueio que coloque em risco o atingimento da meta da *Sprint*?”.

No caso do Vinho Notas, a *Daily Scrum* foi substituída por uma revisão rápida do andamento da *Sprint*, verificando no início do dia o quão próximo se está do atingimento da meta planejada.

## Jira Software

Para ajudar a organizar o trabalho de desenvolvimento, optou-se pelo *Jira Software* da Atlassian[[17]](#footnote-17). A figura 5.1 apresenta a tela inicial do projeto criado dentro da ferramenta. Com ele foi possível organizar e construir um quadro usando o *Scrum* como referência, e com isso ter uma visão mais abrangente do andamento das coisas.

Figura 5.1 – Projeto Vinho Notas

|  |
| --- |
|  |
| Fonte: o autor (2024). |

A figura 5.2 apresenta um pequeno recorte da linha do tempo do projeto, que permite acompanhar o andamento e dessa forma medir o quão próximo do objetivo final o trabalho se encontra.

Figura 5.2 – Linha do temo do projeto Vinho Notas

|  |
| --- |
|  |
| Fonte: o autor (2024). |

Na figura 5.3 tem-se o backlog da Spring 12 do Vinho Notas. Por se tratar de um trabalho de conclusão de curso (TCC), a Sprint 12 está sendo usada para a parte da escrita do TCC.

Figura 5.3 – Backlog da Sprint 12

|  |
| --- |
|  |
| Fonte: o autor (2024). |

E finalmente, o quadro da Sprint 12 pode ser visto na figura 5.4. O quadro é composto pelos campos “*To Do*”, “*In Progress*”, “*Code Review*” e “*Done*”. A primeira se refere ao *backlog* da *Sprint*. Quando uma tarefa é iniciada, ela é movida para a segunda coluna, indicando que agora ela está *in progress*. Se estiver aguardando que o *CI/CD* do GitHub conclua algum processo, o cartão daquela atividade é movido para a terceira coluna, de *code review*. Quando as atividades inerentes ao cartão estiverem concluídas, ele é movido para a última coluna, *done*. Dessa forma, se todas as tarefas estiverem concluídas, diz-se que a *Sprint* atingiu o seu objetivo e assim ela pode ser concluída.

Figura 5.4 – Quadro da Sprint 12

|  |
| --- |
|  |
| Fonte: o autor (2024). |

### Organização das tarefas

Para melhor organizar o trabalho, optou-se pela criação de cartões de tarefas com tipos específicos. O primeiro tipo é o *Epic* que é criado para representar uma parte grande da construção da solução. Ele é quebrado em várias partes menores chamados de *Feature*.

A *Feature* está relacionada com uma funcionalidade ou uma função do produto em desenvolvimento. É nela que o código é produzido, mas conforme a necessidade também pode ser subdividida em partes menores, chamadas de *Subtasks*.

A *Subtask* é a menor parte de uma *Feature* e é criada sempre que for necessário fatiá-las ainda mais porque se tornaram muito grandes, permitindo assim que se trabalhe sempre com tarefas pequenas e rápidas, dando velocidade na construção da solução.

Finalmente, caso seja encontrado algum erro na aplicação resultante de uma funcionalidade que não está se comportando como deveria, então é criada um cartão de *Bug*.

## Versionamento do código

O código produzido foi armazenado em um repositório no GitHub[[18]](#footnote-18), que é uma plataforma baseada em nuvem e que utiliza o Git[[19]](#footnote-19) para controle de versão, gerenciamento e acompanhamento de alterações no código. Dessa forma foi possível criar *branchs*, a partir do código principal, e atuar nessas ramificações de forma segura, isolando cada atividade sem afetar aquilo que já estava pronto.

A *Branch* principal do projeto é a *main*, que é gerada quando se cria o projeto no GitHub. A partir dela foi criado uma cópia nomeada como *develop*. Para cada novo *Epic* uma nova *Branch* é criada a partir da *develop*, que irá se ramificar novamente para cada *Feature*, *Subtask* ou *Bug*.

Quando o código é concluído ele é integrado na *Branch* pai através da abertura de um *pull request*. Para que seja aprovado é preciso rodar um *workflow* criado no GitHub Actions, uma espécie de CI/CD[[20]](#footnote-20) que permite automatizar a compilação, testes, análise de código e empacotamento para publicação. Após concluída a *pipeline* do GitHub Actions, o *pull request* é aceito, o código é integrado e a *Branch* é excluída.

## Etapas do desenvolvimento

Definidas a forma de organizar o desenvolvimento, é hora de definir as etapas para construir o Vinho Notas. A primeira etapa foi a descoberta, o estudo sobre as regras de negócio envolvidas no mundo dos vinhos. Foi preciso entender o que é um vinho, quais os tipos existentes, seus aromas, sabores e cores, além de compreender os rituais envolvidos na degustação, desde a escolha da taça até a temperatura de serviço.

Para a definição da solução optou-se pelo *Lean Incption*, segundo Caroli (2022). Como o resultado deste trabalho, em última análise, é a apresentação de um Produto Mínimo Viável (MVP), então a utilização de um workshop na forma do *Lean Inception* se aplica perfeitamente por apresentar uma maneira estruturada e simples para alcançar o resultado esperado.

As etapas envolvidas foram a definição de visão do produto, o entendimento sobre o que o produto é, não é, faz e não faz. Na sequência foram identificados os objetivos do produto, as personas e a jornada dos usuários. A partir disso foram definidas as funcionalidades, o plano de entrega ou criação do MVP e o Canvas MVP, cujos itens foram apresentados e discutidos no item 4.1.

O *Lean Inception* foi construído dentro do Miro[[21]](#footnote-21), que é uma grande lousa digital, utilizando um *template* do Strategyzer[[22]](#footnote-22) e o resultado pode ser visto na figura 5.5.

|  |
| --- |
| Figura 5.5 – o Business Model Canvas |
|  |
| Fonte: o autor (2024). |

Com o entendimento do que é o negócio e tendo um modelo definido, a próxima etapa foi a definição dos requisitos funcionais e não funcionais[[23]](#footnote-23), criar a estrutura do projeto, ou seja, o repositório no GitHub, a definição do banco de dados, da tecnologia a ser empregada no *backend* e *frontend* e a arquitetura da solução.

A figura 5.6 mostra a imagem inicial do repositório criado. Para os demais itens ver seção 7 deste trabalho que aborta a arquitetura da solução.

|  |
| --- |
| Figura 5.6 – Criação do repositório no GitHub |
|  |
| Fonte: o autor (2024). |

# Mockup da proposta de solução

A versão inicial do Vinho Notas deve ser uma aplicação simples limpa e intuitiva. A ideia inicial é ter um MVP na sua versão 1.0, evoluindo para algo mais estruturado e completo ao longo do tempo.

Para isso, o primeiro passo foi criar um fluxo simulando a interação do usuário com o sistema. A figura 6.1 apresenta o início da jornada do usuário. Com o desenho, alguns comentários ou anotações são feitas com o intuito de ir conversando com a interface e dessa forma poder desenhar uma tela mais próxima possível do que se pretende construir.

|  |
| --- |
| Figura 6.1 – Início da jornada do usuário |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

A partir do desenho inicial, a tela de login é projetada como pode ser visto na figura 6.2. Neste trabalho optou-se por utilizar o Excalidraw[[24]](#footnote-24) para fazer os desenhos de fluxo e tela de sistema. Utilizar o Figma[[25]](#footnote-25) seria uma escolha natural, mas como a ideia é deixar as coisas o mais simples possível e ganhar tempo para criar uma solução MVP em um tempo curto, percebeu-se que o Excalidraw supre todas as necessidades nesse momento.

|  |
| --- |
| Figura 6.2 – Tela de login |
| Uma imagem contendo Interface gráfica do usuário  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

Após efetuar o login, caso cumpra as condições necessárias, como por exemplo ser maior de 18 anos e estar cadastrado previamente, então o usuário é direcionado à tela inicial do sistema, cujo desenho pode ser visto na figura 6.3.

|  |
| --- |
| Figura 6.3 – Tela inicial |
| Interface gráfica do usuário, Texto, Email  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

## Menu Cadastro

A tela inicial deve ser simples, com um menu na parte de cima da tela, com quatro opções apenas, “Home”, “Cadastro”, “Vinho” e “Degustação”. Dessa forma, quando o usuário selecionar o menu Cadastro (ver figura 6.4), lhe será apresentado três opções: usuário, pessoa e endereço.

|  |
| --- |
| Figura 6.4 – Menu Cadastro |
| Texto  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

Para cadastrar o usuário, primeiro deve-se cadastrar uma pessoa que concomitantemente adiciona seu endereço. Pensando em um fluxo, o usuário a partir do menu cadastro pode visualizar uma lista dos endereços como ilustrado na figura 6.5.

|  |
| --- |
| Figura 6.5 – Fluxo de consulta de endereços |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

Com esse desenho foi possível pensar em uma tela listando os endereços onde o usuário possa filtrar por alguma palavra-chave, selecionar um ou mais endereços e fazer alteração ou exclusão do registro. A tela final deve ficar parecida com a figura 6.6.

|  |
| --- |
| Figura 6.6 – Tela de listagem de endereços |
| Tabela  Descrição gerada automaticamente com confiança média |
| Fonte: o autor (2024). |

Com a primeira tela de listagem desenhada, optou-se por padronizar as demais, pensando em um comportamento semelhante para o usuário. Dito de outra forma, ele deve ser capaz de listar alguma coisa na tela, editar e excluir, mas também deve conseguir exportar o conteúdo para um arquivo no formato CSV[[26]](#footnote-26). Isso pode ser visto na figura 6.7 que apresenta a tela de listagem de pessoas.

|  |
| --- |
| Figura 6.7 – Tela de listagem de pessoas |
| Interface gráfica do usuário, Texto, Email  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

Essa tela adiciona um botão “Novo” que o usuário utilizará para cadastrar uma nova pessoa, informando nome completo, o CPF e a data de nascimento. Essas informações são usadas para validar se a pessoa já está cadastrada e se tem mais de 18 anos. As linhas são expansivas, mostrando o endereço da pessoa, que é persistido junto com os demais dados de cadastro.

A figura 6.8 apresenta a listagem dos usuários, cuja tela tem a mesma estrutura das demais, mantendo assim o padrão estabelecido anteriormente. Quando a pessoa cadastrar o usuário ela deve informar o nome, o perfil (na versão inicial apenas Enófilo), e-mail e uma senha de 4 números que não é apresentada na tela de listagem.

|  |
| --- |
| Figura 6.8 – Tela de listagem de usuários |
| Texto  Descrição gerada automaticamente com confiança baixa |
| Fonte: o autor (2024). |

## Menu Vinho

O menu Vinho terá apenas duas opções. Uma tela com a listagem dos vinhos e uma tela com a listagem de avaliações feitas pelo usuário, figura 6.9.

|  |
| --- |
| Figura 6.9 – Tela de vinho |
|  |
| Fonte: o autor (2024). |

A figura 6.10 mostra o fluxo do usuário ao listar os vinhos na tela. A partir do momento que se tem um vinho cadastrado no sistema, é possível avaliá-lo. Por isso pensou-se em adicionar um botão a mais, justamente para fazer a avaliação desse vinho. Porém, deve-se lembrar que não se trata de preencher uma ficha de degustação, mas sim pensando em um enófilo que está abrindo uma garrafa no final do dia, sozinho ou com seu/sua parceiro(a) e decidiu registrar uma avaliação para aquela garrafa.

|  |
| --- |
| Figura 6.10 – Fluxo de listar vinhos |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

A tela deve ficar parecida com a figura 6.11. Cadastrar um vinho requer preencher muitos campos e por isso é importante que o usuário possa mostrar apenas os campos que lhe interessam na sua tela. As demais funcionalidades são as mesmas, garantindo assim a manutenção da padronização.

|  |
| --- |
| Figura 6.11 – Tela de lista de vinhos |
| Interface gráfica do usuário, Texto, Email  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

O usuário pode listar as avaliações ao selecionar o menu Vinho e depois Avaliações. Uma proposta de tela pode ser visualizada na figura 6.12, com as opções de editar, excluir e exportar para um arquivo no formato .CSV.

|  |
| --- |
| Figura 6.12 – Tela de avaliações |
| Texto  Descrição gerada automaticamente com confiança média |
| Fonte: o autor (2024). |

## Menu Degustação

O menu Degustação vai apresentar apenas as fichas de degustação na forma de uma listagem. Figura 6.13 mostra o menu bem simples, com apenas uma opção.

|  |
| --- |
| Figura 6.13 – Menu fichas de degustação |
| Texto, Carta  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

Ao selecionar a opção “Fichas de degustação”, o usuário é direcionado à tela da figura 6.14. Nela o usuário pode visualizar todas as fichas de degustação que ele já preencheu de maneira formal. A degustação forma é aquela feita seguindo certas regras, ou cerimônias, onde o enófilo escolhe um ou mais vinhos e passa por cada uma das quatro etapas, inspeção visual, olfativa, gustativa e avaliação geral. Essa jornada é ilustrada na figura 6.15.

|  |
| --- |
| Figura 6.14 – Tela lista de degustações |
|  |
| Fonte: o autor (2024). |

|  |
| --- |
| Figura 6.15 – Fluxo de degustação |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

## Menu harmonização

O menu de harmonização surgiu da ideia de incorporar uma inteligência artificial generativa ao projeto e dessa forma construir uma espécie de *sommelier* virtual. Sua função consiste em passar informação sobre um determinado vinho ao usuário, sugerir alimentos que possam harmonizar com o vinho e finalmente, sugerir uma refeição completa com entrada, prato principal e sobremesa que harmonize com o vinho.

A figura 6.16 apresenta uma sugestão de tela de harmonização tal como deve ficar no final.

|  |
| --- |
| Figura 6.16 – Tela de sugestão de harmonização |
| Interface gráfica do usuário, Texto, Aplicativo, Email  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

# Arquitetura de Software

Nesta seção pretende-se apresentar o projeto de arquitetura de software desenhado para criar o Vinho Notas. Para Sommerville (2018, p. 147), “O projeto de arquitetura visa compreender como um sistema de software deve ser organizado e projetar a estrutura geral desse sistema”.

A arquitetura escolhida é a Modelo-Visão-Controlador (MVC), separando as camadas de conteúdo e lógica de processamento no modelo, enquanto a visão se encarrega da interface com o usuário e o controlador gerencia o acesso ao modelo e à visão (PRESSMAN E MAXIM, 2021).

Essa arquitetura chamada MVC foi inicialmente chamada de Thing-Model-View-Editor (REENSKAUG, 1979a) e posteriormente para Models-View-Controllers (REENSKAUG, 1979b). Para conhecer a história com mais detalhes ver Reenskaug (2024).

## Banco de dados

É fato que a humanidade vive em uma era guiada por dados gerados da forma mais diversa possível, armazenados em bancos que “... facilitam a integração de diferentes sistemas e aplicativos, permitindo que as informações fluam de maneira eficiente e eficaz entre eles” (DATA SCIENCE ACADAMY, 2023). Para Amadeu (2014, p. 3), “Um banco de dados é uma coleção de dados relacionados”. A autora define os dados como sendo informações que podem ser registradas em algum lugar.

O armazenamento e o gerenciamento dos dados são feitos por um Sistema de Gerenciamento de Banco de Dados (SGBD), que no caso do Vinho Notas optou-se pelo PostgreSQL[[27]](#footnote-27). A escolha se deu por se tratar de um SGBD de código aberto amplamente utilizado para armazenar e gerenciar dados estruturados, além de ser compatível com SQL e JSON (POSTGRESQL, 2014).

Conforme documentação da Google Cloud, “O PostgreSQL também oferece suporte a propriedades ACID (atomicidade, consistência, isolamento e durabilidade) e é altamente tolerante a falhas graças à geração de registros de gravação antecipada” (GOOGLE CLOUD, 2024).

A propriedade ACID é importante para garantir a integridade durante o processamento das transações relacionadas a banco de dados, principalmente em banco de dados relacionais (SINGH, 2024). As transações são uma sequência de uma ou mais operações SQL executadas como uma única unidade atômica (SIMONYAN, 2024).

O teorema CAP afirma que qualquer sistema de dados compartilhados em rede pode ter apenas duas das três propriedades desejáveis (BREWER, 2012). No caso do PostgreSQL, essas propriedades são a consistência e a disponibilidade (SINGH, 2024).

### Diagrama de esquema para o banco de dados

A modelagem de dados consiste em elaborar uma “...coleção de conceitos que podem ser usados para descrever a estrutura de um banco de dados” (AMADEU, 2014, p. 22). Já o modelo conceitual, “... representa as informações que existem no contexto do negócio, com maior foco nos processos” (AMADEU, 2014, p. 35).

|  |
| --- |
| Figura 7.1 – Diagrama de esquema para o banco de dados de cadastro |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

Como será visto adiante, o Vinho Notas trabalha com uma arquitetura de micros serviços onde cada serviço possui seu próprio banco de dados e é responsável por uma ou mais APIs (acrônimo para *Application Programming Interface*). Para uma visão completa sobre APIs ver Louzada, Carvalho e Laranja (2024).

A figura 7.1 apresenta o diagrama de esquema para o banco de dados de cadastro. Esse serviço é formado por várias tabelas, e cada tabela é alimentada por uma API REST[[28]](#footnote-28). Isso significa que quando o usuário faz uma operação de cadastramento ou consulta no *frontend*, é enviado uma requisição via HTTP utilizando o formato JSON (*Javascript Object Notation*) utilizando uma solicitação do tipo POST, GET, DELETE ou PUT[[29]](#footnote-29) (IBM, 2024).

|  |
| --- |
| Figura 7.2 – Diagrama de esquema para o banco de dados de vinho |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

É no serviço de cadastro e consequentemente no seu banco de dados que é feito o registro da pessoa, seu endereço e finalmente, o usuário. As tabelas “tbstate” e “tbcountry” são responsáveis pelas informações de estados brasileiros e países do mundo todo. Essas APIs não estão disponíveis para os usuários do Vinho Notas, porém estão para integração com outros sistemas, como será detalhado mais adiante.

A figura 7.2 apresenta o diagrama de esquema para o banco de dados do serviço de vinho que tem por responsabilidade única o cadastramento do vinho

O diagrama de esquema para o banco de dados de avaliação pode ser visto na figura 7.3. Com o vinho cadastrado o usuário pode escolher um dos vinhos e fazer uma avaliação, estabelecendo uma nota e destacando outras informações conforme será detalhado adiante.

|  |
| --- |
| Figura 7.3 – Diagrama de esquema para o banco de dados de avaliação |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

A figura 7.4 apresenta o diagrama de esquema para o banco de dados do serviço de degustação. Para registrar uma degustação o usuário utiliza o serviço homônimo que emprega a tabela apresentada naquela figura.

A diferença entre uma avaliação e uma degustação é que no primeiro caso o usuário faz uma análise pessoal mais rápida, anotando as suas percepções de forma particular. Por outro lado, na degustação ele tem a opção de escolher entre cada aspecto analisado, onde o sistema o guia no processo de degustar o vinho.

|  |
| --- |
| Figura 7.4 – Diagrama de esquema para o banco de dados de degustação |
| Diagrama  Descrição gerada automaticamente com confiança baixa |
| Fonte: o autor (2024). |

Os diagramas entidade relacionamento podem ser vistos nas figuras a seguir. O primeiro deles é o de cadastro, na figura 7.5. Nesse banco trabalha-se com 5 tabelas, sendo uma para pessoa, que depende da tabela de endereços e é dependência da tabela de usuários. A tabela de endereços, por sua vez, depende das tabelas de estado e país que também apresentam uma relação de dependência.

A figura 7.6 apresenta o diagrama entidade relacionamento para o vinho e para degustação. Esses bancos de dados têm apenas uma tabela, sem relacionamento entre si ou com outras de micro serviços diferentes. Dessa forma é possível garantir o isolamento das transações relacionadas à entidade de registro de vinhos e degustações no sistema. Essa prática foi adotada para todos os bancos de dados e seus respectivos micros serviços evitando assim o acoplamento entre eles.

Figura 7.5 - Diagrama Entidade Relacionamento do cadastro

|  |
| --- |
| Diagrama  Descrição gerada automaticamente com confiança baixa |
| Fonte: o autor (2024) |

A mesma abordagem pode ser vista na figura 7.7, que apresenta o diagrama para o micro serviço de avaliação. Isolando dessa forma cada API em um micro serviço isolado com um banco de dados independente, abre espaço para se trabalhar inclusive com bancos de dados diferentes. Por exemplo, poderia se trabalhar com o PostgreSQL em um micro serviço enquanto outro poderia ter o SQL Server como banco.

Figura 7.6 - Diagrama Entidade Relacionamento do vinho e degustação

|  |  |
| --- | --- |
| Texto  Descrição gerada automaticamente com confiança baixa | Texto  Descrição gerada automaticamente com confiança média |
| Fonte: o autor (2024) | |

Figura 7.7 - Diagrama Entidade Relacionamento para avaliação

|  |
| --- |
| Tabela  Descrição gerada automaticamente |
| Fonte: o autor (2024) |

## O Back End

O *back end*[[30]](#footnote-30) da aplicação foi construído utilizando a linguagem Java[[31]](#footnote-31), com JDK 17[[32]](#footnote-32)[[33]](#footnote-33), utilizando o Maven[[34]](#footnote-34) como compilador e gerenciador de dependências e o Spring Boot[[35]](#footnote-35) como framework para simplificar o desenvolvimento.

Quando o usuário faz uma solicitação na tela da aplicação, dentro do Vinho Notas, ele envia uma requisição para o *back end* que fará o tratamento necessário e entregará uma resposta àquela solicitação.

As requisições enviadas indicam uma ação através de um verbo HTTP como por exemplo o método GET que retorna os dados solicitados, POST que é utilizado para criar um recurso, o PUT para fazer alterações em algo que está persistido no banco de dados e DELETE para apagar uma informação do banco[[36]](#footnote-36).

A figura 7.8 apresenta o desenho de como esse fluxo deve funcionar. A partir da interação do usuário na tela do sistema, ele envia uma requisição HTTP para um *Back end For Front end* (BFF)[[37]](#footnote-37). Este, por sua vez, verifica a quem se destina a solicitação e dispara a requisição para o *back end*. Se for necessária alguma intervenção junto ao banco de dados, o *back end* se encarrega de fazer isso, e após todo o processamento necessário, devolve uma resposta ao BFF que retransmite para a tela do sistema.

|  |
| --- |
| Figura 7.8 – Diagrama do fluxo das requisições |
|  |
| Fonte: o autor (2024). |

Cada *back end*, inclusive o BFF, possui uma arquitetura separada por camadas, atendendo aos princípios do *Clean Architecture* preconizados por Martin (2012). O desenho dessa arquitetura pode ser visto na figura 7.9, onde cada camada é independente da outra.

|  |
| --- |
| Figura 7.9 – Desenho da arquitetura em camadas |
|  |
| Fonte: o autor (2024). |

A primeira camada é a externa, de onde vem as requisições ou para onde irão caso seja necessário. Tudo passa pelo BFF que posteriormente chama os controladores, que por sua vez direciona as requisições à camada de serviços (responsável pela parte lógica da aplicação). A principal é a central, que envolve as entidades, que encapsulam as regras de negócio da aplicação.

## O front end

Optou-se pelo React[[38]](#footnote-38) para criar a interface do programa, que é uma biblioteca JavaScript[[39]](#footnote-39) de código aberto para renderizar interfaces, lançado em 2013 pelo Facebook[[40]](#footnote-40). A escolha se deu pela sua praticidade em criar aplicativos de página única (SPA[[41]](#footnote-41)), atualizando o conteúdo de forma dinâmica sem a necessidade de recarregar a página inteira.

Para se obter uma construção mais ágil e produtiva escolheu-se o Vite[[42]](#footnote-42), usando o JavaScript como linguagem de programação, e garantindo assim que a renderização ocorra do lado do cliente, e não do servidor.

A arquitetura do *front end* é semelhante àquela utilizada no *back end*, no que diz respeito a divisão de responsabilidade, no entanto, trabalhando com um projeto único. Essa separação é baseada nos princípios do SOLID[[43]](#footnote-43), e feita com o uso de componentes, cada qual com uma responsabilidade específica, escritos em linguagem JavaScript com a sintaxe JSX[[44]](#footnote-44), que permite escrever marcação HTML dentro de código JavaScript.

Para evitar a duplicação de código o emprego de componentes é muito importante, porque melhora a distribuição, sustentação, isolamento compartilhado e reuso do código (BERTOLI, 2017).

Quanto ao estilo, CSS, optou-se por duas opções *open source* baseadas em Bootstrap[[45]](#footnote-45), criadas para o React. São elas o PrimeReact[[46]](#footnote-46) e o React-Bootstrap[[47]](#footnote-47). Essa escolha ajudou com a construção dos componentes e a responsividade da aplicação, segundo as especificações do W3C[[48]](#footnote-48).

# Validação da solução

A qualidade em um software se refere a um conjunto de características que garantem que ele atende às necessidades dos usuários e funcione de maneira confiável, eficiente e segura. Ela envolve diversos aspectos, como funcionalidade, usabilidade e confiabilidade, entre outros.

A funcionalidade se refere ao fato de que o software deve atender aos requisitos especificados e realizar as tarefas para as quais foi projetado. A usabilidade quer dizer que ele deve ser fácil de usar, aprender e navegar, com interfaces intuitivas e amigáveis. Também deve ser confiável, ou seja, livre de erros e falhas, e funcionar de maneira consistente em diferentes cenários.

Para garantir a qualidade dessa solução, utilizou-se a técnica do TDD[[49]](#footnote-49), ou seja, do desenvolvimento guiado por testes. Para isso seguiu-se os passos descritos em Beck (2010, p. 31):

1. Escreva um teste.
2. Faça-o rodar.
3. Refatore.

A figura 8.1 apresenta uma versão visual dessa técnica adaptada de Aniche (2017, p.31). O primeiro passo é escrever o teste mais simples possível, executar para ver ele falhar. Depois se corrige a falha implementando a solução e executa o teste novamente, que deverá passar com sucesso. Nesse momento deve-se refatorar o código eliminando duplicidades seguindo os princípios das boas práticas de programação.

Em termos práticos, todas as classes que possuem algum método que é invocado são testadas. Primeiro cria a classe, depois escreve a assinatura do método que será implementado e na sequência a classe de testes, utilizando os frameworks descritos adiante e, finalmente, constrói o teste para aquele método que será implementado.

Como o método não faz nada ainda, pois tem apenas: a assinatura, ou seja, um modificador de acesso público ou privado, algo que será retornado (objeto ou nada), o nome do método e os parâmetros (ou nenhum parâmetro) e um retorno nulo caso a assinatura exija isso. Por isso, ao ser executado o teste deve falhar. Com isso, implementa-se a funcionalidade e testa-se novamente até passar com sucesso.

|  |
| --- |
| Figura 8.1 – TDD |
| Diagrama  Descrição gerada automaticamente |
| Fonte: o autor (2024). |

Para testar o *back end* utilizou-se um conjunto de frameworks, sendo o principal deles o JUnit[[50]](#footnote-50), que é gratuito e de código aberto, amplamente utilizado para testes automatizados. Ele trabalha em conjunto com o Mockito[[51]](#footnote-51), que é útil em testes unitários e de integração, criando objetos falsos (mock), simulando o comportamento de classes reais.

As classes de controladores foram testadas com o uso do WebMvcTest[[52]](#footnote-52), do Spring Boot, que permite testar as requisições HTTP, métodos e *status* de retorno, a desserialização do objeto de entrada e serialização do objeto de saída, e exceções entre outras funcionalidades.

Para as interfaces de repositório de dados, que estendem a classe JpaRepository, utilizou-se o DataJpaTest[[53]](#footnote-53), permitindo assim verificar as interações com o banco de dados. Optou-se em um banco em memória (neste caso o H2[[54]](#footnote-54)) para fins de testar os processos de criação, consulta, edição e exclusão das entidades.

Para o *fron tend* optou-se pelo Jest[[55]](#footnote-55), que é amplamente utilizado para testar códigos JavaScript em projetos usando o React.

Em termos de cobertura de testes, a ideia é que se tenha no mínimo 80% dos métodos e das classes testadas. Para obter essa medida utilizou-se o JaCoCo[[56]](#footnote-56) (*Java Code Coverage*) que é uma biblioteca de cobertura de código gratuita para projetos Java. É muito útil para a medir o quanto do código-fonte Java foi testado, fornecendo relatórios detalhados de cobertura de código, como pode ser visto na figura 8.2, que mostra como exemplo o relatório da cobertura de testes unitários do BFF.

|  |
| --- |
| Figura 8.2 – Relatório do JaCoCo para cobertura de testes unitários do BFF |
|  |
| Fonte: o autor (2024). |

Pelo relatório é possível observar qual o percentual de cobertura de testes em cada camada, permitindo assim focar naquelas em que o indicador for menor em comparação às outras, garantindo a qualidade do código escrito.

# Registros das evidências do projeto

O repositório da solução desenvolvida encontra-se no GitHub, no endereço <https://github.com/vinho-notas/vinho-notas-app>, onde é possível encontrar o *backend* dos micro serviços e o *frontend*. Cada um deles possui um arquivo README.md com a documentação das apis.

O quadro 9.1 apresenta uma compilação de cada micro serviço e seu respectivo endereço de repositório no GitHub.

|  |  |
| --- | --- |
| Quadro 9.1 – Micro serviços e seus endereços | |
| **Microsserviço** | **Endereço** |
| Cadastro | <https://github.com/vinho-notas/vinho-notas-app/tree/main/cadastro> |
| Vinho | <https://github.com/vinho-notas/vinho-notas-app/tree/main/vinho> |
| Avaliação | <https://github.com/vinho-notas/vinho-notas-app/tree/main/avaliacao> |
| Degustação | <https://github.com/vinho-notas/vinho-notas-app/tree/main/degustacao> |
| Harmonização | <https://github.com/vinho-notas/vinho-notas-app/tree/main/harmonizacao> |
| BFF | <https://github.com/vinho-notas/vinho-notas-app/tree/main/bff> |
| *Frontend* | <https://github.com/vinho-notas/vinho-notas-app/tree/main/frontend> |
| Fonte: o autor (2024). | |

O quadro 9.2 apresenta os *endpoints* da API de cadastro. Todos eles partem de uma URL base para rodar localmente, acrescentando o caminho “/api/v1” mais um recurso, que pode ser observado no quadro, indicando tratar-se de uma API Restful.

Quadro 9.2 - Endpoints da API de cadastro

|  |  |  |
| --- | --- | --- |
| **Método HTTP** | **Descrição** | **Funcionalidade** |
| **Países** | **Operações relacionadas a países** | |
| GET | /api/v1/countries/{id} | Retorna um país pelo seu id. |
| PUT | /api/v1/countries/{id} | Atualiza um país. |
| DELETE | /api/v1/countries/{id} | Deleta um país. |
| GET | /api/v1/countries | Retorna todos os países. |
| POST | /api/v1/countries | Cria um país. |
| GET | /api/v1/countries/name/{name} | Retorna um país pelo seu nome. |
| GET | /api/v1/countries/continent/{continent} | Retorna uma lista de países pelo seu continente. |
| **Estados** | **Operações relacionadas a estados** | |
| GET | /api/v1/states/{id} | Retorna um estado pelo seu id. |
| PUT | /api/v1/states/{id} | Atualiza um estado. |
| DELETE | /api/v1/states/{id} | Deleta um estado. |
| GET | /api/v1/states | Retorna todos os estados. |
| POST | /api/v1/states | Cria um estado. |
| GET | /api/v1/states/uf/{uf} | Retorna um estado pela sua UF. |
| GET | /api/v1/states/name/{name} | Retorna um estado pelo seu nome. |
| **Endereços** | **Operações relacionadas a endereços** | |
| GET | /api/v1/address/{id} | Retorna um endereço pelo seu id. |
| PUT | /api/v1/address/{id} | Atualiza um endereço. |
| DELETE | /api/v1/address/{id} | Deleta um endereço. |
| GET | /api/v1/address | Retorna todos os endereços. |
| POST | /api/v1/address | Cria um endereço. |
| **Pessoas** | **Operações relacionadas a pessoas** | |
| GET | /api/v1/persons/{id} | Retorna uma pessoa pelo seu id. |
| PUT | /api/v1/persons/{id} | Atualiza uma pessoa. |
| DELETE | /api/v1/persons/{id} | Deleta uma pessoa. |
| GET | /api/v1/persons | Retorna todas as pessoas. |
| POST | /api/v1/persons | Cria uma pessoa. |
| GET | /api/v1/persons/name/{name} | Retorna uma pessoa pelo seu nome. |
| **Usuários** | **Operações relacionadas a usuários** | |
| GET | /api/v1/users/{id} | Retorna um usuário pelo seu id. |
| PUT | /api/v1/users/{id} | Atualiza um usuário. |
| DELETE | /api/v1/users/{id} | Deleta um usuário. |
| GET | /api/v1/users | Retorna todos os usuários. |
| POST | /api/v1/users | Cria um usuário. |
| GET | /api/v1/users/name/{name} | Retorna um usuário pelo seu nome. |
| Fonte: O autor (2024) | | |

URL significa *"Uniform Resource Locator"* ou em uma tradução livre, Localizador Uniforme de Recursos. É uma sequência de caracteres que especifica onde um recurso pode ser encontrado na Internet e o mecanismo usado para acessá-lo. Por exemplo, ao rodar localmente uma aplicação, sua URL base pode ser http://localhost:8080, indicando que o esquema utiliza o protocolo “HTTP” para navegação, o domínio é “localhost”, indicando que se trata da própria máquina que está sendo executado e “8080” é o número da porta que está sendo utilizada.

Nem todos os *endpoints* da API de cadastro estão disponíveis para o usuário. Optou-se por expor apenas aqueles que se referem a cadastrar uma pessoa, seu endereço e o usuário. Além disso, é possível alterar os dados, excluir ou listar todos. A ideia é que as APIs sejam agnósticas no sentido de permitir que sejam consumidas por qualquer serviço que utilize esses *endpoints*, e que seja independente de linguagem de programação.

O quadro 9.3 apresenta os *endpoints* da API de vinho que segue o mesmo padrão visto anteriormente, ou seja, é uma API agnóstica cujos *endpoints* expostos permitem ao usuário realizar operações relacionadas ao registro, listagem, edição e exclusão de um vinho no banco de dados do sistema.

Quadro 9.3 - Endpoints da API de vinho

|  |  |  |
| --- | --- | --- |
| **Método HTTP** | **Descrição** | **Funcionalidade** |
| **Vinhos** | **Operações relacionadas a vinhos** | |
| GET | /api/v1/wines/{id} | Retorna um vinho pelo id. |
| PUT | /api/v1/wines/{id} | Atualiza um vinho pelo id. |
| DELETE | /api/v1/wines/{id} | Deleta um vinho pelo id. |
| GET | /api/v1/wines | Retorna todos os vinhos. |
| POST | /api/v1/wines | Cria um vinho. |
| Fonte: O autor (2024) | | |

Os *endpoints* da API de degustação podem ser vistos no quadro 9.4. Pode-se perceber que há um padrão entre as APIs, que foram construídas para serem CRUDs completos, mesmo que nem todos os recursos estejam disponíveis para o usuário.

A expressão CRUD é um acrônimo para *Create, Read, Update* e *Delete*, que são métodos obtidos através de requisições HTTP do tipo POST, GET, PUT e DELETE, respectivamente.

Quadro 9.4 - Endpoints da API de degustação

|  |  |  |
| --- | --- | --- |
| **Método HTTP** | **Descrição** | **Funcionalidade** |
| Tasting Card | Operações relacionadas a ficha de degustação | |
| POST | /api/v1/tasting-card | Cadastrar ficha de degustação. |
| GET | /api/v1/tasting-card | Retorna uma lista com todas as fichas de degustação cadastradas. |
| GET | /api/v1/tasting-card/{id} | Retorna uma ficha de degustação cadastrada pelo id. |
| PUT | /api/v1/tasting-card/{id} | Atualiza uma ficha de degustação cadastrada pelo id. |
| DELETE | /api/v1/tasting-card/{id} | Deleta uma ficha de degustação cadastrada pelo id. |
| Fonte: O autor (2024) | | |

Para a API de avaliação, os *endpoints* estão no quadro 9.5, seguindo o mesmo padrão das demais APIs. Por outro lado, o quadro 9.6 apresenta os *endpoints* da API de harmonização. Essa API difere das demais por apresentar apenas métodos HTTP do tipo GET. Isso decorre do fato de que ela consome uma API externa, do Chat GPT[[57]](#footnote-57), que é possível ser feito adicionando uma dependência para o Spring AI[[58]](#footnote-58).

Quadro 9.5 - Endpoints da API de avaliação

|  |  |  |
| --- | --- | --- |
| **Métodos HTTP** | **Descrição** | **Funcionalidade** |
| **Point Scales** | **Operações relacionadas a avaliação de vinhos** | |
| GET | /api/v1/point-scales/{id} | Retorna uma avaliação de vinho pelo id. |
| PUT | /api/v1/point-scales/{id} | Atualiza uma avaliação de vinho pelo id. |
| DELETE | /api/v1/point-scales/{id} | Deleta uma avaliação de vinho pelo id. |
| GET | /api/v1/point-scales | Retorna todas as avaliações de vinho. |
| POST | /api/v1/point-scales | Cria uma avaliação de vinho. |
| Fonte: O autor (2024) | | |

O Spring AI é um *framework* utilizado para a criação da API de harmonização usando inteligência artificial. Com ele é possível se conectar a vários provedores de modelos de inteligência artificial populares, como OpenAI, Microsoft, Amazon, Google e Hugging Face.

Ainda que esteja em pleno desenvolvimento, já é possível utilizar vários recursos como o prompt de comandos onde se passa um texto com os comandos necessários para executar uma ação e obtendo o retorno desejado.

Quadro 9.6 - Endpoints da API de harmonização

|  |  |  |
| --- | --- | --- |
| **Métodos HTTP** | **Descrição** | **Funcionalidade** |
| Pairing | Operações relacionadas a harmonização de vinhos | | |
|  |  |  |
| GET | /api/v1/pairing/pairings | Retorna a harmonização do vinho |
| GET | /api/v1/pairing/menu | Retorna a harmonização do vinho com o menu |
| GET | /api/v1/pairing/information | Retorna informações sobre o vinho |
| Fonte: O autor (2024) | | |

A última API é o BFF, *Backend For Frontend*, criado como um servidor intermediário entre o *frontend* e o *backend* da aplicação. Ele possui os mesmos *endpoints* que as demais APIS, com algumas exceções.

No BFF os métodos HTTP de criar, editar e excluir país e/ou estado não está disponível. O que muda em relação às APIs individualmente é apenas a URL base, por causa da porta. Dessa forma, garante-se que o *frontend* chame o BFF que por sua vez deve direcionar a solicitação ao serviço responsável que retornará uma resposta. Garante-se dessa forma o desacoplamento e o isolamento dos serviços.

# Considerações finais e expectativas

A jornada no curso de Desenvolvimento Full Stack foi marcada por uma constante evolução, desde as primeiras aulas até o desenvolvimento das habilidades necessárias para construir aplicações web completas. As disciplinas percorridas proporcionaram uma base sólida e abrangente, abarcando tanto os aspectos técnicos quanto as metodologias ágeis e inovadoras que permeiam o mercado de software.

No início do curso, foram apresentados os fundamentos da computação e algoritmos, desvendando os pilares que sustentam a lógica por trás da programação. Em seguida, a Programação Orientada a Objetos trouxe as ferramentas essenciais para organizar e estruturar o código de forma eficiente e reutilizável.

Ao longo do curso, foi possível conhecer e criar uma base em linguagens de front end como HTML, CSS e JavaScript, aprimorando a capacidade de criar interfaces web intuitivas e responsivas. As disciplinas de Arquitetura Client-Side e Design de UI guiaram na construção de interfaces com foco na experiência do usuário.

No back-end, não foram exploradas tantas linguagens quanto no front end, mas foi possível trabalhar um pouco com Python (na disciplina de banco de dados relacional) e Java, aprendendo a desenvolver e consumir APIs REST e SOAP. O estudo de Arquitetura Server-Side proporcionou uma compreensão da organização e do funcionamento dos sistemas web do lado do servidor.

As disciplinas de DevOps Básico e Segurança de Software permitiram conhecer as ferramentas necessárias para garantir a confiabilidade, segurança e escalabilidade das aplicações. Houve um aprofundamento importante em metodologias ágeis como Lean Startup e Mentalidade Ágil - LEAN, aprendendo a trabalhar de forma colaborativa e adaptável para entregar resultados de forma rápida e eficiente.

A jornada também proporcionou a exploração de áreas como Inovação e Transformação Digital, compreendendo como as tecnologias disruptivas estão moldando o futuro do mercado. O estudo de Micro Serviços mostrou como construir aplicações mais modulares, escaláveis e resilientes.

Em Banco de Dados Relacional e NoSQL, foram vistos os conceitos e ferramentas para gerenciar e armazenar dados de forma eficiente. As disciplinas de Design Thinking e Empreendedorismo e Novos Modelos de Negócios mostraram uma visão holística do desenvolvimento de software, considerando a viabilidade e o impacto de produtos digitais no mercado.

Ao final do curso, com este TCC, foi possível de aplicar os conhecimentos em um projeto prático de desenvolvimento de uma aplicação web. Essa experiência prática permitiu integrar os conceitos aprendidos e desenvolver uma solução completa, desde a concepção até a implementação e o teste.

Com a formação em Desenvolvimento Full Stack, as expectativas para o futuro são promissoras. É importante estar preparado para os novos desafios do mercado de trabalho, principalmente considerando as mudanças recentes proporcionadas pelas IAs Generativas. Portanto, essa jornada em Desenvolvimento Full Stack foi importante para a evolução na carreira, gerando boas expectativas para o futuro. A ideia a partir de agora é evoluir ainda mais, principalmente no front end.

# Referências

ATLASSIAN. O Coach Agile: manual rápido da Atlassian para desenvolvimento com agilidade. Disponível em: https://www.atlassian.com/br/agile. Acesso em: 27 mar. 2024.

AMADEU, Claudia Vicci (org.). Banco de dados. São Paulo, SP: Pearson, 2014. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 01 abr. 2024.

ANICHE, Mauricio. Test-Driven Development: teste e design no mundo real. São Paulo: Casa do Código, 2017. 194 p.

BARELLI, Suzana. Como serão os brindes de 2024? Estadão. São Paulo, 02 jan. 2024. Disponível em: https://www.estadao.com.br/paladar/le-vin-filosofia/como-serao-os-brindes-de-2024/. Acesso em: 26 mar. 2024.

BECK, Kent. TDD desenvolvimento guiado por testes. Porto Alegre: Bookman, 2010. 240 p.

BERTOLI, Michele. React design patterns and best practices: Build modular applications that are easy to scale using the most powerful components and design patterns that React can offer you right now. Birmingham, UK: Packt Publishing Ltd, 2017.

BREWER, Eric. CAP Twelve Years Later: how the "rules" have changed. Infoq. [S.I]. 30 maio 2012. Disponível em: https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed/. Acesso em: 01 abr. 2024.

CAROLI, Paulo. Direto ao ponto: criando produtos de forma enxuta. São Paulo: Casa do Código, 2015.

CAROLI, Paulo. Lean Inception: saiba como alinhar pessoas e construir o produto certo. 2022. Disponível em: https://caroli.org/lean-inception-3/. Acesso em: 28 mar. 2024.

CAVIQUIOLI, Nara. Aplicativo para catalogar vinhos, você já baixou? Nsc Total. Porto Alegre, 24 mar. 2021. Disponível em: https://www.nsctotal.com.br/colunistas/nara-caviquioli/aplicativo-para-catalogar-vinhos-voce-ja-baixou. Acesso em: 27 mar. 2024.

Croft, N. Degustação de vinhos. São Paulo: Folio, 2002.

DATA SCIENCE ACADAMY. 10 Razões Que Explicam a Importância dos Bancos de Dados nos Dias de Hoje. 2023. Disponível em: https://blog.dsacademy.com.br/10-razoes-que-explicam-a-importancia-dos-bancos-de-dados-nos-dias-de-hoje/. Acesso em: 01 abr. 2024.

GOOGLE CLOUD. PostgreSQL x SQL Server: quais são as principais diferenças? 2024. Disponível em: https://cloud.google.com/learn/postgresql-vs-sql?hl=pt-br. Acesso em: 01 abr. 2024.

GORESNSTEIN, Ari. As expectativas para o mercado de vinhos para 2024. Exame. [S.I.], 03 fev. 2024. Disponível em: https://exame.com/casual/as-expectativas-para-o-mercado-de-vinhos-para-2024/. Acesso em: 26 mar. 2024.

IBM (comp.). O que é uma API REST? 2024. Disponível em: https://www.ibm.com/br-pt/topics/rest-apis. Acesso em: 26 abr. 2024.

Lanari, R. 2020: um ano fora da curva (e dos vinhos). Exame, São Paulo, 14 dez. 2020. Disponível em: https://exame.com/casual/2020-um-ano-fora-da-curva-e-dos-vinhos/. Acesso em: 22 mar. 2024.

LOUZADA, Vinicius; CARVALHO, Caroline; LARANJA, Emerson. API: o que é, para quê serve e qual é a sua importância. 2024. Disponível em: https://www.alura.com.br/artigos/api. Acesso em: 26 abr. 2024.

MAGALHÃES, Tatiana. E-commerce de vinhos: do digital ao phygital. Winext. 12 maio 2022. Disponível em: https://www.wine-xt.com/pt-br/blog/2022/5/13/e-commerce-de-vinhos-do-digital-ao-phygital. Acesso em: 26 mar. 2024.

MALANDRINO, Pier-Jean. Architecture Patterns: backend for frontend (bff) pattern. 2023. Disponível em: https://medium.com/scub-lab/backend-for-frontend-bff-pattern-57de57683264. Acesso em: 02 abr. 2024.

MARTIN, Robert C.. The Clean Architecture. 2012. Disponível em: https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html. Acesso em: 02 abr. 2024.

MARTIN, Robert C.. Solid Relevance. 2020. Disponível em: https://blog.cleancoder.com/uncle-bob/2020/10/18/Solid-Relevance.html. Acesso em: 03 abr. 2024.

Melo, A. S. Q. T&H Turismo & Hotelaria: bebidas e harmonizações. João Pessoa: Editora do CCTA, 2020.

OLIVEIRA, Marcio. TENDÊNCIAS NO MERCADO DO VINHO PARA 2024. Vinoticias. [S.I.], 5 jan. 2024. Disponível em: https://www.vinoticias.com.br/post/tend%C3%AAncias-no-mercado-do-vinho-para-2024. Acesso em: 26 mar. 2024.

OSTERWALDER, Alexander; PIGNEUR, Yves. Business Model Generation - Inovação em Modelos de Negócios: um manual para visionários, inovadores e revolucionários. Rio de Janeiro: Alta Books, 2011. 300 p.

Ozbun, T. Wine industry in Brazil – statistics & facts. Statista, Jan 10, 2024. Disponível em: https://www.statista.com/topics/5228/wine-industry-in-brazil/#topicOverview. Acesso em: 22 mar. 2024.

POSTGRESQL. About. 2024. Disponível em: https://postgresql.org/about/. Acesso em: 01 abr. 2024.

Pressman, R. S., Maxim, B, R. Engenharia de software: uma abordagem Profissional. 9. ed. – Porto Alegre: AMGH, 2021.

REENSKAUG, Trygve. A note on DynaBook requirements. [S. L.]: Xerox, 1979. Disponível em: https://folk.universitetetioslo.no/trygver/1979/sysreq/SysReq.pdf. Acesso em: 26 abr. 2024.

REENSKAUG, Trygve. THING-MODEL-VIEW-EDITOR: an example from a planningsystem. [S. L.]: Xerox, 1979. Disponível em: https://folk.universitetetioslo.no/trygver/1979/mvc-1/1979-05-MVC.pdf. Acesso em: 26 abr. 2024.

REENSKAUG, Trygve. MVC XEROX PARC 1978-79. 2024. Disponível em: https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html. Acesso em: 26 abr. 2024.

SCHWABER, Ken; SUTHERLAND, Jeff. O Guia do Scrum: o guia definitivo para o Scrum: as regras do jogo. 2020. Disponível em: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf. Acesso em: 27 mar. 2024.

Singh, Karan Pratap. "System Design". Acesso em 1 de abril de 2024. GitHub. Disponível em: https://github.com/karanpratapsingh/system-design/blob/main/README.md.

SIMONYAN, Hayk. SQL Transactions and ACID Properties. Medium. San Francisco, Ca, p. 0-0. 08 mar. 2024. Disponível em: https://levelup.gitconnected.com/sql-transactions-and-acid-properties-bb5b670538f8. Acesso em: 01 abr. 2024.

Sommerville, I. Engenharia de software. 10 ed, São Paulo. Pearson Education do Brasil, 2018.

VERHEYEN, Gunther. Scrum - um guia de bolso: um companheiro de viagem inteligente. [S. L.]: Ullizee-Inc, 2019. 112 p. Edição do Kindle

1. *Progressive Web App*. [↑](#footnote-ref-1)
2. <https://www.atlassian.com/software/jira> [↑](#footnote-ref-2)
3. <https://github.com/> [↑](#footnote-ref-3)
4. *Back end for front end.* [↑](#footnote-ref-4)
5. Após instalar a aplicação no desktop ou no celular o arquivo serviceworker.js garante que a aplicação irá funcionar com conexão limitada ou até mesmo em uma situação sem internet, usando para isso um serviço de armazenamento de cache. [↑](#footnote-ref-5)
6. O Copilot está disponível de forma gratuita para usuários Microsoft e sua descrição pode ser obtida no link: <https://www.microsoft.com/pt-br/microsoft-copilot>. [↑](#footnote-ref-6)
7. <https://www.google.com/> [↑](#footnote-ref-7)
8. Sistema operacional de celulares da marca Apple. [↑](#footnote-ref-8)
9. Sistema operacional de celulares das demais marcas. [↑](#footnote-ref-9)
10. <https://www.evino.com.br/> [↑](#footnote-ref-10)
11. <https://www.vivino.com/BR/pt-BR/> [↑](#footnote-ref-11)
12. <https://www.wine.com.br/> [↑](#footnote-ref-12)
13. <https://play.google.com/store/apps/details?id=com.sh0rtshift.wineNotesFREE&pcampaignid=web_share> [↑](#footnote-ref-13)
14. <https://play.google.com/store/apps/details?id=com.delectable.mobile&hl=en_US> [↑](#footnote-ref-14)
15. <https://aws.amazon.com/pt/> [↑](#footnote-ref-15)
16. Os valores estão expressos em: <https://agilemanifesto.org/iso/ptbr/manifesto.html> enquanto os princípios podem ser encontrados em: <https://agilemanifesto.org/iso/ptbr/principles.html> [↑](#footnote-ref-16)
17. <https://www.atlassian.com/software/jira> [↑](#footnote-ref-17)
18. <https://github.com/about> [↑](#footnote-ref-18)
19. <https://git-scm.com/> [↑](#footnote-ref-19)
20. Integração contínua e entrega contínua. [↑](#footnote-ref-20)
21. <https://miro.com/> [↑](#footnote-ref-21)
22. <https://www.strategyzer.com/library/the-business-model-canvas> [↑](#footnote-ref-22)
23. Ver itens 3.1. ao 3.4. deste trabalho [↑](#footnote-ref-23)
24. <https://excalidraw.com/> [↑](#footnote-ref-24)
25. <https://www.figma.com/> [↑](#footnote-ref-25)
26. CSV significa Comma Separated Values e é um arquivo simples, capaz de armazenar tabelas com dados na forma de texto. [↑](#footnote-ref-26)
27. <https://www.postgresql.org/> [↑](#footnote-ref-27)
28. REST significa "*Representational State Transfer*". É um estilo arquitetural utilizado no desenvolvimento de APIs para sistemas distribuídos na web. [↑](#footnote-ref-28)
29. Apesar de existirem outras solicitações, essas são as mais usadas. [↑](#footnote-ref-29)
30. O back end é a parte de um sistema de software que lida com interações, lógica de processamento e armazenamento de dados e outras operações que ocorrem no lado do servidor. [↑](#footnote-ref-30)
31. <https://www.oracle.com/br/java/> [↑](#footnote-ref-31)
32. JDK é o acrônimo para Java Development Kit [↑](#footnote-ref-32)
33. <https://docs.oracle.com/en/java/javase/17/> [↑](#footnote-ref-33)
34. <https://maven.apache.org/> [↑](#footnote-ref-34)
35. <https://spring.io/projects/spring-boot> [↑](#footnote-ref-35)
36. Existem outros métodos não citados aqui e que podem ser visualizados em <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods> [↑](#footnote-ref-36)
37. Para uma definição da arquitetura envolvendo o BFF ver Malandrino (2023). [↑](#footnote-ref-37)
38. <https://react.dev/> [↑](#footnote-ref-38)
39. <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [↑](#footnote-ref-39)
40. <https://www.facebook.com/> [↑](#footnote-ref-40)
41. *Single Page Application* [↑](#footnote-ref-41)
42. <https://vitejs.dev/> [↑](#footnote-ref-42)
43. Para uma melhor compreensão do SOLID veja Martin(2020). [↑](#footnote-ref-43)
44. JSX é uma extensão de sintaxe JavaScript que permite escrever códigos semelhantes ao HTML, em um componente React. Para mais detalhes ver <https://react.dev/learn/writing-markup-with-jsx> [↑](#footnote-ref-44)
45. <https://getbootstrap.com/> [↑](#footnote-ref-45)
46. <https://primereact.org/> [↑](#footnote-ref-46)
47. <https://react-bootstrap.netlify.app/> [↑](#footnote-ref-47)
48. <https://www.w3.org/> [↑](#footnote-ref-48)
49. TDD é a sigla em inglês para *Test-Driven Development*, que quer dizer Desenvolvimento Orientado a Testes. [↑](#footnote-ref-49)
50. <https://junit.org/junit5/> [↑](#footnote-ref-50)
51. <https://site.mockito.org/> [↑](#footnote-ref-51)
52. <https://docs.spring.io/spring-boot/docs/current/api/org/springframework/boot/test/autoconfigure/web/servlet/WebMvcTest.html> [↑](#footnote-ref-52)
53. <https://docs.spring.io/spring-boot/docs/current/api/org/springframework/boot/test/autoconfigure/orm/jpa/DataJpaTest.html> [↑](#footnote-ref-53)
54. <https://www.h2database.com/html/main.html> [↑](#footnote-ref-54)
55. <https://jestjs.io/pt-BR/> [↑](#footnote-ref-55)
56. <https://www.jacoco.org/jacoco/> [↑](#footnote-ref-56)
57. <https://openai.com/> [↑](#footnote-ref-57)
58. <https://spring.io/projects/spring-ai> [↑](#footnote-ref-58)