

# **Apresentação Final**

# **Engenharia de Software**

# **VirtualLib**

...

Prof: Edeilson

Grupo: Cayke Daniel, Lucas Farias, Filipe Coelho, Lucas Yudi, Gabryel Delanne

# Importância das APG-1 e APG-2

- APG-1: alinhamento inicial, entendimento do problema.
- APG-2: evolução do produto e validação das entregas.

# Experiência com Metodologias Ágeis

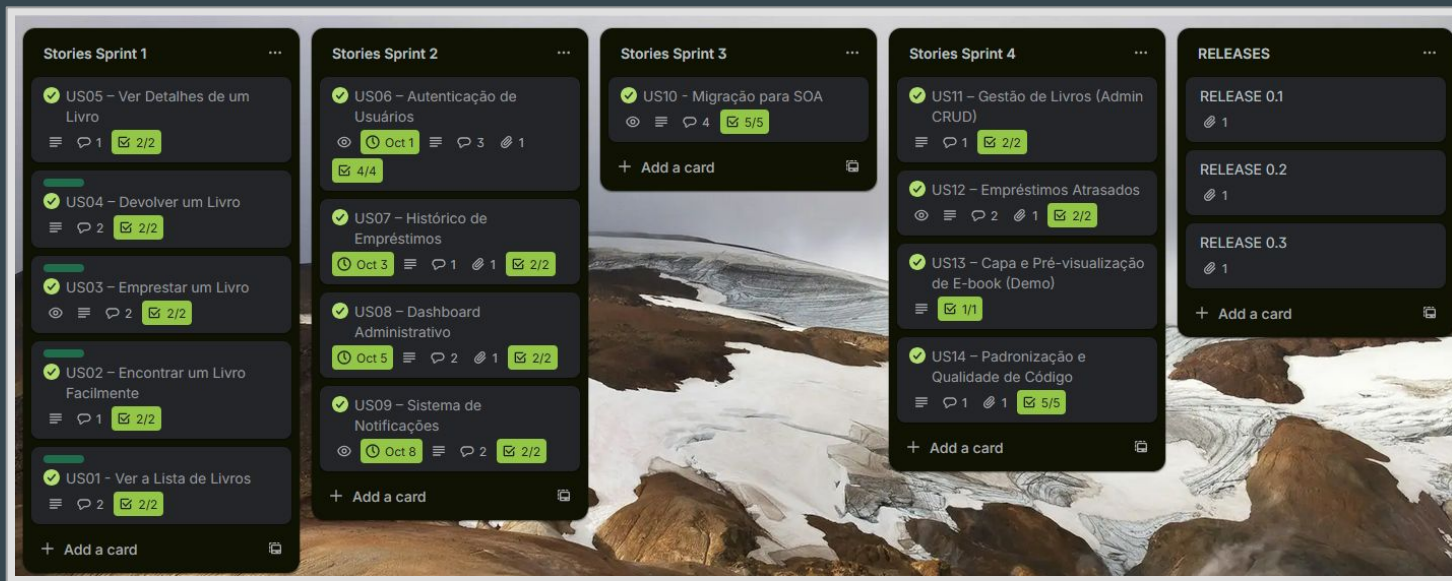
- Sprints
- Divisão de tarefas no board.
- Reuniões rápidas para alinhamento.
- Entregas contínuas e incrementais.

# Desafios Vivenciados

- GitFlow
- Horários
- Bugs e Deploy

# Soluções Aplicadas

- Padronização de branches
- Checklist de PR
- Divisão clara de papéis



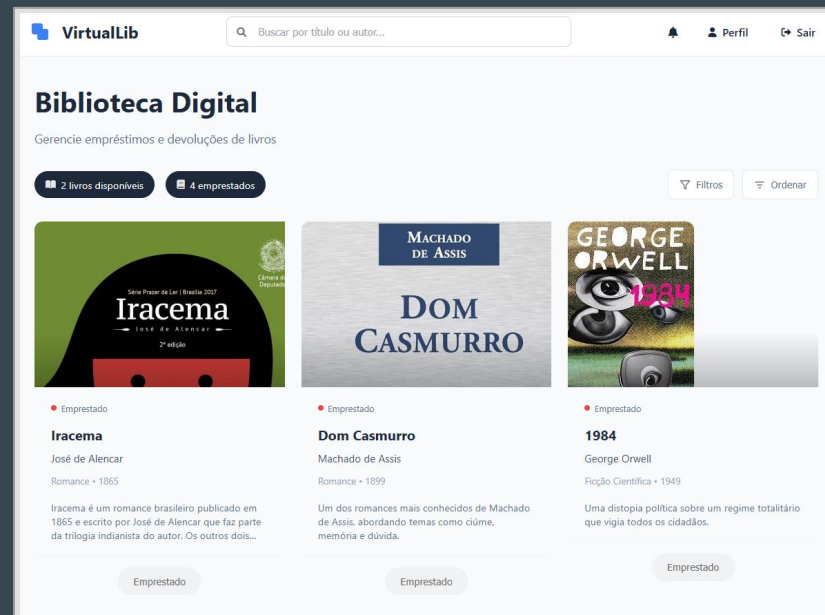
# Pontos Positivos

- Contato real com mercado
- Trabalho em equipe
- Metodologias ágeis
- Prática, não só teoria

aluno	Pontos negativos	pontos positivos	nota
Yudi		pontos positivos na minha opinião é entender o fluxo de trabalho que o mercado exige, aprender a usar o PR, novas branches além da main, prompts de comando como checkout, merge(–force kkkk) etc... aprender a se comportar em frente ao cliente, saber desenvolver uma apresentação entre ser técnico e comercial ao mesmo tempo— nota 10	10
Filipe		Adoção de Fluxo de Trabalho Profissional (GitFlow): O domínio do fluxo de trabalho e das operações Git avançadas (uso de PRs, criação e gestão de múltiplas branches além da main, comandos como checkout e merge) e a capacidade de integrar-se eficientemente a um time de desenvolvimento maduro. Isso minimiza problemas e facilita a colaboração em projetos de grande escala.  Dificuldade Inicial: Organização e Gestão de Prazos, No início, houve uma dificuldade em internalizar a organização de prazos. A intervenção do líder de grupo (que tomou a frente e estabeleceu a cadência) resolveu o problema	10
Luca		trabalhar com praticas de desenvolvimento usadas no mercado; desenvolvemos nossas habilidades de trabalho em equipe, aprendemos a seguir padrões de desenvolvimento em grupo como criação de branches, padronização de branches e commit, fazer Pull Requests e principalmente a lidar com prazos;	10
Cayke	Dificuldade de organizar/desenvolver ao mesmo tempo	Grande aprendizado principalmente relacionado à comunicação e organização durante o período	10
Gabryel		- Organização de projeto e equipe - Interação com git e GitHub - Contatado direto com praticas e ferramentas utilizadas no mercado	10
Total			10

# O Produto

- Funcionalidades principais
- Diferenciais
- Evolução do projeto



# Como entendemos o problema

- Análise de necessidades
- User stories
- Prototipação inicial



# Como priorizamos os requisitos

- Definição do que seria implementado primeiro
- Critérios de valor
- Impacto versus esforço



## Requisitos Funcionais (RF)

- **RF01:** O sistema deve permitir ao usuário visualizar uma lista de livros, exibindo status (disponível/emprestado).
- **RF02:** O sistema deve oferecer uma barra de busca por título ou autor.
- **RF03:** O sistema deve permitir ao usuário marcar um livro como emprestado, alterando seu status no catálogo.
- **RF04:** O sistema deve permitir ao usuário marcar um livro como devolvido, alterando seu status no catálogo.
- **RF05:** O sistema deve exibir detalhes completos de um livro selecionado (sinopse, ano, autor, categoria).



## Requisitos Não Funcionais (RNF)

- **RNF01:** A listagem e a busca devem retornar resultados em tempo real (tempo de resposta  $\leq 2$  segundos).
- **RNF02:** A interface deve ser intuitiva e responsiva.
- **RNF03:** O sistema deve manter consistência nos dados (não mostrar livro como disponível se estiver emprestado).
- **RNF04:** O sistema deve armazenar os dados em um banco confiável com suporte a múltiplos usuários.

# Validação dos Requisitos

- Iterações curtas

# Validação das Tecnologias

- Análise de prós e contras.
- Testes de performance e integração.

# Importância das Sprints

- Organização e previsibilidade.
- Divisão clara do trabalho.
- Redução de riscos.
- Entregas contínuas.

# Entrega de Valor por Iteração

- Cada sprint gera algo utilizável.
- Feedback constante.
- Correções rápidas e direcionadas.

# Importância dos Padrões

- Organização do código.
- Facilita manutenção.
- Escalabilidade.
- Clareza na separação de responsabilidades.

# API e Arquitetura em Serviços

- Independência entre front/back
- Escalabilidade
- Facilidade para integrar

# Testes Unitários

- Segurança para refatorar
- Menos bugs
- Confiança no código



# Controle de Versão (Git)

- Historico claro
- Pull Requests
- Branches organizadas

# GitFlow + Release

- Branch principal estável
- Branches de desenvolvimento
- Processo de release
- Controle de versões



# Conclusão

- Experiência prática
- Trabalho em equipe
- Superação de desafios