

# Trabalho Final Engenharia de Software



## TimerBook



- Tiago Barbosa
- Lucas Monteiro
- Matheus Pontes
- Vitor Kawan
- Bruno
- Kayk Zago

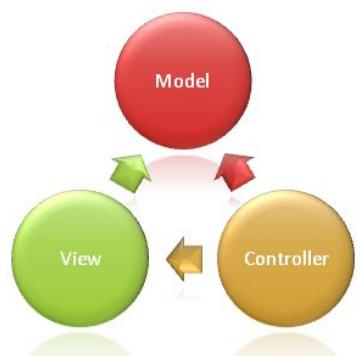
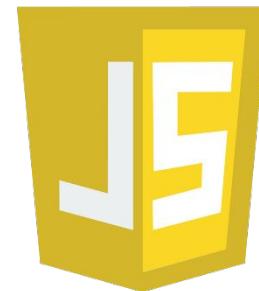


# Parte 1 – Experiência na Discliplina



# Importância das Atividades APG-1 e APG-2

- Aprendizagem de novos conceitos.
- Instalação de dependências.



c) Apresentar as **lições aprendidas**, os **desafios vivenciados na disciplina** (ex: dificuldade de comunicação, o controle de versão do código) e **as soluções propostas para estes desafios** durante todo o ciclo da disciplina.

1. Lições aprendidas:

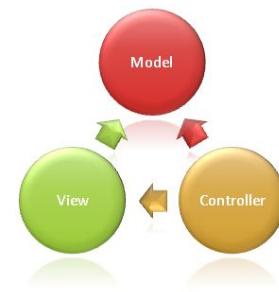
- Aprendizagem dos conceitos de PHP, estrutura MVC, GitFlow e JavaScript
- Importância da organização na produção de um software
- Responsabilidade com as datas de entregas
- Importância de planejamento na construção de um software

2. Desafios:

- Primeiro contato com as novas linguagens e ferramentas
- Cumprimento dos prazos
- Construir o software com PHP puro
- Controle de versão no GitFlow

3. Soluções:

- Ajuda da equipe, monitores e do professor
- Divisão do projeto em tarefas menores
- Reuniões diárias



MySQL™

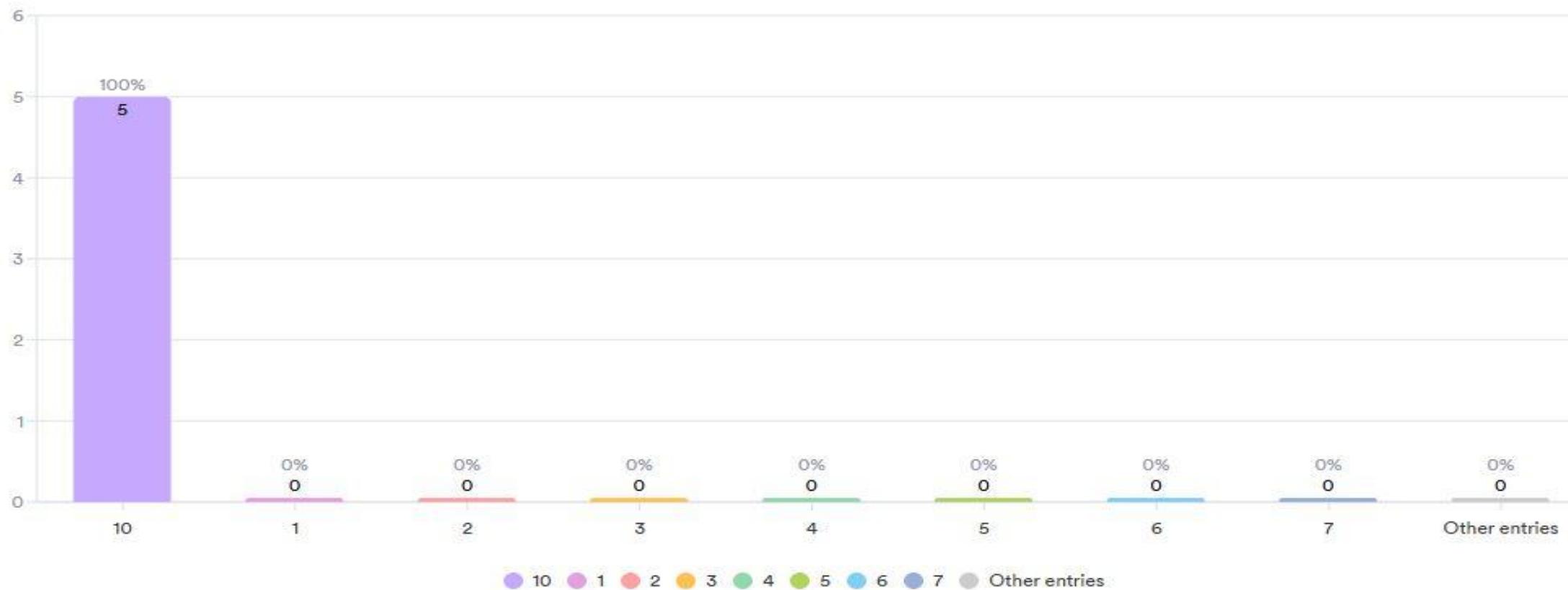
D) Apresentem os pontos positivos e negativos da disciplina (caso haja). Considerem para a sua resposta os seguintes pontos: proporcionar experiência em práticas de mercado da indústria de software; a atuação em times de software; a aplicação de um processo ágil de desenvolvimento; a experiência com o GitFlow e GitHub.

- A disciplina proporciona uma experiência real do processo de criação de um software, com sprints contendo metas a serem cumpridas. A formação de equipes é um ponto chave da preparação para o mercado de trabalho, onde a interação com outras pessoas é constante. O uso da ferramenta de versionamento como o GitHub foi crucial para o controle do projeto, correção de erros e um backup sólido.



## 5 - Numa escala de 1 a 10, qual a sua experiência na disciplina?

5 Responses

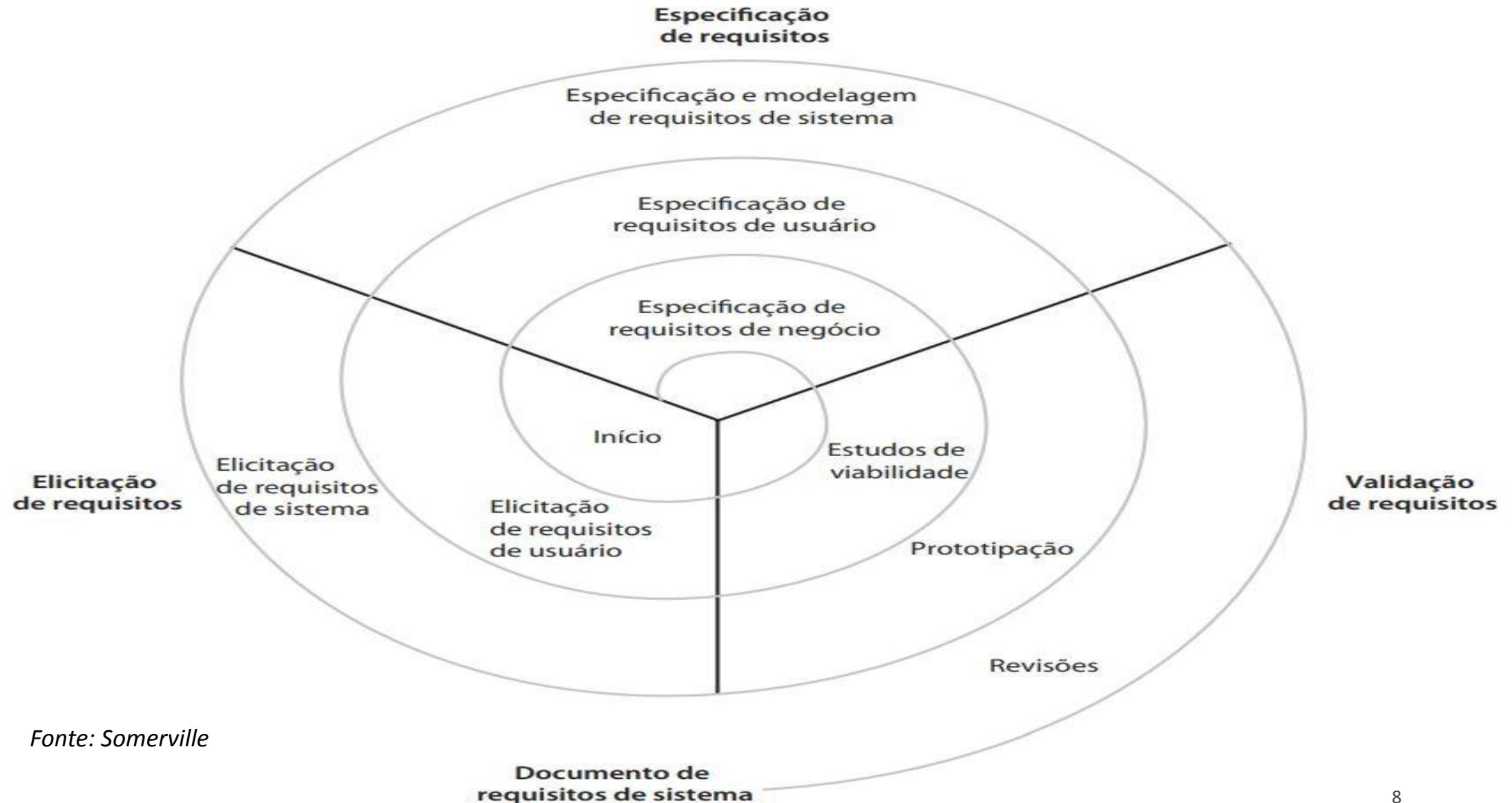




## Parte 2 – Questões de Conhecimento

**“ Imaginem que vocês estão iniciando um novo projeto (um problema desconhecido) para um novo cliente: discorram detalhadamente como vocês lidariam com as questões apresentadas a seguir:”**

# (a) – Quais estratégias seriam utilizadas para entender o problema e todos os seus requisitos?



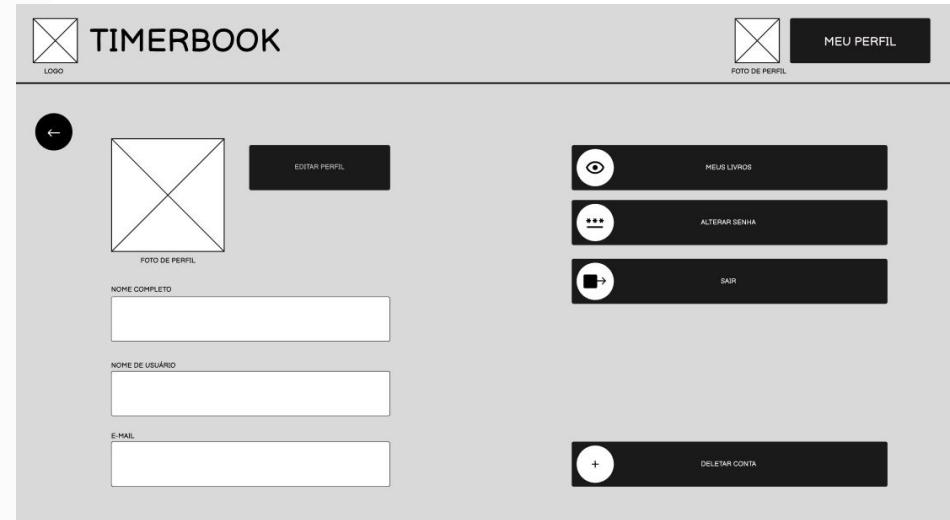
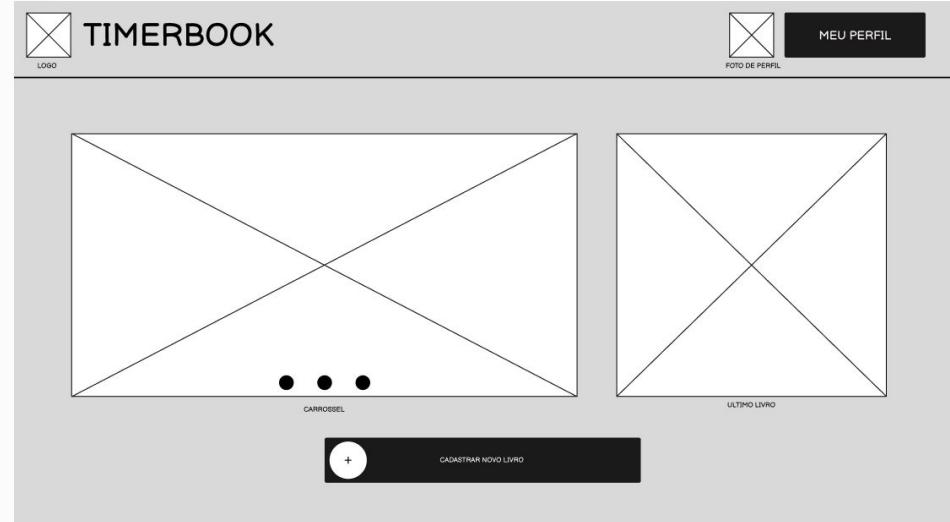
# Definição dos requisitos prioritários do produto

- Identificar o conjunto essencial de funcionalidades.
- Permitir colocar o sistema em funcionamento.
- Classificação dos requisitos:
  - Obrigatório;
  - Importante mas pode esperar;
  - Apenas desejável.



# Validação dos requisitos

- Validação de forma contínua.
- Utilização de Wireframes.
- Revisões para validar se o software atende às expectativas.



d) Como vocês fariam para definição e validação das tecnologias que seriam adotadas?

- Avaliar se o projeto realmente precisa da inserção da nova tecnologia
- Procurar se as novas ferramentas podem atender os requisitos
- Analisar os custos
- Verificar os ajustes que devem ser feitos com a implementação das novas tecnologias
- Monitorar os problemas com uso de testes



# Importância do planejamento das iterações

- Foco em um conjunto reduzido e alcançável de tarefas.
- Previsão da entrega e ajustes rápidos.
- Divisão das tarefas e o acompanhamento foram feitos pelo Trello.

The screenshot shows a Trello board titled "TimerBook - Sistema de Gerenciamento de Leitura". The board is organized into several columns representing different stages of development:

- Sprints:** Lists three sprints: "Sprint 1 - Sistema de Cadastro e Login de Clientes", "Sprint 2 - Tela Inicial, Tela do Administrador e Cadastro de Livros", and "Sprint 3 - Sistema de Avaliação, Ranking de Livros, Perfil do Usuário e".
- User Stories:** Lists user stories grouped by sprint. For Sprint 1, there are S1-D1 and S1-D2. For Sprint 2, there are S2-D1 and S2-D2. For Sprint 3, there are no visible stories.
- Divisão da Sprint 1:** Contains cards for "Tela de Login (HTML e CSS puro)", "Tela de Cadastro (HTML e CSS) + Esqueci\_Minha\_Senha", and "Envio dos dados (imagem) para o back-end".
- Divisão da Sprint 2:** Contains cards for "Front-End" (status 7/7) and "Back-End" (status 6/6). A button "+ Adicionar um cartão" is present.
- Divisão da Sprint 3:** Contains cards for "Front-End" (status 8/8), "Back-End" (status 8/8), and "Testes Unitários" (status 3/3). A button "+ Adicionar um cartão" is present.
- Divisão da Sprint 4:** Contains cards for "Front-End" (status 2/2), "Back-End" (status 4/4), and "Testes Unitários" (status 4/4). A button "+ Adicionar um cartão" is present.
- Divisão da Sprint Final:** Contains cards for "Front-End" (status 1/1), "Back-End" (status 3/3), and "Testes Unitários" (status 2/2). A button "+ Adicionar um cartão" is present.
- Divisão dos Artefatos Finais:** Contains a card for "Tarefas" (status 0/4) and a button "+ Adicionar um cartão".

At the bottom of the board, there are navigation buttons: "Caixa de entrada", "Planejador", "Quadro" (highlighted in blue), and "Mudar de quadros".

f) Qual a importância da definição de valor em uma iteração para o sucesso do produto? Elucidem.

### Importância da Definição de Valor na Iteração

#### Por que isso importa?

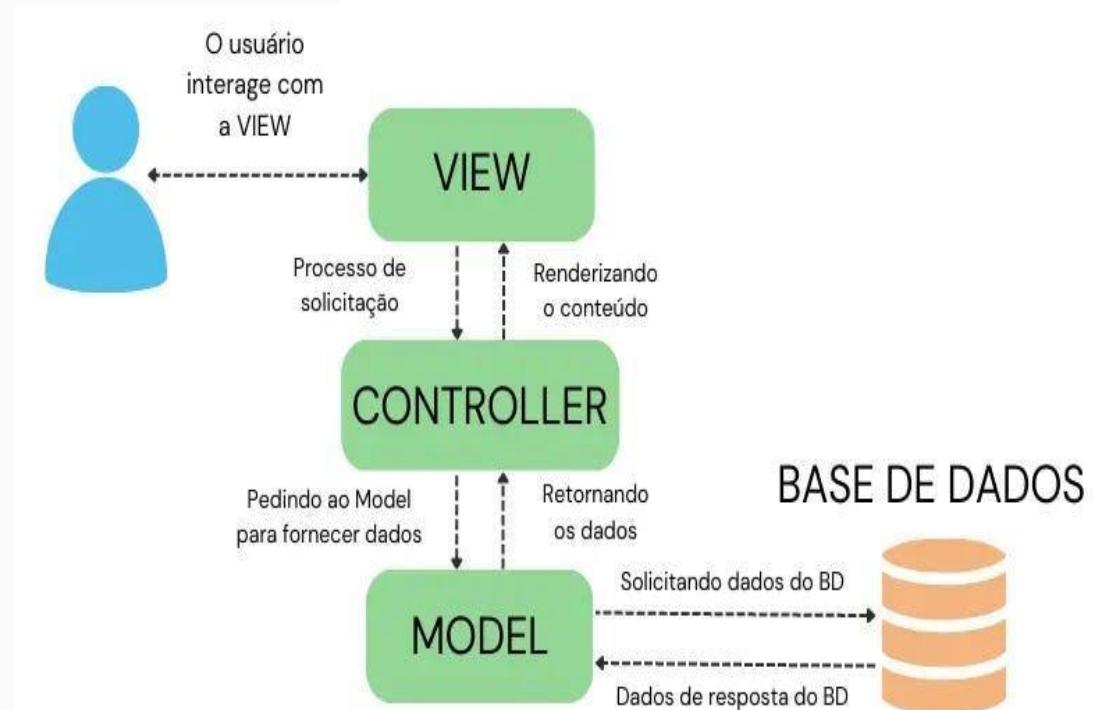
- Dá **clareza** ao time sobre o que entregar
- Mantém o foco e evita perder o “fio da meada”
- Mostra **por que** aquela entrega é necessária
- Evidencia **o que retorna para o cliente**
- Permite **priorizar melhor** e tomar decisões rápidas
- Garante que cada iteração gere **valor real ao produto**



g) Vocês conseguem entender e explicar qual a importância de seguir padrões em projetos, por exemplo, MVC?

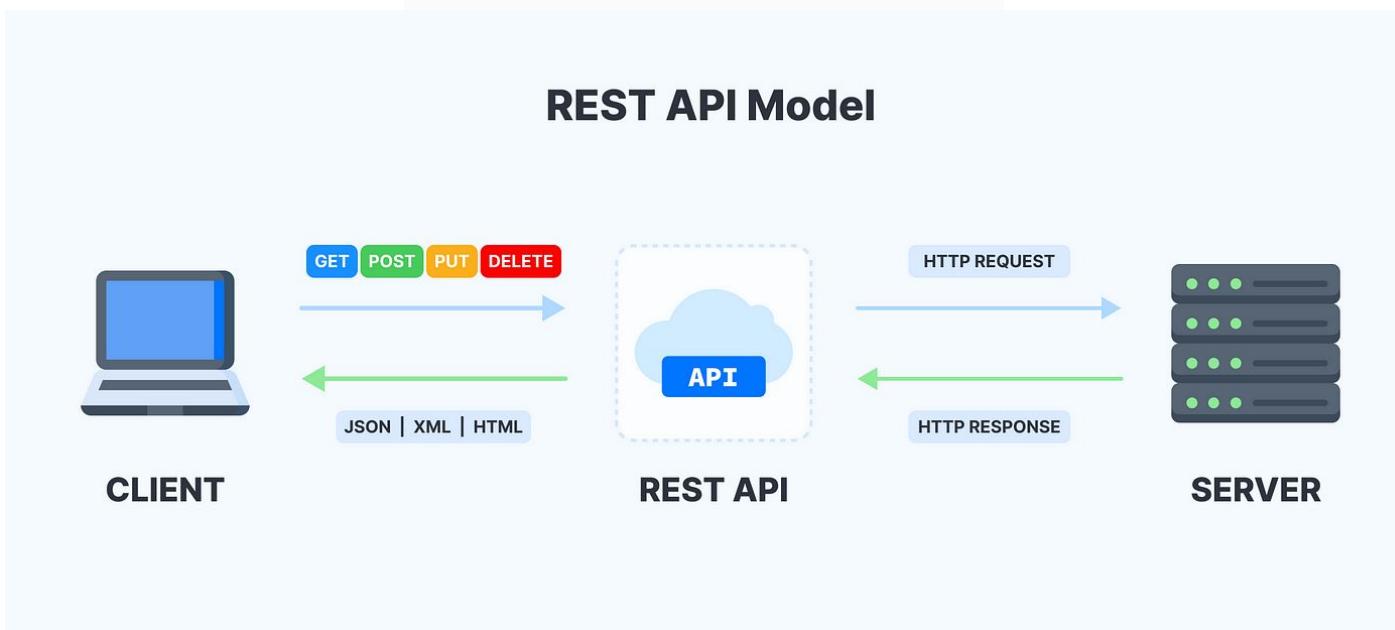
#### Por que usar padrões como MVC?

- Organiza o projeto dividindo responsabilidades
- Código mais limpo, legível e fácil de manter
- Facilita adicionar novas funcionalidades
- Reduz erros e problemas futuros
- Melhora a colaboração entre a equipe
- Facilita testes e reutilização de código



h) Qual a importância de seguir a Arquitetura baseada em Serviços por meio de API REST? Esclareçam.

- A arquitetura baseada em serviços com API REST é importante, porque permite que diferentes sistemas se comuniquem de forma padronizada e eficiente. Ela facilita a manutenção, a escalabilidade e a reutilização de funcionalidades, além de permitir que cada membro da equipe trabalhe em diferentes funções ou features de forma organizada, tornando a integração entre aplicações mais confiável.



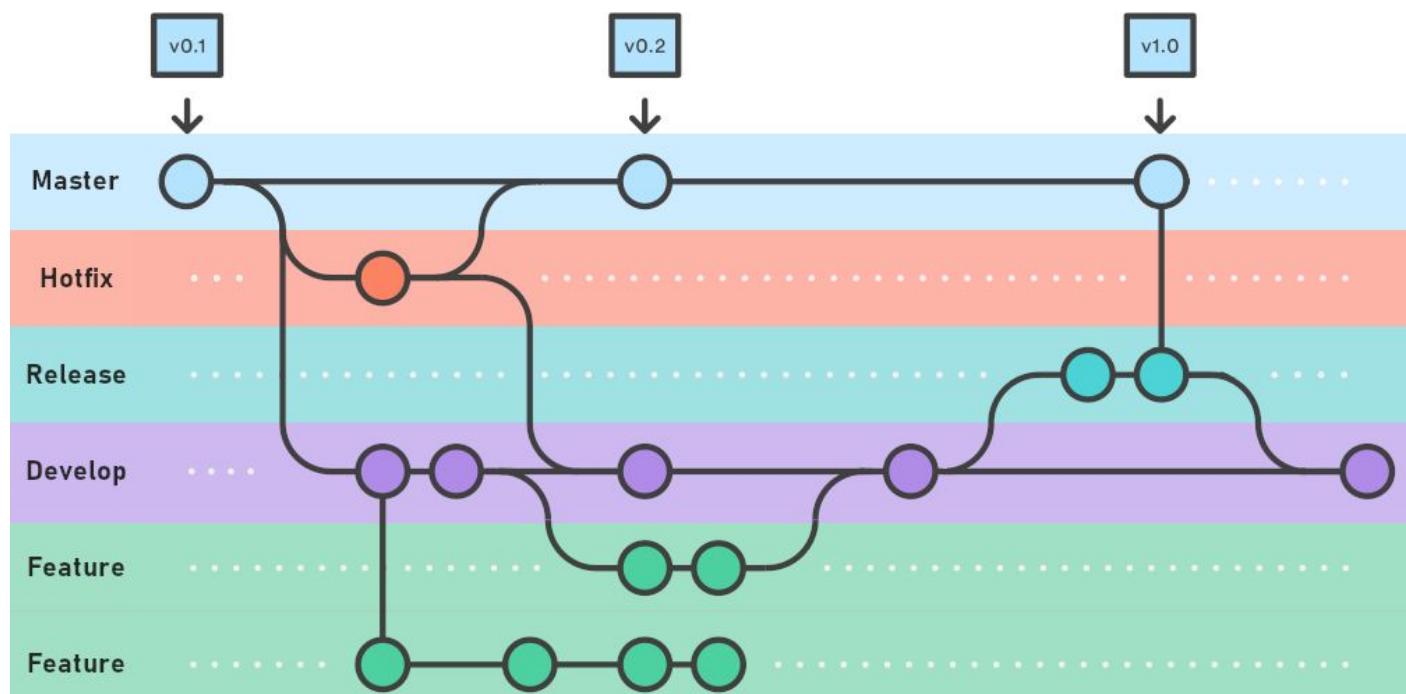
j) Qual a importância do Controle de Versão do código? Como este Controle de Versão é realizado?

- O controle de versão é fundamental para registrar todas as alterações feitas no código, possibilitando reverter para versões anteriores quando necessário. Ele também facilita o trabalho colaborativo, evitando conflitos entre diferentes contribuições.



## k) Como vocês vêm o Fluxo de Trabalho organizado pelo GitFlow?

- O GitFlow organiza o desenvolvimento de software por meio de diferentes branches do github, como Main, Develop, Hotfix, Feature e Release. Esse fluxo de trabalho permite que novas funcionalidades sejam desenvolvidas, testadas e integradas de forma estruturada, garantindo maior controle sobre o código e facilitando a colaboração em equipe e evitando conflitos que teríamos caso todos trabalhassem em cima de uma única branch.



## 1) O que vocês entendem por Release do produto?

- Uma release do produto é uma versão estável do software disponibilizada para os usuários ou clientes. Ela representa um conjunto de funcionalidades completas, corrigidas e testadas, podendo incluir novas features, melhorias ou correções de bugs, ela geralmente é feita após o término de uma iteração.

