

# CRIANDO API COM LARAVEL PHP, PLANEJANDO FEATURES E PADRONIZANDO VERSIONAMENTO

O título desta publicação é cirúrgico ao definir o que será apresentado no presente artigo. Apesar de se tratar de um projeto de iniciação, o processo de desenvolvimento da API será o mais próximo possível de um ambiente profissional, buscando esclarecer as etapas de planejamento e execução, tornando viável e amigável à compreensão de pessoas desenvolvedoras com níveis iniciais de senioridade, como Júnior ou Pleno.

## ETAPAS DO PROCESSO

- Documentação
  - Descrição, Objetivos, Padrões e Diagramas
  - Estrutura de Pastas e Arquivo
  - Registros de Acessos e Erros
  - Ambientes de Desenvolvimento, Homologação e Produção
- Etapas:
  - Planejamento
    - Requisitos
    - Cronograma
  - Desenvolvimento / Testes
    - Estrutura de pastas e arquivos
    - Versionamento: padrões de branches, commits e tags
    - Padrão de Endpoints
    - Aplicação de Testes Unitários
  - Teste de Carga e Balanceamento
    - Teste de Desempenho: quantos usuários/requisições por minuto
    - Teste de Carga: resposta ao aumento exponencial de requisições
    - Teste de Stress: resistência à situações e sobrecargas imprevisíveis
  - Métricas:
    - Disponibilidade
    - Relatório de erros e resoluções
    - Análise de tempo de entrega
  - Entrega
    - Publicação do Projeto em Homologação
    - Publicação do Projeto em Produção
  - Consumo da API

## DADOS TÉCNICOS

Linguagem: PHP

Framework: Laravel

Gerenciador de Pacotes: Composer

ORM: Eloquent

Banco de Dados: MySQL

Controle de versões: GIT por meio da plataforma Github

# DOCUMENTAÇÃO

Como já citado anteriormente, por se tratar de um projeto CRUD de iniciação, não será necessário anexar documentação de requisitos com Business Case e afins. A documentação será apresentada em um projeto resumo das principais funcionalidades, objetivos gerais, proposta de padrões de desenvolvimento e escrita de códigos e diagramas para dar uma ideia mais realista do escopo da aplicação.

API LARAVEL PHP   v1.0.0	
Objetivo Geral	Apresentação do processo de desenvolvimento, iniciação ao teste automatizado, análise de performances, suporte, disponibilidade, restrições e publicação de uma API construída sobre o padrão REST/RESTful.
Objetivo específico	Desenvolver a capacidade analítica de pessoas desenvolvedoras em início de jornada profissional e apresentar caminhos de fluidez para escrita de códigos em PHP.
Recursos	Servidor Web, Banco de Dados (MySQL), PHP 8.^.
Ferramentas	Editor de Código ( <a href="#">VS Code</a> ), Software Gestão de Projetos ( <a href="#">Trello</a> ), Software para modelagem de diagramas ( <a href="#">Draw IO</a> ), Servidor Integrado ( <a href="#">Laragon</a> )
Homologação	<a href="#">Netlify</a> e <a href="#">Heroku</a>
Homologação	<a href="#">Hostinger</a> Hospedagem ( <a href="#">Premium Web Hosting</a> )

CRONOGRAMA		
Ação	Descrição	Status
Documentação	Descrever os requisitos e plano de execução	
Análise e aquisição	Pesquisar hospedagens, repositório e integração	
Modelo de Dados	Descrever a estrutura de banco, tabelas e tipos	
Estruturação	Criar estrutura de pastas, arquivos e conexão	
Testes	Criar testes unitários específicos para cada feature	
Testes de Carga/Stress	Testar performance e suporte em homologação	
Publicação	Fechar versão e publicação em Produção	

# REFERÊNCIAS

## Padrões, Metodologias e Boas Práticas

- **Engenharia de Software Moderna** - Valente, Marco Tulio
- **Engenharia de Requisitos: Software Orientado a Negócios** - Vazquez, Carlos Eduardo; Simões, Guilherme Siqueira
- **Código Limpo** - Martin, Robert C
- **Arquitetura Limpa** - Martin, Robert C
- **Boas Práticas Para Dados na Web** - <https://w3c.br/traducoes/DWBP-pt-br/>
- **Padrões de Aplicação Web** - <https://www.w3c.br/Padroes/WebDesignAplicacoes>