

# Guia Estratégico para o Desenvolvimento do Projeto PET-Tele: Webpages Dinâmicas com PHP e JSON

## Parte 1: Desconstruindo o Desafio: Uma Visão Estratégica

Esta seção inicial enquadra o projeto não apenas como uma tarefa técnica, mas como uma avaliação abrangente do potencial do candidato como membro do PET-Tele. O objetivo é alinhar cada ação realizada com as expectativas explícitas e implícitas da comissão de seleção.

### 1.1 Compreendendo a Filosofia PET-Tele: Além do Código

A etapa final do processo seletivo do PET-Tele é concebida como uma imersão nas práticas e na cultura do grupo. A instrução fundamental é clara: "Assuma que o processo de pesquisa, o estudo a ser efetuado, o projeto a ser realizado e o material a ser entregue, são frutos de um trabalho seu dentro do grupo, como se você já fizesse parte dele".<sup>1</sup> Esta premissa é a base de toda a avaliação. O desempenho não será medido apenas pelo produto final — uma página web funcional —, mas pela maneira como todo o processo é conduzido.

A comissão de seleção utilizará os mesmos princípios empregados na avaliação contínua dos bolsistas atuais.<sup>1</sup> A lista de critérios é extensa e detalhada, incluindo: "comprometimento com a tarefa, empenho no cumprimento da tarefa, cumprimento dos prazos, qualidade na execução e na apresentação dos resultados, independência, iniciativa, propositividade e criatividade".<sup>1</sup> Estes não são termos abstratos, mas sim um roteiro preciso que guiará a análise de cada etapa do trabalho.

A exigência de um "diário de atividades" <sup>1</sup> reforça a importância do processo. Este

documento, juntamente com os critérios de avaliação, revela que o "como" o trabalho é feito é tão, ou mais, importante quanto "o que" é feito. Um projeto tecnicamente perfeito, mas entregue sem um planejamento claro, uma documentação robusta ou uma demonstração de autonomia, será considerado incompleto. O diário de atividades, portanto, transcende um simples registro de tarefas. Ele deve ser construído como uma narrativa do desenvolvimento, documentando não apenas as ações ("Escrevi o código PHP para ler o arquivo JSON"), mas também o raciocínio por trás delas ("Optei por uma estrutura de dados aninhada no JSON para representar a relação entre os grupos e seus cursos, o que simplificará a lógica de iteração no PHP"). Desafios encontrados e a forma como foram superados devem ser detalhados, pois evidenciam independência e capacidade de resolução de problemas. Ideias que vão além dos requisitos mínimos demonstram iniciativa e "propositividade", qualidades altamente valorizadas.

## 1.2 Decodificando os Entregáveis: Uma Lista de Verificação para a Excelência

O sucesso nesta etapa depende da entrega de um conjunto coeso de materiais, cada um com um propósito específico e um alto padrão de qualidade. Os entregáveis obrigatórios são <sup>1</sup>:

1. **Um Diário de Atividades** em formato PDF, relatando cronologicamente todo o processo.
2. **Um Documento Textual** (relatório) em formato PDF, organizado em seções, resumindo a pesquisa e o projeto. Este documento deve ser autoral e elaborado utilizando o Sistema de Preparação de Documentos LaTeX.<sup>1</sup>
3. **Um Documento de Apresentação de Slides**, que resume a pesquisa e o projeto. O uso do Beamer, uma extensão do LaTeX, é sugerido.<sup>1</sup>
4. **O Projeto Prático**, que deverá ser funcional e apresentado em uma demonstração ao vivo para comprovar seu correto funcionamento.<sup>1</sup>

A exigência explícita de ferramentas como LaTeX e a sugestão do Beamer sinalizam a expectativa por um padrão de comunicação acadêmica e profissional rigoroso. Estes não são meros formalismos; são ferramentas que o grupo utiliza e espera que seus membros dominem.

Esses quatro entregáveis não são independentes; eles se complementam para formar uma narrativa coesa sobre a competência do candidato. O diário revela o processo de trabalho e a metodologia. O relatório em LaTeX demonstra a capacidade de formalizar a pesquisa, documentar tecnicamente o trabalho e seguir padrões acadêmicos. A apresentação em Beamer testa a habilidade de sintetizar e comunicar informações complexas de forma clara e eficaz. Finalmente, o projeto funcional comprova a proficiência técnica. Um desequilíbrio entre eles — por exemplo, um excelente projeto com uma documentação deficiente — será

interpretado como uma falha em atender aos padrões integrais do grupo. Consequentemente, a documentação não pode ser uma tarefa relegada para o final. O aprendizado e a utilização do LaTeX e do Beamer devem ser integrados ao cronograma desde o início, garantindo que a elaboração desses materiais ocorra em paralelo com o desenvolvimento do projeto.

## Parte 2: A Base de Conhecimento: Dominando as Tecnologias Essenciais

Esta seção funciona como um guia de estudo direcionado. O objetivo é avançar das definições básicas para uma compreensão prática de como cada tecnologia se encaixa no objetivo do projeto, fornecendo uma base sólida para a fase de implementação.

### 2.1 Fundamentos da Web: O Modelo Cliente-Servidor na Prática

O projeto exige uma compreensão fundamental dos conceitos de Internet, Web e webpages, bem como do modelo que rege seu funcionamento: o modelo Cliente-Servidor.<sup>1</sup> A tarefa central — transformar uma página estática em uma página dinâmica — é a demonstração clássica dessa arquitetura. O PHP é explicitamente definido como uma linguagem que atua no lado do servidor.<sup>1</sup>

Para contextualizar, o fluxo de interação do projeto pode ser descrito da seguinte forma:

1. **Cliente (Navegador Web):** O usuário digita o endereço da sua página (ex: `http://localhost/pet_uff/`) e pressiona Enter. O navegador envia uma requisição HTTP para o servidor solicitando o recurso.
2. **Servidor (Ambiente XAMPP):** O servidor web (Apache) recebe a requisição. Ao identificar que o arquivo solicitado é um `.php`, ele não o envia diretamente. Em vez disso, ele o passa para o interpretador PHP.
3. **Processamento no Lado do Servidor:** O script PHP é executado. Ele realiza as tarefas programadas: localizar os arquivos JSON no sistema de arquivos do servidor, ler seus conteúdos, decodificá-los para uma estrutura de dados manipulável (como um array) e, em seguida, usar essa estrutura para gerar dinamicamente um documento HTML completo.
4. **Resposta do Servidor:** Após a execução do script, o servidor envia o documento HTML recém-gerado de volta para o navegador do cliente como resposta à requisição inicial.
5. **Renderização no Lado do Cliente:** O navegador recebe o HTML puro (sem nenhum

código PHP visível) e o renderiza, exibindo a página formatada para o usuário.

Este entendimento prático é crucial, pois esclarece por que o código PHP nunca é visível para o usuário final e como os dados do servidor são transformados em uma página web interativa.

## 2.2 Front-End: Estrutura e Estilo (HTML & CSS)

As linguagens HTML e CSS são a base da apresentação visual de qualquer página web. O projeto especifica o uso de HTML, HTML-form (formulários) e CSS.<sup>1</sup> O HTML (HyperText Markup Language) é responsável por definir a estrutura e o conteúdo da página, enquanto o CSS (Cascading Style Sheets) controla a sua aparência visual — cores, fontes, espaçamento e layout.

Uma abordagem que demonstra um nível mais elevado de conhecimento e alinhamento com as boas práticas modernas é o uso de **HTML Semântico**. Embora não seja um requisito explícito, sua aplicação reflete o critério de "qualidade na execução". Em vez de usar elementos genéricos como `<div>` e `<span>` para tudo, o HTML semântico emprega tags que descrevem o significado do seu conteúdo. Por exemplo, para estruturar a página dos grupos PET, pode-se usar `<main>` para o conteúdo principal, `<section>` para agrupar a lista de grupos, e `<article>` para cada "card" individual de um grupo. Outras tags como `<header>`, `<footer>` e `<nav>` também devem ser utilizadas para suas finalidades específicas.

A utilização de HTML semântico traz benefícios diretos para a acessibilidade, um pilar do desenvolvimento web moderno. Leitores de tela utilizados por pessoas com deficiência visual conseguem interpretar a estrutura da página de forma mais eficaz, melhorando a navegabilidade. Isso demonstra uma preocupação com a inclusão e um conhecimento que vai além do básico, mostrando iniciativa. Para aprofundar o conhecimento, a documentação da MDN Web Docs é uma referência de alta qualidade para aprender os fundamentos de HTML e CSS.<sup>2</sup>

## 2.3 Back-End: O Motor Dinâmico (PHP)

O PHP (Hypertext Preprocessor) é a tecnologia central do lado do servidor para este projeto.<sup>1</sup> Sua função é executar a lógica que transforma os dados brutos armazenados nos arquivos JSON em conteúdo dinâmico a ser exibido na página. O código PHP é embutido no arquivo

HTML e processado pelo servidor antes que a página seja enviada ao cliente.

Para a execução bem-sucedida do projeto, o domínio de algumas funções e estruturas de controle do PHP é essencial:

- `file_get_contents('caminho/para/arquivo.json')`: Esta função é usada para ler o conteúdo completo de um arquivo (neste caso, um arquivo JSON) e carregá-lo em uma variável como uma string.<sup>7</sup>
- `json_decode($string_json, true)`: Esta é a função chave para a interação com os dados. Ela pega a string JSON lida pela função anterior e a converte em uma estrutura de dados do PHP. O segundo parâmetro, `true`, é importante, pois força a conversão em um array associativo, que é geralmente mais fácil de manipular com loops, em vez de um objeto.<sup>9</sup>
- `foreach ($array as $item)`: Esta estrutura de repetição é ideal para percorrer o array de grupos PET. A cada iteração, a variável `$item` conterá os dados de um grupo, permitindo gerar o HTML correspondente de forma programática.

Para o aprendizado, a documentação oficial do PHP oferece um tutorial introdutório robusto.<sup>11</sup> Guias práticos, como os encontrados em plataformas como FreeCodeCamp, também podem acelerar a compreensão com exemplos aplicados.<sup>13</sup>

## 2.4 Intercâmbio de Dados: O Papel do JSON

O JSON (JavaScript Object Notation) é o formato escolhido para o armazenamento dos dados.<sup>1</sup> Trata-se de um formato de texto leve e de fácil leitura para humanos, que se tornou um padrão de fato para a troca de dados na web. Sua sintaxe é baseada em dois conceitos principais:

- **Objetos:** Uma coleção de pares chave-valor, delimitada por chaves `{}`. As chaves são strings e os valores podem ser strings, números, booleanos, arrays ou outros objetos.
- **Arrays:** Uma lista ordenada de valores, delimitada por colchetes `[]`.

No contexto do projeto, um arquivo JSON conterá um array de objetos, onde cada objeto representa um grupo PET/ProPET e seus atributos (nome, tutor, curso, etc.). Por exemplo, a representação de um único grupo poderia ser:

JSON

```
{
```

```
"groupName": "PET TELECOMUNICAÇÕES - PET TELE",  
"category": "PET",  
"tutor": "Professor Alexandre Santos de La Vega",  
"courses":,  
"contactEmail": "pet.tele@gmail.com"  
}
```

A clareza e a simplicidade do JSON facilitam a manutenção dos dados. Atualizar informações sobre um grupo exige apenas a edição de um arquivo de texto, sem a necessidade de alterar o código PHP, o que demonstra a vantagem de separar os dados da lógica de apresentação.

## Parte 3: O Plano para o Sucesso: Arquitetura do Projeto e Estratégia de Dados

Esta é a fase crítica de planejamento. Um plano robusto nesta etapa tornará a fase de desenvolvimento mais fluida e demonstrará habilidades de previsão e organização — critérios de avaliação chave.

### 3.1 Coleta e Estruturação de Dados

A primeira tarefa prática envolve pesquisa e organização de dados. As instruções são específicas<sup>1</sup>:

- **Fontes de Dados:** A coleta deve começar pelas páginas web do próprio PET-Tele e da UFF sobre os grupos PET e ProPET.<sup>1</sup> A página da PROGRAD/UFF é a fonte primária mais detalhada.<sup>14</sup>
- **Resolução de Conflitos:** As instruções preveem inconsistências. Em caso de discrepância na lista de grupos, a lista do PET-Tele deve prevalecer. Para dados específicos de um grupo (tutor, contato), os dados da página da UFF devem ser usados.<sup>1</sup>
- **Tratamento de Dados Ausentes:** Se houver dados faltantes, o candidato deve tentar obtê-los e, crucialmente, citar as fontes utilizadas. Este é um teste de iniciativa e metodologia de pesquisa.
- **Organização dos Arquivos:** Os dados devem ser segregados em quatro arquivos JSON distintos, baseados em duas classificações: PET vs. ProPET, e "curso único" vs. "conexão de saberes".<sup>1</sup>

Esta tarefa vai além da simples entrada de dados; é uma avaliação da capacidade de pesquisa e resolução de problemas. A maneira como as lacunas de informação são tratadas é fundamental. Simplesmente afirmar que "o dado não foi encontrado" é insuficiente. O diário de atividades deve refletir o processo investigativo. Por exemplo, uma entrada poderia ser: "Os dados de contato do ProPET Engenharia de Petróleo estão ausentes na página da PROGRAD.<sup>14</sup> Realizei uma busca pelo site do Departamento de Engenharia de Petróleo da UFF. Encontrei informações de contato da pós-graduação <sup>15</sup>, mas não específicas do ProPET. Registrarei esta lacuna na documentação final e prosseguirei com os dados disponíveis, demonstrando o esforço de pesquisa." Este tipo de registro evidencia independência e rigor metodológico.

Para garantir consistência, é vital definir uma estrutura de dados padrão antes de iniciar a coleta. A tabela a seguir propõe um modelo para cada objeto de grupo dentro dos arquivos JSON.

**Tabela 1: Estrutura Proposta para os Dados em JSON**

Chave	Tipo de Dado	Descrição	Exemplo	Prioridade da Fonte
groupName	String	Nome oficial e completo do grupo	"PET ENGENHARIA MECÂNICA CAMPUS NITERÓI"	Página da UFF <sup>14</sup>
category	String	Categoria do programa: "PET" ou "ProPET"	"PET"	Página da UFF <sup>14</sup>
type	String	Tipo de grupo: "Curso Único" ou "Conexão de Saberes"	"Curso Único"	Inferido dos cursos <sup>14</sup>
creationDate	String	Data ou ano de criação do grupo	"1996"	Página da UFF <sup>14</sup>

tutor	String	Nome do(a) professor(a) tutor(a)	"Professora Fabiana Rodrigues Leta"	Página da UFF <sup>14</sup>
courses	Array de Strings	Lista de cursos de graduação envolvidos	["Engenharia Mecânica"]	Página da UFF <sup>14</sup>
contactEmail	String	E-mail público de contato do grupo	"uffpetmec@gmail.com"	Página da UFF <sup>14</sup>
website	String	URL do site, blog ou agregador de links	" <a href="https://petmec.uff.br/">https://petmec.uff.br/</a> "	Página da UFF <sup>14</sup>
instagram	String	Nome de usuário do perfil no Instagram (sem o @)	"petgeografiauff"	Página da UFF <sup>14</sup>
address	String	Endereço físico ou localização do grupo	"Rua Passo da Pátria, 156 – Campus da Praia Vermelha..."	Página da UFF <sup>14</sup>
campus	String	Campus da UFF onde o grupo está sediado	"Niterói"	Inferido do endereço <sup>14</sup>

### 3.2 Design de UI/UX: Propondo a Solução

A especificação do projeto contém uma diretriz crucial: "A estrutura de apresentação dos



dados nas webpages não é definida e deverá ser proposta pelo desenvolvedor".<sup>1</sup> Esta é uma oportunidade explícita para demonstrar "criatividade" e "propositividade". Uma simples tabela HTML, embora funcional, mostraria o mínimo de esforço. Uma abordagem mais sofisticada e alinhada com as práticas de design modernas é a criação de um layout baseado em "cards".

Um design de cards consiste em apresentar cada grupo PET/ProPET como uma unidade de conteúdo visualmente distinta e autocontida.<sup>16</sup> Cada card agruparia as informações mais importantes de um grupo (nome, curso, tutor, links de contato) de forma organizada e esteticamente agradável. Este padrão de design é inerentemente responsivo e se adapta bem a diferentes tamanhos de tela, desde desktops a dispositivos móveis.

Para ir além, pode-se propor funcionalidades interativas que melhorem a experiência do usuário (UX). A adição de filtros — por exemplo, botões para alternar a visualização entre "Todos", "Apenas PET" e "Apenas ProPET", ou um menu suspenso para filtrar por campus — demonstra uma mentalidade proativa. Estas funcionalidades não são complexas de implementar com PHP (a lógica de filtragem pode ser aplicada no servidor antes de gerar o HTML), mas agregam um valor significativo ao projeto.

Antes de iniciar a codificação, é uma excelente prática criar um esboço ou wireframe simples da interface proposta. Este esboço pode ser feito à mão ou com uma ferramenta online gratuita e deve ser incluído na seção "Método" do relatório em LaTeX. Isso documenta o processo de planejamento e justifica as decisões de design, mostrando que a solução foi pensada e não improvisada.

### 3.3 Arquitetura de Código e Arquivos

Uma organização de arquivos lógica e limpa é um indicativo de profissionalismo e facilita o desenvolvimento e a manutenção do projeto, refletindo o critério de "qualidade na execução". Uma estrutura de diretórios bem definida separa as diferentes responsabilidades do sistema (dados, lógica, estilo). A seguinte estrutura é recomendada:

```
/pet_uff_project/  
|  
|-- index.php  
|
```

```
|-- /css/  
| |-- style.css  
|  
|-- /data/  
| |-- pet_curso_unico.json  
| |-- pet_conexao_saberes.json  
| |-- propet_curso_unico.json  
| |-- propet_conexao_saberes.json  
|  
|-- /img/  
|-- (imagens, como logos, se utilizadas)
```

- **index.php:** O arquivo principal. Conterá a estrutura HTML da página e o código PHP para carregar os dados, processar filtros e gerar os cards dinamicamente.
- **/css/style.css:** O arquivo de folha de estilos. Centraliza todas as regras de CSS para manter o código HTML limpo.
- **/data/:** Um diretório dedicado a armazenar os quatro arquivos JSON, separando claramente os dados da aplicação.
- **/img/:** Um diretório para armazenar quaisquer recursos de imagem, como um logo da UFF ou ícones para os links de contato.

Esta separação de interesses é um princípio fundamental da engenharia de software e sua adoção, mesmo em um projeto de pequena escala, demonstra maturidade técnica.

## Parte 4: O Sprint de Desenvolvimento: Um Guia de Implementação Passo a Passo

Esta seção oferece um roteiro prático e detalhado para a construção do projeto, desde a configuração do ambiente até a implementação das funcionalidades.

### 4.1 Configurando o Ambiente de Desenvolvimento Local

Para executar código PHP, é necessário um ambiente de servidor. O XAMPP é uma solução gratuita e popular que agrupa o servidor web Apache, o PHP e o banco de dados MySQL em um único pacote de fácil instalação.<sup>19</sup>

### Passos para a Configuração:

1. **Download e Instalação:** Baixe o instalador do XAMPP para o seu sistema operacional a partir do site oficial da Apache Friends.<sup>19</sup> Siga as instruções de instalação, mantendo as opções padrão.
2. **Iniciando os Módulos:** Abra o Painel de Controle do XAMPP. Inicie o módulo "Apache" clicando no botão "Start". A indicação ficará verde, mostrando que o servidor web está ativo.
3. **Localizando o Diretório Raiz:** O XAMPP cria um diretório chamado htdocs dentro da sua pasta de instalação (ex: C:\xampp\htdocs). Este diretório é a raiz do seu servidor web local. Todos os seus projetos web devem ser colocados aqui.
4. **Criando a Pasta do Projeto:** Dentro de htdocs, crie a pasta para o seu projeto, seguindo a arquitetura definida anteriormente (ex: pet\_uff\_project).
5. **Acessando o Projeto:** Abra seu navegador web e acesse o projeto através do URL [http://localhost/pet\\_uff\\_project/](http://localhost/pet_uff_project/). Inicialmente, a página estará vazia ou mostrará um erro, o que é normal até que o arquivo index.php seja criado.

## 4.2 Fase 1: Implementação dos Dados e Lógica Central

Esta fase foca em fazer os dados aparecerem na tela, mesmo sem estilo.

Passo 1: Criar e Popular os Arquivos JSON

Dentro do diretório /data/, crie os quatro arquivos .json necessários. Usando a estrutura definida na Tabela 1, preencha cada arquivo com os dados coletados dos grupos PET e ProPET, garantindo que a sintaxe JSON (chaves entre aspas, vírgulas entre os elementos) esteja correta.

Passo 2: Escrever a Lógica de Leitura em PHP

No arquivo index.php, comece com um bloco de código PHP no topo do arquivo. Este bloco será responsável por toda a manipulação de dados.

PHP

```
<?php
// Função para carregar e decodificar um arquivo JSON
function loadGroups($filePath) {
    if (!file_exists($filePath)) {
        return; // Retorna um array vazio se o arquivo não existir
    }
}
```

```

    $json_data = file_get_contents($filePath);
    return json_decode($json_data, true);
}

// Carregar dados de todos os arquivos
$pet_curso = loadGroups('data/pet_curso_unico.json');
$pet_conexao = loadGroups('data/pet_conexao_saberes.json');
$propet_curso = loadGroups('data/propet_curso_unico.json');
$propet_conexao = loadGroups('data/propet_conexao_saberes.json');

// Unir todos os grupos em um único array
$all_groups = array_merge($pet_curso, $pet_conexao, $propet_curso, $propet_conexao);

// Lógica de filtro (exemplo para ser implementado posteriormente)
$display_groups = $all_groups;
?>

```

### Passo 3: Gerar o HTML Dinamicamente

Abaixo do bloco PHP, crie a estrutura básica do HTML. No <body>, use um loop foreach do PHP para iterar sobre o array \$display\_groups e gerar o HTML para cada card.

#### HTML

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Grupos PET e ProPET da UFF</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <header>
        <h1>Grupos PET e ProPET da UFF</h1>
    </header>
    <main>
        <div class="card-container">
            <?php foreach ($display_groups as $group):?>
                <article class="card">
                    <h2><?php echo htmlspecialchars($group['groupName']);?></h2>
                    <p><strong>Tutor(a):</strong> <?php echo htmlspecialchars($group['tutor']);?></p>
                </article>
            </?php?>
        </div>
    </main>
</body>
</html>

```

```

        <p><strong>Categoria:</strong> <?php echo
htmlspecialchars($group['category']);?></p>
        <p><strong>Campus:</strong> <?php echo
htmlspecialchars($group['campus']);?></p>
        <footer>
            <?php if (!empty($group['website'])):?>
                <a href="<?php echo htmlspecialchars($group['website']);?>"
target="_blank">Site</a>
            <?php endif;?>
            <?php if (!empty($group['contactEmail'])):?>
                <a href="mailto:<?php echo htmlspecialchars($group['contactEmail']);?>">Email</a>
            <?php endif;?>
        </footer>
    </article>
<?php endforeach;?>
</div>
</main>
</body>
</html>

```

*Nota: O uso de htmlspecialchars() é uma prática de segurança essencial para prevenir ataques de Cross-Site Scripting (XSS), garantindo que qualquer dado vindo do JSON seja exibido como texto puro.*

## 4.3 Fase 2: Estilização e Apresentação

Com a lógica funcionando, o foco agora é a aparência e a usabilidade.

Passo 1: Estilizar os Cards com CSS

No arquivo css/style.css, defina as regras para o layout e a aparência dos cards. O uso de CSS Flexbox ou Grid é altamente recomendado para criar um layout de grade responsivo.

CSS

```

/* Exemplo de CSS básico */
body {
    font-family: sans-serif;
    background-color: #f4f4f4;
}

```

```

margin: 0;
padding: 20px;
}

.card-container {
display: grid;
grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
gap: 20px;
}

.card {
background-color: #ffffff;
border: 1px solid #ddd;
border-radius: 8px;
padding: 20px;
box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}

.card h2 {
margin-top: 0;
color: #333;
}

.card footer a {
margin-right: 10px;
text-decoration: none;
color: #007bff;
}

```

## Passo 2: Garantir a Responsividade

A solução deve funcionar bem em diversos dispositivos.<sup>1</sup> Use media queries no CSS para ajustar o layout em telas menores. Por exemplo, em dispositivos móveis, a grade de cards pode se transformar em uma única coluna.

### CSS

```

@media (max-width: 768px) {
.card-container {
grid-template-columns: 1fr; /* Uma coluna em telas menores */
}
}

```

}

## 4.4 Boas Práticas: Testes e Controle de Versão

Testes:

A especificação exige que a solução funcione "nos navegadores Web mais utilizados".<sup>1</sup> Após a implementação, é mandatório testar a página em navegadores como Google Chrome, Mozilla Firefox e Microsoft Edge para garantir consistência visual e funcional.

Controle de Versão com Git e GitHub:

Embora não seja um requisito explícito, o uso de um sistema de controle de versão como o Git demonstra um alto grau de profissionalismo e maturidade técnica. O Git permite rastrear todas as alterações no código, criar um histórico de desenvolvimento e reverter para versões anteriores se necessário. O GitHub serve como uma plataforma para hospedar o repositório de código na nuvem, funcionando como um backup e um portfólio.<sup>22</sup>

Adotar essa prática é uma forma poderosa de impressionar a comissão de seleção. O histórico de commits (git log) funciona como um complemento técnico detalhado para o diário de atividades, mostrando a evolução granular do projeto.

### Mini-guia para iniciar com Git:

1. **Instalar o Git:** Baixe e instale o Git a partir do site oficial.<sup>23</sup>
2. **Inicializar o Repositório:** Navegue até a pasta do seu projeto (pet\_uff\_project) pelo terminal e execute o comando `git init`. Isso cria um repositório Git local.
3. **Fazer o Primeiro Commit:** Adicione todos os arquivos ao controle de versão com `git add.` e, em seguida, salve o estado atual do projeto com `git commit -m "Commit inicial: estrutura do projeto e arquivos JSON"`.
4. **Criar Repositório no GitHub:** Crie uma conta gratuita no GitHub, crie um novo repositório (pode ser privado) e siga as instruções para conectar seu repositório local ao remoto.

Manter um fluxo de trabalho de fazer commits regulares para cada nova funcionalidade ou correção cria um registro profissional do seu trabalho.

## Parte 5: Os Entregáveis Acadêmicos: Elaborando o Relatório e a Apresentação

Esta seção foca na produção da documentação, que deve refletir o rigor acadêmico e os padrões de qualidade do programa PET.

## 5.1 O Diário de Atividades (diario\_de\_atividades.pdf)

Este documento é a crônica do seu projeto. Deve ser um registro detalhado e reflexivo, não apenas uma lista de tarefas. A estrutura deve ser cronológica, com cada entrada contendo data, descrição da atividade, tempo dedicado, desafios e soluções.

O conteúdo estratégico é o que diferenciara um diário mediano de um excelente. É o espaço para *demonstrar* as qualidades avaliadas:

- **Demonstrando "Iniciativa" e "Propositividade":**
  - *Exemplo de Entrada:* "**05/10/2025:** Pesquisei sobre boas práticas de UI/UX para a exibição de listas de dados.<sup>25</sup> Decidi implementar um filtro simples para alternar entre grupos PET e ProPET, visando melhorar a usabilidade da página. Embora não seja um requisito explícito, essa funcionalidade agrega valor e está alinhada com o critério de 'propositividade'."
- **Demonstrando "Independência" e Resolução de Problemas:**
  - *Exemplo de Entrada:* "**10/10/2025:** Ao tentar carregar os dados, o PHP retornou um erro de null após o json\_decode(). Suspeitei de um erro de sintaxe no arquivo JSON. Utilizei uma ferramenta online de validação de JSON e identifiquei uma vírgula extra no final de uma lista. Corrigi o arquivo e o script passou a funcionar corretamente. O problema foi resolvido de forma autônoma."

## 5.2 O Relatório Formal (relatorio\_final.pdf em LaTeX)

Este é o principal documento escrito e deve seguir estritamente o modelo fornecido (modelo\_documento\_generico.pdf).<sup>1</sup> A utilização do LaTeX é obrigatória.<sup>1</sup> Para facilitar o trabalho, o uso de um editor online como o Overleaf é altamente recomendado, pois elimina a necessidade de instalação e configuração local.<sup>1</sup>

A estrutura do relatório deve ser a seguinte:

- **Capa:** Preencher conforme o modelo, com identificação da instituição, título do trabalho, autor e datas.
- **Corpo - Objetivo e Motivação:**
  - **Objetivo:** Definir em um parágrafo o escopo do trabalho, baseado na especificação



do projeto: "Desenvolver webpages dinâmicas para exibir informações sobre os grupos PET e ProPET da UFF, utilizando HTML, CSS e PHP, com dados armazenados em arquivos de formato JSON".<sup>1</sup>

- **Motivação:** Listar as motivações apresentadas no documento do projeto, como o interesse do PET-Tele em reestruturar suas páginas estáticas e atualizar os dados dos grupos da UFF.<sup>1</sup>
- **Desenvolvimento - Introdução Teórica:** Resumir a pesquisa realizada sobre as tecnologias. Explicar o modelo cliente-servidor, o papel do HTML para a estrutura, do CSS para o estilo, do PHP como linguagem de script do lado do servidor e do JSON como formato de armazenamento de dados.
- **Desenvolvimento - Método:** Esta é uma seção crucial para descrever *como* o trabalho foi feito. Detalhar a estratégia de coleta de dados, as decisões de design de UI/UX (incluir aqui o wireframe ou esboço da interface), a arquitetura de arquivos escolhida e a lógica de programação implementada.
- **Desenvolvimento - Resultados:** Apresentar o produto final. Esta seção deve conter os itens específicos de documentação da solução prática <sup>1</sup>:
  - Uma breve explicação sobre o uso da solução (como usar os filtros, por exemplo).
  - Imagens (screenshots) das webpages propostas, mostrando a aparência em desktop e, se possível, em mobile.
  - Se o projeto for hospedado online, indicar o URL de acesso público.
- **Desenvolvimento - Conclusões:** Fazer uma reflexão sobre o projeto. Discutir os aprendizados, as dificuldades superadas e as vantagens da solução dinâmica em comparação com uma abordagem estática. Sugerir possíveis melhorias futuras (ex: um painel administrativo para editar os dados).
- **Apêndices:** Incluir as listagens completas dos códigos (index.php, style.css) e trechos representativos dos arquivos JSON.<sup>1</sup>
- **Agradecimentos e Referências Bibliográficas:** Agradecer ao grupo PET-Tele pela oportunidade. Listar todas as fontes consultadas (documentação oficial, tutoriais, artigos) em um formato de citação padrão.

### 5.3 A Apresentação (apresentacao.pdf em Beamer)

A apresentação de slides deve ser um resumo conciso e visualmente atraente do trabalho realizado.<sup>1</sup> A sugestão de usar Beamer reforça a expectativa de um padrão acadêmico.

#### Estrutura Sugerida para os Slides:

1. **Slide de Título:** Título do projeto, nome do candidato, Processo Seletivo PET-Tele 2025-2.
2. **Introdução (1 slide):** Apresentar o objetivo do projeto: modernizar a exibição de dados

dos grupos PET/ProPET da UFF.

3. **Fundamentação Teórica (2-3 slides):** Resumir as tecnologias utilizadas (HTML, CSS, PHP, JSON) e como elas interagem no modelo cliente-servidor.
4. **Metodologia e Planejamento (2-3 slides):** Mostrar o processo: a estratégia de coleta de dados, a proposta de design (exibir o wireframe) e a arquitetura do projeto.
5. **Demonstração ao Vivo (transição para o navegador):** Este é o momento da demonstração prática de 5 minutos.
6. **Resultados e Funcionalidades (1-2 slides):** Após a demo, exibir screenshots da página final e destacar as principais características (ex: "Design Responsivo com Cards", "Carregamento Dinâmico de Dados via PHP/JSON", "Funcionalidade de Filtro Proposta").
7. **Conclusão (1 slide):** Resumir os principais aprendizados, o sucesso no cumprimento dos objetivos e o valor da solução desenvolvida.
8. **Agradecimentos e Perguntas (1 slide):** Um slide final para agradecer a atenção e abrir para perguntas.

## Parte 6: A Grande Final: Apresentação e Demonstração do Projeto

A apresentação final é o culminar de todo o trabalho. A estrutura de tempo é rigorosa: 60 minutos por candidato, divididos em 40 minutos para apresentação, 5 minutos para a demonstração prática e 5 minutos para perguntas.<sup>1</sup> A avaliação abrangerá a qualidade dos slides, a clareza da exposição, o domínio do conteúdo e a segurança do apresentador.<sup>1</sup>

### Estratégias para o Sucesso:

- **Ensaio e Gestão do Tempo:** Praticar a apresentação várias vezes é fundamental. Cronometre a fala para garantir que se encaixe nos 40 minutos. Ensaie a transição para a demonstração ao vivo e a própria demonstração para que não ultrapasse os 5 minutos.
- **Preparação Técnica:** No dia da apresentação, chegue com antecedência. Certifique-se de que o ambiente de desenvolvimento (XAMPP) está funcionando perfeitamente e que a página do projeto já está aberta no navegador. Tenha um plano de contingência, como um vídeo curto da aplicação funcionando, em caso de falhas técnicas inesperadas.
- **Antecipação de Perguntas:** Prepare-se para responder a perguntas prováveis, como:
  - "Por que você escolheu essa abordagem de design (cards)?"
  - "Qual foi o maior desafio técnico que você enfrentou e como o superou?"
  - "Se você tivesse mais uma semana, que outra funcionalidade adicionaria ao projeto?"
- **Conexão com os Critérios de Avaliação:** Durante a apresentação, conecte sutilmente as suas ações aos critérios de avaliação. Por exemplo, ao apresentar os filtros, pode-se

dizer: "Para exercitar a 'propositividade' solicitada, propus e implementei uma funcionalidade de filtro que, embora não estivesse no escopo mínimo, melhora significativamente a experiência do usuário ao navegar pelos dados."

## Apêndice: Cronograma Detalhado do Projeto

Este cronograma serve como uma ferramenta de gerenciamento de projeto, dividindo o trabalho em fases e tarefas gerenciáveis, alinhadas com os prazos oficiais do edital.<sup>1</sup>

**Tabela 2: Cronograma Detalhado do Projeto (19 de Setembro a 30 de Outubro de 2025)**

Fase	Período	Tarefas Principais	Entregáveis/Marcos
<b>1. Pesquisa e Planejamento</b>	19/09 - 29/09	<ul style="list-style-type: none"><li>- Estudo aprofundado de HTML, CSS, PHP, JSON.</li><li>- Início do Diário de Atividades.</li><li>- Coleta e análise de dados dos grupos PET/ProPET.<sup>14</sup></li><li>- Definição da estrutura JSON (Tabela 1).</li><li>- Esboço/wireframe da interface de usuário.</li><li>- Definição da arquitetura de arquivos.</li><li>- Preparação para a apresentação de entendimento.</li></ul>	<b>24/09: Reunião de Alocação de Projetos. 30/09 - 02/10: Apresentação "Entendimento do Projeto".</b>

<b>2. Desenvolvimento Central</b>	30/09 - 13/10	<ul style="list-style-type: none"> <li>- Configuração do ambiente XAMPP. - Criação e preenchimento dos 4 arquivos JSON. - Desenvolvimento da lógica PHP para leitura e união dos dados. - Criação da estrutura HTML básica com o loop foreach. - Início da redação do relatório em LaTeX (Capa, Objetivo, Motivação, Introdução Teórica). - Manutenção diária do Diário de Atividades.</li> </ul>	<b>14/10 e 16/10: Apresentação "Andamento do Projeto".</b>
<b>3. Estilização, Refinamento e Documentação</b>	15/10 - 25/10	<ul style="list-style-type: none"> <li>- Implementação completa do CSS para estilizar os cards. - Desenvolvimento da responsividade com media queries. - Implementação de funcionalidades extras (ex: filtros). - Testes completos em múltiplos navegadores. - Finalização do relatório em LaTeX (Método, Resultados, Conclusões, Apêndices com códigos). - Criação</li> </ul>	<ul style="list-style-type: none"> <li>- Versão final do código-fonte. - Versão final do relatório e diário.</li> </ul>

		dos slides da apresentação em Beamer. - Revisão e formatação final do Diário de Atividades.	
<b>4. Submissão e Preparação Final</b>	26/10 - 28/10	- Organizar todos os arquivos para entrega. - Enviar e-mail com todos os documentos até a data limite. - Ensaiai a apresentação e a demonstração ao vivo. - Preparar respostas para possíveis perguntas.	<b>26/10 (Domingo): PRAZO FINAL PARA SUBMISSÃO DOS DOCUMENTOS.</b>
<b>5. Apresentação Final</b>	29/10 - 30/10	- Realizar a apresentação final do projeto para a comissão de seleção.	<b>29/10 e 30/10: Apresentação Final.</b>
<b>Divulgação dos Resultados</b>	03/11	- Aguardar a comunicação oficial do resultado.	<b>03/11: Divulgação dos Resultados Finais.</b>

#### Referências citadas

1. Sel\_2025\_2\_Instrucoes\_Etapa\_Final (1).pdf
2. Primeiros passos com CSS - Aprendendo desenvolvimento web - MDN, acessado em setembro 27, 2025,  
[https://developer.mozilla.org/pt-BR/docs/Learn\\_web\\_development/Core/Styling\\_basics](https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Styling_basics)
3. HTML básico - Aprendendo desenvolvimento web - MDN, acessado em setembro 27, 2025,  
[https://developer.mozilla.org/pt-BR/docs/Learn\\_web\\_development/Getting\\_started/Your\\_first\\_website/Creating\\_the\\_content](https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content)

4. Introdução ao HTML - Aprendendo desenvolvimento web - MDN, acessado em setembro 27, 2025, [https://developer.mozilla.org/pt-BR/docs/Learn\\_web\\_development/Core/Structuring\\_content](https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Structuring_content)
5. CSS básico - Aprendendo desenvolvimento web | MDN - Mozilla, acessado em setembro 27, 2025, [https://developer.mozilla.org/pt-BR/docs/Learn\\_web\\_development/Getting\\_started/Your\\_first\\_website/Styling\\_the\\_content](https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Getting_started/Your_first_website/Styling_the_content)
6. HTML: Boas práticas em acessibilidade - Aprendendo desenvolvimento web | MDN, acessado em setembro 27, 2025, [https://developer.mozilla.org/pt-BR/docs/Learn\\_web\\_development/Core/Accessibility/HTML](https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Accessibility/HTML)
7. How to parse a JSON File in PHP? - GeeksforGeeks, acessado em setembro 27, 2025, <https://www.geeksforgeeks.org/php/how-to-parse-a-json-file-in-php/>
8. Read, Decode, Encode, Write JSON in PHP - nidup, acessado em setembro 27, 2025, <https://www.nidup.io/blog/manipulate-json-files-in-php/>
9. Como ler um arquivo Json - php - Stack Overflow em Português, acessado em setembro 27, 2025, <https://pt.stackoverflow.com/questions/76737/como-ler-um-arquivo-json>
10. Como ler JSON com PHP (json\_decode) - Hora de Codar, acessado em setembro 27, 2025, [https://horadecodar.com.br/como-ler-json-com-php-json\\_decode/](https://horadecodar.com.br/como-ler-json-com-php-json_decode/)
11. Um simples tutorial - Manual - PHP, acessado em setembro 27, 2025, [https://www.php.net/manual/pt\\_BR/tutorial.php](https://www.php.net/manual/pt_BR/tutorial.php)
12. Manual do PHP, acessado em setembro 27, 2025, [https://www.php.net/manual/pt\\_BR/index.php](https://www.php.net/manual/pt_BR/index.php)
13. O manual do PHP – Guia para iniciantes em PHP - freeCodeCamp, acessado em setembro 27, 2025, <https://www.freecodecamp.org/portuguese/news/o-manual-do-php-guia-para-iniciantes-em-php/>
14. Pet e Propet|Universidade Federal Fluminense, acessado em setembro 27, 2025, <https://www.uff.br/prograd/pet-e-propet/>
15. Contato – Programa de Pós-graduação em Engenharia de Produção - Universidade Federal Fluminense, acessado em setembro 27, 2025, <https://tpp-uff.com.br/contato/>
16. Card UI design: fundamentals and examples - Justinmind, acessado em setembro 27, 2025, <https://www.justinmind.com/ui-design/cards>
17. Card UI Design: The Best Practices and 25 Best Examples - Mockplus, acessado em setembro 27, 2025, <https://www.mockplus.com/blog/post/card-ui-design>
18. Card UI Design Examples and Best Practices for Product Owners - Eleken, acessado em setembro 27, 2025, <https://www.eleken.co/blog-posts/card-ui-examples-and-best-practices-for-product-owners>
19. XAMPP Installers and Downloads for Apache Friends, acessado em setembro 27, 2025, [https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html)
20. Como Instalar XAMPP no Windows - Solutio Web, acessado em setembro 27,

- 2025, <https://www.solutioweb.com.br/blog/como-instalar-xampp-no-windows>
21. Iniciando com PHP Como Configurar o XAMPP e Rodar seu Primeiro Código - YouTube, acessado em setembro 27, 2025, <https://www.youtube.com/watch?v=Owxf5uSrcpU>
  22. O guia do iniciante para o Git e o GitHub - freeCodeCamp, acessado em setembro 27, 2025, <https://www.freecodecamp.org/portuguese/news/o-guia-do-iniciante-para-o-git-e-o-github/>
  23. Introdução ao Git e GitHub para Iniciantes - Sérgio Toledo, acessado em setembro 27, 2025, <https://www.sergiotoledo.com.br/tutoriais/diversos/introducao-ao-git-e-github-para-iniciantes>
  24. Git e GitHub: Um Guia Básico Iniciantes - Brasil Code, acessado em setembro 27, 2025, <https://www.brasilcode.com.br/git-e-github-um-guia-basico-iniciantes/>
  25. Organização de Informações em UI Design: Estratégias e Técnicas para Organizar Informações em Interfaces - Awari, acessado em setembro 27, 2025, <https://awari.com.br/organizacao-de-informacoes-em-ui-design-estrategias-e-tecnicas-para-organizar-informacoes-em-interfaces/>
  26. Dashboards: um guia de boas práticas para UX e UI Designers - Blog Ateliware, acessado em setembro 27, 2025, <https://blog.ateliware.com/dashboard-design/>