

Especificação para Backend: Geração de Relatório PDF de Cronograma

1. Objetivo

Implementar a funcionalidade de geração de relatório PDF para um cronograma específico. O frontend já está preparado para chamar um endpoint da API, enviar o ID do cronograma e receber o arquivo PDF codificado em **base64**.

A principal requisito é que o PDF gerado seja **visualmente idêntico** ao layout definido em um template HTML já existente.

2. Endpoint a ser Implementado

- **Método:** **POST**
- **URL:** **/api/cronogramas/:id/pdf**
- **Parâmetro de URL:** **id** (o ID do cronograma a ser impresso).

3. Lógica de Negócio Detalhada

O endpoint deve executar os seguintes passos:

1. **Receber a Requisição:** Obter o **id** do cronograma a partir dos parâmetros da URL.
2. **Buscar Dados Completos:** Consultar o banco de dados para obter todos os dados do cronograma correspondente ao **id**, incluindo a **lista completa de todas as suas atividades associadas**.
3. **Carregar o Template HTML:** Ler o conteúdo do arquivo de template **CRONOGRAMAMESJUNHO.html**, que já está no projeto e define o layout exato do relatório.
4. **Preenchimento Dinâmico do HTML (Data Injection):** Esta é a etapa mais crítica. A lógica do backend deve substituir "placeholders" (ou manipular o DOM do HTML) com os dados do cronograma. Será necessário:
 - Injetar informações no cabeçalho: **nomeUBSF**, **enfermeiro**, **medico**, e o período (**mês/ano**).
 - **Distribuir as atividades na tabela:** O template tem uma tabela complexa que simula um calendário. É preciso mapear cada atividade para a sua célula **<td>** correta, com base na data e no **diaSemana** da atividade. Células sem atividades devem permanecer vazias.
5. **Converter HTML para PDF:** Utilizar uma biblioteca de conversão (como **Puppeteer** ou similar, que já foi citada no planejamento do projeto) para renderizar o HTML preenchido e convertê-lo em um arquivo PDF. É crucial que a biblioteca consiga renderizar o CSS do template corretamente.
6. **Codificar em Base64:** Converter o buffer do arquivo PDF gerado para uma string no formato **base64**.
7. **Retornar a Resposta:** Enviar a resposta para o frontend no formato JSON especificado abaixo.

4. Formato da Resposta da API

O frontend espera receber a resposta da API seguindo estritamente este formato:

Em caso de Sucesso (Status 200)

```
{
  "success": true,
  "message": "PDF gerado com sucesso.",
  "data": {
    "pdfBase64": "JVBERi0xLjQKJ..." // String longa representando o arquivo PDF
    em base64
  },
  "timestamp": "2024-07-28T10:30:00.000Z"
}
```

Em caso de Erro (Status 404, 500, etc.)

O frontend já está preparado para tratar as respostas de erro padrão da API.

```
{
  "success": false,
  "message": "Cronograma com ID 'xyz' não foi encontrado.",
  "data": null,
  "timestamp": "2024-07-28T10:30:00.000Z"
}
```

5. Resumo da Tarefa

O objetivo final é que o frontend possa chamar `POST /api/cronogramas/:id/pdf` e receber em troca uma string `base64` do PDF, perfeitamente formatado de acordo com o template HTML.

Qualquer dúvida sobre o template ou o fluxo, estou à disposição.