

# Projeto Final

## Módulo 1

Gerenciamento de Pacientes e Prontuários



# Descrição

- Crie uma aplicação para Gerenciamento de Pacientes e Prontuários
- Os dados deverão ser enviados e recuperados da API (A API será executada utilizando o **json-server**)
- As **únicas** bibliotecas externas permitidas no desenvolvimento do projeto são:
  - Bootstrap (conteúdo visto em sala: *flex-box, margin, padding, col e row*)
  - Font-awesome para os ícones
- Data de entrega: **02/05/2023**
- Os arquivos do projeto deverão estar em um repositório público no GitHub
- A aplicação deverá ser publicada na Vercel (e API no Render ou Railway)
- **Qualquer tipo de plágio não será tolerado. Em caso de cópia, o aluno será imediatamente desligado do curso .**

# Detalhes da Entrega

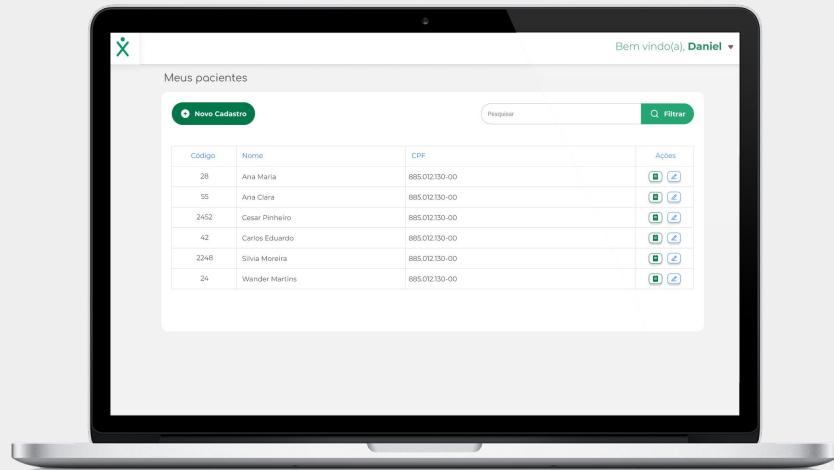
- Grave um vídeo para apresentar o seu projeto
  - Demonstre o funcionamento da sua aplicação (cadastro, edição, exclusão, listagem e funcionalidades extras, se houverem)
  - Apresente o código-fonte explicando o fluxo utilizado para o desenvolvimento
  - Opções de App para gravação: Loom (extensão do Chrome) ou Zoom
- Crie uma pasta com o seu nome completo na pasta Projeto Final correspondente à sua turma ([Projeto Final](#))
  - Inclua o link do GitHub, link da aplicação na Vercel e o vídeo (arquivo ou link)
- **Só serão aceitos arquivos incluídos até 23:59h do dia 02/05/2023 (terça-feira)**

# Plantões

- Durante o mês de projeto, os professores farão plantões de 18h às 19:30h para tirar dúvidas
- Não haverá chamada e a presença não é obrigatória
- Os plantões ocorrerão no mesmo link do Meet da respectiva turma

# Layout: Figma

Arquivo Aberto



# Funcionalidades

- Tela de Login
- Tela de Cadastre-se
- Listar Pacientes
  - Filtrar lista
- Cadastrar Paciente
- Editar Paciente
- Excluir Paciente
- Visualizar Prontuário
  - Filtrar lista
- Incluir sessão
- Incluir fato relevante

# Funcionalidades Obrigatórias

- **Tela de Login** (Layout estático)
- **Tela de Cadastre-se** (Layout estático)
- **Listar Pacientes** (Layout e comunicação com a API)
  - **Filtrar lista**
- **Cadastrar Paciente** (Layout e comunicação com a API)
- **Editar Paciente** (Layout e comunicação com a API)
- **Excluir Paciente** (Layout e comunicação com a API)
- **Visualizar Prontuário** (Layout estático)
  - Filtrar lista
- Incluir sessão
- Incluir fato relevante

# Funcionalidades Optativas

- **Tela de Login** (Comunicação com a API)
- **Tela de Cadastre-se** (Comunicação com a API)
- Listar Pacientes
  - Filtrar lista
- Cadastrar Paciente
- Editar Paciente
- Excluir Paciente
- **Visualizar Prontuário** (Comunicação com a API)
  - **Filtrar lista**
- **Incluir sessão** (Layout e comunicação com a API)
- **Incluir fato relevante** (Layout e comunicação com a API)



**As atividades  
obrigatórias  
irão compor**

**70%**

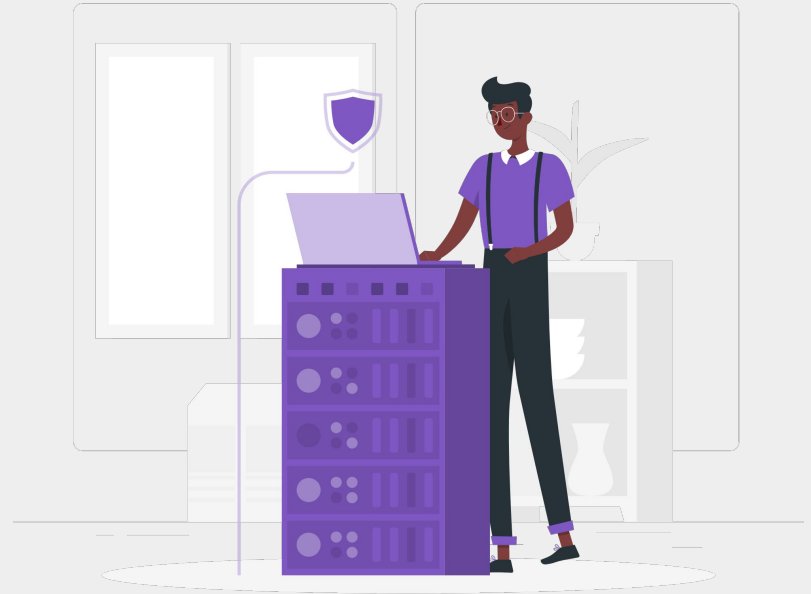
**da nota final**

**As atividades  
optativas  
irão compor**

**30%**

**da nota final**

# Publicação



# Vercel

- A Vercel será utilizada para hospedar **a aplicação front-end** (Gerenciador de Pacientes e Prontuários)
- O código na sua aplicação deve estar em um repositório no GitHub
- Crie uma conta no site da Vercel: [Sign Up – Vercel](#)
- Ao entrar clique em **Add new > Project**
- Conecte o seu GitHub e escolha o repositório na lista e clique em **Import**
- Na próxima tela clique em **Deploy**
- A Vercel fornecerá a URL para acesso à sua aplicação

# Render (API)

- O Render poderá ser utilizado para hospedar **a API** (a parte do json-server)
- Clone este repositório para o seu GitHub clicando na opção **Use this template:**  
<https://github.com/jesperorb/json-server-heroku>
- Defina no arquivo db.json o conteúdo base. Exemplo:

```
{  
  "tasks": []  
}
```

- Entre no site do Render: [Render](#)
- Clique **GET STARTED FOR FREE > Sign up for Render [GitHub]**
- Faça o login no seu GitHub

# Render (API)

- Clique em **New Web Service**
- Do lado direito da página, clique em **Configure account** abaixo do símbolo do GitHub, selecione a sua conta e clique em **Install** para vinculá-la ao Render
- Escolha o repositório que deseja publicar (o criado anteriormente) e clique em **Connect**
- Defina um nome para o *web service* e ao final da página clique em **Create Web Service**
- Após o sucesso da publicação o link localizado abaixo do nome do *web service* poderá ser usado
- Use o link gerado na sua aplicação front-end substituindo o <http://localhost:3000>

# Railway (API) – Outra opção

- O Railway poderá ser utilizado para hospedar **a API** (a parte do json-server)
- Clone este repositório para o seu GitHub clicando na opção **Use this template:**  
<https://github.com/jesperorb/json-server-heroku>
- Defina no arquivo db.json o conteúdo base. Exemplo:

```
{  
  "tasks": []  
}
```

- Entre no site do Railway: [Railway](#)
- Clique **Start a New Project > Deploy from GitHub repo > Login With GitHub**
- Faça o login no seu GitHub

# Railway (API)

- Clique em **Deploy from GitHub repo** novamente, escolha o repositório e clique em **Deploy Now**
- Após o sucesso da publicação clique em **Settings**, vá até **Domains** e clique em **Generate Domain**
- Use o link gerado na sua aplicação substituindo <http://localhost:3000>
- É possível que a plataforma exija a inclusão de um cartão de crédito caso a sua conta do GitHub não seja suficiente para a verificação (use um cartão virtual para que você possa excluí-lo após a entrega do projeto)
- Observe a quantidade de horas disponíveis na lateral superior direita para evitar custos
- Exclua sua aplicação do Railway após a gravação do vídeo para evitar custos

# Conteúdo Auxiliar





# Conteúdo Auxiliar

- Tratamento de erros
  - <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/try...catch>
- Formatação da data
  - Opção 1: `new Date(date.split('-').join('/'))`
  - Opção 2: `new Date(`${date}T00:00:00-03:00`).toLocaleDateString()`

# Sugestões



# Sugestões para implementações extras

- Responsividade
- Ordenação em cada coluna (crescente e decrescente)
- Paginação
- Auto-contraste
- **Documentação JSON-SERVER:** <https://github.com/typicode/json-server#routes>

**Obrigada!**

