

FullStack [Trindade]

Módulo 2 - Projeto Avaliativo

SUMÁRIO

1 INTRODUÇÃO	1
2 ENTREGA	2
3 REQUISITOS DA APLICAÇÃO	2
4 ROTEIRO DA APLICAÇÃO	2
4.1 ORGANIZAÇÃO DE PROJETO	3
4.2 FORMATO DO SISTEMA	3
4.3 EXEMPLOS DE APLICAÇÃO	7
4.4 DOCUMENTAÇÃO NO README.MD	11
4.5 GRAVAÇÃO DE VÍDEO	11
5 CRITÉRIOS DE AVALIAÇÃO	11
6 PLANO DE PROJETO	15

1 INTRODUÇÃO

A **LABMedicine LTDA**, empresa líder no segmento tecnológico para gestão hospitalar, está com um projeto novo intitulado **LABMedical**, um software audacioso para gestão de inventário médico. O seu perfil chamou a atenção dos gestores, para criar a aplicação Front-End do software, que deverá ser construída utilizando a biblioteca **React**.

Está animado para iniciar o desenvolvimento?

Leia atentamente os itens abaixo para ficar por dentro das regras de negócio e endpoints que devem ser criados nesta aplicação. Lembre-se também de ler atentamente as regras de entrega do projeto.

2 ENTREGA

O código deverá ser inserido e versionado no **GitHub** em modo privado, o **Trello** deverá ser criado em modo de acesso público, e o vídeo deverá ser inserido no **Google Drive** em modo **leitor** para qualquer pessoa com o link. Ambos os links deverão ser disponibilizados na tarefa **Módulo 2 - Projeto Avaliativo**, presente na semana 11 do AVA até o dia **26/07/2023** às **23h55.privado**

O repositório deverá ter as seguintes pessoas adicionadas:

- Robert Alexandre de Oliveira Santos - **robertsantosti**
- Operação DEVinHouse - **lab365-operacao**

Importante:

1. Não serão aceitos projetos submetidos **após a data limite da atividade**, e, ou **alterados** depois de entregues. Lembre-se de **não modificar** o código no GitHub até receber sua nota e feedback.
2. Não esqueça de **submeter os links no AVA**. Não serão aceitos projetos em que os links não tenham sido submetidos.

3 REQUISITOS DA APLICAÇÃO

A aplicação que deverá ser realizada **individualmente** deve contemplar os seguintes requisitos:

1. O sistema deverá ser desenvolvido utilizando a biblioteca **React**.
2. Você **deverá** modelar o layout da aplicação com os formatos, tipografias, cores e organização de layout que achar mais adequado.
 - a. Caso julgue necessário, poderá usar Bootstrap para estilização do *layout*.
3. Você deverá planejar o projeto utilizando o modelo *Kanban* na ferramenta **Trello**;
4. Você deverá gravar um **vídeo** de apresentação do sistema.
5. O sistema deve seguir o **Roteiro da Aplicação**.

4 ROTEIRO DA APLICAÇÃO

A **LABMedicine LTDA**, deseja gerenciar o inventário médico de hospitais e postos de saúde, através de um sistema de gestão.

4.1 ORGANIZAÇÃO DE PROJETO

Crie um quadro no Trello para organizar as tarefas de criação do projeto. Utilize o modelo *Kanban* com as seguintes listas:

- *Backlog*
- *To-Do*
- *Doing*
- *Blocked*
- *Review*
- *Done*

Crie todas as tarefas necessárias ao desenvolvimento do projeto na lista *Backlog* e arraste cada cartão por vez entre as listas durante o desenvolvimento do projeto. Dica: Você pode criar etiquetas/*tags*, *checklists* e prazos para ajudar na organização das tarefas.

Não utilize o Trello de exercícios, crie um novo Trello com acesso público para o projeto, utilizando o seu e-mail de estudante. Compartilhe o link do Trello na tarefa de submissão no AVA.

4.2 FORMATO DO SISTEMA

A aplicação que deverá ser realizada **individualmente** deve contemplar os seguintes requisitos:

1. Ser construída utilizando a biblioteca React:
 - a. Utilizar roteamento para gerenciamento de páginas.
 - b. Gerenciar o estado global quando necessário.
 - c. Utilizar animações de loading e transições entre páginas.
 - d. Utilizar consumo da API [ViaCEP](#) para cadastro de endereço.
2. Utilizar favicon, título de página e demais assets.
3. Utilizar o localStorage ou JSON Server para guardar as informações cadastradas.
4. Utilizar o GitHub como versionador de código:
 - a. Utilização do padrão baseado em GitFlow com main, features e releases.
 - b. Utilização de commits curtos e concisos.

As páginas e funcionalidades devem seguir o formato:

1. **Página de Login:** Contendo um formulário com campos de email e senha, botões para criar uma conta, entrar na aplicação, e um link para resetar a senha.

- a. O botão de criar conta deve abrir uma página ou modal com formulário para criar um usuário, sendo exigido e-mail válido e senha com mais de 8 caracteres. Para a senha utilize dois campos e valide se a mesma é equivalente. Salve o usuário no localStorage (ou utilize JSON Server).
 - b. A funcionalidade de "esqueceu sua senha" não precisa ser implementada. Caso não seja implementada, utilize um *pop-up*, *toast* ou *alert* para avisar que esta funcionalidade está em construção.
 - c. O botão de entrar deve direcionar o usuário para a página de Início, caso o e-mail e senha sejam válidos e estejam cadastrados.
- 2. **Menu Lateral:** Com exceção da página de login, deverá estar presente em todas as outras páginas, apresentando todos os botões de navegação.
 - a. O botão de sair deve 'deslogar' e direcionar o usuário para a página de login.
 - b. Deverá existir um botão para esconder e mostrar o menu lateral.
- 3. **Toolbar:** Com exceção da página de login, deverá estar presente em todas as outras páginas, apresentando o título da página, nome do usuário (médico) e ícone para representar a foto do usuário.
 - a. Opcionalmente poderá ser adicionado outras funcionalidades como *dropdown* e botão de sair, por exemplo.
- 4. **Página de Início (Estatísticas e Informações):** Deve conter as estatísticas do sistema e busca de pacientes.
 - a. As estatísticas devem ser ao menos a quantidade de pacientes cadastrados, quantidade de consultas cadastradas e a quantidade de exames. Você está livre para adicionar novas estatísticas caso ache necessário.
 - b. Deverá existir um campo para buscar pacientes pelo nome, telefone ou e-mail.
 - c. A listagem de pacientes deve mostrar um *card* para cada um com ao menos um ícone, nome completo, idade, contato (telefone ou e-mail), plano de saúde e botão para ver mais informações.
 - d. Ao clicar no *card* (ou em "ver mais"), a página de cadastro de paciente deve ser aberta, com os botões de edição, deleção e salvamento ativados. Nesta página, o usuário (médico) poderá editar as informações ou deletar um paciente caso o mesmo não possua consultas e exames vinculados.
- 5. **Página de Cadastro de Paciente:** Deve conter um formulário para cadastro de paciente com botões para editar, deletar e salvar.
 - a. Durante o cadastro, os botões de editar e deletar devem ficar desativados (desabilitados).

b. Campos do formulário:

- i. Nome Completo: Obrigatório, com máximo e mínimo de 50 e 5 caracteres, respectivamente.
 - ii. Gênero: Obrigatório com *dropdown* de opções pré-definidas.
 - iii. Data de Nascimento: Obrigatório, data válida.
 - iv. CPF: Obrigatório com o formato 000.000.000-00
 - v. RG com órgão expedidor: Obrigatório, com máximo de 20 caracteres.
 - vi. Estado Civil: Obrigatório com *dropdown* de opções pré-definidas.
 - vii. Telefone: Obrigatório com o formato (99) 9 9999-99999
 - viii. E-mail: Não obrigatório e com validação.
 - ix. Naturalidade: Obrigatório, com máximo e mínimo de 50 e 5 caracteres, respectivamente.
 - x. Contato de Emergência: Obrigatório com o formato (99) 9 9999-99999
 - xi. Lista de Alergias: Não obrigatório.
 - xii. Lista de Cuidados Específicos: Não obrigatório.
 - xiii. Convênio: Não obrigatório.
 - xiv. Número do Convênio: Não obrigatório.
 - xv. Validade do Convênio: Não obrigatório.
 - xvi. Endereço: Cep, Cidade, Estado, Logradouro, Número, Complemento, Bairro e Ponto de Referência.
- c. Deverá utilizar a API do ViaCEP para buscar os dados de endereço.
- d. Deverá verificar os dados informados antes de cadastrar.
- e. Deverá criar um identificador único para cada paciente cadastrado.
- f. Deverá apresentar animação ao salvar.

6. **Página de Cadastro de Consulta:** Deve conter uma busca de paciente e um formulário para cadastro de consulta com botões editar, deletar e salvar.

a. Durante o cadastro, os botões de editar e deletar devem ficar desativados (desabilitados).

b. Campos do formulário:

- i. Motivo da consulta: Obrigatório, com máximo e mínimo de 60 e 6 caracteres, respectivamente.
- ii. Data da Consulta: Obrigatório, buscando data atual do sistema e liberando para edição.
- iii. Horário da Consulta: Obrigatório, buscando horário atual do sistema e liberando para edição.
- iv. Descrição do Problema: Obrigatório, com máximo e mínimo de 1000 e 15 caracteres, respectivamente.
- v. Medicação Receitada: Não obrigatório.

- vi. Dosagem e Precauções: Obrigatório, com máximo e mínimo de 250 e 15 caracteres, respectivamente.
 - c. Deverá verificar os dados informados antes de cadastrar.
 - d. Deverá criar um identificador único para cada consulta cadastrada.
 - e. Deverá apresentar animação ao salvar.
- 7. **Página de Cadastro de Exame:** Deve conter uma busca de paciente e um formulário para cadastro de consulta com botões editar, deletar e salvar.
 - a. Durante o cadastro, os botões de editar e deletar devem ficar desativados (desabilitados).
 - b. Campos do formulário:
 - i. Nome do Exame: Obrigatório, com máximo e mínimo de 50 e 5 caracteres, respectivamente.
 - ii. Data do Exame: Obrigatório, buscando horário atual do sistema e liberando para edição.
 - iii. Horário do Exame: Obrigatório, buscando horário atual do sistema e liberando para edição.
 - iv. Tipo do Exame: Obrigatório, com máximo e mínimo de 30 e 5 caracteres, respectivamente.
 - v. Laboratório: Obrigatório, com máximo e mínimo de 30 e 5 caracteres, respectivamente.
 - vi. URL do Documento: Não obrigatório.
 - vii. Resultados: Obrigatório, com máximo e mínimo de 1000 e 15 caracteres, respectivamente.
 - c. Deverá verificar os dados informados antes de cadastrar.
 - d. Deverá criar um identificador único para cada exame cadastrado.
 - e. Deverá apresentar animação ao salvar.
- 8. **Página de Listagem de Prontuário:** Deverá possuir uma barra de pesquisa, listagem de pacientes cadastrados e botão para abrir os detalhes.
 - a. A barra de busca deverá buscar o paciente pelo nome ou identificador cadastrado.
 - b. Ao clicar no *card* do paciente, uma nova página deve ser aberta mostrando as informações cadastradas, e o histórico de consultas e exames.
- 9. **Página de Prontuário de Paciente:** Deverá possuir informações concisas e histórico de consultas e exames.
 - a. As informações a serem exibidas são: nome do paciente, convênio, contato de emergência, lista de alergias e lista de cuidados específicos.

- b. O histórico de exames e consultas deve mostrar todas as consultas e exames realizados pelo paciente, sendo mostrados em ordem cronológica.
- c. Nas consultas, deve haver todas as informações cadastradas, dando destaque ao motivo, data e hora. Também deve haver um botão de editar, que quando clicado, abrirá a página de cadastro da consulta, com os botões de edição, deleção e salvamento ativados. Nesta página, o usuário (médico) poderá editar as informações ou deletar uma consulta.
- d. Nos exames, deve haver todas as informações cadastradas, dando destaque ao nome, laboratório, data, hora e anexo. Também deve haver um botão de editar, que quando clicado, abrirá a página de cadastro de exame, com os botões de edição, deleção e salvamento ativados. Nesta página, o usuário (médico) poderá editar as informações ou deletar um exame.

4.3 EXEMPLOS DE APLICAÇÃO

A aplicação deverá conter os requisitos apresentados anteriormente, sendo codificada em **html**, **css** e **javascript** através da biblioteca **React**, também deverá ser utilizado **markdown** para o readme.md.

As imagens a seguir demonstram **exemplos** da aplicação que deverá ser desenvolvida. Você deverá inovar e criar o seu próprio layout que **deve necessariamente** ser diferente do apresentado no exemplo abaixo. A estrutura de páginas pode ser baseada no exemplo, mas os ícones, tipografias, cores, organização, formas e etc, devem ser implementadas de forma diferente.

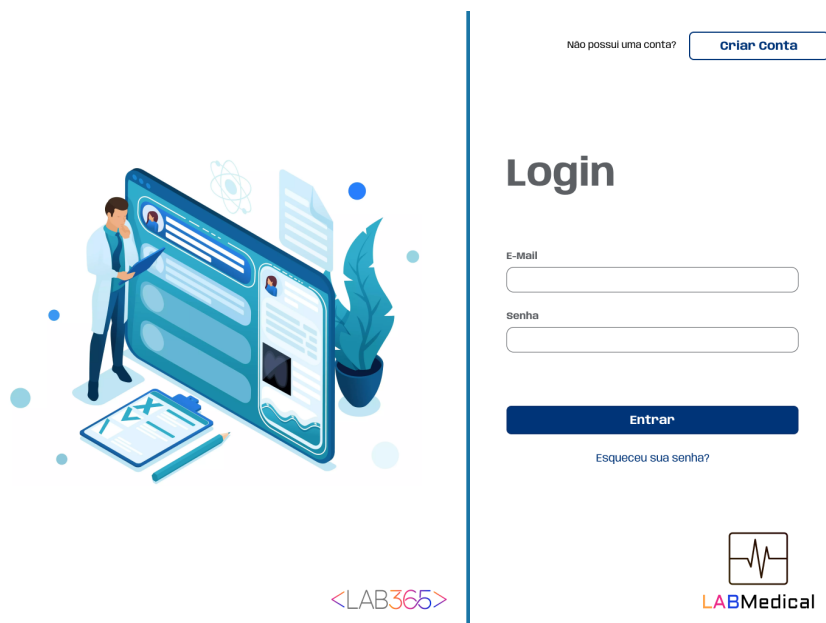


Imagem 1: Exemplo de página de Login



Imagem 2: Exemplo de página de Início (Estatística e Informações)

The form is titled 'CADASTRO DE PACIENTE' and includes a user profile 'Cesar Abascal'. It contains several sections for data entry:

Preencha os campos para cadastrar

Editar Deletar Salvar

Identificação

Nome Completo: Gênero: Data Nascimento:

CPF: RG: Estado Civil:

Telefone: E-Mail: Naturalidade:

Convênio

Convênio: Número da Carteira: Validade:


Dados de Endereço

CEP: Cidade: Estado:

Logradouro: Número:

Complemento: Bairro: Ponto de Referência:

Imagem 3: Exemplo de página de Cadastro de Paciente


LABMedical

Genal

INICIO

SAIR

Pacientes

+ CADASTRAR

LISTAR PRONTUÁRIO

Exames

+ CADASTRAR CONSULTA

+ CADASTRAR EXAME

Encolher Menu

CADASTRO DE CONSULTA

Cesar Abascal

Encontre o paciente

Buscar

Consulta de [Nome Paciente]

Editar

Deletar

Salvar

Motivo da Consulta

Data da Consulta

Hora da Consulta

dd/mm/aaaa


hh:mm:ss

Descrição do Problema

Medicação Receitada

Dosagem e Precauções

Imagem 4: Exemplo de página de Cadastro de Consulta


LABMedical

Genal

INICIO

SAIR

Pacientes

+ CADASTRAR

LISTAR PRONTUÁRIO

Exames

+ CADASTRAR CONSULTA

+ CADASTRAR EXAME

Encolher Menu

CADASTRO DE EXAME

Cesar Abascal

Encontre o paciente

Buscar

Exame de [Nome Paciente]

Editar

Deletar

Salvar

Nome do Exame

Data da Exame

Hora da Exame

dd/mm/aaaa

hh:mm:ss

Tipo do Exame

Laboratório

URL do Documento do Exame

Resultado do Exame

Imagem 5: Exemplo de página de Cadastro de Exame

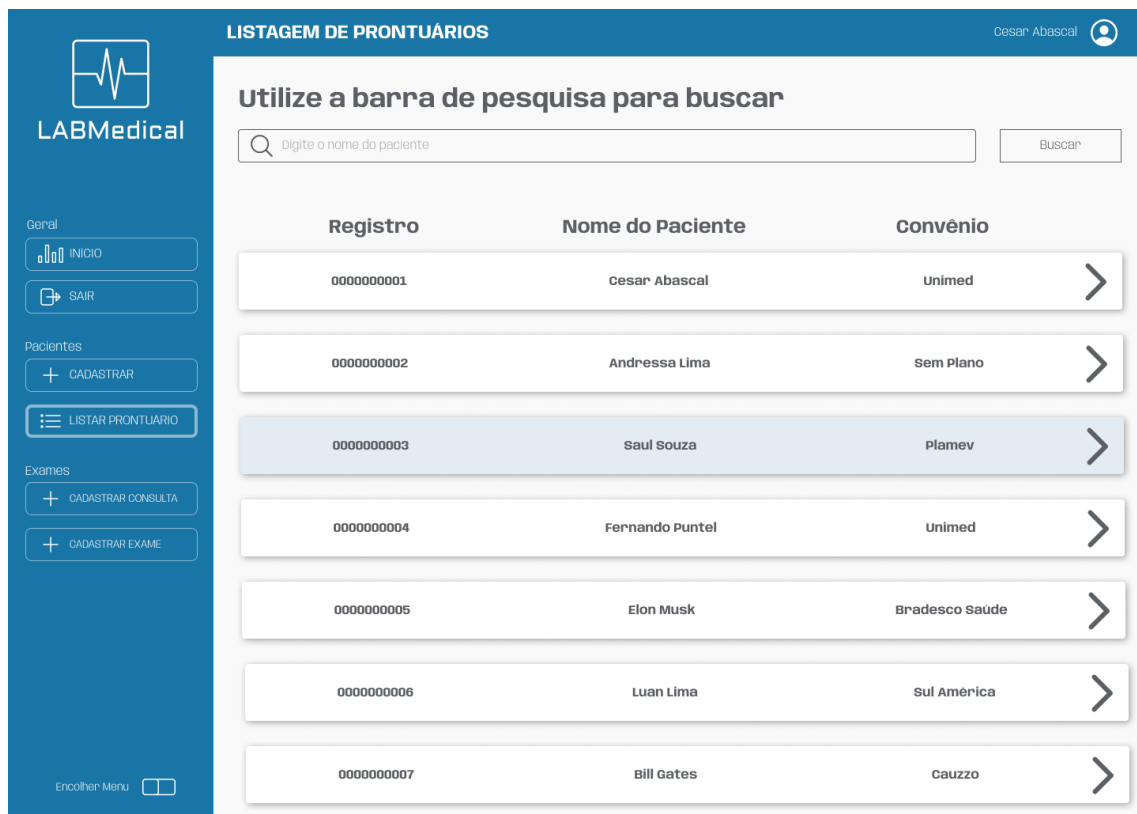


Imagem 6: Exemplo de página de Listagem de Prontuários de Paciente

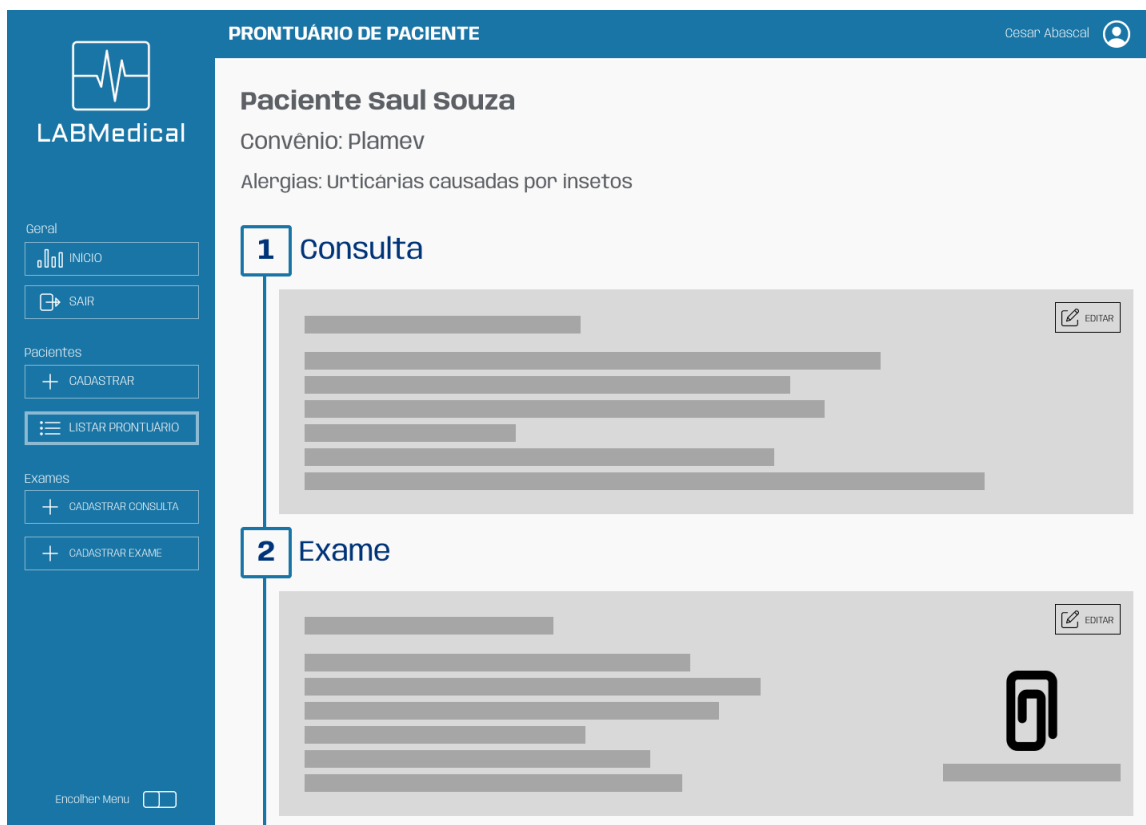


Imagem 7: Exemplo de página de Prontuário de Paciente

4.4 DOCUMENTAÇÃO NO README.MD

Crie um arquivo `readme.md` no repositório do seu projeto no GitHub, para documentar a sua solução, bem como demonstrar as técnicas e linguagens utilizadas, além do escopo do projeto e como o usuário pode executar o seu sistema.

Algumas dicas interessantes para utilizar na criação do seu portfólio são:

- Criar um nome para o seu software;
- Descrever qual o problema ele resolve;
- Descrever quais técnicas e tecnologias utilizadas. Aqui você também pode inserir alguma imagem ou diagrama para melhor entendimento;
- Descrever como executar;
- Descrever quais melhorias podem ser aplicadas;
- Entre outras coisas.

4.5 GRAVAÇÃO DE VÍDEO

Além do desenvolvimento deste sistema você deverá gravar um vídeo, com tempo **máximo** de 5 minutos, abordando os seguintes questionamentos:

- Qual o objetivo do sistema? E demonstração de funcionamento.
- O que deve ser realizado para executar o sistema?
- Como você organizou as tarefas antes de começar a desenvolver?
- Quais as *branches* você criou e quais os objetivos para cada uma?
- Você acha que faltou algo no seu código que você poderia melhorar?

Você poderá gravar na vertical ou na horizontal. É importante que apareça seu rosto e esteja em um local com boa iluminação. Para realizar a entrega do vídeo, coloque em uma pasta do **Google Drive** em modo leitor para qualquer pessoa com o link, e compartilhe o mesmo na submissão do projeto no AVA. Uma dica interessante é você inserir o vídeo no `readme.md` do seu projeto no repositório do GitHub.

5 CRITÉRIOS DE AVALIAÇÃO

A tabela abaixo apresenta os critérios que serão avaliados durante a correção do projeto. O mesmo possui variação de nota de 0 (zero) a 10 (dez) como nota mínima e máxima, e possui peso de **60% sobre a avaliação do módulo**.

Serão **desconsiderados e atribuída a nota 0 (zero)** os projetos que apresentarem plágio de soluções encontradas na internet ou de outros colegas. Lembre-se: Você está livre para utilizar outras soluções como base, mas **não é permitida** a cópia.

Organização e Apresentação do Projeto

Nº	Critério de Avaliação	0	0,65
1	Criou um Trello e cartões para organização do projeto?	Não criou ou não submeteu o link (acesso) ao Trello.	Criou um Trello, cadastrando todas as atividades necessárias no <i>backlog</i> , e as arrastou uma a uma em cada uma das listas durante o desenvolvimento do projeto, conforme elencado no item 4.1
2	No GitHub foi aberto um branch separado para cada etapa, com mensagens do commit claras e adequadas?	Os commits foram realizados diretamente na master/main, e não foi usado mensagens claras.	Foram abertas branches para cada funcionalidades, e os commits foram realizados de forma concisa com mensagens adequadas.
3	Criou uma documentação com readme.md?	Não criou a documentação.	Criou uma documentação completa com ao menos todos os tópicos sugeridos no item 4.4.
Nº	Critério de Avaliação	0	1,05
4	Realizou a gravação de um vídeo?	Não foi realizada a gravação do vídeo.	Gravou o vídeo e abordou todos os tópicos listados no item 4.5.

Desenvolvimento adequado da Aplicação

Nº	Critério	0	0,25	0,70
5	Inovou, construindo um layout agradável, responsivo e com acessibilidade?	Não inovou e construiu um layout sem responsividade e acessibilidade.	Inovou, construindo um layout agradável, personalizado e responsivo.	Inovou, construindo um layout agradável, personalizado, responsivo e com acessibilidade.
6	Estruturou o projeto, separando os recursos de acordo com o padrão de projetos?	Não foi efetuado a separação das páginas e componentes e a criação do router e store estão sendo feitas direto no main.	Foi efetuado a separação parcial de páginas e componentes ou inicialização do router e store em arquivos separados, com poucos erros na padronização.	Foi efetuado a separação de views e componentes, criação da store e router em seus devidos diretórios, separando regras de negócios e criação, sempre que necessário, em arquivos específicos.

7	Desenvolveu a página de login com a funcionalidade de autenticação funcional e com devidos tratamentos no formulário. Além de conter todos os elementos descritos no requisitos como acesso à página de cadastro, login e esqueci a senha?	Não desenvolveu a página ou entregou sem estar funcional.	Desenvolveu a página com a autenticação funcional porém com falhas na validação do formulário, alertas ou falta de redirecionamento.	Desenvolveu a página de login com todos os elementos do requisito, autenticação funcional com redirecionamento, alertas de avisos e falhas, além criar o guard com redirecionamento para página inicial se usuário já estiver autenticado.
8	Criou o template da aplicação, contendo o Menu Lateral e Toolbar funcional de acordo com os requisitos?	Não desenvolveu o template para a aplicação.	Desenvolveu o template, porém sem o Menu Lateral ou Toolbar ou com algum dos elementos desses componentes não funcionais.	Desenvolveu o template utilizando Menu Lateral e Toolbar, com todas as funcionalidades de acordo com os requisitos e com o guard permitindo o acesso somente com autenticação.
9	Desenvolveu a página de início contendo as estatísticas, listagem de pacientes, pesquisa e acesso a um modal ou página de edição do paciente?	Não desenvolveu a página ou apenas efetuou a criação da página com dados estáticos.	Desenvolveu a página de início, com todas as estatísticas e listagem de pacientes.	Desenvolveu a página de início, com todas as estatísticas, listagem de pacientes, pesquisa e acesso a mais detalhes, edição e deleção.
10	Desenvolveu a página de cadastro de paciente de acordo com os requisitos, efetuando validação no formulário e armazenando os dados no localStorage (ou JSON Server)?	Não desenvolveu a página ou não está funcional.	Desenvolveu parcialmente a página ou com erros de validação, não salvando corretamente no localStorage ou não efetuando a busca de CEP com a API ViaCep.	Desenvolveu a página de cadastro de paciente funcional, com todas as validações, efetuando consulta na API, guardando os registros no localStorage e permitindo edição e deleção corretamente.

11	Desenvolveu a página de cadastro de consulta, guardando as informações no localStorage (ou JSON Server), efetuando as devidas validações de formulário?	Não desenvolveu a página ou não está funcional.	Desenvolveu a página de cadastro, armazenando a consulta no localStorage, porém com falhas na validação.	Desenvolveu a página de cadastro com todas as validações funcionais, armazenando a consulta no localStorage utilizando a regra de negócio, além de retornar para o usuário mensagem de conclusão.
12	Desenvolveu a página de cadastro de exame, guardando as informações no localStorage (ou JSON Server), efetuando as devidas validações de formulário?	Não desenvolveu a página ou não está funcional.	Desenvolveu a página de cadastro, armazenando o exame no localStorage, porém com falhas na validação.	Desenvolveu a página de cadastro com todas as validações funcionais, armazenando o exame no localStorage utilizando a regra de negócio, além de retornar para o usuário mensagem de conclusão.
13	Desenvolveu a página de listagem de prontuário com filtro?	Não desenvolveu a página de listagem ou entregou com dados estáticos.	Desenvolveu a página de listagem, porém não buscou os dados corretamente ou não aplicou o filtro.	Desenvolveu a página de listagem com todos os requisitos funcionais.
14	Desenvolveu a página de detalhes do prontuário?	Não desenvolveu a página ou não está funcional.	Desenvolveu a página, porém a mesma não possui os botões de edição e anexos.	Desenvolveu a página totalmente funcional, seguindo todas as especificações solicitadas.

6 PLANO DE PROJETO

Ao construir a aplicação proposta, o aluno estará colocando em prática os aprendizados em:

- **Versionamento:** O que é versionamento de código, aplicações e utilização no Github.
- **HTML e CSS:** HTML5 (elementos semânticos) e CSS (seletores, principais estilos, layouts e flexbox).
- **JavaScript:** Variáveis, Tipos de Dados, Estrutura de Controle de Fluxo (condicional e repetição), Operadores (aritméticos, lógicos e relacionais), Funções, ES6+, Escopo, Arrow Functions, Operadores Rest e Spread, Funções de Arrays, Manipulação do DOM, Utilização de Seletores, Eventos, Classes, Atributos, Encapsulamento, Herança, Polimorfismo, Módulos, Funções Assíncronas, JSON, LocalStorage, Interval e Timeout.
- **React:** React básico, Renderização de Componentes, Props, Prop Types, Hooks Principais (useState, useEffect e useRef), eventos, renderização de listas, Hooks Avançados (useReducer e useContext), React Router, Formulários, Prop Drilling, Composition, Estilos (Styled Components e Bootstrap), Developer Tools e Deploy.