

# FullStack [Education]

## Módulo 2 - Projeto Avaliativo

### SUMÁRIO

1 INTRODUÇÃO	1
2 ENTREGA	2
3 REQUISITOS DA APLICAÇÃO	2
4 ROTEIRO DA APLICAÇÃO	3
4.1 FORMATO DO SISTEMA	3
4.2 EXEMPLO DE APLICAÇÃO	7
4.3 DOCUMENTAÇÃO NO README.MD	8
4.4 USO DO TRELLO	8
4.5 GRAVAÇÃO DE VÍDEO	8
5 CRITÉRIOS DE AVALIAÇÃO	9
6 PLANO DE PROJETO	12

### 1 INTRODUÇÃO

O **LAB365 Developer**, empresa líder no segmento tecnológico de gestão estudantil, está com um novo projeto intitulado **labPCP**, um software audacioso para gestão de processos educacionais. O seu perfil chamou a atenção dos gestores, para criar a aplicação Front-End do software, que deverá ser construída utilizando o *framework* **Angular**.

Está animado para iniciar o desenvolvimento?

Leia **atentamente** os itens abaixo para ficar por dentro das regras de negócio e rotas/telas que devem ser criadas na 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 planejamento deverá ser realizado em um quadro *Kanban* no **Trello**, e o vídeo deverá ser inserido no **Google Drive** em modo leitor para qualquer pessoa com o link. Os **três** links deverão ser disponibilizados na tarefa **Módulo 2 - Projeto Avaliativo**, presente na semana 11 do AVA até o dia **14/08/2024** às **22h (sem possibilidade de extensão de prazo)**.

O repositório deverá ser privado, com as seguintes pessoas adicionadas:

- Docente Eduardo Rodrigues da Silva - **silva-eduardrs**
- Docente Rayane Cristina de Oliveira Silva - **raydevcp20**
- Operação LAB365 - **lab365-operacao**

Também deve ser adicionado ao quadro trello os seguintes emails:

- Docente Rayane Cristina de Oliveira Silva - **rayanecristina.19@gmail.com**
- Docente Eduardo Rodrigues da Silva - **silva.eduardrs@gmail.com**

Não serão aceitos projetos submetidos **após a data limite da atividade**, e, ou alterados depois de entregues.

### Importante:

1. Não modifique o seu projeto antes de receber a nota e feedback. Caso contrário, o mesmo não será corrigido.
2. Não esqueça de **submeter os links no AVA**. Não serão aceitos projetos em que os links não tenham sido submetidos.
3. Crie um quadro Trello **diferente do quadro de exercícios**. No quadro de exercícios, os cartões diferentes serão deletados.

### 3 REQUISITOS DA APLICAÇÃO

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

1. O sistema deverá ser desenvolvido utilizando o *framework* **Angular**.
2. O sistema deve seguir o **Roteiro da Aplicação**.
3. Você deverá **modelar o layout** da aplicação com os formatos, tipografias, cores e organização de layout que achar melhor.
  - a. Caso julgue necessário, poderá usar Bootstrap, MUI (Material UI) e/ou outra ferramenta para estilização do *layout*.
4. Você deverá se preocupar com a **usabilidade** da aplicação.
  - a. Pense no contraste das cores para não atrapalhar na visualização.
5. Você deverá utilizar o GitHub como versionador de código.
6. Você deverá organizar as tarefas a serem realizadas no projeto, em um quadro no **Trello**. Para isso você **deverá criar um novo** quadro em sua conta de estudante (**não** use o quadro de exercícios). Lembre-se que o desenvolvimento ocorre de forma mais fluída quando já se tem o planejamento prévio daquilo que será codificado.
7. Você deverá **gravar um vídeo** de apresentação do sistema.

## 4 ROTEIRO DA APLICAÇÃO

A aplicação deverá conter os requisitos apresentados anteriormente, sendo codificada em **TypeScript** através do *framework* **Angular**, também deverá ser utilizado **markdown** para descrever o projeto no `readme.md`.

### 4.1 FORMATO DO SISTEMA

A aplicação deverá seguir o seguinte formato:

1. Ser construída utilizando o *framework* **Angular**:
  - a. Utilizar roteamento para gerenciamento de páginas.
  - b. Gerenciar o estado global quando necessário.
  - c. Utilizar animações de loading e ou transições entre páginas.
    - i. Exemplo: Spinner de loading
  - 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*, *develop*, *features* e *releases* (*release é opcional*).
  - b. Utilização de commits curtos e concisos.
5. Utilizar Trello para organização das tarefas a serem realizadas:
  - a. Criar um quadro com estrutura *kanban* (*backlog*, *todo*, *doing*, *blocked*, *review* e *done*) para organização das tarefas.
  - b. Criar cartões com tarefas a serem realizadas e arrastar cada um conforme a necessidade.
6. Possuir as seguintes páginas e funcionalidades:
  - a. **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.
    - i. Crie um mock inicial com alguns usuários para fazer o login. Este mock deve conter usuários com os perfis: Administrador, Docente e Aluno.
    - ii. A funcionalidade "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.
    - iii. A funcionalidade "criar uma conta" não precisa ser implementada. Caso não seja implementada, utilize um *pop-up*, *toast* ou *alert* para avisar que esta funcionalidade está em construção.

- iv. O botão de entrar deve direcionar o usuário para a página de Início, caso o e-mail e senha sejam válidos e estejam cadastrados, caso contrário, deve ser sinalizado “Usuário e/ou senha incorretos” através de um *pop-up*, *toast* ou *alert*.
- b. **Menu Lateral:** Deverá estar presente em todas as outras páginas, com exceção da página de login, apresentando os seguintes botões de navegação:
  - i. O botão de sair deve 'deslogar' e redirecionar o usuário para a página de login.
  - ii. Deverá conter botões de opções de navegação para as outras telas do portal:
    1. Início: redireciona para a “*Página de Início*”, item d ou item e, a depender do perfil de acesso do usuário.
    2. Cadastro Docente: redireciona para a “*Página de Cadastro de Docente*”, item f. Visível apenas para usuários Administradores.
    3. Cadastro de Aluno: redireciona para a “*Página de Cadastro de Aluno*”, item g. Visível apenas para usuários Administradores.
    4. Cadastro de Turma: redireciona para a “*Página de Cadastro de Turma*”, item h. Visível apenas para usuários Administradores e Docentes.
    5. Cadastro de Avaliação: redireciona para a “*Página de Cadastro de Avaliação/Nota*”, item i. Visível apenas para usuários Administradores e Docentes. Página de Listagem de Docentes
    6. Listagem de Docentes: redireciona para a “*Página de Listagem de Docentes*”, item j. Visível apenas para usuários Administradores.
    7. Notas: redireciona para a “*Página de Notas do Aluno*”, item k. Visível apenas para usuários Alunos.
  - iii. Um botão para recolher/esconder e expandir/mostrar o menu lateral.
- c. **Toolbar:** Deverá estar presente em todas as outras páginas, com exceção da página de login, apresentando o título da página, nome do usuário logado e ícone para representar a foto do usuário.
  - i. Opcionalmente poderá ser adicionado outras funcionalidades como *dropdown* e botão de sair, por exemplo.
- d. **Página de Início (Administradores e Docentes):** Deve conter as seguintes funcionalidades:
  - i. Uma seção com as estatísticas do sistema. Elas devem representar ao menos a quantidade de alunos cadastrados, quantidade de docentes cadastrados e a quantidade de turmas.  
Você está livre para adicionar novas estatísticas caso ache necessário.
    1. Esta seção somente será apresentada para usuários Administradores
  - ii. Deverá existir um campo para buscar alunos pelo nome, telefone ou e-mail.
  - iii. A listagem de alunos deve mostrar um *card* para cada um com ao menos um ícone, nome completo, idade, contato (telefone ou e-mail) e um botão:

1. Caso o perfil do usuário seja Administrador: o botão possuirá o texto “*ver mais*” e, ao ser clicado, o usuário deve ser redirecionado para a “*Página de Cadastro de Aluno*”, funcionalidade especificada no *item g*, que deve ser aberta, com os botões de edição, deleção e salvamento ativados. Nesta página, o usuário poderá editar as informações ou deletar um aluno caso o mesmo não possua turmas e avaliações vinculadas. Para isso, a tela deve preencher os campos com as informações daquele aluno.
  2. Caso o perfil do usuário seja Docente: o botão possuirá o texto “*lançar nota*” e, ao ser clicado, o usuário deve ser redirecionado para a “*Página de Cadastro de Avaliação/Nota*”, funcionalidade especificada no *item i*.
- e. **Página de Início (Alunos)**: A tela inicial para o aluno deverá conter duas sessões:
- i. Sessão Minhas Avaliações: Nesta sessão serão exibidas as últimas 3 avaliações realizadas pelo aluno, sendo exibido o nome da avaliação, a matéria e a data de realização. O clique em uma avaliação redireciona o usuário para a “*Página de Notas do Aluno*”, funcionalidade especificada no *item k*.
  - ii. Sessão Minhas matérias: Nesta sessão serão listadas 3 matérias que o usuário está cursando atualmente. Pode ser realizado o mock para a exibição das matérias
  - iii. Sessão Cursos extras: Sessão onde mostrará cursos extras e ou matérias eletivas que o aluno poderá fazer para complementar sua grade curricular. Pode ser realizado o mock para a exibição das matérias
- f. **Página de Cadastro de Docente**: Deve conter um formulário para cadastro de Docente com botões para editar, deletar e salvar.
- i. Essa tela só deverá ser acessada pelo usuário do tipo administrador.
  - ii. Durante o cadastro, os botões de editar e deletar devem ficar desativados (desabilitados).
  - iii. Campos do formulário:
    1. Nome Completo: Obrigatório, com máximo e mínimo de 64 e 8 caracteres, respectivamente.
    2. Gênero: Obrigatório com *dropdown* de opções pré-definidas.
    3. Data de Nascimento: Obrigatório, data válida.
    4. CPF: Obrigatório com o formato 000.000.000-00
    5. RG com órgão expedidor: Obrigatório, com máximo de 20 caracteres.
    6. Estado Civil: Obrigatório com *dropdown* de opções pré-definidas.
    7. Telefone: Obrigatório com o formato (99) 9 9999-99999
    8. E-mail: Obrigatório e com validação do formato de email.
    9. Senha: Obrigatório sendo no mínimo 8 caracteres.
    10. Naturalidade: Obrigatório, com máximo e mínimo de 64 e 8 caracteres, respectivamente.
    11. Endereço: Cep, Cidade, Estado, Logradouro, Número, Complemento, Bairro e Ponto de Referência.

- 12. Matérias que o docente ministra: Obrigatório; campo multi-select com opções de matérias pré-setadas.
  - iv. Deverá utilizar a API do ViaCEP para buscar os dados de endereço.
  - v. Deverá verificar os dados informados antes de cadastrar.
  - vi. Deverá criar um identificador único para cada docente cadastrado.
  - vii. Deverá apresentar uma confirmação ao salvar. Exemplo: *toast* ou *alert*.
- g. Página de Cadastro de Aluno:** Deve conter um formulário para cadastro de Aluno com botões para editar, deletar e salvar.
- i. Essa tela só deverá ser acessada pelo usuário do tipo administrador.
  - ii. Durante o cadastro, os botões de editar e deletar devem ficar desativados (desabilitados).
  - iii. Campos do formulário:
    - 1. Nome Completo: Obrigatório, com máximo e mínimo de 64 e 8 caracteres, respectivamente.
    - 2. Gênero: Obrigatório com *dropdown* de opções pré-definidas.
    - 3. Data de Nascimento: Obrigatório, data válida.
    - 4. CPF: Obrigatório com o formato 000.000.000-00
    - 5. RG com órgão expedidor: Obrigatório, com máximo de 20 caracteres.
    - 6. Telefone: Obrigatório com o formato (99) 9 9999-99999
    - 7. E-mail: Não obrigatório e com validação.
    - 8. Senha: Obrigatório sendo no mínimo 8 caracteres.
    - 9. Naturalidade: Obrigatório, com máximo e mínimo de 64 e 8 caracteres, respectivamente.
    - 10. Endereço: Cep, Cidade, Estado, Logradouro, Número, Complemento, Bairro e Ponto de Referência.
    - 11. Turma: Obrigatório, campo multi-select com a listagem de turmas cadastradas no portal.
  - iv. Deverá utilizar a API do ViaCEP para buscar os dados de endereço.
  - v. Deverá verificar os dados informados antes de cadastrar.
  - vi. Deverá criar um identificador único para cada aluno cadastrado.
  - vii. Deverá apresentar uma confirmação ao salvar. Exemplo: *toast* ou *alert*.
- h. Página de Cadastro de Turma:** Deve conter um formulário para cadastro de Turmas com botões editar, deletar e salvar.
- i. Essa tela só deverá ser acessada pelos usuários dos tipos Administradores e Docentes.
  - ii. Durante o cadastro, os botões de editar e deletar devem ficar desativados (desabilitados).
  - iii. Campos do formulário:
    - 1. Nome da Turma: Obrigatório, com máximo e mínimo de 64 e 8 caracteres, respectivamente.

2. Data de Início/Término: Obrigatório, buscando data atual do sistema e liberando para edição.
  3. Horário da Turma: Obrigatório, buscando horário atual do sistema e liberando para edição.
  4. Professor: Obrigatório, dropdown com a listagem de professores cadastrados no portal.
    - a. Caso o usuário possua o perfil Administrador, deverão ser listados todos os professores/docentes cadastrados no sistema.
    - b. Caso o usuário possua o perfil Docente, o dropdown deverá vir preenchido com o nome do próprio usuário selecionado e não permitir alteração.
  - iv. Deverá verificar os dados informados antes de cadastrar.
  - v. Deverá criar um identificador único para cada turma cadastrada.
  - vi. Deverá apresentar uma confirmação ao salvar. Exemplo: *toast* ou *alert*.
- i. **Página de Cadastro de Avaliação/Nota:** Deve conter um dropdown para a seleção de uma turma para a inclusão de uma avaliação e um formulário para cadastro das notas com botões editar, deletar e salvar.
- i. Essa tela só deverá ser acessada pelos usuários dos tipos Administradores e Docentes.
  - ii. Durante o cadastro, os botões de editar e deletar devem ficar desativados (desabilitados).
  - iii. Campos do formulário:
    1. Professor: Obrigatório, dropdown com a listagem de professores cadastrados no portal.
      - a. Caso o usuário possua o perfil Administrador, deverão ser listados todos os professores/docentes cadastrados no sistema.
      - b. Caso o usuário possua o perfil Docente, o dropdown deverá vir preenchido com o nome do próprio usuário selecionado e não permitir alteração.
    2. Nome da Matéria: Obrigatório, dropdown contendo as opções de matérias pré-setadas.
    3. Nome da Avaliação: Obrigatório, campo de texto aberto para a inserção do nome da avaliação.
    4. Data da Avaliação: Obrigatório, buscando horário atual do sistema e liberando para edição.
    5. Aluno: Obrigatório, deve ser selecionado em um dropdown, pesquisável pelo nome do aluno, o aluno que receberá essa nota.
    6. Nota Avaliação: Obrigatório, numérico com o valor mínimo e máximo sendo 0 e 10.
  - iv. Deverá verificar os dados informados antes de cadastrar.
  - v. Deverá criar um identificador único para cada avaliação/nota cadastrada.



- vi. Deverá apresentar uma confirmação ao salvar. Exemplo: *toast* ou *alert*.
- j. **Página de Listagem de Docentes:** Deverá possuir uma barra de pesquisa, listagem de docentes cadastrados e botão para abrir os detalhes.
  - i. Essa tela só deverá ser acessada pelo usuário do tipo administrador.
  - ii. A barra de busca deverá buscar o docente pelo nome ou identificador cadastrado.
  - iii. A listagem de docentes deve mostrar um *card* para cada um com ao menos o número do registro (id), nome completo, contato (telefone ou e-mail) e um botão "*ver mais*".
  - iv. Ao clicar no botão "*ver mais*", o usuário deve ser redirecionado para a "*Página de Cadastro de Docente*", funcionalidade especificada no *item f*, que deve ser aberta, com os botões de edição, deleção e salvamento ativados. Nesta página, o usuário poderá editar as informações ou deletar um docente caso o mesmo não possua turmas e avaliações vinculadas. Para isso, a tela deve preencher os campos com as informações daquele docente.
- k. **Página de Notas do Aluno:** Deverá possuir histórico de avaliações e notas do aluno.
  - i. Essa tela só deverá ser acessada pelo usuário do tipo aluno.
  - ii. As informações a serem exibidas são: turma, docente da turma, matéria, nome da avaliação e nota.
  - iii. O histórico de notas deve mostrar todas as avaliações realizadas pelo aluno, sendo mostrados em ordem cronológica.
  - iv. Dica: Você pode utilizar uma tabela com todas as informações, categorizada por matéria ou turma.

## 4.2 EXEMPLO DE APLICAÇÃO

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

As imagens a seguir demonstram **exemplos da aplicação** que deverá ser desenvolvida. Você poderá **desenvolver igual ao modelo, ou criar um layout diferente** com estilo próprio.

Imagem 1: Página de Login

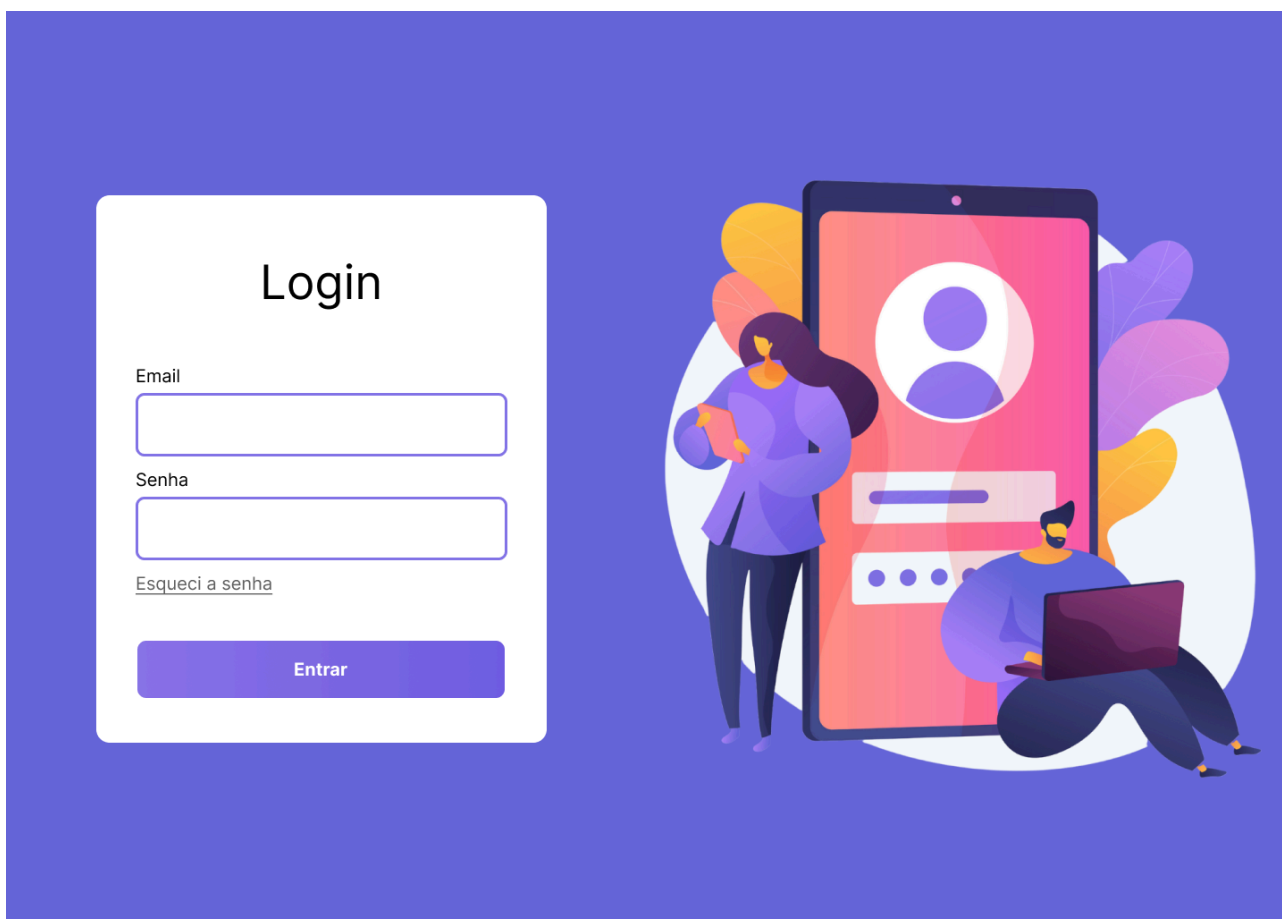


Imagem 2: Página de Início (Estatística e Informações)

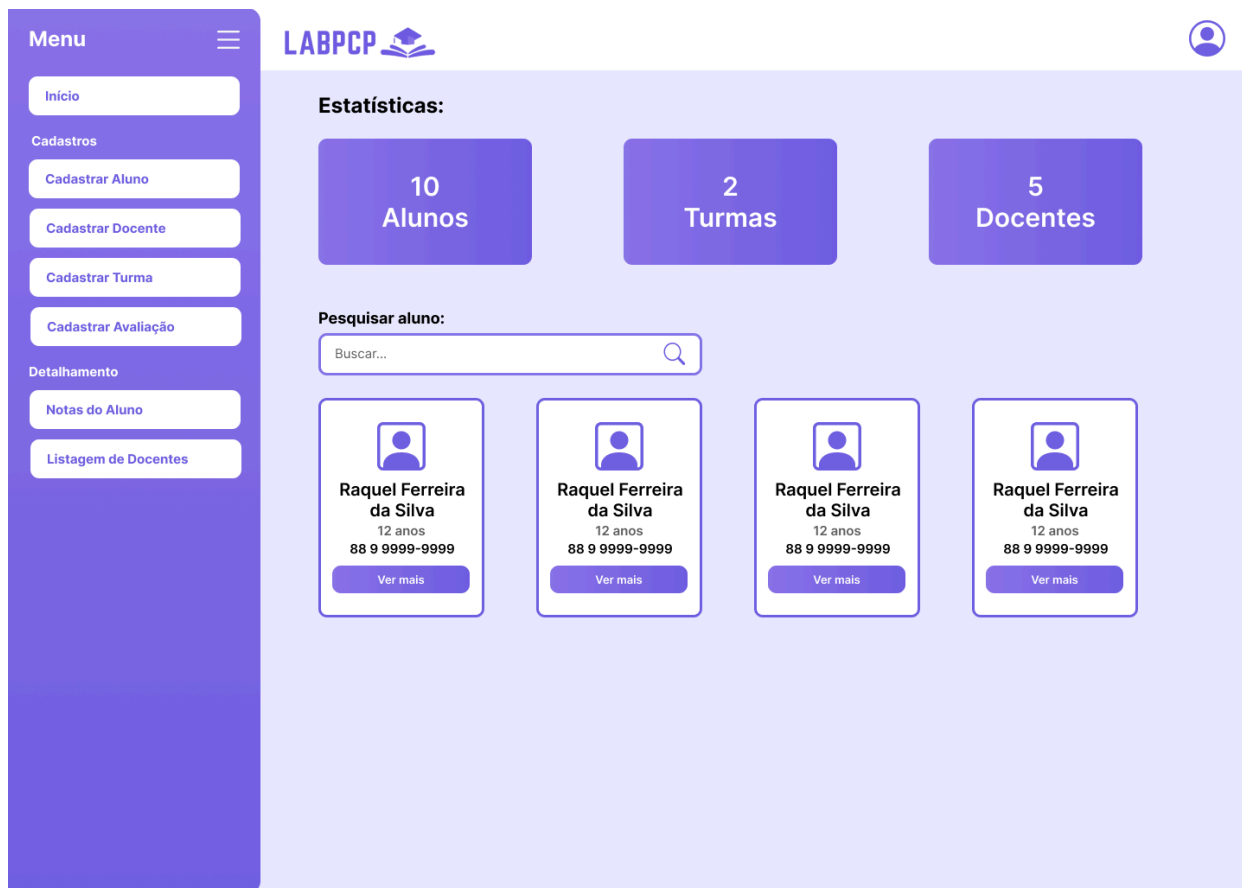


Imagem 3: Página de Cadastro/Edição de Docente

Menu

Início

Cadastros

Cadastrar Aluno

Cadastrar Docente

Cadastrar Turma

Cadastrar Avaliação

Detalhamento

Notas do Aluno

Listagem de Docentes

LABPCP

Cadastrar novo docente

Editar

Excluir

Nome completo

Telefone

Gênero

Estado civil

Data de nascimento

Email

Senha

CPF

RG com órgão expedidor

Naturalidade

Matérias

CEP

Rua

Número

Cidade

Estado

Complemento

Salvar

Imagem 4: Página de Cadastro/Edição de Turma

Menu

Início

Cadastros

Cadastrar Aluno

Cadastrar Docente

Cadastrar Turma


Cadastrar Avaliação


Detalhamento

Notas do Aluno

Listagem de Docentes

LABPCP





## Cadastrar nova turma

Professor

Nome da turma

Data de início

Data de término

Horário

Salvar

Editar

Excluir

Imagem 5: Página de Cadastro/Edição de Aluno

Menu

Início

Cadastros

Cadastrar Aluno

Cadastrar Docente

Cadastrar Turma

Cadastrar Avaliação

Detalhamento

Notas do Aluno

Listagem de Docentes

LABPCP

Cadastrar novo aluno

Editar

Excluir

Nome completo

Telefone

Gênero

Turma

Data de nascimento

Email

Senha

CPF

RG com órgão expeditor

Naturalidade

CEP

Rua

Número

Cidade

Estado

Complemento

Salvar

14

Imagem 6: Página de Cadastro/Edição de Nota/Avaliação

Menu

Início

Cadastros

Cadastrar Aluno

Cadastrar Docente

Cadastrar Turma

Cadastrar Avaliação

Detalhamento

Notas do Aluno

Listagem de Docentes

LABPCP

Cadastrar nova avaliação/nota

Editar

Excluir

Professor

Aluno

Nome da matéria

Nome da avaliação

Data da avaliação

Nota

Salvar

Imagem 7: Página de Listagem de Docente

Menu

Início

Cadastros

Cadastrar Aluno

Cadastrar Docente

Cadastrar Turma

Cadastrar Avaliação

Detalhamento

Notas do Aluno

Listagem de Docentes

LABPCP

## Listagem de docentes

Pesquisar docente:

Buscar...

Id	Nome completo	Telefone	Ação
0001	Raquel Ferreira da Silva	(99) 9 9999-9999	<a href="#">Ver mais</a>
0002	Raquel Ferreira da Silva	(99) 9 9999-9999	<a href="#">Ver mais</a>
0003	Raquel Ferreira da Silva	(99) 9 9999-9999	<a href="#">Ver mais</a>
0004	Raquel Ferreira da Silva	(99) 9 9999-9999	<a href="#">Ver mais</a>



Imagem 8: Página de Notas do aluno

Menu

Início

Cadastros

Cadastrar Aluno

Cadastrar Docente

Cadastrar Turma

Cadastrar Avaliação

Detalhamento

Notas do Aluno

Listagem de Docentes

LABPCP

Notas do Aluno

Informações do aluno

Nome: Raquel Feirreira da Silva  
Email: raquel@gmail.com  
Gênero: Feminino  
Telefone: (99) 9 9999-9999  
CPF: 999.999.999-99

Turma

Docente: Claudia Santos  
Nome da turma: Biologia  
Horário: 13:00

Histórico de notas

Id	Nome avaliação	Data	Matéria	Nota
0001	Teste AV1	12/03/2024	História	8
0002	Teste AV2	12/04/2024	História	7
0003	Seminário	12/05/2024	História	10

#### 4.3 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.

## 4.4 USO DO TRELLO

Na ferramenta **Trello**, crie uma área de trabalho (*workspace*) e um quadro (*board*) utilizando a metodologia *kanban* (*backlog*, *todo*, *doing*, *blocked*, *review* e *done*) para realizar a gestão do seu projeto. A partir das informações extraídas deste documento, organize as tarefas a serem realizadas em cartões (*cards*) e os arraste a cada etapa. Note que você pode fazer um paralelo com a metodologia do *GitFlow*, criando cartões com o mesmo nome, por exemplo *feature/create-login-page*, desta forma, o projeto estará organizado e documentado sendo possível rastrear cada etapa e sanar as dúvidas quando necessário. Não esqueça de habilitar o quadro para modo público e colocar o link no `readme.md` e na submissão no AVA.

**Importante:** Não utilize o quadro de exercícios para realizar tarefas de projeto. Todos os cards diferentes aos de exercícios serão deletados de forma automática.

## 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.

### Apresentação da Solução

Nº	Critério de Avaliação	0	1,00
1	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.

### Uso adequado do Trello

Nº	Critério de Avaliação	0	0,25	0,5
2	Aluno atribuiu adequadamente seu nome no card, bem como especificações da tarefa?	O aluno não criou/atribuiu o card.	Aluno criou o card, mas não colocou seu nome ou comentários especificados nos requisitos.	Aluno criou o card, atribuiu seu nome nele e colocou as especificações de implementação no mesmo.
3	O aluno movimentou adequadamente o card no board, seguindo a metodologia kanban?	Aluno não movimentou o card no board.	Aluno movimentou o card de maneira inadequada no board.	O aluno movimentou adequadamente o card, movimentando 1 por vez, e respeitando as regras no Kanbanize.

### Uso adequado do GitHub

Nº	Critério de Avaliação	0	0,25	0,5
4	As mensagens do commit estão claras e adequadas?	Os commits do aluno não tiveram nenhuma mensagem	Os alunos fizeram commits com comentários, mas esses estavam inadequados ou incompletos com relação ao conteúdo das alterações	As mensagens do commit estão claras e adequadas em relação ao conteúdo das alterações
5	Foi aberto um branch separado para cada etapa, funcionalidade (gitflow)?	Os commits foram realizados diretamente na master/main.	Foram abertas branches para cada funcionalidade, porém foram realizados commits de mesma	Foram abertas branches para cada funcionalidades, e os commits foram

			funcionalidade em branchs diferentes.	realizados de forma concisa nas mesmas.
--	--	--	--	--

### Desenvolvimento adequado da Aplicação

Nº	Critério	0	0,10 a 0,50	0,60
6	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.
7	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 componentes, criação da store e router em seus devidos diretórios, separando regras de negócios e criação, sempre que necessário, em arquivos específicos.
8	Desenvolveu a página de login com a funcionalidade de autenticação funcional e com devidos tratamentos no formulário.	Não desenvolveu a página ou não está 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.
9	Criou o template da aplicação, contendo o Menu Lateral e Toolbar funcional de acordo com os requisitos?	Não desenvolveu os componentes ou não está funcional.	Desenvolveu os componentes, porém sem alguns dos elementos visuais ou sem a validação de exibição após login	Desenvolveu o template utilizando Menu Lateral e Toolbar, com todas as funcionalidades de acordo com os requisitos e com a validação de exibição apenas após o login.
10	Desenvolveu a página de início de administradores e docentes, contendo as estatísticas, pesquisa, listagem de alunos, e redirecionamento para a página solicitada	Não desenvolveu a página ou não está funcional.	Desenvolveu a página de início, com todas as estatísticas e listagem de alunos com dados estáticos e/ou sem redirecionamento	Desenvolveu a página de início, com todas as estatísticas, pesquisa, listagem alunos e acesso a página solicitada no item.
11	Desenvolveu a página de início do aluno, contendo todas as sessões e realizando o	Não desenvolveu a página ou não está funcional.	Desenvolveu a página de início, com dados estáticos e/ou sem redirecionamento	Desenvolveu a página de início, com todas as sessões e realizando o redirecionamento para a página solicitada.

	redirecionamento para a página de notas			
12	Desenvolveu a página de cadastro de docentes 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 docentes funcional, com todas as validações, efetuando consulta na API, guardando os registros no localStorage e permitindo edição e deleção corretamente.
13	Desenvolveu a página de cadastro de alunos, guardando as informações no localStorage (ou JSON Server), efetuando as devidas validações de formulário?	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 com todas as validações funcionais, armazenando a avaliação/nota no localStorage e retornando para o usuário a mensagem de conclusão.
14	Desenvolveu a página de cadastro de turma, guardando as informações no localStorage (ou JSON Server), efetuando as devidas validações de formulário?	Não desenvolveu a página ou não está funcional.	Desenvolveu parcialmente a página ou com erros de validação, não salvando corretamente no localStorage	Desenvolveu a página de cadastro com todas as validações funcionais, armazenando a avaliação/nota no localStorage e retornando para o usuário a mensagem de conclusão.
15	Desenvolveu a página de cadastro de avaliações/notas, efetuando as validações e armazenando o dado corretamente	Não desenvolveu a página ou não está funcional.	Desenvolveu parcialmente a página ou com erros de validação, não salvando corretamente no localStorage	Desenvolveu a página de cadastro com todas as validações funcionais, armazenando a avaliação/nota no localStorage e retornando para o usuário a mensagem de conclusão.
16	Desenvolveu a página de cadastro de listagem de docentes, com todos os elementos solicitados e realizando o redirecionamento para a página solicitada	Não desenvolveu a página ou não está funcional.	Desenvolveu a página, porém sem todos os elementos visuais, não realizando o redirecionamento solicitado ou com dados fixos em tela.	Desenvolveu a página totalmente funcional, seguindo todas as especificações solicitadas e realizando as integrações para consumo dos dados.
17	Desenvolveu a página de notas dos alunos, com todos os elementos solicitados	Não desenvolveu a página ou não está funcional.	Desenvolveu a página, porém sem todos os elementos visuais ou com dados fixos em tela	Desenvolveu a página totalmente funcional, seguindo todas as especificações solicitadas e realizando as integrações para consumo dos dados.

## 6 PLANO DE PROJETO

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

- **Kanban Board:** O que é Kanban e utilização no Trello.
- **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
- **Angular:** Setup Inicial (instalação e configuração), Angular CLI, Estrutura, Bootstrap, Componentes, Property Binding, Passagem de Valores, Injeção de Dependência, Eventos de Componentes, Rotas, Navegação Segura, Decorators, Diretivas (ngIf e ngFor) e ng-content.