

Requisitos funcionais

- 1.1 - Fazer transição de formulário Login/Cadastro, assim que o botão cadastrar for acionado.
- 1.2 - Caso o algoritmo identifique um erro pelo usuário, informar na tela.
- 1.3 - Cadastrar usuário no banco de dados, caso todas as informações estiverem corretas.
- 1.4 - Fazer login se o algoritmo validar se a matrícula e a senha forem compatíveis com os dados no banco de dados, fazer a verificação de cargo pela matrícula para ver quais os acessos que o usuário poderá ter acesso.
- 1.5 - Operador poderá visualizar apenas, Atividade, Perfil e Calendário
- 1.6 - Coordenador poderá visualizar, Atividade, Perfil, Calendário e a aba de envio de atividade
- 1.7 - O Administrativo terá o acesso completo do site apenas trocando a aba de envio de atividade por Cadastro Usuário e gerenciamento de usuario
- 1.8 - Redirecionar o usuário para a tela inicial.
- 1.9 - Carregar botões na tela inicial, com Perfil, Atividade, Calendário e Sobre nós.
- 2.0 - Caso usuário clique no botão perfil.
- 2.1 - Redirecionar o usuário para a aba Perfil para visualização das suas informações.
- 2.2 - Caso usuário clique no botão Atividade.
- 2.3 - Se usuário notar que tem algo de errado, poderá pedir para o Administrativo alterar seus dados com um link do e-mail.
- 2.4 - Deverá ser redirecionado na aba de Atividades para visualização, onde poderá ver sua atividade diária que foi mandada pelo coordenador.
- 2.5 - Poderá marcar se concluiu a atividade com uma caixa de verdadeiro ou falso, assim que concluir todas, poderá enviar as atividades para o banco de dados.
- 2.6 - Se usuário clicar na Aba de calendário, será redirecionado para a aba.
- 2.7 - Na aba calendário ele poderá visualizar suas atividades realizadas.
- 2.8 - O Login Adm terá todo o acesso do site e poderá visualizar os dados dos usuários, atualizar e cadastrar novos usuário.
- 2.9 - Já o login de coordenador o único diferencial do operador é que ele poderá mandar atividades para seu operador.

Requisitos não funcionais

- 1.1 - A aplicação foi feita em C# utilizando o framework ASP.NET Core, com a estrutura de código com o Blazor server
- 1.2 – A estrutura do código é separada em BackEnd, DTO (Data Transfer Object) e o FrontEnd
- 1.3 – Funciona da seguinte maneira, o Front pede requisição para o DTO que faz essa “ponte” para o Back e lá executa essa requisição e manda novamente para o DTO e assim chega no front
- 1.4 – O sistema de segurança do sistema é feito pelo token no local storage
- 1.5 – Tempo de resposta do sistema deve ser em menos de 2 segundos
- 1.6 – Somente quem tiver com o cargo Adm deverá ter acesso completo ao sistema
- 1.7 – Fácil usabilidade, pois não tem muitas abas ou muitas funções complicadas
- 1.8 – Aplicação com uma segurança ótima, pois as informações que aparecem na tela são buscadas pelo token no local storage, assim quando o usuário fechar a tela o token vai sumir e as informações também
- 1.9 – A aplicação tem tempo de login de 18 minutos ausente no site