

HOTSPOT:

1 ° Passo - Elaboração do Wireframe da Tela de Login

Observamos sites que são da empresa, utilizamos do Manual da Marca SENAI para desenvolvermos um modelo da tela seguindo os padrões, com as ferramentas do Figma realizamos um Wireframe e podemos visualizá-la.

2 ° Passo - Estilização para Web da Tela de Login

Com a utilização do Wireframe anterior fizemos alguns ajustes para a tela Web, pois o Figma só é utilizado de base, pois quando é passado para a Web, boa parte das vezes fica desproporcional.

3 ° Passo - Responsivo

Fizemos um Wireframe para o Responsivo baseado no Web, porém modificamos algumas coisas, como por exemplo a cor da Tela, e o ajustamos para ser funcional em celulares de diferentes tamanhos.

4 ° Passo - Implementação do Método de Login

Fizemos as funcionalidades de autenticação de email e senha, para fazer o login e ter acesso a internet

5 ° Passo - Implementação no Mikrotik

Depois de ter feito as funcionalidades, implementamos o design e método funcional no roteador

6 ° Passo - Teste do Hotspot no Notebook

Foi feito o teste do hotspot, com usuários já cadastrados no roteador, na versão web pelo notebook

7 ° Passo - Teste do Hotspot no Celular

Foi feito o teste do hotspot, com usuários já cadastrados no roteador, na versão mobile pelo celular

PAINEL:

BACK-END:

Realizado em C#

1º Passo - Criação das Modelagens

Foi realizado as modelagens físicas, conceitual e lógico.

2º Passo - Criação do Banco de dados SQL

Através do modelo físico realizamos a criação do banco de dados, criando o dml, ddl e dql.

Tabelas criadas: TipoUsuario, Usuário e CadastrarCampanhas.

3º Passo - Criação da aplicação

Após o banco de dados criados, realizamos a criação da aplicação no Visual Studio Code 2019, usando a linguagem C#.

4º Passo - Instalações de pacotes

Instalamos alguns pacotes necessários para o nosso projeto.

Sendo eles: `BCrypt.Net-Next`, `Microsoft.AspNetCore.Authentication.JwtBearer`, `Microsoft.AspNetCore.Mvc.NewtonsoftJson`, `Microsoft.EntityFrameworkCore`, `Microsoft.EntityFrameworkCore.Analyzers`, `Microsoft.EntityFrameworkCore.SqlServer`, `Microsoft.EntityFrameworkCore.SqlServer.Design`, `Microsoft.EntityFrameworkCore.Tools`, `Microsoft.Extensions.Configuration.FileExtensions`, `Microsoft.Extensions.Configuration.Json`, `Swashbuckle.AspNetCore`, `System.Data.SqlClient` e `System.IdentityModel.Tokens.Jwt`

5º Passo - Criação das Domains/Context - Scaffold

Através dos pacotes, realizamos o comando de scaffold no cmd do projeto, onde gerou as Domains e as Context.

6º Passo - Criação das Interfaces

Após a criação das domains definimos o que cada tabela ia realizar.

Todas as tabelas tiveram: ListarTodos, BuscarPorId, Deletar, Atualizar e Cadastrar.

Somente a tabela de CadastrarCampanha tiveram dois métodos diferentes, um deles foi AtualizarBool (Atualiza a boolean) e AtivoList (Lista somente as campanhas ativas)

7º Passo - Criação dos Repositories

No repositories definimos o métodos da interfaces e usamos a context com o ctx, colocando cada método em suas classes

8º Passo - Criação das Controllers

Puxamos o repositories e definimos através de um if return, as mensagens que apareceria caso tentasse utilizar uma de nossas tabelas e caso ela fosse sucedida ou até mesmo um BadRequest.

9º Passo - Criação da LoginViewModel

Criamos o método de login e a mensagem que retornaria, para logar na nossa aplicação teria que fazer um cadastro com email e senha.

10º Passo - Adicionar métodos na Startup

Definir o swagger, método de quanto tempo o usuário poderá ficar logado, através da chave de autenticação e puxar o método StaticFiles.

11º Passo - Criação das pastas Utils/Upload/Criptografia

Pasta Upload - Serve para definir extensões, dados, delete, return e validação.

Pasta Criptografia - Server para criptografar as senhas através de validação e comparação.

12º Passo - Criação das pastas StaticFiles/Images

Salva as imagens cadastradas do painel dentro destas pastas.

13º Passo - Adicionar o banco de dados no appsettings.json

Adicionamos o banco de dados na appsettings para aumentar a segurança da aplicação.

14º Passo - Testes

Testamos o projeto através do swagger e do postman

15º Passo - Deploy na azure

Subimos a aplicação na nuvem, através da plataforma azure.

FRONT-END:

Realizado em React JS

1º Passo - Método de Login: O Login foi criado com autenticação jwt, criptografia e uma separação entre os usuários que vão acessar a plataforma. A separação propriamente dita é realizada logo nesta primeira tela para que possa ser feita a devida triagem entre o usuário ADM e o usuário MARKETING e TELEVISÃO para que possa ser redirecionado a tela correspondente de sua função.

2º Passo - Método de Logout: O método de logout é efetuado a partir de uma limpeza do valor do token registrado no localStorage da aplicação criada. Sem estes valores salvos, o usuário não tem a oportunidade de se conectar novamente sem realizar outro login. Independente do usuário que tente se conectar, seja ADM, MARKETING ou TELEVISÃO, será necessário ser feito o login novamente, a menos que a senha seja salva.

3º Passo - Cadastro de Campanha: A função de cadastro de campanha é composta por um formulário que solicita o “Nome da Campanha, Data de Início e de Expiração, Descrição e a Imagem” e é pego no token jwt o Id do Usuário que está logado.

4º Passo - Registro de Campanha: A função de registro de campanha é uma tabela que retorna os dados das campanhas cadastradas, onde é retornado: “Nome da Campanha, Usuário que Cadastrou, Data de Início e Expiração, e os botões de ações”, as ações sendo o botão de excluir campanha e o de ativa-lá. Como o próprio nome diz, o botão de excluir campanha, exclui ela permanentemente, e o botão de ativar campanha que foi feito com um método que vai verificar o valor do campo “CampanhaAtiva” se esse valor for 0 significa que a campanha está inativa, fazendo com que o botão a transforme em ativa assim que for clicado, e vice-versa.

5º Passo - Listagem de Campanha: A Tela da Campanha é simples, metade da tela comporta a imagem cadastrada e a outra metade seu título e descrição, apenas campanhas *Ativas* são exibidas (recomendado ativar apenas uma campanha por vez). Só pode ser acessada por usuários do tipo “Televisão”.

6º Passo - Cadastros de Usuários: Composta por um formulário onde tem os campos “*Nome, Email, Senha e Tipo Usuário*”, após completar as informações deve-se clicar em “Cadastrar” e a página será redirecionado para a “Lista de Usuários”.

7º Passo - Lista de Usuários: Uma tabela com as informações: ID, Usuário, Email, Tipo Usuário e Ações. Contém um botão de excluir para cada usuário existente e abaixo da tabela um para cadastrar que redireciona a tela de “Cadastro de Usuários”.

8º Passo - Deploy: Subimos a aplicação na nuvem, através da plataforma vercel.