**Projeto Definitivo: CAST CURSOS**

**Link GitHub:** [*https://github.com/Thiagodz7/ProjetoDefinitivo.git*](https://github.com/Thiagodz7/ProjetoDefinitivo.git)

**Scripts de Dados:   
  
Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

**Evidencias de Testes:**

**Testes de Cadastros e Visualização dos dados:**

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

**Teste de validação do Campo:**

**Interface gráfica do usuário, Site

Descrição gerada automaticamente**

**Teste Confirmação da Senha:**

**Tela de celular com aplicativo aberto

Descrição gerada automaticamente**

**Teste de Get das datas:**

**Texto

Descrição gerada automaticamente**

**Teste Comprovação de Funcionamento de Logs:**

**Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente**

**Descrição técnica da solução implementada:**

Para a realização da solução solicitada pelo cliente foi realizada com as seguintes ferramentas: DotNet, C#, Entity Framework, Angular, SqlServer, Html, Css, JavaScript, TypeScript, PostMan e Swagger. Todas através do Visual Studio Code e Visual Studio Community.

A solução contém uma conexão direta entre BackEnd e FrontEnd graças a API utilizada e modulada no Controller do BackEnd, a ferramenta entity framework foi essencial para criar as models que possibilitam não mais facilidade de acesso ao banco de dados, como também, maior facilidade de manipular as tabelas que lá estiverem criadas, foi utilizado também a ferramenta angular para a construção do FrontEnd onde através do TypeScript manipulei os dados vindos diretamente do banco de dados através da API, o TypeScript foi essencial para isso, praticamente toda a lógica do sistema foi através do component de cada tela, como é possível verificar abaixo:

Texto

Descrição gerada automaticamente

Os Services foram utilizados não só para trazerem informações via API/Requesição HTTP, mas também para manipular a visualização das tabelas(Semelhante ao InnerJoin) porém diretamente no código:

Texto

Descrição gerada automaticamente

Como visto no código acima, é chamada vários serviços em um único componente, isso foi feito para acessar dados de outras tabelas na mesma página, ou seja, no mesmo documento html, e não só para isso, foi usado também para manipular dados e cadastrar dados em outras tabelas com um único comando(exemplo: tabela Log após um cadastro na tabela cursos), obvio que isso fez com que o código ganhasse uma boa quantidade a mais: Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Percebe-se que no Método “**OnSubmit**()” ele cadastra um curso novo e também cadastra um log, tudo isso através de serviços.

Para não alongar muito na escrita, finalizo mostrando a ferramenta utilizada para versionamento do código para o GitHub: Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Ferramenta de extensão do próprio Visual Code, e para o gitignore foi usado a linha de comando:

Texto

Descrição gerada automaticamente