

Arquitetura de Implementação do Projeto Hydra

1. Resumo do Projeto

Objetivo: Desenvolver uma solução para calcular o consumo hídrico de estabelecimentos, promovendo conscientização sobre o uso eficiente de água.

Stack Tecnológica:

- Frontend: React.js com React Router para navegação.
- Backend: .NET 8.0 (API RESTful com autenticação JWT).
- Banco de Dados: MySQL 8.0.
- Infraestrutura: Azure para hospedagem e serviços gerenciados.
- Controle de Código: GitHub com GitHub Actions para CI/CD.

2. Arquitetura Geral

Camadas do Sistema:

- Frontend: Interface do usuário para entrada e exibição de dados.
- Backend: Lógica de negócios e API RESTful.
- Banco de Dados: Persistência dos dados em MySQL.
- Infraestrutura: Serviços de nuvem gerenciados no Azure e Github.

Fluxo de Funcionalidades e Dados

Cadastro de Usuários:

- O usuário insere dados como nome, e-mail, senha e informações básicas do estabelecimento no formulário do frontend.
- Frontend valida os dados localmente e envia uma requisição POST ao backend
- Backend realiza validações adicionais (ex: verificar duplicidade de e-mail).
- Os dados são armazenados na tabela Users no banco de dados.
- Backend retorna uma mensagem de sucesso ou erro ao frontend, que exibe o resultado ao usuário.

Inserção de Dados de Consumo:

- Usuário logado acessa a tela de entrada de consumo.
- Preenche o formulário com informações de consumo direto e indireto
- O frontend envia uma requisição POST ao backend contendo os dados preenchidos.
- O backend valida os dados, armazena-os e prepara para o cálculo.

Cálculo do Consumo Hídrico:

- O backend processa os dados armazenados, aplicando as fórmulas e métodos definidos para calcular o consumo hídrico total.
- Backend retorna o resultado ao frontend em formato JSON.
- O frontend exibe o resultado ao usuário de forma amigável, utilizando gráficos e indicadores visuais, com a possibilidade de download de um relatório em PDF.

Recomendações de Redução:

- Com base nos dados de consumo registrados, o backend analisa padrões e identifica oportunidades de redução.
- Dicas personalizadas são geradas e exibidas ao usuário junto com o relatório de consumo.

3. Detalhes Técnicos da Implementação

Frontend**Estrutura de Rotas:**

- /: Página inicial.
- /cadastro: Formulário para registro de usuários.
- /login: Tela de autenticação.
- /entradaconsumo: Formulário para entrada de dados de consumo hídrico.
- /ajuda: Informações sobre o projeto.
- /contato: Página de contato.

Componentes:

- Ajuda.jsx
- Cadastro.jsx
- Contato.jsx
- EntradaConsumo.jsx
- Home.jsx
- Login.jsx

Ferramentas e Pacotes:

- React Router: Para navegação entre páginas.
- Axios: Para gerenciar requisições à API.

Backend

Estrutura de camadas:

- Controller: Define os endpoints e manipula requisições.

Exemplos: UsersController.cs e ConsumptionController.cs

- Service: Implementa a lógica de negócios.

Exemplo: JwtService.cs

- Repository: Interage com o banco de dados utilizando Entity Framework Core.

Exemplo: AppDbContext.cs

Endpoints RESTful:

/api/users:

- POST /register: Cadastro de usuários.
- POST /login: Geração de token JWT.

/api/consumption:

- POST /calculate: Envia dados para cálculo.

Banco de dados

Estrutura básica (MySQL 8.0):

- Tabela `Users`: Armazena informações de usuários (nome, email, senha hash, etc.)

Infraestrutura

Azure:

- Hospedagem do backend como App Service.
- Banco de dados MySQL configurado como serviço gerenciado.

Pipeline de CI/CD

Configuração no GitHub:

- Build: Testar e compilar frontend e backend.
- Testes: Executar testes unitários e de integração.
- Deploy: Deploy automático para o Azure após aprovação.