

INF-0999 - Trabalho Final

Grupo

Leonardo Rener de Oliveira
Victor Akira Hassuda Silva

1. Introdução

A aplicação desenvolvida busca conectar pessoas com ONGs que atendam suas necessidades, com foco em pessoas de baixa renda atendidas por assistentes sociais. Assim as ONGs cadastradas podem entrar em contato e auxiliar essas pessoas, contribuindo em temas como a erradicação da pobreza e da fome.

2. Objetivo

O objetivo deste trabalho é colocar em prática os conhecimentos adquiridos ao longo do curso. Para isso, foram disponibilizados 4 temas, dos quais escolhemos Erradicar a Pobreza e a Fome através da criação de um sistema de informação.

O sistema consiste em realizar um cadastro das ONGs pelo Brasil e das famílias em situação de vulnerabilidade. Assim, através deste sistema, um assistente social consegue realizar uma entrevista e cadastrar essas famílias e as ONGs que estiverem cadastradas podem visualizar as famílias que precisam de atendimento.

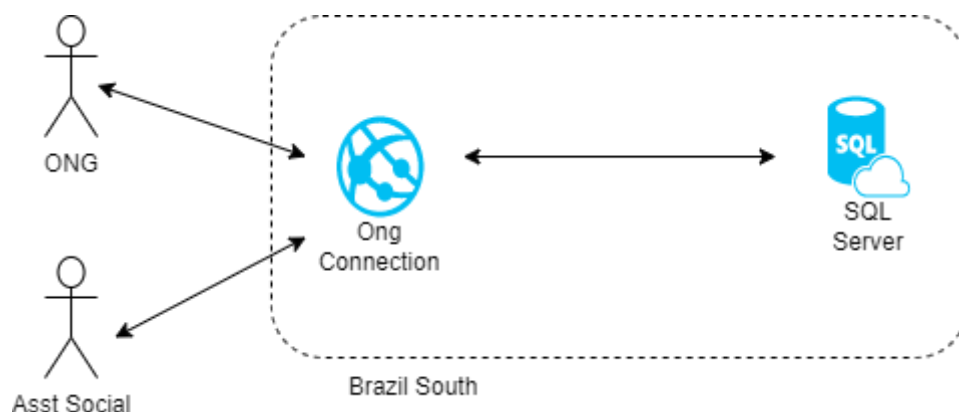
3. Considerações

Algumas considerações foram levadas em conta ao desenvolver este sistema:

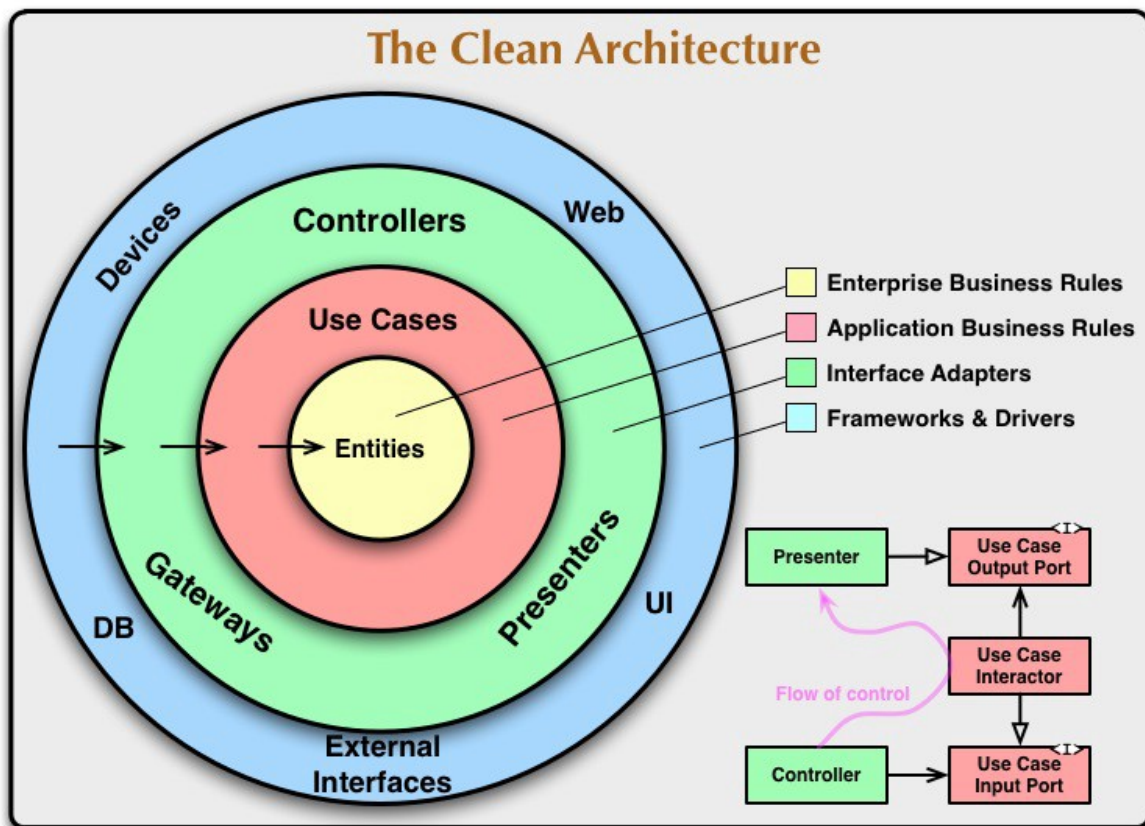
1. O sistema será utilizado por ONGs e por Assistente Sociais
2. As ONGs farão o autosserviço de se cadastrarem
3. Os assistentes sociais são os responsáveis por realizar o cadastro das famílias após uma breve entrevista para que se possa identificar a necessidade familiar

4. Desenvolvimento

O sistema Ong Connection é um sistema Web hospedado no Azure Cloud com banco de dados também hospedado na nuvem. Todo o sistema foi desenvolvido em C#.Net seguindo os modernos padrões de arquitetura de Software.



A aplicação segue o padrão DDD (Domain Driven Design), onde cada uma das camadas do domínio possui uma responsabilidade. Esse padrão está incluso nos princípios de design do Clean Architecture proposto por Robert Cecil Martin.



O sistema se divide em 4 camadas distintas, sendo a primeira a camada de Serviço, onde temos a aplicação Web seguindo o padrão MVC (Model View Controller).

A segunda camada é a Aplicação, onde ficam as regras que comandam as necessidades da camada de serviço, como registrar um usuário ou cadastrar uma família por exemplo.

Na terceira temos a camada de Domínio ou camada de regras de negócios, onde ficam todas as regras específicas ao negócio, como entidades do banco de dados ou regras específicas a serem aplicadas para tratar os dados recebidos ou a enviar para a Aplicação.

Na última camada temos a camada de infraestrutura, onde reside qualquer integração que seja necessária para conversar com o mundo externo à aplicação, como o Banco de Dados.

Para desenvolver o sistema, foram utilizados os seguintes frameworks:

- ASP.NET Core MVC para gerar a aplicação base e comandar as requisições da Web
- Identity Framework para gerenciar os usuários da aplicação
- Entity Framework para gerenciar as entidades do Banco de Dados, como realizar criação, leitura, atualização e deleção de registros e tabelas.

- Dependency Injection Framework para gerar uma aplicação onde as camadas são fracamente conectadas, facilitando assim a manutenção do software

O código encontra-se armazenado no Azure DevOps em um repositório Git privado. Durante o desenvolvimento não foi utilizado nenhum controle de branch ou algo similar, por se tratar apenas de um MVP e ter um curto prazo para entrega, ambos commitaram diretamente na branch Main e trabalharam em arquivos diferentes para evitar conflitos. Caso fosse um cenário mais realista, em hipótese alguma teria um desenvolvedor trabalhando diretamente na Main todas as formas de trabalho seriam feitas em branches a partir da Main, seguindo o Trunk Based Development.

Na hospedagem de todo o sistema, foi utilizado o Azure Cloud e por se tratar de uma aplicação de baixo volume de acesso e baixo processamento, foram utilizados os recursos mais básicos para a hospedagem.

Para realizar o deploy de toda a infraestrutura necessária, foi utilizado o Terraform juntamente do Azure Pipeline. Assim, todo commit na branch Main é automaticamente enviado para ser disponibilizado na nuvem.

E para evitar que toda a infraestrutura seja recriada todas as vezes que a pipeline é executada, o Terraform foi integrado com o Storage Account para salvar o State file, arquivo que contém toda a infraestrutura atual e caso haja mudanças entre a infra a ser criada e o state file, o Terraform identifica as mudanças e as aplica.

Os serviços utilizados pelo Terraform e pelo sistema foram:

- **Key Vault** para armazenar os IDs e Secrets necessários para a aplicação e o usuário e senha do SQL Server
- **Storage Account** para armazenar o State file do Terraform
- **Servidor Sql Server** para hospedar o Banco de dados
- **Sql Server** como provedor de Banco de dados
- **Windows Server** para hospedar o App Service
- **App Service** para gerenciar o Web App

O grupo gerenciou o trabalho seguindo a metodologia Scrum, mas não foi definida uma sprint ou dailies para sabermos o status. Os cards foram criados depois de ter o tema definido e após um rápido brainstorming para uma definição de quais seriam as funcionalidades necessárias para o sistema.

A metodologia foi crucial para a organização do grupo e para evitar conflitos ao desenvolver as funcionalidades, a fim de evitar que 2 pessoas trabalhassem em uma mesma história.

Foram utilizadas as técnicas de brainwriting, para geração de ideias, e de prototipação de baixa fidelidade, onde foram prototipadas ideias para algumas das principais telas da aplicação, com o objetivo de garantir uma melhor experiência para os usuários.

The image displays three wireframe screens for a system, each with a title at the top:

- Tela de Cadastro**: A registration form with a header "Selecione o tipo de conta" and two radio buttons: "ONG" (selected) and "Assistente Social". Below are input fields for "Nome", "Email", "Senha", and "Confirmar Senha", followed by a "Registrar" button.
- Tela de especificações da ONG**: A screen with two main sections. The left section, "Área de Atuação", includes fields for "Estado", "Cidades", "Bairro", and "Quantidade de pessoas que consegue atender". The right section, "Perfil atendido", includes fields for "Público alvo" and "Idade alvo".
- Tela de pessoas que a ONG atende**: A screen with two columns. The left column, "Atendendo", has a header "Nome da pessoa..." and a large empty box below. The right column, "Novas solicitações", has a header "Nome da pessoa..." and two rows of input fields, each with "Aceitar" and "Recusar" buttons.

Estas interfaces de usuário foram desenvolvidas utilizando o framework Bootstrap, este framework provê componentes de interface e front-end para sites e aplicações web. Foram utilizados principalmente os componentes Card, Buttons e classes do estilo Forms, como Select, Input Group e Radios, utilizadas nas interfaces com formulários.

Para garantir uma segurança razoável contra as principais formas de ataque identificadas pelo OWASP, as boas práticas de programação e configuração do sistema foram obedecidas.

Assim, para evitar ataques de injeção, o Entity Framework impede que sejam inseridos dados com caracteres especiais no banco de dados. Por mais improvável que seja, caso ocorra um erro do tipo, o front-end impede que seja executado qualquer script, pois o Asp Net Core trata todos os valores recebidos como texto, impedindo a execução de scripts maliciosos e também não expõe erros ocorridos no banco de dados.

Para realizar transições entre as páginas, nenhum dado sensível, como ID dos registros do banco de dados, é exposto. Apenas os dados que são visíveis para o usuário são utilizados.

E por fim, o Identity Framework, proporciona a segurança dos usuários, gerenciando a criação de usuários e papéis. É importante salientar que o Identity Framework armazena a senha criada pelo usuário, mas utilizando técnicas de Hashing para não expor o valor real.

5. Conclusão

Ao finalizar o MVP, podemos identificar vários pontos de melhorias, caso o sistema deixe de ser apenas uma prova de conceito e se torne um sistema real, como:

- Utilizar GUIDs (Global Unique Identifier) como identificadores de registros
- Adicionar uma área administrativa, onde o usuário com privilégios de administrador possa gerenciar diretamente os usuários da ferramenta
- Adicionar um filtro para que as ONGs consigam visualizar melhor as famílias a serem atendidas

Com esse trabalho foi possível aplicar todos os conhecimentos adquiridos durante o curso e se aproximar do formato de trabalho para desenvolver uma aplicação do zero no mundo real.