

.Objetivo do Projeto e Descrição da Solução

Atualmente não são todos os ambientes que possuem profissionais que se comunicam de maneira fluente a linguagem de sinais (ou libras, aqui no Brasil), e isso causa um impacto nas relações e experiências individuais de pessoas com deficiência auditiva ou surdas. Acessibilidade não é uma opção, mas uma necessidade e a ferramenta desenvolvida e que lhes está sendo apresentada quer colaborar para com a inclusão de indivíduos com necessidades especiais.

Acreditamos que com a implementação de uma ferramenta que reconheça a linguagem de sinais e faça a tradução em texto para o atendente, a fluência em libras deixa de se tornar um requisito, tornando as vagas mais acessíveis, bem como o atendimento.

O principal objetivo do projeto é tornar mais acessível a questão de atendimento e de resolução dos problemas individuais do cliente sem que ele precise estar acompanhado de alguém com capacidade de falar, ou sem que ele precise de um funcionário fluente em libras, tornando-o mais independente e também, deixando de lado a necessidade da fluência em linguagem de sinais como requisito para contratação do funcionário.

O projeto consiste no desenvolvimento de uma ferramenta que possibilita a comunicação do cliente que possui necessidades especiais (nesse caso, pessoas mudas e surdas que saibam ler) e o atendente que não possui fluência em Libras. A ferramenta detecta os gestos feitos pelo cliente por meio de uma câmera e a inteligência artificial processa os dados. Após o processamento de dados, a IA interpreta e mostra ao atendente o que foi dito pelo cliente. A resposta para a requisição do cliente será feita por meio de texto, mostrando a ele numa tela. O atendimento encerrará quando o atendente informar o fim à ferramenta e o programa será reiniciado.

Funcionalidades

F01 - Cadastrar atendente

Descrição: A plataforma deve permitir o cadastro do funcionário.

F02 - Comunicação com microcomputador

Descrição: A plataforma deve se comunicar com o Raspberry.

F03 - Capturar sinais

Descrição: O aplicativo deverá capturar os sinais em libras emitidos pelo cliente através da câmera do dispositivo ou do totem.

F04 - Traduzir sinais

Descrição: O sistema deverá processar os dados obtidos e traduzir em formato de texto através do uso de uma inteligência artificial.

F05 - Salvar sinais

Descrição: Os sinais capturados deverão ser salvos no banco de dados.

F06 - Salvar tradução

Descrição: A plataforma deve salvar as traduções no banco de dados.

F07 - Mostrar o resultado

Descrição: O aplicativo mostrará a tradução em formato de texto na tela.

.Implementação do backend

A implementação do backend será feita através da criação das classes que compõem o core do sistema; Funcionário, Atendimento, Pergunta, Sinal, e Resposta.

A classe Funcionário possuirá os seguintes atributos: nome, email, senha, data de cadastro.

A classe Atendimento possuirá o id, data, duração(em minutos) e o título do assunto tratado.

A classe Pergunta terá o id, o texto que representa a requisição do cliente e a duração(em segundos) necessária para captar a informação gestual.

A classe Resposta possuirá o id, texto respondendo a requisição feita e duração(em segundos).

A classe Sinal terá o id, texto que corresponde a tradução do gesto feito pelo cliente e o tempo que foi necessário para ser reconhecido.

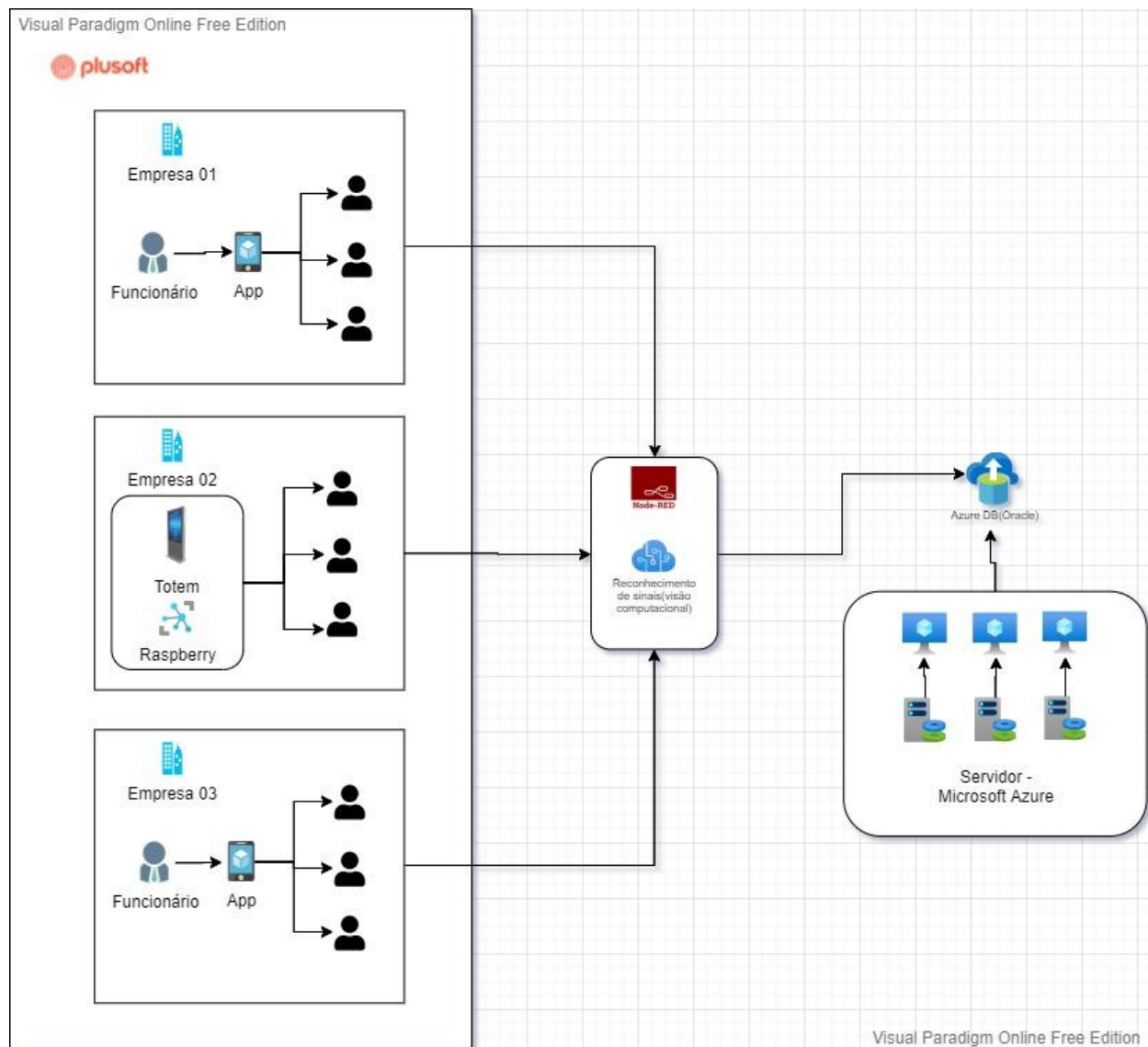
Criando uma estrutura que permita o controle dos dados gerados através das interações entre clientes e atendentes.

Serão implementadas classes DAO para salvar as informações referentes aos atendimentos feitos no aplicativo. As classes implementadas serão: FuncionarioDao, PerguntaDao, RespostaDao e SinalDao.

.Listagem das tecnologias e frameworks

- **Raspberry:** Microcontrolador responsável pela execução do programa.
- **ORACLE:** Banco de dados.
- **SPRING/Java:** Framework para construção da API.
- **MQTT:** Serviço de mensageria utilizado pelo sistema.
- **OpenCV/Python:** É utilizado para reconhecimento e processamento de imagens que serão enviadas pelo aplicativo.

.Arquitetura de Solução



1º Cenário - Envio via celular: Haverá um aplicativo, onde o funcionário gravará um vídeo do cliente fazendo os sinais em libras do que ele deseja, este vídeo terá uma curta duração. Após gravado será enviado via Rest com Node-Red com a visão computacional, que executará o algoritmo para o reconhecimento do sinal e retornará um arquivo .txt, contendo o que o cliente deseja. O banco de dados será responsável por salvar os dados do atendimento (data, duração, assunto do atendimento e o funcionário que o executou) além de salvar o tempo de resposta do algoritmo e qual foi a mensagem processada, utilizando dos servidores da Azure para manter o banco ativo.

2º Cenário - Atendimento via totem: Haverá um totem, onde o cliente gravará um vídeo fazendo os sinais em libras do que deseja, este vídeo terá uma curta duração. Após gravado será enviado ao raspberry via código MQTT e depois chamará via Rest o Node-Red com a visão computacional, que executará o algoritmo para o reconhecimento do sinal e retornará um arquivo .txt, contendo o que o cliente deseja. O banco de dados será responsável por salvar os dados do atendimento(data, duração, assunto do atendimento e que o atendimento foi feito via totem) além de salvar o tempo de resposta do algoritmo e qual foi a mensagem processada, utilizando dos servidores da Azure para manter o banco ativo.