

INSTITUTO FEDERAL DO PARANÁ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FELIPE ANDRADE

BEYOND SOUND: uma plataforma de ensino de libras.

IRATI
2022

RESUMO

Atualmente há uma dificuldade em realizar o aprendizado de libras, por diversos motivos, sejam eles por barreira cultural ou mesmo por falta de acesso.

Segundo pesquisas apontam que cerca de 98% das pessoas que possuem deficiência auditiva, outro fator relevante são as pessoas que não conseguem estudar por conta dessa barreira, cerca de 30% das pessoas não possuem escolaridade, tendo em vista esse problema este projeto visa desenvolver uma ferramenta para auxiliar no ensino e a disseminação de libras, através de duas opções sendo elas um quiz e a outra um painel de comunicação.

Para a construção desse *software* iremos utilizar a metodologia chamada de *FDD* que com suas ferramentas irá auxiliar na entrega rápida para o cliente, diferente no FDD normal que é composto por uma equipe de diferentes cargos, o projeto será desenvolvido somente por um desenvolvedor, para complemento das será desenvolvido em um arquitetura baseada em microserviços, juntamente com o paradigma de orientação a objetos.

Foi realizado uma comparação de 3 sistemas que visam auxiliar os surdos, o primeiro comparado é um site que disponibiliza notícias e tem um quiz para exercitar sua comunicação, o segundo ele é um aplicativo que possuem algumas ferramentas de consulta para o usuário e por fim o último que também é um app, sua funcionalidade também é de disponibilizar um quiz para interação dos usuários.

Dentro dos levantamento de requisitos é utilizado duas formas para obter as informações para o projeto, sendo através de um *Brainstorming* e de entrevistas, os dois foram realizados pela intérprete de libras que atua no campus IFPR Irati, além das modelagem que foram utilizadas, de primeiro momento um diagrama de classes de uso para auxiliar a enxergar os papéis dos usuários, seguido por diagramas de classe, atividades e objeto.

Neste projeto são utilizadas algumas ferramentas importantes como o *Typescript* para desenvolvimento dos dois lados, tanto do *front-end* quanto do *back-end*, além de ferramentas para controle de qualidade como o *jenkins* entre outros.

Por fim, foi desenvolvido dos cronogramas separados para auxílio do entendimento do prazo de entregas da parte da documentação e programação, sendo organizadas por ordem de prioridade.

Palavras-chave: *Feature Driven Development*, Microserviços, Levantamento de Requisitos, Cronograma, Diagrama.

Lista de Ilustrações

Figura 1 - Pesquisa sobre pessoas que sabem falar libras.	3
Figura 2 - Metodologia FDD	8
Figura 3 - Gráfico sobre microsserviços	10
Figura 4 - Quiz	12
Figura 5 - Apps recomendados	13
Figura 6 - Quiz	13
Figura 7 - Tradutor	14
Figura 8 - Leitor de objetos	16
Figura 9 - Dicionário	17

LISTA DE TABELAS E QUADROS

Quadro 1 - Quadro de comparações	16
Quadro 2 - Ferramentas Utilizadas.	29
Quadro 3 - Cronograma da Documentação.	30
Quadro 4 - Cronograma da programação.	31
Quadro 4.1 - Cronograma de Programação	32

LISTA DE ABREVIATURAS

FDD	<i>Feature Driven Development</i>
TDD	<i>Test Driven Development</i>
RF	Requisito Funcional
RN	Requisito de Negócio
RNF	Requisito Não Funcional
POO	Programação Orientada a Objeto
OMS	Organização Mundial de Saúde

Sumário

Capítulo 1 - Introdução	1
1.1 - Contextualização do problema	1
1.2 - Motivação	1
1.3 - Objetivo	2
1.4 - Justificativa	2
1.5 - Organização do texto	4
1.6 - Conclusão	4
Capítulo 2 - Fundamentação Teórica	5
2.1 - Introdução	5
2.2 Levantamento de Requisitos	5
2.3 - Metodologias Ágeis	6
2.3.1 - Feature Driven Design (FDD)	6
2.3.2 - FDD Adaptado	9
2.4 Arquitetura	9
2.5 - Paradigmas de Programação	10
2.6 - Linguagem de Modelagem	12
2.7 - Trabalhos Relacionados	12
Librasol	13
Quiz de Libras	14
Librasil	15
Considerações	16
2.8 - Conclusão	17
Capítulo 3 - Levantamento de Requisitos.	18
3.1 - Introdução	18
3.2 - Descrição detalhada do cenário/problema	18
Previsão de utilização do sistema	19
3.3 - Diagrama de Caso de Uso	19
3.3.1 - Descrição dos requisitos e regras de negócio.	19
Mapeamento: Caso de Uso para Requisitos Funcionais	19
Capítulo 4 - Modelagem	27
4.1 - Introdução	27
4.2 - Diagrama de atividade	27
4.3 - Diagrama de classe	28
4.4 - Diagrama de Objetos	28
4.5 Conclusão	29
Capítulo 5 - Desenvolvimento	29
5.1 - Introdução	29

5.2.Ferramentas Utilizadas	30
5.5 - Cronogramas	30
5.5.1 - Referente às etapas de desenvolvimento da documentação	31
Quadro 3 - Cronograma da documentação	31
5.5.2 - Referente ao processo de desenvolvimento.	31
Capítulo 6 - Conclusão	32
6.1 - Contribuições deste trabalho para o cenário.	32
6.2 - Considerações finais	32
6.3 - Trabalhos Futuros	32
REFERÊNCIAS	32

Capítulo 1 - Introdução

1.1 - Contextualização do problema

Atualmente uma das maiores dificuldades para quem sofre de surdez está em ter acesso às informações, essa dificuldade está associada a muitos fatores seja pela barreira de comunicação, por uma comunicação presencial ou mesmo através do uso da internet.

Além desse problema os surdos sofrem diversas dificuldades relacionadas a parte de cultura e lazer, pois existem poucas ações inclusivas desenvolvidas para que as pessoas surdas tenham acesso ao lazer e cultura (ACADEMIA DE LIBRAS, 2019).

Segundo Felipe Franco, o ensino essencial para surdos tem sido ausente no quesito escolar. Sendo assim o resultado é uma defasagem de alunos que possuem essa deficiência, segundo o MEC, uma pesquisa realizada em 2012 havia cerca de 27 mil alunos surdos matriculados, no entanto hoje esse número caiu cerca de 20%, também relata sobre a ausência de libras nas escolas ser um problema para quem necessita do ensino, também é preocupante para intérpretes não é um curso muito visado, visto que apenas 7 de 59 universidades federais oferecem o curso de graduação para libras (CODIGO INF, 2021).

1.2 - Motivação

Segundo Thiago Vieira (AGENCIA NACIONAL, 2016) que são levados para que seja ainda mais difícil a aprendizagem e a questão cultural da população brasileira, são destacados as seguintes dificuldades:

- Resistência dos pais que não possuem deficiência auditiva.
- Dificuldade de formação bilíngue dos pais e professores.
- No ambiente escolar a mudança de hábito de todo corpo docente.
- Falta de formação por parte de profissionais da educação.

Outro ponto relevante é a falta de tradutores e intérpretes de libras, como descreve a matéria escrita pela câmara dos deputados no qual explica como essa lei prevê que as escolas federais devem garantir serviços aos alunos surdos, já

nas demais escolas sendo elas privadas ou públicas é recomendado esse acesso.(CAMERA DOS DEPUTADOS, 2016)

1.3 - Objetivo

Auxiliar na disseminação de libras implementando uma ferramenta/sistema que auxilie no estudo/aprendizado/disseminação de libras por parte das pessoas interessadas. Para viabilizar esse cenário, será desenvolvido um programa o qual pretende viabilizar que, fluentes e pessoas interessadas na linguagem de libras, tenham acesso a um ambiente adequado de compartilhamento de conhecimento.

Objetivos Específico:

- Reunir usuários para aprendizado através de perguntas e respostas de um quiz
- Reunir fluentes e não fluentes para um ambiente comum de aprendizado.
- Fazer com que pessoas fluentes possam contribuir para comunidade de linguagem de libras.

1.4 - Justificativa

Atualmente há uma dificuldade em que pessoas se sensibilizem ou mesmo pessoas que possuam deficiência tenham habilidade em se comunicar utilizando a linguagem de libras. Porém, esta sensibilização é de extrema importância, sendo que a mesma é a língua oficial dentro do Brasil *segundo a definição da Lei Nº 10.436, de 24 de abril de 2002, em seu Artigo 1, parágrafo único:*

“Entende-se como Língua Brasileira de Sinais – Libras a forma de comunicação e expressão, em que o sistema linguístico de natureza visual-motora, com estrutura gramatical própria, constituem um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil.”

Dados gerados pela Organização Mundial da Saúde(OMS) em uma pesquisa realizada em 2015, apontam que no Brasil cerca de 14% da população brasileira apresenta surdez ou problemas auditivos(JORNAL USP, 2017).

Segundo Ricardo Bento profissional professor titular de Otorrinolaringologia da Faculdade de Medicina (FM) da Universidade de São Paulo e profissional

entrevistado na matéria, esse número tem a tendência a aumentar, pois a vida média dos brasileiros aumentaram nos últimos anos e com o passar da vida a parte auditiva do indivíduo acaba tendo perdas significativas na audição(JORNAL USP, 2017).

Quando o assunto é educação, as pessoas surdas se enquadram em uma porcentagem muito baixa de formação. Estudo feito pelo Instituto Locomotiva e a Semana da Acessibilidade Surda, 7% são brasileiros que possuem ensino superior completo, 15% tem o ensino médio, 46% foram até o fundamental, e 32% não possuem grau de escolaridade(ASSEMBLEIA DE SÃO PAULO, 2021).

Outra pesquisa realizada pelo Instituto Brasileiro de Geografia e Estatística(IBGE) foi realizada uma pesquisa nacional para verificar a quantidade de pessoas que fazem o uso da Língua Brasileira de Sinais(Libras), foi realizada com integrantes com mais de 4 anos de idade, apresentavam os seguintes números, 1,8% sabe usar a língua, 3% se comunicam mais com muita dificuldade e 64,2% não sabem falar de maneira nenhuma, como mostra a imagem abaixo:

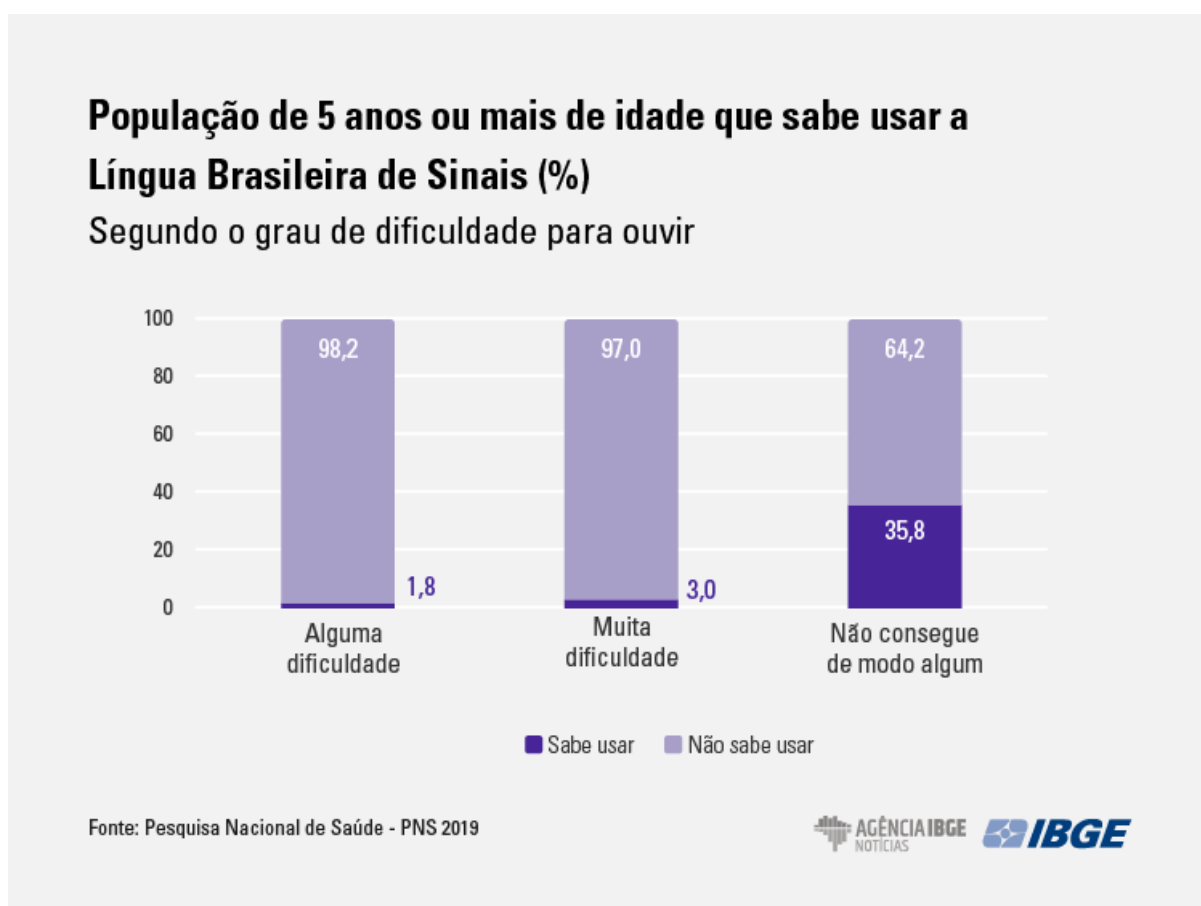


Figura 1 - Pesquisa sobre pessoas que sabem falar libras.

1.5 - Organização do texto

No capítulo seguinte será descrito a referência teórica desde a metodologia escolhida, seus papéis como foi adaptada para este projeto, arquiteturas escolhidas, paradigmas e linguagem de modelagem, no capítulo 3 contém uma descrição de como foi realizado o levantamento de requisitos, desde a pesquisa, formas e como foi planejado para levantar as principais demandas para o projeto, no capítulo 4 estão os modelos disponibilizados pela *UML* (Linguagem de Modelagem Unificada), tendo os diagramas de caso de uso, classe e objetos; por fim, no capítulo 5, será descrito como foi realizado as etapas tanto de escrita deste projeto quanto o desenvolvimento de *software*.

1.6 - Conclusão

Neste capítulo abordamos o motivo que será desenvolvido essa ferramenta, e qual público ela visa auxiliar, também foi abordado alguns fundamentos teóricos e pesquisas realizadas por órgãos públicos, e como será organizado o texto até o final do projeto sendo separado por 6 capítulos.

Capítulo 2 - Fundamentação Teórica

2.1 - Introdução

Neste capítulo será abordado como surgiu cada ferramenta utilizada dentro deste projeto, será descrito desde a escolha da metodologia, a abordagem utilizada no levantamento de requisitos e as tecnologias utilizadas de forma mais técnica e com informações de autores da área.

2.2 Levantamento de Requisitos

Para o levantamento de requisitos serão utilizadas duas formas diferentes para obtenção dos dados, o primeiro escolhido foi a entrevista pois se trata de uma técnica simples e que traz bons resultados na fase inicial do projeto(DEVMEDIA), para ela existem duas formas de ser conduzida, a primeira é chamada de entrevista estruturada quantos os tópicos já são pré-definidos, com o intuito de buscar respostas assertivas para aquela questão específica, e entrevista não estruturadas que é uma forma mais aberta de obter informações do entrevistado(MENTORING, 2016)

Brainstorming é uma técnica para geração de ideias. Ela permite que diversas pessoas possam contribuir para o cenário do problema trazendo os seguintes benefícios trazidos pela (DEVMEDIA)

- Seleção dos participantes: Os participantes selecionados devem contribuir diretamente durante a sessão. Trazer pessoas bem informadas, vindas de diferentes grupos garantirá uma boa representação;
- Explicar a técnica e as regras a serem seguidas: O líder irá conduzir explicando os conceitos básicos do *brainstorming* e as regras a serem seguidas durante a sessão;
- Produzir uma boa quantidade de ideias: As pessoas do grupo irão gerar diversas ideias dependendo da quantidade que foi exigida pelo tópico do *brainstorming*. Os participantes são convidados, um por vez, a dar uma única idéia. Se alguém tiver problema, passa a vez e espera a próxima rodada.

2.3 - Metodologias Ágeis

Segundo Sbrocco(2012, p.87) os conceitos existentes hoje relacionados ao desenvolvimento ágil que existiam por volta de 1990, motivados por conta de métodos que os travavam, caracterizado por muita documentações rígidas e planejamentos muito antecipados, sendo assim desenvolvedores viram a necessidade de criar novos métodos, denominados como métodos ágeis.

Esse problema já tinha se tornado crônico, era comum um atraso de vários anos entre a validação de uma necessidade de negócios pela empresa e a entrega de uma aplicação funcional para supri-la. Durante esse período, as mudanças nos mercados e nas demandas corporativas causaram o cancelamento de partes significativas dos projetos de aplicações antes mesmo da entrega. Esse desperdício de tempo e recursos fez com que vários desenvolvedores buscassem uma alternativa.

Em 2001, ocorreu uma reunião no qual marcou o surgimento desses novos paradigmas de desenvolvimento, segundo o manifesto ágil eram eles: Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas. Estas pessoas representam várias metodologias já utilizadas, como *SCRUM*, *Extreme Programming*, *DSDM*, *Adaptive Software Development*, *Crystal*, *Feature Driven Development*, entre outras. O documento original, e seus criadores, estão no site do manifesto ágil e se encontra até hoje no ar, com a última interação em 2016.(MANIFESTO AGIL, 2001)

2.3.1 - Feature Driven Design (FDD)

A metodologia *Feature Driven Development* (FDD), que em sua tradução literal significa Metodologia Dirigida a Funcionalidades, foi criada por Jeff de Luca e Peter Coad. A FDD é uma metodologia ágil, robusta e muito utilizada nos dias atuais. A motivação para criação dessa metodologia deu-se no contexto do sistema bancário internacional da *United Overseas Bank*, que segundo Sbrocco e Macedo (SCROCCO, p.99) inicialmente foi considerado inviável de ser desenvolvido no prazo predeterminado. A eficiência da metodologia *FDD* foi então demonstrada quando o sistema que contava com 2000 funcionalidades foi entregue em 15 meses por uma equipe de 50 pessoas.

A *FDD* proporciona algumas características bem marcantes, a primeira é que atende equipes de diferentes tamanhos, pois a mesma oferece um conjunto sólido de atribuições capazes de ser ampliadas se necessário, mesmo em projetos descentralizados ou até mesmo em projetos *web*.

A produção do software engloba um conjunto de métricas de qualidade a serem aplicadas durante o desenvolvimento do projeto, além de aplicar o *Test Driven Development* (Desenvolvimento orientado a testes), como base para medir a qualidade do projeto(DEV MEDIA).

Com esta metodologia, se consegue proporcionar uma forma de trabalho que agrada a todos os envolvidos no projeto, sugerindo formas de interação e controle fácil. Os desenvolvedores se sentem muito à vontade, relata Sbrocco(2012, p.103) durante a implementação pois a *FDD* possui um conjunto de regras de fácil entendimento e de resultados rápidos, trazendo também vantagens para o cliente. Com a entrega por funcionalidades, disponibilizando elas para o cliente, é possível ter uma métrica se a funcionalidade foi realizada da maneira que o cliente considera como prioritária.

É importante dizer que o *FDD* é composto por cinco etapas, onde as três primeiras só ocorrem uma única vez, já as duas últimas ocorrem a cada interação. São cinco os processos da metodologia ágil *FDD*:

- desenvolver um modelo Abrangente;
- construir uma lista de funcionalidades;
- planejar através de funcionalidades;
- projetar através de funcionalidades;
- construir através de funcionalidades (CEDRO TECH, 2019).

Essa prática sugere que a partir dos requisitos construa-se uma lista de funcionalidades decompondo funcionalmente o domínio do negócio. Categorizado em três níveis:

- Áreas de negócios;
- Atividades do negócio;
- Passos da atividade

Os desenvolvedores devem sempre reconstruir o software ou seja ativar as funcionalidades e modificações realizadas, assim os demais desenvolvedores utilizam e trabalham em uma versão atualizada do projeto. Os clientes também se beneficiam utilizando sempre a última compilação do projeto. A metodologia prega essa sequência, ou seja, o cliente sempre estará com o sistema atualizado, e sempre terá o feedback instantâneo para funcionalidade, Assim como

mostra a Figura 1, que demonstra em uma linha reta todo o passo a passo até chegar a entrega.



Figura 2 - Metodologia *FDD*

Para Sbrocco(2012, p. 101) são listados 5 funções papéis muito claros que são eles: gerente de projeto, arquiteto-chefe, equipe de modelagem, programador-chefe e equipe de funcionalidades. Nesta seção será explicado o que cada papel tem como obrigação dentro da equipe, para

- Gerente de Projeto: Tem como função o contato direto com o cliente e captar todos os requisitos, bem como as possíveis restrições. Cabe ao mesmo formar a equipe de desenvolvimento de acordo com as funcionalidades, acompanhar todo o desenvolvimento, atentando sempre as boas práticas do *FDD*, o gerente também deve escrever um estudo completo juntamente com especialistas sobre as regras de negócio do projeto para que os demais desenvolvedores e membros da equipe tomem ciência de como tudo funciona;
- Arquiteto-chefe/especialista: A equipe deve possuir um profissional especialista no assunto do projeto a ser desenvolvido, para sanar;
- Equipe de modelagem: Definido pelo gerente é capaz de elaborar a lista de funcionalidades(*features*) do sistema em questão. Esse profissional deve dominar, entre outras habilidade, técnicas de modelagem de sistemas com conhecimento de *UML* suficiente para trabalhar com diversos diagramas;
- Programador-chefe: Esse colaborador deve refinar a lista de features e transformá-las em modelo de objetos. Deve também organizar os trabalhos, escolhendo a linguagem e as formas de armazenamento bem como as integrações *build*, revisando e aprovando a liberação para o cliente;

- Equipe de funcionalidades: São os desenvolvedores propriamente ditos, profissionais que devem conhecer as ferramentas de desenvolvimento necessários para o desenvolvimento do projeto, além de como o sistema funciona;

2.3.2 - FDD Adaptado

Diferente de uma equipe corporativa o FDD teve que ser adaptado por falta de recurso humano, o único que será responsável pelo desenvolvimento de todos os passos da metodologia acaba sendo aplicada pelo autor, as funcionalidades serão testadas por usuários próximos sendo colegas de trabalho e amigos próximos além de reunir o feedback de usuários que possuem a deficiência.

A priori o papel do cliente será desempenhado pela intérprete de libras do campus IFPR Irati, relatando o que é necessário para a melhora e principalmente relatar sua experiência na área. Apesar de ser um nicho bem específico futuramente será disponibilizado para que mais pessoas possam utilizá-lo e os mesmos vão passando feedbacks para melhoramento contínuo do produto, conforme prega a metodologia, como trabalho futuro será disponibilizado uma forma de comunicação para que as solicitações dos usuários possam ser atendidas.

2.4 Arquitetura

A arquitetura escolhida para este projeto é baseada em micro serviços para Martin Fowler é uma abordagem para desenvolver um único aplicativo porém separado em pequenos pedaços cada um tendo uma responsabilidade própria e única, os micro serviços oferecem alguns benefícios como(MARTIN FOWLER, 2021):

- *“Limites fortes do módulo : os microsserviços reforçam a estrutura modular, o que é particularmente importante para equipes maiores.”*
- *“Implantação independente : serviços simples são mais fáceis de implantar e, como são autônomos, são menos propensos a causar falhas no sistema quando dão errado.”*
- *“Diversidade de tecnologia : com microsserviços, você pode combinar vários idiomas, estruturas de desenvolvimento e tecnologias de armazenamento de dados.”*

Outro ponto que destaca é a produtividade, como mostrada abaixo pela figura 3, mostrando que no início um projeto monolítico, que consiste em todo projeto seja o *front-end* (lado do cliente) a página que é visualizada através de um navegador *back-end* (lado do servidor), lado do processamento de dados. Tendo essa introdução no começo nota-se que o monolítico leva uma vantagem por ser mais simples de ser aplicado, porém com o passar do tempo ele vai decaindo a produtividade, diferente do micro-serviço que com o passar do tempo consegue se manter estável na produtividade.

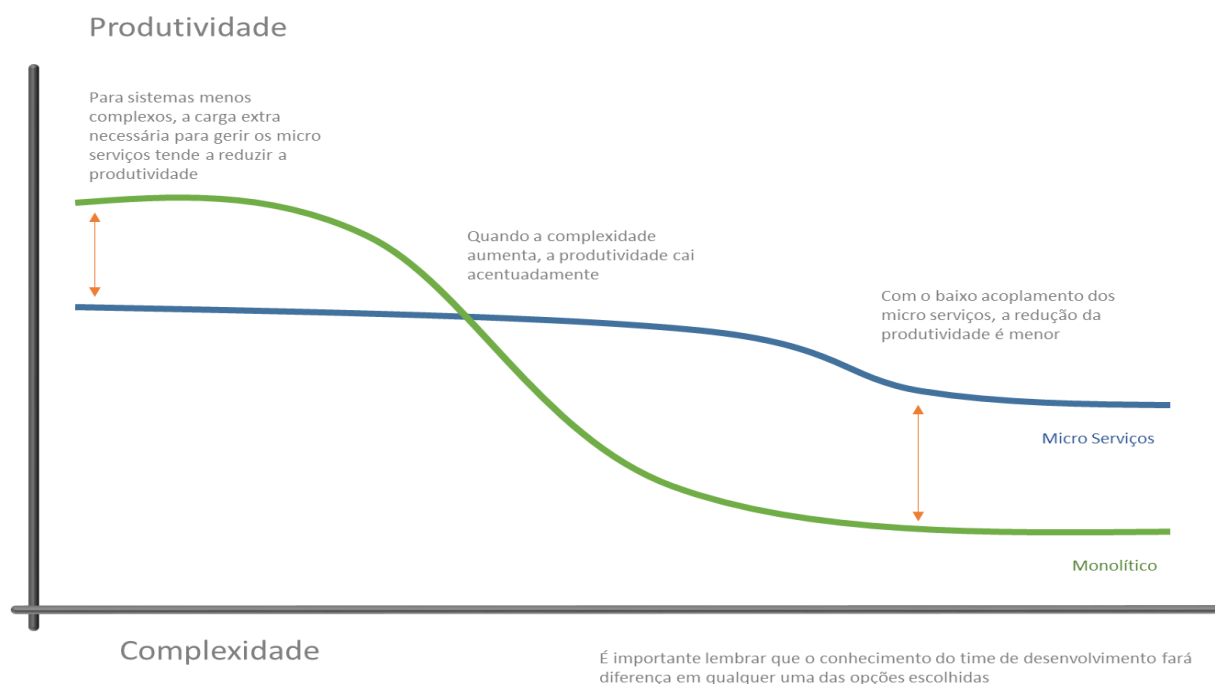


Figura 3 - Gráfico sobre microserviços

2.5 - Paradigmas de Programação

Neste projeto o paradigma de escolha é a Orientação a Objeto pois a mesma tem como objetivo não repetir códigos, separar entre funcionalidades e cada objeto é tratado como individual o que será necessário para este trabalho, caso outros desenvolvedores desejem contribuir para o projeto, com ela consigo encapsular funções para facilidade de localização dentro do mesmo.

Ela surgiu como uma alternativa a essas características da programação estruturada. O intuito da sua criação também foi o de aproximar o manuseio das

estruturas de um programa ao manuseio das coisas do mundo real, daí o nome "objeto" como algo genérico, que pode representar qualquer coisa tangível(ALURA, 2022).

A programação estruturada, quando codificada de maneira correta, ela terá um desempenho maior ao da orientação a objetos, o motivo é que por ser um paradigma sequencial, cada linha de código é executada uma após a outra, tendo poucos desvios, diferente da POO(Programação Orientada a Objetos), além da programação estrutura ter uma compatibilidade melhor com o uso dos hardwares.(DEV MEDIA).

Porém com o POO ele acaba trazendo pontos muito fortes, principalmente para as aplicações modernas, como o desempenho das aplicações não é o fator mais preocupante devido a limitações das máquinas atuais, o que facilitou na utilização de POO por seu principal conceito que é a reutilização de código(DEV MEDIA).

Abaixo como mostra na figura 3, ao lado esquerdo temos um diagrama apontando como funciona a programação estruturada, basicamente sendo composta por uma camada maior sendo os dados globais, dividindo em diferentes procedimentos, já na orientação a objetos, cada objeto é único, ou seja cada objeto terá sua responsabilidade não passando os dados globalmente, o que facilita outro conceito muito importante em poo que é o encapsulamento das funcionalidades, ou seja cada objeto tem sua função única e específica.

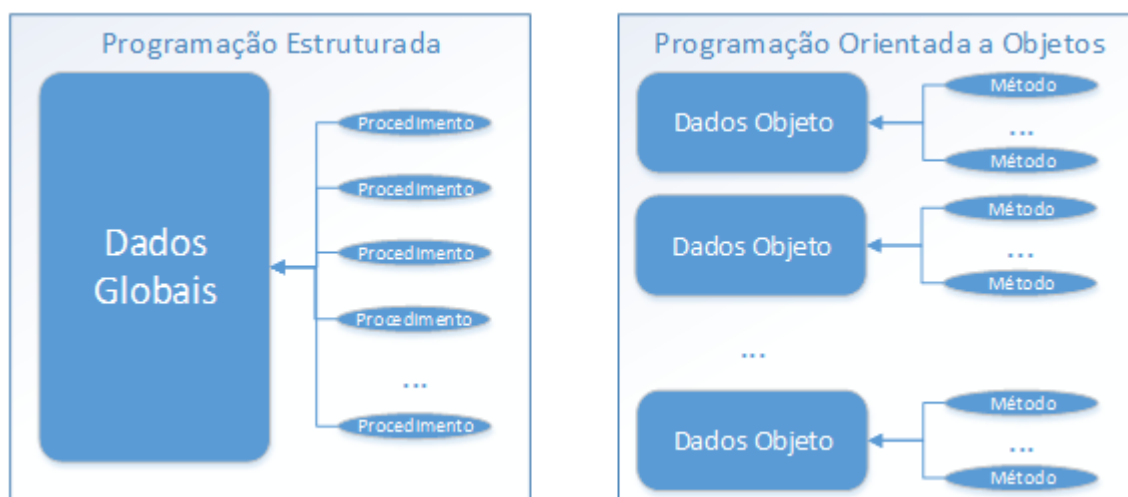


Figura 3 - Programação Estruturada vs Programação Orientada a Objetos

2.6 - Linguagem de Modelagem

A linguagem a ser utilizada é a Unified Modeling Language(*UML*) tem o surgimento através do sucesso de sistemas grandes e complexos.

Os esforços começaram no início de outubro de 1994 quando Rumbaugh se juntou a Booch em uma empresa chamada Rational, com o objetivo de de unificar os métodos de duas empresas Booch e OMT, decorrido o trabalho foi lançado a primeira versão declarada como *Unifed Process*(Processo Unificado), nesta mesma época Jacobson se associou a Rational e o escopo da tarefa acabou se tornando a UML. Porém somente em 1997 a UML foi aprovada pelo Object management Group, que consiste em um grupo de empresa que ratifica padrões de orientação a objetos.(PROJETO DIÁRIO, 2013).

Os diagramas da UML se dividem em dois grandes grupos: diagramas estruturais e diagramas comportamentais.

A uml ela é separada por dois tipos de diagramas, chamados de Diagramas comportamentais que são encarregados pela parte dinâmica do sistema, ela será encarregada por mostrar as funcionalidades, como o processo do negócio deve ser tratado, comportamentos, trocas de mensagem e como responde elas, e o diagrama estrutural, que ta para parte estática do sistema ou seja como será estruturado uma classe, métodos, interfaces, serviços, componentes e arquitetura do sistema(ATÉ O MOMENTO, 2019)

2.7 - Trabalhos Relacionados

Neste capítulo serão apresentados três aplicativos/páginas web no qual será descrito o que ele oferece as vantagens e as desvantagens dos mesmos. Como será mostrado duas formas de auxílio à aprendizagem, foi focado mais a pesquisa na área de quiz do projeto, pois a outra plataforma de ensino que se trata de reunir pessoas para treinar libras, não constam hoje no mercado as pesquisas realizadas foram através de plataformas de pesquisa como o *google* e através da *play store*(loja de aplicativos do sistema operacional *android*)

Librasol

O Librasol é uma página web que tem como objetivo: trazer vários conteúdos relacionados a libras, manter atualizado às pessoas que possuem essa deficiência trazendo notícias já na página inicial(figura 4), navegando a página temos a disposição uma lista de aplicativos para o ensino(figura 5) fornecendo um quiz para o usuário treinar a sua fluência; possui as legislações as quais contemplam o surdo concentrando assim toda a informação necessária para o mesmo.



Figura 4 - Quiz

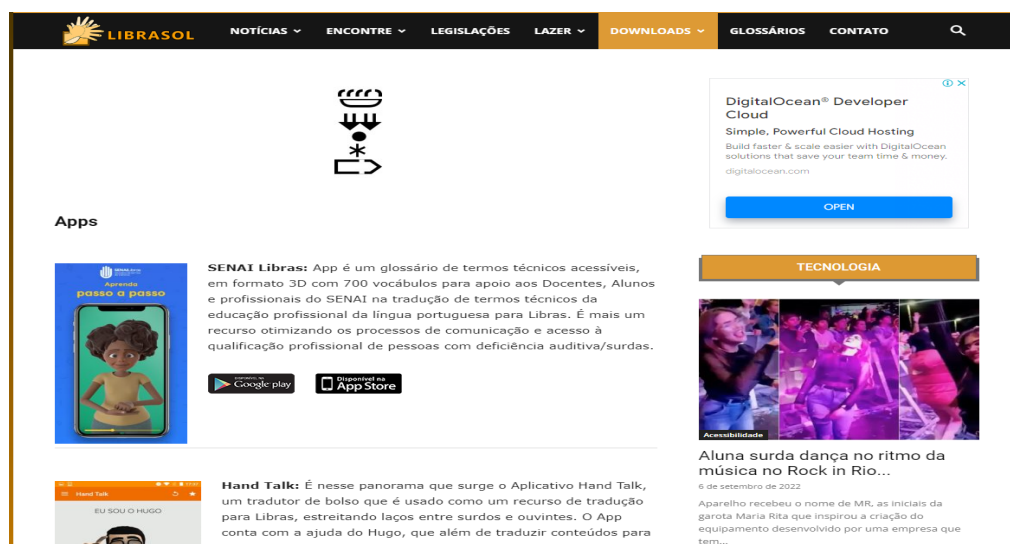


Figura 5 - Apps recomendados

Quiz de Libras

O Quiz de libras é um aplicativo disponibilizado apenas para mobile, pode ser baixado através do play store. Este tem o objetivo de apresentar questões para treinar seu nível de libras com base em temas e situações(figura 6), também há uma parte de tradutor(figura 7), no qual disponibiliza algumas imagens que podem te auxiliar na comunicação com outra pessoa.

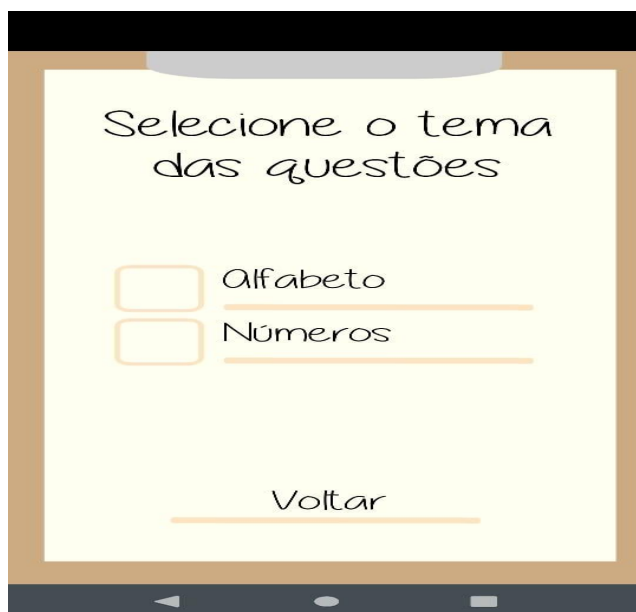


Figura 6 - Quiz

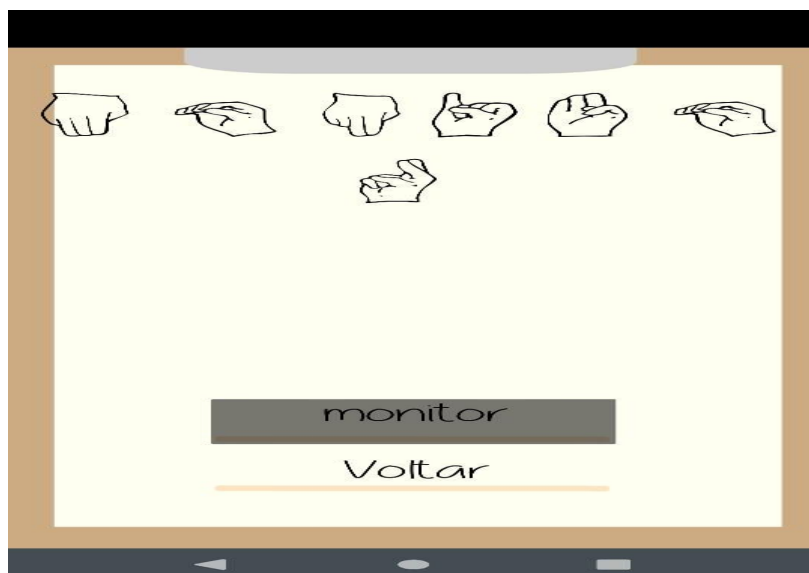


Figura 7 - Tradutor.

Librasil

Librasil consiste em disponibilizar um dicionário para o usuário com diferentes palavras do dia a dia, traz uma descrição de como deve ser a posição da mão e do rosto(figura 9), assim auxiliando o usuário a reproduzir o que ele precisa, também há uma forma de detecção de imagem, que através da sua câmera de celular(figura 8), você consegue tirar fotos e o mesmo indica o que pode ser e a precisão tem sobre a imagem.

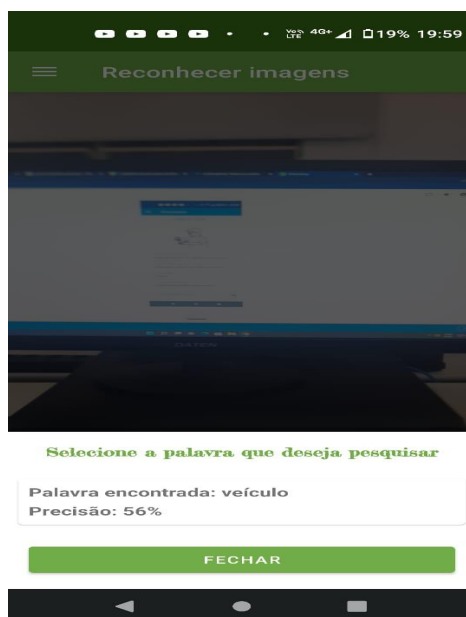


Figura 8. Leitor de objetos

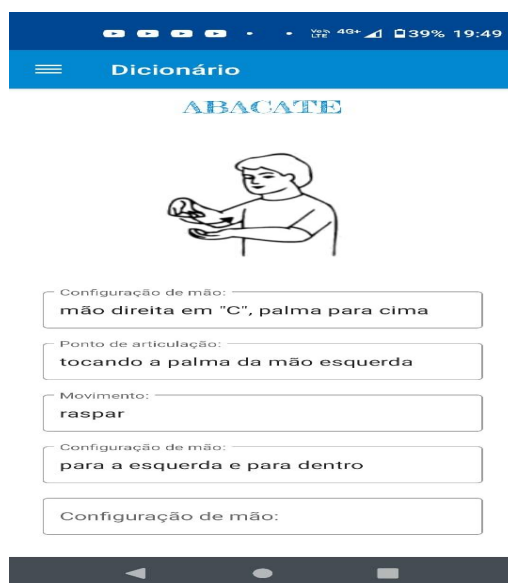


Figura 9 - Dicionário

Considerações

Librasol passa a experiência intuitiva para acessar as informações e oferece muitas informações sobre esse nicho específico, disponibilizando para o usuário um acervo bem grande de informações sobre a área, outro ponto positivo é a responsividade. Em dispositivos menores (e.g., *smartphones*) este é mais “preciso” do que em telas de computadores de mesa. Outro ponto negativo se refere a quantidade de anúncios que o mesmo possui em sua página web poluição visual demasiada acarretada pela quantidade de propagandas/anúncios).

A experiência que tive com o Quiz de Libras foi confusa, pois ele disponibiliza uma métrica de pontos conforme o acerto da questão, porém caso o usuário erre, ele volta para questão não indicando o erro, mais sim como se tivesse voltado para o estado inicial, o qual me gerou uma confusão inicial. O tradutor, de fato, foi útil, pois pessoas que não tem nenhuma base, podem treinar e tentar ser objetivo para aprender o contexto que ela mais precisa utilizar essa linguagem.

Com o Librasil a experiência não é muito agradável pois, ao acessar o dicionário de palavras, este apresenta certa lentidão para buscar as mesmas. Este sistema não têm indicação de “carregando o conteúdo”, o qual passa a impressão de um possível erro. Adicionalmente, a funcionalidade de detecção de imagem também não é precisa, através da imagem(figura 10) mostra uma situação que tirei foto de um monitor de computador, e o mesmo trouxe uma informação nada correspondente ao mesmo, abaixo foi realizado um quadro sobre alguns pontos de cada *software*.

	Librasol	Librasil	Quiz de Libras
Ambiente	Web	Mobile	Mobile
Anuncios	Sim	Não	Não
Informações sobre surdez	Sim	Não	Não
Dicionário de Palavras	Não	Sim	Sim
Detectar Objetos	Não	Sim	Não
Quiz	Sim	Não	Sim
Responsividade	Sim	Sim	Sim

Quadro 4 - Quadro de comparações

2.8 - Conclusão

Neste capítulo foram levantados o que são metodologias ágeis, a respectiva metodologia escolhida para o projeto (i.e., FDD), a forma que a mesma foi adaptada para o desenvolvimento de apenas um programador. Adicionalmente, também foi levantado como será feita a arquitetura do projeto (i.e., arquitetura de microserviços) Por fim, nesta seção foi abordado a linguagem de modelagem (i.e., UML) algumas soluções do mercado e a experiência que tive enquanto usuário destes sistemas.

Capítulo 3 - Levantamento de Requisitos.

3.1 - Introdução

Neste capítulo serão levantadas as formas que foram capturadas as necessidades relatadas pelos usuários, que se inicia por uma entrevista realizada pela intérprete do CAMPUS IFPR Irati, como o aluno em que ela realiza a interpretação teve sua experiência com o aplicativo, também serão relatados os resultados das pesquisas quantitativas e como foi divulgado para obtenção de respostas.

Também terá uma descrição detalhada de um diagrama de caso de uso, que relata em alto nível, qual o papel do ator do *software* e como será sua interação, seguida por uma lista de funcionalidades e suas descrições detalhadas.

3.2 - Descrição detalhada do cenário/problema

Para resolver o problema levantado, será necessário realizar um módulo de cadastro de usuários, sendo divididos em usuário comum, fluente, tal que o fluente deverá passar por uma análise e ser aceito por um usuário administrador.

O processo de login tendo a possibilidade do usuário se cadastrar na plataforma, podendo selecionar a opção de usuário comum ou fluente, caso opte pela opção do fluente, o mesmo deverá ser notificado que passará por uma avaliação, também será necessário que o usuário encaminhe um pequeno vídeo se apresentando para os administradores, além de conter as opções de esqueci o usuário e senha.

Ao realizar a autenticação, deverá apresentar duas plataformas separadas, um quiz inicialmente disponibilizando pequenos vídeos ou gifs para interpretação do usuário, seguido por alternativas que podem auxiliar na resposta correta do mesmo, a cada resposta correta deverá ter um contador de experiência para que seja retirado um relatório de usuários que mais acertaram durante um período ou mesmo alguma forma de aquisição de benefícios dentro do quiz, como direito a ter uma resposta incorreta sem perder experiência. Para alimentar o banco de dados de perguntas e respostas, será necessário um aplicativo móvel para gravação desses vídeos possibilitando o mesmo a adicionar as questões que podem conter aquela pergunta cadastrada, a cada pergunta cadastrada deverá passar por um processo de aprovação pelo administrador.

E um painel de comunicação, que inicialmente o usuário cadastrado como fluente, poderá ter seu nome listado para que usuários comuns possam agendar ou interagir em tempo real, o software deverá conter um pequeno vídeo do fluente se apresentando, uma breve descrição sobre seu conhecimento e interesses, por fim a página deverá conter um calendário listando as datas disponíveis para agendar um horário para conversação com o fluente, após agendado deverá gerar um link com uma reunião a priori pelo *Google Meet*.

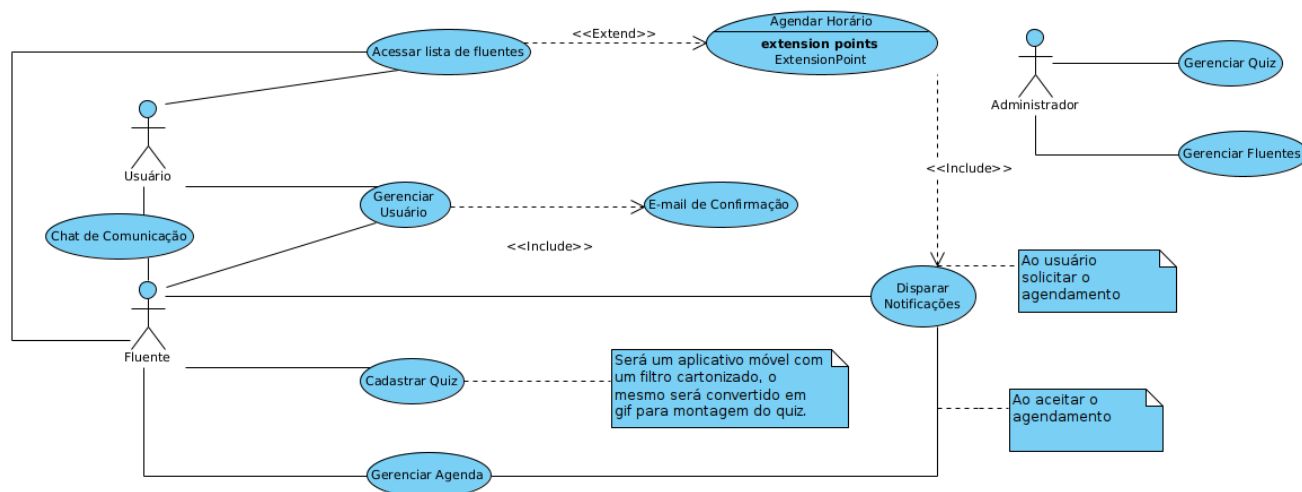
Previsão de utilização do sistema

Logo após o desenvolvimento das funcionalidades de login e quiz, definidas como prioridade, já estará disponível inicialmente para um uso fechado de alunos do ifpr, notificando a intérprete para divulgação da plataforma.

3.3 - Diagrama de Caso de Uso

O diagrama conta com três atores, descritos como usuário, fluente e administrador e um total de 12 funcionalidades. O usuário será responsável pelo gerenciamento do usuário dentro do sistema, funcionalidades como cadastrar e inativar o usuário, irá poder realizar o acesso ao painel e terá disponibilidade de acessar a lista de fluentes, que clicando em um fluente e acessando seu calendário, pode realizar o agendamento, também pode clicar na opção para abrir um chat online, que poderá conversar em tempo real. por fim terá a disponibilidade de acessar o quiz clicando nos cenários disponíveis para responder uma das questões disponibilizadas para entretenimento.

Para os fluentes eles poderão cadastrar um quiz, que será disponibilizado um aplicativo mobile para o mesmo, ao realizar seu cadastro terá que ficar pendente de uma aprovação do usuário administrador, poderá gerenciar sua agenda de usuários que desejam treinar a sua conversação, e por fim o usuário administrador que ficará responsável por gerenciar os quizzes e usuários que são cadastrados no painel.



3.3.1 - Descrição dos requisitos e regras de negócio.

Mapeamento: Caso de Uso para Requisitos Funcionais

Caso De Uso	Requisito Funcional
UC01 - Gerenciar Usuário	RF001: Cadastrar usuário. RF002: Enviar link de confirmação do usuário.

UC02 – Gerenciar Quiz	RF003: Cadastrar quiz com perguntas e respostas. RF004: Aprovar ou reprovar se o quiz irá ficar disponível.
UC03 - Lista de Fluentes	RF005: Acessar o painel para comunicação entre usuário e fluente.
UC04 - Chat de comunicação	RF006: Opção de chat entre usuário e fluente.
UC05 - Acessar Quiz	RF007: Ganho de experiência para o usuário. RF008: Responder perguntas e respostas.
UC06 - Gerenciar Fluentes	RF009: Poder aceitar fluentes para a plataforma. RF010: Inativar fluentes da plataforma.
UC07 - Gerenciar Quiz	RF011: Aprovar/Rejeitar os quizzes cadastrados.
UC08 - Gerenciar Agenda	RF012: Aprovar/Rejeitar agendamento com os usuários.

Descrição, individual, dos Requisitos Funcionais

Caso De Uso:	UC01 - Gerenciar Usuário
Identificador:	RF001
Nome do requisito:	Cadastrar usuário
Módulo:	Módulo de cadastro
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – Cargo/Função:	Felipe Andrade;
Versão:	v 1.0
Descrição de Cenário Principal:	Esta funcionalidade estará disponível para que o usuário comum ou o fluente acabem se cadastrando na plataforma
Descrição de Cenário Alternativo:	Será possível que o Administrador do painel cadastre o usuário, caso o mesmo relate problemas em seu cadastro.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / Usuário (s):	Aluno, professor, administrador.
Prioridade:	Alta;
Informações de entrada:	Usuário: nome, idade, email e senha; Fluente: nome, idade, email, senha, tempo de fluência, anexo de comprovação de fluência. Administrador: Nome, senha.
Informações de saída:	Após o procedimento do cadastro de um cliente, o sistema deve informar ao ator se o cadastro foi concluído com sucesso.
Restrições lógicas:	Verificar se o e-mail está inválido; Validar se o e-mail já não consta na base de dados; Sugerir uma senha com letras e número.
Restrições tecnológicas:	Não há restrições.
Regras do negócio:	RN01: caso o usuário esteja se declarando como professor, será necessário da aprovação do administrador.
Requisitos não funcionais:	RNF01: o sistema deve possuir uma interface gráfica amigável e de fácil utilização; RNF02: o sistema de cadastro deve ser rápido;
Observações:	N/A;

Caso De Uso:	UC01 - Gerenciar usuários
Identificador:	RF002
Nome do requisito:	Enviar link de confirmação do usuário
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade;
Versão:	v01
Descrição de cenário principal	Para este requisito, ao cadastrar o usuário ou fluente será necessário que o mesmo realize ativação através do e-mail.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Usuário, Fluente.
Prioridade:	Alta;
Informações de entrada:	Não há entrada.
Informações de saída:	Ao confirmar o link, será notificado informando que o cadastro foi ativado com sucesso.
Restrições lógicas:	Caso o e-mail esteja inválido não disparar e retornar erro para o usuário.
Restrições tecnológicas:	N/A;
Regras do negócio:	Não há regras de negócio.
Requisitos não funcionais:	RNF01: o e-mail de ativação deve ser amigável;
Observações:	N/A;

Caso De Uso:	UC02 - Gerenciar Quiz
Identificador:	RF003
Nome do requisito:	Cadastrar quiz com perguntas e respostas
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade;
Versão:	v01
Descrição de cenário principal	Para este requisito o usuário poderá através de um aplicativo gravar e montar as perguntas e respostas.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Fluente, Administrador.
Prioridade:	Alta;
Informações de entrada:	Não há entrada.
Informações de saída:	Informar que o cadastro foi concluído com sucesso, e que está pendente de aprovação.
Restrições lógicas:	N/A;
Restrições tecnológicas:	N/A;
Regras do negócio:	N/A
Requisitos não funcionais:	RNF01: o video deve ser feito com filtro em cartoon; RNF02: interação de montar a pergunta e resposta.
Observações:	N/A;

Caso De Uso:	UC02 - Gerenciar Quiz.
Identificador:	RF004
Nome do requisito:	Aprovar ou reprovar se o quiz irá ficar disponível.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022

Autor – cargo/função:	Felipe Andrade
Versão:	v01
Descrição de cenário principal:	O usuário administrador irá receber uma lista de perguntas e respostas para aprovação.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Administrador.
Prioridade:	Alta;
Informações de entrada:	Um valor verdadeiro ou falso.
Informações de saída:	Informar que deu certo a operação.
Regras do negócio:	N/A;
Requisitos não funcionais:	RNF01: É preciso que o usuário tenha uma experiência visual agradável.
Observações:	N/A;

Caso De Uso:	UC03 - Lista de Fluentes
Identificador:	RF005
Nome do requisito:	Acessar o painel para comunicação entre usuário e fluente.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade;
Versão:	v01
Descrição de cenário principal	Para este requisito o usuário terá possibilidade de receber uma lista de todos os fluentes cadastrados no painel de comunicação.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Usuário.
Prioridade:	Baixa;
Informações de entrada:	Usuário e senha.
Informações de saída:	Lista de fluentes.
Restrições lógicas:	N/A;
Restrições tecnológicas:	N/A;
Regras do negócio:	N/A
Requisitos não funcionais:	RNF01: apresentar os fluentes de forma organizada.
Observações:	N/A;

Caso De Uso:	UC04 - Chat de comunicação
Identificador:	RF006
Nome do requisito:	Opção de chat entre usuário e fluente.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade
Versão:	v01
Descrição de cenário principal:	Será disponibilizado um chat entre o usuário e o fluente.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Usuário, Fluente.
Prioridade:	Baixa;
Informações de entrada:	Textos que o usuário enviar.
Informações de saída:	Textos que o outro lado deve receber.
Regras do negócio:	N/A;

Requisitos não funcionais:	RNF01: É preciso que o usuário tenha uma experiência visual agradável. RNF02: Precisa ser em tempo real.
Observações:	N/A;

Caso De Uso:	UC05 - Acessar Quiz
Identificador:	RF008
Nome do requisito:	Responder o Quiz.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade
Versão:	v01
Descrição de cenário principal:	O usuário terá a opção de responder o quiz.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Usuário, Fluente.
Prioridade:	Alta;
Informações de entrada:	Resposta selecionada.
Informações de saída:	Informação se a resposta está correta..
Regras do negócio:	N/A;
Requisitos não funcionais:	RNF01: É preciso que o usuário tenha uma experiência visual agradável. RNF02: Precisa ser em tempo real.
Observações:	Não há observações.

Caso De Uso:	UC06 - Gerenciar Fluentes
Identificador:	RF009
Nome do requisito:	Poder aceitar fluentes para a plataforma.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade
Versão:	v01
Descrição de cenário principal:	O Administrador terá uma lista de fluentes cadastrados dentro do painel, podendo aceitar ou não o mesmo.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Administrador.
Prioridade:	Baixa;
Informações de entrada:	Id do usuário e booleano aceitando ou não.
Informações de saída:	Operação realizada ou não.
Regras do negócio:	N/A;
Requisitos não funcionais:	N/A;
Observações:	N/A;

Caso De Uso:	UC06 - Gerenciar Fluentes
Identificador:	RF010
Nome do requisito:	Inativar fluentes da plataforma.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade
Versão:	v01

Descrição de cenário principal:	O Administrador terá uma lista de fluentes ativos do painel, tendo a possibilidade de inativar a qualquer momento.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Administrador.
Prioridade:	Baixa;
Informações de entrada:	Id do usuário.
Informações de saída:	Operação realizada ou não.
Regras do negócio:	N/A;
Requisitos não funcionais:	N/A;
Observações:	N/A;

Caso De Uso:	UC08 - Cadastrar Quiz
Identificador:	RF012
Nome do requisito:	Gravar um pequeno vídeo, enviar perguntas e a resposta correta.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade
Versão:	v01
Descrição de cenário principal:	O Fluente/Administrador poderá gravar um pequeno vídeo encaminhando as perguntas e a resposta sobre aquele quiz.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Fluente, Administrador.
Prioridade:	Alta;
Informações de entrada:	Id do usuário.
Informações de saída:	Operação realizada ou não.
Regras do negócio:	RN001: O vídeo deve ter o filtro cartoonizado;
Requisitos não funcionais:	N/A;
Observações:	N/A;

Caso De Uso:	UC09 - Gerenciar Agenda
Identificador:	RF013
Nome do requisito:	Aprovar/Rejeitar agendamento com os usuários.
Data de criação:	15/10/2022
Data da última alteração:	15/10/2022
Autor – cargo/função:	Felipe Andrade
Versão:	v01
Descrição de cenário principal:	O Fluente irá receber um e-mail sobre qual data e horário o usuário solicitou para a sessão de conversação.
Fontes:	Intérprete de Libras IFPR Irati – Sarah;
Ator (s) / usuário (s):	Fluente.
Prioridade:	Baixa;
Informações de entrada:	Booleano aceitando ou não.
Informações de saída:	Enviar um e-mail para o usuário informando a resposta.
Regras do negócio:	RN01: o e-mail deve ser respondido até 48 horas, caso não tenha sido será expirado e não poderá responder mais;
Requisitos não funcionais:	N/A;
Observações:	N/A;

Capítulo 4 - Modelagem

4.1 - Introdução

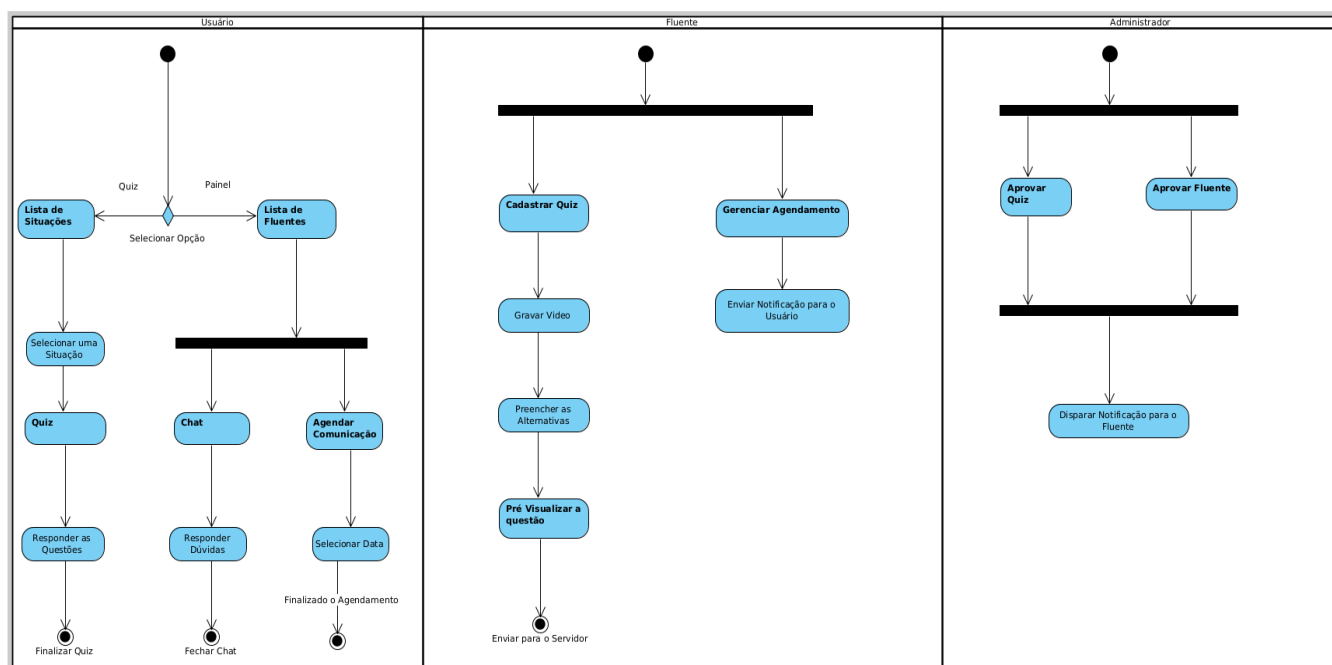
Nesta seção será mostrada a modelagem do sistema, iniciando com o diagrama de atividade, que irá disponibilizar uma visualização em forma de sequência, como a interação das funcionalidades e quais ações o ator terá dentro do *software*.

Seguindo temos o diagrama de classes, com o intuito de mostrar quais classes serão construídas dentro do projeto, de maior valor se dá às informações que contém em cada classes, contendo seus atributos e tipos e os métodos que

serão compostas por aquela classe. E por fim o diagrama de objetos que será um exemplo de dados que serão preenchidos dentro das classes, e qual contexto o software irá auxiliar.

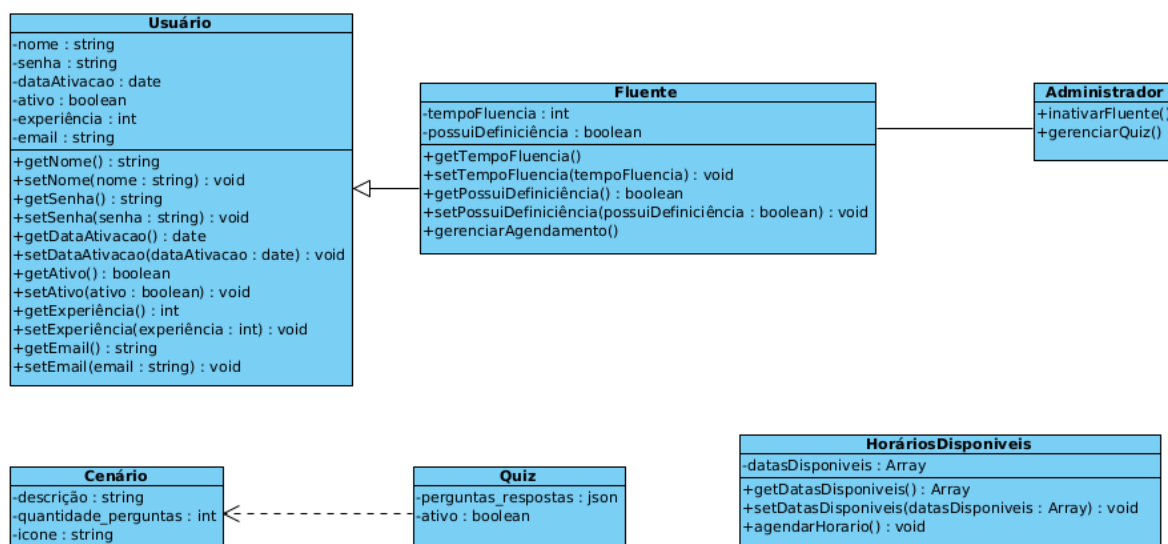
4.2 - Diagrama de atividade

Para este diagrama, foi representado o fluxo que cada ator terá dentro do projeto, tentando separar em três cenários, no qual o primeiro se dá pelo usuário que irá conter três etapas caso selecione o quiz, a lista de situações que serão cenários do cotidiano, ao selecionar uma situação irá acessar o quiz correspondente, clicado na resposta irá receber a notificação se acertou ou erro.



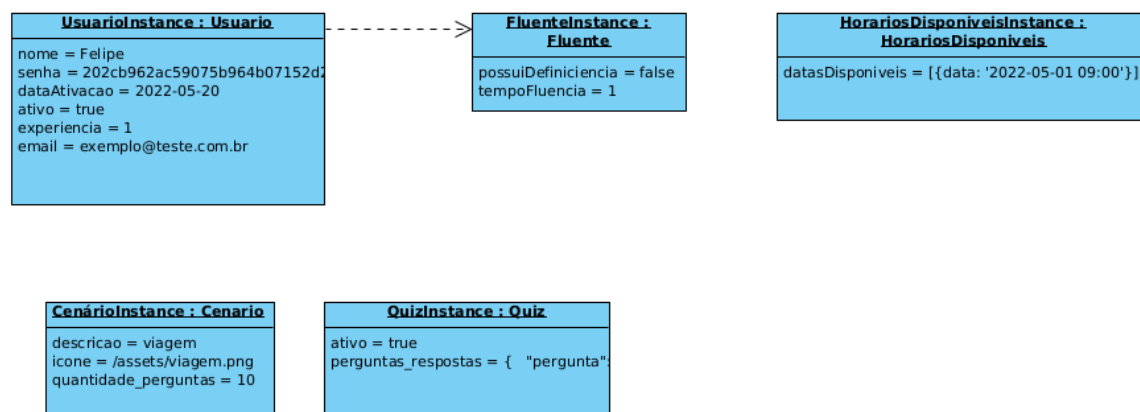
4.3 - Diagrama de classe

Para este diagrama foram necessários um total de 6 classes inicialmente, basicamente separando os tipos de usuário, fluente e administrador no qual as classes de fluente e administrador herdarão de usuário, por fim teremos duas classes responsáveis pelo controle do quiz e para persistência dos horários agendados teremos a classe de horários disponíveis.



4.4 - Diagrama de Objetos

Para o diagrama de objetos, foram implementados alguns exemplos para a equipe de desenvolvimento tenha uma base para desenvolver as funcionalidades e realizar algumas validações que não podem ser recebidas pelos atributos.



4.5 Conclusão

Utilizando as ferramentas da UML para realizar a modelagem teve como resultado um melhor acompanhamento de todos os atributos que eram necessários para o projeto. Com o diagrama de atividades pode-se observar o fluxo que cada tipo de usuário terá dentro do software e quais decisões ele pode tomar, já com o diagrama de classes, consegue-se ter uma noção para realizar o

desenvolvimento, agrega um valor muito alto com a utilização do paradigma orientado a objeto, com e por fim o diagrama de objetos temos noção de quais dados podem ser inseridos em cada tabela, um modelo.

Capítulo 5 - Desenvolvimento

5.1 - Introdução

Nesta etapa iremos descrever e apontar quais tecnologias abordadas, suas versões, será abordado o benefício que determinada tecnologia trouxe para o projeto, também está previsto trazer algumas imagens do software de algumas telas chaves do produto, destacadas no diagrama de atividades. Também será descrito a rotina de segurança, os cuidados principalmente dos usuários e como será feita sua rotina de backup, por fim será disponibilizado algumas tabelas de cronogramas tendo como ênfase o estágio de modelagem e entrega de funcionalidades.

5.2.Ferramentas Utilizadas

Quadro 2 - Ferramentas Utilizadas

5.5 - Cronogramas

Nesta etapa será disponibilizado o cronograma de duas etapas do projeto os cronogramas serão separados pelos meses e representados por números do calendário, por exemplo 07 para julho, o primeiro cronograma será composto por ações relacionadas a parte da documentação, desde a escrita a aplicações da metodologia, levantamento de requisitos e seus diagramas

5.5.1 - Referente às etapas de desenvolvimento da documentação

Processos	8	9	10	11	12	1	2	3
Metodologia	x							
Fundamentação Teórica	x							
Levantamento de Requisitos	x							
Diagrama de Classe de Uso	x							
Diagrama de Atividade		x	x	x	x	x	x	
Diagrama de Classes		x	x	x				
Diagrama de Objetos		x	x	x				
Possíveis Ajustes			x	x	x	x	x	

Quadro 3 - Cronograma da documentação

5.5.2 - Referente ao processo de desenvolvimento.

Funcionalidades	12	1	2	3
RF001 - Cadastrar Usuários - Implementação	x			

RF001 - Cadastrar Usuários - Testes	x			
RF002 - Enviar um link de confirmação do usuário - Implementação	x			
RF002 - Enviar um link de confirmação do usuário - Testes	x			
RF003 - Realizar login - Implementação	x			
RF003 - Realizar login - Testes	x			
RF004 - Cadastrar Quiz - Implementação		x		
RF004 - Cadastrar Quiz - Testes		x		
RF009 - Gerenciar Fluentes - Implementação		x		
RF009 - Gerenciar Fluentes - Testes		x		
RF005 - Gerenciar Quiz - Implementação		x		
RF005 - Gerenciar Quiz - Testes		x		
RF012 - Gerenciar Quiz Cadastrado - Implementação			x	
RF012 - Gerenciar Quiz Cadastrado - Testes			x	
RF008 - Responder perguntas e respostas - Implementação			x	
RF008 - Responder perguntas e respostas - Testes			x	
RF007 - Ganho de experiência do usuário - Implementação			x	
RF007 - Ganho de experiência do usuário - Testes			x	
RF013 Gerenciar agenda com os usuários - Implementação				x
RF013 Gerenciar agenda com os usuários - Testes				x
RF010 - Inativar fluentes da plataforma - Implementação				x
RF010 - Inativar fluentes da plataforma - Testes				x
RF005 - Painel para comunicação entre usuário e fluente - Implementação				x
RF005 - Painel para comunicação entre usuário e fluente - Testes				x
Deploy	x	x	x	x

Quadro 4 - Cronograma de Programação

Capítulo 6 - Conclusão

Nesta seção será descrito as contribuições que este trabalho para divulgação dessa língua, como ela contribuiu para o cenário, o que será realizado daqui para frente e as referências do projeto.

6.1 - Contribuições deste trabalho para o cenário.

6.2 - Considerações finais

6.3 - Trabalhos Futuros

REFERÊNCIAS

SBROCCO JOSÉ HENRIQUE DE CARVALHO; PAULO CESAR DE MACEDO. Metodologias ágeis: Engenharia de software sob medida. Editora Érica; 1ª edição (22 março 2012)

LOCATELLI, T. Libras: aspectos, desafios e possibilidades proporcionadas pela tecnologia. 17 de agosto de 2018, Disponível em: [https://www.](https://www.nucleodoconhecimento.com.br/educacao/libras-pela-tecnologia)

[nucleodoconhecimento.com.br/educacao/libras-pela-tecnologia](https://www.nucleodoconhecimento.com.br/educacao/libras-pela-tecnologia) ,

Acesso em: 29 set. 2022.

Quase 30 milhões de brasileiros sofrem de surdez. 19 de setembro de 2017, Disponível em: <https://jornal.usp.br/atualidades/quase-30-milhoes-de-brasileiros>

[-sofrem-de-surdez/#:~:text=Dados%20da%20Organiza%C3%A7%C3%A3o%20Mundial%20da%3E](#) Acesso em: 1 ago. 2022.

NEGRA, G. I. DA M. IBGE confirma: surdez não é sinônimo de Libras. 31 de agosto de 2021, Disponível em: <https://www.geledes.org.br/ibge-confirma-surdez-nao-e-sinonimo-de-libras>. Acesso em: 1 ago. 2022.

SIGNUM WEB, E. Estatísticas sobre os surdos no Brasil conheça e aprenda sobre esse tema. 2 de novembro de 2020, Disponível em: <https://blog.signumweb.com.br/curiosidades/surdos-no-brasil> , Acesso em: 2 ago. 2022.

RED HAT, O que é metodologia Ágil? 19 de julho de 2022, Disponível em: <https://www.redhat.com/pt-br/devops/what-is-agile-methodology#como-ela-surgiu> Acesso em: 1 set. 2022.

MANIFESTO AGIL, Manifesto para Desenvolvimento Ágil de Software. 2001 Disponível em: <http://agilemanifesto.org/iso/ptbr/manifesto.html> Acessado em: 1 set. 2022.

MARTIN FOWLER, Microservices Guide. 21 de agosto de 2019, Disponível em: <https://www.martinfowler.com/microservices> Acessado em: 2 set. 2022.

TRYBE. UML: o que é, para que serve e quando usar essa linguagem de notação? 25 de agosto de 2022, Disponível em: <https://blog.betrybe.com/tecnologia/uml>. Acessado em: 2 set. 2022.

PLANALTO, Lei Federal N. 10436 de 24 de abril de 2002: Oficializa a Língua Brasileira de Sinais em território nacional. Brasília 2002. Disponível em: http://www.planalto.gov.br/ccivil_03/LEIS/2002/L10436.htm Acessado em: 19 de setembro de 2022.

MANIFESTO AGIL, Manifesto para Desenvolvimento Ágil de Software.

2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html> Acesso em: 19 de setembro de 2022.

ATÉ O MOMENTO, O que é UML - o que é, para que serve, quando usar, e muito mais! 31 de janeiro de 2019, Disponível em: <https://www.ateomomento.com.br/diagramas-uml> Acesso em: 12 de outubro de 2022.

ASSEMBLEIA DE SÃO PAULO, Dia Internacional da Linguagem de Sinais procura promover a inclusão de pessoas surdas, de 23 de Setembro de 2021. Disponível em: <https://www.al.sp.gov.br/noticia/?23/09/2021/dia-internacional-da-linguagem-de-sinais-procura-promover-a-inclusao-de-pessoas-surdas> Acesso em: 20 de out. de 2022.

AGENCIA NACIONAL, Um em cada quatro idosos tinha algum tipo de deficiência em 2019, 26 de outubro de 2021. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/31447-um-em-cada-quatro-idosos-tinha-algum-tipo-de-deficiencia-em-2019>, Acesso em: 20 de out. de 2022.

PSICOPEDAGOGIA CRÍTICA, LIBRAS: principais dificuldades no uso da língua de sinais, 25 de julho de 2016. Disponível em: <https://psicopedagogiacritica.wordpress.com/2016/07/25/libras-as-principais-dificuldades-no-uso-da-lingua-de-sinais>. Acesso em: 20 out. 2022.

CÂMARA DOS DEPUTADOS, Especialistas pedem mais intérpretes de Libras em universidades; MEC aponta dificuldades - Notícias. Disponível em: <https://www.camara.leg.br/noticias/577880-especialistas-pedem-mais-interpretes-de-libras-em-universidades-mec-aponta-dificuldades>. Acesso em: 20 out. 2022.

ACADEMIA DE LIBRAS, Dificuldades e Desafios dos Surdos na Sociedade (Brasil), 23 de dezembro de 2019. Disponível em: <https://academiadelibras.com/blog/dificuldades-dos-surdos-na-sociedade>

CODIGO INF, A falta do ensino de Libras nas escolas preocupa estudantes e intérpretes, 21 maio 2021. Disponível em: <http://codigo.inf.br/aun/educacao/a-falta-do-ensino-de-libras-nas-escolas-preocupa-necessidad-os-e-interpretes/> Acesso em: 21 out. 2022

DEVMEDIA, Introdução ao FDD - Feature Driven Development. Disponível em: <https://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/27971> Acesso em: 21 out. 2022

CEDRO TECH, Feature Driven Development (FDD), 12 de abril de 2019, Disponível em: <https://blog.cedrotech.com/feature-driven-development-fdd>. Acesso em: 22 out. 2022.

DEVMEDIA, Técnicas para levantamento de Requisitos. Disponível em: <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151> Acesso em: 22 out. 2022;

MENTORING, Entrevista: como método de levantamento de requisitos. 13 de junho de 2016, Disponível em: <https://projectmentoring.wordpress.com/2016/06/13/entrevista-como-metodo-de-levantamento-de-requisitos>. Acesso em: 22 out. 2022.

ALURA, POO: o que é programação orientada a objetos?, 11 de maio de 2022, Disponível em: <https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos> Acesso em: 22 out. 2022

DEVMEDIA, POO: Os 4 pilares da Programação Orientada a Objetos. Disponível em: <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>.

PROJETO DIÁRIO, FELIZARDO, J. H. L. DA F. História da UML. 25 de julho de 2013, Disponível em: <https://www.projetodiario.net.br/historia-da-uml> Acesso em: 22 out. 2022.