



ETEC JOÃO GOMES DE ARAUJO

Curso Técnico em Informática para internet

Felipe Antunes Fernandes De Oliveira

Guydo Ventura Gonçalves

Rafaela Kaori Eiti

**UPLINKERCODE: Desenvolvimento de detector e tradutor de
linguagem de programação**

Pindamonhangaba – SP

2023

CEETEPS - CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA
ETEC JOÃO GOMES DE ARAUJO
CURSO TÉCNICO EM INFORMÁTICA PARA INTERNET

Felipe Antunes Fernandes De Oliveira
Guydo Ventura Gonçalves
Rafaela Kaori Eiti

**UPLINKERCODE: Desenvolvimento de detector e tradutor de
linguagem de programação**

Trabalho de Conclusão de Curso apresentado
ao CEETEPS – Centro Estadual de Educação
Tecnologia Paula Souza – Etec João Gomes de
Araújo como requisito parcial para a certificação
do Curso Técnico em Informática para internet.

Profª Orientadora: Isaura Maria dos Santos
Fortes

Pindamonhangaba – SP

2023

Felipe Antunes Fernandes De Oliveira

Guydo Ventura Gonçalves

Rafaela Kaori Eiti

**UPLINKERCODE: Desenvolvimento de detector e tradutor de
linguagem de programação**

ETEC JOÃO GOMES DE ARAUJO

CURSO TÉCNICO EM INFORMÁTICA PARA INTERNET

Data: _____

Resultado: _____

COMISSÃO AVALIADORA:

Prof. _____

Assinatura _____

Prof. _____

Assinatura _____

Prof. _____

Assinatura _____

Dedicamos este trabalho a todos que nos apoiaram nessa jornada na ETEC JGA e por ter feito possível a realização do desenvolvimento do software UpLinkerCode. Também aos profissionais da área de tecnologia e estudantes.

Agradecemos as nossas famílias e amigos que deram suporte nesse tempo até o dia de hoje, aos colegas do curso que foi de extrema importância para a produção do trabalho de conclusão.

“A lógica é a anatomia do pensamento.”

Henry Goodwin

RESUMO

A tecnologia da informação está cada vez mais inserida no mundo atual e assim cada vez mais cresce a demanda de profissionais da área para a melhoria contínua e haver cada vez mais avanço. O objetivo do trabalho realizado é de analisar a situação do mercado de tecnologia brasileiro, as dificuldades que estão presentes para encontrar bons desenvolvedores e para a formação das futuras gerações, análise da viabilidade da criação de um software para traduzir a linguagem de programação utilizada em um código para uma segunda linguagem. Foram realizadas pesquisas com estudantes e profissionais da área e foi possível concluir que é um mercado aquecido atualmente e que só tende a crescer cada vez mais, por isso há espaço para o *software* ser utilizado por estudantes e/ou desenvolvedores.

Palavras-Chave: *software*, tecnologia da informação, lógica de programação.

Índice de Imagens

Figura 1: Reportagem sobre a falta de profissionais de T.I no mercado brasileiro.....	14
Figura 2: Reportagem sobre a falta de profissionais de T.I. no mercado brasileiro.....	15
Figura 3: Logotipo do software de design Figma.	15
Figura 4: Logotipo do Javascript.....	18
Figura 5: Logotipo do HTML	18
Figura 6: Logotipo do CSS.....	19
Figura 7: Logotipo do CSS.....	19
Figura 8: Logotipo do SGBD MySQL.....	21
Figura 9: Faixa etária dos participantes da pesquisa online.	25
Figura 10: A viabilidade de utilização do software pelos estudantes.	25
Figura 11: A dificuldade no aprendizado da lógica de programação	25
Figura 12: Gênero dos participantes da pesquisa online.....	26
Figura 13: Design system da paleta de cores da UpLinkerCode	27
Figura 14: Design system das fontes e tamanho da UpLinkerCode	28
Figura 15: Design system das fontes e tamanho da UpLinkerCode	28
Figura 16: Design system das fontes e tamanho da UpLinkerCode	29
Figura 17: Design system dos botões da UpLinkerCode	29
Figura 18: Design system dos botões da UpLinkerCode	30
Figura 19: Design system dos botões da UpLinkerCode.....	30
Figura 20: Design system dos botões da UpLinkerCode.....	31
Figura 21: Wireframe da landing page modelo 1	31
Figura 22: Wireframe da landing page modelo 2	32
Figura 23: Wireframe da página principal modelo 1	32
Figura 24: Wireframe da página principal modelo 2	33
Figura 25: Protótipo de alta fidelidade do modelo 1	34
Figura 26: Front-end da página inicial.....	35
Figura 27: Front-end da página inicial.....	35
Figura 28: Front-end página da tradução.....	36

Figura 29: API do ChatGPT	36
Figura 30: Protótipo e login.....	38
Figura 31:Código do menu de login.....	38
Figura 32: Bibliotecas e código para conexão com o Banco de dados.....	39
Figura 33: Função cadastrar.....	40
Figura 34: Função login	41
Figura 35: Cadastro dos usuários mostrando o salt utilizado e a senha_hashed gerada.....	41
Figura 36: Dados salvos no Banco de dados	42
Figura 37: Login mostrando a comparação entre a senha salva na base de dados (senha_hashed) e a recém criptografada (senha_hashed_input)	42
Figura 38: Diagrama Entidade Relacionamento	43
Figura 39: Ambiente MySQL Workbench com comando para criar o banco de dados	44
Figura 40: Criação tabelas “clientes” e “planos”	44
Figura 41: Inserindo clientes diretamente no banco de dados.....	45
Figura 42: Inserindo clientes diretamente no banco de dados.....	45
Figura 43: Logotipo da ULC.....	46
Figura 44: Canvas inicial da UpLinkerCode.....	46

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Objetivos	9
1.1.1 Objetivo Geral	9
1.1.2 Objetivos específicos.....	10
1.2 Justificativa.....	10
2. REVISÃO DA LITERATURA	12
2.1. As dificuldades no aprendizado em lógica de programação	13
2.2. As novas gerações x Tecnologia da Informação	14
2.3. Tecnologias utilizadas no desenvolvimento do software da ULC	15
2.3.1. Figma.....	15
2.3.3. Canvas.....	17
2.4 TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO DO CÓDIGO ..	17
2.4.1 <i>Front-end</i>	17
2.4.2 – Banco de dados.....	19
2.1.3 <i>Back end</i>	22
3. METODOLOGIA	23
4 ANÁLISE E DISCUSSÃO DE RESULTADOS	26
4.1 Desenvolvimento do software de tradução da ULC.....	27
4.1.1 Idealização do software	27
4.1.2 Desenvolvimento do protótipo no Figma.....	27
4.1.3 <i>Wireframe</i> desenvolvido pela ULC.....	31
4.1.4 Protótipo desenvolvido para o Software.....	33
4.2 Desenvolvimento da página	34
4.2.1 Desenvolvimento <i>front-end</i>	34
4.2.2 Desenvolvimento <i>back-end</i>	36
4.2.3 Desenvolvimento banco de dados.....	42
4.3 Viabilidade da UPLINKERCODE.....	45

4.3.1 Logotipo da uplinkercode.....	45
4.3.2 Canvas.....	46
5. CONCLUSÃO	49
5.1 Considerações Finais	49
5.2 Sugestão para trabalhos futuros	49
REFERÊNCIAS.....	50
APÊNDICES	54
Apêndice A – Questionário.....	54

1 INTRODUÇÃO

A ampla aplicabilidade da tecnologia nos diversos setores da economia levou ao surgimento de uma infinidade de linguagens de programação, com isso, as empresas se veem obrigadas a investir em cursos para os seus desenvolvedores e/ou contratar novos desenvolvedores para realizar projetos específicos. Essas soluções impactam diretamente no custo e tempo de produção, além de afetar a qualidade do *software*.

O *software* desenvolvido no trabalho, o UpLinkerCode (ULC), nasceu com o intuito de auxiliar os programadores a desenvolver códigos em outras linguagens diferentes de maneira fácil para otimizar o tempo e custo do profissional, com o objetivo principal de ajudar no aprendizado de quem está iniciando os estudos em programação e auxiliar quem está aprendendo uma linguagem nova.

O objetivo do *software* da UpLinkerCode, é ser mais uma ferramenta para atrair os jovens para a carreira de tecnologia, democratizando o acesso e facilitando o entendimento das linguagens, não deixando ser algo desinteressante e aparentemente difícil, assim podendo distanciar cada vez mais novos talentos para a área. Na entrevista com o professor, o FERRO (2019), cita “Ele conta que os jovens não veem a TI como opção de trabalho, porque acreditam que a profissão seja muito complexa e, ao mesmo tempo, desinteressante”.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo é auxiliar nos trabalhos que envolvem a manipulação de documentos e a escrita de códigos de programação, fazendo com que a resposta dos diversos arquivos conectados e selecionados pelo usuário seja mais rápida e eficaz, além de auxiliar na compreensão de diversas linguagens e na possibilidade de fazer a transição de diversos tipos que sejam mais agradáveis ao usuário.

1.1.2 Objetivos específicos

- ✓ Pesquisar e compreender a aplicabilidade de sistemas de identificação e tradução de linguagens de programação, assim como o atual processo de aprendizagem de programação para novatos;
- ✓ Pesquisar e aplicar os sistemas citados em múltiplas linguagens por um só sistema;
- ✓ Estudar as ferramentas e bibliotecas necessárias para o desenvolvimento do sistema;
- ✓ Definir os custos do desenvolvimento;
- ✓ Desenvolver o sistema utilizando Python;
- ✓ Implementar o sistema de acordo com as capacidades previstas;
- ✓ Analisar a qualidade de tradução e a funcionalidade dos códigos criados;
- ✓ Testar e concluir a execução do sistema; e
- ✓ Documentar o processo de criação e os resultados obtidos.

1.2 Justificativa

A recente e repentina mudança nas práticas de trabalho e estudo, estimulada pelo aprimoramento tecnológico e a pandemia da Covid-19, causou uma aproximação considerável da maioria dos profissionais e estudantes à tecnologia.

“Situações extraordinárias como a que foi causada pela pandemia do coronavírus exigiram agilidade para adaptação e aceleraram a necessidade de muitas empresas de dar início à transformação digital.”
MUNIZ, et al (2022, pág. 7)

Meios online de acessar conteúdos e trabalho começaram a ser muito utilizados, em uma maneira de contornar as dificuldades econômicas causadas pelo evento.

Mesmo após essa situação, muitas das pessoas acostumaram-se com essas práticas, tanto os cidadãos quanto as empresas começaram a adotar meios tecnológicos de realizar suas tarefas no cotidiano. Perante a essa mudança, que afetou diversas áreas econômicas e sociais, a tecnologia e a internet se tornaram ainda mais entrelaçadas à vida humana.

Com essa junção, várias novas ferramentas foram criadas, no mercado da programação, para acelerar a produção de projetos, e a capacidade de trabalhar a distância sem perda de produtividade atraiu mais pessoas para a área, e por isso, mais funcionários tiveram de ser treinados, e mais interessados procuraram pelo estudo, que sem guias ou cursos, é difícil de se seguir. Enquanto isso, com o uso de documentos de texto no trabalho ou estudo, há uma certa lentidão na execução de certas tarefas, principalmente no que se diz salvar arquivos e compartilhá-los, o que possibilita problemas como perda de arquivos, falhas de comunicação e desorganização.

As soluções da UpLinkerCode impactam diretamente no custo e tempo de produção, além de afetar a qualidade do software. Já o processo de atualização de arquivos com proteção facilita a velocidade de alteração de arquivos, agilizando as tarefas e trazendo segurança ao trabalho.

2. REVISÃO DA LITERATURA

A tecnologia da informação está ganhando cada vez mais espaço no mercado de trabalho brasileiro e com mais ênfase após a pandemia de COVID-19, como explica MUNIZ, et al (2022, pág. 7) “Situações extraordinárias como a que foi causada pela pandemia do coronavírus exigiram agilidade para adaptação e aceleraram a necessidade de muitas empresas de dar início à transformação digital.”, houve um aumento da demanda de profissionais da área de tecnologia atualmente.

Acompanhando a corrida pelos profissionais de tecnologia, existe um dado preocupante que é a falta de profissionais no mercado para suprir a necessidade do mercado que está cada vez mais tecnológico e inovador. Essa escassez está estagnando o crescimento da tecnologia no Brasil.

A falta de mão de obra qualificada no setor de Tecnologia da Informação (TI) tem gerado preocupação em entidades que representam a classe. Em situações díspares, a falta de mão de obra pode impactar o crescimento do setor em breve..., a repercussão do impacto da escassez de mão de obra em TI poderá ocasionar inclusive em uma perda de receita de R\$ 115 bilhões até 2020 no segmento, em todo o país. (CARDOSO, pg. 697)

Outro fator é a necessidade de o desenvolvedor ter raciocínio lógico e matemático bom para solucionar os problemas encontrados nos códigos, ao mesmo tempo que os estudantes brasileiros não têm uma base sólida em exatas na educação básica. Segundo Santos, et al (2015): “Ao analisarmos as habilidades que representam cada nível verificamos, a partir dos dados apresentados que, de maneira geral, no 3º ano do EM os alunos não possuem habilidades essenciais aos olhos do Ensino da Matemática.”.

Essa defasagem é ainda mais marcada quando consideramos a ideia que a tecnologia e seu uso representam. A internet e os mais diversos meios tecnológicos ficaram marcados como ferramentas principalmente de lazer e muitas vezes, o uso desses meios para a programação não é considerado ou reconhecido. Portanto, essa mentalidade causa a falta de conhecimento geral da população e dos jovens estudantes sobre a programação.

O mundo educacional já não é o mesmo após a tecnologia estar presente em seu cotidiano, sua utilização no ambiente escolar é comum entre os

alunos, onde os mesmos sabem facilmente utilizar um aplicativo de celular, um programa no computador, mas ao mesmo tempo não sabem ou nunca pararam para pensar em como os mesmos funcionam, como eles são feitos, quem os criam.
(LOVATTI, pg. 114)

LOVATTI, et al (2017) ainda cita que “a programação proporcionará a ampliação de ideias e o desenvolvimento raciocínio lógico ..., a programação está presente na vida dos alunos e assim faz-se necessário conhecer essa nova linguagem e recurso tecnológico.”. Portanto, é visível que a programação deveria ter um papel maior no ensino, e uma integração dela na vida dos estudantes deveria proporcionar um aumento dos profissionais requisitados na área e o desenvolvimento dos alunos.

Tomando como base os dados apresentados, foi desenvolvido o *software* de tradução de códigos: a UpLinkerCode, que tem o intuito de auxiliar os desenvolvedores iniciantes ou desenvolvedores que já estão atuando na área e que estejam estudando e/ou que necessita utilizar uma linguagem de programação nova em um projeto. Auxiliando na entrada de novos profissionais na tecnologia.

2.1. As dificuldades no aprendizado em lógica de programação

Uma parte de extrema importância para exercer a atividade de um programador é a lógica de programação, sendo a base para aprender as linguagens de programação e os *frameworks* necessários, como foi citado pela ALMEIDA (2008) “A lógica de programação tem como objetivo mostrar métodos para solucionar problemas, além de determinar a sequência lógica para o desenvolvimento de um programa”. As dificuldades para continuar com os estudos se iniciam quando não há uma base boa na parte lógica e matemática.

Este não é um cenário tão improvável de se imaginar, quando o ensino básico brasileiro tem dificuldade em formar alunos com uma boa base de matemática dos ensinos médios, assim acaba atrapalhando e/ou dificultando no progresso desses alunos nos estudos.

Exemplos críticos no desempenho em matemática é o resultado da avaliação obtida pelo desempenho dos estudantes brasileiros divulgados pelo Ministério da Educação (MEC). O desempenho dos alunos do Ensino

Médio em Matemática piorou em duas avaliações, cujos resultados foram divulgados no Sistema Nacional de Avaliação da Educação Básica (SAEB). (SANTOS, et al, p. 18)

Deste modo, acaba distanciando cada vez mais as pessoas da tecnologia, por encontrar sérias dificuldades no início e com os “gurus” da tecnologia que querem passar fórmulas mágicas para entrar no mundo da tecnologia da informação de um jeito “fácil” e rápido.

2.2. As novas gerações x Tecnologia da Informação

Além dos problemas com o aprendizado em lógica de programação, outro grande desafio das empresas de tecnologia para o futuro é gerar interesse nas novas gerações para as áreas *tech*. Segundo FERRO (2021), consultando um profissional da área, atrair os jovens é uma tarefa difícil e que vai demandar estratégias para suprir a falta de profissionais que tende a piorar com o passar dos anos.

Ao realizar uma rápida busca sobre o cenário atual do mercado de tecnologia, mostra a preocupação sobre a alta demanda de profissionais de programação e a falta no mercado, como mostra o FERRO (2019) “exigida pelo processo de transformação digital que vivemos e essa defasagem pode piorar, já que a expectativa é de que, até 2024, o setor abra mais de 400 mil novas vagas.”.

Figura 1: Reportagem sobre a falta de profissionais de T.I no mercado brasileiro.



Área de TI deve gerar quase 420 mil vagas até 2025, mas faltam profissionais

Segundo a associação das empresas deste setor, existe uma demanda média anual de 159 mil profissionais no país, mas o Brasil só forma 53 mil...

13 de mar. de 2023



Fonte: g1.globo.com, 2023

Figura 2: Reportagem sobre a falta de profissionais de T.I. no mercado brasileiro.

 Canaltech

Vagas de emprego para serviços de TI crescem quase 80% em 10 anos no Brasil

Um levantamento realizado pela Associação das Empresas Brasileiras de Tecnologia da Informação do Paraná (Assespro-PR) revelou um...

2 de fev. de 2023



Fonte: Canaltech.com, 2023

O problema já estava sendo visado desde o início do milênio, mas o estopim do drama, foi a pandemia de COVID-19 que acelerou em alguns anos a transformação digital nas nossas vidas, o professor entrevistado pelo FERRO (2019), acrescenta que: “A pandemia acelerou o processo de automatização de serviços, mas a disponibilidade de profissionais não segue o mesmo rumo, o que, para o professor Rodello, exige que o País enfrente o “desafio de atrair os jovens para a profissão”.

2.3. Tecnologias utilizadas no desenvolvimento do software da ULC

2.3.1. Figma

O protótipo, foi desenvolvido pelo *software* de *design* de sites e aplicativos: Figma.

Contendo o *design system* do site, o protótipo navegável e dois modelos para escolha, como exemplo: página na versão claro ou escuro.

Figura 3: Logotipo do software de design Figma.



Fonte: Figma.com, 2023

2.3.1.1 *Design system*

Segundo a MCDONALD (2019), o *design patterns* é utilizado para melhorar o *design* do site ou aplicativo para o usuário utilizar de maneira fácil e prendendo a atenção dele, com padrão de cores, fontes, o design limpo e bonito aos olhos.

“UI patterns (user interface patterns) are found in the digital sphere of web sites, applications, native mobile apps and other software or devices. They suggest function, interaction and intent. UI patterns document reusable parts of an interface that share a purpose.” (MCDONALD, 2019)

A importância de criar um bom *design system*, é a padronização da identidade visual, de linguagem e outras características marcantes para a marca, assim auxiliando para o cliente lembrar da marca com algum estímulo.

2.3.1.2 *Wireframe*

O *Wireframe* segundo a CAZIBA, 2018 é utilizado para dar referência para o desenvolvedor *Front-end* com o tamanho dos objetos utilizados, em como deve ser estruturado de acordo com alguns estudos realizados, como os locais mais quentes do site, os locais mais fáceis de acessar e entre outros. Junto com os detalhes de design que analisa as cores e na decoração para prender a atenção do usuário.

*“Wireframes are usually referred to as a visual guideline that explains our digital product structure, such as its hierarchy and its key elements. Usually, wireframes are mostly used when we are creating a web or mobile application...
During the process of creating wireframes, we also start connecting the dots, or in other case the screens, by creating the product flow”* (CANZIBA, p. 58)

Uma característica que diferencia o *wireframe* do *mockup*, é de não ser de alta fidelidade, sendo riscado no papel na maioria das vezes.

2.3.1.3. *Mockup*

Ao realizar o *Wireframe* inicial, foi desenvolvido o protótipo para mostrar como seria desenvolvido o design do *software* com alto grau de fidelidade, como foi

comentado por PINHEIRO (2016), "Protótipo é uma simulação do produto final. É como uma maquete interativa que pode ter qualquer grau de fidelidade. O principal objetivo da construção de protótipos é testar se o fluxo do produto é agradável e coerente".

2.3.3. Canvas

A ferramenta Canvas, é utilizada para dar início e colocar no papel a ideia de um novo negócio ou já existente para analisar a viabilidade, o investimento necessário para dar andamento ao negócio e ao investimento inicial, realizando adaptações periodicamente de acordo com a realidade vivida pelo negócio atualmente.

"Has become the de facto diagnostic tool for understanding the value creation potential of businesses, our goal was to develop a similar construct that is easy to use and emphasizes the special characteristics of platform ecosystems"(SORRI, 2019)

2.4 TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO DO CÓDIGO

Para o desenvolvimento do Software foram utilizadas algumas tecnologias na parte de banco de dados, *front-end* e *back-end*.

2.4.1 Front-end

No desenvolvimento do *front-end* do *software*, foram utilizadas a linguagem Javascript e tecnologias como HTML 5, CSS 3.

2.4.1.1 Javascript

O Javascript é considerado uma linguagem de programação, por ser aplicada a lógica de programação no seu uso, como diz o FLANAGAN (2013) "Javascript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais".

Figura 4: Logotipo do Javascript



Fonte: logospng.org, 2023

É utilizado tanto na parte de design (*front-end*), quanto na parte de trás do site (*back-end*), dependendo da forma em que for utilizá-la. No *front-end*, possui a característica de deixar o site mais dinâmico e interativo, utilizando o ECMA Script.

2.4.1.2 HTML 5

Segundo SILVA (2015) “Quando a HTML foi inventada, os conteúdos eram essencialmente hipertextos. Assim hoje a HTML, é uma linguagem para marcação de conteúdos web em geral”, ela acaba não sendo considerada uma linguagem de programação por não utilizar lógica no seu código, mas servindo como base para inserir as linguagens.

Figura 5: Logotipo do HTML



Fonte: w3.org, 2023

2.4.1.3 CSS 3

O CSS3 acompanha sempre a HTML, para realizar a estilização das marcações aos conteúdos do site, podendo inserir imagens, planos de fundo e outras formatações para caracterizar o site de acordo com a marca e/ou o objetivo proposto. Como mostra o SILVA (2015) “O CSS é a abreviação para o termo *Cascading Style Sheet*, traduzido para o português como folha de estilo em forma de cascata.”.

Figura 6: Logotipo do CSS.



Fonte: logospng.org, 2023

2.4.2 – Banco de dados

O banco de dados foi desenvolvido utilizando o Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR) MySQL juntamente com a ferramenta visual de design e modelagem de banco de dados MySQL *Workbench*. Com essas tecnologias conseguimos criar, implementar e gerenciar todas as funcionalidades do banco de dados como: criação de tabelas, adicionar, deletar e pesquisar informações na base de dados. Esses softwares também foram utilizados para auxiliar no planejamento e organização do projeto ao permitirem a criação do Diagrama Entidade Relacionamento (DER).

Figura 7: Logotipo do CSS.



Fonte: mysql.com, 2023

2.4.2.1 – MySQL

Segundo Dyer (2015) o MySQL é SGBDR *open source* e *multithreaded* que se tornou popular devido a permissão para uso gratuito sobre as normas da *General Public License* (GPL), somado a suas diversas ferramentas, estabilidade, performance e confiabilidade.

“It has been estimated that there are more than six million installations of MySQL world wide, and reports of over 50,000 downloads a day of MySQL installations softwares. The sucesso of MySQL as a leading database is due not to its price – after all, Other cost-free and open source databases are available – but also its reliability, performance, and features” (DYER, 2015 p. 03).

2.4.2.2 – Sistema Gerenciador de Banco de Dados (SGBDR)

Para Gehrke e Ramakrishnan (2008, p. 07) o SGBD é *software* projetado para otimizar o armazenamento e gerenciamento dos dados em nosso projeto, “Armazenando-se dados em um SGBD em vez de em uma coleção de arquivos do sistema operacional, é possível utilizar os recursos do SGBD para gerenciar os dados de uma forma mais robusta e eficiente.

À medida que cresce o volume de dados e o número de usuários – centenas de *gigabytes* de dados e milhares de usuários são comuns nos bancos de dados corporativos atuais –, o suporte de um SGB torna-se indispensável”

O MySQL utiliza o modelo entidade-relacionamento (ER) que, de acordo com Ramakrishnan e Gehrke (2008), permite descrever os dados em termos de objetos e seus relacionamentos, criando uma estrutura baseada em linhas e colunas que formam uma tabela de informações relacionadas onde é possível definir a complexa relação entre as diversas tabelas de um sistema. Cada linha representa um objeto em que é passado um código de identificação (ID) único juntamente com outras propriedades como: nome do objeto e sua descrição.

Segundo Moes e Sheldon (2005) importante que o SGBDR permita manipular as informações contidas na base de dados ao mesmo tempo que as protege contra corrupção e inconsistência, mantendo os metadados necessários que caracterizam os dados armazenados.

Para os autores a maior diferença entre um SGBD e um SGBDR é que o último não apenas suporta a estruturação dos dados em forma de tabelas, mas também permite o relacionamento entre essas tabelas.

2.4.2.3 - Diagrama Entidade Relacionamento (DER)

Diagrama Entidade-Relacionamento (DER) é uma ferramenta de modelagem de dados que é amplamente utilizada em bancos de dados relacionais. De acordo com Elmasri e Navathe (2016), o DER é "uma técnica gráfica que retrata entidades, atributos e relacionamentos entre entidades". Ele é composto por três componentes principais: entidades, atributos e relacionamentos.

As entidades representam objetos do mundo real, como clientes, produtos e pedidos. Os atributos representam características dessas entidades, como nome, endereço e telefone para clientes. Os relacionamentos representam a forma como essas entidades estão relacionadas entre si, como um pedido que pertence a um cliente específico.

2.4.2.4 – MySQL *Workbench*

O MySQL *Workbench* é uma ferramenta gráfica de modelagem e gerenciamento de banco de dados MySQL. Segundo Fehily (2012, p. 405), "o *Workbench* é uma ferramenta de modelagem visual que permite criar, editar e gerenciar esquemas de banco de dados MySQL de forma gráfica".

De acordo com Soto (2019, p. 3), o ele oferece "uma interface gráfica que permite a criação e edição de tabelas, colunas, índices e restrições, além de permitir a execução de consultas SQL e a administração do banco de dados".

Figura 8: Logotipo do SGBD MySQL



Fonte: mysql.com, 2023

2.1.3 *Back end*

O *back-end* desenvolvido tem como base a linguagem de programação C e Portugal, para realizar a parte interna da funcionalidade do website.

2.1.3.1 – C

A linguagem de programação C, é utilizada para ler e trazer as informações solicitadas pelo cliente, segundo JUNIOR (2019), a linguagem C possui algumas características que o difere das outras como a sua estruturação sendo simples, linguagem de alto desempenho, considerado portátil por quem utiliza e a sintaxe dela é de alto nível.

Imagem 9: Logotipo da linguagem de programação C



Fonte: Microsoft Learn, 2023.

2.1.3.2 – Portugal

O Portugal não é conhecido como uma linguagem de programação como o caso da linguagem C citado anteriormente, sendo utilizado apenas para fins didáticos para quem está iniciando os estudos em programação, principalmente na parte de back end.

Segundo o MENDES, et al (2022), “Mesmo sendo uma pseudolinguagem, o Portugal também possui suas regras de escrita e que precisam ser seguidas para que os algoritmos funcionem corretamente.”, dessa forma auxiliando mais na formação do raciocínio lógico para desenvolver os códigos em alguma linguagem de programação futura.

Imagem 10: Logotipo da linguagem Portugol



Fonte: UNIVALI-LITE, 2023

2.1.3.3 - ChatGPT

O ChatGPT é uma ferramenta nova e útil utilizando a inteligência artificial para auxiliar no cotidiano ou no trabalho com apoio de banco de dados que foi alimentado com várias informações.

“O ChatGPT é um modelo de linguagem avançado desenvolvido pela OpenAI, é projetado para interagir com os usuários de maneira conversacional e pode ser usado para muitos propósitos e aplicações.” (LUNKE, 2023).

3. METODOLOGIA

No trabalho de conclusão de curso apresentado, foram realizadas análises de uma pesquisa qualitativa exploratória, com base em coleta de dados bibliográfica, documental e em pesquisas de campo com desenvolvedores e estudantes que estão iniciando na área de tecnologia sobre o espaço no mercado de tecnologia para a implantação do software UpLinkerCode nas escolas e/ou estudos individuais.

A área da tecnologia da informação está cada vez mais ocupando espaço no mercado de trabalho e pode-se notar por algumas pesquisas que mostram dados de 2022.

“Uma pesquisa da Associação das Empresas de Tecnologia da Informação e Comunicação e de Tecnologias Digitais (Brasscom), os empregos no setor de Tecnologia da Informação e Comunicação, que a associação chama de TIC, cresceram 14,4% no ano passado (2022).” DIAS, Maria (n.d.)

Apesar do aumento na quantidade de vagas e com previsões como a da Forbes (2022), que diz: “A migração digital acelerada pela pandemia criou oportunidades de

trabalho no setor de TI. Até 2025, haverá cerca de 800 mil novas vagas, segundo projeção da Brasscom”, mostra que a tendência é de continuar contratando mais profissionais do setor por estar em falta no mercado, não apenas nacional, mas também internacionalmente. A parte mais preocupante é sobre a falta de profissionais qualificados para as vagas abertas.

Segundo FERRO (2021, n.p.), “O setor de tecnologia da informação enfrenta sérios problemas para recrutar funcionários, ... foram 344% novos postos para profissionais plenos de TI e 173% para juniores no semestre passado”, então há vagas no mercado de trabalho, mas não estão sendo formado profissionais bons nas áreas existentes.

A equipe fez o uso de pesquisas bibliográficas com artigos, livros sobre desenvolvimento de software, e a utilização de um formulário para analisar sobre a aceitação do software da UpLinkerCode pelo público-alvo e mercado pesquisado em nossa amostra.

A pesquisa teve o foco em desenvolvedores e também em pessoas de outras áreas, mas com possível interesse em aprender programação, com 67 respostas registradas no formulário online e 08 na pesquisa de campo. A pesquisa online foi iniciada em 16 de março de 2023 e finalizou-se em 03 de abril de 2023. O formulário foi respondido por meio de abordagem de diversos públicos em redes sociais e na ETEC João Gomes de Araújo. A pesquisa de campo foi realizada em 17 de abril de 2023 a 21 de abril de 2023 com os alunos da ETEC João Gomes de Araújo e alunos da IFSP no campo de Campos do Jordão.

Nos formulários são questionados sobre as dificuldades enfrentadas pelos desenvolvedores no início da carreira ou para as pessoas que estão se inserindo atualmente no mercado, além da relevância que essas ou outras pessoas, de outras áreas, enxergam no futuro da programação. Os resultados estão disponíveis na seção de apêndice da pesquisa.

É esperado que ao analisar as respostas obtidas, seja possível viabilizar a ideia da implantação do software da UpLinkerCode no mercado e ajudando os desenvolvedores a otimizar os estudos.

Há vários estudos mostrando o medo dos profissionais sobre o futuro do mercado de tecnologia brasileiro com a falta de desenvolvedores, a probabilidade de atrasar o desenvolvimento da tecnologia nas terras tupiniquim, como foi explicado pelo

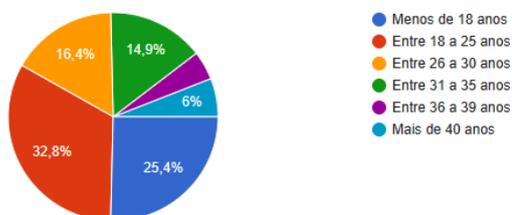
RTS (2009) “: a concepção linear da conversão do avanço científico em avanço tecnológico que, por sua vez, irá gerar desenvolvimento econômico e social”.

O software da ULC, foi desenvolvido para mudar o cenário catastrófico que pode vir a acontecer futuramente. Realizando as pesquisas de campo e online, pode-se analisar que a maior parte dos estudantes e/ou desenvolvedores, têm entre menos de 18 anos a 30 anos, que podem mudar a realidade do mercado se trabalhado corretamente.

Figura 9: Faixa etária dos participantes da pesquisa online.

Qual a sua faixa etária?:

67 respostas



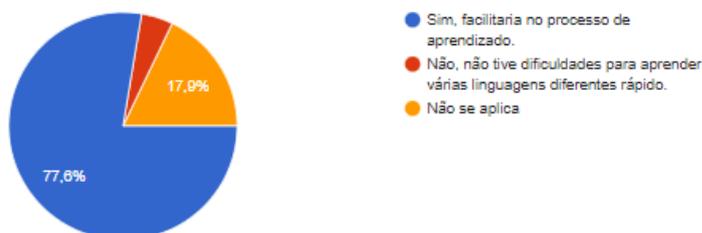
Fonte: Acervo próprio, 2023.

Figura 10: A viabilidade de utilização do software pelos estudantes.

Você acha que um software que traduz códigos de uma linguagem para outra de forma simultânea e disponibilizando documentações da linguagem, ia ter ajudado nos seus estudos inicialmente?

[Copiar](#)

67 respostas

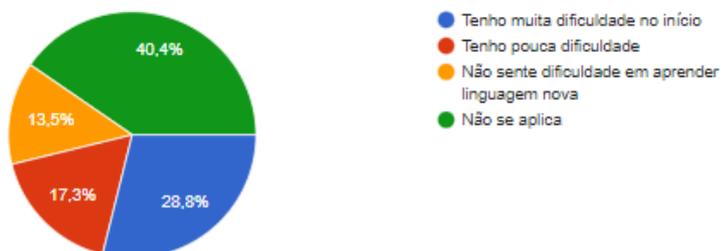


Fonte: Acervo próprio, 2023.

Figura 11: A dificuldade no aprendizado da lógica de programação

Se a resposta anterior for sim, você sente dificuldades para aprender uma linguagem de programação nova ou em lógica de programação?

52 respostas



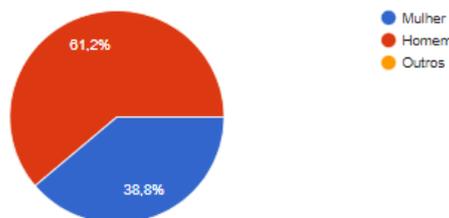
Fonte: Acervo próprio, 2023.

Um outro ponto importante é a distinção de gêneros até os dias de hoje na tecnologia, com a maior prevalência de homens na área e sem muita inclusão de mulheres, sendo um dado importante para poder aumentar a mão de obra na tecnologia da informação. A crença de haver rótulos para cada carreira ou área pode acabar afastando muitos talentos novos da área.

Figura 12: Gênero dos participantes da pesquisa online.

Qual o seu gênero?:

67 respostas



Fonte: Acervo próprio, 2023.

4 ANÁLISE E DISCUSSÃO DE RESULTADOS

Apresentação da forma em que foi idealizado o projeto pelos integrantes da ULC para o desenvolvimento do Software.

4.1 Desenvolvimento do software de tradução da ULC

4.1.1 Idealização do software

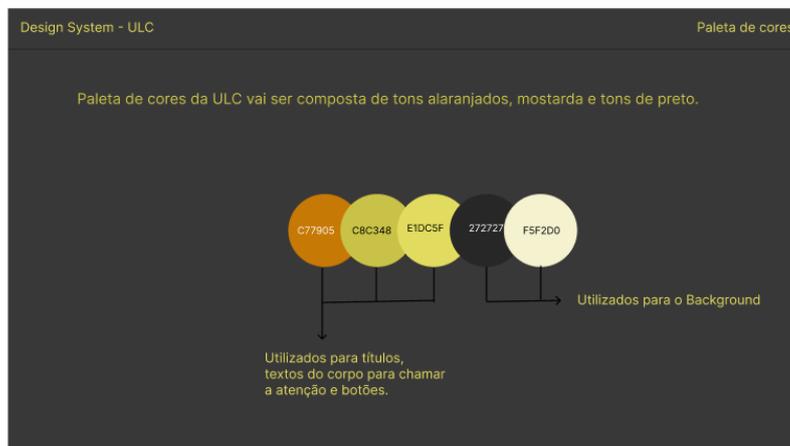
A ideia do software, surgiu quando os desenvolvedores do projeto recordaram as dificuldades que enfrentaram no início dos estudos com programação e foi analisado que outros estudantes também têm/tiveram dificuldades para estudar e aplicar a lógica de programação nos códigos.

A UpLinkerCode, nasceu para auxiliar quem está iniciando os estudos em programação deixando a experiência mais fácil e sem dificuldades.

4.1.2 Desenvolvimento do protótipo no Figma

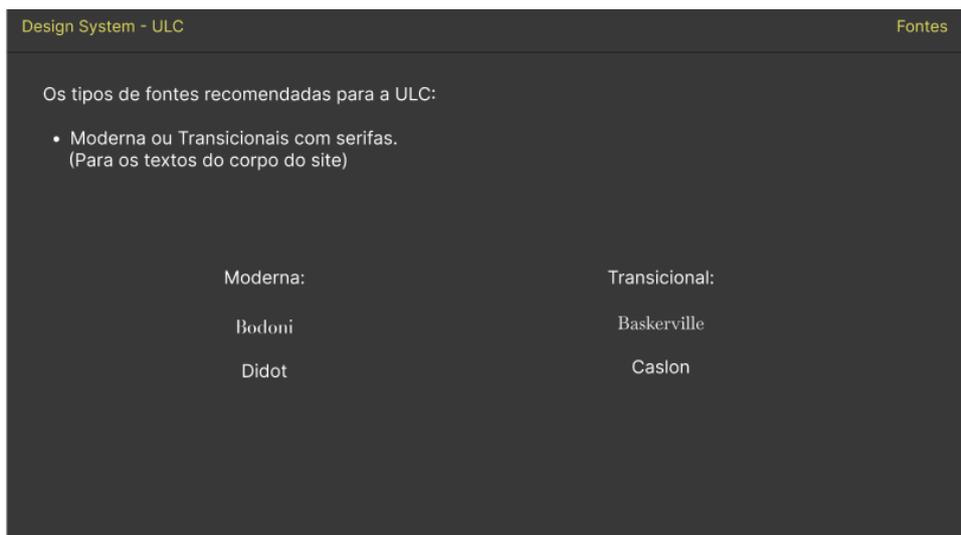
O *design system* da UpLinkerCode, foi baseado na paleta de cores variando entre as cores amarelo, laranja e preto para combinar com a identidade visual do negócio, as fontes utilizadas moderna e transicional, nos tamanhos 24 a 36 para títulos e de 16 a 24 para demais textos.

Figura 13: Design system da paleta de cores da UpLinkerCode



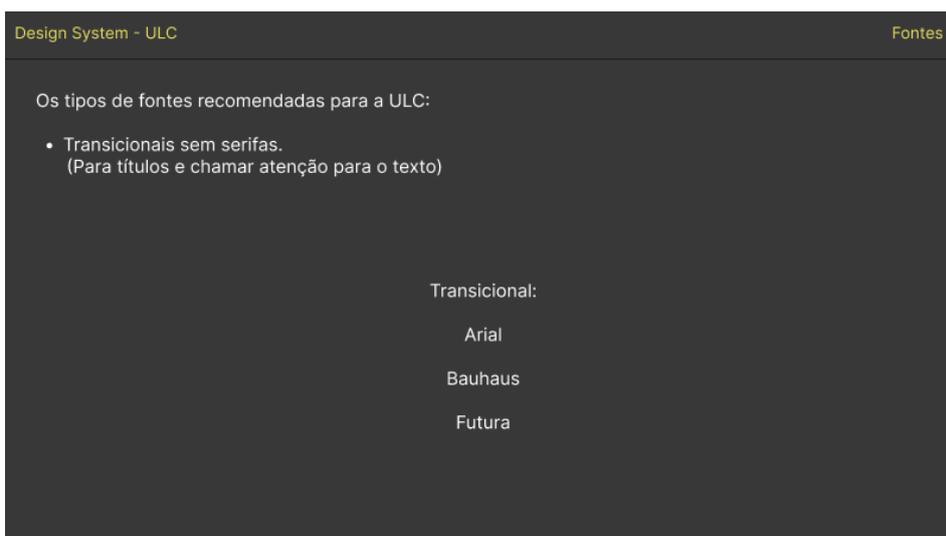
Fonte: Acervo próprio, 2023.

Figura 14: Design system das fontes e tamanho da UpLinkerCode



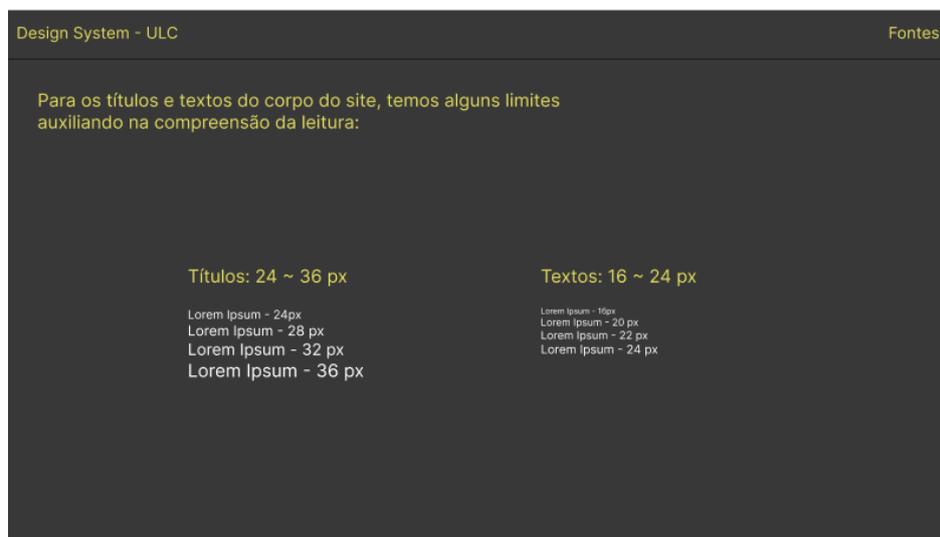
Fonte: Acervo próprio, 2023.

Figura 15: Design system das fontes e tamanho da UpLinkerCode



Fonte: Acervo próprio, 2023.

Figura 16: Design system das fontes e tamanho da UpLinkerCode



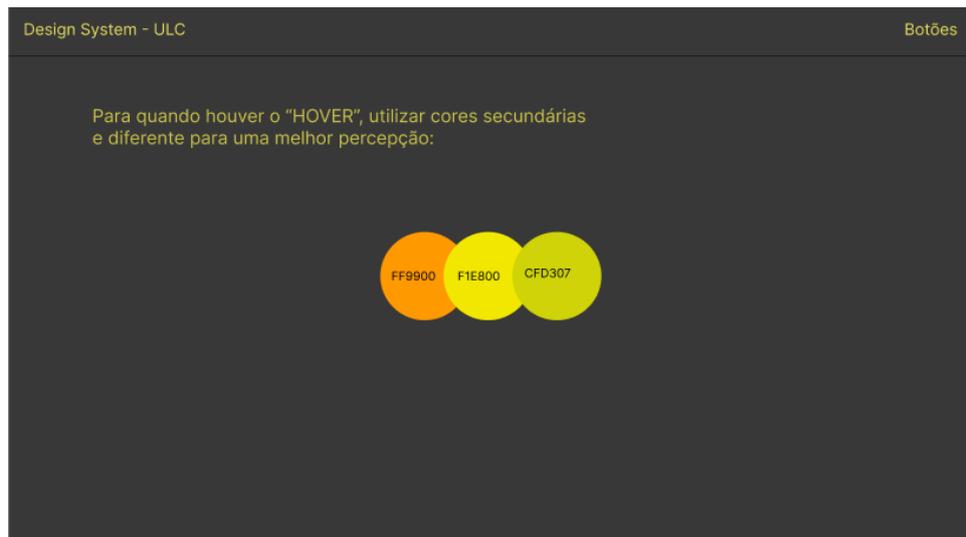
Fonte: Acervo próprio, 2023.

Figura 17: Design system dos botões da UpLinkerCode



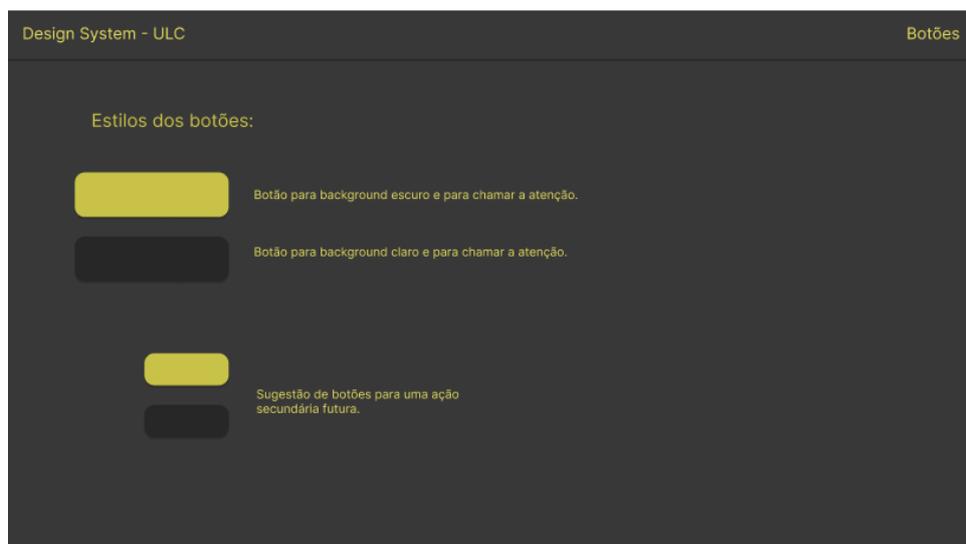
Fonte: Acervo próprio, 2023.

Figura 18: Design system dos botões da UpLinkerCode



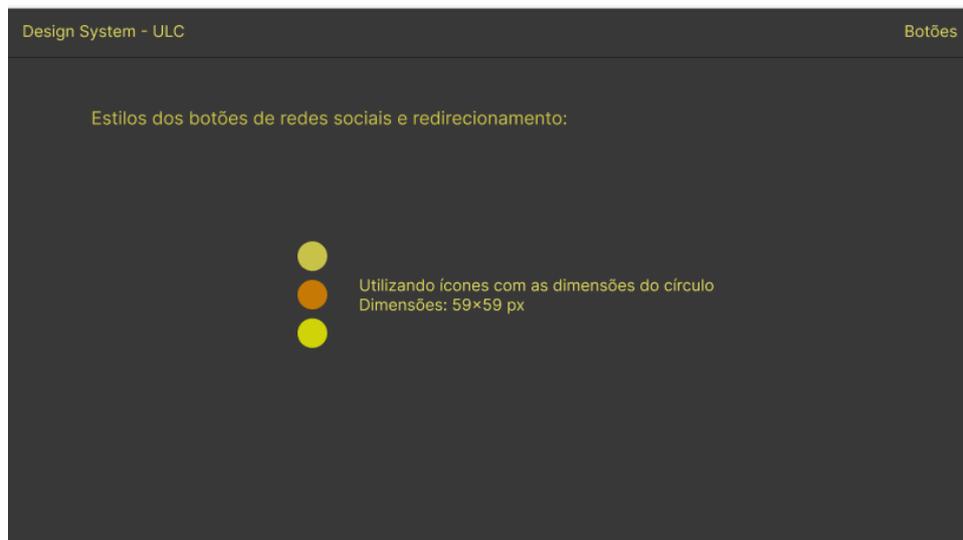
Fonte: Acervo próprio, 2023.

Figura 19: Design system dos botões da UpLinkerCode.



Fonte: Acervo próprio, 2023.

Figura 20: Design system dos botões da UpLinkerCode.

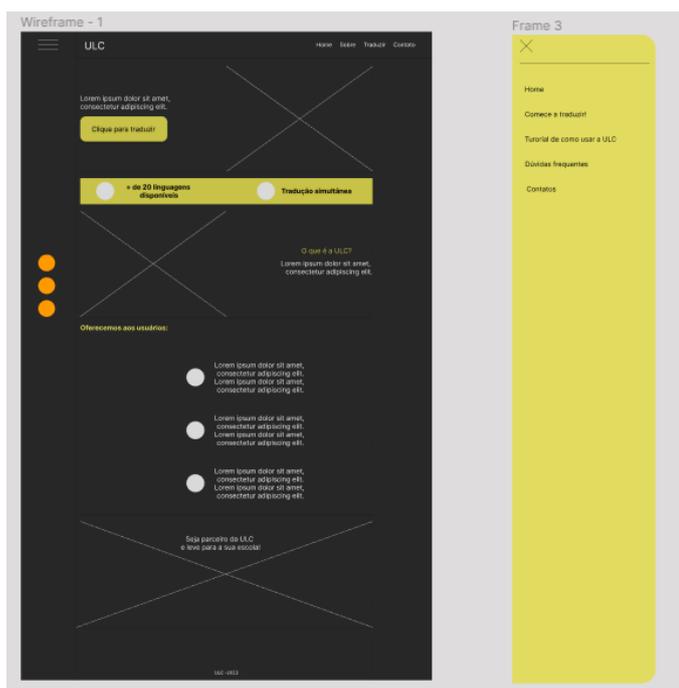


Fonte: Acervo próprio, 2023.

4.1.3 Wireframe desenvolvido pela ULC

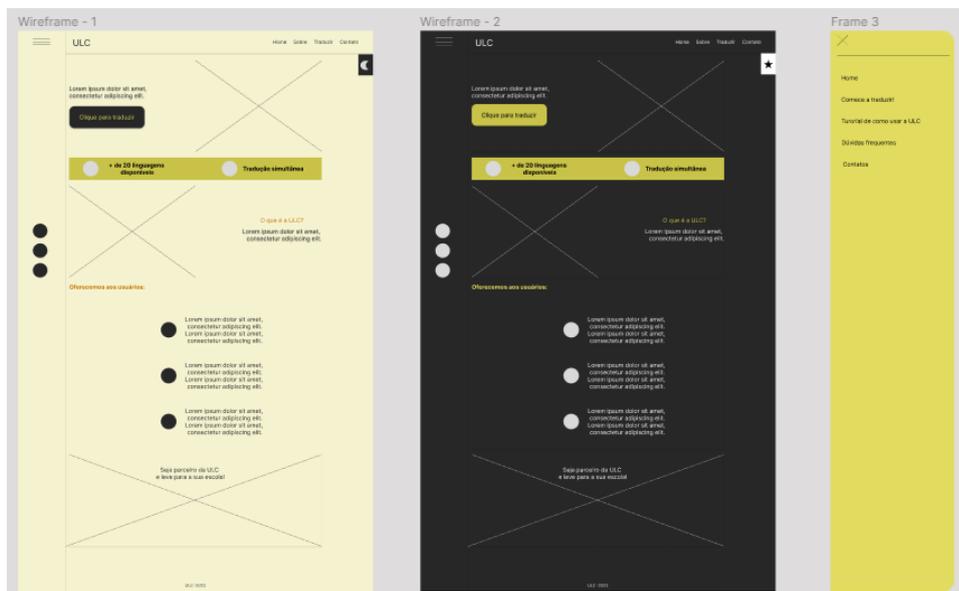
O *Wireframe* utilizado na ULC, foi idealizado pelos próprios criadores dando a identidade de todos no site, com dois modelos para ser realizado uma análise de qual utilizar no *mockup*.

Figura 21: Wireframe da landing page modelo 1



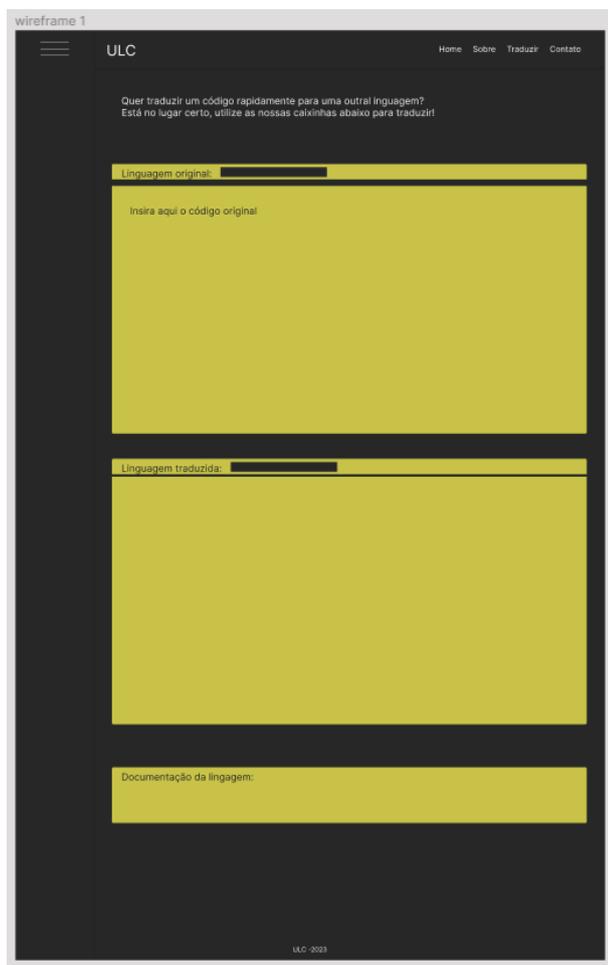
Fonte: Acervo próprio, 2023.

Figura 22: Wireframe da landing page modelo 2



Fonte: Acervo próprio, 2023.

Figura 23: Wireframe da página principal modelo 1



Fonte: Acervo próprio, 2023.

Figura 24: Wireframe da página principal modelo 2

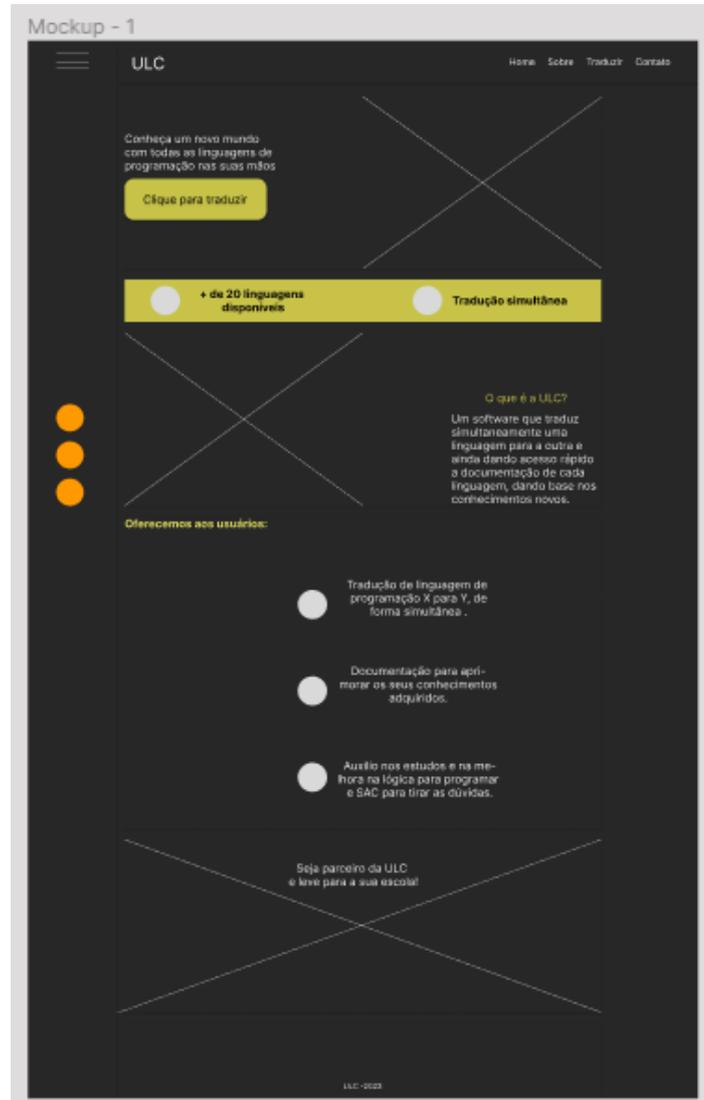


Fonte: Acervo próprio, 2023

4.1.4 Protótipo desenvolvido para o Software

Após uma análise dos *wireframes* apresentados, foi escolhido o modelo 1 por ter melhor *layout* e ser mais intuitivo para utilizar.

Figura 25: Protótipo de alta fidelidade do modelo 1



Fonte: Acervo próprio, 2023.

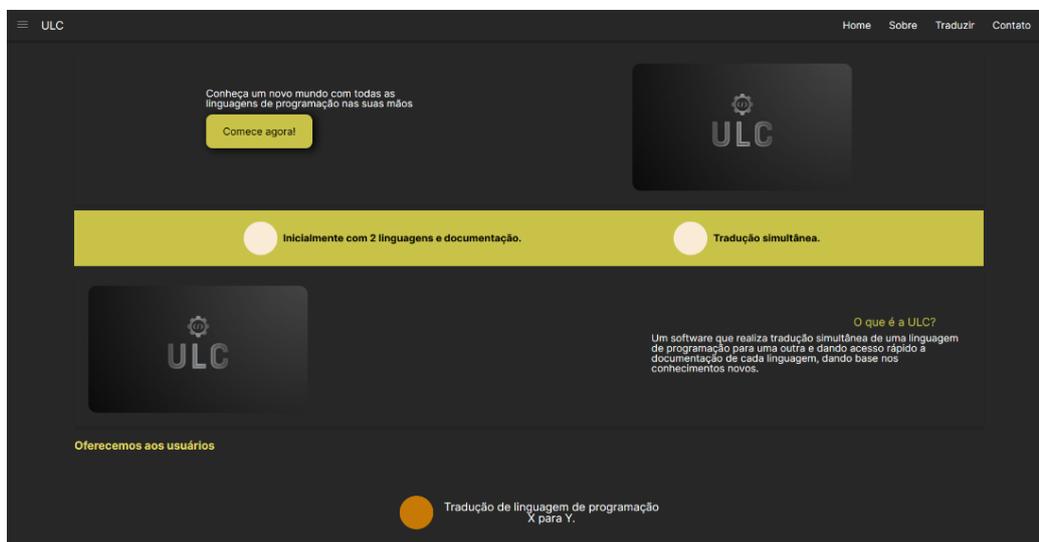
4.2 Desenvolvimento da página

4.2.1 Desenvolvimento *front-end*

O *front-end* foi baseado no design do *mockup*, utilizando o HTML 5, CSS 3 e JavaScript para deixar o site mais dinâmico e com responsividade. Primeiramente foi desenvolvido o HTML com os textos, as imagens que foram utilizadas e as separações de cada parte e assim, estilizando a página com o CSS 3 para organizar as imagens, os textos nas posições corretas e as cores da paleta desenvolvida no *design system*.

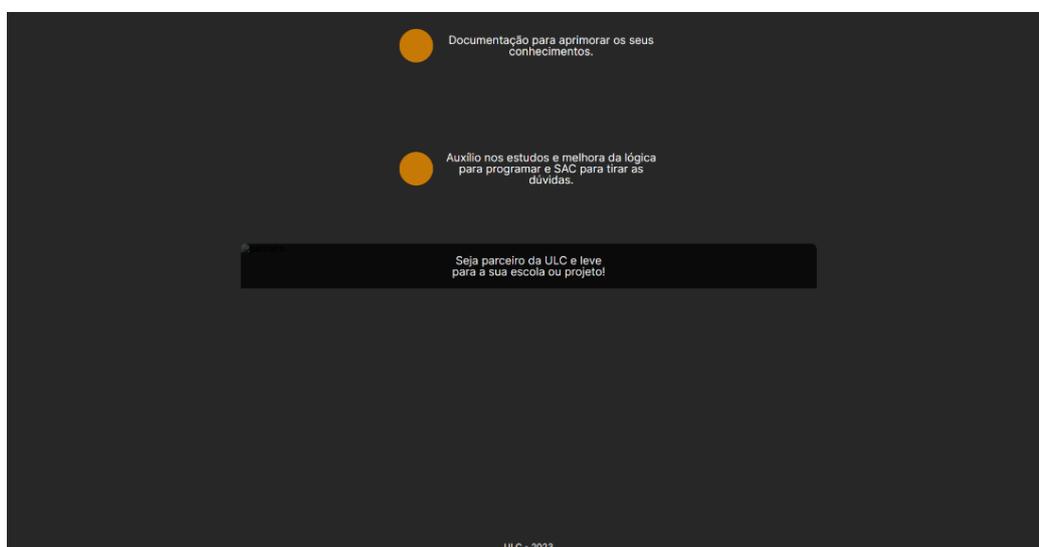
Para finalizar o Javascript foi responsável por deixar o site mais dinâmico para o usuário, que foi desenvolvido com uma linguagem de programação.

Figura 26: Front-end da página inicial.



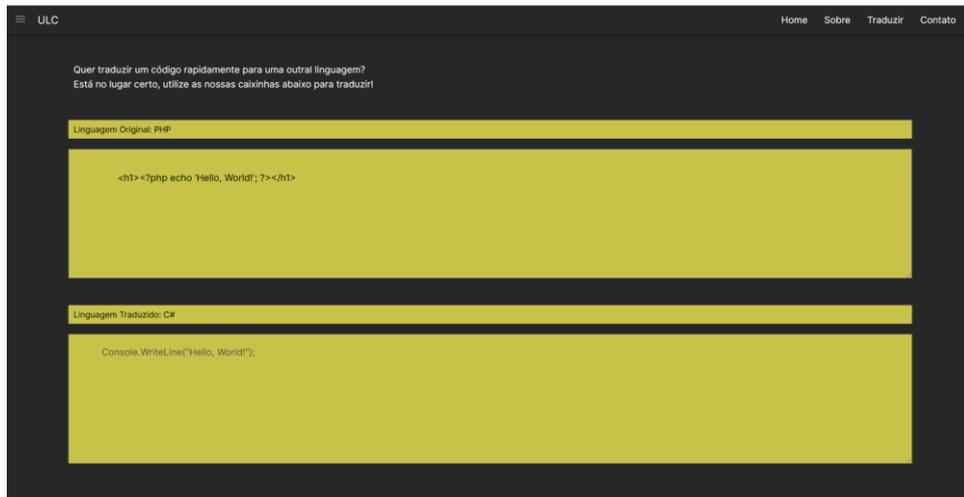
Fonte: Acervo próprio, 2023.

Figura 27: Front-end da página inicial.



Fonte: Acervo próprio, 2023

Figura 28: Front-end página da tradução.



Fonte: Acervo próprio, 2023.

4.2.2 Desenvolvimento *back-end*

No protótipo inicial do software teve o auxílio do ChatGPT para realizar o processo de tradução para ser utilizado e estudado até deixar o código do projeto maduro e pronto.

Figura 29: API do ChatGPT

```

1  from flask import Flask, request, jsonify
2  import requests
3  import json
4
5  app = Flask(__name__)
6
7  @app.route('/', methods=['POST'])
8  def traduzir_codigo():
9      OPENAI_API_KEY = "
10     link = "https://api.openai.com/v1/chat/completions"
11     id_modelo = "gpt-3.5-turbo"
12
13     body_mensagem = {
14         "model": id_modelo,
15         "messages": [{"role": "user", "content": request.form['codigo_original']}]}
16     }
17
18     headers = {
19         "Authorization": f"Bearer {OPENAI_API_KEY}",
20         "Content-Type": "application/json"
21     }
22
23     body_mensagem = json.dumps(body_mensagem)
24
25     requisicao = requests.post(link, headers=headers, data=body_mensagem)
26     resposta = requisicao.json()
27     mensagem = resposta["choices"][0]["message"]["content"]
28
29     return jsonify({"traducao": mensagem})
30
31 if __name__ == '__main__':
32     app.run()
33

```

Fonte: Acervo próprio, 2023.

4.2.2.1 Sistema de login:

O sistema de *login* atua como uma porta de entrada de um site que permite aos usuários acessar uma área restrita. Ele autentica a identidade do usuário e permite o acesso a informações privadas ou possibilita a interação com recursos específicos. Segundo o artigo de PAPPA, R. et al. (2021), o sistema de *login* é uma "camada crítica de segurança" que protege contra "acessos não autorizados" e "ataques de hackers". Além disso, ele também pode fornecer aos usuários uma experiência personalizada e permitir que eles salvem suas informações e configurações pessoais.

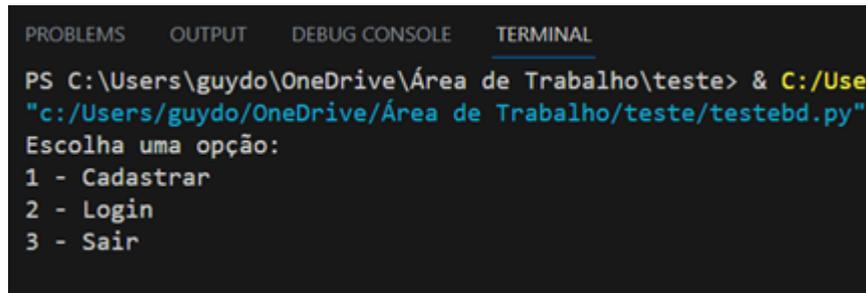
A prática de salvar senhas com *hash* e *salt* é fundamental para garantir a segurança das informações de usuários em sistemas online. O uso de *hash*, que é um algoritmo criptográfico que transforma a senha original em uma sequência de caracteres ilegível, impede que a senha seja facilmente decifrada caso seja comprometida. Já o *salt*, que é um valor aleatório adicionado à senha antes de ser "hashada", dificulta ainda mais a quebra da senha.

Conforme afirmado por O'Gorman et al. (2015), essa técnica é essencial para proteger as senhas dos usuários e evitar vazamentos de informações sensíveis. Com isso, a privacidade e a confiança dos usuários são preservadas, e a empresa ou site que oferece o serviço demonstra responsabilidade e comprometimento com a segurança.

4.2.2.2 Protótipo sistema de *login*

Foi desenvolvido uma aplicação de linha de comando para testar a integração entre o sistema de *login* e o banco de dados, visando verificar pontos como: conexão com a base de dados, captura de *inputs*, formato das informações armazenadas e conversão da senha do usuário para o formato *hash* e *salt*. Quando ativado, o programa inicia um menu simples com as opções de cadastro – salva as informações do usuário em nossa base de dados –, *login* – simula o acesso do usuário ao *site* – e sair.

Figura 30: Protótipo e login



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\guydo\OneDrive\Área de Trabalho\teste> & C:/Users/guydo/OneDrive/Área de Trabalho/teste/testebd.py
Escolha uma opção:
1 - Cadastrar
2 - Login
3 - Sair

```

Fonte: Acervo Próprio, 2023

A função “*while True*” indica que o bloco de código dentro dele será executado repetidamente sem parar, a menos que uma condição de parada seja atingida que, em nosso programa é a escolha da opção “3 – sair”, a qual possui um *break* que finaliza o programa.

Figura 31: Código do menu de login

```

# Menu de opções
while True:
    print("Escolha uma opção:")
    print("1 - Cadastrar")
    print("2 - Login")
    print("3 - Sair")

    opcao = input()

    if opcao == "1":
        cadastrar()
    elif opcao == "2":
        login()
    elif opcao == "3":
        break
    else:
        print("Opção inválida.")

```

Fonte: Acervo Próprio, 2023

Em seguida utilizamos a biblioteca “*mysql.connector*” estabelecer a comunicação entre o aplicativo e o bancos de dados. De acordo com a documentação oficial do MySQL, essa biblioteca é um “*driver* Python autônomo que fornece acesso a bancos de dados MySQL e MariaDB usando o protocolo MySQL”. Ela permite que o desenvolvedor crie, leia, atualize e delete informações em um banco de dados MySQL diretamente através de código Python. Esta biblioteca é chamada em diversos momentos ao longo do código.

Figura 32: Bibliotecas e código para conexão com o Banco de dados

```
testebd.py > login
1  import hashlib
2  import uuid
3  import mysql.connector
4
5  # Conectando ao banco de dados
6  mydb = mysql.connector.connect(
7      host="localhost",
8      user="root",
9      password="",
10     database="ulc1"
11 )
```

Fonte: Acervo Próprio, 2023

Na etapa seguinte criamos o sistema de cadastro utilizando as bibliotecas `hashlib`, responsável por realizar operações de hash, e `uuid`, a qual gera um identificador único (salt) que será concatenado a senha do usuário para, então, serem convertidos em uma hash. Na imagem 3 podemos observar o código utilizado para encriptar a senha do usuário.

Na linha 22 do código em questão gera um valor aleatório de *bits* e em seguida a função `hex()` converte esse valor em uma string hexadecimal de 32 caracteres que será armazenada na variável chamada `salt`. A linha 25 realiza a concatenação da senha digitada pelo usuário com o valor gerado do salt, armazenando o resultado na variável `senha_salt`. A linha 28 é responsável por criar uma hash criptográfica da combinação de uma senha e um salt, utilizando o algoritmo SHA-256. A função `encode()` converte a combinação em uma sequência de bytes, necessária para a função `hashlib.sha256()` processar o valor. Por fim, a função `hexdigest()` converte a hash em uma string hexadecimal, que pode ser armazenada com segurança no banco de dados.

Podemos observar o código que realiza a inserção dos dados na tabela "clientes" no banco de dados. A função "cursor()" do objeto "mydb" representa a conexão com o banco de dados e é utilizada para criar um cursor que será usado para executar as operações no banco de dados. A variável "query" armazena uma string que contém o comando SQL "INSERT INTO", que é usado para inserir novos registros na tabela enquanto "values" contém os dados que serão inseridos. O método "execute()" é usado para executar a query SQL, inserindo os dados na tabela. Por fim,

a função "commit()" é chamada no objeto "mydb", confirmando que as alterações no banco de dados devem ser efetivadas.

Figura 33: Função cadastrar

```

12
13 # Função para cadastrar novo cliente
14 def cadastrar():
15     # Recebendo dados do cliente
16     nome = input("Digite seu nome: ")
17     email = input("Digite seu email: ")
18     celular = input("Digite seu celular: ")
19     senha = input("Digite sua senha: ")
20
21     # Gerando salt aleatório
22     salt = uuid.uuid4().hex
23
24     # Adicionando salt à senha do cliente
25     senha_salt = senha + salt
26
27     # Hashing da senha com SHA256
28     senha_hashed = hashlib.sha256(senha_salt.encode()).hexdigest()
29
30     # Inserindo cliente no banco de dados
31     cursor = mydb.cursor()
32     query = "INSERT INTO clientes (clienteNome, clienteEmail, clienteCelular, senhaHashed, salt, idPlanoCliente) VALUES (%s, %s, %s, %s, %s, %s)"
33     values = (nome, email, celular, senha_hashed, salt, 1) # idPlanoCliente = 1 por padrão
34     cursor.execute(query, values)
35     mydb.commit()

```

Fonte: Acervo Próprio, 2023

Para realizar a validação do usuário, a função "login()" recebe como entrada o seu e-mail e senha que são armazenados em variáveis com o mesmo nome. Em seguida é feito uma pesquisa no banco de dados por um cliente com o e-mail fornecido e, quando encontrado, a função extrai o salt e a senha criptografada que foram criados durante o cadastro.

Feito isso é realizada uma nova concatenação entre a senha digitada durante o login com o salt extraído da nossa base de dados. Essa string é criptografada utilizando o mesmo algoritmo de hash (SHA-256), o resultado obtido é então comparado com a senha armazenada no banco de dados. Se as senhas forem iguais, o login é realizado com sucesso. Caso contrário, o programa retornará a mensagem "Senha incorreta". Se o e-mail não for encontrado a mensagem de "Email não cadastrado" será mostrada.

Figura 34: Função login

```

46 # Função para fazer login
47 def login():
48     ...# Recebendo dados de login do cliente
49     ...email = input("Digite seu email: ")
50     ...senha = input("Digite sua senha: ")
51
52     ...# Buscando cliente no banco de dados pelo email
53     ...cursor = mydb.cursor()
54     ...query = "SELECT * FROM clientes WHERE clienteEmail = %s"
55     ...values = (email,)
56     ...cursor.execute(query, values)
57     ...result = cursor.fetchone()
58
59     ...if result:
60         ...# Extraindo salt e senha do cliente
61         ...salt = result[5]
62         ...senha_hashed = result[4]
63
64         ...# Adicionando salt à senha digitada
65         ...senha_salt = senha + salt
66
67         ...# Hashing da senha com SHA256
68         ...senha_hashed_input = hashlib.sha256(senha_salt.encode()).hexdigest()
69
70         ...if senha_hashed == senha_hashed_input:
71             ...print("Login realizado com sucesso!")
72         ...else:
73             ...print("Senha incorreta.")
74     ...else:
75         ...print("Email não cadastrado.")
76

```

Fonte: Acervo Próprio, 2023

Figura 35: Cadastro dos usuários mostrando o salt utilizado e a senha_hashed gerada.

```

Escolha uma opção:
1 - Cadastrar
2 - Login
3 - Sair
1
Digite seu nome: teste
Digite seu email: teste@email.com
Digite seu celular:
Digite sua senha: teste123
Cadastro realizado com sucesso!

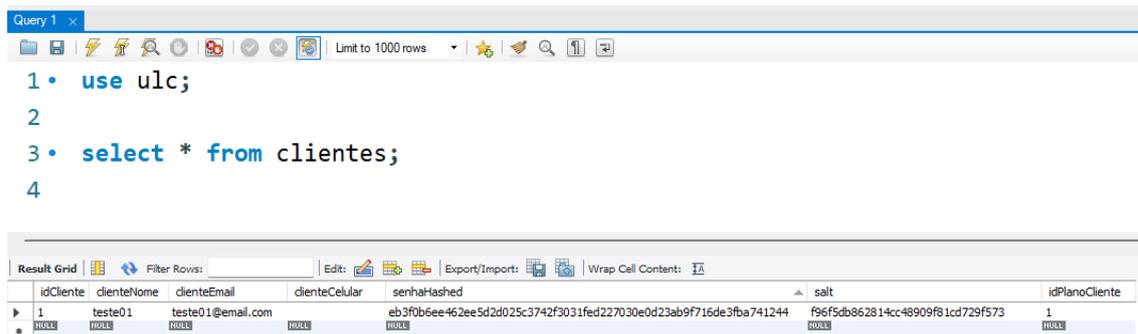
salt:
1a39770820004061811f3e8be1ff5cdd

senha_hashed:
473c6a47ff8ac97551256f95cee60a9abac2a36f74a13ed00bfffde862e1c976c

```

Fonte: Acervo Próprio, 2023.

Figura 36: Dados salvos no Banco de dados



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. Below the toolbar, the SQL query is displayed:

```

1 • use ulc;
2
3 • select * from clientes;
4

```

Below the query, a 'Result Grid' is shown with the following data:

idCliente	clienteNome	clienteEmail	clienteCelular	senhaHashed	salt	idPlanoCliente
1	teste01	teste01@email.com		eb3f0b6ee462ee5d2d025c3742f3031fed227030e0d23ab9f716de3fba741244	f96f5db862814cc48909f81cd729f573	1

Fonte: Acervo Próprio, 2023

Figura 37: Login mostrando a comparação entre a senha salva na base de dados (senha_hashed) e a recém criptografada (senha_hashed_input)

```

Escolha uma opção:
1 - Cadastrar
2 - Login
3 - Sair
2
Digite seu email: teste@email.com
Digite sua senha: teste123
salt:
1a39770820004061811f3e8be1ff5cdd

senha_hashed:
473c6a47ff8ac97551256f95cee60a9abac2a36f74a13ed00bffde862e1c976c

senha_hashed_input:
473c6a47ff8ac97551256f95cee60a9abac2a36f74a13ed00bffde862e1c976c

Login realizado com sucesso!

```

Fonte: Acervo Próprio, 2023

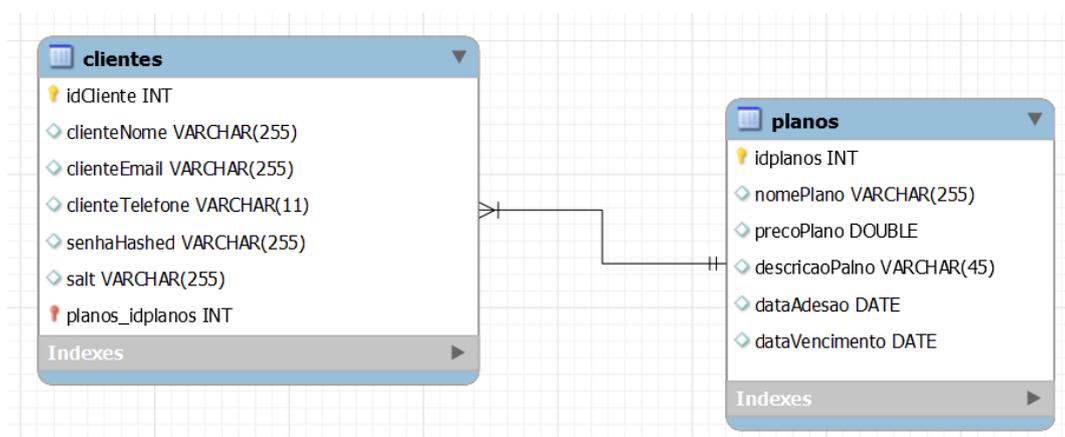
4.2.3 Desenvolvimento banco de dados.

4.2.3.1 – Desenvolvimento do DER

O planejamento do banco de dados teve início pela criação do DER, ilustrando a organização necessária para o controle e armazenamento dos dados do projeto. Como resultado obtivemos duas tabelas, clientes e planos, possuindo uma relação 1:n.

A tabela “clientes” contém informações relativas aos usuários do ULC, nela são armazenados dados para a identificação do cliente, formas de contato e o plano selecionado. Visando aprimorar a segurança do nosso sistema, as senhas de nossos clientes são armazenadas utilizando criptografia hash e salt. Esta técnica será abordada futuramente. Já a tabela “planos” armazena informações sobre as diversas modalidades de assinaturas que a ULC disponibiliza.

Figura 38: Diagrama Entidade Relacionamento



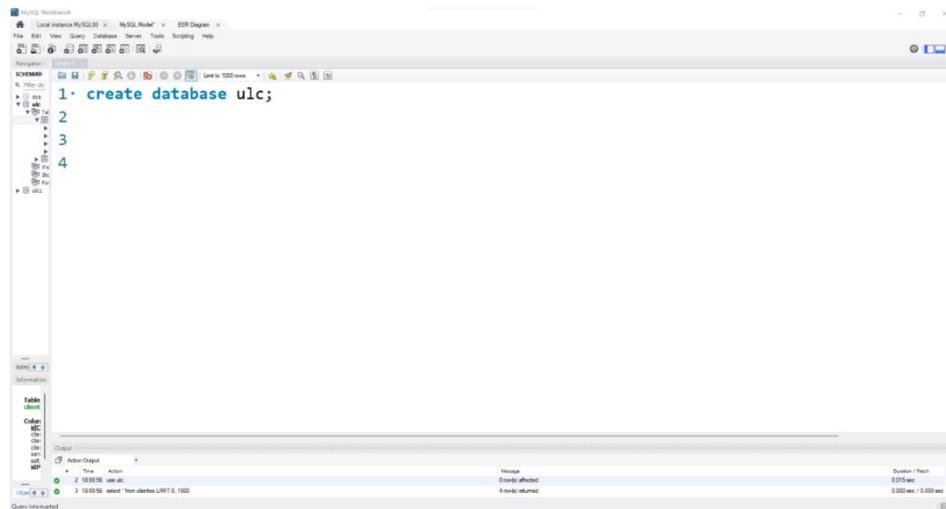
Fonte: Acervo próprio, 2023

4.1.5.2 – Implementação

Após a definição do DER demos início a implementação do banco de dados utilizando a ferramenta MySQL *workbench* 8.0 CE, onde as tabelas e seus atributos foram gerados através de *Queries* contendo comando SQL. Entre os comandos disponibilizados pelo SGBD, utilizamos:

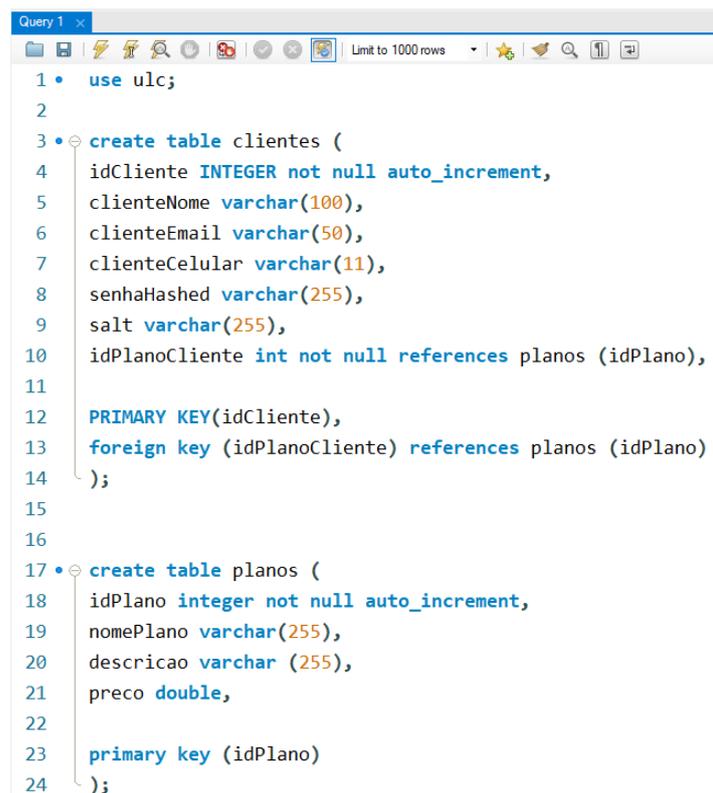
1. *create database*: Cria um banco de dados.
2. *Create table*: cria uma tabela.
3. *Primary key*: define uma coluna como chave primária da tabela, ou seja, um identificador único do objeto.
4. *Foreing key*: cria uma restrição que relaciona uma coluna da tabela atual com uma coluna de outra tabela, garantindo a integridade referencial dos dados.
5. *Insert into*: Adiciona um registro a tabela.
6. *Select * from*: Retorna todos os registros de uma tabela.

Figura 39: Ambiente MySQL Workbench com comando para criar o banco de dados



Fonte: Acervo próprio, 2023

Figura 40: Criação tabelas “clientes” e “planos”



Fonte: Acervo Próprio, 2023

Figura 41: Inserindo clientes diretamente no banco de dados

```

Query 1 x
Limit to 1000 rows
1 • use ulc;
2
3 • INSERT INTO clientes (clienteNome, clienteEmail, clienteCelular, senhaHashed, salt, idPlanoCliente) VALUES
4 ('Lucas Silva', 'lucas.silva@teste.com', '11988887777', '', '', 1),
5 ('Ana Souza', 'ana.souza@teste.com', '11977776666', '', '', 2),
6 ('Pedro Santos', 'pedro.santos@teste.com', '11966665555', '', '', 3),
7 ('Julia Oliveira', 'julia.oliveira@teste.com', '11955554444', '', '', 2),
8 ('Mariana Lima', 'mariana.lima@teste.com', '11944443333', '', '', 1);

```

Fonte: Acervo Próprio, 2023

Figura 42: Inserindo clientes diretamente no banco de dados

```

Query 1 x
Limit to 1000 rows
1 • use ulc;
2
3 • select * from clientes;
4

```

idCliente	clienteNome	clienteEmail	clienteCelular	senhaHashed	salt	idPlanoCliente
1	Lucas Silva	lucas.silva@teste.com	11988887777			1
2	Ana Souza	ana.souza@teste.com	11977776666			2
3	Pedro Santos	pedro.santos@teste.com	11966665555			3
4	Julia Oliveira	julia.oliveira@teste.com	11955554444			2
5	Mariana Lima	mariana.lima@teste.com	11944443333			1
*	NULL	NULL	NULL	NULL	NULL	NULL

Fonte: Acervo Próprio, 2023

4.3 Viabilidade da UPLINKERCODE

4.3.1 Logotipo da uplinkercode

A logotipo da UpLinkerCode (ULC), foi desenvolvida com as cores da paleta original com tons de preto e cinza e com elementos que remetem ao desenvolvedor web e a tecnologia da informação.

Figura 43: Logotipo da ULC

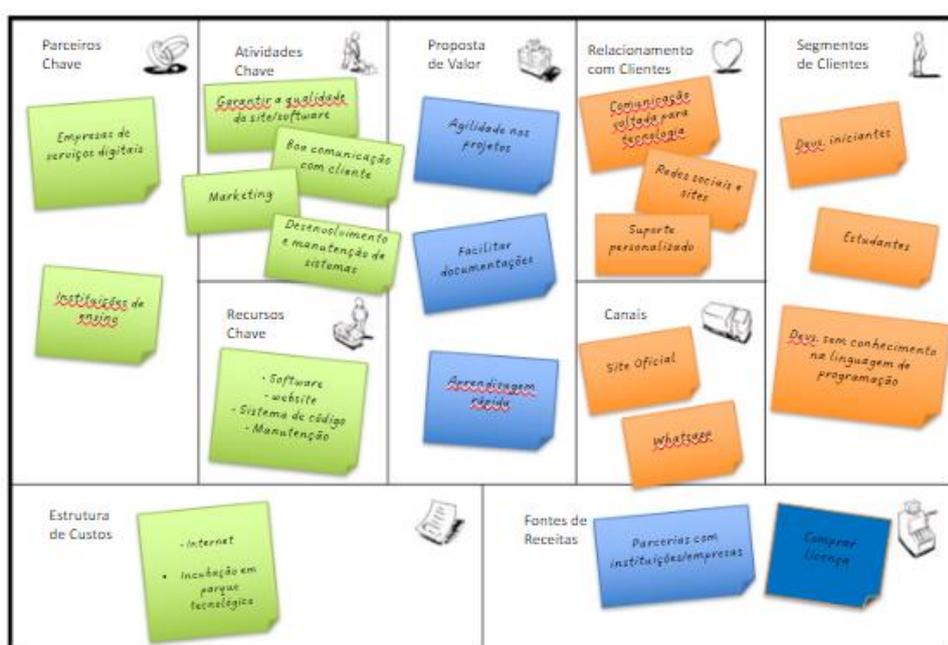


Fonte: Acervo próprio, 2023

4.3.2 Canvas

No canvas desenvolvido pela UpLinkerCode, foram levantadas algumas informações importantes para a realização do modelo de negócio de acordo com o cenário da tecnologia da região que irá atuar inicialmente.

Figura 44: Canvas inicial da UpLinkerCode



Fonte: Acervo próprio, 2023

4.3.2.1 Parceiros chave

São os principais parceiros estratégicos da ULC, para utilizar os serviços ou para ajudar na divulgação da marca.

No caso são as empresas de serviços digitais, que podem se interessar no software para a melhoria na gestão de tempo dos seus serviços e/ou produtos e as instituições de ensino, que podem utilizar para auxiliar os alunos no aprendizado complementando as aulas.

4.3.2.2 Atividades chave

Quais são as principais atividades para manter a qualidade e o bom funcionamento do produto e/ou serviço que está sendo vendido. Para manter um bom funcionamento de um software é necessário a manutenção do código, manter o sistema atualizado evitando bugs futuros, a participação do cliente com o marketing e para aumentar a área de divulgação da marca pelo mercado.

4.3.2.3 Proposta de valor

A proposta de valor, é importante para analisar quais são os benefícios que ele vai trazer para quem adquirir, ou seja, quais são as dores dos clientes que o produto vai solucionar. Com o software, espera-se que o cliente otimize o seu tempo criando ou arrumando um código e tenha um aprendizado rápido caso for um estudante que está iniciando na área de programação.

4.3.2.4 Relacionamento com o cliente

A forma como a empresa vai conversar com o seu cliente, pode fazer muita diferença no sucesso do negócio, como foi mostrado no modelo inicialmente a ULC, vai utilizar uma linguagem mais tecnológica, suporte personalizado de acordo com as necessidades do cliente e por meio das redes sociais para ter mais proximidade com o público-alvo.

4.3.2.5 Segmentos de clientes

Definir o seu público-alvo e assim chegar em uma persona com as suas características e gostos definidos, é importante para não tentar agradar a todos e assim afetando o desempenho do negócio que poderia ser promissor.

O público-alvo da ULC, são os desenvolvedores iniciantes e estudantes que estão começando a sua jornada na tecnologia e os desenvolvedores que já possuem conhecimento em alguma linguagem e que necessita trabalhar com uma linguagem nova e diferente.

4.3.2.6 Recursos chave

O que é necessário ter para manter o produto, são as partes principais para a realização do negócio, sendo o produto e/ou serviço prestado.

A ULC, por ser um software não vai precisar de muitos recursos, precisando apenas de: software, website, manutenção e o código web.

4.3.2.7 Canais

Os canais são os locais em que o futuro cliente vai conseguir encontrar a empresa, no caso os potenciais clientes vão poder encontrar a ULC no site oficial e nas redes sociais como o *Whatsapp*.

4.3.2.8 Estruturas de custos

São os gastos que vão existir no seu negócio, como por exemplo o custo fixo e variável que precisam estar presentes para analisar o investimento necessário. Com o Canvas da ULC, vai precisar apenas de Internet e incubadora em parques tecnológicos para continuar no desenvolvimento do projeto e receber investimentos.

4.3.2.9 Fontes de receitas

Por último tem as fontes de receitas, que é a forma com que o seu negócio vai conseguir gerar receitas. A ULC, possui apenas o software para gerar receita para a marca.

5. CONCLUSÃO

5.1 Considerações Finais

Ao analisar os resultados obtidos com as pesquisas realizadas, pode-se considerar que o *software* UpLinkerCode possui espaço favorável no mercado da tecnologia, acompanhando o aumento da procura para novos e bons programadores, cada vez mais alunos entrando na área e junto vem a dificuldade de aprendizado em uma parcela.

O foco da UpLinkerCode, é auxiliar nos estudos dos alunos para facilitar o entendimento de lógica de programação, dessa forma diminuindo a grande taxa de desistências durante o início e não comprometendo ainda mais a falta de profissionais bem qualificados.

Porém analisando o contexto, foi possível perceber que a dificuldade de encontrar novos profissionais não está apenas no aprendizado inicial da programação, mas a educação básica pública brasileira e o contexto em que os jovens vivem atualmente, traz um cenário desfavorável para a tecnologia. A dificuldade com matemática nas escolas pode acabar atrapalhando no desenvolvimento do raciocínio lógico da pessoa e atualmente os jovens se interessam mais por profissões ligadas a internet, mas em outras partes, achando que a programação é chata e sem graça.

Com a conscientização dos jovens e dos mais velhos sobre o cenário desastroso do mercado tecnológico, o auxílio do software da ULC, é esperado dar um incentivo a mais para a entrada de novos desenvolvedores no mercado.

5.2 Sugestão para trabalhos futuros

Para o desenvolvimento de um trabalho futuro, pode-se realizar um aprimoramento da ideia de tradutor de linguagens para auxiliar no aprendizado de quem está iniciando os estudos na área de programação e/ou desenvolver algum projeto que também ajude outras áreas de TI, como a área de dados, *front-end* e outros.

REFERÊNCIAS

ALMEIDA, Marilane. **CURSO ESSENCIAL DE LÓGICA DE PROGRAMAÇÃO**. - São Paulo: Digerati Books, 2008.

ÁREA DE TI DEVE GERAR QUASE 420 MIL VAGAS ATÉ 2025, MAS FALTAM PROFISIONAIS. G1.com, 2023. Disponível em: Área de TI deve gerar quase 420 mil vagas até 2025, mas faltam profissionais | Jornal Hoje | G1 (globo.com). Acesso em: 27/03/2023.

CANZIBA, Elvis. **Hands-on UX Design for developers**. 1. ed. Birmingham: Packt Publishing, 2018.

DAVID, Tobias de; CARDOSO, Érico Edú Corrêa. **A falta de profissionais de tecnologia de informação no mercado de trabalho**. A falta de profissionais de tecnologia de informação no mercado de trabalho | Cardoso | Uma Nova Pedagogia para a Sociedade Futura (emnuvens.com.br). Acesso em: 09/04/2023.

FERRO, Pedro. **FALTA DE PROFISSIONAIS DE TI SÓ TENDE A AUMENTAR, DIZEM ESPECIALISTAS**. Disponível em: <https://jornal.usp.br/atualidades/falta-de-profissionais-de-ti-so-tende-a-aumentar-dizem-especialistas/>. Acesso em: 27/03/2023.

FIGMA. Logotipo Figma. Disponível em: <https://www.figma.com/>. Acesso em: 09/04/2023.

FLANAGAN, David. **JAVASCRIPT: O GUIA DEFINITIVO**. 6^o Edição. Tradução para a língua portuguesa copyright 2013, Bookman Companhia Editora LTDA.

FORBES. **Profissões do futuro: as certificações mais buscadas pelo mercado de TI**. Disponível em:

<https://forbes.com.br/carreira/2022/10/profissoes-do-futuro-as-certificacoes-mais-buscadas-pelo-mercado-de-ti/>. Acesso em:27/03/2023.

GAZETA DO POVO, **Brasil vai precisar de quase 800 mil profissionais de TI até 2025**. Disponível em: <https://www.gazetadopovo.com.br/gazz-conecta/brasil-vai-precisar-de-quase-800-mil-profissionais-de-ti-ate-2025/>. Acesso em:13/03/2023.

JUNIOR, Peter Jandl. **CURSO BÁSICO DA LINGUAGEM C**. Novatec Editora Ltda, 2019.

LUNKES, Leomar. **TRANSFORMANDO CHATGPT E MIDJOURNEY EM OPORTUNIDADES DE NEGÓCIO: 100 IDEIAS CRIATIVAS PARA IMPULSIONAR SUA RENDA**. Bibliomundi, 2023.

LOGOS PNG: DOWNLOAD DE LOGOMARCAS. Logotipo Javascript. 1024x1024 pixels. Disponível em: <https://logospng.org/logo-javascript/>. Acesso em: 10/06/2023.

LOGOS PNG: DOWNLOAD DE LOGOMARCAS. Logotipo CSS3. 1024x1024 pixels. Disponível em: <https://logospng.org/logo-css-3/>. Acesso em: 10/06/2023.

LOVATTI, Bruna Gomes; VIEIRA, Lara Santos; MARQUES, Kédyma; SCOLFORO, Monica Altoe. **A PROGRAMAÇÃO NO ENSINO BÁSICO: FORMANDO ALUNOS PARA SOCIEDADE TECNOLÓGICA**. Disponível em: <https://multivix.edu.br/wp-content/uploads/2018/04/revista-ambiente-academico-edicao-5-artigo-8.pdf>. Acesso em: 06/04/2023.

MACDONALD, Diana. **PRACTICAL UI PATTERNS FOR DESIGN SYSTEMS**. Victoria, VIC, Australia; Apress, 2019.

MENDES, Joice Barbosa; MUNIZ, Rafael da Silva. **LÓGICA DE PROGRAMAÇÃO COM PORTUGOL**. Casa do código, 2022.

Microsoft learn. DOCUMENTAÇÃO DO MICROSOFT C++, C E ASSEMBLER. 800X800 pixels. Disponível em: <https://learn.microsoft.com/pt-br/cpp/?view=msvc-170>. Acesso em: 06/04/2023.

MUNIZ, Antônio; SOUZA, Anderson Gonzaga de; LABRIOLA, Luiz Eduardo; BRANCO, Leandro; DIAS, Luciana T. C.; BOSIO, Marcello. **JORNADA TRANSFORMAÇÃO DIGITAL NO BRASIL: UNINDO PRÁTICAS E CASES DE EXPERIÊNCIAS BRASILEIRAS QUE POTENCIALIZAM A ECONOMIA DIGITAL NO BRASIL**. Rio de Janeiro; Brasport Livros e Multimídia Ltda, 2022.

MYSQL. Logotipo MySQL. 162x111 pixels. Disponível em: <https://www.mysql.com/>. Acesso em: 10/06/2023.

MYSQL. Logotipo MySQL Workbench. 162x111 pixels. Disponível em: <https://dev.mysql.com/downloads/workbench/>. Acesso em: 10/06/2023.

Rede de tecnologia social (RTS). **TECNOLOGIAS SOCIAIS: CAMINHOS PARA A SUSTENTABILIDADE**. Gráfica brasil, Brasília, 2009.

PEREIRA, Juliana Batista; TOLENTINO-NETO, Luiz Caldeira Brant de (2015). ***O que os dados do SAEB nos dizem sobre o desempenho dos estudantes em matemática?*** Educação Matemática Pesquisa, 17(2), pp. 309-333.

PIGNATI, Giovana. **VAGAS DE EMPREGO PARA SERVIÇOS DE TI CRECEM QUASE 80% EM 10 ANOS NO BRASIL**. Canaltech, 2023. Disponível em: <https://canaltech.com.br/mercado/vagas-de-emprego-para->

servicos-de-ti-crescem-quase-80-em-10-anos-no-brasil-238482/. Acesso em: 10/06/2023.

PINHEIRO, Allan Petterson da Silva. **UX DESIGN INTRODUZIDO NO DESENVOLVIMENTO DE INTERFACES GRÁFICAS**. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/235/9445/1/21400530.pdf>. Acesso em: 20/04/2023.

SANTOS, Josiel Almeida; FRANÇA, Kleber Vieira; SANTOS, Lúcia S. B. dos. **DIFICULDADES NA APRENDIZAGEM DE MATEMÁTICA**. 2007. Trabalho de conclusão de curso (Licenciatura em matemática) - Centro Universitário Adventista de São Paulo Campus São Paulo.

SILVA, Francisco Leocassio da; MOREIRA, Irlan Arley Targino. **Análise das dificuldades na aprendizagem de programação no curso de análise e desenvolvimento de sistemas do IFRN/Pau dos Ferros**. *In*: ENCONTRO UNIFICADO DE COMPUTAÇÃO DO PIAUÍ (ENUCOMPI), 14, 2021, Picos. Disponível em: <https://sol.sbc.org.br/index.php/enucompi/article/view/17752>. Acesso em: 12/04/2023.

SORRI, K.; SEPPÄNEN, M.; STILL, K.; VALKOKARI, K. **BUSINESS MODEL INNOVATION WITH PLATFORM CANVAS**. *Journal of Business Models*, 2019, 7(2), 1-13.

SILVA, Maurício Samy. **FUNDAMENTOS DE HTML5 E CSS3**. Novatec Editora Ltda, 2015.

UNIVALE-LITE. Portugol-Studio. 1280x640 pixels. Disponível em: <http://univali-lite.github.io/Portugol-Studio/>. Acesso em: 25/06/2023.

W3. W3c html5 logo. 1024X1024 pixels. Disponível em: <https://www.w3.org/html/logo/>. Acesso: 13/06/2023.

APÊNDICES

Apêndice A – Questionário

2- Perguntas para pesquisa online aberto a comunidade.

1- Qual a sua faixa etária?

- Menos de 18 anos
- Entre 18 a 25 anos
- Entre 26 a 30 anos
- Entre 31 a 35 anos
- Entre 36 a 39 anos
- Mais de 40 anos.

2- Qual o seu gênero?

- Mulher
- Homem
- Outros

3- Qual a sua cidade?

- Pindamonhangaba
- Taubaté
- São José dos Campos
- São Paulo
- Guaratinguetá
- Outros

4- Você estuda programação?

- Sim
- Não

5- Se a sua resposta anterior for sim, você sente dificuldades para aprender uma linguagem de programação nova ou em lógica de programação?

- Tive muita dificuldade no início

- Tive pouca dificuldade
- Não senti dificuldades em aprender linguagem nova
- Não se aplica
- Outros

6- Em geral, a democratização do acesso a tecnologia pode ajudar a inserir mais pessoas no mercado da Tecnologia da informação?

- Sim
- Não
- Talvez
- Outros.

7- Qual a importância de aprender a lógica de programação:

- 1
- 2
- 3
- 4
- 5

8- Em qual plataforma você estuda as linguagens de programação?

- Youtube
- Plataformas pagas
- Stack Overflow
- Pesquisas no google
- Outros.

9- Você acha que um software que traduz códigos de uma linguagem para outra de forma simultânea e disponibilizando documentações da linguagem, ia ter ajudado nos seus estudos inicialmente?

- Sim, facilitaria no processo de aprendizado.
- Não, não tive dificuldades para aprender várias linguagens diferentes rápido.
- Não se aplica

10- Em cursos profissionalizantes e/ou em escolas, ao seu ver **existe a necessidade de implantação de um software** que otimiza o tempo de estudos em programação e lógica?

- Sim
- Não
- Talvez

11-Quais das opções abaixo você acha que **complementam** os seus estudos?

- E-book
- Guias com dicas
- Vídeo aulas
- Projetos como exemplo

12- Qual a faixa de preço que te interessa pagar por um software que otimiza e facilita o aprendizado em programação?

- Até 60,00
- Entre 61,00 a 70,00
- Entre 71,00 a 80,00
- Entre 81,00 a 90,00
- Entre 91,00 a 100,00
- Mais de 100,00

13- Quais os locais que você prefere receber notícias sobre a ULC?

- Redes sociais (Instagram, Facebook, Twitter, TikTok,...)
- E-mail
- Propagandas
- Panfletos, outdoor

14- Uma rede social para acompanhar os conteúdos sobre TI, dicas e o trabalho da ULC te interessaria

- Sim

- Não