
1. Introdução

Com o avanço da tecnologia, parte do mercado de vendas e serviço teve sua migração para o mundo digital. É visto que hoje as grandes empresas possuem a maior parte de sua receita através de vendas online, pois o número de pessoas que são impactadas através da internet é muito maior do que no mundo físico. Isso fez com que grandes negócios se tornassem maiores ainda, aumentando com isso também, a desigualdade com outros empreendedores. Principalmente para aqueles que não tem conhecimento, ou poder para investir no mercado digital. Desde a divulgação em redes sociais até a criação de um *E-commerce*, para ter sucesso no mundo digital, é preciso ter o conhecimento de como utilizar as ferramentas certas para impactar o maior número de pessoas, o que depende muitas vezes de um profissional especializado, que acaba sendo inacessível para pessoas com baixa renda.

Em contraste dessa dificuldade em obter apoio especializado, é o número de recém-formados na área de tecnologia, marketing digital, publicidade e novos integrantes na indústria 4.0 no geral que não conseguem ingressar em um primeiro emprego. Onde muitas vezes a motivação da recusa, está na falta de experiência comprovada, que se mostra difícil de obter devido à falta de oportunidade.

Dado o problema, tem-se por objetivo promover a conexão entre estes novos profissionais juntamente com os solicitantes mais afetados pela desigualdade digital. De forma com que esta demanda possa evoluir ambas as partes, porém sem gerar custos financeiros. Com o intuito apenas de capacitar o aspirante a uma nova profissão promovendo a experiência com um problema real, e ajudar o demandante a evoluir seu negócio gerando uma maior competitividade frente a concorrência desigual de grandes empresas.

Para que isto se torne possível, é idealizado o desenvolvimento de uma plataforma digital para realizar essa união. Utilizando-se de conhecimentos da engenharia de software para levantar os requisitos de uma aplicação que seja de fácil acesso a vários dispositivos, dado que o público-alvo muitas vezes não terá condições de possuir um equipamento robusto.

2. Referencial Teórico

2.1. Impacto Social

Durante os últimos anos vêm sendo muito discutido a respeito das transformações tecnológicas e sociais que afetam a forma como as pessoas interagem, se comunicam, desenvolvem soluções para seus problemas pessoais e profissionais e encaram suas rotinas de trabalho (LAST et al., 2014; WANG et al., 2019). A quarta revolução industrial (ou Indústria 4.0), pode ser considerada como um produto destas transformações, motivada pela necessidade em flexibilizar processos de decisão em diferentes níveis de mercado e de contexto social e um catalisador frente a estas transformações, capaz de impulsionar cada vez mais a competitividade pautada em informação, decisão e de mudanças de papel no mercado de trabalho (MORRAR et al., 2017; VAIDYA et al., 2018).

No entanto, existe uma grande preocupação quanto ao advento de populações mais carentes frente a todas estas transformações. Cada vez mais a desigualdade tecnológica vem se mostrando presente em diversos cenários, e em um mundo onde tudo se converge para a

digitalização, a inclusão digital se torna o principal pilar para a inclusão social. A digitalização atua como um potencializador da informação, aumentando a velocidade e o volume das informações. Enquanto isso é uma ferramenta poderosa para grandes empresas divulgarem seus serviços, aumenta a desigualdade e diminui a competitividade para aqueles sem o mesmo nível de influência (DEMO, 2005).

Quando trazemos esse cenário para a concorrência de mercado e a divulgação de serviços, a disparidade entre empreendedores inseridos no meio digital versus aqueles que não estão é algo que chama a atenção. Hoje boa parte da população consumidora realiza suas compras sem sair de casa, isso faz com que serviços que possuem disponibilidade via internet tenham um grande salto no número de vendas. Sem falar na evolução do Marketing digital, que demonstra um alto poder na quantidade de pessoas impactadas por conta da facilidade com que a informação trafega nas redes (ROSA et al., 2017).

Paralelo a isso, é visto que cada vez mais a necessidade de profissionais capacitados para atuar neste novo formato de indústria tem aumentado. Porém mesmo com essa alta procura, alguns ingressantes têm dificuldades em encontrar uma primeira oportunidade, e um dos grandes motivos para isso acontecer é a falta de experiência profissional. Isso porque entende-se que exista um *gap* de capacitação técnica e profissional entre o que é apresentado nas escolas e universidades e o que é exigido realmente no mundo corporativo (FORTUS et al., 2005; KAMARUZAMAN et al., 2019). E conforme apresentado por Menhorn e Slomka, (2013), esse problema cresce cada vez mais justamente por não gerar essa vazão suficiente de egressos, que possuam tanto habilidades técnicas quanto não-técnicas para acompanhar as tendências corporativas e tecnologias que são lançadas à indústria e à sociedade.

2.2. Levantamento de Tecnologias

Para construir a plataforma, foi realizado um estudo de caso para escolher a tecnologia que mais se adequasse ao que está sendo construído, levando em consideração a curva de aprendizado da *stack*, suporte da comunidade, ferramentas disponíveis, e nível de conhecimento dos integrantes. Após analisar algumas linguagens de programação voltadas a Web (JAVINPAUL, 2021), como PHP, Java, Python, foi optado por utilizar um *Superset* da linguagem *Javascript* (MDN Contributors, 2022), o *Typescript* (MICROSOFT, 2023).

Typescript é uma linguagem de programação de código aberto desenvolvida pela Microsoft que estende o *JavaScript*, adicionando tipos estáticos opcionais. Essa extensão proporciona aos desenvolvedores uma ferramenta poderosa para detecção de erros durante a fase de desenvolvimento, melhorando a robustez e a manutenibilidade do código. Com um compilador que converte o *TypeScript* em *JavaScript* padrão, a linguagem oferece suporte avançado a recursos modernos, como classes, interfaces e módulos. Possui uma ampla comunidade que veem contribuindo fortemente para o ecossistema da linguagem, facilitando assim, encontrar conteúdo e *Frameworks* que facilitem o desenvolvimento.

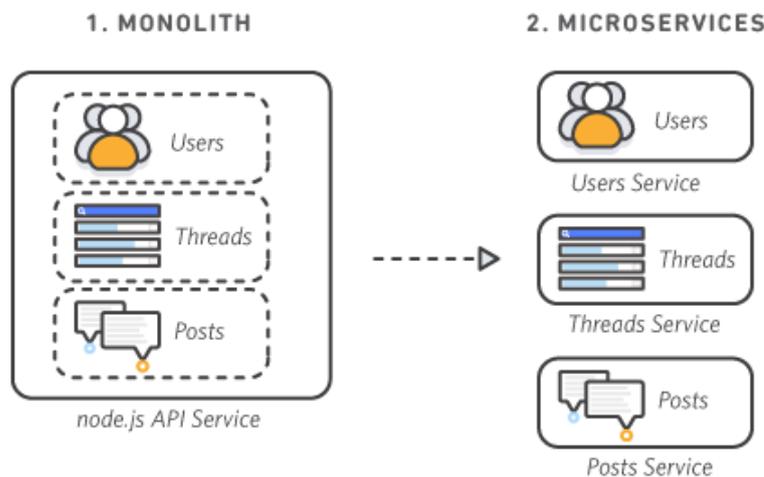
Um dos *Frameworks* para *Backend* que a linguagem fornece é o *Express*. O *Express* no *Node.js* (NODE.JS contributors, 2022) destaca-se por sua simplicidade, oferecendo um *framework* minimalista e flexível. Com *middleware* eficiente, facilita a manipulação de solicitações e respostas, enquanto a gestão de rotas é simplificada. Sua extensibilidade permite fácil integração de funcionalidades adicionais. Com uma comunidade ativa, o *Express* é amplamente adotado e oferece flexibilidade na escolha de ferramentas. Essas características fazem do *Express* uma

escolha popular, proporcionando eficiência e facilidade de uso no desenvolvimento web (KINSTA, 2023). Todas essas características corroboraram para a escolha da ferramenta para o projeto.

Outro ponto também levantado, foi a utilização de microsserviços para a composição da arquitetura da plataforma. Nos dias de hoje, já é conhecido pela grande maioria dos desenvolvedores as vantagens de se trabalhar com microsserviços, uma arquitetura de desenvolvimento em que uma aplicação é dividida em componentes independentes e interconectados. Essa abordagem oferece vantagens como escalabilidade mais fácil, manutenção simplificada, e *deploys* independentes. Cada microsserviço pode ser desenvolvido, implantado e escalado separadamente, promovendo agilidade, resiliência e facilitando a colaboração entre equipes. Essa arquitetura é particularmente eficaz em ambientes complexos, permitindo o desenvolvimento e aprimoramento contínuos de partes específicas do sistema (AMAZON, 2023).

E uma das ferramentas que facilitam o trabalho com microsserviços, são os contêineres, que se popularizaram com o surgimento da ferramenta *Docker*. Com base na documentação (DOCKER, 2023) *Docker* é uma plataforma de virtualização de contêineres que simplifica o desenvolvimento, distribuição e execução de aplicativos. Ele encapsula aplicativos e suas dependências em contêineres autossuficientes, garantindo consistência em diferentes ambientes. Com eficiência de recursos, possibilita a execução rápida e fácil escalabilidade de aplicações. Sua abordagem leve e portátil transforma o processo de desenvolvimento, permitindo a criação e implantação eficientes de ambientes isolados. Essa tecnologia é amplamente utilizada para desenvolvimento ágil, facilitando a colaboração entre equipes e a implementação consistente em diversas plataformas.

Figura 1 – Monolito VS Microsserviços.



Fonte: <https://aws.amazon.com/pt/microservices>, 2023

Para realizar a comunicação entre os microsserviços, foi levantado a utilização de filas de mensageria que como apresentado em (RIBEIRO, 2023) oferecem uma comunicação assíncrona eficiente entre sistemas distribuídos, melhorando a escalabilidade e a confiabilidade. Permitem o desacoplamento de componentes, possibilitando o processamento de mensagens em tempos diferentes, além de proporcionar tolerância a falhas, garantindo a entrega e processamento mesmo em cenários complexos e heterogêneos. O serviço utilizado para realizar essa

implementação foi o NATS, que é um sistema de mensagens de código aberto simples, seguro e de alto desempenho para aplicativos nativos da nuvem, e arquiteturas de microsserviços (NATS, 2023).

Por fim, como parte da infraestrutura da aplicação, foi considerado também algumas possibilidades de ferramentas para o desenvolvimento do *FrontEnd*, a camada de interação com o usuário. Dentre as opções hoje disponíveis, como *Angular*, *Ember*, *Vue* etc, foi optado pela utilização do *React JS* (FACEBOOK, 2022), mais especificamente uma variante da biblioteca o *Next JS*.

Construído sobre o *React*, oferece uma experiência aprimorada no desenvolvimento web. Combinando as vantagens do *React* com recursos adicionais, o *Next.js* simplifica a renderização, permitindo a criação eficiente de aplicações otimizadas para SEO. Sua arquitetura facilita a navegação e suporta renderização híbrida, gerando páginas estáticas ou dinâmicas conforme necessário. No mercado, essa integração do *Next.js* com *React* ganha destaque, sendo adotada por diversas empresas devido à sua eficácia no desenvolvimento ágil, proporcionando interfaces modernas, eficientes e escaláveis (NASCIMENTO, 2022).

3. Metodologia

Para conseguir chegar no resultado esperado, foi empregado o uso de metodologias ágeis para acelerar o desenvolvimento da aplicação, e permitir validar a ideia a cada pequena entrega de valor. Para isso, nos baseamos nos princípios do Manifesto Ágil abordados em (HIGHSMITH, 2009) e (SHARP, 2018).

Mais especificamente, foi planejado a construção do produto utilizando algumas estratégias da metodologia Scrum que foi derivada e teve como seu principal fundador, um dos signatários do Manifesto Ágil (Sutherland, 2014). Abaixo estão listadas algumas das estratégias utilizadas.

3.1 Backlog

Para conseguir mensurar todas as atividades que deverão ser realizadas para o sucesso do produto, foi descrito uma série de histórias (tarefas) que deverão ser concluídas ao longo do desenvolvimento, atribuindo a elas um objetivo e uma complexidade para que aquilo seja realizado. Isso é descrito pelo autor como *Backlog*, e representa basicamente o futuro do produto, e onde se quer chegar com ele.

Quadro 1 – Exemplo de Backlog utilizado.

Histórias	Complexidade
Prototipação da tela de login	Baixa
Descrever regras de negócio	Média
Criação das tabelas no banco de dados	Alta

Definição da arquitetura do projeto	Média
Desenho da logo da plataforma	Baixa

Fonte: Autores, 2023

3.2 Sprint

Foi limitado a janela de desenvolvimento das atividades baseado em *Sprints* de 15 dias. Dentro desse período, foi focado em realizar delimitadas histórias do *Backlog* baseando-se em sua complexidade para limitar o número de atividades por *Sprint*. E o critério de sucesso para cada história é que ao final do prazo isso esteja pronto para teste com o usuário final, onde no caso os usuários foram pessoas próximas que se voluntariaram para testar o sistema.

3.3 Ciclo PDCA

Como abordado anteriormente, uma das grandes vantagens em utilizar o Scrum é a possibilidade de validar as alterações feitas a cada *Sprint* juntamente com o cliente. Pois ao final de cada ciclo, uma nova funcionalidade estará disponível para ser testada, e com isso é possível obter *Feedback* do usuário se a entrega que foi feita está alinhada com as expectativas do projeto.

E para isso, um dos métodos utilizados em projetos ágeis é o ciclo PDCA (*Plan, Do, Check, Act*). Basicamente a ideia do ciclo é, conseguir planejar as alterações que precisam ser feitas no produto, realizar essas mudanças, validar com os usuários se as mudanças foram positivas ou negativas, e com isso agir para corrigir o que poderia ser melhor, e investir em continuar fazendo aquilo que deu certo.

Figura 2 – Ciclo PDCA.



Fonte: <http://www.fbsolutions.com.br/erp-totvs-protheus/o-que-e-pdca-plan-do-check-act>, 2017

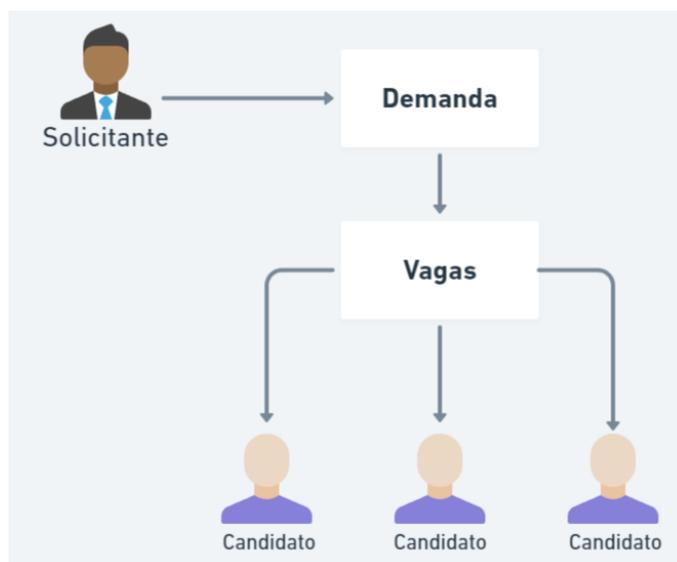
4. Análise e discussão dos resultados

Após realizar o estudo do problema, e levantar formas em que ele poderia ser resolvido, foi descrito um fluxo ideal, as entidades que deveriam compor o sistema, e qual o papel que cada uma deveria desempenhar para atingir o objetivo final. A ideia era conseguir mapear o que

a plataforma deveria possibilitar o usuário fazer, quais os requisitos mínimos deveriam existir, e quais funcionalidades precisariam ser criadas.

O fluxo pensado foi, um empreendedor possui a demanda de digitalizar o seu produto de alguma forma, e para isso precisará da ajuda de profissionais especializados que possam resolver seu problema. Não necessariamente, um mesmo problema deve ser resolvido por apenas uma pessoa, haverá casos em que a solicitação abrangerá diferentes tipos de especialização.

Figura 3 – Diagrama do fluxo principal.



Fonte: Autores, 2023

Assim, foi definido o fluxo base da aplicação, uma entidade chamada de “Solicitante”, possui uma “Demanda” a ser resolvida. Esta demanda possui um número de determinadas “Vagas” que serão preenchidas por “Candidatos” que ajudarão o solicitante com o problema descrito. Essas vagas não necessariamente serão para um mesmo papel, onde dependendo da complexidade do problema, poderão estar envolvidos profissionais com características diferentes. Como por exemplo, na construção de um *Website* personalizado, seria necessário que pelo menos um desenvolvedor e um *designer* participassem do problema.

Com isso, foi possível determinar as funcionalidades básicas da aplicação, e os requisitos funcionais que deverão existir:

- O usuário deverá conseguir se cadastrar na plataforma, e realizar o seu login com Email e senha.
- O usuário deve ter a possibilidade de incluir informações de contato no seu perfil, além de outras informações como foto, descrição etc.
- Para usuários do tipo “Candidato”, deverá ser possível adicionar qual a área de atuação (desenvolvedor, designer, publicitário etc.)
- Os usuários devem ter a possibilidade de visualizar o perfil de outros usuários.
- O sistema terá de fazer a diferenciação entre os tipos de usuário “Solicitante” e “Candidato”.
- Um “Solicitante” deve conseguir fazer o cadastro de “Demandas”, com um determinado número de “Vagas”, e realizar o aceite ou recusa dos candidatos.

-
- Um “Candidato” deverá conseguir visualizar as “Demandas” disponíveis, e se disponibilizar para atuar em uma das “Vagas” caso está ainda não esteja preenchida.
 - Uma “Demanda” deve possuir um *status* que determine se ela já foi concluída, ou se ainda está disponível para atuação.
 - Dentro de uma “Demanda” deverá ser possível adicionar uma descrição do que está sendo pedido, além de arquivos de anexo caso necessário.
 - Deverá ser possível diferenciar cada tipo de “Vaga” disponível dentro de uma “Demanda”, o tipo de vaga determinará a área de atuação exigida como competência do profissional.
 - Os usuários deverão poder realizar comentários dentro de uma “Demanda”.

Com estas definições básicas da aplicação, foi possível realizar a modelagem da arquitetura do projeto, e a divisão em microsserviços baseado nas entidades que foram criadas, mapeando o que cada um deverá gerenciar.

4.1. Microsserviços

Conforme mencionado anteriormente, para a construção do sistema, foi utilizado uma arquitetura baseada em microsserviços. Cada serviço foi criado com a mesma estrutura, utilizando a linguagem *Typescript* e com *Framework Express*. Além disso, cada um possui um *Schema* de banco de dados separado, onde o banco utilizado foi o *PostgreSQL*.

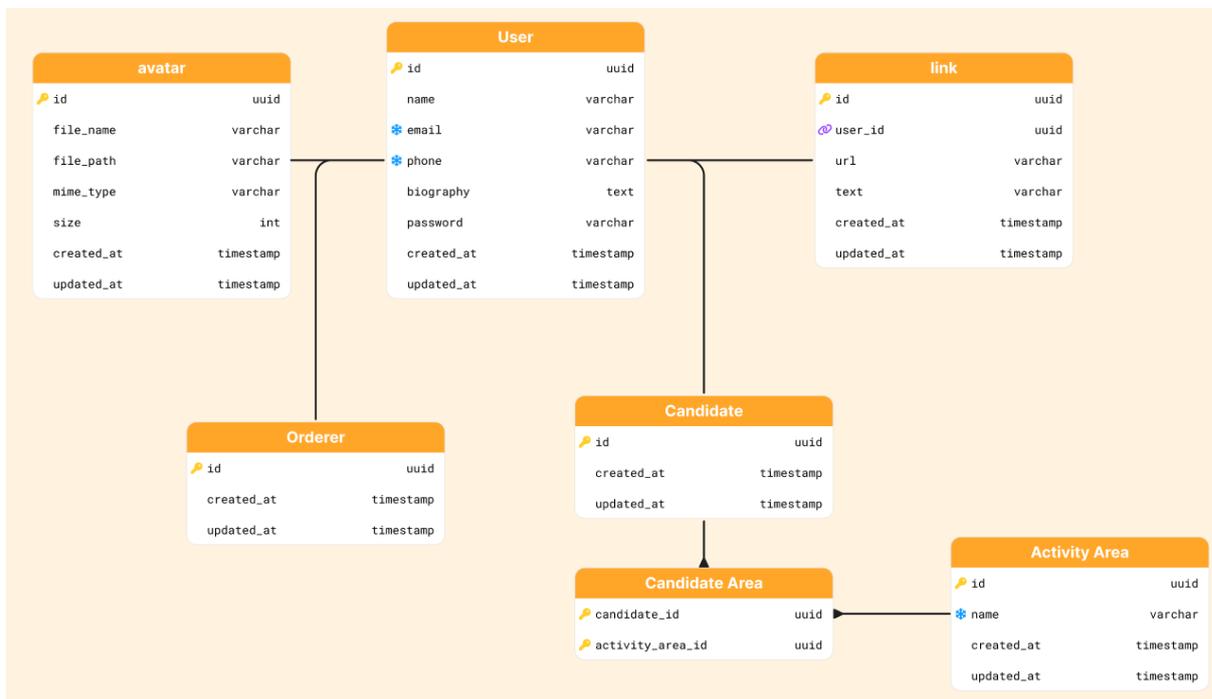
A comunicação entre os microsserviços decorre da utilização de filas de mensageria, que disparam ações a serem realizadas conforme eventos específicos acontecem, como por exemplo no cadastro de um novo usuário. Para isso foi utilizado de uma ferramenta de gerenciamento de mensagens, o NATS.

4.1.1. Microsserviço *Auth*

Responsável por realizar a autenticação do usuário na plataforma, o serviço de *Auth* foi implementado utilizando o padrão de segurança com *Access Token* e *Refresh Token*. O funcionamento desse método é basicamente o seguinte, assim que um usuário realiza a sua autenticação na plataforma por meio do login e senha, é retornado a ele um *Access Token* e um *Refresh Token* (ambos JWT), o papel do primeiro é garantir a informação de que o usuário está autenticado no sistema, e poderá seguir o fluxo de navegação livremente, porém essa autenticação tem um tempo de expiração que no padrão atual está em torno de uma hora. Assim que é feita uma requisição para o *Backend* e é verificado que o *Access Token* expirou o tempo de vida útil, entra em cena o *Refresh Token*, que possui um tempo de expiração maior hoje configurado para 15 dias. Este *token* é único por usuário e sessão e é utilizado para gerar um novo *Access Token* com um novo período de expiração, fazendo assim com que o usuário permaneça autenticado na aplicação, e garantindo a segurança de que a ninguém mais tenha este acesso.

Além da função de autenticação, o serviço de *Auth* também é responsável por armazenar as informações básicas dos usuários, foto de avatar, e fazer o gerenciamento de permissões baseado no perfil de “Solicitante” e “Candidato”. Abaixo está representado o diagrama do *schema* do serviço e seus relacionamentos:

Figura 4 – Diagrama do banco de dados *Auth*.

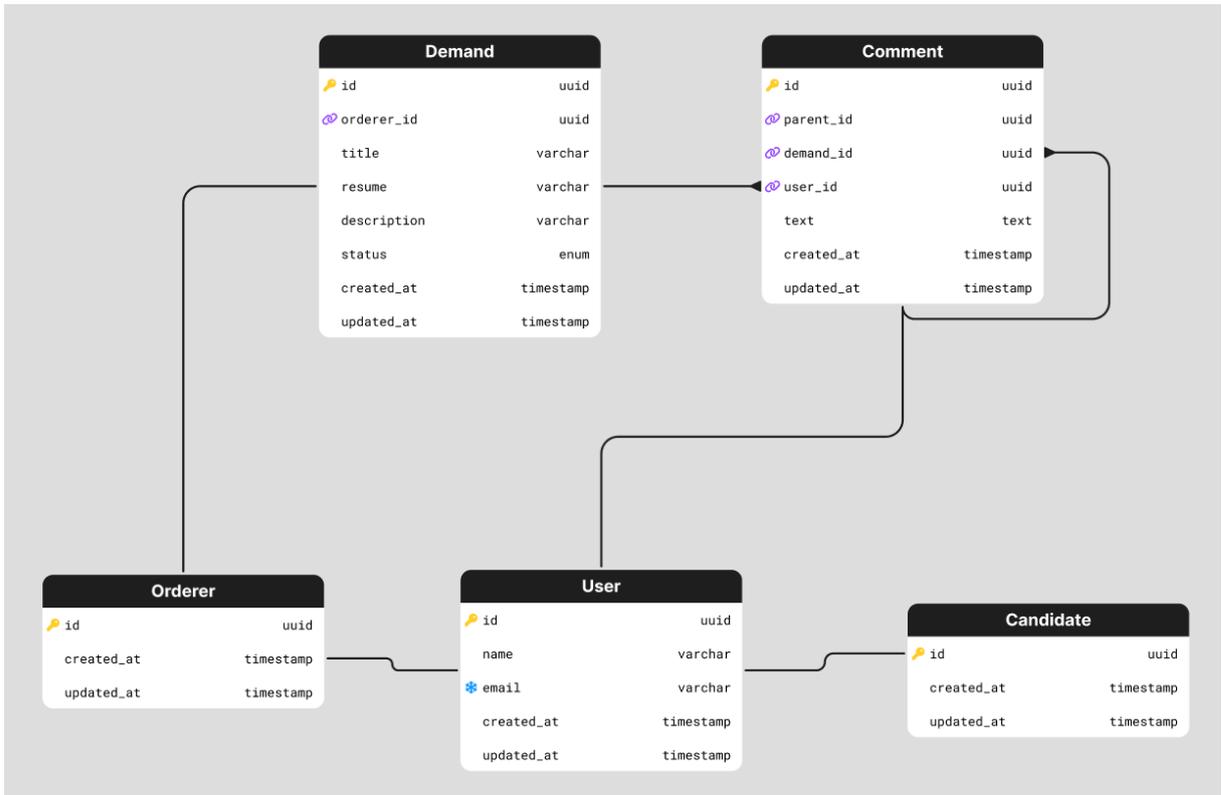


Fonte: Autores, 2023

4.1.2. Microserviço *Demand*

O microserviço *Demand* é responsável por fazer o gerenciamento de todas as “Demandas” da plataforma. É por meio dele que se torna possível realizar o cadastro de uma nova solicitação, e manter ela atualizada podendo realizar edições, além de também possibilitar a inserção de comentários, e fazer a manipulação das vagas disponíveis para a demanda. Abaixo está representado o diagrama do *schema* do serviço e seus relacionamentos:

Figura 5 – Diagrama do banco de dados *Demand*.

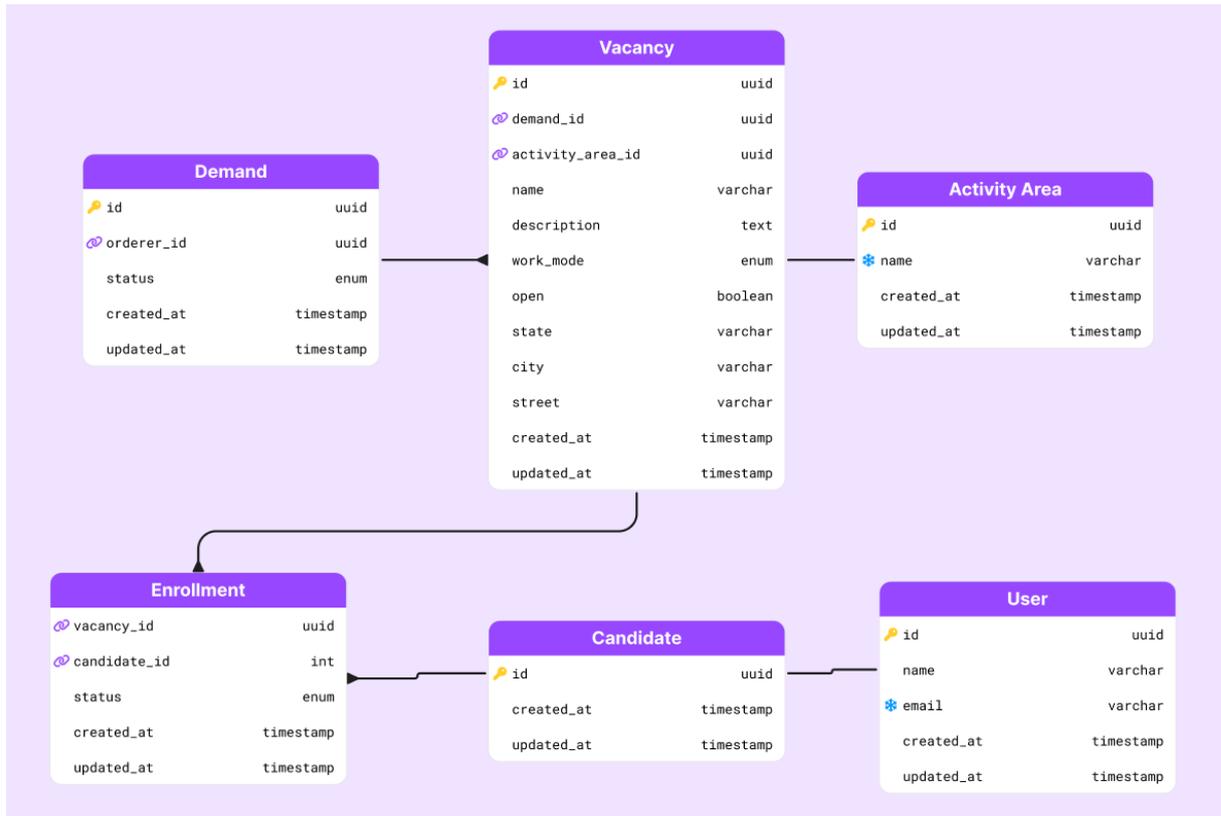


Fonte: Autores, 2023

4.1.3. Microserviço *Vacancy*

E por último temos o serviço responsável por realizar o cadastro dos tipos de vagas disponíveis, onde serão inseridas informações como o a área de atuação daquela vaga, se há algum candidato já inscrito, qual a modalidade de atuação, remoto ou presencial, e caso seja presencial, o endereço do local. Abaixo está representado o diagrama do *schema* do serviço e seus relacionamentos:

Figura 6 – Diagrama do banco de dados *Vacancy*.



Fonte: Autores, 2023

4.2. Camada de interação com o usuário

Uma preocupação na hora de criar a parte visual do projeto, foi em relação a acessibilidade da plataforma. Pensando nisso, foi realizado todo o design e desenvolvimento focando na ideia de *Mobile First*, que se resume em basicamente desenhar as telas para dispositivos móveis primeiro, e posteriormente adaptar o *layout* para resoluções maiores.

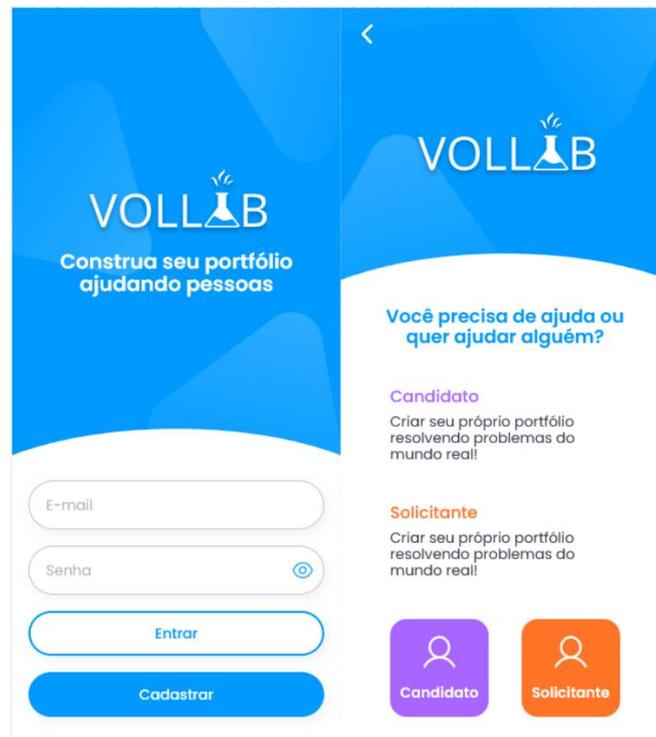
A aplicação *Frontend* foi toda desenvolvida utilizando o *Framework Nest.js* e assim como no *Backend* também foi utilizado o *Typescript* como linguagem. Além disso, foi utilizado algumas bibliotecas que auxiliaram na parte de animações, e interação com o servidor através da arquitetura REST.

Para dar mais personalidade ao projeto, foi criado uma identidade visual, juntamente com a marca *Vollab*, contendo também uma logo personalizada.

4.2.1. Login e cadastro de usuários

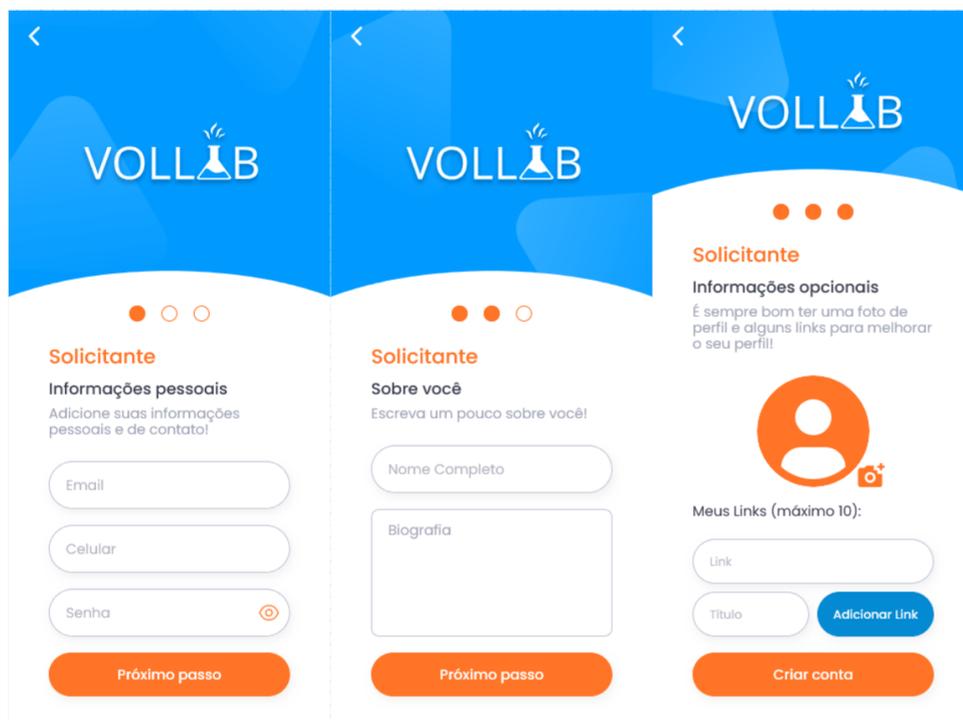
A seguir, algumas representações da tela de login e cadastros de usuários na aplicação, nos dois perfis disponíveis, solicitantes e candidatos.

Figura 7 – Telas de login.



Fonte: Autores, 2023

Figura 8 – Cadastro de usuário Solicitante.

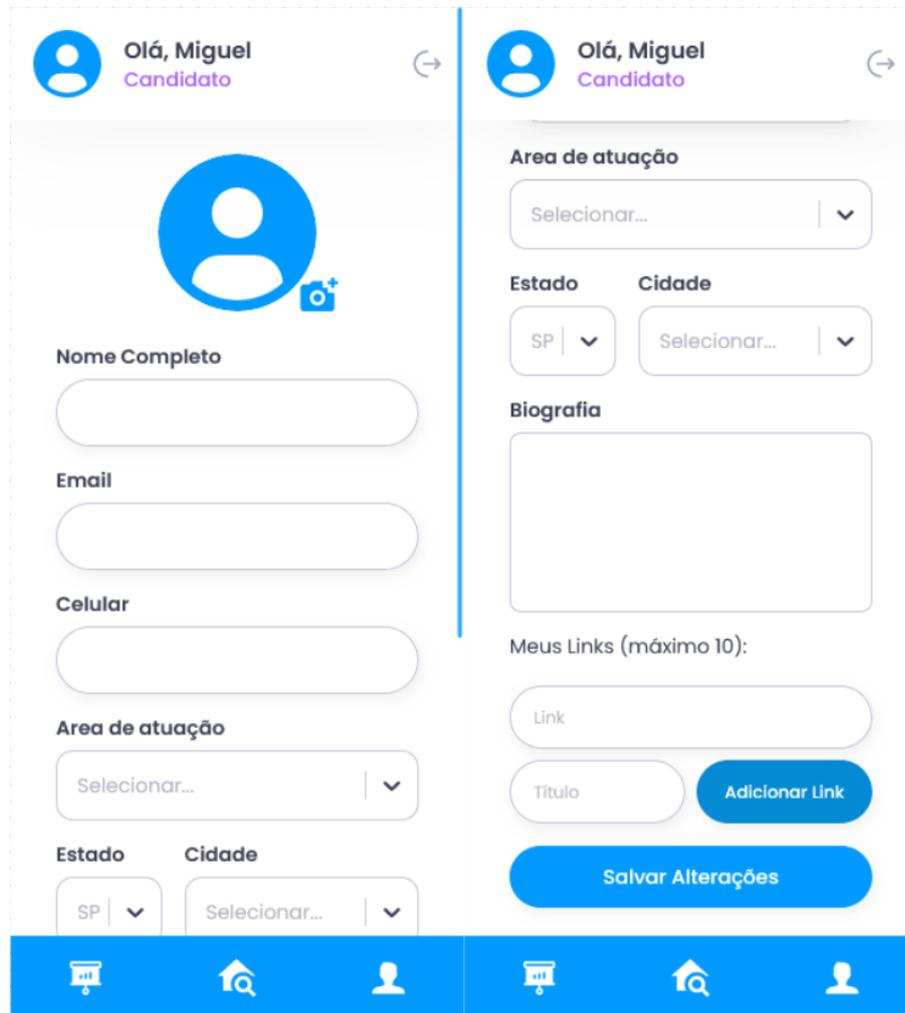


Fonte: Autores, 2023

4.2.2. Perfil de usuário

Telas de visualização de perfil de usuário, e edição de informações.

Figura 9 – Perfil de usuário.

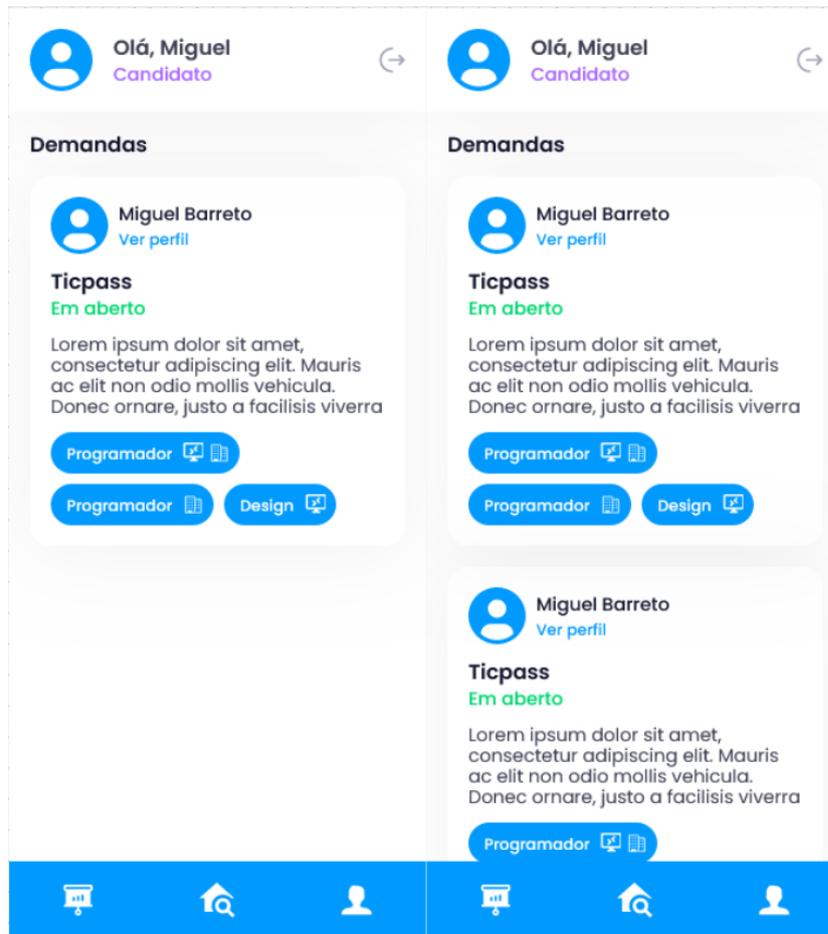


Fonte: Autores, 2023

4.2.3. Listagem de demandas

Exemplos de demandas disponíveis para um usuário com o perfil “Candidato”.

Figura 10 – Listagem de demandas.



Fonte: Autores, 2023

4.3. Código fonte

O código fonte da aplicação está disponibilizado publicamente em uma plataforma, e pode ser acessado através do link <https://github.com/orgs/Vollab/repositories>.

5. Considerações finais

Diante do exposto, a construção e implementação da plataforma digital proposta demonstra ser uma solução promissora para mitigar os desafios enfrentados pelos empreendedores menos favorecidos e pelos profissionais iniciantes na indústria 4.0. O avanço tecnológico, especialmente a migração para o mundo digital, impôs uma barreira para aqueles que não possuem os recursos financeiros ou o conhecimento necessário para adentrar esse universo altamente competitivo.

A utilização de tecnologias recentes na criação da plataforma representa um passo significativo para garantir eficiência, e escalabilidade. Tendo sua infraestrutura pautada com o uso de microsserviços e processamento assíncrono de eventos com filas de mensageria, conseguimos garantir uma estrutura robusta, e com alta capacidade de crescimento. Além disso, há a construção de um design centrado no usuário de celular, reconhecendo a predominância desse

dispositivo entre o público-alvo. A interface intuitiva e responsiva visa proporcionar uma experiência otimizada, facilitando a acessibilidade e atendendo às necessidades do público-alvo que, muitas vezes, não dispõe de equipamentos robustos.

Há também possibilidades de inserir melhorias que podem ampliar a porcentagem de sucesso do produto, como por exemplo a introdução de um chat online para os usuários, sistemas de postagens como as de um blog, inclusão de mais informações nos perfis dos candidatos, melhorias no algoritmo de busca de demandas, personalização de filtros de pesquisa, e demais alterações. Podendo ser validadas com os usuários ativos, medir o engajamento com as novas funcionalidades, e replanejar novas mudanças, seguindo os princípios apresentados do PDCA.

A proposta de conectar empreendedores necessitados de digitalização com profissionais em busca de experiência é uma abordagem que visa a promoção de um ambiente colaborativo e inclusivo. Ao eliminar a barreira financeira e facilitar a entrada de novos talentos no mercado, a plataforma cria um ciclo positivo de benefícios mútuos. A resolução de problemas reais proporciona aos profissionais iniciantes uma oportunidade valiosa de comprovação de experiência, preenchendo uma lacuna que muitas vezes impede a inserção no mercado de trabalho.

Assim, a plataforma não apenas atende à demanda de ambos os públicos-alvo, mas também contribui para reduzir a desigualdade digital ao oferecer oportunidades igualitárias de crescimento. Em um cenário onde a tecnologia pode ser uma ferramenta poderosa para inclusão, a iniciativa proposta se destaca como um modelo de como a inovação pode ser aplicada para criar um impacto positivo e sustentável na sociedade.

Referências

AMAZON. **O que são microsserviços?** Disponível em: <<https://aws.amazon.com/pt/micro-services>>. Acesso em: 10 nov. 2023.

DEMO, P. **INCLUSÃO DIGITAL. Cada vez mais no centro da inclusão social. Inclusão Social**, [S. l.], v. 1, n. 1, 2005. Disponível em: <https://revista.ibict.br/inclusao/article/view/1504>. Acesso em: 15 out. 2022.

DOCKER. **Overview of the get started guide**. Disponível em: <<https://docs.docker.com/get-started/>>. Acesso em: 5 nov. 2023.

FACEBOOK. **React**. React.Org, 2022, Disponível em: <<https://pt-br.reactjs.org>>. Acesso em: 10 nov. 2022.

FB Solutions. **O que é PDCA – PLAN-DO-CHECK-ACT. 2021. Imagem no formato PNG disponível no site**. Disponível em: <<http://www.fbsolutions.com.br/erp-totvs-protheus/o-que-e-pdca-plan-do-check-act/>>. Acesso em: 1 de dezembro de 2023

FORTUS, David, et al. **Design - based science and real - world problem - solving**. International Journal of Science Education 27.7 (2005): 855-879.

HIGHSMITH, Jim. **Agile project management: creating innovative products**. Pearson education, 2009.

JAVINPAUL. **Top 5 Programming languages for Web development in 2022**. Medium, 2021, Disponível em: <<https://medium.com/javarevisited/top-5-programming-languages-for-web-development-in-2021-f6fd4f564eb6>>. Acesso em: 30 nov. 2022.

KAMARUZAMAN, F. MOHD, et al. **Comparison of Engineering Skills with IR 4.0 Skills**. International Journal of Online Engineering 15.10 (2019).

KINSTA. **O que é Express.js? Tudo o Que Você Precisa Saber**. 2023, Disponível em: <<https://kinsta.com/pt/base-de-conhecimento/o-que-e-express-js/>>. Acesso em: 10 nov 2023

LAST, Heiner, et al. **Industry 4.0**. Business & information systems engineering 6.4 (2014): 239-242.

MDN Contributors. **JavaScript**. MDN. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 10 nov. 2022.

MENHORN, Benjamin; SLOMKA, Frank. **Confirming the design gap. In: Advances in Computational Science, Engineering and Information Technology**. Springer, Heidelberg, 2013. p. 281-292.

MICROSOFT. **TypeScript**. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 10 nov. 2023

MORRAR, Rabeh, HUSAM Arman, and SAEED Mousa. **The fourth industrial revolution (Industry 4.0): A social innovation perspective**. Technology Innovation Management Review 7.11 (2017): 12-20.

NASCIMENTO, Felipe. **NextJS: por que usar?** Disponível em: <<https://www.alura.com.br/artigos/next-js-vantagens>>. Acesso em: 10 nov. 2023

NATS Authors. **NATS**. Disponível em: <<https://nats.io/>>. Acesso em: 1 dez. 2023

NODE.JS contributors. **NodeJs**. NodeJs.Org, Disponível em: <<https://nodejs.org/en>>. Acesso em: 12 nov. 2022.

RIBEIRO, Guilherme. **Porque você deveria utilizar mensageria na sua aplicação**. Disponível em: <<https://www.dinamize.com.br/blog/por-que-voce-deveria-utilizar-mensageria-na-aplicacao>>. Acesso em: 10 nov. 2023

ROSA, Renato et al. **The Importance of Digital Marketing Using the Influence of Consumer Behavior**. REVISTA DE TECNOLOGIA APLICADA (RTA) v.6, n.2, (2017): 28-39.

SHARP, Jason H.; LANG, Guido. **Agile in teaching and learning: Conceptual framework and research agenda**. Journal of Information Systems Education, v. 29, n. 2, p. 45-52, 2018.

SUTHERLAND, Jeff. **Scrum: A arte de fazer o dobro do trabalho na metade do tempo**. Leya, 2014.

TYPESCRIPT, Logo. **Arquivo de imagem no formato SVG**. Disponível em: <https://pt.m.wikipedia.org/wiki/Ficheiro:Typescript_logo_2020.svg>. Acesso em: 1 de dez. de 2023

VAIDYA, Saurabh et al. **Industry 4.0 –a glimpse**. Procedia Manufacturing 20 (2018): 233-238.

WANG, Dong et al. **The age of social sensing**. Computer, v. 52, n. 1, p. 36-45, 2019.