

Conectivo Conecta

Arthur Gramiscelli

Guilherme Campos Barata Diniz

Leonardo Antunes Barreto Noman

Lucas Guilherme Verdán Moreira

Weber Marques de Oliveira

**Instituto de Informática e Ciências Exatas– Pontifícia Universidade de Minas Gerais
(PUC MINAS)
Belo Horizonte – MG – Brasil**

agbranco@sga.pucminas.br

guicbdeniz@hotmail.com

leonoman07@gmail.com

lgvmoreira@sga.pucminas.br

weber.marques@sga.pucminas.br

Resumo

Conectivo é uma Organização da Sociedade Civil da comunidade da Ventosa com o objetivo de solucionar problemas reais da população local. Um de seus desafios é de interligar oportunidades de emprego com moradores da comunidade, transformando positivamente a vida do empregado. Para isso, é necessária uma plataforma que conecte vagas com trabalhadores disponíveis e aptos. Este trabalho tem, portanto, como objetivo, gerar uma aplicação móvel que conecte prestadores de serviços da comunidade com possíveis empregadores.

1 Introdução

1.1 Contextualização

O Trabalho Interdisciplinar de Software IV é uma disciplina que tem como propósito fazer uma ligação concreta entre as matérias práticas e teóricas ensinadas pelo curso de Engenharia de Software, PUC Minas. É uma matéria extensionista,

ou seja, tem como finalidade resolução de problemas existentes de contexto social usando o conhecimento adquirido pelo ensino e pesquisa dentro da Universidade.

Este trabalho está sendo desenvolvido no segundo semestre de 2020, ano em que a pandemia causada pelo COVID-19 modificou de forma extrema o ritmo de vida de toda população mundial. O isolamento social como medida de segurança afetou negativamente tanto a qualidade de vida das pessoas pela necessidade da socialização como também pela retração econômica devido a mesma. (Douglas, M. 2020).

Um dos pontos positivos da pandemia foi o aumento das doações de cestas básicas por instituições públicas e privadas em Belo Horizonte. Com isso veio a necessidade de organizar de forma mais sistemática quem e quanto cada família já recebeu, para que tivesse uma distribuição equitativa. Surgiu aí o Conectivo, criado por Jimmy e Francisco com intuito de ajudar a comunidade da Ventosa em Belo Horizonte.

O Conectivo é uma Organização da Sociedade Civil que tem como objetivo impactar positivamente a comunidade da Ventosa. Usando a Teoria da Mudança como meio estratégico para mapear o desenvolvimento socioeconômico daqueles que lá vivem. Esse método foca nas entradas e saídas do sistema, e usa de métricas objetivas para entender o que se passa pela comunidade.

1.2 Problema

Uma das soluções propostas pelo nosso parceiro foi desenvolver um aplicativo que funcione de forma semelhante ao LinkedIn, uma rede social de empregos. O aplicativo permitirá que empresas parceiras ao projeto registre vagas de emprego por lá, e que os moradores locais se candidatem para essas vagas, com seus respectivos perfis indicando seu histórico profissional.

1.3 Objetivo Geral

Desenvolver uma aplicação robusta que permita integrar de forma ágil oportunidades de emprego àqueles que o procuram.

1.4 Objetivos Específicos

- Utilizar dos conhecimentos dos participantes adquiridos na matéria de Interface Humano Computador para criar uma aplicação mobile com interface clara e de alto nível de usabilidade;
- Desenvolver um código de qualidade de acordo com os princípios SOLID;
- Fazer o levantamento de requisitos para que todas as funcionalidades demandadas sejam atendidos;
- Adquirir conhecimento sobre todas as etapas do processo de desenvolvimento de software na prática.

1.5 Justificativa

Em Julho de 2020 foi relatado pela CDL (Câmara de Dirigentes Lojistas) e da Abrasel (Associação Brasileira de Bares e Restaurantes) que o número de desempregados em Belo Horizonte chegaria em até 140 mil devido ao fechamento do comércio (D'Oliveira, Rafael. 2020), medida de prevenção protocolada pelo prefeito Alexandre Kalil.

Como observado pelos dados compilados pela Our World in Data (Roser, M, 2020), o Brasil foi um dos países mais frágeis em relação ao espalhamento do COVID-19. Sabe-se de suas consequências médicas e suas inevitáveis complicações sociais de curto e médio prazo. Uma das formas de combater o desemprego de alta volatilidade é justamente fazendo uma intermediação entre os interessados em empregar e aqueles que estão procurando urgentemente um novo emprego.

2 Referencial Teórico

Neste projeto foi utilizado de diferentes ferramentas das diferentes áreas do conhecimento. Das tecnologias de desenvolvimento: um framework, um compilador, uma metodologia ágil, um padrão de projeto, um banco de dados e duas linguagens de

programação. Estas estão descritas abaixo. Das tecnologias de comunicação e organização: WhatsApp, Discord, Google Drive, Github, Trello e Microsoft Teams.

2.1 React Native

Para montar a interface de usuário decidiu-se utilizar do React Native. Como citado em sua documentação oficial “React Native é um framework de código aberto para construção de aplicações Android e iOS, usando React e capacidades nativas da plataforma trabalhada.” (React Native Docs, 2020, traduzido). Com ele, é possível utilizar JavaScript para descrever a aparência e comportamento da interface de usuário da aplicação.

Native utiliza conceitos de *view* descrito em “Em desenvolvimento Android e iOS, um **view** é um bloco básico de construção de uma IU: um pequeno elemento retangular na tela que pode ser usado para expor textos, imagens ou para responder entradas de usuários” (React Native Docs, 2020, traduzido).

2.2 Desenvolvimento Ágil

Para organizar, planejar e alocar melhor as demandas e os recursos, será utilizado o processo de desenvolvimento ágil, popularizado pelo manifesto ágil. Seus princípios são: “Através deste trabalho, passamos a valorizar: Indivíduos e interações mais que processos e ferramentas”, “Software em funcionamento mais que documentação abrangente”, “Colaboração com o cliente mais que negociação de contratos”, “Responder a mudanças mais que seguir um plano” (Beck, K. 2001).

2.3 Transferência Representacional de Estado (REST)

Para manter uma boa comunicação entre a aplicação desenvolvida e o sistema gerenciador de banco de dados utilizado, foi utilizado de uma interface de programação de aplicações (API, em inglês) RESTful. Richard Taylor cita REST como “um conjunto de restrições que tentam minimizar a latência e a comunicação de rede, ao mesmo tempo maximizando a independência e a escalabilidade dos componentes implementados.” (Taylor, R. 2002, traduzido).

2.4 Node.js

Foi utilizado do Node.js, um ambiente de execução de Javascript “arquitetado para construir aplicações de rede escaláveis” (About Node.js, 2020, traduzido) para desenvolver a API de conexão com o banco de dados da aplicação.

2.5 TypeScript

Foi utilizado do TypeScript, “uma linguagem de código aberto que constrói códigos JavaScript, (...), adicionando a ele definições estáticas de tipos” (What is Typescript, traduzido), para minimizar erros em tempo de execução.

Em 2017 foi publicado um artigo relacionado a tipagem de JavaScript e foi concluído que “(...) nossa descoberta central é de que ambos sistemas de tipagem estáticos (estudados) encontram uma porcentagem importante de bugs públicos: tanto Flow 0.30 quanto TypeScript 2.0 detectaram 15%!” (Gao, Z. 2017).

2.6 MySQL

Para persistência de dados, foi utilizado o sistema de gerenciamento de banco de dados MySQL. Como é exposto em suas documentações oficiais, “MySQL, o sistema de gerenciamento de banco de dados SQL de código aberto mais popular do mundo, é desenvolvido, distribuído e apoiado pela Oracle Corporation” (What is MySQL, traduzido, 2020). Todos os bancos de dados do MySQL são relacionais, descrito na documentação como “um banco de dados relacional armazena dados em tabelas separadas ao invés de salvar todos os dados em um grande depósito.” (MySQL databases are relational, traduzido).

2.7 Extensão

Como uma disciplina extensionista este trabalho propõe unir os conhecimentos teóricos aprendidos na faculdade nas áreas de ensino e pesquisa para solucionar algum

problema social. Tem como objetivo levar para fora da academia desenvolvimento prático para alguma parte da sociedade, juntamente com um desenvolvimento cidadão.

3. Metodologia

Quanto aos objetivos o tipo de pesquisa utilizada foi exploratória. O objetivo deste trabalho foi desenvolver uma aplicação que a ideia já existia mas esta foi otimizada para o fim específico do cliente. É também uma pesquisa qualitativa uma vez que procura atender uma demanda que não manifesta por meios objetivos dos números, é uma demanda social em os que os resultados são mais informativos sobre o problema.

A primeira etapa do trabalho foi fazer uma entrevista com o cliente. Todos cinco integrantes do projeto participaram ativamente das entrevistas a fim de melhor explorar nossas dúvidas sobre este, e, conjuntamente ao cliente, propor uma solução adequada para sua demanda. A segunda etapa foi basear o planejamento pelo Trello utilizando do Scrumban como metodologia geral. A terceira etapa foi o desenvolvimento prático do sistema utilizando das ferramentas citadas no referencial teórico (React Native, TypeScript, MySQL, Node.js). Adiciono aqui o uso do Git e GitHub para versionamento do código, cada branch do projeto foi baseada no número do card do Trello. A quarta etapa foi apresentar o projeto ao cliente e aos avaliadores. Todas essas etapas se repetiram em cada sprint (iteração).

Os stakeholders deste projeto são: Francisco e Jimmy por parte do Conectivo, os desenvolvedores integrantes deste projeto, e a universidade PUC Minas. O Conectivo solicitou a construção deste projeto à PUC Minas. Isso foi possível pelo fato da Universidade ser um local de ensino, pesquisa e extensão. Este projeto representa justamente esses três pontos. Os desenvolvedores participaram tanto para melhorar as habilidades técnicas quanto para ajudar de forma concreta uma parte carente da sociedade.

4. Resultados

4.1 Requisitos

4.1.1 Requisitos Funcionais

A tabela 1 mostra nossos requisitos funcionais, seu número de ordem, descrição e prioridade. Os requisitos funcionais são aqueles que definem uma função de um sistema de software ou seu componente.

Tabela 1. Requisitos Funcionais

Número de Ordem	Requisito	Descrição	Prioridade
1	Cadastrar perfil.	O usuário cadastra perfil.	Alta
2	Deletar perfil.	O usuário deleta perfil.	Média
3	Atualizar perfil.	O usuário atualiza perfil.	Alta
4	Visualizar perfil.	O usuário visualiza perfil de outros usuários.	Média
5	Fazer login.	O usuário faz login com seu perfil.	Alta
6	Enviar mensagem de contratação.	O sistema envia mensagens de contratação.	Alta
7	Cadastrar mensagem padrão de contratação.	O administrador cadastra mensagem padrão de contratação.	Média
8	Solicitar aceitação de termos de uso.	O sistema solicita aceitação de termos de uso.	Alta
9	Cadastrar termo de uso.	O administrador cadastra termos de uso.	Alta
10	Enviar prova de existência.	O empregador usuário envia prova de existência através de algum documento.	Alta
11	Verificar existência de empresas cadastradas.	O administrador verifica existência da empresa cadastrada.	Alta
12	Criar oferta de emprego.	O empregador usuário cria oferta de emprego.	Alta
13	Registrar em vaga de emprego.	O empregado usuário registra em vaga de emprego.	Alta
14	Enviar mensagem de vaga.	O sistema envia mensagens de vaga.	Alta

15	Gerar relatórios de vagas.	O empregador usuário gera relatório de empregados interessados.	Média
----	----------------------------	---	-------

4.1.2 Requisitos Não Funcionais

Na tabela 2 estão descritos nossos requisitos não funcionais. Estes são os requisitos que dizem respeito a como os requisitos funcionais serão entregues.

Tabela 2. Requisitos Não Funcionais

Número de Ordem	Requisito	Descrição	Prioridade
1	Ser disponível online.	O sistema deve sempre estar disponível na Internet.	Alta
2	Limitar permissões a informações	Informações administrativas devem ser limitadas a permissão de cada usuário.	Alta
3	Ser disponível em aplicações nativas de celular android.	O sistema deve estar disponível para uso em aparelhos celulares com sistema android.	Alta

4.2 Diagrama de Classe

A figura 1 mostra o diagrama de classe de nossa aplicação, uma representação da estrutura e relações das classes que servem de modelo para objetos. Esta é uma representação da Unified Modeling Language (UML).

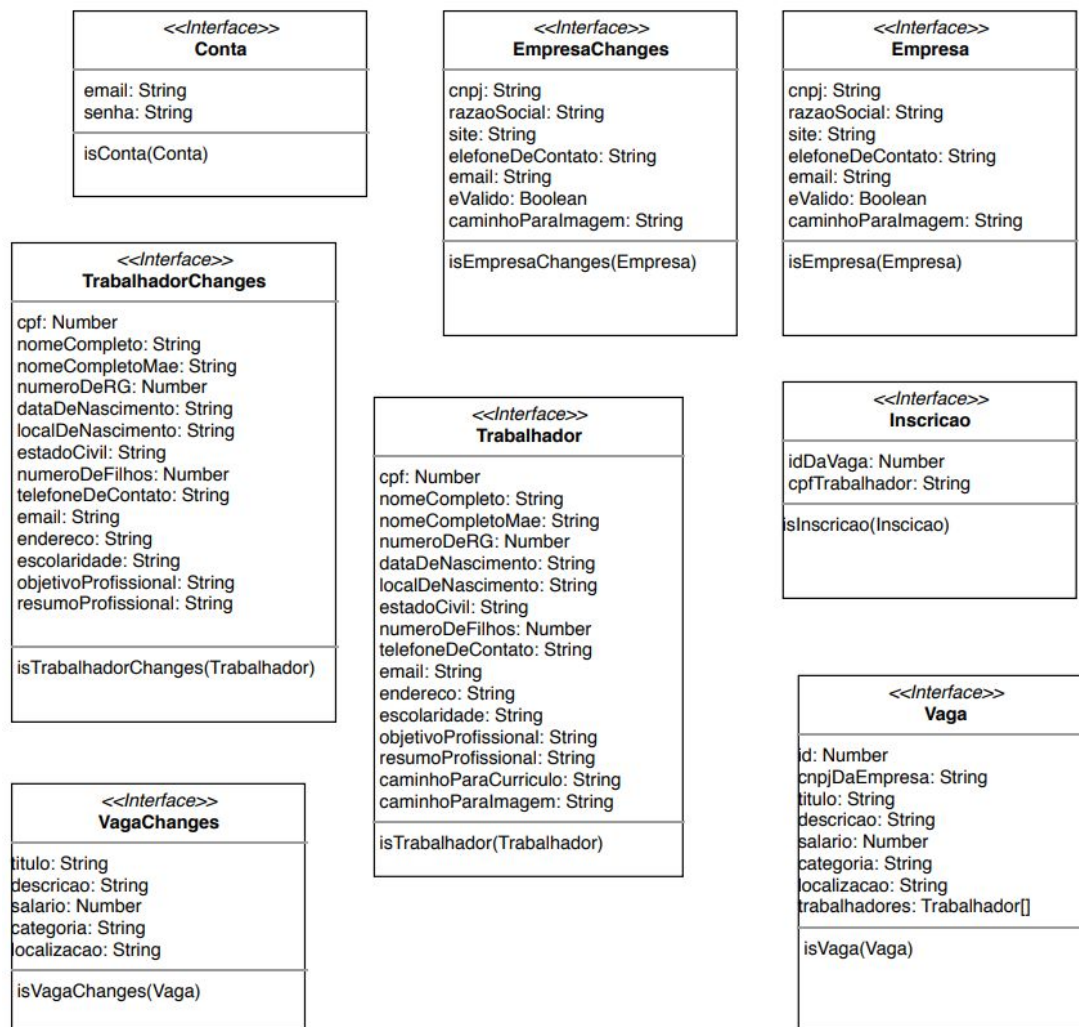


Figura 1. Diagrama de Classe

4.3 Modelo Relacional do Banco de Dados

Na figura 2 está representado nosso modelo relacional é um modelo de dados representativo (ou de implementação), adequado a ser o modelo subjacente de um Sistema Gerenciador de Banco de Dados (SGBD), que se baseia no princípio de que todos os dados estão armazenados em tabelas (ou, matematicamente falando, relações).

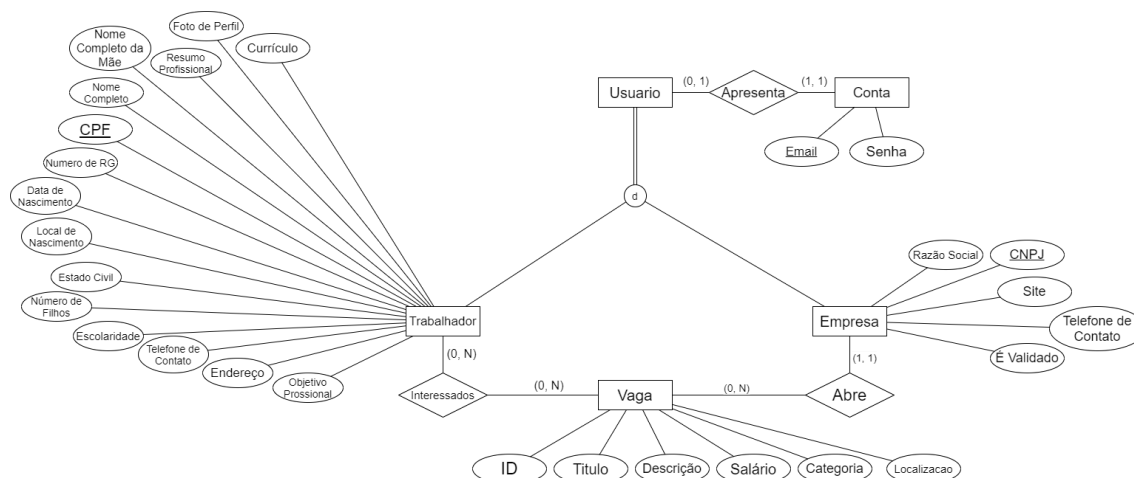


Figura 2. Modelo DER

4.4 Arquitetura do Software

4.4.1 Organização das camadas

Model e Control - Geridas pelo BackEnd. Cada funcionalidade apresenta uma pasta de controle e uma pasta de modelo.

View - FrontEnd. Através dos componentes React.

4.4.2 Softwares Utilizados

4.4.2.1 Linguagens

JavaScript.

4.4.2.2 IDE

Visual Studio Code

4.4.3 Frameworks

4.4.3.1 Frontend

React.js

4.4.3.2 Backend

Node.js

4.4.4 SGBD

MySQL

4.4.5 Controle versão

Git/GitHub

4.4.6 Outros softwares

Comunicação: Discord, Microsoft Teams.
--

Diagramas: Draw.io.

Design: Figma.

Organização: Trello, Google Drive.

4.4.7 Outras informações

Docker para integrar SGBD, Frontend e Backend.
--

4.5 Artefatos

As seguintes imagens mostram nossa aplicação. A figura 3 representa nossa tela de login. Tela onde o usuário pode entrar dentro do sistema ou escolher para se registrar. Na tela de registro de trabalhador, figura 4, o usuário preenche os campos com as informações requeridas para se cadastrar. Na tela de criação de vaga, figura 5, o usuário (empresa) preenche as informações para criar uma vaga onde o trabalhador poderá se cadastrar. A figura 6 são essas vagas em listagem. A figura 7 é a visualização de uma vaga. A figura 8 mostra uma tela onde o usuário consegue visualizar as informações de seu perfil e, se assim desejar, editá-las.

17:15 [notification icons]

[signal icons] [battery 80%]

Conecta

Email

Senha

Tipo de Conta

Selecione aqui ▼

CONFIRMAR

REGISTRAR

[android navigation bar]

Figura 3. Tela de Login



Figura 6. Tela de Visualização das Vagas



Figura 7. Tela Visualizando Vaga

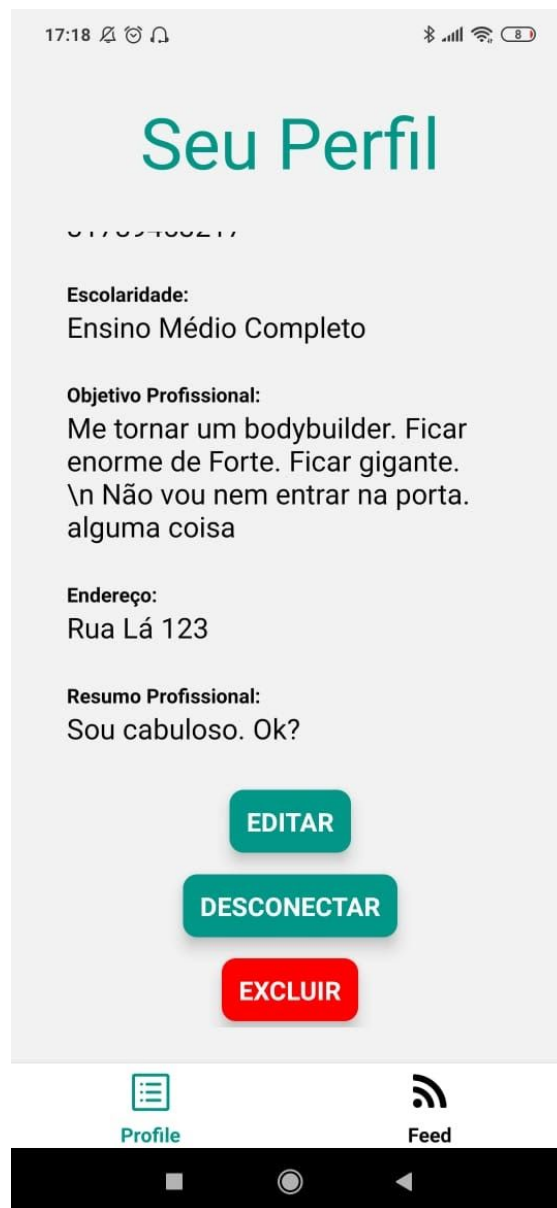


Figura 8. Tela de Perfil do Usuário

5. Conclusões e trabalhos futuros

Como proposta inicial o trabalho tinha o objetivo de se criar um aplicativo que pudesse rodar na grande maioria dos smartphones, criando assim uma maior acessibilidade e abrangência do uso do mesmo, visto que seria mais utilizado em áreas cuja a renda monetária das pessoas de um modo geral não é grande. O software em si, foi formulado em conjunto com o cliente responsável, com o objetivo de criar uma conexão entre empresas da área, interessadas em trabalhadores, e pessoas da comunidade que tem como objetivo adquirir um trabalho. Por meio de um fluxo de cadastro de vagas disponíveis por parte do contratante, e a visualização e cadastro nessas vagas por parte dos moradores da comunidade, o software iria fazer esse link entre as duas partes. A partir daí, as empresas contratantes teriam acesso ao contato e perfil dos usuários dentro da rede, e por meio disso poderia fazer a seleção e contratação dos mesmos.

O que foi desenvolvido e proposto para o trabalho do TIS, seria a entrega de todo esse fluxo de cadastro descrito acima, o que foi entregue. Para o futuro, é previsto e tem se como objetivo, o deploy dessa aplicação, para que assim se tenha um real acesso dos usuários a mesma, além de que, fazer também um site referente a esse aplicativo. Também foi proposto pelo cliente, o desenvolvimento de outros projetos que tem relação com esse App futuramente, projetos como um site de cursos de especialização que daria diplomas que seriam válidos tanto nessa nova plataforma que foi criado com em outras também.

Link do vídeo:

<https://www.youtube.com/watch?v=JRz5b5YVDzo&feature=youtu.be>

Link do repositório:

<https://github.com/icei-pucminas/plf-es-2020-2-tiiv-8499100-conectivo>

Referências

Beke, K. (2001). Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 24 ago. 2020.

Douglas, Margaret. BMJ 2020;369:m1557. Acesso em: 24 de Agosto de 2020.

Gao, Z. *et al.*, (2017) "To Type or Not to Type: Quantifying Detectable Bugs in JavaScript," 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), Buenos Aires, pp. 758-769, doi: 10.1109/ICSE.2017.75.

D'Oliveira, Rafael. Fechamento do comércio pode deixar 140 mil desempregados em BH. BHAZ, 2020. Disponível em: <https://bhaz.com.br/2020/07/16/fechamento-comercio-desempregados-bh/>. Acesso em: 24 de Agosto de 2020.

React Native Docs. Disponível em: <https://reactnative.dev/docs/intro-react-native-components>. Acesso em: 24 ago. 2020.

Node.js About Docs. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 24 ago. 2020.

Roser, Max *et al.*, (2020) - "Coronavirus Pandemic (COVID-19)". Publicado online em OurWorldInData.org. <https://ourworldindata.org/coronavirus>

Taylor, R. (2002) “Principled design of the modern web architecture”, ACM Trans. Internet Technol. (TOIT) 2, (2), 115–150.

Typescript What is?. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 24 ago. 2020.

What is MySQL?. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. Acesso em: 25 ago. 2020.