

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**IMPLEMENTAÇÃO DE SOLUÇÃO SISTÊMICA PARA GESTÃO DE DADOS
EM PESQUISA CIENTÍFICA: PROJETO INTEGRADO SOBRE OS
IMPACTOS DAS MEDIDAS DE ISOLAMENTO E DISTANCIAMENTO
SOCIAL PELA COVID-19**

Victor Lima Veiga

Teresópolis/ RJ
Novembro, 2021

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**IMPLEMENTAÇÃO DE SOLUÇÃO SISTÊMICA PARA GESTÃO DE DADOS
EM PESQUISA CIENTÍFICA: PROJETO INTEGRADO SOBRE OS
IMPACTOS DAS MEDIDAS DE ISOLAMENTO E DISTANCIAMENTO
SOCIAL PELA COVID-19**

Victor Lima Veiga

Trabalho de Conclusão de Curso apresentado ao
Centro Universitário Serra dos Órgãos como
requisito obrigatório para obtenção do título de
Bacharel em CIÊNCIA DA COMPUTAÇÃO.
Orientador(a): Luciana Nunes Ferreira

Teresópolis/ RJ
Novembro, 2021

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**IMPLEMENTAÇÃO DE SOLUÇÃO SISTÊMICA PARA GESTÃO DE DADOS
EM PESQUISA CIENTÍFICA: PROJETO INTEGRADO SOBRE OS
IMPACTOS DAS MEDIDAS DE ISOLAMENTO E DISTANCIAMENTO
SOCIAL PELA COVID-19**

Victor Lima Veiga

Trabalho de Conclusão de Curso aprovado como requisito parcial para obtenção do título de Bacharel em Ciência da Computação no Centro Universitário Serra dos Órgãos pela banca examinadora:

Luciana Nunes Ferreira - professor auxiliar e assessora pedagógica, DEaD-Unifeso

Cátia Araujo Farias - professora titular e assessora pedagógica, DEaD-Unifeso

Roberta Amaral – Diretora da DPPe

Teresópolis / RJ
09 de novembro de 2021

Dedicatória

Dedico este projeto de trabalho de conclusão de curso a minha família e aos amigos que me deram apoio para a conclusão do mesmo.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me deu forças para que eu pudesse concluir esse projeto. Sou grato a minha orientadora Luciana Nunes pela orientação deste trabalho, por todas as dicas, pela paciência e por me ajudado a focar totalmente na solução. Também tenho uma enorme gratidão a professora Cátia Farias, coordenadora da pesquisa que me deu todo apoio para que eu pudesse implementar esse trabalho. Agradeço também a minha família e a os meus amigos que, direta ou indiretamente, colaboraram para a realização deste trabalho.

LISTA DE ILUSTRAÇÕES

Figura 1 – Print da Home page e do termo de consentimento.....	20
Figura 2 – Print dos dados do pesquisador responsável.....	21
Figura 3 – Print das 5 primeiras questões.	22
Figura 4 – Print das barras de status circular.....	23
Figura 5 – Print das barras de avanço.	23
Figura 6 – Print das validações feitas pelo formulário.	24
Figura 7 - Print de uma questão de opção única.....	24
Figura 8 - Print de uma questão de múltiplas opções.....	25
Figura 9 - Listagem de relações do banco de dados.	38
Figura 10 - Relação colunas e linhas.....	38
Figura 11 - Configuração do SO instalado na hospedagem da TI do UNIFESO.	39
Figura 12 - Classe Dados.....	41
Figura 13 - Estrutura Django.	42
Figura 14 - Configuração de rotas.	43
Figura 15 - <i>ViewSet</i>	44
Figura 16 - <i>Serializer</i>	44
Figura 17 - Estrutura <i>front-end</i>	47
Figura 18 - Arquivos de entrada do React.....	48
Figura 19 - Função Render da página.....	48
Figura 20 - <i>Constructor</i> da classe.	49
Figura 21 - Configuração do projeto na hospedagem.....	50
Figura 22 - Configuração NGINX.....	51
Figura 23 - Estrutura da solução.....	52
Figura 24 - Área do Pesquisador.	53

LISTA DE TABELAS

Tabela 1 – Tabela com comparações das soluções de pesquisa.....	19
Tabela 2 – Tabela de recursos oferecidos.	19
Tabela 3 - Dicionário de Dados.....	26
Tabela 4 - Limites gerais do PostgreSQL.	37

RESUMO

Desde que foi identificado o novo coronavírus na cidade de Wuhan, na província de Hubei, China, a Organização Mundial da Saúde (OMS) considerou o surto pandêmico como uma Emergência de Saúde Pública de Importância Internacional que, pelo Regulamento Sanitário Internacional, corresponde ao nível mais alto de alerta da Organização (BRASIL, 2020).

Em meio a tal cenário, naquela ocasião, surgiu a necessidade de novas adaptações nas organizações para com suas rotinas de trabalho, de ensino e de aprendizagem. Desde então, passou-se a vivenciar o denominado, o novo normal, em que a convivência social foi redefinida de forma que a sociedade deu prosseguimento a sua rotina. Em decorrência dessas novas ações em que o espaço virtual ganhou evidências quanto aos processo de trabalho e relações pessoais e interpessoais, propôs-se o presente trabalho de implementação tecnológica relacionado à pesquisa científica contemplada no Plano de Iniciação Científica e Pesquisa (PICPq 2020-21) do Unifeso: “Projeto Integrado Sobre os Impactos das Medidas de Isolamento e Distanciamento Social pela Covid-19”, no que consistiu na construção de um banco de dados a partir de especificações técnicas adequadas para registrar os dados levantados no questionário online aplicado junto à comunidade do Unifeso.

Palavras-chave: Banco de Dados; Covid-19; Implementação Tecnológica.

ABSTRACT

Since the new Coronavirus was identified in Wuhan City, Hubei Province of China, World Health Organization (WHO) has considered the pandemic outbreak as a Public Health Emergency of International Concern that, by the International Health Regulation, corresponds to the highest level of alert of the organization (BRASIL, 2020).

Amid this scenario, in that occasion, new adjustments became necessary in organizations' work, education and learning routines. Since then, everyone started to experience what is called the new normal, where social coexistence was redefined in a way that society continued its routine. As a result of these new actions, in which the virtual environment has gained focus in the matter of work and personal relations, it was suggested the present work of technologic implementation related to the scientific research contemplated on "Plano de Iniciação Científica e Pesquisa" (PICPq 2020-21) of Unifeso: "Projeto Integrado Sobre os Impactos das Medidas de Isolamento e Distanciamento Social pela Covid-19", which consisted on building a database from suitable technical specifications to register data collected from an online quiz applied alongside Unifeso's community.

Keywords: Database, Covid-19, Technological Implementation.

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	MOTIVAÇÃO	12
1.2	OBJETIVO GERAL	12
1.3	OBJETIVOS ESPECÍFICOS.....	12
1.4	ORGANIZAÇÃO DO TRABALHO.....	13
2	TRABALHOS RELACIONADOS	14
2.1	DRAGONFLY.....	14
2.2	IDENTIFICAÇÃO AUTOMÁTICA DE CONHECIMENTOS RELACIONADOS AOS CASOS DE DENGUE NO ESTADO DO PIAUÍ EM BANCOS DE DADOS PÚBLICOS POR MEIO DE REDES DE REGRAS DE ASSOCIAÇÃO FILTRADAS.	14
2.3	HISTÓRIA DOS BANCOS DE DADOS	15
2.4	ANÁLISE DE SISTEMAS DE GERENCIAMENTO DE BANCOS DE DADOS PARA ARMAZENAMENTO DE DADOS CLIMÁTICOS	15
2.5	VANTAGENS E LIMITAÇÕES DOS QUESTIONÁRIOS ELETRÔNICOS VIA INTERNET NO CONTEXTO DA PESQUISA CIENTÍFICA	16
2.6	UTILIZAÇÃO DO GOOGLE FORMS NA PESQUISA ACADÊMICA.....	17
2.7	O BANCO DE DADOS DE ESTRUTURA DE CRISTAL DO AMERICAN MINERAGLOGIST .	17
2.8	BANCO DE DADOS DA ATUALIDADE ITALIANA INDICADORES DE ESTRESSE, IPSI 1.4.	17
2.9	RELAÇÃO DOS TRABALHOS MENCIONADOS COM O PROJETO	18
3	SOLUÇÃO E METODOLOGIA	20
3.1	SOLUÇÃO	20
3.2	METODOLOGIA.....	25
4	DESENVOLVIMENTO E TECNOLOGIAS UTILIZADAS.....	36
4.1	DESENVOLVIMENTO.....	36
4.1.1	Apresentar o conceito de <i>back-end</i> e <i>front-end</i>	36
4.1.2	PostgreSQL 12	36
4.1.3	Linux CentOS 7	38
4.1.4	Python 3.6.8: Linguagem de programação utilizada para construção do <i>back-end</i> 40	
4.1.5	Django: <i>Framework</i> Python para desenvolvimento do <i>back-end</i>	41
4.1.6	<i>Django Rest Framework</i> : <i>Framework</i> Python para construção do <i>web service</i> utilizando a arquitetura <i>Rest API</i>	42

4.1.7	Javascript ES6: Linguagem de programação utilizada para construção do <i>Front-end</i>	45
4.1.8	Node.JS 14x: Software utilizado para executar código JavaScript no <i>back-end</i>	45
4.1.9	ReactJs: Biblioteca JavaScript para desenvolvimento do <i>front-end</i>	46
4.1.10	uWSGI: Software para construção de serviços de hospedagem	50
4.1.11	Nginx: Servidor HTTP	50
4.2	HOSPEDAGEM	51
5	CONCLUSÃO E TRABALHOS FUTUROS	53
6	REFERÊNCIAS BIBLIOGRÁFICAS	55

1 INTRODUÇÃO

Desde 2019, o mundo tem sido assolado pelo novo coronavírus (SARS-CoV-2), causador da doença do coronavírus 2019 (Covid – Corona Virus Disease). A partir de então, de forma global, surgiu a necessidade de novas medidas para convivência social a fim de conter o avanço e propagação do vírus que é transmitido por gotículas respiratórias. Na readaptação da humanidade para com suas rotinas, sejam trabalhistas, pessoais ou educacionais, o impacto das medidas tomadas requer estudos e análises que venham resultar em práticas assertivas de retomada ao processo de desenvolvimento socioeconômico, bem como possam inferir em modelos mais aptos a garantir a prevenção de situações de riscos associados à saúde pública, favorecendo a implementação de planos e programas de gerenciamento de riscos à saúde de forma geral. Neste sentido, um grupo de pesquisadores do Unifeso, liderados pela pesquisadora Profa. Dra. Cátia Araujo Farias, assessora pedagógica da DEaD, buscaram analisar os efeitos das medidas de isolamento e de distanciamento social nas relações no âmbito familiar e profissional da comunidade acadêmica no contexto da pandemia da Covid19. Para tanto, necessitavam organizar dados de pesquisa sobre os efeitos das medidas de isolamento e de distanciamento social, no âmbito das relações profissionais e familiares dos estudantes, dos professores, dos técnico-administrativos e dos gestores das unidades acadêmicas, assistenciais e administrativas do Unifeso, em relação aos aspectos que interferiram na vida familiar em razão do teletrabalho e vice-versa, assim como as relações interpessoais entre seus pares e as consequências sobre sua produtividade. Tendo essa necessidade de construir um banco de dados a partir de especificações técnicas adequadas para registrar os dados de forma segura e sigilosa, surgiu o presente trabalho, em que por meio de um formulário online, obtivesse as informações necessárias para proposto no projeto integrado. Dessa forma, o presente trabalho vem trazendo a execução dessa pesquisa, que já foi disponibilizada através de formulário online (<https://unifeso.edu.br/pesquisacovid19/>), as informações aqui recolhidas que estão compondo o banco de dados para ser utilizado como meio de pesquisa nas áreas de conhecimento da universidade Unifeso, conforme ensejo do projeto integrado no que tange a integração com outros centros de pesquisa e, fundamentalmente entre as três grandes áreas do conhecimento do Unifeso, considerando as linhas de pesquisa desenvolvidas com o Plano de Iniciação Científica e Pesquisa (PICPq), no apoio ao desenvolvimento de projetos de pesquisa dos estudantes dos cursos de graduação e pós-graduação.

Aqui será apresentado de que forma se deu essa transformação para dados, através da ciência da computação, os meios utilizados para que as informações estivessem disponíveis possibilitando assim o acesso dos usuários para consulta.

1.1 MOTIVAÇÃO

Após o conhecimento da temática do projeto integrado, bem como da necessidade de construção do banco de dados para registrar os dados obtidos da pesquisa sobre os efeitos das medidas de isolamento e de distanciamento social, no âmbito das relações profissionais e familiares da comunidade acadêmica do Unifeso, em relação aos aspectos que interferiram na vida familiar em razão do teletrabalho e vice-versa, assim como as relações interpessoais entre seus pares e as consequências sobre sua produtividade, buscou-se empenho na propositura de auxílio técnico, de maneira a garantir os prepostos do referido projeto.

Na ocasião, concorreu-se a uma bolsa de pesquisa científica para a área de ciências da computação e, após minucioso processo de seleção, obteve-se a oportunidade de participar do grupo de pesquisa com o referido trabalho de “Implementação de Solução Sistêmica para Gestão de Dados em Pesquisa Científica”, de maneira a contribuir com o registro das informações de forma segura e sigilosa, garantindo a criação do banco de dados a ser disponibilizado a outros pesquisadores interessados na temática estudada, bem como aos estudantes das três grandes áreas do conhecimento do Unifeso, no apoio ao desenvolvimento de seus projetos de pesquisa de finalização de curso.

1.2 OBJETIVO GERAL

Construir de um banco de dados a partir de especificações técnicas adequadas para registrar os dados levantados no questionário online aplicado junto à comunidade do Unifeso.

1.3 OBJETIVOS ESPECÍFICOS

1. Coletar dados através de um formulário online a partir de formulação de questões abertas e fechadas.
2. Testar modelos de formulários e métodos de entrevista online para análise qualitativa, quantitativa e quali-quantitativa.

3. Disponibilizar os dados obtidos à comunidade do Unifeso por meio de recursos online.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está estruturado da seguinte forma: Introdução, em que se apresenta a motivação do estudo, bem como de seus objetivos. No referencial teórico, tem-se a descrição dos principais trabalhos relacionados a questão de banco de dados, bem como uma relação dos trabalhos relacionados especificamente com o projeto. Em seguida, no item metodologia, descreve-se a proposta do projeto e demonstra-se a solução implementada, para então, ser descrito as tecnologias utilizadas e como cada uma foi implementada na solução, assim como as suas maiores dificuldades e vantagens. Finaliza-se o trabalho acadêmico com as considerações finais em que se sinaliza oportunidades de trabalhos futuros, que darão continuidade a este projeto.

2 TRABALHOS RELACIONADOS

2.1 DRAGONFLY

O projeto consiste na criação do banco de dados através de informações que podem ser inseridas por qualquer usuário da comunidade científica de maneira livre. Os pesquisadores disponibilizam o site aberto para consulta e inclusão em tempo integral, para que qualquer usuário do ramo possa utilizar das informações já alocadas, assim como inserir novas informações, que precisam ser validadas pelos gestores do site online e, uma vez aprovada, a informação já fica disponível no banco junto ao nome do contribuinte em questão. Dessa maneira, o projeto visa construir um banco de dados através de armazenamento de dados de características morfológicas básicas de insetos europeus, com mérito nas informações sobre toda as fases da vida do inseto.

2.2 IDENTIFICAÇÃO AUTOMÁTICA DE CONHECIMENTOS RELACIONADOS AOS CASOS DE DENGUE NO ESTADO DO PIAUÍ EM BANCOS DE DADOS PÚBLICOS POR MEIO DE REDES DE REGRAS DE ASSOCIAÇÃO FILTRADAS.

O projeto visa recolhimento de dados com objetivo de identificar padrões comportamentais da dengue no estado do Piauí. Conta com a base de dados nacional do SINAN (Sistema de informação de agravos de notificação), este sistema reúne dados de notificação de doenças epidemiológicas de municípios de todo o Brasil, porém é facultado aos estados e municípios o envio de dados ao SINAN, não sendo possível abranger todos os cenários do país.

O projeto conta também com a EMBRAPA-Meio Norte (CPAMN), é uma empresa brasileira de pesquisa agropecuária vinculada ao ministério da Agricultura, Pecuária e Abastecimento, de onde as informações de dados meteorológicos foram retiradas para que junto com os dados do SINAN fosse utilizada a técnica de extração de dados para construir informações.

Em todos os casos as informações são anônimas, não expondo os infectados e seus dados. O projeto tem como resultado principal ajudar a mapear casos de dengue e dar poder de tomada de decisão a órgãos governamentais públicos que poderão criar políticas de prevenção e contenção da dengue e de outras doenças epidemiológicas.

2.3 HISTÓRIA DOS BANCOS DE DADOS

O artigo relata a história do banco de dados para diversas áreas do conhecimento, traz uma retrospectiva bem simplificada sobre os avanços da tecnologia e as contribuições das empresas tornando possível os avanços tecnológicos fazendo com que a ferramenta possa chegar em todas as partes do mundo de alguma forma.

A primeira geração de banco de dados se dá em 1960 ainda de maneira bem lenta, partindo para os primeiros modelos de banco de dados relacional a partir dos anos de 1970, protagonizados por E.F Codd e P. Chen que revolucionaram a forma com que as pessoas pensavam em armazenamento de dados. De 1980 em diante muitos avanços foram possíveis abrindo caminho para que o armazenamento de dados que conhecemos nos dias de hoje. O artigo também traz a visão dos autores em relação ao futuro dos meios de armazenamento de dados, como o conceito bancos de dados distribuídos ou federados, e também a utilização de armazenamento de dados pessoais em nuvem.

2.4 ANÁLISE DE SISTEMAS DE GERENCIAMENTO DE BANCOS DE DADOS PARA ARMAZENAMENTO DE DADOS CLIMÁTICOS

A dissertação busca identificar formas eficientes de armazenamento e recuperação de dados de Séries Temporais obtidos a partir de torres meteorológicas.

O autor levanta uma problemática no ramo da pesquisa meteorológica, a disponibilidade de dados e acesso aos mesmos são muitas das vezes muito complexas e não possuem uma boa credibilidade e confiabilidade, já que os dados existentes sofrem modificações indevidas e indesejadas, permanecendo em pesquisas posteriores. Então, esta dissertação é voltada para verificar o desempenho de SGDBs (sistemas gerenciadores de banco de dados) relacionais e não relacionais, analisando o seu bom armazenamento, monitoramento e gestão das informações.

Percebe-se a descrição de um pouco sobre a história do banco de dados, a sua definição e seu surgimento. Também é apresentando os principais modelos de banco de dados, sua classificação e modo de funcionamento.

O estudo busca a todo momento comparar duas tecnologias muito utilizadas e disseminadas na área de armazenamento de dados. De um lado, o SGDB PostgreSQL que possui em sua essência o modelo objeto relacional, seguindo a padronização SQL (Structured Query Language). O PostgreSQL é muito utilizado no ramo comercial de desenvolvimento de software complexos, concorrendo com outras alternativas como SQL Server e Oracle.

É também analisado o SGDB Cassandra, um sistema gerenciador de banco de dados não relacional de código fonte aberto a comunidade. SGDBs com características NoSQL possuem uma boa escalabilidade e são distribuídos, utilizam do modelo de dados de linhas particionadas com consistência ajustável. O Cassandra faz particionamento, replicação, detecção de falhas e recuperação automática, porém não garante a consistência dos dados em todas as réplicas.

No estudo em questão é feito um Benchmark, ou seja, um padrão de testes em massa para avaliar o desempenho de algo. Para a escolha do software que realizaria os testes, foi utilizado como critério a capacidade de analisar os dois bancos de dados do estudo, então YCSB foi escolhido para realizar a tarefa. O YCSB, é um framework de código aberto da Yahoo, que fornece um conjunto de programas muito utilizados para avaliação de desempenho de SGDBs.

Depois de toda a sessão de testes, foi concluído que o SGDB Cassandra foi o que se revelou mais adequado para a implantação nas plataformas de gerenciamento de dados climáticos. Isso se deu, porque o Cassandra apresentou bons resultados ao inserir dados no banco.

É possível identificar com muita facilidade que cada SGDB tem seus pontos fortes e fracos com funcionalidade únicas, o objetivo do autor não é levantar uma discussão sobre o melhor gerenciador de banco de dados, mas sim, qual que possui melhor desempenho para trabalhar com dados meteorológicos.

2.5 VANTAGENS E LIMITAÇÕES DOS QUESTIONÁRIOS ELETRÔNICOS VIA INTERNET NO CONTEXTO DA PESQUISA CIENTÍFICA

Nesse artigo os autores vêm mostrar vantagens e desvantagens da aplicação de formulários online dentro do âmbito da pesquisa científica.

As principais desvantagens identificadas na aplicação de questionários eletrônicos relatadas no artigo em questão são: os entrevistados precisam ter acesso a internet; dificuldade em incentivar participantes a enviarem suas respostas; a baixa confiabilidade dos dados, uma vez que qualquer indivíduo pode se submeter a pesquisa e falsificar informações já que são anônimas.

Em contrapartida, se comparada a uma pesquisa via correios, a pesquisa eletrônica tem baixo custo, maior alcance, agilidade e rapidez nas respostas.

Os autores consideram que todas as metodologias de pesquisa apresentam limitações e cabe ao pesquisador identificar qual se adapta aos requisitos do projeto de pesquisa científica.

2.6 UTILIZAÇÃO DO GOOGLE FORMS NA PESQUISA ACADÊMICA

O artigo apresenta os resultados de uma oficina acadêmica que buscou despertar o interesse de docentes e discentes a usar o Google Forms na pesquisa acadêmica. Disponibilizada pela Google, é uma ferramenta online gratuita que possibilita a criação de formulários online e disponibiliza o conteúdo facilmente nas plataformas para acesso dos usuários através de links ou endereço web. O aplicativo é de fácil utilização e já se integra automaticamente as demais ferramentas da Google como Gmail, Sheets, entre outros podendo assim compartilhar os resultados da pesquisa científica.

2.7 O BANCO DE DADOS DE ESTRUTURA DE CRISTAL DO AMERICAN MINERAGLOGIST

Trata-se de um banco de dados que contém estruturas cristalinas. Para esse banco é fornecido um conjunto de software interativo que pode ser usado para visualizar e manipular as estruturas cristalinas, computam diferentes propriedades de um cristal, como geometria, difração padrões e densidades de elétrons procrístais. O banco é mantido pela *Mineralogical Society of America* e pela University do Arizona de onde cedem acesso gratuito aos usuários. É um banco de fácil acesso que

Inclusive é disponibilizado como meio de ajuda na educação em sala de aula da universidade.

2.8 BANCO DE DADOS DA ATUALIDADE ITALIANA INDICADORES DE ESTRESSE, IPSI 1.4

O banco de dados italiano de indicadores atuais de estresse (IPSI) é um banco de dados italiano georreferenciado, disponível gratuitamente repositório de informações sobre o campo de tensões cristais. Consiste em orientações horizontais de tensão que foram analisados, compilados em um formato padronizado e classificados de acordo com a confiabilidade e comparabilidade em escala global. O banco de dados contém uma coleção de informações sobre tensão contemporânea dentro da crosta rasa das seguintes categorias principais de indicadores de tensão: rupturas de poços; mecanismos focais de

terremotos; sequências sísmicas e dados de escorregamento de falha ativo. O presente banco de dados (IPSI 1.4) lançado em janeiro de 2020 está acessível através de uma interface web que facilita a descoberta, acessibilidade, interoperabilidade e reutilização do conjunto de dados. Além disso, contém 928 registros atualizados até dezembro de 2019 com acréscimo de 10% em relação ao primeiro, e informações de metadados aprimoradas. A distribuição uniforme de dados de estresse em um determinado território é relevante para modelagem da crosta terrestre ou como ponto de partida em muitos estudos aplicados. Portanto, é necessário continuar coleta de novos dados e atualização dos mapas de estresse atuais para obter avaliações mais confiáveis.

2.9 RELAÇÃO DOS TRABALHOS MENCIONADOS COM O PROJETO

Dragonfly: O projeto disponibiliza através da internet uma interface web construída com tecnologias básicas da web como HTML, Javascript e CSS, sendo gerenciadas por um servidor feito em PHP. O gerenciador de banco de dados utilizado é o MySQL. Através do site, disponível online 24 horas, é possível consultar e filtrar as características das libélulas, de acordo com a necessidade do pesquisador, e então baixar todos os dados filtrados em um arquivo CSV.

Através do artigo sobre a história dos bancos de dados que o projeto definiu o tipo de esquema que a solução implementaria: Ficou decidido que seria utilizado o esquema de um banco de dados relacional.

Na Dissertação “Análise de Sistemas de Gerenciamento de Banco De Dados para Armazenamento de Dados Climáticos”: Faz uma análise utilizando várias métricas de desempenho e benefícios dos gerenciadores de banco de dados PostgreSQL e Cassandra. Como resultado das análises é possível identificar que o banco de dados Cassandra tem maior desempenho ao gravar e atualizar informações, porém ao consultar dados o PostgreSQL é mais recomendado do que o Cassandra.

Abaixo é possível comparar como foi implementado o projeto Dragonfly e o projeto de pesquisa institucional feito dentro do UNIFESO (Pesquisa que foi usada como motivação para esse trabalho).

Tabela 1 – Tabela com comparações das soluções de pesquisa.

	DRAGONFLY	PROJETO
MODELO	Relacional	Relacional
SGDB	MySQL	PostgreSQL
FRONT-END	HTML, CSS, Javascript	ReactJs
BACK-END	PHP	Django
HOSPEDAGEM	Czech University of Life Sciences Prague	UNIFESO
DISPONIBILIDADE	24 Horas	24 Horas
ACESSO AOS DADOS	Comunidade	Comunidade Unifeso
DOWNLOAD	Arquivo .CSV	Arquivo .CSV

Fonte: dragonfly-database

Tabela 2 – Tabela de recursos oferecidos.

	DRAGONFLY	PROJETO
Os dados são de fácil acesso?	Sim	Sim
Possui área do pesquisador?	Sim	Sim
Possui um dicionário de dados?	Não	Sim
Permite filtrar os dados?	Sim	Não
Permite contribuições de novos dados?	Sim	Sim
A submissão ao formulário é anônima?	Não	Sim
Formulário Inteligente?	Sim	Sim

Fonte: dragonfly-database

3 SOLUÇÃO E METODOLOGIA

3.1 SOLUÇÃO

O formulário online desenvolvido foi hospedado nas instalações da TI do UNIFESO, e já se encontra online podendo ser acesso através do seguinte endereço: (<https://unifeso.edu.br/pesquisacovid19>).

O entrevistado pode usar qualquer dispositivo que navegue na internet para se submeter a pesquisa, como um *smartphone*, *tablet*, notebook ou computador. Google Chrome, Firefox, Mozilla ou Safari; basta abrir algum desses programas para navegar no formulário.

Ao carregar a página é exibido no centro da tela um botão para que o entrevistado clique e inicie a pesquisa. Então, é mostrado ao usuário um termo de consentimento e informações sobre a pesquisa, pesquisadores e seus objetivos (Figura 1 e 2):

Figura 1 – *Print* da Home page e do termo de consentimento.



Fonte: Autor próprio

Figura 2 – *Print* dos dados do pesquisador responsável.

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO – TCLE
×

14 de agosto de 2018) e possui aprovação do Conselho de Ética-Unifeso (Documento nº 4.274.898).

Sua participação é voluntária e você pode deixar de responder a qualquer momento, sem que isso lhe cause qualquer problema.

Se você tiver alguma dúvida, você pode contatar a coordenadora da pesquisa:

Nome do pesquisador(a):	Luciana Nunes Ferreira
Cargo/Função:	Professor Auxiliar e Assessora Pedagógica, DEaD-Unifeso
Instituição:	Centro Universitário Serra dos Órgãos - Unifeso
e-mail:	luciananunes@unifeso.edu.br
Telefone:	(21) 98189-3827

☐ Eu concordo em participar da pesquisa, conforme assinatura exarada no TCLE. Li e entendi em que consiste minha participação.

Você deve concordar com os termos antes de continuar

Fechar

Continuar

Fonte: Autor próprio

É necessário que o participante da pesquisa leia e aceite o termo de consentimento. Uma vez dando o consentimento, a pesquisa será iniciada. Para tanto, o usuário deve marcar a caixa com o texto “Eu concordo em participar da pesquisa, conforme assinatura exarada no TCLE. Li e entendi em que consiste minha participação”. Após esse procedimento, portanto, as perguntas de cunho pessoal são demonstradas ao entrevistado e a pesquisa começa, conforme a figura 3.

Figura 3 – *Print* das 5 primeiras questões.

The figure displays two sequential screenshots of a mobile application interface for a COVID-19 survey. The interface is designed for a smartphone, with a status bar at the top showing the time and battery level. The background of the app is a light gray with a faint image of a building.

Left Screenshot (12:10):

- Top Bar:** Shows the time 12:10 and the URL `pesquisacovid19.unifeso.edu.br`.
- Navigation:** Two circular buttons with left and right arrows are at the top.
- Progress Indicator:** A circular progress bar at the top right shows 0% completion.
- Questions:**
 - Bairro:** Text input field containing "Ermitage".
 - Cidade:** Text input field containing "Teresópolis".
 - Selecione o seu estado:** Dropdown menu showing "RJ - Rio de Janeiro".
- Instructions:** "Preencha as informações de onde você vive e clique no botão Continuar".
- Bottom Bar:** Copyright notice "Copyright © 2021. Unifeso - Desenvolvido por Victor Veiga" and navigation icons (back, forward, home, and a list icon).

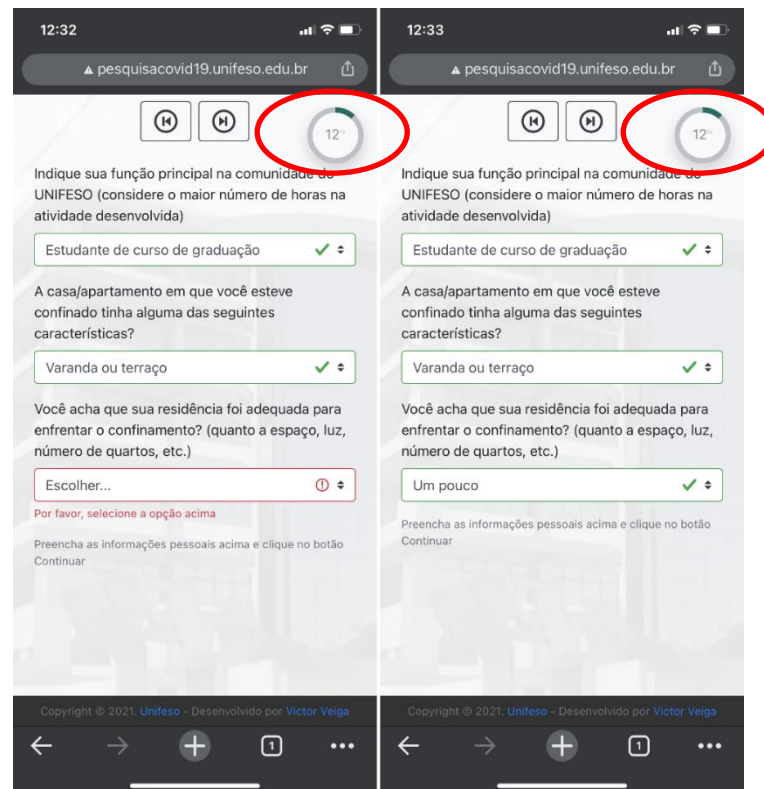
Right Screenshot (12:11):

- Top Bar:** Shows the time 12:11 and the same URL.
- Navigation:** Same circular buttons at the top.
- Progress Indicator:** The circular progress bar at the top right now shows 3% completion.
- Questions:**
 - Data de nascimento:** Text input field containing "26 Mar 1999".
 - Com que gênero você se identifica?:** Dropdown menu showing "Masculino".
- Instructions:** "Preencha as informações pessoais acima e clique no botão Continuar".
- Bottom Bar:** Same copyright notice and navigation icons as the left screenshot.

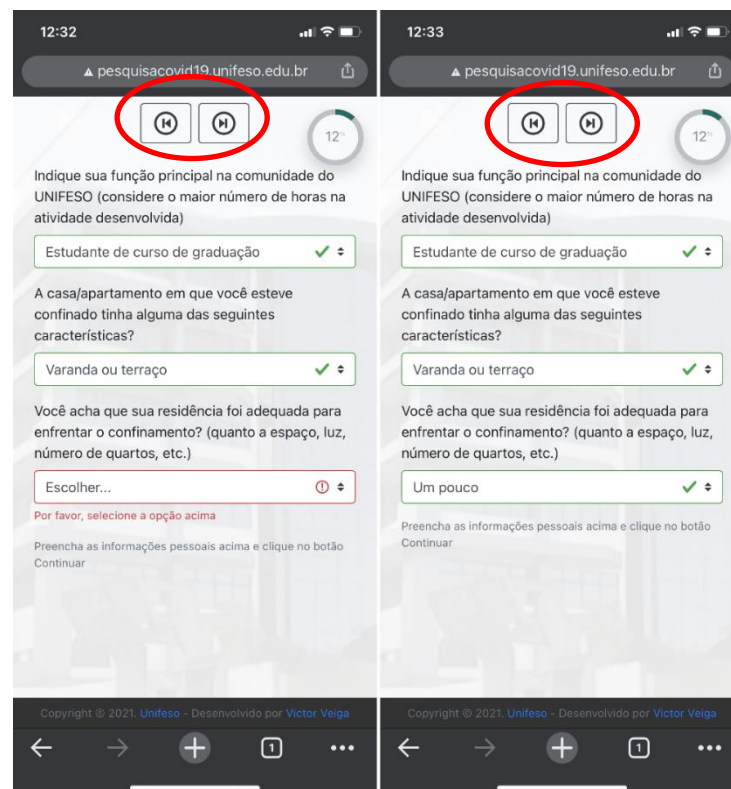
Fonte: Autor próprio

São exibidas ao participante grupos de questões de maneira paginada. Dessa forma fica mais fácil para o usuário responder as questões, já que ele consegue ver tudo que precisa responder naquele momento, ao invés de exibir tudo numa página de rolagem. Além disso, existe uma barra de *status* circular (Figura 4) que informa ao entrevistado o seu progresso durante a pesquisa, dando a ele a percepção de quanto ainda falta responder para terminar a pesquisa.

Para navegar no formulário o entrevistado pode clicar nos dois botões na parte superior da página, o botão esquerdo volta para o grupo de perguntas anterior e botão direito avança para o próximo grupo (Figura 5). O formulário conta com uma validação de informações submetidas, que não permite que o usuário avance para próxima página sem que antes as informações prestadas estejam de acordo com as validações impostas no dicionário de dados, assim garantindo a consistência da informação coletada (Figura 6).

Figura 4 – *Print* das barras de *status* circular.

Fonte: Autor próprio

Figura 5 – *Print* das barras de avanço.

Fonte: Autor próprio

Figura 6 – *Print* das validações feitas pelo formulário.

Fonte: Autor próprio.

Na figura 7, tem-se o exemplo de uma questão fechada de opção única. As questões foram elaboradas no formato aberto e fechado, ou seja, formada por opções de resposta já preenchidas para o entrevistado escolher e outras discursivas. Foram construídas de maneira a melhor auxiliar os pesquisadores em suas abordagens de estudos científicos, de forma qualitativa, quantitativa ou quali-quantitativa (Figura 8).

Figura 7 - *Print* de uma questão de opção única.

Fonte: Autor próprio

Figura 8 - *Print* de uma questão de múltiplas opções.

The figure consists of two side-by-side screenshots of a mobile application interface. Both screenshots show a survey question with a progress bar at the top right.

Left Screenshot (13:33): The question is "Caso tenha respondido 'sim' na pergunta anterior, indique o motivo? (você pode responder mais de uma)". The progress bar is at 31%. The options are:

- Devido ao número de pessoas precisando utilizar a internet em casa
- Devido a necessidade de lazer (assistir filmes, games, Tv etc.)
- Devido ao serviço de home office
- Devido ao serviço de home office e utilização do Ambiente Virtual de Aprendizagem (AVA)
- Devido a utilização do Ambiente Virtual de Aprendizagem (AVA)
- Outra

 A note at the bottom says "Selecione uma ou mais opções acima".

Right Screenshot (13:35): The question is "Quais os sistemas operacionais que você mais utiliza?". The progress bar is at 56%. The options are:

- Windows
- Distribuição Linux
- Mac OS
- iOS
- Android
- Não sei especificar
- Outro

 Below the "Outro" option, there is a text input field labeled "Especificar outro sistema operacional" with a red border and a warning icon. A note below the input field says "Por favor, especifique o sistema operacional". At the bottom, it says "Escolha uma da opções acima".

Both screenshots show a copyright notice at the bottom: "Copyright © 2021. Unifeso - Desenvolvido por Victor Veiga".

Fonte: Autor próprio

Para a implementação desse projeto de pesquisa, foi optado por criar um formulário online, ao invés de usar algum já existente, de maneira a oferecer aos participantes da pesquisa uma boa experiência de entrevista. Além disso, o questionário desenvolvido consegue ajustar-se de acordo com as respostas do usuário, ou seja, depois das perguntas comuns, as perguntas específicas são demonstradas por página e, conforme o entrevistado vai respondendo, o questionário vai se adaptando e exibindo perguntas relacionadas ao contexto em que o entrevistado está inserido.

3.2 METODOLOGIA

As perguntas do formulário foram produzidas anteriormente e organizadas em um formato de dicionário de dados, totalizando cerca de 46 questões, sendo 13 discursivas e 33 fechadas.

Na tabela 3, tem-se o dicionário de dados, correspondentes às questões formuladas.

Tabela 3 - Dicionário de Dados

Nº	NOME	TAMANHO	OCORRÊNCIA	PERGUNTA	CONDIÇÃO
1	BAIRRO	100	1-1	Que bairro reside?	
2	CIDADE	50	1-1	Que cidade reside?	
3	UF	2	1-1	UF da cidade	UF válida
4	NASCIMENTO	10	1-1	Em que ano você nasceu	Data válida, YYYY-MM-DD
5	GENERO	1	1-1	Com que gênero você se identifica? M - Masculino F - Feminino O - Outro	Valores válidos: M, F, O
6	GENERO_OUTRO	50	0-1	Especificar o genero	Obrigatório se SEXO for O - Outro
7	COR	1	1-1	Com que cor você se identifica? B - Branca P - Parda N - Preta A - Amarela	Valores válidos: B, P, N e A
8	RELACIONAMENTO	1	1-1	Qual é o seu estado civil? S - Solteiro/a C - Casado/a ou união estável D - Desquitado/a, separado/a judicialmente ou divorciado/a V - Viúvo/a	Valores válidos S, C, D ou V

9	INSTRUCAO	1	1-1	Qual é o seu nível máximo de estudos concluídos? 1 - Ensino fundamental 2 - Ensino médio 3 - Ensino técnico 4 - Ensino superior 5 - Pós-graduação (especialização, mestrado ou doutorado) 6 - Não sabe 7 - Outro	Valores válidos 1,2,3,4,5,6 ou 7
10	INSTRUCAO_OUTRO	255	0-1	Especificar os estudos	Obrigatório se INSTRUCAO for O - Outro
11	TRABALHO_ESTUDO	1	1-1	Qual é a sua situação de trabalho/estudo? 1 - Estudante e empregado com carteira assinada 2 - Dona/o de casa ou cuidador/a de pessoas dependentes 3 - Empregado com contrato temporário 4 - Empregado com carteira assinada 5 - Empregador 6 - Outro	Valores válidos 1,2,3,4,5, ou 6
12	TRABALHO_ESTUDO_OUTRO	255	0-1	Especificar a situação de trabalho/estudo	Obrigatório se TRABALHO_ESTUDO for O - Outro

13	FUNCAO_UNIFESO	1	1-1	Assinale sua função principal na comunidade do UNIFESO (considere o maior número de horas na atividade desenvolvida): 1 - Estudante de curso de graduação. 2 - Estudante de curso de pós-graduação/residência. 3 - Docente. 4 - Profissional de saúde. 5 - Função administrativa. 6 - Serviços gerais. 7 - Outra.	
14	FUNCAO_UNIFESO_ESPEC	100	0-1	Especificar a função	Obrigatório se FUNCAO_UNIFESO for 7 - Outra
15	MORADIA_CHARACTERISTICAS	1	1-1	A casa/apartamento em que você está confinado tem alguma das seguintes características? 1 - Varanda ou terraço 2 - Jardim privado 3 - Outras áreas externas 4 - Nenhum espaço exterior	Valores válidos 1,2,3 ou 4
16	RESIDENCIA_ADEQUADA	1	1-1	Você acha que sua residência é adequada para enfrentar o confinamento? (quanto a espaço, luz, número de quartos, etc.) 1 - Nada 2 - Um pouco 3 - Moderadamente 4 - Bastante 5 - Muito	Valores válidos 1,2,3,4 ou 5

17	TIPO_CONEXAO	1	1-1	Qual a sua conexão de internet ? 1 - Banda larga (internet residencial ou corporativa) 2 - Internet móvel 3G/4G 3 - Internet via satélite 4 - Não sei indicar 5 - Outro(Especificar)	Valores válidos 1, 2, 3,4 ou 5
18	TIPO_CONEXAO_ESPEC	100	0-1	Especificar a conexão sua conexão com a internet	Obrigatório se TIPO_CONEXAO for 5
19	TIPO_REDE	1	1-1	Tipo da rede que você costuma se conectar. 1 - Rede pública (faculdade, hospital, rodoviária, shopping) 2 - Rede privada (redes empresariais e de segurança governamental) 3 - Rede doméstica (redes fornecidas diretamente por provedores de internet)	Valores válidos 1,2 ou 3
20	DISPOSITIVO_USADO	1	1-1	Qual dispositivo que você mais utilizada para navegar na internet? (Tarefas cotidianas, redes sociais e trabalho) 1 – Celular ou Smartphone 2 – Computador ou notebook. 3 – Tablet 4 – Outro	Valores válidos 1,2,3 ou 4
21	DISPOSITIVO_USADO_ESPEC	100	0-1	Especificar o dispositivo utilizado para navegar na internet	Obrigatório se DISPOSITIVO_USADO for 4
22	TIPO_USUARIO	1	1-1	Qual o seu perfil de usuário da internet? 1 - Perfil de usuário iniciante, não possui uma boa familiaridade com o acesso à internet.	Valores válidos 1,2 ou 3

				<p>2 - Perfil de usuário intermediário (velocidade de conexão (até 200 mega) para fazer downloads e uploads que requerem uma melhor conexão;</p> <p>3 - Perfil de usuário completo internet rápida ou ultrarrápida para gaming, videoconferências com pessoas de outros países e outros serviços digitais inclusos, como plataformas streaming.</p>	
23	TROCOU_PERFIL_USUARIO	1	1-1	<p>Você precisou trocar o perfil de usuário da internet, durante o confinamento?</p> <p>S - Sim</p> <p>N - Não</p>	Valores válidos S ou N
24	MOTIVO_TROCA_USUARIO	1	0-6	<p>Caso tenha respondido “sim” na pergunta anterior, indique o motivo? (você pode responder mais de uma):</p> <p>1 - Devido ao número de pessoas precisando utilizar a internet em casa.</p> <p>2 - Devido a necessidade de lazer (assistir filmes, games, Tv etc.)</p> <p>3 - Devido ao serviço de home office.</p> <p>4 - Devido ao serviço de home office e utilização do Ambiente Virtual de Aprendizagem (AVA).</p> <p>5 - Devido a utilização do Ambiente Virtual de Aprendizagem (AVA).</p> <p>6 - Outra</p>	<p>Obrigatório se TROCOU_PERFIL_USUARIO for Sim</p> <p>Valores válidos 1,2,3,4,5 ou 6</p>

25	EQUIPAMENTO_HOME_OFFICE	1	1-1	Você possui os equipamentos necessários para trabalhar em home office? S – Sim N – Não	Valores válidos S ou N
26	OBSTACULO_EQUIPAMENTO	1	0-1	Se não, quais são os obstáculos? 1 - Falta de equipamentos (Computadores / notebooks) 2 - Falta de uma conexão à internet estável 3 - Falta de recursos financeiros 4 - Outro	Obrigatório se EQUIPAMENTO_HOME_OFFICE for Não
27	OBSTACULO_EQUIP_ESPEC	100	0-1	Especificar outros obstáculos do motivo da falta de equipamentos	Obrigatório se OBSTACULO_EQUIPAMENTO for 3
28	APPS_UTILIZADOS	2	1-16	Quais aplicativos ou software você utiliza? (Pode escolher mais de uma opção) 1 – Facebook 2 – LinkedIn 3 – Youtube 4 – Skype 5 – Snapchat 6 – Twitter 7 – Vimeo 8 – WhatsApp 9 - Netflix 10 - Amazon Prime Video 11 - Disney+ 12 - Globoplay 13 - Paramount+ 14 – Telegram 15 – Instagram	Valores válidos 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 ,16,17,18,19,20,21,22,23,24,25,26

				16 – Hangout 17 – Bible Youversion 18 – ResearchGate 19 – Flickr 20 – Tiktok 21 – E-mail 22 – Kindle 23 - Waze 24 - Cifra Club 25 - Aplicativos para acessar instituições financeiras 26 - Aplicativos para o aprendizado de idiomas	
29	SO_UTILIZADO	1	1-1	Quais os sistemas operacionais que você mais utiliza? 1 – Windows 2 – Distribuição Linux 3 – Mac OS 4 – iOS 5 – Android 6 - Não sei especificar 7 – Outro	Valores validos 1,2,3,4,5 ou 6
30	SO_UTILIZADO_ESPEC	100	0-1	Especificar o sistema operacional	Obrigatório se SO_UTILIZADO for 7
31	UTILIZOU_VIRTUALIDADE	1	1-1	Durante a pandemia do novo coronavírus você utilizou a virtualidade para substituir alguma de suas atividades cotidianas? S - Sim. N - Não.	Valores válidos S ou N
32	VIRTUAL_ATIVIDADE	1	0-6	Caso tenha respondido SIM, assinale a(s) atividade(s) desenvolvida(s) por você (pode marcar mais de uma)	Valores válidos 1,2,3,4,5 ou 6. Obrigatório se

				1 - Estudar/realizar atividades referentes apenas ao curso de graduação. 2 - Estudar para qualificação profissional. 3 - Trabalhar na modalidade online (home office). 4 - Fazer compras (supermercado/farmácia/padaria/vestuário). 5 - Pagar contas online (água/luz/telefone/condomínio). 6 - Outra	UTILIZOU_VIRTUALIDADE for Sim
33	VIRTUAL_ATIVIDADE_ESPEC	100	0-1	Especificar atividades desenvolvidas por você	Obrigatório se VIRTUAL_ATIVIDADE for 6 - Outra
34	VIRTUAL_ATIVIDADE_MOTIVO	1	0-4	Caso tenha respondido NÃO, assinale o(s) motivo(s) que identificados por você (pode marcar mais de um) 1 - Trabalho em unidade de saúde. 2 - Não achei necessário substituir nenhuma atividade cotidiana. 3 - Minha atividade não pode ser exercida de modo online. 4 - Outro	Valores válidos 1,2,3 ou 4. Obrigatório se UTILIZOU_VIRTUALIDADE for Não
35	VIRTUAL_AT_MOTIVO_ESPEC	100	0-1	Especificar motivos de não ter utilizado da virtualidade para substituir alguma atividade	Obrigatório se VIRTUAL_ATIVIDADE_MOTIVO for 4 - Outro
36	DIFICULDADE_USO_APP	1	0-1	Você teve dificuldades em utilizar aplicativos/software/sistemas durante o período de confinamento?	Obrigatório se TIPO_USUARIO for 2 ou 3, Valores válidos S ou N

37	CONTATAR_SUPORE_APP	1	0-1	Você precisou contatar o suporte ao usuário de algum software, aplicativo ou sistema que você fez uso?	Obrigatório se TIPO_USUARIO for 2 ou 3, Valores válidos S ou N
38	SAT_SERV_TEL	1	1-1	Conte-nos se você ficou satisfeito com os serviços prestados pela companhia de telefone: S - Sim. N - Não. O - Não fiz o uso desse serviço.	Valores válidos S, N ou O
39	SAT_SERV_INTERNET	1	1-1	Conte-nos se você ficou satisfeito com os serviços prestados pela companhia de internet: S - Sim. N - Não. O - Não fiz o uso desse serviço.	Valores válidos S, N ou O
40	SAT_SERV_TV	1	1-1	Conte-nos se você ficou satisfeito com os serviços prestados pela companhia de TV a cabo: S - Sim. N - Não. O - Não fiz o uso desse serviço.	Valores válidos S, N ou O
41	SAT_SERV_LUZ	1	1-1	Conte-nos se você ficou satisfeito com os serviços prestados por aplicativos da companhia de Luz: S - Sim. N - Não. O - Não fiz o uso desse serviço.	Valores válidos S, N ou O
42	SAT_SERV_STREAMING	1	1-1	Conte-nos se você ficou satisfeito com os serviços prestados por aplicativos de serviços de streaming: S - Sim.	Valores válidos S, N ou O

				N - Não. O - Não fiz o uso desse serviço.	
43	SAT_SERV_FINAN	1	1-1	Conte-nos se você ficou satisfeito com os serviços prestados por aplicativos de instituições financeiras (Bancos, financiadoras, corretoras de valores...): S - Sim. N - Não. O - Não fiz o uso desse serviço.	Valores válidos S, N ou O
44	SAT_SERV_GOV	1	1-1	Conte-nos se você ficou satisfeito com os serviços prestados por aplicativos de instituições governamentais (FGTS, Meu INSS, CDT entre outros...): S - Sim. N - Não. O - Não fiz o uso desse serviço.	Valores válidos S, N ou O
45	SAT_SERV_UNIVER	1	1-1	Conte-nos se você ficou satisfeito com os serviços online disponibilizados pela Universidade (Portal do aluno, ava, biblioteca online...): S - Sim. N - Não. O - Não fiz o uso desse serviço.	Valores válidos S, N ou O

Fonte: Autor próprio

4 DESENVOLVIMENTO E TECNOLOGIAS UTILIZADAS

4.1 DESENVOLVIMENTO

4.1.1 Apresentar o conceito de *back-end* e *front-end*

1. O que é *Front-End*? Tudo que envolve a parte visual, artístico, criativo de um site ou aplicação para integração dos usuários. É desenvolvida em linguagens como JavaScript, HTML e CSS, que são as técnicas de base do desenvolvimento web, que são executadas no dispositivo que o usuário tem em mãos.
2. O que é *Back-End*? É o que envolve a parte de desenvolvimento de uma aplicação, os códigos que permitem a parte visual funcionar corretamente, o que liga o banco de dados às informações do navegador, ou seja, implementa arquiteturas para viabilizar as funcionalidades do site, sistema ou app. As linguagens de programação mais utilizadas no Back-End são PHP, Python, C#, Java, Ruby, entre outras. Elas possibilitam o bom funcionamento dos servidores e a segurança dos dados, as funções do *back-end* são totalmente invisíveis para o usuário do aplicativo.

4.1.2 PostgreSQL 12

É um sistema para gerenciamento de banco de dados do tipo relacional. Suas funções principais são, armazenar dados com segurança e retornar esses dados em resposta a solicitações de outros aplicativos de software.

A primeira versão foi lançada em 97 na Universidade de Berkeley na Califórnia, é desenvolvido e mantido pelo *PostgreSQL Global Development Group*, composto por várias empresas e contribuintes individuais.

Escrito em linguagem C, é *Open Source*, pode ser utilizada por qualquer pessoa em seus projetos e está disponível em UNIX, Window, OS X e outros. No Site oficial <https://www.postgresql.org/> pode-se fazer o download do software gratuitamente e instalar no computador.

Tem todos os requisitos que um banco de dados que trabalha com transações precisa, tem suporte a views, procedimentos armazenados entre outros.

Algumas das características do PostgreSQL são:

1. Suporta particionamento;
2. Possui controle de concorrência multiversão;
3. Busca de texto completo;
4. Possui linguagem procedural, entre outras características.

PostgreSQL tem uma conectividade simplificada, trabalha via rede TCP/IP padrão. Uma vez estabelecida a conexão, a comunicação é feita através de envio de comando. Sua linguagem combina declarações padrão SQL:2008 mais alguns comandos específicos do Postgres, tem seus próprios comandos. Dessa forma pode-se criar aplicações em várias linguagens e acessar o banco de dados Postgres com facilidade.

O Postgres tem alguns limites, que pode variar de versão para versão (disponíveis para download e atualização no site oficial), como tamanho de tabelas, tamanho de linha, tamanho de campos.

Na tabela 4, tem-se os limites gerais do PostgreSQL.

Tabela 4 - Limites gerais do PostgreSQL.

Item	Valor
Tamanho máximo de banco de dados	Ilimitado
Tamanho máximo de tabela	32 TB
Tamanho máximo de linha	1,6 TB
Tamanho máximo de campos	1 GB
Nº máximo de linhas por tabela	Ilimitado
Nº máximo de colunas por tabela	250 - 1600
Nº máximo de índices por tabela	Ilimitado

Fonte: Tabela retirada da videoaula no canal do Youtube da Bóson Treinamentos: Introdução ao PostgreSQL – Curso de Postgres

Foi definido para este projeto em questão o SGDB PostgreSQL por ser livre e gratuito, relacional, robusto, expansível e suporta uma quantidade grande de gravações de registros, beneficiando assim o projeto de pesquisa que esse trabalho implementa. Na figura 9, tem-se a listagem de tabelas que compõem a estrutura do banco de dados que atende ao projeto.

Figura 9 - Listagem de relações do banco de dados.

List of relations			
Schema	Name	Type	Owner
public	auth_group	table	postgres
public	auth_group_permissions	table	postgres
public	auth_permission	table	postgres
public	auth_user	table	postgres
public	auth_user_groups	table	postgres
public	auth_user_user_permissions	table	postgres
public	django_admin_log	table	postgres
public	django_content_type	table	postgres
public	django_migrations	table	postgres
public	django_session	table	postgres
public	participante_dados	table	postgres
(11 rows)			

Fonte: Autor própria

Na relação “participante_dados” é onde é armazenado os dados dos participantes. As outras relações são criadas automaticamente pelo framework gerenciador Django. Na figura 10, é possível conferir como os dados são dispostos. As colunas representam as perguntas, cada coluna é um campo que faz referência ao nome da pergunta no dicionário de dados, conforme a tabela 3.

Figura 10 - Relação colunas e linhas.

Query Editor

Query History

1

SELECT * FROM public.participante_dados

2

ORDER BY id ASC

Data Output

Explain

Messages

Notifications

	id [PK] integer	bairro character varying (100)	cidade character varying (100)	uf character varying (2)	nascimento date	genero character varying (1)
1	1	Ermitage	Teresópolis	RJ	1999-03-26	M

Fonte: Autor próprio

4.1.3 Linux CentOS 7

O CentOS, abreviação de Community ENTERprise Operating System, é uma distribuição Linux de classe corporativa derivada de códigos fonte gratuitamente distribuídos pela *Red Hat Enterprise Linux* e mantida pelo *CentOS Project* e conta com uma família de distribuições:

- *CentOS*: Distribuição gratuita para servidores;

- *RHEL*: Distribuição paga e com suporte direto da empresa;
- *Fedora*: Distribuição voltada para máquinas de trabalho e com pacotes mais recentes.

A numeração das versões é baseada na numeração do *Red Hat Enterprise Linux*. Geralmente se usa o *CentOS* como servidor, mas também pode ser utilizado em desktop, apesar de não ser o foco dele. O download pode ser efetuado através do site [centos.org](https://www.centos.org) onde contém também, manual de instrução para instalação.

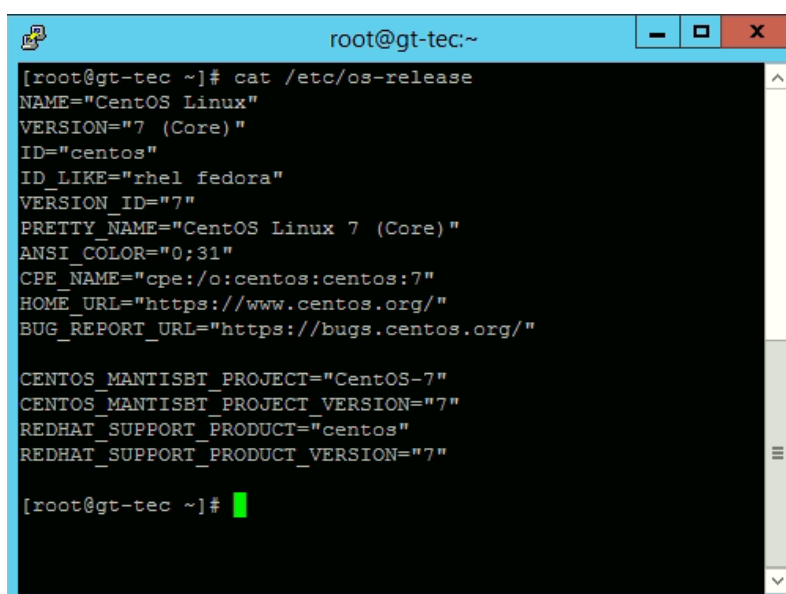
CentOS proporciona acesso aos softwares padrão da indústria, incluindo compatibilidade com os pacotes de softwares preparados especificamente para os sistemas da *Red Hat Enterprise Linux*. Isso lhe dá um bom nível de segurança e suporte.

Em dezembro de 2020, a *Red Hat* encerrou o desenvolvimento do *CentOS*. No entanto, a *Red Hat* continuará a oferecer suporte à distribuição de lançamento contínuo relacionada, *CentOS Stream*.

O *back-end* do projeto de pesquisa foi implementado no sistema operacional *CentOS*, todos os serviços necessários para receberem os dados dos participantes foi instalado nesse sistema. Por ser gratuito e muito estável, esses foram os fatores determinantes que influenciaram na escolha do sistema utilizado.

Na figura 11, tem-se a configuração do SO instalado na hospedagem da TI do Unifeso.

Figura 11 - Configuração do SO instalado na hospedagem da TI do UNIFESO.



```
root@gt-tec:~  
[root@gt-tec ~]# cat /etc/os-release  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"  
ANSI_COLOR="0;31"  
CPE_NAME="cpe:/o:centos:centos:7"  
HOME_URL="https://www.centos.org/"  
BUG_REPORT_URL="https://bugs.centos.org/"  
  
CENTOS_MANTISBT_PROJECT="CentOS-7"  
CENTOS_MANTISBT_PROJECT_VERSION="7"  
REDHAT_SUPPORT_PRODUCT="centos"  
REDHAT_SUPPORT_PRODUCT_VERSION="7"  
  
[root@gt-tec ~]#
```

Fonte: Autor próprio

4.1.4 Python 3.6.8: Linguagem de programação utilizada para construção do *back-end*

Python para que serve? Python é uma linguagem *Open-Source* de propósito geral usado bastante em *data science*, *machine learning*, desenvolvimento de web, desenvolvimento de aplicativos, automação de scripts, *fintechs* e mais.

Stack Overflow considerou Python a linguagem de renome que mais cresce, e a linguagem que os programadores mais desejam aprender. Como por exemplo em 2014 nos Estados Unidos, 80% das 10 melhores universidades de ciência da computação (e 69% dos top 39) ensinam o python em cursos introdutórios.

Desde que o Python foi criado em 1991 por Guido van Rossum, essa linguagem de programação vem crescendo no mercado se tornando uma das linguagens de programação multifuncionais líderes no mundo hoje. Os desenvolvedores de *back-end* utilizam seus códigos organizados e altamente legíveis para criar scripts funcionais para lidar com atribuições de *back-end*.

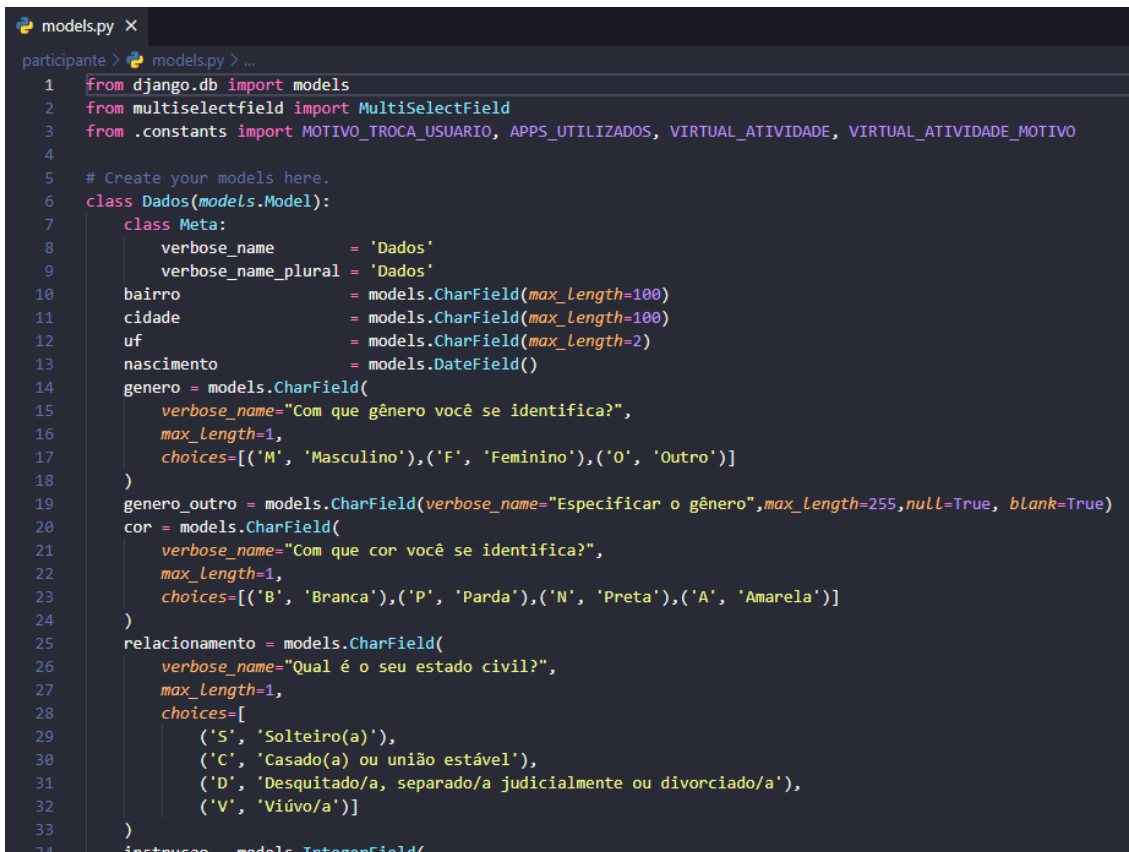
Uma pesquisa de 2020 feita por *Stack Overflow* revelou que é a principal linguagem de programação de *back-end* que os desenvolvedores esperam aprender. Python liderou essa categoria por quatro anos consecutivos. Ele também foi classificado em terceiro lugar entre as linguagens de programação mais amadas existentes. (STACK OVERFLOW, 2020)

Algumas das principais características do Python:

- Fácil de ler, escrever e de ser mantido;
- Eficiente para fazer protótipos;
- Altamente escalável e poderoso;
- Moldável: permite facilmente acessar e adicionar bibliotecas para funções específicas;
- Uma linguagem de programação *Open-Source* com uma grande comunidade;
- Códigos incorporáveis;
- Bibliotecas enormes;

Dadas as principais características do Python, fica fácil de entender o motivo de usar essa tecnologia no projeto. Como o Python suporta o paradigma de Orientação a Objetos, foi construído uma classe modelo chamada “Dados” em um pacote denominado “participante”, estendendo da classe “models.Model” do pacote “django.db” (Figura 12).

Figura 12 - Classe Dados.



```

models.py x
participante > models.py > ...
1 from django.db import models
2 from multiselectfield import MultiSelectField
3 from .constants import MOTIVO_TROCA_USUARIO, APPS_UTILIZADOS, VIRTUAL_ATIVIDADE, VIRTUAL_ATIVIDADE_MOTIVO
4
5 # Create your models here.
6 class Dados(models.Model):
7     class Meta:
8         verbose_name = 'Dados'
9         verbose_name_plural = 'Dados'
10    bairro = models.CharField(max_length=100)
11    cidade = models.CharField(max_length=100)
12    uf = models.CharField(max_length=2)
13    nascimento = models.DateField()
14    genero = models.CharField(
15        verbose_name="Com que gênero você se identifica?",
16        max_length=1,
17        choices=[('M', 'Masculino'),('F', 'Feminino'),('O', 'Outro')]
18    )
19    genero_outro = models.CharField(verbose_name="Especificar o gênero",max_length=255,null=True, blank=True)
20    cor = models.CharField(
21        verbose_name="Com que cor você se identifica?",
22        max_length=1,
23        choices=[('B', 'Branca'),('P', 'Parda'),('N', 'Preta'),('A', 'Amarela')]
24    )
25    relacionamento = models.CharField(
26        verbose_name="Qual é o seu estado civil?",
27        max_length=1,
28        choices=[
29            ('S', 'Solteiro(a)'),
30            ('C', 'Casado(a) ou união estável'),
31            ('D', 'Desquitado/a, separado/a judicialmente ou divorciado/a'),
32            ('V', 'Viúvo/a')]
33    )
34    instrucao = models.IntegerField(

```

Fonte: Autor próprio

A classe “Dados” gerencia as operações de recuperar e persistir os dados das respostas dos participantes. É o modelo dessa classe que torna possível desenvolver a lógica dos dados retornados e enviados para o banco de dados, cada propriedade representa uma coluna lá na relação “participante_dados” conforme figura 10. Em cada propriedade é passado um tipo de campo que recebe as regras estabelecidas no dicionário de dados da Tabela 3.

4.1.5 Django: Framework Python para desenvolvimento do back-end

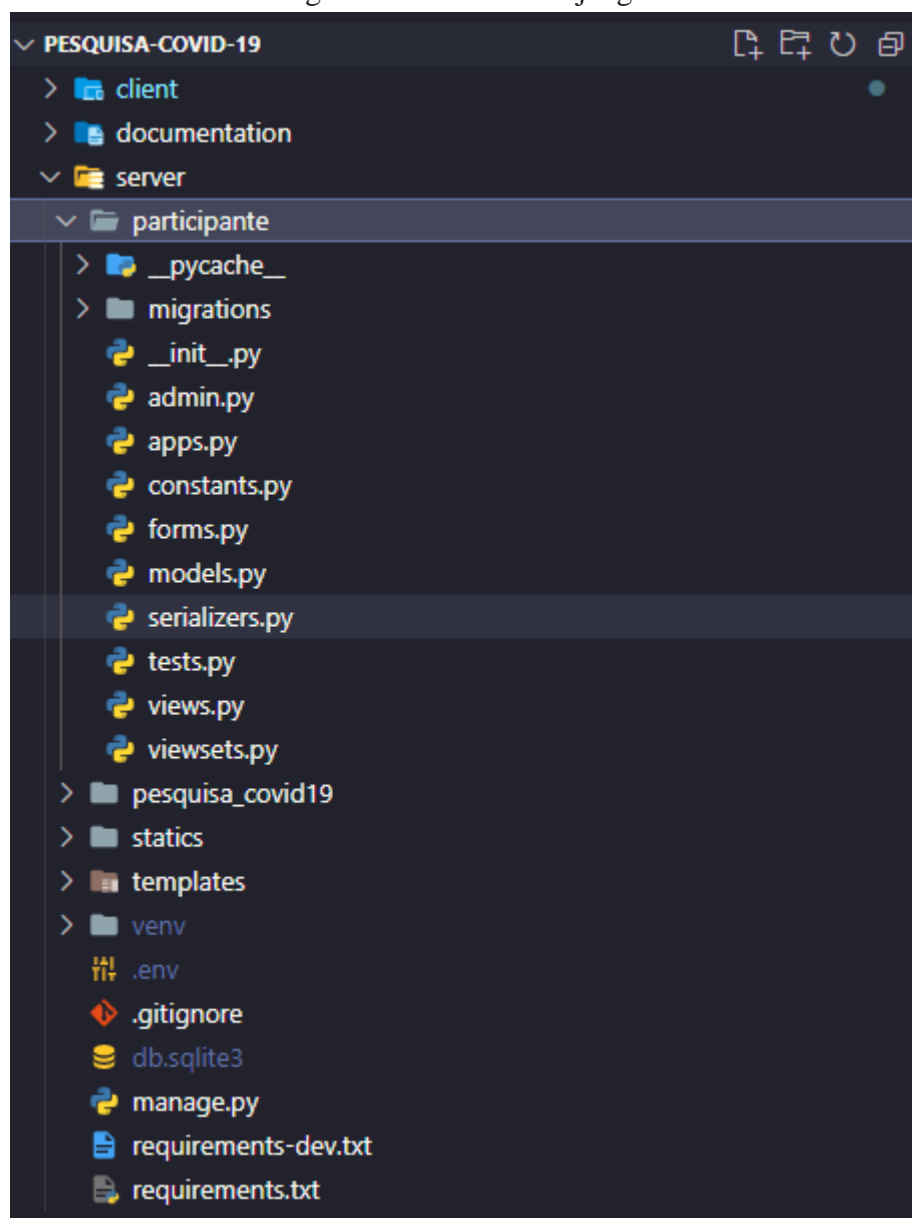
É um Framework onde pode se desenvolver de forma muito rápida, aplicações web, utiliza o Python como base nessa criação.

Atende o usuário de várias formas, desde relação de banco de dados até o retorno de *templates* para o lado do cliente, tem a premissa de facilitar o desenvolvimento de aplicações web com mais qualidade e menos códigos. Fornece base para construir páginas webs bem elaboradas, com ótima qualidade e poucos códigos.

Estendendo as classes do Django foi possível montar a estrutura do pacote “participante”, dessa forma o próprio *framework* se encarregou de “conversar” com o

gerenciador de banco de dados e retornar as interações necessárias para o lado do *front-end*. (Figura 13)

Figura 13 - Estrutura Django.



Fonte: Autor próprio

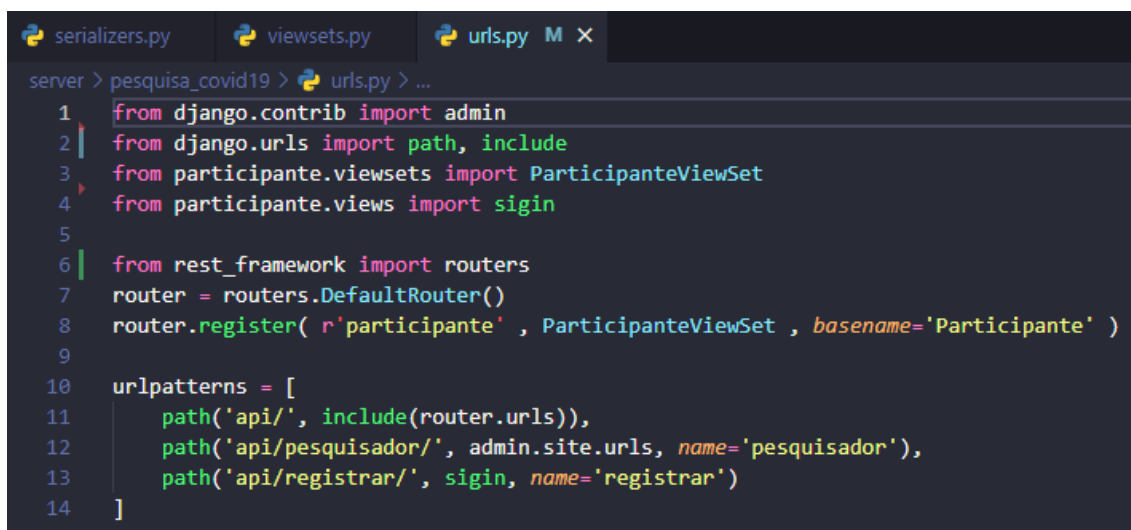
4.1.6 *Django Rest Framework: Framework Python para construção do web service utilizando a arquitetura Rest API*

De acordo com o próprio site (DJANGO-REST-FRAMEWORK, 2021), a estrutura Django REST é um kit de ferramentas poderoso e flexível para a construção de APIs da Web.

Traz uma série de facilidades para criar APIS, provê a parte de serialização, rotas, views.... É um projeto financiado de forma colaborativa tendo como patrocinadores: premium, Sentry, Stream, ESG, Rollbar, Cadre, Kloudless, Lights On Software, Retool e bit.io. A estrutura REST é altamente recomendável e apenas oficialmente compatível com o lançamento de patch mais recente de cada série Python e Django.

Quaisquer configurações globais para uma API de estrutura REST são mantidas em um único dicionário de configuração denominado `REST_FRAMEWORK` (Figura 14).

Figura 14 - Configuração de rotas.



```

server > pesquisa_covid19 > urls.py > ...
1  from django.contrib import admin
2  from django.urls import path, include
3  from participante.viewsets import ParticipanteViewSet
4  from participante.views import signin
5
6  from rest_framework import routers
7  router = routers.DefaultRouter()
8  router.register( r'participante' , ParticipanteViewSet , basename='Participante' )
9
10 urlpatterns = [
11     path('api/', include(router.urls)),
12     path('api/pesquisador/', admin.site.urls, name='pesquisador'),
13     path('api/registrar/', signin, name='registrar')
14 ]

```

Fonte: Autor próprio

No arquivo de configurações de urls é possível determinar os *end-points* da aplicação, todas as requisições vindas do cliente são direcionadas ao endereço “api/”, que serão encaminhadas para o *router*, se “participante” for a chamada da requisição, então é invocado o *viewset* “ParticipanteViewSet”. Na figura 15 tem-se o *viewset* da solução.

Um *viewset* para o *Django Rest Framework*, é um visualizador que não fornece nenhum manipulador de método “get” ou “post”, ao invés disso ele implementa funções que se assemelham as operações básicas utilizadas em base de dados relacionais (CRUD): “create”, “read”, “update”, “delete”.

Como esse sistema só recebe os dados dos participantes, não pode permitir a API a listar, deletar nem atualizar os registros depois de uma submissão de um entrevistado. Abaixo é possível identificar que os métodos *list*, *destroy*, *retrieve* e *update* foram sobrescritos, dessa forma o sistema se torna mais seguro e não permite falhas ou invasões de pessoas mal-intencionadas:

Figura 15 - Viewset.

```

serializers.py  viewsets.py X  urls.py M
server > participante > viewsets.py
1 from rest_framework.viewsets import ModelViewSet
2 from rest_framework.response import Response
3 from .models import Dados
4 from .serializers import DadosSerializer
5
6 class ParticipanteViewSet(ModelViewSet):
7     serializer_class = DadosSerializer
8
9     def get_queryset(self):
10         return Dados.objects.all()
11
12     def list(self, request, *args, **kwargs):
13         return Response(data={'message': 'Requisição não permitida.'}, status=400)
14
15     def create(self, request, *args, **kwargs):
16         return super(ParticipanteViewSet, self).create(request, *args, **kwargs)
17
18     def destroy(self, request, *args, **kwargs):
19         return Response(data={'message': 'Requisição não permitida.'}, status=400)
20
21     # Recupera apenas o item passado por parâmetro
22     def retrieve(self, request, *args, **kwargs):
23         return Response(data={'message': 'Requisição não permitida.'}, status=400)
24
25     def update(self, request, *args, **kwargs):
26         return Response(data={'message': 'Requisição não permitida.'}, status=400)
27
28     # (PATCH) Atualiza parcialmente um objeto, ao invés de atualizar o objeto inteiro como faz o update
29     def partial_update(self, request, *args, **kwargs):
30         return Response(data={'message': 'Requisição não permitida.'}, status=400)

```

Fonte: Autor próprio

Figura 16 - Serializer.

```

serializers.py X  viewsets.py  urls.py M
server > participante > serializers.py > ...
1 from rest_framework.serializers import ModelSerializer, ValidationError
2 from rest_framework import fields
3 from .models import Dados
4 from .constants import MOTIVO_TROCA_USUARIO, APPS_UTILIZADOS, VIRTUAL_ATIVIDADE, VIRTUAL_ATIVIDADE_MOTIVO
5
6 class DadosSerializer(ModelSerializer):
7     class Meta:
8         model = Dados
9         fields = '__all__'
10
11     motivo_troca_usuario = fields.MultipleChoiceField(choices=MOTIVO_TROCA_USUARIO, allow_blank=True, allow_null=True)
12     apps_utilizados = fields.MultipleChoiceField(choices=APPS_UTILIZADOS)
13     virtual_atividade = fields.MultipleChoiceField(choices=VIRTUAL_ATIVIDADE, allow_blank=True, allow_null=True)
14     virtual_atividade_motivo = fields.MultipleChoiceField(choices=VIRTUAL_ATIVIDADE_MOTIVO, allow_blank=True, allow_null=True)
15
16     def validate(self, data):
17         erros = {}
18
19         if data.get('genero', '') == '0' and (data.get('genero_outro', '') == ''):
20             erros['genero_outro'] = 'O campo genero_outro é obrigatório quando o valor "0" é informado em genero.'
21
22         if data.get('instrucao', 0) == 7 and (data.get('instrucao_outro', '') == ''):
23             erros['instrucao_outro'] = 'O campo instrucao_outro é obrigatório quando o valor "7" é informado em instrucao.'
24
25         if data.get('trabalho_estudo', 0) == 6 and (data.get('trabalho_estudo_outro', '') == ''):
26             erros['trabalho_estudo_outro'] = 'O campo trabalho_estudo_outro é obrigatório quando o valor "6" é informado em trabalho_estudo.'
27
28         if data.get('funcao_unifeso', 0) == 7 and (data.get('funcao_unifeso_espec', '') == ''):
29             erros['funcao_unifeso_espec'] = 'O campo funcao_unifeso_espec é obrigatório quando o valor "7" é informado em funcao_unifeso.'
30
31         if data.get('tipo_conexao', 0) == 5 and (data.get('tipo_conexao_espec', '') == ''):
32             erros['tipo_conexao_espec'] = 'O campo tipo_conexao_espec é obrigatório quando o valor "5" é informado em tipo_conexao.'
33
34         if data.get('dispositivo_usado', 0) == 4 and (data.get('dispositivo_usado_espec', '') == ''):
35             erros['dispositivo_usado_espec'] = 'O campo dispositivo_usado_espec é obrigatório quando o valor "4" é informado em dispositivo_usado.'

```

Fonte: Autor próprio.

A figura 16 apresenta o se serializador da aplicação. Os serializadores no *Django Rest Framework* permite a tradução de estruturas de dados complexas, como *QuerySets* e instância de modelos e classes a serem convertidos em tipos de dados nativos do Python, com o intuito de facilitar a manipulação. No DRF o *serializer* também serve como uma “desserialização”, transformando os tipos nativos do Python em uma estrutura compatível com os dados complexos.

4.1.7 Javascript ES6: Linguagem de programação utilizada para construção do *Front-end*

JavaScript é uma das linguagens de programação *front-end* mais utilizadas. Ele permite interação entre clientes e usuários e apresenta recursos exclusivos comparado com outras linguagens.

JavaScript (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinamicamente fraca e multiparadigma (protótipos, orientado a objeto, imperativo e, funcional). Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da *World Wide Web*. JavaScript permite páginas da Web interativas e, portanto, é uma parte essencial dos aplicativos da web. A grande maioria dos sites usa, e todos os principais navegadores têm um mecanismo JavaScript dedicado para executá-lo.

Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido, porém os mecanismos JavaScript agora estão incorporados em muitos outros tipos de software host, incluindo em servidores e bancos de dados da Web e em programas que não são da Web, como processadores de texto e PDF, e em tempo de execução ambientes que disponibilizam JavaScript para escrever aplicativos móveis e de desktop, incluindo *widgets* de área de trabalho.

4.1.8 Node.JS 14x: Software utilizado para executar código JavaScript no *back-end*

Node.js é um interpretador de JavaScript assíncrono com código aberto orientado a eventos, criado por Ryan Dahl em 2009, focado em migrar a programação do Javascript do cliente (*front-end*) para os servidores, criando aplicações de alta escalabilidade (como

um servidor web), manipulando milhares de conexões/eventos simultâneas em tempo real numa única máquina física.

O Node.JS (ambiente de execução Javascript no servidor) foi implementado baseado no interpretador V8 JavaScript Engine (interpretador de JavaScript em C++ com código aberto do Google, utilizado no Chrome), com desenvolvimento mantido pela fundação Node.js em parceria com a Linux Foundation.

O Node.JS se fez necessário nesse projeto porque através dele foi possível manipular e gerenciar o desenvolvimento do *front-end* com o React.JS.

4.1.9 ReactJs: Biblioteca JavaScript para desenvolvimento do *front-end*

É uma biblioteca JavaScript criada pelo Facebook para o desenvolvimento de aplicações *front-end*, essa ferramenta ajuda no desenvolvimento de sites e páginas e pode ser utilizada com outras ferramentas para agregar mais recursos às aplicações web e mobile.

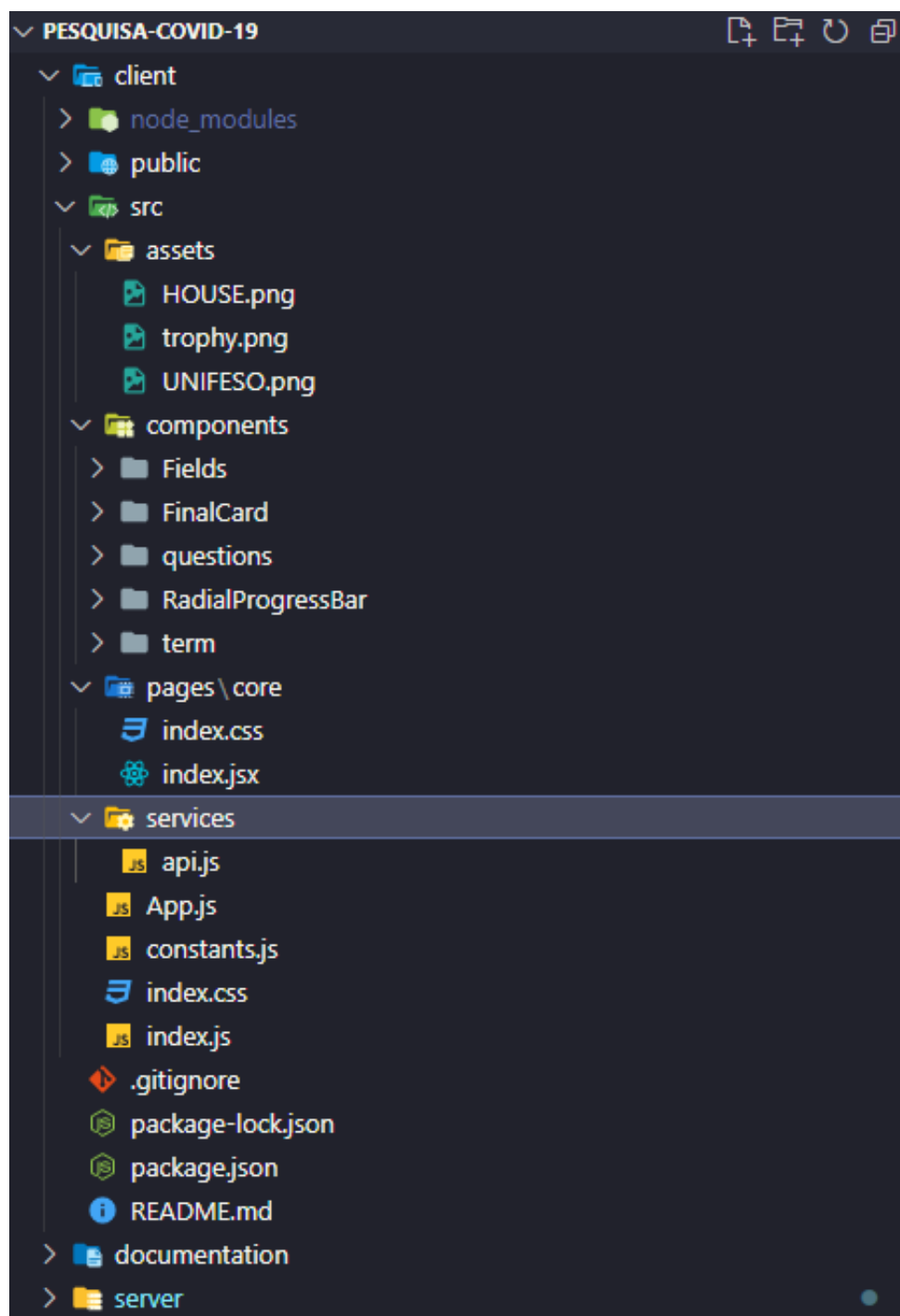
O React pode ser utilizada com qualquer tipo de linguagem de *back-end*, a aplicação pode ser em PHP, Java etc. Além disso, ela funciona em conjunto com outras bibliotecas ou frameworks JavaScript, como *Bootstrap* e *JQuery*. Possui algumas funcionalidades que o tornam uma biblioteca eficiente para o desenvolvimento de aplicações *front-end* com elementos interativos como: Virtual DOM, que é uma representação estruturada de todos os elementos da página e é utilizado para facilitar as alterações nesses elementos; JSX, uma extensão para o JavaScript, oferece algumas funcionalidades a mais e permite escrever de forma simples JavaScript e HTML no mesmo código. A principal característica do React é permitir a criação de componentes que podem ser reutilizados em diversas páginas, além de ser fácil de aprender, permite a reutilização de componentes, apresenta fácil implementação entre outras. Para utilizar o React é preciso ter conhecimento em HTML, JavaScript e CSS.

Para agilizar o desenvolvimento desse projeto foi utilizado o *framework front-end React Bootstrap 4.0*, com ele foi possível criar componentes visuais mais bonitos, elegantes e intuitivos.

O projeto do *front-end* foi organizado utilizando uma estrutura bem popular aceita por desenvolvedores de toda a comunidade. Todo arquivo programável é centralizado dentro da pasta “src”, a pasta “node_modules” é onde é armazenado os pacotes do React e bibliotecas de terceiros, e dentro da pasta “public” existe o HTML global da aplicação,

configurações, ícones e recursos via CDN são alterados nessa pasta. Na raiz do *front-end* possui o arquivo “package.json” criado pelo “npm” (Gerenciador de pacotes do Node.JS), nesse arquivo (Figura 17) são configuradas as dependências do projeto, versão, e execução de *scripts*. A aplicação dessa solução é *Single Page*, ou seja, possui uma única página, na qual os elementos visuais na tela são alterados de maneira controlada pelos estados.

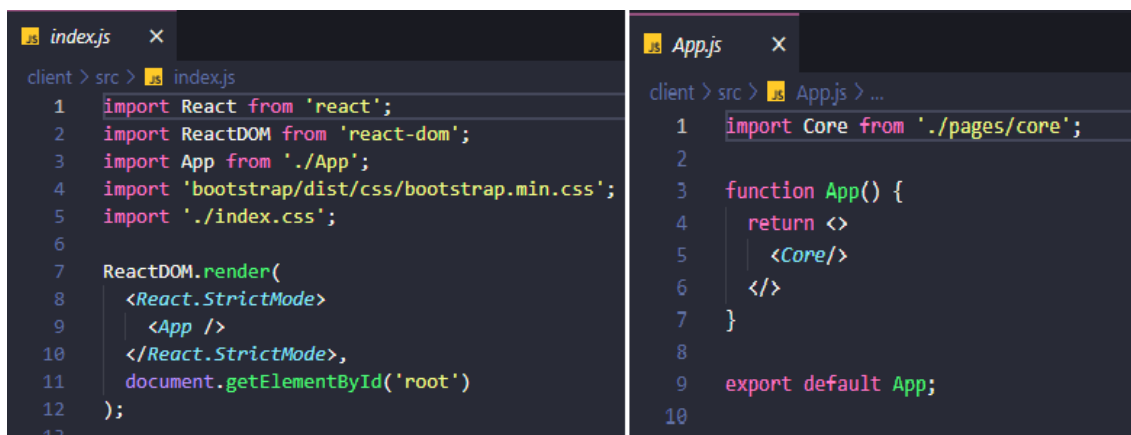
Figura 17 - Estrutura *front-end*.



Fonte: Autor próprio

A pasta “assets” (Figura 17) possui recursos de mídia, como imagens, logos, áudios e etc. Na “componets” (Figura 17) é onde os componentes do sistema foram criados, tudo para o React é um componente, então, um botão, um campo de entrada de texto, uma barra de progresso é componente. Foi necessário criar esses recursos antes de montar o formulário. (Figura 18 e 19)

Figura 18 - Arquivos de entrada do React.



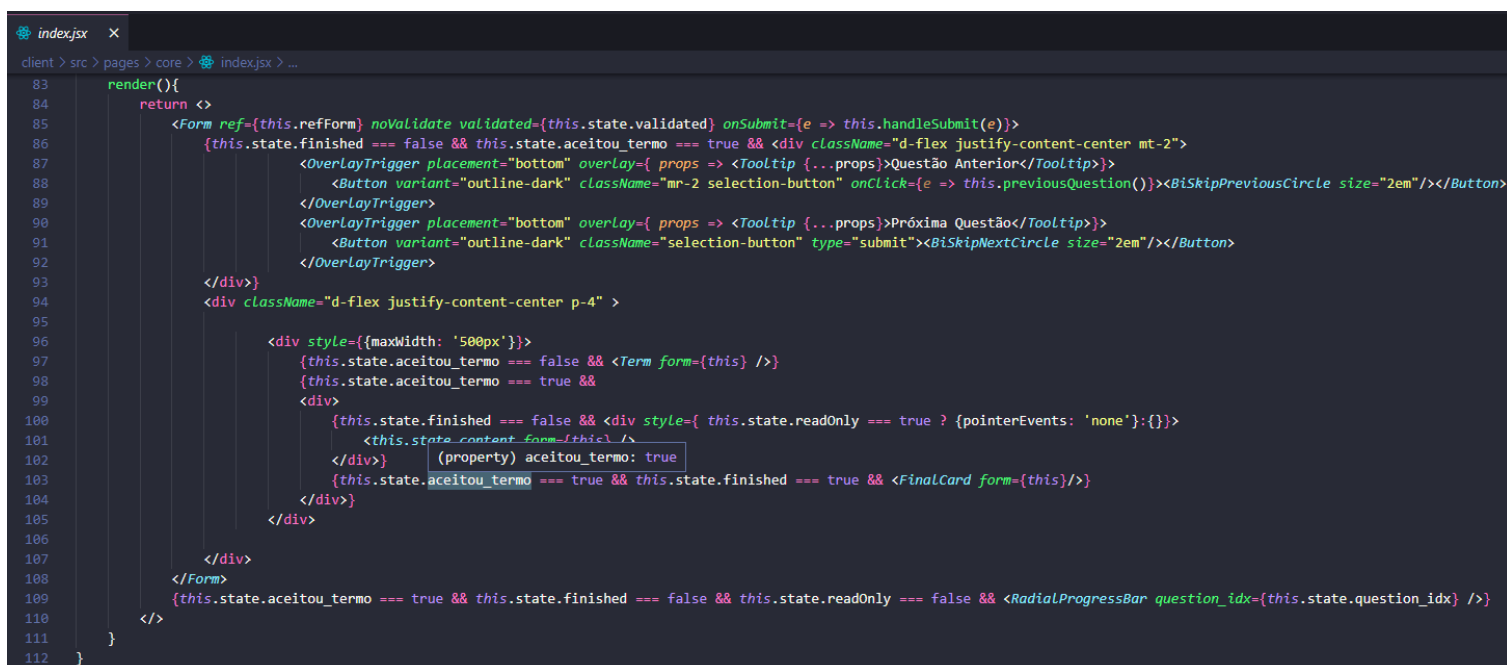
```

client > src > index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import App from './App';
4 import 'bootstrap/dist/css/bootstrap.min.css';
5 import './index.css';
6
7 ReactDOM.render(
8   <React.StrictMode>
9     <App />
10   </React.StrictMode>,
11   document.getElementById('root')
12 );
13

client > src > App.js > ...
1 import Core from './pages/core';
2
3 function App() {
4   return <>
5     <Core/>
6   </>
7 }
8
9 export default App;
10
  
```

Fonte: Autor próprio

Figura 19 - Função Render da página.



```

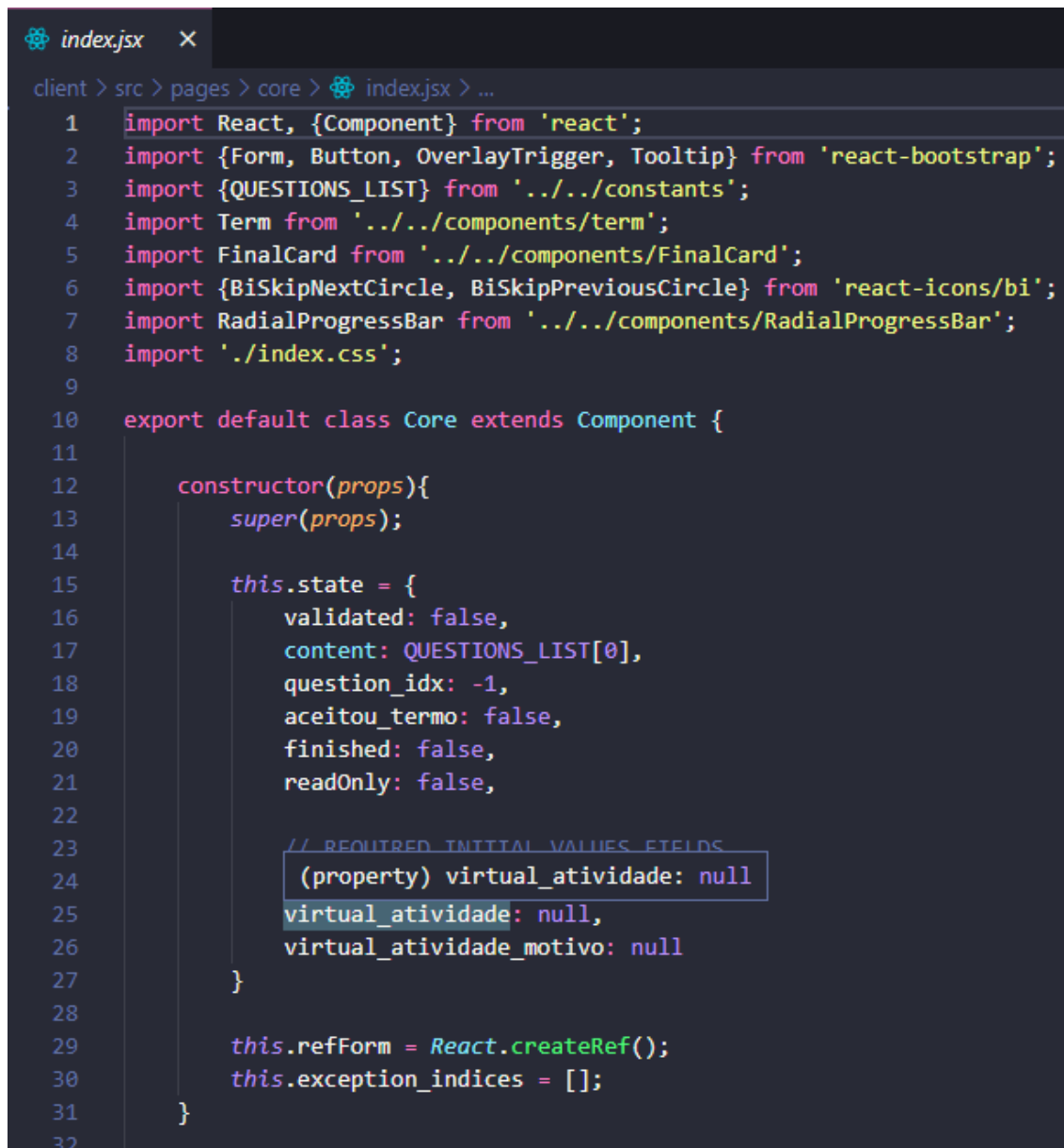
client > src > pages > core > index.jsx > ...
83 render(){
84   return <>
85     <Form ref={this.refForm} noValidate validated={this.state.validated} onSubmit={e => this.handleSubmit(e)}>
86       {this.state.finished === false && this.state.aceitou_termo === true && <div className="d-flex justify-content-center mt-2">
87         <OverlayTrigger placement="bottom" overlay={ props => <Tooltip {...props}>Questão Anterior</Tooltip>>
88           <Button variant="outline-dark" className="mr-2 selection-button" onClick={e => this.previousQuestion()}><BiSkipPreviousCircle size="2em"/></Button>
89         </OverlayTrigger>
90         <OverlayTrigger placement="bottom" overlay={ props => <Tooltip {...props}>Próxima Questão</Tooltip>>
91           <Button variant="outline-dark" className="selection-button" type="submit"><BiSkipNextCircle size="2em"/></Button>
92         </OverlayTrigger>
93       </div>
94       <div className="d-flex justify-content-center p-4">
95         <div style={{maxWidth: '500px'}}>
96           {this.state.aceitou_termo === false && <Term form={this} />}
97           {this.state.aceitou_termo === true &&
98             <div>
99               {this.state.finished === false && <div style={ this.state.readOnly === true ? {pointerEvents: 'none'}: {}>
100                 <this.state.constantForm={this} />
101               </div>
102               (property) aceitou_termo: true
103             </div>
104             {this.state.aceitou_termo === true && this.state.finished === true && <FinalCard form={this}/>}
105           </div>
106         </div>
107       </div>
108     </Form>
109     {this.state.aceitou_termo === true && this.state.finished === false && this.state.readOnly === false && <RadialProgressBar question_idx={this.state.question_idx} />}
110   </>
111 }
112 }
  
```

Fonte: Autor próprio.

No *front-end* foi utilizado como padrão na maioria das páginas, componentes de classe, ou componentes que herdam da classe “*Component*” do React, conforme Figura 20. Dessa forma fica mais fácil sobrescrever o comportamento padrão do React, se

necessário. Utilizando o método de classes, é possível desfrutar de todas as ferramentas que o paradigma de Orientação a Objetos oferece em um projeto *front-end*.

Figura 20 - *Constructor* da classe.



```

1  import React, {Component} from 'react';
2  import {Form, Button, OverlayTrigger, Tooltip} from 'react-bootstrap';
3  import {QUESTIONS_LIST} from '../constants';
4  import Term from '../components/term';
5  import FinalCard from '../components/FinalCard';
6  import {BiSkipNextCircle, BiSkipPreviousCircle} from 'react-icons/bi';
7  import RadialProgressBar from '../components/RadialProgressBar';
8  import './index.css';
9
10 export default class Core extends Component {
11
12     constructor(props){
13         super(props);
14
15         this.state = {
16             validated: false,
17             content: QUESTIONS_LIST[0],
18             question_idx: -1,
19             aceitou_termo: false,
20             finished: false,
21             readOnly: false,
22
23             // REQUIRED INITIAL VALUES FIELDS
24             (property) virtual_atividade: null
25             virtual_atividade: null,
26             virtual_atividade_motivo: null
27         }
28
29         this.refForm = React.createRef();
30         this.exception_indices = [];
31     }
32

```

Fonte: Autor próprio.

O código fonte dessa solução, juntamente com todas as instruções de instalação e execução do ambiente de desenvolvimento está disponível à comunidade através do github no seguinte endereço: <https://github.com/victorveiga/pesquisa-covid-19.git>

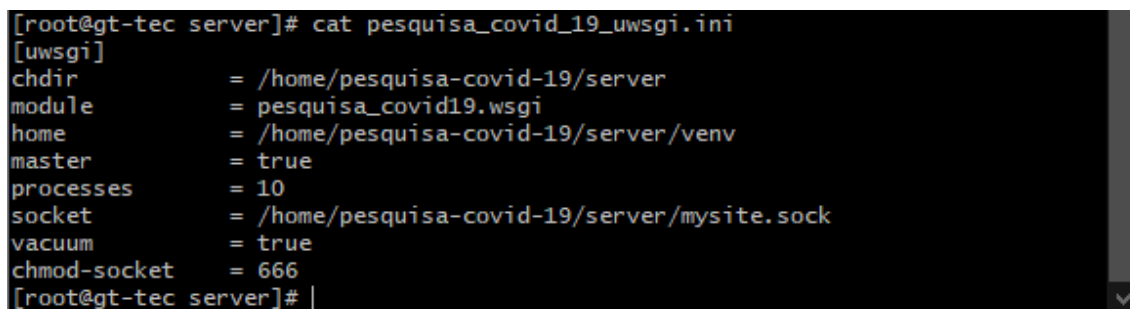
4.1.10 uWSGI: Software para construção de serviços de hospedagem

uWSGI é um aplicativo de software que visa desenvolver uma pilha completa para a construção de serviços de hospedagem

Essa ferramenta é frequentemente usada para servir aplicações web Python em conjunto com servidores web como Cherokee e Nginx, que oferecem suporte direto para o protocolo wsgi nativo do uWSGI.

Para *deploy* da solução nas instalações da TI do Unifeso, foi utilizado o uWSGI juntamente com o Nginx, Na Figura 21, é possível conferir como ficou a configuração do projeto *back-end* na hospedagem:

Figura 21 - Configuração do projeto na hospedagem.



```
[root@gt-tec server]# cat pesquisa_covid_19_uwsgi.ini
[uwsgi]
chdir          = /home/pesquisa-covid-19/server
module         = pesquisa_covid19.wsgi
home           = /home/pesquisa-covid-19/server/venv
master         = true
processes      = 10
socket         = /home/pesquisa-covid-19/server/mysite.sock
vacuum         = true
chmod-socket   = 666
[root@gt-tec server]#
```

Fonte: Autor próprio.

4.1.11 Nginx: Servidor HTTP

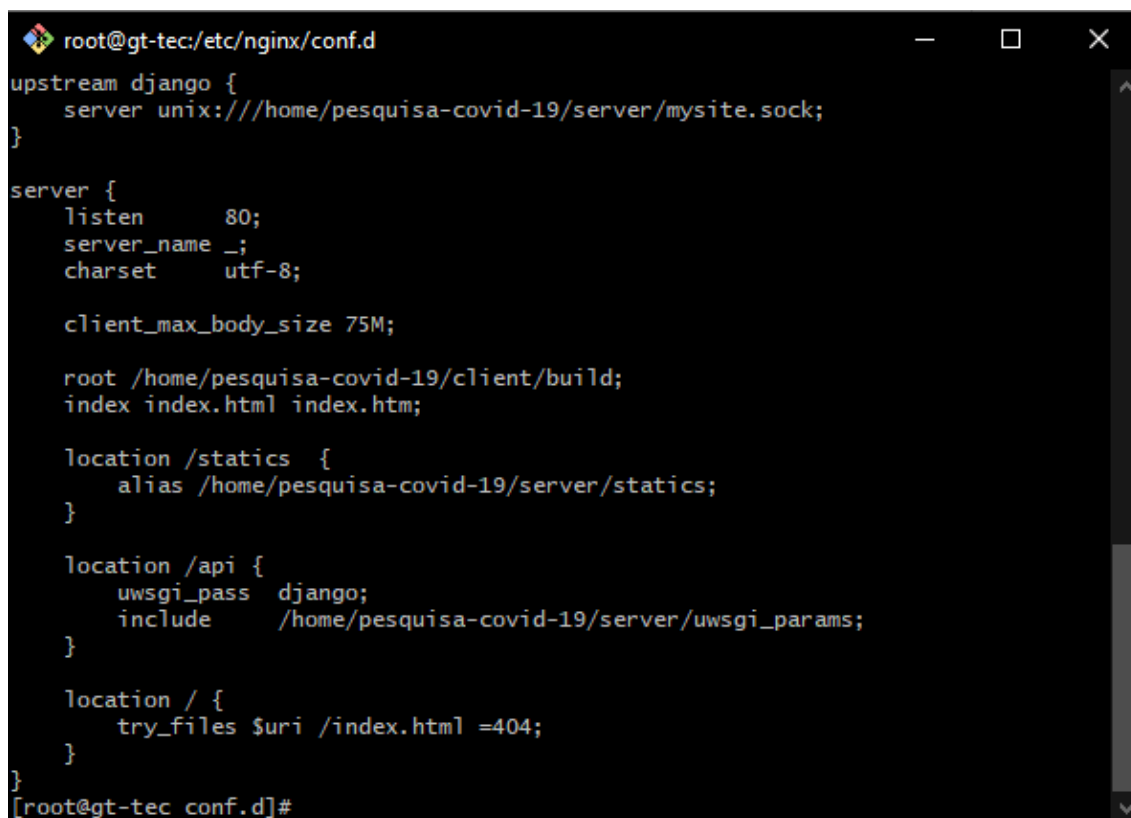
É um servidor leve de HTTP, proxy reverso, proxy de e-mail IMAP/POP3, feito por Igor Sysoev em 2005.

Ele é um serviço, um programa que roda e que serve para responder requisições web, ele ouve uma porta para responder requisições HTTP.

Após receber a requisição, ele analisa e repassa para outros servidores na rede, reduzindo assim a carga sobre outros servidores e consumindo menos memória que o APACHE.

É possível verificar na figura 22, como foi configurado o NGINX. Na porta padrão 80 é servido o serviço do *front-end* e do *back-end*, o que diferencia os serviços são as rotas, para a rota padrão “/” retorna os arquivos estáticos construídos no ReactJs, na rota “/api” é onde recebe as requisições vindas dos participantes da pesquisa. Na rota “/statics” são servidos arquivos estáticos do painel do administrador do Django, na qual foi utilizado para ser a área do pesquisador.

Figura 22 - Configuração NGINX.

A terminal window with a dark background and light-colored text. The title bar shows 'root@gt-tec:/etc/nginx/conf.d'. The content is an NGINX configuration file. It starts with an 'upstream django' block pointing to a Unix socket. Then a 'server' block is defined, listening on port 80, with server_name '_' and charset 'utf-8'. It sets 'client_max_body_size 75M;'. The 'root' is set to '/home/pesquisa-covid-19/client/build;' and the 'index' is 'index.html index.htm;'. There are three 'location' blocks: '/statics' pointing to '/home/pesquisa-covid-19/server/statics;', '/api' using 'uwsgi_pass django;' and including '/home/pesquisa-covid-19/server/uwsgi_params;', and a default location '/' that tries files and returns 404 if not found. The prompt at the bottom is '[root@gt-tec conf.d]#'.

```
root@gt-tec:/etc/nginx/conf.d
upstream django {
    server unix:///home/pesquisa-covid-19/server/mysite.sock;
}

server {
    listen      80;
    server_name _;
    charset     utf-8;

    client_max_body_size 75M;

    root /home/pesquisa-covid-19/client/build;
    index index.html index.htm;

    location /statics {
        alias /home/pesquisa-covid-19/server/statics;
    }

    location /api {
        uwsgi_pass  django;
        include     /home/pesquisa-covid-19/server/uwsgi_params;
    }

    location / {
        try_files $uri /index.html =404;
    }
}
[root@gt-tec conf.d]#
```

Fonte: Autor próprio

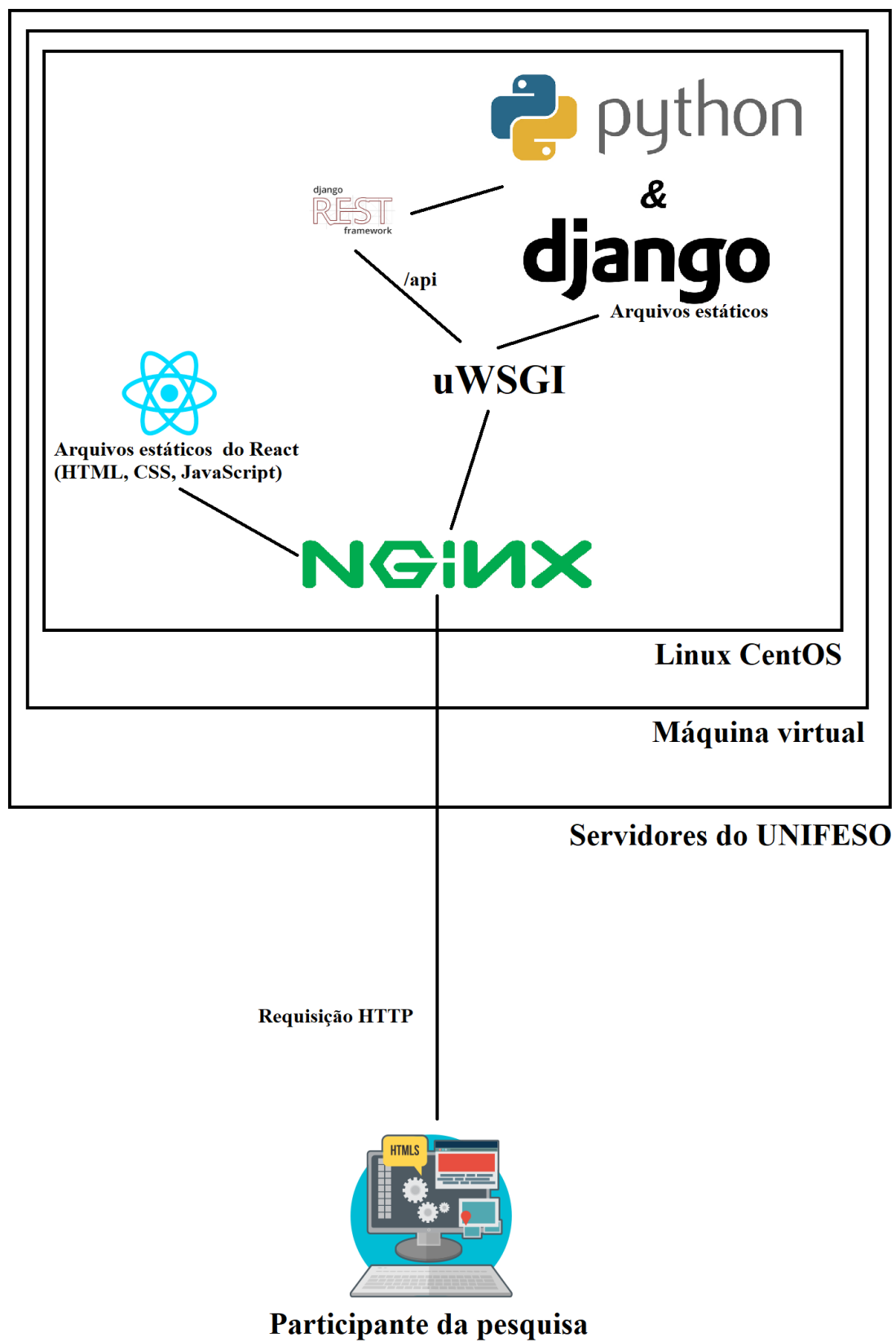
4.2 HOSPEDAGEM

Para que todos os serviços ficassem online e disponível 24 horas por dia para que qualquer pessoa com o link conseguisse ter acesso a pesquisa, foi necessário utilizar uma hospedagem de sites ou aplicativos. Existe inúmeras maneiras de colocar um app online, mas a maioria das soluções possui um custo muito alto. Para que a solução do projeto ficasse disponível foi fornecido pela TI do Unifeso, uma máquina virtual dentro de seus servidores, dessa forma foi possível realizar todo o processo de deploy, sem nenhum custo, já que a hospedagem do projeto está sendo mantido pela instituição.

A máquina virtual disponibilizada possui 4GB de RAM, com 100Gib de espaço de armazenamento, com 3 núcleos de processamento sobre o processador “Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz”.

Na figura 23, é possível ter uma visão geral da estrutura construída para solução e todas as tecnologias utilizadas.

Figura 23 - Estrutura da solução.



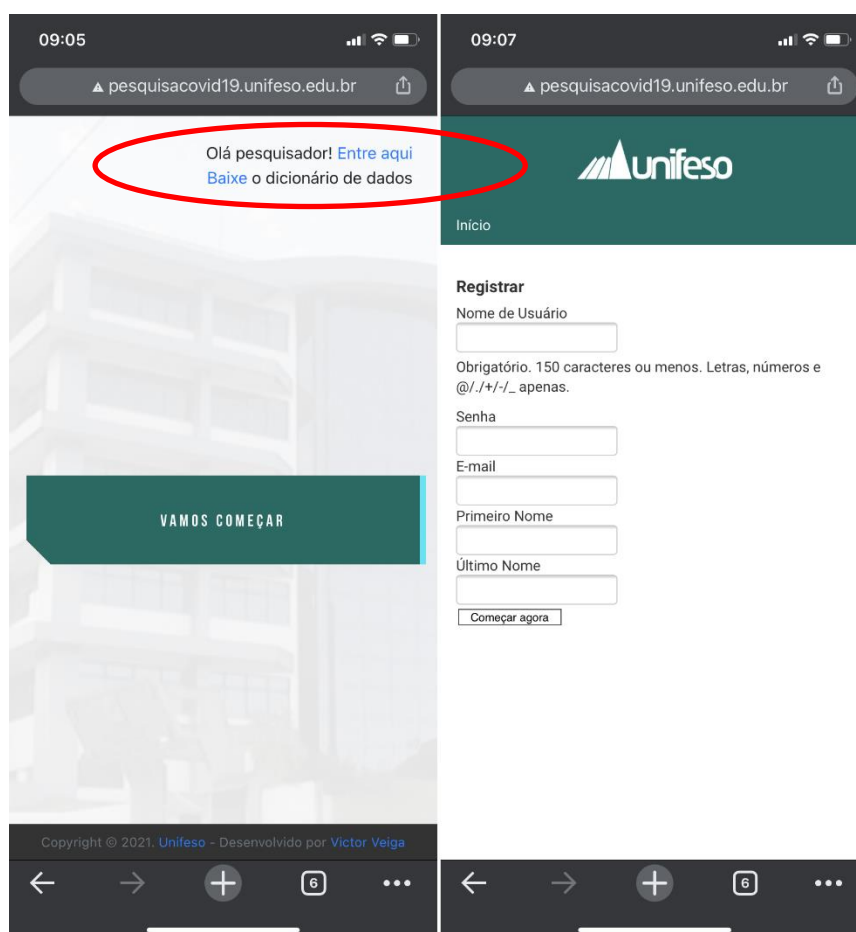
Fonte: Autor próprio.

5 CONCLUSÃO E TRABALHOS FUTUROS

O intuito da solução implementada nesse projeto era disponibilizar de maneira online um questionário de pesquisa no campus do Unifeso, e a partir disso construir um banco de dados acessível a comunidade. Conforme o esperado, todos os objetivos foram alcançados. A pesquisa se iniciou no dia 16 de setembro de 2021. Com o apoio da diretoria do CCT, da diretoria da DPPE e do *Marketing* do Unifeso, foi enviado aos estudantes, professores e colaboradores um convite incentivando a participarem da pesquisa.

O formulário ainda permanece online para receber submissões. Foi liberado uma área do pesquisador, onde é possível ter acesso aos dados em um arquivo com formato CSV, basta ele realizar um simples cadastro e entrar na área de Pesquisa e Administração. No canto superior direito é possível entrar na área do pesquisador ou baixar o dicionário de dados, conforme Figura 24.

Figura 24 - Área do Pesquisador.



Fonte: Autor próprio.

Esse projeto apenas implementou através de uma solução tecnológica sistêmica a pesquisa “Projeto Integrado Sobre os Impactos das Medidas de Isolamento e Distanciamento Social pela Covid-19”. Analisar os dados obtidos não é o propósito desse trabalho, o maior objetivo é incentivar e contribuir para a ampliação da produção acadêmica dos estudantes do Unifeso.

6 REFERÊNCIAS BIBLIOGRÁFICAS

FARIAS, Cátia Araujo; FERREIRA, Luciana Nunes; SOARES, Richard Pereira; VEIGA, Victor Lima. DESENVOLVIMENTO DE SOLUÇÃO SISTÊMICA PARA GESTÃO DE DADOS EM PESQUISA CIENTÍFICA: PROJETO INTEGRADO SOBRE OS IMPACTOS DAS MEDIDAS DE ISOLAMENTO E DISTANCIAMENTO SOCIAL PELA COVID-19. CONFESO, [s. l.], ano 2020, ed. 5, 20 nov. 2020.

HARABIŠ, FILIP; HRONKOVÁ, JANA. European database of the life-history, morphological and habitat characteristics of dragonflies (Odonata). *European Journal of Entomology*, v. 117, n. 1, p. 302-308, 14 jul. 2020.

SILVA, Joan D. S; CALÇADA, Jâina Carolina Meneses; REZENDE, Solange Oliveira; CALÇADA, Dario Brito. Automatic identification of knowledge related to dengue cases in the state of Piauí in public databases using filtered-association rules networks. *Revista de Informática Teórica e Aplicada - RITA*, Porto Alegre, Instituto de Informática da UFRGS, v. 27, n. 3, p. 40-49, 2020.

BERG, K. L.; SEYMOUR, T.; GOEL, R. History Of Databases. *International Journal of Management & Information Systems (IJMIS)*, v. 17, n. 1, p. 29-36, 31 Dec. 2012.

KUHNEN, I. A. Análise de sistemas de gerenciamento de banco de dados para armazenamento de dados climáticos. Cuiabá, 2016, 59f. Dissertação (Mestrado em Física Ambiental) - Instituto de Física, Universidade Federal de Mato Grosso.

VASCONCELLOS, LILIANA; GUEDES, LUIS FERNANDO ASCENÇÃO. E-Surveys: Vantagens e Limitações dos Questionários Eletrônicos via Internet no Contexto da Pesquisa Científica. *SemeAd - Seminários em Administração FEA/USP*, [S. l.], p. 1-16, 18 ago. 2007.

MOTA, Janine da Silva. UTILIZAÇÃO DO GOOGLE FORMS NA PESQUISA ACADÊMICA. *Revista Humanidades e Inovação*, Palmas-Tocantins, ano 2019, v. 6, n. 12, 9 set. 2019. Relatos de Experiência, p. 372-380.

DOWNS, ROBERT T. DOWNS; HALL-WALLACE, MICHELLE. The American Mineralogist crystal structure database. *American Mineralogist*, [S. l.], ano 2003, v. 88, p. 247-250, 2 jan. 2003.

MARIUCCI, Maria Teresa; MONTONE, Paola. Database of Italian present-day stress indicators, IPSI 1.4. *Nature*, [S. l.], ano 2020, n. 298, 8 set. 2020. *Scientific Data*, p. 1-11.

ANHANGUERA. O que é e qual é a diferença entre Front-End e Back-End? Anhanguera, 2021. Disponível em: <<https://blog.anhanguera.com/o-que-e-front-end-e-back-end/>>. Acesso em: 29 Outubro 2021.

BRITO, E. PostgreSQL 12 lançado com várias melhorias de desempenho. *Edivaldobrito.com.br*, 2019. Disponível em:

<<https://www.edivaldobrito.com.br/postgresql-12-lancado-com-varias-melhorias-de-desempenho/>>. Acesso em: 29 Outubro 2021.

POSTGRESQL. PostgreSQL 12.8 Documentation. postgresql.org, 2021. Disponível em: <<https://www.postgresql.org/docs/12/index.html>>. Acesso em: 29 Outubro 2021.

GETTING help / Documentation. wiki.centos.org, 2021. Disponível em: <<https://wiki.centos.org/Documentation>>. Acesso em: 30 Outubro 2021.

WIKIPEDIA. CentOS. Wikipedia, 2021. Disponível em: <<https://pt.wikipedia.org/wiki/CentOS>>. Acesso em: 30 Outubro 2021.

CLARK, M. Top 10 Linguagens de Programação para backend. Back4app, 2021. Disponível em: <<https://blog.back4app.com/pt/top-10-linguagens-de-programacao-para-backend/>>. Acesso em: 01 Novembro 2021.

STACK OVERFLOW. 2020 Developer Survey. Stack Overflow, 2020. Disponível em: <<https://insights.stackoverflow.com/survey/2020>>. Acesso em: 01 Novembro 2021.

KIM DESMOND. What is Python used for? Top 5 Python uses. CodingNomads, 2020. Disponível em: <<https://codingnomads.co/blog/python/what-is-python-used-for-python-uses>>. Acesso em: 1 Novembro 2021.

PYTHON. Python 3.6.15 documentation. Python, 2021. Disponível em: <<https://docs.python.org/3.6/>>. Acesso em: 1 Novembro 2021.

DJANGO. Documentação do Django. djangoproject, 2021. Disponível em: <<https://docs.djangoproject.com/pt-br/3.1/>>. Acesso em: 1 Novembro 2021.

DJANGO-REST-FRAMEWORK. Django Rest Framework. django-rest-framework.org, 2021. Disponível em: <<https://www.django-rest-framework.org/>>. Acesso em: 1 Novembro 2021.

MDN WEB DOCS. JavaScript. MDN Web Docs, 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 2 Novembro 2021.

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. JavaScript. Wikipédia, 2021. Disponível em: <<https://pt.wikipedia.org/wiki/JavaScript>>. Acesso em: 2 Novembro 2021.

DEVMEDIA. Guia de Node.js. Devmedia, 2021. Disponível em: <<https://www.devmedia.com.br/guia/node-js/40312>>. Acesso em: 2 Novembro 2021.

NODEJS. About Node.js. nodejs.org, 2021. Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 2 Novembro 2021.

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. Node.js. Wikipédia, 2021. Disponível em: <<https://pt.wikipedia.org/wiki/Node.js>>. Acesso em: 2 Novembro 2021.

FREIRES, N. Um guia para usar React JS. geekhunter, 2019. Disponível em: <<https://blog.geekhunter.com.br/um-guia-para-usar-react-js/>>. Acesso em: 3 Novembro 2021.

REACT. Introdução. React, 2021. Disponível em: <<https://pt-br.reactjs.org/docs/getting-started.html>>. Acesso em: 3 Novembro 2021.

NGINX. Nginx Documentation. Nginx, 2021. Disponível em: <<https://nginx.org/en/docs/>>. Acesso em: 3 Novembro 2021.

UWSGI. The uWSGI project. uWSGI, 2021. Disponível em: <<https://uwsgi-docs.readthedocs.io/en/latest/>>. Acesso em: 3 Novembro 2021.