

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ELIMAR SANCHES KAUFFMANN

**GERENCIAMENTO DE PROJETO DE SOFTWARE PARA ESTAÇÕES
SOLARIMÉTRICAS E FOTOVOLTAICAS IMPLANTADAS NOS CAMPI DA
UTFPR**

CURITIBA

2022

ELIMAR SANCHES KAUFFMANN

**GERENCIAMENTO DE PROJETO DE SOFTWARE PARA ESTAÇÕES
SOLARIMÉTRICAS E FOTOVOLTAICAS IMPLANTADAS NOS CAMPI DA
UTFPR**

**SOFTWARE PROJECT MANAGEMENT FOR SOLARIMETRIC AND
PHOTOVOLTAIC STATIONS IMPLEMENTED ON THE CAMPUSES OF UTFPR**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação do Curso de Bacharelado em Sistemas de Informação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. José Antonio Buiar

Coorientador: Prof. Dr. Winderson Eugenio dos Santos

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ELIMAR SANCHES KAUFFMANN

**GERENCIAMENTO DE PROJETO DE SOFTWARE PARA ESTAÇÕES
SOLARIMÉTRICAS E FOTOVOLTAICAS IMPLANTADAS NOS CAMPI DA
UTFPR**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Bacharel do Curso de Bacharelado em Sistemas
de Informação da Universidade Tecnológica Federal
do Paraná (UTFPR).

Data de aprovação: 04/Julho/2022

Luiz Augusto Pelisson
Mestrado
Universidade Tecnológica Federal do Paraná

Jorge Assade Leludak
Doutorado
Universidade Tecnológica Federal do Paraná

Jose Antonio Buiar
Doutorado
Universidade Tecnológica Federal do Paraná

CURITIBA

2022

AGRADECIMENTOS

O autor agradece à Terezinha Pinheiro, sua tia de consideração, que lhe deu total suporte durante seu início de curso, na cidade de Curitiba-PR. Ao Formação Solidária, que por meio de seu cursinho preparatório, o Cursinho Solidário, pode estudar de forma adequada para entrar no curso que almejava. À Fundação Casa do Estudante Universitário do Paraná, que lhe deu abrigo e inúmeras aprendizagens, podendo participar da gestão da Casa.

Agradece à Universidade Tecnológica Federal do Paraná, que possibilitou a realização de um sonho, sendo o primeiro da família a ter uma formação superior, e não qualquer formação, mas uma das melhores do Brasil, em uma Federal, com professores incríveis, aos quais o autor também é grato.

Também à UTFPR pelo apoio e infraestrutura disponibilizados para o desenvolvimento desta pesquisa e à COPEL-Distribuição pelo apoio e financiamento dos recursos para realização deste projeto de P&D “ANEEL P&D 2866-0464/2017 - Metodologia Para Análise, Monitoramento e Gerenciamento da GD por Fontes Incentivadas”.

RESUMO

Devido ao alto consumo das matrizes energéticas na geração de diversos tipos de energias, pesquisadores e entidades governamentais procuram formas alternativas para suprir essa demanda na utilização de combustíveis fósseis. No Brasil, através de um incentivo da ANEEL, projetos de pesquisas estão em desenvolvimento nas universidades e institutos federais com o objetivo de estimular novas formas para a produção de energia. Na UTFPR, um projeto já implantado utilizando Estações Solarimétricas e Fotovoltaicas, em parceria com a COPEL, tem como uma de suas bases a busca para o encaminhamento de políticas públicas mais assertivas voltadas ao setor elétrico. Com isso, este estudo tem como objetivo analisar o gerenciamento do projeto de um software responsável pelo monitoramento e gerenciamento dessas estações que hoje estão em funcionamento nesta instituição.

Palavras-chave: Gestão de Projeto. Estações Solarimétricas. Estações Fotovoltaicas.

ABSTRACT

Owing to the high consumption of energy matrices in the generation of various types of energy, researchers and government entities are looking for alternative ways to meet this demand in the use of fossil fuels. In Brazil, through an incentive from ANEEL, research projects are under development at universities and federal institutes with the aim of stimulating new ways for energy production. At UTFPR, a project already implemented using Solarimetric and Photovoltaic Stations, in partnership with COPEL, has as one of its bases the search for forwarding more assertive public policies aimed at the electric sector. Thus, this study aims to analyze the project management of a software responsible for monitoring and managing these stations that are currently in operation in this institution.

Keywords: Project Management. Solarimetric Stations. Photovoltaic Stations.

LISTA DE FIGURAS

Figura 1 – Exemplo de modelagem usando o BR Modelo Web.	24
Figura 2 – Termos utilizados na pesquisa do Estado da Arte.	26
Figura 3 – Diagrama de trabalho.	35
Figura 4 – Exemplo de consulta de dados utilizando o aplicativo Planilhas Google.	45
Figura 5 – Tela inicial do software para as estações solarimétricas e fotovoltaicas pronto.	49
Figura 6 – Grupos de Usuários.	51
Figura 7 – Exemplo de grupo de usuário - perfil Admin (Gestor).	51
Figura 8 – Tela de criação de Chave de API.	52
Figura 9 – Exemplo de Chave Gerada.	52
Figura 10 – Tela de Cricação de Comando para Inversores.	53
Figura 11 – E-mail de confirmação de comando para inversor.	53
Figura 12 – E-mail de cancelamento de comando para inversor.	53
Figura 13 – Diagrama Entidade-Relacionamento	76
Figura 14 – Diagrama Lógico	77

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ANEEL	Agência Nacional de Energia Elétrica
API	Interface de Programação de Aplicações
As Builts	Como construído
BD	Banco de Dados
CdTe	Telureto de Cádmio
CERN	Centro Europeu de Pesquisa Nuclear
CIGS	Cobre-Índio-Gálio
CNAE	Classificação Nacional de Atividades Econômicas
CSV	Comma-separated values
Cybersecutit	Segurança de computadores
Datasheet	Folha de Especificações
DER	Diagrama de Entidade e Relacionamento
EPE	Empresa de Pesquisa Energética
EPE	Empresa de Pesquisa Energética
EPESOL	Estação de Pesquisa em Energia Solar
ER	Entidade-Relacionamento
FTP	Protocolo de Transferência de Arquivo (do inglês, File Transfer Protocol)
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
Information Security	Segurança da informação
INPE	Instituto Nacional de Pesquisas Espaciais
IP	Internet Protocol

ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LABENS	Laboratório de Energia Solar
m-Si	Silício Monocristalino
MER	Modelo Entidade e Relacionamento
p-Si	Silício Policristalino
P&D	Pesquisa e Desenvolvimento
ROL	Receita Operacional Líquida
SFVCR	Sistema Fotovoltaico Conectado à Rede
SGBD	Sistema Gerenciador de Banco de Dados
SONDA	Sistema de Organização Nacional de Dados Ambientais
SONDA	Sistema de Organização Nacional de Dados Ambientais
SPE	Superintendência de Pesquisa e Desenvolvimento e Eficiência Energética
SQL	Structured Query Language, ou Linguagem de Consulta Estruturada
TD	Taxa de desempenho
TI	Tecnologia da Informação
UML	Unified Modeling Language
URL	Uniform Resource Locator
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 – INTRODUÇÃO	12
2 – REFERENCIAL TEÓRICO	14
2.1 Energia Solar no Brasil	14
2.1.1 Estações Solarimétricas	14
2.1.2 Usinas Fotovoltaicas	15
2.2 Modelagem Computacional	15
2.3 Software	16
2.3.1 Aplicações para a Web	17
2.3.2 Sistema de aquisição de dados	18
2.3.3 Sistema de Monitoramento	19
2.3.4 Interface	19
2.3.5 Interfaces Amigáveis	19
2.3.6 Cybersecurity e Information Security	19
2.3.7 API – Application Programming Interface	19
2.3.8 Modelagem	20
2.3.9 Casos de Uso	21
2.3.10 Requisitos Funcionais	21
2.3.11 Requisitos Não Funcionais	22
2.3.12 Modelagem de Dados	22
2.3.13 Diagrama de Entidade e Relacionamento	23
2.4 Gestão de Projetos de Software	24
2.4.1 Gestão de Projeto	24
2.4.2 Projeto de Software	25
2.4.3 Gerente de Projetos de Software	25
2.4.4 Ciclo de Vida de Projetos de Software	26
2.5 Estado da Arte	26
3 – PROJETO LABENS	29
3.1 Histórico	29
3.1.1 Projeto UTFPR	29
3.1.2 ANEEL - Chamada Nº.001/2016	30
3.1.3 COPEL - Chamada Pública COPEL 001/2017	31
3.2 Laboratório de Energia Solar LABENS	31
3.3 Aplicação	32

4 – METODOLOGIA	34
4.1 Partes envolvidas	35
4.2 Entendimento lógico e físico do projeto	36
4.3 Entendimento e levantamento dos requisitos de software	38
4.4 Requisitos e prazos a serem repassados à empresa contratada	39
4.5 Levantamento dos requisitos da empresa para desenvolver	40
4.6 Busca da empresa para desenvolver o sistema	41
4.7 Triagem das empresas e Contratação	41
4.8 Alinhamento do escopo do projeto, prioridades e cronogramas em definição com a empresa a ser contratada	42
4.9 Interface entre LABENS e Empresa contratada	42
4.10 Materiais e definições de apoio	43
4.11 Especificação para Exportação de Dados Utilizando APIs	44
4.12 Especificações quanto à Segurança	46
5 – RESULTADOS	48
5.1 Gerenciamento de Projeto de Software	48
5.2 Acompanhamento do Desenvolvimento do Sistema	49
5.3 Especificações quanto à Segurança e à Exportação de Dados	50
6 – CONSIDERAÇÕES FINAIS	55
Referências	56
Apêndices	60
APÊNDICE A – Reuniões Realizadas	61
APÊNDICE B – Diagramas de Casos de Uso	72
APÊNDICE C – Análise de Requisitos	74
C.1 Requisitos Funcionais	74
C.2 Requisitos Não Funcionais	75
APÊNDICE D – Diagrama Entidade-Relacionamento	76
APÊNDICE E – Modelo Lógico do Banco de Dados	77

1 INTRODUÇÃO

Com o alto consumo das matrizes energéticas sendo utilizadas para suprir as mais diversas necessidades dos processos de consumo e produção, a energia elétrica passa a ser fundamental para a subsistência do ser humano. A forma predominante de produção desse tipo de energia parte de fontes não renováveis, conforme Dutra et al. (2013). Entretanto, esse formato de produção provoca sérios impactos ambientais percebidos pela população mundial e comprovados por cientistas.

Na busca de um modelo sustentável e renovável de fonte de energia, novas iniciativas vêm sendo pesquisadas, como a energia solar fotovoltaica, que é uma forma de geração de energia capaz de suprir algumas necessidades com certa vantagem em relação às formas tradicionais de geração (PEREIRA, 2008).

Por ser bem desenvolvida tecnologicamente para a produção de energia elétrica, a energia solar fotovoltaica no Brasil, nos últimos anos, vem superando alguns desafios do país, mostrando certa evolução no setor (OLIVEIRA; PERIN, 2016). Essas conquistas, em parte, se devem à obrigação de investimento em P&D no setor elétrico pela lei nº 9.991/2000, a qual determina que as empresas concessionárias, permissionárias e autorizadas do setor de energia elétrica apliquem anualmente um percentual de sua Receita Operacional Líquida (ROL) em pesquisa e desenvolvimento (P&D).

A Universidade Tecnológica Federal do Paraná (UTFPR) foi uma das contempladas com essa modalidade de investimento (COPEL, 2017b), passando a instalar Estações Solarimétricas e Fotovoltaicas em alguns de seus campi. Nesse contexto tornou-se necessária a elaboração de um modelo computacional que possibilite o desenvolvimento de um sistema para monitoramento e gerenciamento remotos destas estações.

O sistema deve possibilitar a professores e pesquisadores o acompanhamento da geração de energia, enquanto compara com informações referentes ao clima, como vento, umidade, insolação, entre outras, permitindo entender melhor a correlação entre o ambiente e a geração de energia, bem como otimizar a manutenção das estações pelo acompanhamento em tempo real, com alertas pré-programados, histórico de manutenção, credenciamento e controle de usuários do sistema.

Este trabalho tem como objetivo analisar o gerenciamento de um projeto de software que permita monitorar, de forma remota e presencial, funcionalidades e processos baseados nos componentes e nas tecnologias já presentes em equipamentos instalados no projeto do LABENS, que hoje está em funcionamento nos campi da UTFPR, de acordo com o projeto “Eficiência Energética e Minigeração em Instituições Públicas de Educação Superior” publicado pela ANEEL (Agência Nacional de Energia Elétrica).

A partir desse objetivo principal, são definidos os seguintes objetivos específicos:

- Organizar requisitos existentes e analisar se é necessário levantar novos, além de acompanhar o desenvolvimento do sistema para verificar se os requisitos estão sendo cumpridos e realizar o trabalho de intermediação entre usuários e desenvolvedores do sistema;
- Especificar formas para exportação de dados e informações relevantes aos pesquisadores do LABENS, utilizando APIs de consulta de dados;
- Especificar formas para a segurança da aplicação, o sistema de monitoramento e a API.

Este trabalho está organizado na forma de capítulos. O Capítulo 2 apresenta de forma breve os conceitos teóricos necessários. o Capítulo 3 apresenta de forma geral o Laboratório de Energia Solar - LABENS e a organização do projeto desenvolvido neste laboratório. Por sua vez, o Capítulo 4 descreve a metodologia utilizada no desenvolvimento deste trabalho. O Capítulo 5 apresenta os resultados atingidos e, por fim, no Capítulo 6 estão apresentadas as considerações finais.

2 REFERENCIAL TEÓRICO

2.1 Energia Solar no Brasil

Quando comparado ao da Europa, o índice de radiação solar no Brasil é muito bom. Na região do litoral norte do estado de Santa Catarina se encontra o menor nível de irradiação do país, na ordem de $1.500 \text{ kWh}/m^2$ de irradiação global anual na superfície horizontal. Enquanto que o maior verificado é no norte do estado da Bahia, na ordem de $2.350 \text{ kWh}/m^2$ de irradiação global horizontal anual. Na Alemanha, um dos países líderes na geração de energia por fonte solar, sua incidência varia de 1.200 a $1.950 \text{ kWh}/m^2$ ao ano (OLIVEIRA; PERIN, 2016). Comparando-se a produtividade estimada total anual média da Alemanha com a do Paraná, percebe-se que este estado possui média de 58,75% superior à Alemanha (TIEPOLO, 2015).

Pelo fato do Brasil possuir grandes Bacias Hidrográficas em várias partes de sua extensão territorial, a principal fonte geradora de energia elétrica é a hidráulica (64,1%), vindo em seguida o gás natural (13%), biomassa (7,3%), derivados de petróleo (6,9%), carvão e seus derivados (3,2%), nuclear (2,5%) e eólico (2,0%), existindo, portanto, muito espaço de crescimento da participação da energia solar na matriz elétrica (TEM-PASS; SOUZA; IATSKIU, 2016). Potencial este que não se pode subestimar, pois, inclusive, há uma demanda reprimida de consumidores ávidos por terem uma central microgeradora fotovoltaica (SANTOS; URBANETZ; GOUVEA, 2018).

2.1.1 Estações Solarimétricas

As Estações Solarimétricas são conjuntos de equipamentos que mensuram parâmetros solares, como radiação solar global, direta e difusa, entre outras, a depender das tecnologias instaladas. Existe uma quantidade mínima de equipamentos necessários para se considerar uma estação de monitoramento solar completa, sendo eles: um pireliômetro para medir irradiância direta normal (G_n), um piranômetro para aquisição da irradiância horizontal global (G) e um piranômetro sombreado para medir a irradiância horizontal difusa (G_{dif}), além disso, para a aquisição de dados G_n e G_{dif} é necessário um rastreador solar (PEREIRA et al., 2017).

Por a energia solar ter uma alta variabilidade temporal, e a mesma estar associada aos padrões meteorológicos atuantes, é importante conhecer o comportamento climatológico da região de interesse para a exploração dessas fontes de energia. No Brasil, ainda existe necessidade de instalações de estações solarimétricas distribuídas, pois o número existente não supre a crescente demanda de medição dos recursos solares. Além do número baixo, existe restrição quanto à disponibilidade dos dados brutos (LIRA; SOARES; SANTOS, 2016).

2.1.2 Usinas Fotovoltaicas

Estações Fotovoltaicas são conjuntos de painéis fotovoltaicos, que se estabelecem em conjuntos de módulos solares, com a função de capturar e transformar a radiação solar em eletricidade. Para essa transformação acontecer, os fótons contidos na luz solar são convertidos em energia elétrica, por meio do uso de células solares (ANEEL, 2005).

A geração fotovoltaica de energia elétrica no Brasil é mais comum nas regiões Norte e Nordeste, em comunidades isoladas da rede de energia elétrica (ANEEL, 2005). Em dezembro de 2018, dos 1,8 GW de projetos centralizados de geração solar fotovoltaica neste país, o Nordeste sediava 69,2%, e dos 0,5 GW em geração distribuída, sediava cerca de 19,0% (BEZERRA, 2018).

Essa geração de energia renovável, por ser um meio de geração de energia limpa e infinita – devido o sol ser sua fonte –, de baixa agressão ao meio ambiente e que propicia a redução do impacto ambiental, vem sendo cada vez mais explorada e recebendo maiores investimentos, tornando-se mais viável e acessível a toda a população Rodrigues (2018).

2.2 Modelagem Computacional

Os modelos são criados para uma melhor compreensão do que será realmente construído. Quando for algo físico, pode-se fazer algo igual, porém em menor escala, como uma maquete de um prédio. Mas quando a entidade a ser construída for um software, o modelo deve assumir uma forma diferente. É necessário representar as informações que o software transforma, a arquitetura e as funções que possibilitam essa transformação, as características que os usuários desejam e o comportamento do sistema durante as transformações. Os objetivos que devem ser cumpridos em dois níveis de abstração são: 1) Descrição do software – do ponto de vista do cliente; 2) Descrição mais técnica do software, em etapa posterior (PRESSMAN, 2011).

No trabalho de engenharia de software, podem ser criadas duas classes de modelos: os modelos de requisitos, que representam os requisitos dos clientes (descrevendo em três domínios: da informação, funcional e comportamental); e os modelos de projeto, que oferecem uma especificação concreta para a construção do software, especificando a arquitetura, a interface para o usuário e os detalhes quanto aos componentes (PRESSMAN, 2011).

Pressman (2011) descreve 5 princípios da modelagem de requisitos e 9 princípios da modelagem de projetos.

Princípios da modelagem de requisitos:

1. O universo de informações de um problema deve ser representado e compreendido;
2. As funções que o software desempenha devem ser definidas;
3. O comportamento do software, como consequência de eventos externos, deve ser representado;

4. Os modelos que descrevem informações, funções e comportamentos devem ser divididos para que revelem detalhes por camadas, ou hierarquicamente;
5. A análise deve partir da informação essencial para o detalhamento da implementação.

Princípios da modelagem de projetos:

1. O projeto deve ser roteirizado para a modelagem de requisitos;
2. Sempre considere a arquitetura do sistema a ser construído;
3. O projeto de dados é tão importante quanto o projeto das funções de processamento;
4. As interfaces, tanto internas quanto externas, devem ser projetadas com cuidado;
5. O projeto de interface do usuário deve ser voltado às necessidades do usuário final. Entretanto, em todo caso, deve enfatizar a facilidade de uso;
6. O projeto no nível de componentes deve ser funcionalmente independente;
7. Os componentes devem ser relacionados livremente tanto entre componentes quanto com o ambiente externo;
8. Representações de projetos, ou modelos, devem ser de fácil compreensão;
9. O projeto deve ser desenvolvido interativamente. A cada interação, o projetista deve se esforçar para obter maior grau de simplicidade.

2.3 Software

Pressman (2011) menciona que se fosse fazer uma definição mais simplificada, similar às usadas em livro-texto, teria-se que o software consiste em 3 pontos: 1) instruções (programas de computador) que, quando executadas, devem executar ações desejadas; 2) uma estrutura adequada de dados, que permite uma melhor manipulação das informações; 3) descrições de operações de uso e de uso dos programas. Até seria possível buscar maiores definições, mais extensas sobre o que é, mas não melhoraria de forma considerável o entendimento do que é software. Para tal, é necessário observar suas características que o diferem de outras construções do ser humano. Por ser mais um elemento de um sistema lógico do que físico, tem características que são distintas daquelas do hardware (PRESSMAN, 2011).

Outra peculiaridade que vale ressaltar em relação ao software é que ele não se desgasta, mas sim, se deteriora Pressman (2011). O que acontece, ao contrário do hardware, é que ele possui maiores taxas de erros no seu início de vida, que são corrigidos ao longo do tempo, achatando a curva de erros. Entretanto, muitas modificações são feitas para suprir novas realidades, e essas mudanças inserem erros que também deverão ser corrigidos, de modo que essas modificações vão deteriorando o software.

No caso do hardware, é possível trocar uma peça com problemas por completo por uma igual, nova e funcional, já no caso de software, não. Pois se tem algo errado, foi por erro de projeto, seja por falta de alguma especificação, ou por outro motivo. Ou seja, sua correção será algo específico e particular para aquele problema. Desta forma, este trabalho visa justamente trazer um projeto com essa modelagem para que esses erros

sejam evitados na proposta abordada.

Para Pressman (2011), a maioria dos softwares continua a ser construída de forma personalizada, sob encomenda, o que, inclusive, é a realidade do que está sendo desenvolvido neste trabalho, ou seja, um software personalizado para as necessidades do LABENS.

2.3.1 Aplicações para a Web

No início da década de noventa, um cientista chamado Tim Berners-Lee desenvolveu o princípio da World Wide Web. Sua principal motivação fora melhorar a maneira em que seus colegas do Centro Europeu de Pesquisa Nuclear (CERN) acessavam às informações que se encontravam espalhadas entre os diferentes sistemas que usavam. As atividades de busca que deveriam ser simples se tornavam complexas, pois precisavam acessar diferentes computadores e ter distintos conhecimentos para manipular todos esses sistemas (BERNERS-LEE, 1996).

Para solucionar esse problema, buscou-se desenvolver um sistema que unificasse o modo de acesso, não importando o local físico de onde se iria acessar, a plataforma e até mesmo o tipo de informação desejada. Sistema o qual teria como padrão o acesso feito por um único programa navegador (*Web Browser*), construído sobre uma infraestrutura simples, sem estado e rápida, através do protocolo de transferência de hipertexto, ou HTTP (*Hyper Text Transfer Protocol*). Além disso, Berners-Lee definiu uma linguagem que seria reconhecida pelo navegador, servindo também como forma de estruturação e apresentação das informações: o HTML (*Hyper Text Markup Language*) (BERNERS-LEE, 1996).

Pressman (2011) lista sete grandes categorias de softwares, os quais apresentam desafios contínuos para os engenheiros de software: 1) Software de Sistema – conjunto de programas feitos para atender outros programas; 2) Software de aplicação – programas sob medida que solucionam uma necessidade específica do negócio; 3) Software científico/da engenharia – tem sido caracterizado por algoritmos "number crunching" (para "processamento numérico pesado"); 4) Software embutido – residente num produto ou sistema e utilizado para implementar características e funções para o usuário final e para o próprio sistema; 5) Software para linha de produtos – projetado para prever capacidade específica de utilização por muitos clientes diferentes; 6) Software de inteligência artificial – faz uso de algoritmos não numéricos para solucionar problemas complexos que não são passíveis de computação ou de análise direta e 7) Aplicações para a Web – chamadas de "WebApps", essa categoria de software centralizada de redes abarca uma vasta gama de aplicações.

De forma mais simplificada, entende-se que as WebApps podem ser pouco mais que um conjunto de arquivos de hipertexto interconectados, os quais apresentam informações gráficas limitadas. Mas que com as novas tecnologias, como a Web 2.0, já se transformaram em sofisticados ambientes computacionais. Deixando de somente prover recursos especiali-

zados, funções computacionais e conteúdo para os usuários finais, mas também passaram a integrar bancos de dados corporativos e aplicações comerciais (PRESSMAN, 2011).

2.3.2 Sistema de aquisição de dados

É importante entender o que é um dado e como ele se difere de uma informação para que haja maior compreensão do que são os sistemas de aquisição de dados. Oliveira (2006) descreve dado como sendo qualquer elemento identificado em sua forma bruta que, por si só, não conduz à compreensão de determinado fato ou situação. De modo que o dado, para um entendimento humano, precisa passar por uma transformação. Oliveira (2006) explica que informação é o dado trabalhado que permite ao executivo tomar uma decisão. Ou seja, a transformação necessária para o entendimento de um fato ou uma situação. Essa transformação ocorre por meio de um sistema de informações, no qual se realizam processos de transformação de dados em informações (OLIVEIRA, 2006).

Sistema é a disposição das partes (ou elementos) de um todo coordenadas entre si, que funcionam como uma estrutura organizada (FERREIRA, 1994). Para Oliveira (2006), a definição é similar, em que o sistema é um conjunto de partes interagentes e interdependentes que, juntamente, formam um todo unitário com determinado objetivo e efetuam uma função específica.

Para um melhor entendimento do que é um sistema, Oliveira (2006) explica os 6 componentes de um sistema: 1) Os objetivos dos usuários e do próprio sistema; 2) As entradas do sistema, que devem estar em sintonia com os objetivos; 3) O processo de transformação do sistema, função em que dada uma entrada, é gerada uma saída; 4) As saídas do sistema, que são os resultados do processo de transformação; 5) Os controles e as avaliações do sistema, o que necessita de um padrão, podendo ser avaliado se as saídas estão coerentes com os objetivos; 6) A retroalimentação (*feedback*) do sistema, visando tornar o sistema autorregulador.

O objetivo dos Sistemas de Aquisição de Dados é controlar a aquisição dos dados, de modo a permitir o usuário a parametrizar, comandar e monitorar o processo de aquisição dos dados. Esses sistemas armazenam os sinais captados na forma de arquivos que, quando necessário, podem ser consultados posteriormente. Esses sistemas também podem possibilitar a visualização e edição dos dados, bem como a geração de relatórios e outros documentos (PANHAN, 2002).

Os dados coletados de sistemas elétricos podem ser, após armazenados no banco de dados, lidos em diferentes formatos, contudo, para lê-los, é importante tornar o processo de suas leituras independentes do restante da ferramenta, permitindo maiores modificações, sem interferência no restante do projeto (AGOSTINI, 2002).

2.3.3 Sistema de Monitoramento

Sistema de Monitoramento é um programa de computador que contribui na aquisição de dados, de forma total ou parcial, podendo ser composto de vários módulos, em que cada um possui sua finalidade (PANHAN, 2002).

2.3.4 Interface

Representa o conjunto de dispositivos, métodos e filosofias responsáveis por conectar e integrar o humano e a máquina – de forma mais precisa, o computador (PANHAN, 2002).

2.3.5 Interfaces Amigáveis

Define-se por interfaces amigáveis a representação da informação exibida por diversas mídias, tais como voz, sons, imagens, animação, texto, da maneira mais simples, auto explicável e agradável ao usuário (PANHAN, 2002).

2.3.6 Cybersecurity e Information Security

“Information Security” é um termo maior que se preocupa com a confidencialidade, integridade e disponibilidade dos dados (OLCOTT, 2019). O termo “Cybersecurity” ou “Cyberspace Security” é referido pela ISO/IEC 27032 de 2012 como proteção de privacidade, integridade e acessibilidade de informações no ciberespaço. Sendo o ciberespaço reconhecido como uma interação de pessoas, software e serviços tecnológicos mundiais (ISO, 2012). Ou seja, o primeiro engloba tanto o que é digital quanto o que não é, já o segundo, que é o nosso escopo, foca somente no digital. É importante identificar onde estão os dados mais críticos, que devem ser protegidos, e quais os riscos de serem comprometidos ou atacados, assim como a tecnologia que será usada para a proteção deles (OLCOTT, 2019).

2.3.7 API – Application Programming Interface

Hunter (2017), em seu livro, "*Irresistible APIs – Designing Web APIs That Developers Will Love*", traz que uma API (Application Programming Interface ou, em tradução, Interface de Programação de Aplicações) é uma interface em um sistema de computador. Sendo que, historicamente, as APIs começaram como interfaces altamente acopladas entre computadores e sistemas. As APIs da web, que são muito mais livres e menos vinculadas, vêm evoluindo por algum tempo, mas recentemente os desenvolvedores viram uma grande explosão nas web APIs disponíveis para eles. Mas muitas dessas APIs foram desenvolvidas sem pensar no usuário final, gerando resultados frustrantes. O termo API tem sido usado na maior parte da história da computação para se referir às interfaces entre sistemas de computador ou entre diferentes programas em um sistema

existente. Frequentemente, esses sistemas eram pares, em que nenhum dos sistemas era especificamente um servidor ou um cliente. Por exemplo, um servidor de correio pode ter usado um banco de dados para armazenar informações, mas os sistemas foram inerentemente projetados em conjunto, fortemente acoplados, trabalhando juntos sem problemas. Mais recentemente, o termo foi expandido para incluir web APIs, sistemas em que um cliente – que pode ser qualquer coisa, desde um navegador da web até um celular aplicativo – entra em contato com um servidor da web e opera nos dados desse servidor. A principal diferença aqui é que os desenvolvedores que estão escrevendo os clientes não são os mesmos que estão escrevendo a interface – o sistema é realmente desacoplado.

Jin, Sahni e Shevat (2018), de forma similar, trazem que uma API é a interface que um programa de software apresenta a outros programas, para humanos e, no caso de APIs da web, para o mundo via a internet. O design de uma API desmente muito o programa por trás – modelo de negócios, recursos do produto, um bug ocasional. Embora as APIs sejam projetadas para funcionar com outros programas, são principalmente destinadas a serem entendidas e usadas por humanos que escrevem esses outros programas. APIs são os blocos de construção que permitem a interoperabilidade para grandes plataformas de negócios na web. APIs são como contas criadas e mantidas em todas as contas de software em nuvem de uma empresa, ações que ajudam a pedir pizza, por exemplo. APIs são como os dados de previsão do clima, em que são compartilhados de uma fonte respeitável como o National Weather Service a centenas de aplicativos de software especializados em suas apresentações. As APIs processam cartões de crédito e permitem que as empresas possam coletar seu dinheiro sem problemas, sem se preocupar com questões financeiras, suas leis e regulamentações correspondentes à sua região.

2.3.8 Modelagem

Sommerville (2011) aborda que, possivelmente, o maior problema que se enfrenta no desenvolvimento de sistemas de software é o de engenharia de requisitos. Esta se relaciona com a definição do que o sistema deverá fazer, as restrições de desenvolvimento, as propriedades essenciais e desejáveis e a operação do sistema.

O objetivo da engenharia de requisitos é prover para as partes um entendimento escrito do problema. Entretanto, entender os requisitos está entre as tarefas mais complicadas enfrentadas por um engenheiro de software. Entendimento o qual pode ser obtido por intermédio de alguns artefatos: cenários de uso, listas de funções e características, modelos de análise ou uma especificação (PRESSMAN, 2011).

A atividade de levantamento de requisitos, que também é conhecida como elicitación de requisitos, corresponde a uma etapa de compreensão do escopo do problema aplicado ao desenvolvimento do software, tendo como principal objetivo possibilitar a mesma visão do problema a ser resolvido, tanto pelo usuário quanto pelo desenvolvedor. Nessa fase, desenvolvedores e clientes, em conjunto, buscam levantar e definir as necessidades dos

futuros usuários do sistema, sendo comum que esses requisitos sejam escritos em uma notação informal, com linguagem natural (BEZERRA, 2015).

2.3.9 Casos de Uso

O criador dos casos de uso, Ivar Jacobson, traz como definição que eles são sequências de ações que são executadas pelos sistemas e que produzem um resultado de valor observável para um ator. Nesse ínterim, um ator pode ser qualquer agente externo (entidade) que, de alguma forma, tem interação com o sistema. Eles descrevem detalhadamente as funcionalidades do sistema, listam as ações que o sistema realizará, mapeiam o escopo e facilitam a comunicação com os envolvidos. Os casos de uso é um dos diagramas da UML (*Unified Modeling Language*). Por meio dessa modelagem, pode-se obter uma visão do relacionamento e interdependência entre os casos de uso, de forma a facilitar a documentação (JACOBSON, 2003).

Usa-se o diagrama da UML para a representação gráfica dos casos de uso e para definir os relacionamentos entre eles. O agente do sistema é representado por um ator, usam-se elipses para representar os casos de uso, e os relacionamentos entre eles são representados por linhas pontilhadas (JACOBSON, 2003).

Os diagramas de casos de uso da UML são usados para modelar os requisitos funcionais e também são descritos e detalhados no documento de especificação de casos de uso, usando uma linguagem formal e técnica. A composição deles se dá por fluxos de eventos, os quais descrevem o comportamento de um caso de uso dada uma sequência de comandos declarativos que especifica as etapas de execução deste dado caso. O modelo de documento de especificação de requisitos pode ser criado e adaptado por cada organização, seguindo as suas necessidades de uso (JACOBSON, 2003).

2.3.10 Requisitos Funcionais

Os requisitos funcionais de um sistema são declarações de serviços que o sistema deverá prover e como será seu comportamento em algumas situações (descreve o que o sistema deverá fazer) (SOMMERVILLE, 2011). Esses requisitos descrevem o comportamento do sistema, as ações que terá para cada uma das entradas, ou seja, aquilo que o sistema deve fazer (COSTA; DIAS, 2017).

Bezerra (2015) traz que os requisitos funcionais definem as funcionalidades do sistema, já apresentando três exemplos: 1) "O sistema deve permitir que cada professor realize o lançamento de notas das turmas nas quais lecionou."; 2) "O sistema deve permitir que um aluno realize a sua matrícula nas disciplinas oferecidas em um semestre letivo."; 3) "Os coordenadores da escola devem poder obter o número de aprovações, reprovações e trancamentos em cada disciplina oferecida em um determinado período".

2.3.11 Requisitos Não Funcionais

Os requisitos não funcionais de um sistema são as restrições sobre os serviços e as funções oferecidas pelo sistema (SOMMERVILLE, 2011), sendo que eles expressam como o procedimento deve ser feito. Para tal, relacionam-se com padrões de qualidade, tais como, robustez, confiabilidade, performance, e possuem grande importância por definirem se um sistema será eficiente ou não no que se propõe. Nos requisitos não funcionais também se apresentam restrições e especificações de uso para os requisitos funcionais (COSTA; DIAS, 2017).

Bezerra (2015) apresenta os requisitos não funcionais, como aqueles que declaram as características de qualidade que o sistema deve possuir e que estão relacionadas às suas funcionalidades. Trazendo cinco tipos: 1) Confiabilidade: que corresponde a medidas quantitativas da confiabilidade do sistema, tais como tempo médio entre as falhas, recuperação de falhas ou quantidade de erros por milhares de linhas de código-fonte; 2) Desempenho: requisitos que definem os tempos de resposta esperados para as funcionalidades do sistema; 3) Portabilidade: restrições quanto às plataformas de hardware e software nas quais o sistema será implantado e quanto ao grau de facilidade para transportar o sistema para outras plataformas; 4) Segurança: limitações sobre a segurança do sistema em relação a acessos não autorizados; 5) Usabilidade: requisitos que se relacionam ou afetam a usabilidade do sistema. Exemplos incluem requisitos sobre a facilidade de uso e a necessidade ou não de treinamento dos usuários.

2.3.12 Modelagem de Dados

A modelagem é uma maneira de representação do ambiente observado por meio de um modelo, de forma a permitir a análise dos relacionamentos entre os objetos estudados, a formação do conceito desses objetos, a abstração das características comuns entre eles e o agrupamento em conjunto, viabilizando a validação da organização dos dados e tornando essa organização mais estável (RASLAN; CALAZANS, 2014).

Também é possível ver a modelagem como a exposição de como as informações se relacionam por meio das notações e dos diagramas, em que se pode tirar as inconsistências de dados e redundâncias. Inclusive, o não uso dessa ferramenta pode implicar em maiores gastos e complexidade no desenvolvimento, dado o crescimento desorganizado dos dados (BENCK; GURA, 2011).

A modelagem de dados para retratar a realidade do estudo necessita, como uma primeira etapa, realizar o levantamento e a análise dos requisitos, visando entender e documentar a maneira em que os dados de interesse são obtidos. Assim, cria-se um esquema conceitual de alto nível (RASLAN; CALAZANS, 2014). Para Elmasri e Navathe (2005), o modelo de dados conceitual é uma descrição objetiva dos requisitos de dados, tendo também descrições detalhadas dos tipos de entidades, relacionamentos e descrições, para,

então, com o auxílio de um SGBD, seguir na etapa de implementação real do projeto no banco de dados.

Heuser (1998) explica dois modelos: o Modelo Conceitual, que é um modelo mais abstrato de dados, o qual descreve a estrutura de um banco de dados independente de sua classificação. Ou seja, suas características, seus objetos e seus relacionamentos entre elementos e realidade têm a representação fiel do ambiente observado, não dependendo de limitações impostas pelas tecnologias, técnicas de implementação ou dos dispositivos físicos; e o Modelo Lógico, que é um modelo de dados que representa a estrutura de um banco de dados. Neste caso, considera os limites impostos por algum tipo de banco de dados, como os hierárquicos, em rede, relacional e orientado a objetos.

2.3.13 Diagrama de Entidade e Relacionamento

O Modelo Entidade-Relacionamento (MER) é usado na primeira etapa do desenvolvimento de um projeto de banco de dados. O MER é um diagrama baseado em figuras geométricas, as quais, a depender da necessidade, possuem rótulos, e todas devem fazer sentido, representando a estrutura conceitual do banco de dados e a associação de seus componentes. Essa representação do MER pelas figuras é chamada de Diagrama de Entidade e Relacionamento (DER). E, por intermédio dele, é possível chegar ao modelo relacional, que é formado por representações dos elementos gráficos em tabelas interligadas, dadas algumas regras de conversão ou de mapeamento, fase a qual também é conhecida como projeto lógico (PEQUENO et al., 2020).

O trabalho desenvolvido por Heuser (2009) apresenta os principais conceitos da abordagem Entidade-Relacionamento:

Entidade - representação de um fato no mundo real, sendo que se deseja salvar seus dados em um banco de dados. No diagrama ER, essa é simbolizada por um retângulo, em que cada um dos retângulos representam um conjunto de objetos que se tem o interesse de armazenar as informações;

Relacionamento - um conjunto de associações entre as entidades. O relacionamento é representado por um losango conectado por linhas às entidades que pertencem ao relacionamento. Quando se cria um relacionamento, é importante definir o número de ocorrências de instâncias das entidades que participam dele, através das cardinalidades mínima e máxima;

Atributo - são características que descrevem uma entidade ou um relacionamento. Os atributos podem ser de diferentes tipos, obrigatório/opcional, monovalorados/multivalorados, simples/composto;

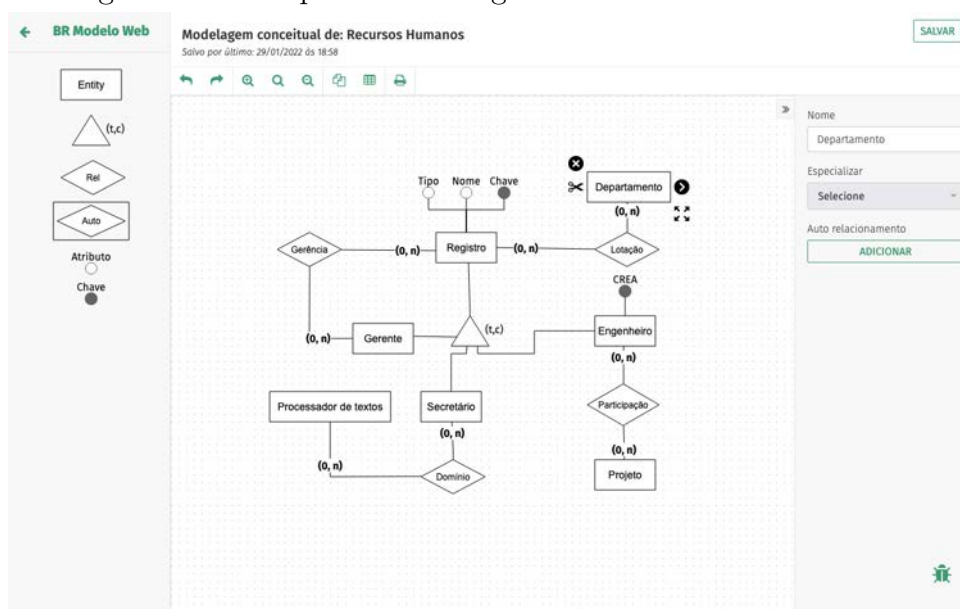
Generalização / especialização - técnica de herança em que as entidades mais especializadas herdam as características ou os relacionamentos de uma ou mais entidades genéricas. Isso ocorre para evitar a repetição das características ou dos relacionamentos na representação de diversas entidades. Sua simbologia de representação é

um triângulo isósceles;

Entidade associativa - faz a abstração de uma associação de relacionamento com outro relacionamento, o qual não é permitido no modelo ER. Para tal, o relacionamento é promovido, passando a se comportar como uma entidade associativa e resolvendo o problema. A representação se dá por intermédio de um retângulo circundando o losango do relacionamento e as entidades conectadas a ele;

O software "brModeloWeb" é uma aplicação web voltada para a modelagem conceitual e lógica dos bancos de dados relacionais. Com ele, pode-se criar uma modelagem de Entidade-Relacionamento e transformá-la em uma modelagem lógica de BD relacional. Ainda é possível converter a modelagem lógica em física por meio do SQL, com o esquema do BD projetado, e entrar com a modelagem lógica, gerando a física (NETO, 2016).

Figura 1 – Exemplo de modelagem usando o BR Modelo Web.



Fonte: WEB (2022)

2.4 Gestão de Projetos de Software

Nesta seção iremos entender melhor o conceito de gestão de projetos, especialmente na área de desenvolvimento de software.

2.4.1 Gestão de Projeto

Projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. A natureza temporária dos projetos indica que eles têm início e fim definidos. Identifica-se a chegada ao fim quando os objetivos do projeto são alcançados, ou quando este é encerrado porque seus objetivos não serão ou não podem ser alcançados,

ou até quando a necessidade do projeto deixa de existir. O projeto também poderá ser encerrado se o cliente (cliente, patrocinador ou financiador) desejar encerrá-lo (GUIDE (2017)).

Ahmed (2012) explica que o gerenciamento de projeto pode ser amplamente definido como o início de uma atividade para atingir alguns objetivos declarados, usando recursos, orçamento e tempo limitados. Durante o projeto tais recursos são consumidos em um período limitado de tempo. Após a conclusão do projeto, os recursos não consumidos devem ser liberados. Como cada projeto é iniciado para um cliente, tem-se a necessidade de, no final, avaliar a satisfação dele em relação às entregas.

2.4.2 Projeto de Software

Um projeto de software é o procedimento completo de desenvolvimento de software, desde o levantamento de requisitos até o teste e manutenção, sendo realizado de acordo com as metodologias de execução em um período de tempo especificado para atingir o produto de software almejado (AHMED, 2012). Villafiorita (2014) discorre que quando se pensa em projetos de software, o primeiro pensamento é o desenvolvimento de aplicativos, mas embora em muitos casos isso seja verdade, projetos relacionados a software podem tomar diferentes formas. Mesmo quando o objetivo principal é desenvolver um sistema, a codificação é apenas uma das atividades requeridas. O gerenciamento de projetos de software é a integração de técnicas de gerenciamento em desenvolvimento de software, sendo tal necessidade advinda da crescente complexidade nos desenvolvimentos dos softwares.

2.4.3 Gerente de Projetos de Software

Um gerente de projeto de software é uma pessoa que assume a responsabilidade de executar o projeto de software. O mesmo conhece as etapas do estado de entrega que o software deve passar. O gerente pode nunca se envolver diretamente na produção do produto final, mas ele controla e gerencia as atividades envolvidas na produção. Um gerente de projeto monitora de perto o processo de desenvolvimento, prepara e executa vários planos, organiza os recursos necessários e adequados e mantém a comunicação entre todos os membros da equipe para abordar questões de custo, orçamento, recursos, tempo, qualidade e satisfação do cliente (VILLAFIORITA, 2014; AHMED, 2012; ARTERO, 2016).

O gerente de projetos é a pessoa escolhida pela organização que desenvolve o projeto para comandar a equipe responsável por alcançar os objetivos do projeto. O papel desse líder é diferente de um gerente funcional ou gerente de operações. Usualmente, o líder funcional se concentra em proporcionar a supervisão de gerenciamento de uma unidade funcional ou de negócios, e os gerentes de operações são responsáveis pela eficiência das operações de negócios (GUIDE, 2017).

2.4.4 Ciclo de Vida de Projetos de Software

Segundo GUIDE (2017), ciclo de vida do projeto é a série pela qual um projeto passa do início ao fim. Sendo, geralmente, as fases sequenciais e os seus nomes e números determinados pelas necessidades de gerenciamento e controle das organizações envolvidas no projeto, a natureza do projeto em si e a sua área de aplicação. São evidenciadas algumas fases do ciclo de vida que podem orientar na elaboração do projeto, sendo elas: início do projeto; organização e preparação; execução do trabalho do projeto; e encerramento do projeto.

Quando se trata da realização do projeto de software, nada impede que o gerente do projeto desenvolva com mais eficiência suas fases, de modo a ter um controle diferenciado para cada uma. Cada projeto pode apresentar mais de uma fase, sendo elas caracterizadas e desenvolvidas conforme a sua necessidade, e não sendo exclusivamente necessário terminar uma fase para dar início à outra, com a possibilidade de subdividi-las (ARTERO, 2016).

2.5 Estado da Arte

Para um melhor entendimento de projetos semelhantes já trabalhados, foram feitas pesquisas em base de dados, tais como: Google Acadêmico, Google Scholar, IEE, Scielo, Roca e CAPES. A busca foi ampla, em relação à nacionalidade, não limitando-se à brasileira. Os termos utilizados para a pesquisa estão apresentados na Figura 2.

Figura 2 – Termos utilizados na pesquisa do Estado da Arte.

Gestão de Projeto de Software, Estações Solarimétricas, Estações Fotovoltaicas, Supervisão de Usinas Solares Fotovoltaicas, Software Project Management, Software Project Management and Photovoltaic, Solarimetric Station, Photovoltaic Station, Photovoltaic Station
AND
(Software OR Development OR Project), Photovoltaic Station
AND
Software
AND
(Development OR Project), Software Development Project
AND
Photovoltaic, Design of Management Systems
AND
Solar, Design of Management Systems AND Photovoltaic

Fonte: Autoria própria

Uma pré-triagem, lendo o título e a descrição das produções científicas, foi feita, conseguindo localizar cerca de 51 que tinham alguma relação com este trabalho. Após uma leitura mais aprofundada em cada uma delas, elencou-se três trabalhos mais próximos a este, a serem descritos.

Ruschel (2015), com sua dissertação, desenvolve um software que, a partir de modelos matemáticos e sabendo as condições ambientais, estima a potência instantânea esperada do sistema fotovoltaico ao qual está supervisionando no projeto "Inserção da Geração Solar Fotovoltaica Urbana Conectada à Rede em Porto Alegre". Para o desenvolvimento desse sistema, utilizou-se outro já existente como base, mudando as entradas e saídas para que se possa cumprir com o objetivo do trabalho. Após o desenvolvimento, o sistema passou por validação, a partir da comparação entre dados experimentais medidos em um sistema fotovoltaico conectado à rede instalado no laboratório e uma simulação no programa desenvolvido, replicando as mesmas condições observadas no sistema real. De forma geral, os resultados apresentaram concordância.

Em (MUNOZ, 2017), o objetivo foi o desenvolvimento de uma interface gráfica para a análise de funcionamento de painéis solares fotovoltaicos, como sugere em seu título "Interfaz grafica para el analisis del funcionamiento de paneles solares fotovoltaicos de concentración". O autor tanto projetou quanto desenvolveu a interface. Para o desenvolvimento da aplicação, separou-se em três etapas: coleta de dados das estações fotovoltaicas que estão localizados em pastas, a manipulação dos dados obtidos e a representação gráfica. Todas essas fases são desenvolvidas usando a ferramenta MATLAB. O projeto conclui com seus objetivos, trazendo uma interface de fácil utilização para o acompanhamento dos dados gerados pelas estações.

Pontes (2021), em sua dissertação de mestrado intitulada "Implantação de estação solarimétrica e de sistema supervisorio com ScadaBR e plataforma IoT em usina fotovoltaica na UNILAB-CE", objetiva apresentar a implantação de uma estação solarimétrica comercial e de sistemas supervisorios baseados em ScadaBR e plataforma de monitoramento IoT da estação solarimétrica e da usina de minigeração solar fotovoltaica, instalada no campus das Auroras da Universidade da Integração Internacional da Lusofonia Afro-Brasileira – UNILAB, no Ceará. O sistema supervisorio ScadaBR possibilita monitorar remotamente as estações, e, para acesso remoto por dispositivo móvel, foi desenvolvido um protótipo de sistema de aquisição de dados e de comunicação baseado em plataforma eletrônica IoT. Também foi desenvolvido um sistema de monitoramento via website, usando aplicativo para dispositivos móveis e através de SCADA. O sistema desenvolvido cumpriu com seus objetivos, permitindo o monitoramento tanto de uma estação solarimétrica quanto de uma usina solar fotovoltaica, a nível de hardware e de software.

Para um melhor entendimento em relação a como esses projetos se relacionam entre si e com este trabalho de conclusão de curso, algumas características foram levantadas, tais como: 1) Gestão de projeto: o quão voltado à gerência do projeto é, ou seja, se para além de projetar e desenvolver, importou-se em compartilhar como foi o processo de gerenciar o projeto; 2) Projeto de software: se existe a preocupação em projetar, definir escopo e requisitos e evidenciar o que foi e como foi projetado, antes do desenvolvimento; 3) Desenvolvimento de software: o quão voltado para o desenvolvimento do software, pensando

em linhas de código; 4) Desenvolvimento de hardware: o quão focado em desenvolver sistemas físicos para o projeto; 5) Sistema solarimétrico: entendimento se no escopo dos projetos trabalham com tecnologias de medição do clima; 6) Sistema fotovoltaico: identificar a relação da pesquisa com a geração de energia solar através desses sistemas.

Ruschel (2015) realiza o desenvolvimento de um software que estima a potência instantânea esperada do sistema fotovoltaico ao qual está supervisionando, mas com uma grande limitação em relação ao projeto de software para esse desenvolvimento, sendo sua preocupação maior codificar o sistema para que fosse funcional e cumprisse com seus objetivos, e não em descrever o projeto realizado e também não especificando a gestão do projeto. Também não teve intervenções em sistemas solarimétricos, somente em sistemas fotovoltaicos.

Munoz (2017) tem uma abordagem diferente da de Ruschel (2015), uma vez que seu projeto é voltado para o desenvolvimento de interfaces gráficas para a análise de funcionamento de painéis solares fotovoltaicos, e, com isso, houve uma preocupação um pouco maior em especificar o projeto de software e o desenvolvimento dele. Mas também não evidenciou como foi a gestão do projeto de software como um todo, somente se limitando ao projeto das telas e ao desenvolvimento delas. Outro ponto é que a gestão também se limitou aos sistemas fotovoltaicos, não se vinculando a sistemas solarimétricos.

Pontes (2021) apresenta o projeto mais completo, considerando os requisitos anteriormente estabelecidos para este trabalho, tendo foco em descrever como foi o projeto de software, assim como o desenvolvimento do projeto proposto, o mesmo sendo válido para o projeto e desenvolvimento de hardware. O projeto também voltou-se tanto a estações solarimétricas, quanto a estações fotovoltaicas. Contudo, ainda assim não teve preocupação em descrever como foi a gestão de projeto de software para essas estações.

Os projetos cumprem com os seus objetivos, mas são falhos em questões de gestão de projeto de software, em especial as estações solarimétricas e fotovoltaicas. Percebe-se que os autores dos trabalhos realizaram todas as etapas de forma independente, diminuindo a complexidade do compartilhamento de conhecimento, uma vez que desenvolviam todas as etapas do projeto. Nesse contexto, conforme aponta tal levantamento bibliográfico, em relação à ausência de textos científicos que abordassem sobre a gestão como foco central de discussão, o presente trabalho visa colaborar com o campo de estudo ao trazer tal perspectiva. Uma vez que, ao contrário dos projetos encontrados, este irá trabalhar as questões de desenvolvimento de software para sistemas solarimétricos e fotovoltaicos, mas com a perspectiva de gestão de projeto de software, não somente projeto e desenvolvimento.

3 PROJETO LABENS

Para alcançar os objetivos específicos do trabalho, foi necessário entender o funcionamento do LABENS. A infraestrutura que engloba as estações solarimétricas e fotovoltaicas. A relação lógica, do fluxo de dados gerados, enviados e armazenados. O uso atual que era feito em relação aos dados disponibilizados, pelos bolsistas e professores. O uso que poderiam fazer, de forma mais simplificada e automatizada. Como poderia ser feito para simplificar a estrutura, dado que o futuro seria incerto em relação a bolsistas para fazer a manutenção, e quanto mais simplificada e documentada, melhor.

3.1 Histórico

3.1.1 Projeto UTFPR

Em 2017, a UTFPR assinou um protocolo de intenções objetivando a cooperação técnica e científica entre a Universidade e a Copel Distribuição, dois contratos de projetos, sendo um voltado à eficiência energética e outro orientado à P&D, e dois termos de cooperação técnica (ligados à contratação dos projetos de pesquisa) com a Copel Distribuição. Sendo, os projetos contratados, dos campi Curitiba e Pato Branco, aprovados na Chamada Pública da Copel DIS.

No escopo do projeto aprovado no campus de Curitiba para cumprir as metas da linha de pesquisa fotovoltaica, a Universidade se propôs a implantar três estações solarimétricas completas que farão parte da rede SONDA (Sistema de Organização Nacional de Dados Ambientais), outras quatro estações do padrão EPE (Empresa de Pesquisa Energética) e seis Módulos de Avaliação de SFVCR (sistema fotovoltaico conectado à rede) a serem instalados entre vários campi (Curitiba, Ponta Grossa, Cornélio Procópio, Pato Branco, Campo Mourão e Medianeira). Devido à diversidade de sensores e às localidades, as equipes que trabalham com os dados gerados precisam tê-los centralizados em um servidor local, com sistema de gerenciamento desses equipamentos. Entretanto, por estar centralizado, os pesquisadores e técnicos necessitam realizar a coleta de cada Estação individualmente, ou até mesmo ir ao local onde os sensores se localizam para coleta manual.

Considerando que o projeto precisa ter um sistema que seja compatível com suas necessidades, necessita-se elaborar um modelo computacional que possibilite o desenvolvimento de um sistema baseado em computador, o qual permita o gerenciamento remoto do projeto. Esse modelo terá como objetivo a estruturação de todo o projeto já existente, possuindo como componentes a serem desenvolvidos: uma API (Interface de aplicação entre programas), a qual consiste em um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software, um banco de dados e sistemas para utilização de forma local e remota, os quais serão responsáveis por entregar dados e informações

relacionados aos processos de geração de energia solar, sendo os mesmos gerados com o auxílio dos registros documentados. Ambler (2004) "a documentação nos auxilia na comunicação durante o projeto e no entendimento nas atividades de manutenção, sendo um fator necessário quando é preciso estabelecer uma comunicação com uma equipe externa de trabalho, sendo um mecanismo essencial no suporte à comunicação, quando esta é combinada com reuniões, teleconferências, correio eletrônico e ferramentas colaborativas".

3.1.2 ANEEL - Chamada N^o.001/2016

Pensando na eficácia dos métodos alternativos para concepção de energia, a Agência Nacional de Energia Elétrica – ANEEL, através da Superintendência de Pesquisa e Desenvolvimento e Eficiência Energética – SPE, publicou, em outubro de 2016, a Chamada N^o. 001/2006 Projeto Prioritário de Eficiência Energética e Estratégico de P&D: "Eficiência Energética e Mini-geração em Instituições Federais de Ensino Superior"(ANEEL, 2016).

Conforme consta no Módulo 5, Seção 5.1 do PROPEE (Procedimentos do Programa de Eficiência Energética, aprovado pela Resolução Normativa n^o. 556/2013), Projeto Prioritário é aquele considerado "[...] de grande relevância e/ou abrangência, cuja finalidade é testar, incentivar ou definir ações de destaque como política pública para incrementar a Eficiência Energética no país"(ANEEL, 2013). E, em relação aos Projetos Estratégicos de P&D, "...são aqueles cujos temas são considerados de grande relevância para o setor elétrico brasileiro, compreendendo estudos e desenvolvimentos que integram a geração de novo conhecimento tecnológico e exija um esforço conjunto e coordenado de várias empresas e entidades executoras"(ANEEL, 2016).

O objetivo principal desse programa de Eficiência Energética é promover o uso eficiente e racional de energia elétrica, por meio de projetos que demonstrem a importância e a viabilidade econômica de ações que combatam o desperdício e que promovam o aumento da eficiência energética, tanto de equipamentos como de processos (ANEEL, 2016).

Já o objetivo do Programa de Pesquisa e Desenvolvimento é alocar, da forma mais apropriada, recursos humanos e financeiros nos projetos que demonstrem a originalidade, a aplicabilidade, a relevância e a viabilidade econômica de produtos e serviços, nos processos finais de uso de energia (ANEEL, 2016). Buscando-se, através dessa alocação de recursos, incentivar a cultura da inovação, com estímulo à pesquisa e ao desenvolvimento no Setor Elétrico brasileiro, com a criação de equipamentos, aprimoramento das prestações de serviço que contribuam para a segurança do fornecimento de energia elétrica, a modicidade tarifária (direito subjetivo do usuário de ter assegurado o seu acesso ao serviço público, seja ele prestado direta ou indiretamente pelo Estado), diminuição do impacto ambiental do setor e da dependência tecnológica do país (ANEEL, 2016).

Os programas também foram direcionados às 63 (sessenta e três) Universidades Federais. Instituições Públicas de Ensino Superior, que desempenham um importante papel no desenvolvimento científico e tecnológico do país. Instituições que possuem como terceira

maior despesa contas com a energia elétrica ANEEL (2016). De acordo com a ANEEL (2016), grande parte dos gastos com energia elétrica se deve ao uso de equipamentos ineficientes e práticas inadequadas de instalação, uso e manutenção dos aparelhos, entre outros motivos.

Essas características das Universidades foram muito importantes para a realização dessa chamada, que beneficiam à sociedade brasileira como um todo e, em especial, às instituições contempladas.

3.1.3 COPEL - Chamada Pública COPEL 001/2017

De modo que a Copel Distribuição S.A., concessionária do serviço público de distribuição de energia elétrica, subsidiária integral da Companhia Paranaense de Energia – COPEL, com finalidade de selecionar propostas de projetos plenamente aderentes à Chamada Nº.001/2016, realiza a “CHAMADA PÚBLICA COPEL 001/2017 PROJETO PRIORITÁRIO DE EFICIÊNCIA ENERGÉTICA E ESTRATÉGICO DE P&D” (COPEL, 2017a). Essa chamada objetiva tanto a concretização de projetos de eficiência energética, quanto a implantação de sistemas de geração própria de energia (na escala da minigeração), para que sirvam de referência e facilitem a implantação de projetos e ações de mesma natureza no setor público.

Nesse contexto, a concessionária publicou um edital de chamada pública, para o recebimento de propostas de projetos por Instituições Públicas de Educação Superior, que possibilitem o estabelecimento de centro de referência, por intermédio de pilotos e demonstrativos. Esperou-se também que a execução desses projetos servissem de subsídios para a formulação de políticas públicas de combate ao desperdício de energia elétrica (nas unidades consumidoras da administração pública) (COPEL, 2017a).

3.2 Laboratório de Energia Solar LABENS

O Laboratório de Energia Solar – LABENS¹, da UTFPR – Campus Curitiba desenvolve pesquisas relacionadas à energia solar, para determinação da distribuição da radiação solar e do potencial fotovoltaico no estado do Paraná, e, também, na análise dos índices de mérito de sistemas fotovoltaicos conectados à rede elétrica de energia instalados na própria Universidade. Esse esforço empregado pelos professores da Universidade, que administram o Laboratório, objetiva resultados que possam auxiliar na disseminação dessa importante fonte de energia no Estado e de subsidiar cursos para formação de profissionais habilitados para atuar nesta área (LABENS, 2019).

Dentre os projetos desenvolvidos no LABENS, destacam-se o Estacionamento Solar Fotovoltaico para carregamento de veículos elétricos e Conversão de um veículo Mercedes Benz Classe A190; Desenvolvimento do Atlas de Energia Solar do Estado do

¹<http://labens.ct.utfpr.edu.br/>

Paraná – 1^o Edição, em parceria com outras instituições; Contribuição com a publicação da segunda edição do Atlas Brasileiro de Energia Solar; Análise dos índices de mérito (Produtividade, Taxa de Desempenho e Fator de Capacidade); Análise da sujidade dos sistemas; Fatores que influenciam no desempenho de SFVCR (LABENS, 2019).

Além dos projetos supracitados, desenvolvem o Projeto P&D ANEEL/COPEL Distribuição – PD 2866-0464/2017 – Metodologia para Análise, Monitoramento e Gerenciamento da Geração Distribuída por Fontes Incentivadas. Projeto que será escopo para o desenvolvimento deste Trabalho de Conclusão de Curso. Uma vez que o mesmo foi divulgado na turma de TCC1 dos autores, e que um deles se inscreveu no edital para ser bolsista de pesquisa de nível auxiliar técnico no projeto, sendo deferido e passando a ter acesso às informações e aos demais envolvidos no projeto.

3.3 Aplicação

Os dados gerados pelas estações são para diversas finalidades, como para elaboração de artigos e congressos periódicos, dissertações de mestrado, monografias de especialização, trabalhos de conclusão de curso na graduação, entre outras. Alguns desses trabalhos foram acompanhados pelos autores, sendo entendidas diretamente as demandas que possuíam de dados para a geração de informações relevantes para o desenvolvimento dos trabalhos.

A Dissertação de Mestrado intitulada "Análise e Estimativa das Componentes da Radiação Solar Através de Dados Medidos na Rede EPESOL DA UTFPR no Estado do Paraná", da mestranda Muriele Bester de Souza, dentre alguns pontos abordados, trabalhou a estimativa dos valores da radiação difusa, direta normal e global no plano inclinado na latitude entre as estações SONDA e EPE implantadas em Curitiba, cujos piranômetros instalados apresentam graus de precisão distintos (SONDA: padrão secundário – CMP10; EPE: primeira classe – CMP6). As medidas coletadas, foram usadas para comparações, com ATLAS do Paraná e com medidas do Instituto Nacional de Meteorologia.

Renata Lautert Yang, bolsista de mestrado do LABENS, em sua dissertação "Estimação da Taxa de Desempenho de Sistemas Fotovoltaicos Conectados à Rede com Diferentes Tecnologias no Estado do Paraná", buscou estimar a taxa de desempenho (TD) para sistemas fotovoltaicos conectados à rede elétrica (SFVCR) no Paraná para quatro tecnologias de módulos fotovoltaicos disponíveis no projeto: silício monocristalino (m-Si), silício policristalino (p-Si), disseleneto de cobre-índio-gálio (CIGS) e telureto de cádmio (CdTe), instalados nos seis campi. O conhecimento da TD é importante para determinar o desempenho dos sistemas, descontando as diversas formas de perdas existentes, em especial as da própria célula fotovoltaica.

A mestranda Camila de Oliveira Silveira, em sua dissertação "Análise do Desempenho de Sistemas Fotovoltaicos Conectados à Rede Elétrica com Diferentes Tecnologias no Estado do Paraná", acompanhou e comparou, por meio de índices de mérito calculados com medições reais de geração de energia e radiação, o desempenho entre os sistemas

implantados em cada um dos campi (sistema de m-Si, CIGS, silício cristalino, p-Si).

Através de sua dissertação, Diego Plazza Hilgert trouxe o "Projeto e Implantação de SFVCR com Diferentes Tecnologias no Estado do Paraná", que, ao contrário dos projetos anteriores, focou mais no antes da geração, detalhando os processos do projeto de instalação de quatro tecnologias distintas de células fotovoltaicas nos seis campi. Por isso, o desenvolvimento de sua dissertação pode contribuir com as documentações do projeto, para um maior entendimento da estrutura física.

4 METODOLOGIA

Neste capítulo é explicado o desenvolvimento do projeto, entendimento da organização pessoal, física e lógica dos requisitos, da organização dos requisitos, alinhamento de escopo, prioridade e cronogramas, busca da empresa para o desenvolvimento dos requisitos e o acompanhamento com ela, bem como as homologações das entregas. É realizada a abordagem de como foi realizada a gestão de projeto de software para as estações solarimétricas e fotovoltaicas implantadas nos campi da UTFPR.

O autor do presente trabalho integrava o grupo de bolsistas de graduação, sendo um dos cinco que compunham a equipe de TI do LABENs. Sendo de sua responsabilidade o entendimento e levantamento dos requisitos necessários para o sistema a ser desenvolvido, busca pela empresa a ser contratada para o desenvolvimento, auxílio nas especificações dos requisitos das empresas, interface e gerência do projeto com a empresa contratada, documentação das reuniões e do sistema, definição das permissões do sistema, replicação do conhecimento adquirido entre os demais membros do LABENs, entendimento e apresentação de alternativas para o uso da API de consulta que foi desenvolvida.

Em todas essas frentes, algumas disciplinas curriculares do curso tiveram grande importância para um melhor desenvolvimento das atividades. Tendo como principal destaque

- Técnicas De Programação - permitiu um maior entendimento dos conceitos de modelagem orientada a objeto;
- Introdução A Banco De Dados - tendo em vista que grande parte do trabalho foi modelar um DER, pensando juntamente quais seriam as melhores maneiras de se armazenar e recuperar os dados;
- Análise e Projeto de Sistemas e Engenharia De Software I e II - foram de grande importância para gerenciar o projeto, incluindo toda a documentação desenvolvida;
- Introdução À Interação Humano-Computador - permitiu conduzir o desenvolvimento da interface do sistema de monitoramento de melhor maneira;
- Desenvolvimento De Aplicações Web - foi fundamental para um melhor desempenho, pois inúmeros conceitos aprendidos nessa matéria auxiliaram no desenvolvimento do projeto, em especial porque o sistema de gerenciamento e monitoramento desenvolvido é uma aplicação web.

Dentre as responsabilidades, a mais complexa foi a gerência do projeto, a qual foi realizada durante a pandemia, em meio a inúmeras incertezas, inclusive de se o projeto seria viável. Após algumas análises e discussões, foi optado por dar continuidade ao projeto, e o que era para ser realizado presencialmente, com o autor presente na empresa acompanhando os desenvolvimentos, passou a ser online, através de ferramentas de videoconferência elencada para tal. O maior desafio foi em relação à saúde mental, pois os trabalhos remotos forçaram

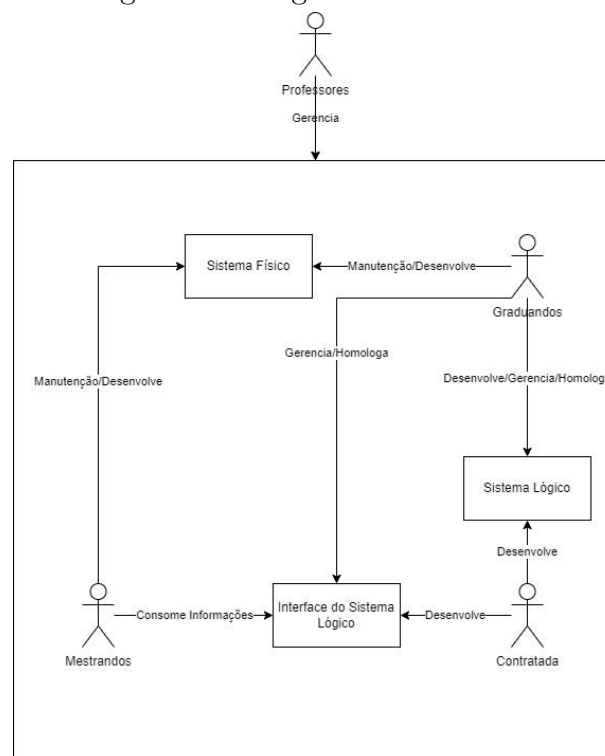
mudanças em como os trabalhos seriam desenvolvidos, mas a tecnologia viabilizou o projeto com uma curva de adaptação alta. Apesar de ser difícil fazer um comparativo de como seria de maneira presencial, dado que não se teve tempo de acontecer nenhuma reunião presencial com a empresa, o projeto conseguiu cumprir o prazo final dado pelo financiador.

4.1 Partes envolvidas

A primeira etapa do projeto foi ter um melhor entendimento em relação às partes envolvidas no projeto Solar do LABENS. Por isso, este capítulo traz, de forma resumida, os envolvidos nas pesquisas dentro do laboratório e as suas respectivas responsabilidades. Sendo o autor um dos pesquisadores do laboratório, como bolsista de Iniciação Científica Técnica.

Para o desenvolvimento bem-sucedido do trabalho, o projeto teve como autores 5 professores, 8 mestrandos, 10 graduandos e uma empresa de desenvolvimento de software contratada. Na Figura 3 está apresentada a estrutura obtida.

Figura 3 – Diagrama de trabalho.



Fonte: Autoria própria

Professores responsáveis gerais pelo projeto, gerenciando ele como um todo, auxiliando os bolsistas em suas frentes de trabalho e desenvolvendo pesquisas sobre a temática; **Bolsistas de Mestrado** contribuíram para o desenvolvimento dos sistemas solares e fotovoltaicas, incluindo documentação e manutenção. Também trabalhando com os dados coletados para o desenvolvimento de pesquisas científicas;

Graduandos podem ser divididos em dois grupos, um que fez um trabalho similar aos mestrados, mais direcionado ao cuidado das estações, e outro voltado para as questões de tecnologia da informação. Esse segundo grupo é responsável por gerência, homologação, desenvolvimento e manutenção da parte lógica e parte da física (cuidando da comunicação com os ST-Ones, Dataloggers e Raspberry pi 3, que estão conectados às estações). Esses bolsistas também cuidam do servidor (tanto lógico quanto físico) que hospedam o site, o banco de dados, o FTP e os sistemas de gerenciamento e monitoramento das estações (ou seja, toda a infraestrutura virtual); **Empresa Contratada** passa a ser uma das partes envolvidas mais à frente no projeto, mas é a que desenvolve o sistema de monitoramento das estações, trabalhando diretamente com o grupo de bolsistas de TI, desenvolvendo tanto a parte lógica quanto a interface do sistema.

4.2 Entendimento lógico e físico do projeto

Após o entendimento das partes envolvidas, as quais auxiliaram em dúvidas pontuais não elucidadas por documentação, passou-se para a segunda etapa, a qual foi entender a infraestrutura em que seria trabalhada, tanto lógica quanto física do projeto. As estações por serem divididas em 6 campi, sendo que dificultaria a locomoção para conhecer o funcionamento de todas elas presencialmente. Portanto, somente um deles teve uma visita presencial, para maior entendimento, o campus de Curitiba. Mas, tanto nesse como nos demais, foram analisados através de seus "As Builts" ou "como construído", ou seja, projetos com representações técnicas da construção (plantas, cortes, fachadas etc). Tal termo é tratado em norma técnica da ABNT (Associação Brasileira de Normas Técnicas), no caso, a NBR 14.645:2001 – Elaboração do “como construído” (As Built) para edificações.

Em tal projeto, foi possível consultar informações, como: normas, relação e composição do projeto básico, informações gerais sobre a obra, superfície de instalação do sistema, proteção física, os sistemas fotovoltaicos conectados à rede, englobando os módulos fotovoltaicos, os inversores, as conexões elétricas, as proteções, o aterramento, os conectores, as conexões lógicas, as conexões de automação e informações sobre a estação EPE, assim como uma lista básica dos materiais utilizados.

Juntamente com esses projetos e o acompanhamento presencial no campus Centro-Curitiba, foi possível levantar a seguinte estrutura:

Computador - um computador servidor instalado no campus Centro-Curitiba da UTFPR para hospedagem das aplicações, onde a infraestrutura de TI está virtualizada. Servidor que possui um processador Intel Xeon E-2124G, 32GB de memória RAM e dois HDs de 2TB cada, instalados em RAID 1 via software. Isso significa que os dois

discos são cópias um do outro, o que garante segurança em caso de falha de um dos discos, resultando em uma capacidade útil de 2TB. Dentro desse servidor, foram criadas algumas máquinas virtuais, tais como: FTP Server, que recebe os dados das EPESOLs via FTP, guarda os dados brutos e envia os dados das estações SONDA para o INPE. Possui 2 núcleos virtuais de processamento, 6GB de RAM e 330GB de armazenamento, divididos em 2 partições: uma de 30GB, em que está o sistema, e uma de 300GB, onde são salvos os dados brutos, sendo que o sistema operacional é o Debian 10. Uma máquina para o banco de dados da aplicação, possuindo 2 núcleos virtuais de processamento, 2GB de RAM e 700GB de armazenamento, sendo que o sistema operacional é o Debian 10. E uma para a Aplicação, a qual possui 2 núcleos virtuais de processamento, 4GB de RAM e 70GB de armazenamento, sendo que o sistema operacional é o Debian 10.

Plantas - seis plantas micro-geradoras de energia elétrica fotovoltaica, nas cidades de Curitiba, Ponta Grossa, Pato Branco, Medianeira, Campo Mourão e Cornélio Procopio; cada planta contendo dois tipos de CPU remota: i) computador de automação (SCADA) programado em linguagem IoT Node-Red e ii) Datalogger (CR1000x) programado em linguagem proprietária Campbell Scientific.

Raspberry Pi - um Raspberry Pi 3, programado em python, instalado no campus Neoville-Curitiba da UTFPR (utilizado para os inversores bidirecionais).

Todas as CPUs estão conectadas à internet e enviam dados para um servidor FTP. Sendo o fluxo de dados iniciando pelas estações, com a seguinte lógica: sensores das estações fotovoltaicas, pensando nos inversores, direcionavam o fluxo de dados para computadores de automação (SCADA). Somente no caso da estação de Curitiba, que possuía um sistema de inversores bidirecionais, que direcionavam as informações para um Raspberry Pi 3, enquanto que as estações solarimétricas direcionavam os dados gerados para um Datalogger (CR1000x).

Todos equipamentos devidamente conectados à internet, para envio dos dados coletadas. Em caso de perda de conexão, os dados são armazenados no próprio equipamento e enviados assim que a conectividade for retomada. O Datalogger trabalha com coletas de 10 em 10 minutos, enquanto que os inversores trabalham com uma frequência de coleta de dados de minuto em minuto. O envio era diário referente a toda coleta.

De início, parte dos dados eram enviados a um servidor de FTP privado, outra parte a uma conta do Dropbox, e o inversor bidirecional ainda não possuía rotina de envio configurada. E quem tivesse a necessidade de trabalhar com as informações precisava utilizar os dados trabalhando com planilhas de dados, sendo que as informações eram enviadas em formato ".csv". Caso precisasse de mais de um dia de dados, seria necessário requisitar a um bolsista de TI que fizesse um compilado de todo o período desejado. E, mesmo assim, os dados de cada dispositivo estariam em planilhas diferentes, o que dificultava a correlação, quando necessário.

Após algumas análises em conjunto com a equipe de TI, decidiu-se centralizar as coletas em um servidor FTP próprio do LABENS. Criou-se uma aplicação para que os dados armazenados fossem buscados e importados para dentro do banco de dados da aplicação diariamente, para, então, serem trabalhados a partir do próprio banco ou das aplicações que façam uso dele. Essa rotina foi elaborada para substituir o processo anterior que era muito custoso a quem precisasse analisar as informações, dado o empenho de juntar esses dados.

Esse fluxo foi possível, dado uma estruturação das nomenclaturas realizadas previamente, para a busca e inserção dos dados no banco de dados.

4.3 Entendimento e levantamento dos requisitos de software

Após entendimento dos envolvidos no projeto, da infraestrutura física e lógica, bem como da comunicação de dados, iniciou-se a etapa de entendimento e levantamento dos requisitos. Como o principal objetivo do sistema é servir aos seus usuários, a etapa inicial foi entender pelos professores e bolsistas como realizavam o trabalho até então. Bem como com acesso aos próprios dados gerados pelas estações, para melhor entendimento dessas informações.

De forma mais prática, um acompanhamento de como os envolvidos transformavam os dados coletados em informações foi feito. Esse entendimento foi realizado pedindo para que bolsistas e professores mostrassem o trabalho que realizavam diariamente. Assim como foi pedido para descreverem os desafios, dificuldades e o que tomava mais tempo deles para o desenvolvimento dos trabalhos. Assim como o que eles viam como possibilidades de melhorias, e o que gostariam que houvesse em um sistema para auxiliar nos trabalhos, de modo a entender os Casos de Usos e Requisitos mais comuns, para, se possível, priorizar no desenvolvimento do modelo computacional para uma futura implementação. De forma geral, percebeu-se que a aquisição dos dados pelos bolsistas e professores para seu uso era empenhoso. Uma vez que precisava procurar dentro de pastas de um servidor FTP, onde estavam os .CSVs que precisavam trabalhar, e depois de identificá-los, ainda precisavam manusear todos eles, lidando ainda com os desafios de conversão, quando usado em formato “.XLSX”. Tais informações foram documentadas para posterior uso.

De forma quase paralela, foram estruturadas todas as informações coletadas pelos bolsistas e professores, montando tabelas de Casos de Uso para então gerar os seus Requisitos. Um levantamento dos equipamentos, das possíveis variáveis geradas pelos equipamentos de coleta de dados e as grandezas dessas variáveis foi feito, para, então, o desenvolvimento de um modelo relacional, com desenvolvimento de um Diagrama de Estado e Relacionamento e de um Modelo Lógico, para facilitar a visualização do fluxo desses dados, e como seriam os relacionamentos entre as tabelas, para um posterior uso delas. Também, foi desenvolvido um dicionário de dados, para definições e representações de elementos de dados.

Após pronto, algumas rodadas de validações foram feitas, tanto com os bolsistas, quanto com os professores. No percurso alguns ajustes foram solicitados, bem como alguns requisitos foram percebidos como desnecessários e outros como necessários de inclusão. Também, pode-se perceber quais eram os requisitos mais importantes, que deveriam ser realizados primeiro ou que não poderiam ser esquecidos. Os apêndices B, C e D apresentam alguns resultados obtidos no processo de análise.

4.4 Requisitos e prazos a serem repassados à empresa contratada

Como idealização inicial de cronograma e escopo, para o desenvolvimento do projeto junto com a empresa contratada, foi-se dividido entre cinco módulos, da seguinte maneira:

1. COMUNICAÇÃO – Prazo de 5 dias para desenvolvimento e 1 dia para a documentação;
2. ARMAZENAMENTO – Prazo de 7 dias para desenvolvimento e 2 dias para a documentação;
3. GESTÃO – Prazo de 15 dias para desenvolvimento, 5 dias de revisão e 5 dias para a documentação;
4. ACESSO – Prazo de 8 dias para desenvolvimento, 5 dias de revisão e 2 dias para a documentação;
5. API – Prazo de 20 dias para desenvolvimento, 10 dias de revisão e 5 dias para a documentação.

Totalizando 90 dias.

Resumidamente, os módulos foram descritos e apresentados previamente da seguinte maneira:

1. COMUNICAÇÃO:
 - Recebimento de todos os pacotes de comunicação dos dispositivos remotos, sem perda de pacotes, garantido através de protocolo específico para garantia da entrega dos pacotes;
2. ARMAZENAMENTO:
 - Separação adequada dos campos presentes em cada pacote, de um total estimado entre 100 a 200 diferentes tipos de campos, armazenando-os na tabela Leituras e respectivas tabelas de Relacionamentos para fins de categorização da informação;
3. GESTÃO:
 - Interfaceamento através de páginas web para ações de inclusão, consulta, alteração e listagem em arquivo texto, de todos os registros, presentes em todas as tabelas de Entidades do DER (exceto a tabela Leitura, cujos dados serão estáticos, com inclusão automática e ação de consulta a ser realizada através de API própria), de acordo com os critérios de liberação de acesso, conforme o Grupo a que o usuário acessante pertença;

- Execução de *Backup* diário de todo o banco de dados, numa fila de *Backup* dos últimos 30 dias;

4. ACESSO:

- Organização dos usuários em quatro Grupos distintos. A saber: Administrador, Gestor, Pesquisador e Visitante, promovendo a liberação de acesso às ações de gestão/API do sistema de acordo com as diferentes regras de permissão de cada Grupo;
- Registro de todos os *logs* realizados por usuários no sistema, em tabela própria do banco de dados, acessível apenas para o usuário Administrador;

5. API:

- Negociação automática com diferentes aplicativos computacionais, destinado à identificação, certificação e categorização do nível de acesso solicitado;
- Identificação e validação dos parâmetros de pesquisa recebidos, a fim de realizar consulta na tabela de Leituras, devolvendo o resultado, conforme selecionado pelo cliente em um dos formatos JSON ou CSV. A consulta poderá ser realizada em amplo espectro de seleção de campos assim como de regras de filtragem por campos, incluindo resultados destes através de operadores lógicos *booleanos* e/ou resultados de comparadores relacionais, em toda a abrangência de tabelas do banco de dados;
- Registro de todos os *logs* realizados por API no sistema, em tabela própria do banco de dados, acessível apenas para o usuário Administrador.

4.5 Levantamento dos requisitos da empresa para desenvolver

Alguns critérios de seleção foram levantados na busca da empresa para o desenvolvimento do sistema de monitoramento das estações fotovoltaicas e solarimétricos, em que somente poderiam pessoas jurídicas, ou seja, empresas, eliminando-se assim pessoas físicas. Deveriam estar estabelecidas com sede em Curitiba ou região metropolitana, facilitando-se assim o contato entre as partes. Devendo ter atuação na área de desenvolvimento de software há pelo menos 5 anos, necessitando que ao menos uma atividade em seu CNAE (Classificação Nacional de Atividades Econômicas) fosse relacionado ao desenvolvimento de software, podendo ser tanto CNAE primário quanto CNAE secundário, respeitando assim a atividade fim que se necessitariam os serviços da contratada. Por fim, a empresa deverá estar regularizada, em relação aos compromissos fiscais e atender aos critérios de contratação por parte de empresas públicas, dado que a contratação seria através da FUNTEFPR (Fundação de Apoio à UTFPR).

4.6 Busca da empresa para desenvolver o sistema

Para encontrar possíveis empresas para desenvolver o sistema, o primeiro passo foi realizar a busca das quais estivessem localizadas em Curitiba ou região metropolitana, sendo o critério mais fácil de se identificar, com as ferramentas disponíveis. Para encontrar essas empresas, algumas buscas no buscador do Google foram realizadas. Para todas as empresas encontradas, buscou-se seu site ou redes sociais, verificando-se a qual segmento e porte de projetos atendiam, pois caso não fizessem sentido para o projeto em questão, não necessitaria-se entrar em contato. Após as buscas e já a eliminação de empresas que visivelmente não fariam sentido, cerca de 13 empresas foram localizadas com a rede social, o site, o e-mail ou o telefone de contato. Também, uma descrição prévia do que foi identificado olhando seu site e redes sociais foi feita, a título de entendimento sobre as empresas.

4.7 Triagem das empresas e Contratação

Após encontrar essas 13 empresas, a próxima etapa foi entrar em contato com cada uma delas, explicando de forma preliminar qual seria a demanda, e se a empresa teria interesse em marcar uma reunião para melhor entendimento. Um documento no corpo do e-mail foi enviado, com a seguinte estrutura: 1) Do objeto: explicando, de forma geral, qual é a demanda em relação ao desenvolvimento do software. Apresentando um diagrama físico, com especificações macro, dos dispositivos envolvidos nas trocas, geração e armazenamento dos dados. Também, uma breve explicação de quais demandas o sistema a ser desenvolvido deveria atender; 2) Dos critérios de seleção da empresa executora: em que somente poderiam pessoas jurídicas, estabelecidas com sede em Curitiba ou região metropolitana, com atuação na área de desenvolvimento de software há pelo menos 5 anos, tendo, ao menos, uma atividade em seu CNAE (Classificação Nacional de Atividades Econômicas) relacionada ao desenvolvimento de software. Por fim, a empresa deverá estar regularizada, em relação aos compromissos fiscais e atender aos critérios de contratação por parte de empresas públicas; 3) Dos prazos de execução: após a assinatura do contrato, a empresa selecionada deveria realizar a entrega dos artefatos de software em até 90 dias; 4) Envio de proposta de preço e cronograma de execução: proposta realizada pela empresa interessada em realizar o serviço, e que fosse apta para tal; 5) Do pagamento: orientações referentes a como se daria o pagamento, explicando que o recebimento estaria vinculado às entregas e que necessitaria de um aceite por parte do LABENs para tal.

Cinco empresas retornaram com interesse em entender mais sobre o projeto, antes de realizar o envio de um orçamento. Reuniões foram marcadas e as dúvidas das empresas foram sanadas. O que foi benéfico para ambas as partes, pois, de um lado, a possível empresa a ser contratada, tinha um entendimento mais pleno do escopo do trabalho, e do outro o laboratório entendia mais sobre a empresa que iria pleitear a proposta.

Ao final, quatro empresas enviaram suas ofertas, respondendo o e-mail encaminhado, para apreciação e aceite ou não pelo LABENS. As propostas foram analisadas, e a mais aderente foi aceita, felizmente, a com menor preço também. A equipe jurídica e financeira da FUNTEFPR fez suas análises específicas, e então a empresa, estando apta, foi contratada.

4.8 Alinhamento do escopo do projeto, prioridades e cronogramas em definição com a empresa a ser contratada

Os requisitos macros, prioridades e prazos haviam sido definidos em documento inicial de contato, apresentado na seção 4.4, mas foram reforçados e melhor explicados na primeira reunião com a empresa. Nessa reunião, foram apresentadas as documentações mais relevantes para um melhor entendimento do projeto. O Apêndice D apresenta o Diagrama Entidade-Relacionamento e o Apêndice E apresenta o Diagrama Lógico existente do banco de dados. Alguns questionamentos foram feitos, e alguns possíveis ajustes foram comentados. Sendo acordado o envio de outras informações, como os arquivos de saída dos dispositivos que monitoram as estações para melhor entendimento, bem como as suas respectivas documentações.

4.9 Interface entre LABENS e Empresa contratada

Após definida a empresa para realizar o desenvolvimento do sistema de monitoramento e gerenciamento das estações, foi definido em conjunto que semanalmente haveria ao menos uma reunião com eles, para acompanhamento do desenvolvimento do projeto, validando se os requisitos estavam sendo cumpridos. A ideia inicial seria ter um acompanhamento presencial, em que um integrante da equipe de bolsistas iria acompanhar o projeto na empresa que desenvolve, mas, dado o contexto da pandemia, todas as reuniões foram realizadas via a plataforma Google Meet. Um resumo das interações está apresentado no Apêndice A.

Após o desenvolvimento do sistema de monitoramento e gerenciamento, foi elaborado um repasse técnico para complementar toda a documentação juntamente com o código fonte. A ideia foi dividir o repasse em sete encontros de até duas horas cada, conforme apresentado na Tabela 1, explicando de forma ampla cada uma das principais estruturas utilizadas para o desenvolvimento do sistema. A ideia foi gravar as explicações, para que, futuramente, caso seja necessário, tenha esse conteúdo complementar, de modo a acelerar o entendimento do que foi realizado. Desses repasses técnicos, alguns bolsistas e professores acompanharam, tirando dúvidas.

Tabela 1 – Aulas para documentação do Sistema WEB - GEBIT e LABENS

Data	Resumo
24/09/2020	Overview Laravel - Estrutura, ORM, Conexão Banco e Autenticação
28/09/2020	Eloquent, ORM, Rotina de Importação e Scheduler
02/10/2020	Views (Blade), Controller (Telas)
06/10/2020	Vue.js, API Controller
08/10/2020	API de consulta, comandos
14/10/2020	Servidor (Nginx + Banco) e Publicação
16/10/2020	Dúvidas e finalização

4.10 Materiais e definições de apoio

Para um trabalho efetivo da empresa contratada, algumas entregas de documentação e definições foram feitas. Por exemplo, especificações lógicas do servidor, configurações nos dispositivos, validações de casos de usos e requisitos, entre outras demandas. Para atender essas demandas, pessoas específicas no LABENS eram envolvidas, para auxílio e realização dos trabalhos específicos que auxiliassem na viabilidade do desenvolvimento do projeto final. Por exemplo, dada a complexidade do sistema físico em número de dispositivos, foi necessário buscar algum padrão de nomenclatura, em especial para as variáveis, dado que elas passavam de 600. Para tal, foi feito um levantamento de inventário, de dispositivos que compunham esse inventário, das grandezas que cada um desses dispositivos mediam, e, por fim, das variáveis que cada um possuía. Para auxiliar, foram consultadas as documentações desses dispositivos, bem como analisado o arquivo de saída de cada um deles, para, então, elaborar uma relação com todas essas variáveis e colocar um padrão de nomes. Durante o levantamento, aproveitou-se para definir outras colunas de informações relevantes, além de suas nomenclaturas.

O nome de inventário dos dispositivos ficou como um nome mais genérico, para que fosse mais detalhado quando fosse um dispositivo mais específico de um dado campus mais específico. Exemplos: "datalogger", "inversor_1K5", "inversor_3K". Para os dispositivos, houve a abreviação do nome de inventário, separado por "-", juntamente com o número de série (em alguns casos o número de série leva a abreviação do campus em que está instalado, para diferenciar dentre os iguais). Por exemplo: "datalogger" foi abreviado para "datlog" e somado com o seu número de série "9402", ficando "datlog-9402". Outro exemplo seria o "abrasp", que é abreviação para "abrigo_aspirado_termohigrometro", que juntou com o número de série "cm000", ficando "abrasp-cm000". Para grandezas, a nomenclatura foi um pouco distinta, buscando trazer na primeira metade do nome menção ao que a grandeza estava medindo, e na segunda metade, separado por "-", o dispositivo. Por exemplo: "car-bid1" que mede a carga percentual estimada do banco de baterias do inversor 1. Outro exemplo seria o "cdte-temp", que mede a temperatura média dos módulos de avaliação

CdTe. Após a definição das grandezas, foram geradas as variáveis, sendo nomeadas pelo nome da grandeza juntamente com o campus de onde originava a medição. No caso da grandeza apresentada como exemplo "car-bid1", que possui o dispositivo gerador dessa grandeza no campus Curitiba, nomeou-se a variável como "car-bidi1CT". A grandeza "cdte-temp" deu origem às variáveis "cdte-tempCT", "cdte-tempMD", "cdte-tempCM", "cdte-tempPG", "cdte-tempPB" e "cdte-tempCP", que são variáveis advindas respectivamente dos campi Curitiba, Medianeira, Campo Mourão, Ponta Grossa, Pato Branco e Cornélio Procopio.

Para além das informações de nomenclatura, cada uma das tabelas recebeu informações complementares, como no caso de inventário, em que foi elaborada uma coluna para descrever a função do item, outra para o modelo, para o período de manutenção (em dias) e um campo específico para o *datasheet*. Na tabela dos dispositivos, além do nome e número de série, foram colocados campos para identificar se o dispositivo estava ativo, a data de instalação dele, observações relevantes quanto à sua instalação e um campo de descrição sobre o funcionamento do objeto. Em grandezas, foi atribuído um campo para a unidade (porcentagem, voltagem, quilometragem, entre outras), o valor mínimo e o valor máximo que pode existir, a natureza (se é do tipo medida ou calculada), e um campo para descrever a grandeza. As variáveis foram definidas em medidas de confiança e incerteza quanto às medidas, informações advindas dos próprios dispositivos.

4.11 Especificação para Exportação de Dados Utilizando APIs

Respeitando um dos objetivos específicos deste trabalho, foi realizada a especificação de uma API (Interface de Programa de Aplicação), a qual permite tanto internos quanto externos ao projeto realizarem consultas aos dados gerados, limitando-se a somente acessar informações que deveriam ter acesso, e também permitindo realizar a exportação desses dados.

A principal motivação de se coletar os dados é a possibilidade de utilização deles por parte de pesquisadores. Por isso, além de coletar e armazenar os dados, é necessário disponibilizá-los de forma fácil para uso. O sistema a ser desenvolvido, permite o monitoramento e a visualização dos dados gerados, mas somente para quem possui acesso ao sistema, sendo relevante o desenvolvimento de uma outra maneira para a utilização desses dados fora do sistema. De modo que foram especificadas formas para exportação dos dados e das informações relevantes aos pesquisadores, resultando na elaboração de uma Interface de Programa de Aplicação (API).

A ideia foi estruturar uma maneira em que, pelo próprio sistema, fosse possível realizar a criação de *tokens*, ou seja, chaves que possibilitassem as requisições via a API. Uma interface de criação de chaves de acesso no próprio sistema foi feita, possibilitando os usuários a somente criarem chaves que permitam a requisição de dados que queiram permitir o acesso. E, ainda, para um maior controle, para cada requisição são guardados os

seguintes atributos de log: o “timestamp” da requisição, qual foi a “chave” que requisitou, qual o “user-agent” do usuário, qual o “ip” do requerente, e qual o “status” da requisição (se obteve sucesso, se o limite de consultas por hora foi alcançado ou se a chave de acesso está expirada). Tais informações são relevantes, para validar se quem está consumindo os dados pela API é realmente a pessoa autorizada a consultá-los.

Essa chave de acesso basicamente é uma string criptografada com 128 caracteres. Quando usada, são registradas informações de quem está usando, tanto o endereço de IP, quanto o navegador que realiza as requisições de dados. Além da criptografia e da verificação do dispositivo que realiza as requisições, foi atribuído um limite de requisições que podem ser feitas com a chave gerada. Tal limitação foi feita, para não sobrecarregar o servidor com requisições infinitas desnecessárias, evitando-se assim que um mal uso gere problemas no servidor que recebe as requisições, por excesso de chamadas. Esse limite é configurado por quem gera essa chave de acesso, possivelmente um professor ou outro operador do sistema. Para facilitar o uso da API para quem deseja trabalhar com os dados de saída das requisições, foi implementado na própria chave de acesso o vínculo do formato de dados que deseja de retorno, com as opções de formatos CSV ou JSON. Essa escolha pode ser muito útil, pois em algumas aplicações, que normalmente são feitas por desenvolvedores, será usado o JSON.

Já em casos de uso de estudantes ou pesquisadores, o CSV é mais prático, podendo, inclusive, recuperar os dados diretamente nas planilhas eletrônicas, somente usando uma fórmula da planilha e passando a URL contendo a chave de acesso como parâmetro, conforme ilustrado na Figura 4. O autor elencou esse uso, em especial pelo fato de os pesquisadores utilizarem muito as planilhas, sendo um modo mais prático de uso, dado que, em algumas situações, iriam exportar os dados e no final do processo iriam utilizar em uma planilha eletrônica.

Figura 4 – Exemplo de consulta de dados utilizando o aplicativo Planilhas Google.

3	https://dados.labens.ct.utfpr.edu.br/api/My45NzluMTY5LTk4IiwicHJ2IjoNTg			
4				
5	id_variavel	timestamp	valor	flag
6	glo_std-so01CM	8/26/2020 14:30:00	1.444717	0
7	glo_min-so01CM	8/26/2020 14:30:00	718.8898	0
8	glo_max-so01CM	8/26/2020 14:30:00	723.5204	0
9	dif_avg-so01CM	8/26/2020 14:30:00	95.9282	0
10	dif_std-so01CM	8/26/2020 14:30:00	0.0670754	0
11	dif_min-so01CM	8/26/2020 14:30:00	95.83139	0
12	dif_max-so01CM	8/26/2020 14:30:00	96.08446	0
13	dir_avg-so01CM	8/26/2020 14:30:00	780.837	0
14	dir_std-so01CM	8/26/2020 14:30:00	1.441931	0
15	dir_min-so01CM	8/26/2020 14:30:00	777.9679	0
16	dir_max-so01CM	8/26/2020 14:30:00	782.3997	0

Fonte: Autoria própria

4.12 Especificações quanto à Segurança

Algumas definições e alguns cuidados quanto à segurança do sistema foram feitas, observando-se os seguintes dados de interesse:

- a) dados gerados pelas estações solarimétricas e fotovoltaicas;
- b) registros de log;
- c) aqueles inseridos por usuários a título de gestão, cadastros de manutenção, usuários, dispositivos, grandezas, variáveis, contatos, dentre outros;
- d) chaves de acesso para consultas via API;
- e) cadastro de comandos de fator de potência para os inversores das estações;
- f) e permissões de usuários.

Os dados do item a) são de grande importância para o laboratório, uma vez que existe investimento nos equipamentos, espaço e pessoal para a coleta desses dados, tendo sua importância em sua coleta e manutenção correta, bem como a disponibilidade somente ao pessoal autorizado. Em relação ao item b), todos os logs são importantes para entendimento do funcionamento do sistema, verificação e validação do pessoal ativo que utiliza o sistema. O item c) traz informações importantes geradas por usuários aptos a inserir, sendo capital intelectual investido no preenchimento. O item d) é importante, pois, se mal configurado, pode permitir acesso a informações não abertas, bem como prejudicar o servidor que hospeda a API, por excesso de requisições. No item e) existe uma atenção especial, pois se não for bem configurado, comandos errados ou mal intencionados podem danificar os equipamentos. O item f) é também importante, pois são as permissões de usuários que autorizam ou desautorizam acesso a determinadas informações e ações no sistema.

Sistemas que não possuem quaisquer medidas de segurança são facilmente comprometidos quando atacados. Um sistema web que não tem protocolo de segurança, ou seja, criptografia, está vulnerável à interceptação de dados entre cliente e servidor (ALVES, 2014). Um sistema que não possui autenticação para login, permite que qualquer usuário tenha acesso às informações dele. Além disso, se existem níveis diferentes de permissões que cada um deve ter, é necessário que sejam criados perfis de usuários diferentes. Acesso direto ao banco de dados para coleta de dados pode também ser perigoso, pois quem acessa diretamente tem acesso às tabelas e pode fazer praticamente qualquer coisa com os dados existentes. Também é interessante ter um mediador entre o BD e o cliente para a implementação das próprias regras de negócio. Neste projeto, um dos acessos mais críticos é o de envio de comandos para os inversores, em que a configuração de um fator errado de potência neles pode causar problemas. E essa configuração errada pode acontecer por ao menos três situações: 1) Envio acidental de valores incorretos; 2) Valor incorreto inserido propositalmente; 3) Falha de sistema que gerou valores errados; 4) Erros humanos que podem acontecer, e visando a integridade dos dados, é importante exigir confirmação do usuário, quanto a alguma modificação importante (GUERRA, 2020). Se

possível e necessário, até mesmo o uso de um segundo fator de confirmação, em questão de disponibilidade de dados, e também de segurança, é importante que a aplicação e o banco de dados estejam em servidores distintos. Em especial porque caso ocorra de o servidor de aplicação ficar indisponível, o de dados não necessariamente ficará.

Tendo a descrição dos dados de interesse e dos possíveis riscos de segurança, pode-se descrever as tecnologias e abordagens idealizadas visando a sua proteção. São elas: a implementação do HTTPS (Hyper Text Transfer Protocol Secure); a criação da confirmação de comandos de create, update e delete; a utilização de máscaras nos campos (validação de C.P.F para o login e a inserção de dados); desenvolvimento de quatro níveis de usuários: Admin, Manager, Guest e User; a criação de virtualização de servidores (apesar de ser um único servidor físico, o servidor de aplicação e de dados estão em máquinas virtuais diferentes); o envio de e-mail solicitando confirmação para mudanças específicas; uma interface de programação de aplicação; a utilização do NIGINX como web server (gateway de API); e a criação de chaves de acesso para as APIs (JSON Web Token).

5 RESULTADOS

Este capítulo apresenta os resultados obtidos na gestão do projeto de software para as estações solarimétricas e fotovoltaicas implantadas nos campi da UTFPR. Alguns exemplos de telas do artefato final, o software, são apresentados para um melhor entendimento do resultado.

5.1 Gerenciamento de Projeto de Software

O projeto foi desenvolvido em meio a uma pandemia, sendo necessárias algumas mudanças em relação a como se desenvolveria o projeto. A principal delas, foi em relação à comunicação, a qual precisou ser 100% digital, tendo somente reuniões online via videoconferências. Contudo, apesar da necessidade de se adaptar frente a essa nova realidade, houveram algumas praticidades quanto às reuniões serem feitas de forma remoto, afetando diretamente a gestão do projeto. Por exemplo, as reuniões que envolviam diferentes partes poderiam ser sequenciais, uma reunião seguida da outra, sem a necessidade de locomoção de nenhuma das partes. Também, houveram casos em que foi necessário convidar mais pessoas para uma reunião já agendada, e, durante a reunião, por ser remota, a pessoa conseguiu entrar rapidamente, em especial por não necessitar locomover-se até o outro local.

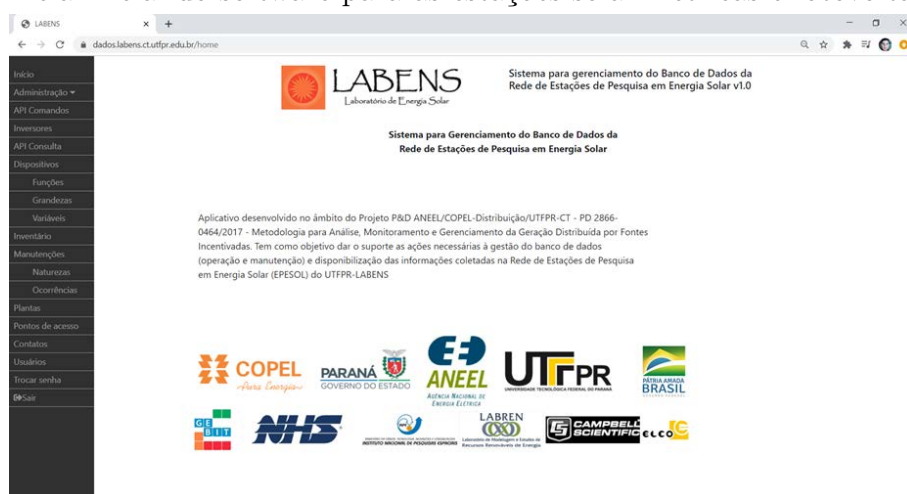
Apesar das mudanças trazidas pela pandemia, o maior desafio foi no entendimento do projeto como um todo, dada sua complexidade, tendo diferentes tecnologias de sistemas fotovoltaicos e solarimétricos instalados em diferentes campi e com tecnologias embutidas distintas, gerando variáveis distintas. Um exemplo da dimensão dessa complexidade, está no número de variáveis geradas, passando de 600. Mas, com a devida leitura das documentações, e visita a um dos campus, bem como inúmeras reuniões com outros envolvidos no projeto do LABENS, foi possível o entendimento. Permitindo assim, trabalhar no projeto e na gestão do desenvolvimento do software. O segundo maior desafio encontrado foi delimitar o escopo do projeto, pois os recursos eram finitos, mas as ideias e necessidades eram inúmeras. Levantar e entender todas as sugestões de requisitos, para filtrar os mais importantes a serem priorizados, foi desafiador, mas os próprios usuários auxiliaram nessa definição, quando perguntados da relevância desse uso para eles. Mas, mesmo assim, com inúmeras ideias novas surgindo a cada momento, foi necessário trabalhar em um aditivo ao contrato, implementando um novo módulo de desenvolvimento, voltado para o envio de comandos para os inversores, via uma API específica a ser desenvolvida para esse propósito.

O projeto possuía prazo de término já definido pelo financiador, sendo necessário, muitas vezes, respostas rápidas à empresa contratada, para que pudessem cumprir com os prazos repassados a eles. Em alguns módulos, existiram atrasos, por consequência de

atrasos tardios de algumas das partes, mas, em outros módulos, foram mais rápidos do que o estimado, compensando assim o tempo. Mas sempre com reuniões periódicas, semanais, para acompanhamento do projeto, das entregas e para sanar eventuais dúvidas. Como algumas das demandas da empresa contratada eram dúvidas sobre questões que já estavam documentadas, o maior trabalho foi, para além de entregar a documentação, explicar eventuais dúvidas que poderiam ter para além da documentação. E, sempre que possível, documentações pendentes iam sendo desenvolvidas antes que a empresa precisasse delas, para que quando chegassem até os módulos que fossem necessitar delas, já estivessem prontas.

Algumas prioridades tiveram que serem revistas, inclusive pela questão do tempo de término das documentações, homologações e revisões de prioridades. Sendo necessário ou adiantar algum requisito, ou até mesmo substituí-lo, também tendo a necessidade de um aditivo. Mas, ainda assim, os prazos de término final foram respeitados e os requisitos propostos foram finalizados e homologados a tempo, evidenciando o sucesso do projeto, inclusive na contratação da empresa.

Figura 5 – Tela inicial do software para as estações solarimétricas e fotovoltaicas pronto.



Fonte: Autoria própria

5.2 Acompanhamento do Desenvolvimento do Sistema

Os requisitos para o sistema foram levantados e organizados, para posteriormente serem apresentados à empresa contratada para a realização do desenvolvimento do software. Sendo a maior parte desses requisitos apresentados de forma clara, visto que após a explicação desenvolveram conforme solicitado. Mas, em alguns casos, na hora da homologação, percebeu-se que alguns (a minoria) não foram tão claros, visto que houve a necessidade de explicar novamente quais eram os requisitos propostos inicialmente. Também, novos requisitos surgiram e tiveram que ser inclusos durante o andamento do desenvolvimento do

projeto. Sendo esses, quando definidos como importantes, acrescidos no escopo do projeto, para futura implementação.

A cada término dos módulos de desenvolvimento, eram liberadas versões parciais de teste, para homologação, em que se faziam alguns testes de usabilidade, e todos os eventuais problemas e desconformidades encontrados eram documentados e repassados para a empresa contratada para correção. Sempre seguindo nessa dinâmica, em que a empresa desenvolvia e a equipe do LABENs validava para então homologar, ou não, cada módulo, com processos sempre intermediados pelo gestor do projeto. Foram, então, desenvolvidas e homologadas todas as 50 telas do sistema, aproximadamente.

5.3 Especificações quanto à Segurança e à Exportação de Dados

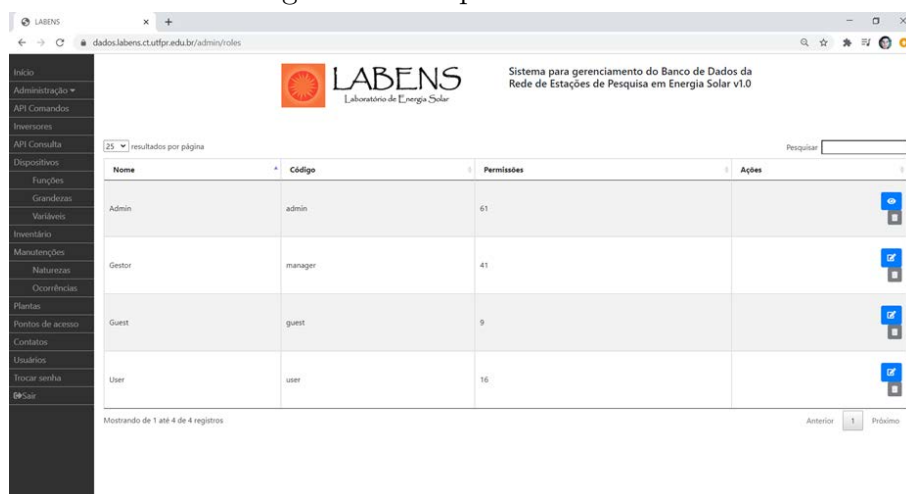
Algumas especificações quanto à segurança foram feitas para maior proteção dos dados. Por exemplo, a implementação do HTTPS (Hyper Text Transfer Protocol Secure), garantindo um tráfego de dados mais seguro na rede. Especificações quanto ao login, o qual é através do C.P.F. do usuário, tendo número máximo de tentativas de login com senha errada igual a 5, sendo que quando atingido esse limite, é necessário esperar 1 minuto para retornar às tentativas, sendo preciso que, em caso de tentativa de recuperar a senha, um administrador do sistema deverá reconfigurá-la, e, em caso de ociosidade no sistema, após duas horas, o login é desconectado do sistema, por inatividade. E após logado ao sistema, todas as tentativas de edição, criação e remoção de dados são passíveis de confirmação da ação a ser realizada, para que comandos acidentais não ocorram.

Ainda pensando na segurança do sistema, foram criados quatro níveis de usuários, com diferentes níveis de permissões: Admin, Manager, Guest e User (do maior ao menor nível de permissão, respectivamente). Em que: Admin possui 61 permissões, sendo capaz de realizar todas as ações de CRUD (Create, Read, Update, Delete) no sistema, enquanto Guest só pode ler (Read). E todos os perfis de usuários possuem data de expiração da conta, em que, após o período pré-determinado de validade esta será ou excluída ou inativada, garantindo assim que usuários temporários não sejam esquecidos com acesso após perderem o vínculo com o projeto.

Além de ser desenvolvido um botão para permitir baixar os dados através do sistema, foram desenvolvidas uma API específica para uso externo e consultas com a possibilidade de retorno dos dados em formato .CSV e .JSON. De modo que se possa usar para conectar com sistemas, softwares ou até mesmo planilhas eletrônicas a API, para a coleta e manipulação dos dados gerados. Para um maior controle do uso da API, garantindo que usuários específicos tenham acesso a informações específicas, foram desenvolvidas chaves de API, ou seja, chaves usadas através da API, que permitem somente o retorno de determinadas consultas.

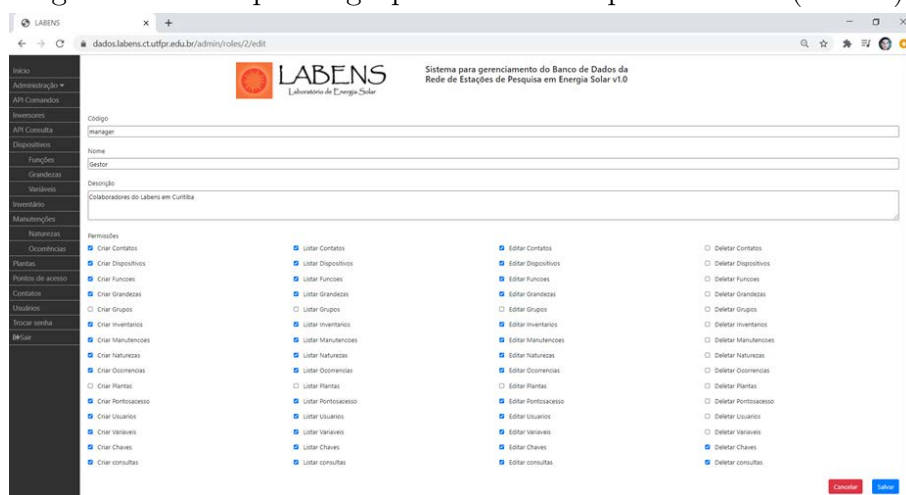
Na Figura 8, pode-se visualizar a tela para a configuração da chave de API. Na qual pode-se atribuir: um nome para posterior gestão das chaves; uma data limite de

Figura 6 – Grupos de Usuários.



Fonte: Autoria própria

Figura 7 – Exemplo de grupo de usuário - perfil Admin (Gestor).



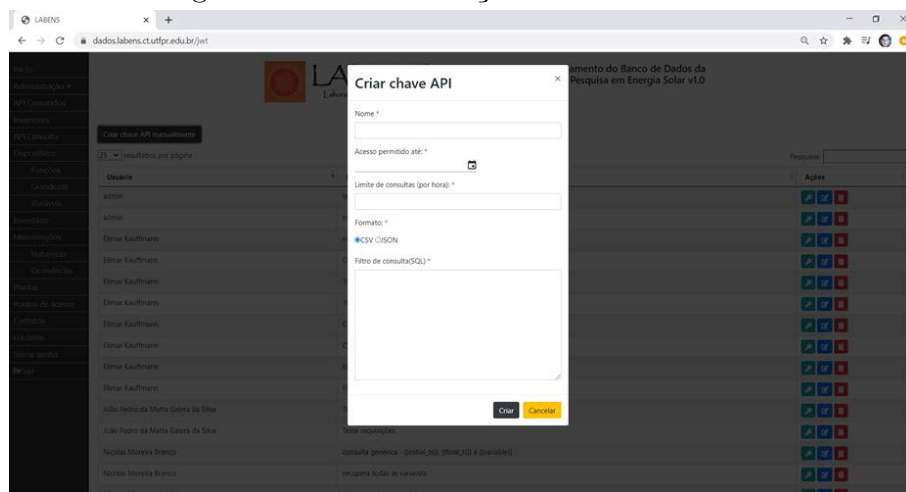
Fonte: Autoria própria

validade, sendo que, após este período, a chave é excluída automaticamente, sem o gestor ter a preocupação de buscar as chaves existentes e excluir uma a uma, quando não lhe interessar mais essas chaves; a quantia de requisições que podem ser feitas por ora, para que não sobrecarregue o sistema em caso de inúmeras tentativas sequenciais de uma mesma consulta; o formato de retorno da requisição, conforme mencionado; e o filtro de consultas, que deve ser escrito em SQL (Structured Query Language), o qual realizará a consulta no banco de dados, mas somente com permissão de leitura.

Depois de salvar a chave, é gerada uma string criptografada de 128 caracteres, a qual pode ser utilizada para consultas através da API. A Figura 9 ilustra como é essa chave gerada, podendo ser usada na estrutura apresentada a seguir.

https://dados.labens.ct.utfpr.edu.br/api/ CHAVE_DE_ACESSO

Figura 8 – Tela de criação de Chave de API.



Fonte: Autoria própria

A string "CHAVE_DE_ACESSO" deve ser substituída pela chave criptografada de 128 caracteres. Também é possível realizar consultas passando parâmetros, de forma a otimizar o retorno, sem precisar de um pós-tratamento para filtrar a requisição. Sendo a consulta similar a essa estrutura: "https://dados.labens.ct.utfpr.edu.br/api/CHAVE_DE_ACESSO?data=2020-06-06", em que o parâmetro é a data, 2020-06-06, obtendo somente como retorno da consulta para esse dado filtro o dia específico passado.

Figura 9 – Exemplo de Chave Gerada.

Chave de Acesso

My45NzluMTY5LTk4liwicHJ2IjoiNTg3MDg2M2Q0YT
YyZDc5MTQ0M2ZhZjkzNmZjMzY4MDMxZDExMGM
0ZiJ9.4rQMps_zmCxN2YgU9m7MvtUoBrpKW58PVxF
aK4pHo0k

Copiar chave API

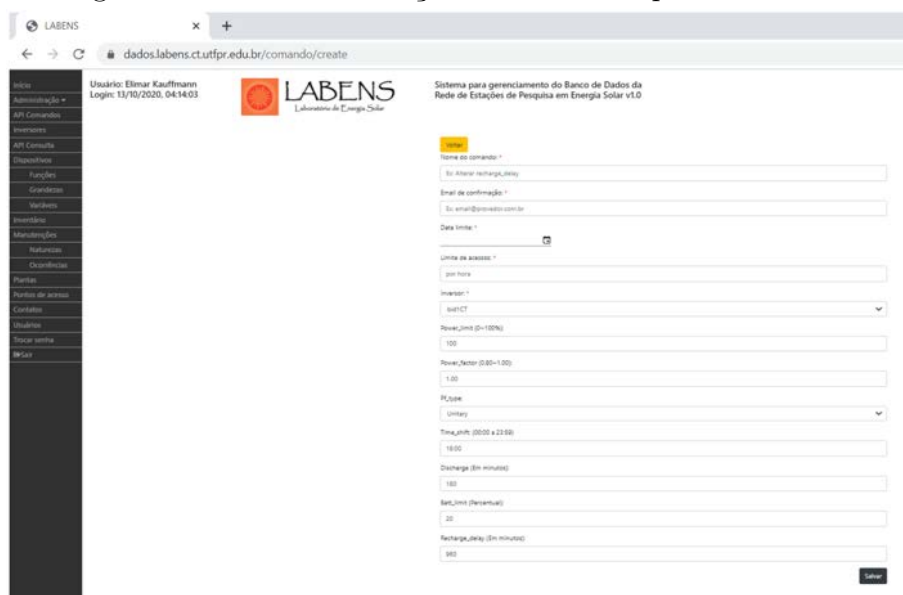
Fechar

Fonte: Autoria própria

Além da API de consulta as informações geradas pelos sistemas fotovoltaicos e solarimétricos, foi desenvolvida uma API para o envio de comandos para os inversores dos sistemas fotovoltaicos. Porém, alguns cuidados foram necessários, dado que comandos errados podem danificar o equipamento. Na Figura 10, pode-se visualizar a tela onde se cria um comando para os inversores, sendo que, após o comando ser criado, um e-mail é enviado para o endereço de e-mail configurado na criação do comando, conforme Figura

11, sendo necessário esse usuário estar logado no sistema, para confirmar o comando. Caso a confirmação não seja feita em até uma hora, o comando é cancelado por chegar ao tempo limite de espera, e um e-mail é enviado avisando sobre o cancelamento, conforme 11.

Figura 10 – Tela de Criação de Comando para Inversores.



Fonte: Autoria própria

Figura 11 – E-mail de confirmação de comando para inversor.



Fonte: Autoria própria

Figura 12 – E-mail de cancelamento de comando para inversor.



Fonte: Autoria própria

Se o comando for confirmado no tempo correto, irá acontecer a tentativa de comunicação, ou seja, o envio de comando para o inversor. Se em 24 horas for possível realizar o comando corretamente, um e-mail para quem criou o comando e para todos

os administradores é enviado, e, se não, em caso de falha, notifica quem o criou e quem tentou executá-lo (através do e-mail).

Para um maior controle de acesso ao sistema, um controle de registros de logins foi realizado, sendo possível identificar qual usuário realizou login, e em que período e por qual dispositivo. Além desse controle por login, também foram implementados registros por uso de requisições de APIs por chaves geradas, e de tentativas de envio de comandos para os inversores.

De modo que as especificações quanto à segurança e à exportação de dados, bem como a implementação delas, foram satisfatórias até o momento do término do projeto.

6 CONSIDERAÇÕES FINAIS

O presente trabalho realizou a análise do gerenciamento do projeto do software para monitoramento de estações solarimétricas e fotovoltaicas implantadas em alguns campi da UTFPR de forma bem-sucedida. Mas, houveram alguns desafios no percurso, sendo os principais: a gerência totalmente remota, pois foi durante a pandemia da Covid-19; a localização de projetos correlacionados, pois existe uma grande variedade de trabalhos científicos voltados para a gerência de software, para o trabalho com estações solarimétricas ou fotovoltaicas, mas poucos com as três temáticas juntas, o que exigiu um empenho maior em juntar as partes para se realizar uma gerência bem-sucedida.

Os principais objetivos foram cumpridos, sendo especificados meios para a exportação de dados e informações relevantes aos pesquisadores, com alguns mecanismos de segurança, utilizando APIs como interface, mas mantendo a praticidade para quem não tem domínio de linguagens de programação, uma vez que se pode utilizar a API em uma planilha eletrônica, que é de mais amplo conhecimento. O sistema, de forma geral, também conta com alguns mecanismos de segurança para evitar problemas com pessoas má intencionadas ou para eventuais erros que podem ser cometidos pelos usuários, bem como faz gestão do projeto como um todo, organizando requisitos existentes, analisando a necessidade de novos, acompanhando o desenvolvimento do sistema com a empresa contratada, validando o cumprimento desses requisitos e realizando o trabalho de intermediação entre usuários e desenvolvedores do sistema.

Dado todo o conhecimento adquirido e os resultados obtidos, seria interessante aprofundar em alguns requisitos que não foram priorizados em relação a outros, devida as limitações de recursos. Deixando, assim, o sistema com outras funcionalidades que também poderiam ser úteis aos possíveis usuários.

Como sugestão de trabalho futuro indicamos a possibilidade de realizar análises com gráficos gerados pelo próprio sistema. Ou, até mesmo, a geração automática de relatórios, com filtros específicos a serem monitorados. Também alertas de inconsistência nos dados monitorados, dado um padrão a ser seguido de monitoramento, ou seja, os mínimos e os máximos possíveis, bem como a média que deveria seguir, pensando na média histórica.

Para além da continuidade desses requisitos, durante o desenvolvimento deste projeto, percebeu-se que para alguns usuários seria interessante ter a possibilidade de realizar as mesmas funcionalidades do sistema web, em dispositivos móveis, como o *smartphone*.

Referências

- AGOSTINI, M. N. **Nova Filosofia para o Projeto de Software para Sistemas de Energia Elétrica Usando Modelagem Orientada a Objetos**. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2002. Citado na página 18.
- AHMED, A. **Software Project Management**. 6000 Broken Sound Parkway NW, Suite 300: Taylor & Francis Group, 2012. ISBN 978-1-4398-4656-8. Citado na página 25.
- ALVES, P. **O que é HTTPS e como ele pode proteger a sua navegação na internet**. 2014. Url=<https://www.techtudo.com.br/noticias/noticia/2014/02/o-que-e-https-e-como-ela-pode-protger-sua-navegacao-na-internet.html>. Citado na página 46.
- AMBLER, S. W. **Agile Documentation**. [S.l.: s.n.], 2004. Citado na página 30.
- ANEEL. **Agile Documentation**. 2. ed. [S.l.]: Catalogação na Fonte Centro de Documentação - CEDOC, 2005. ISBN 85-87491-09-1. Citado na página 15.
- ANEEL. **PROCEDIMENTOS DO PROGRAMA DE EFICIÊNCIA ENERGÉTICA - PROPEE**. [S.l.], 2013. Citado na página 30.
- ANEEL. **Chamada Nº. 001/2016 Projeto Prioritário de Eficiência Energética e Estratégico de P&D: “Eficiência Energética e Minigeração em Instituições Públicas de Educação Superior”**. [S.l.], 2016. Citado 2 vezes nas páginas 30 e 31.
- ARTERO, M. A. **Gestão de projetos de software**. Avenida Paris, 675 – Parque Residencial João Piza, CEP: 86041-100 — Londrina — PR: Editora e Distribuidora Educacional S.A., 2016. ISBN 978-85-8482-671-1. Citado 2 vezes nas páginas 25 e 26.
- BENCK, L. L. N.; GURA, E. F. **Construção de um data warehouse, aliado a uma ferramenta open source ireport na geração de informações para tomada de decisão**. 2011. Citado na página 22.
- BERNERS-LEE, T. **The World Wide Web: Past, Present and Future**. 1996. Url=<https://www.w3.org/People/Berners-Lee/1996/ppf.html>. Citado na página 17.
- BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. Rua Sete de Setembro, 111 - 16º Andar, Rio de Janeiro-RJ: Elsevier Editora Ltda., 2015. ISBN 978-85-352-2627-0. Citado 2 vezes nas páginas 21 e 22.
- BEZERRA, F. D. **Energia solar fotovoltaica**. p. 18, 2018. Citado na página 15.
- COPEL. **Chamada Pública Copel 001/2017 Projeto Prioritário de Eficiência Energética e Estratégico de P&D**. [S.l.], 2017. Citado na página 31.
- COPEL. **RESULTADO PROJETOS SELECIONADOS – CHAMADA PÚBLICA VPDE 001/2017**. [S.l.], 2017. Citado na página 12.
- COSTA, G. d. S.; DIAS, Y. **Gain: Proposição de um método de gamification aplicado à promoção da inovação da tecnologia**. 2017. Citado 2 vezes nas páginas 21 e 22.

DUTRA, J. C. et al. Uma análise do panorama das regiões missões e noroeste do estado do rio grande do sul sob o prisma da energia eólica e solar fotovoltaica como fontes alternativas de energia. **Revista Paranaense de Desenvolvimento - RPD**, v. 34, n. 124, 2013. ISSN 2236-5567. Disponível em: <<http://www.ipardes.pr.gov.br/ojs/index.php/revistaparanaense/article/view/547>>. Citado na página 12.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. Av. Ermano Marchetti, 1435 - Lapa - São Paulo - SP: Pearson Education do Brasil Ltda., 2005. ISBN 85-88639-17-3. Citado na página 22.

FERREIRA, A. B. d. H. **Novo dicionário da língua portuguesa**. [S.l.]: Editora Nova Fronteira, 1994. Citado na página 18.

GUERRA, B. **O que é integridade de dados e por que ela é importante**. 2020. Url=<https://blog.in1.com.br/o-que-%C3%A9-integridade-de-dados-e-por-que-ela-%C3%A9-importante>. Citado na página 46.

GUIDE, P. **A Guide to the PROJECT MANAGEMENT BODY OF KNOWLEDGE**. 14 Campus Boulevard, Newtown Square, Pennsylvania 19073-3299 USA: Project Management Institute, Inc., 2017. ISBN 978-1-62825-184-5. Citado 2 vezes nas páginas 25 e 26.

HEUSER, C. A. **Projeto de Banco de Dados**. [S.l.]: Digital Source, 1998. Citado na página 23.

HEUSER, C. A. **Projeto de Banco de Dados**. Porto Alegre: Artmed Editora SA, 2009. ISBN 978-85-7780-452-8. Citado na página 23.

HUNTER, K. L. **Irresistible APIs - Designing web APIs That Developers Will Love**. 20 Baldwin Road, PO Box 761, Shelter Island, NY 11964: Manning Publications Co., 2017. ISBN 9781617292552. Citado na página 19.

ISO. **ISO/IEC 27032 - Information technology - Security Techniques - Guidelines for cybersecurity**. [S.l.], 2012. Citado na página 19.

JACOBSON, I. Use cases and aspects - working seamlessly together. **ETH Zurich**, p. 22, 2003. Citado na página 21.

JIN, B.; SAHNI, S.; SHEVAT, A. **Designing web APIs - Building APIs That Developers Love**. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2018. ISBN 978-1-492-02692-1. Citado na página 20.

LABENS. **LABENS - Laboratório de Energia Solar da UTFPR**. 2019. Url=<http://labens.ct.utfpr.edu.br/>. Citado 2 vezes nas páginas 31 e 32.

LIRA, A. L. d. O.; SOARES, B. d. L.; SANTOS, S. d. A. Estação solarimétrica de referência – instalação, operação e manutenção. **Revista Brasileira de Energia Solar**, v. 7, n. 2, p. 132–139, 2016. Citado na página 14.

MUNOZ, A. R. Interfaz gráfica para el análisis del funcionamiento de paneles solares fotovoltaicos de concentración. p. 149, 2017. Citado 2 vezes nas páginas 27 e 28.

NETO, M. B. d. S. brmodeloweb: Ferramenta web para ensino e modelagem de banco de dados. 2016. Citado na página 24.

OLCOTT, J. **Cybersecurity Vs. Information Security: Is There A Difference?** 2019. Url=<https://www.bitsight.com/blog/cybersecurity-vs-information-security>. Citado na página 19.

OLIVEIRA, D. d. P. R. de. **Sistemas, Organização & Métodos**. São Paulo, SP, BR: Editoras Atlas S.A., 2006. ISBN 85-224-4515-X. Citado na página 18.

OLIVEIRA da R. A. R.; PERIN, G. F. Panorama da energia solar fotovoltaica no brasil. **Revista Brasileira de Energia Solar**, v. 7, n. 2, p. 140–147, 2016. Citado 2 vezes nas páginas 12 e 14.

PANHAN, A. M. **Sistema de Aquisição de Dados e Monitoramento Remoto para Câmaras Frias e Sistemas de Refrigeração**. 94 p. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, 2002. Citado 2 vezes nas páginas 18 e 19.

PEQUENO, P. V. d. O. et al. Conceptor - a tool for creating and maintaining the entity-relationship model and automatic generation of sql statements for the database. 2020. Citado na página 23.

PEREIRA, B. R. Energia solar fotovoltaica: fundamentos e aplicações. 2008. Citado na página 12.

PEREIRA, E. B. et al. Atlas brasileiro de energia solar. v. 2, 2017. Citado na página 14.

PONTES, W. M. C. S. **Implantação de Estação Solarimétrica e de Sistema Supervisório com SCADABR e Plataforma IoT em Usina Fotovoltaica na UNILAB-CE**. 149 p. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, 2021. Citado 2 vezes nas páginas 27 e 28.

PRESSMAN, R. S. **Engenharia de Software Uma Abordagem Profissional**. Porto Alegre, RS, BR: AMGH Editora Ltda., 2011. ISBN 978-85-63308-33-7. Citado 5 vezes nas páginas 15, 16, 17, 18 e 20.

RASLAN, D. A.; CALAZANS, A. T. S. Data warehouse: conceitos e aplicações. 2014. Citado na página 22.

RODRIGUES, D. B. A importância da energia solar: Fotovoltaica. p. 40, 2018. Citado na página 15.

RUSCHEL, C. S. **Desenvolvimento de software para supervisão de usinas solares fotovoltaicas**. 106 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015. Citado 2 vezes nas páginas 27 e 28.

SANTOS, W. E. dos; URBANETZ, J.; GOUVEA, D. C. de. Monitoramento de painel fotovoltaico através de computação em névia integrado à rede gsm. p. 6, 2018. Citado na página 14.

SOMMERVILLE, I. **Engenharia de Software**. Rua Nelson Francisco, 26 - São Paulo - SP - Brasil: Pearson Education do Brasil, 2011. ISBN 978-85-7936-108-1. Citado 3 vezes nas páginas 20, 21 e 22.

TEM-PASS, E. H.; SOUZA, M. B. de; IATSKIU, T. Estudo econômico para implantação de sistemas fotovoltaicos conectados à rede elétrica de energia no estado do paraná. p. 131, 2016. Citado na página 14.

TIEPOLO, G. M. **Estudo de Potencia de Geração de Energia Elétrica Através de Sistemas Fotovoltaicos Conectados à Rede no Estado do Paraná**. 230 p. Dissertação (Mestrado) — Pontifícia Universidade Católica do Paraná Escola Politécnica, Curitiba Paraná, 2015. Citado na página 14.

VILLAFIORITA, A. **Introduction to Software Project Management**. 6000 Broken Sound Parkway NW, Suite 300: CRC Press, 2014. ISBN 978-1-4665-5954-7. Citado na página 25.

WEB, B. M. **Ferramenta para modelagem de banco de dados**. 2022. Url=<https://www.brmodeloweb.com/>. Citado na página 24.

Apêndices

APÊNDICE A – Reuniões Realizadas

Este apêndice relaciona as reuniões ocorridas entre GEBIT e LABENS, reativas ao desenvolvimento do sistema.

Participaram das reuniões com a empresa contratada, para o avanço do projeto, dois representantes da empresa contratada, incluindo um analista de sistemas de informação formado pela própria UTFPR, um dos professores do LABENS e o autor deste trabalho. Em algumas reuniões específicas houve a participação de outros elementos para pautas pontuais. Os registros das ATAs foram feitas pelo autor, seguindo o resumo delas:

Reunião 1 (08/04/2020) - A primeira reunião consistiu em apresentar o DER existente do banco de dados. Alguns questionamentos foram feitos, e alguns possíveis ajustes foram comentados. Acordou-se que durante a semana todas as tabelas do banco ⁶³ APÊNDICE A – Reuniões Realizadas Este apêndice relaciona as reuniões ocorridas entre GEBIT e LABENS, relativas ao desenvolvimento do sistema. Participaram das reuniões com a empresa contratada, para o avanço do projeto, dois representantes da empresa contratada, incluindo um analista de sistemas de informação formado pela própria UTFPR, um dos professores do LABENS e o autor deste trabalho. Em algumas reuniões específicas houve a participação de outros elementos para pautas pontuais. Os registros das ATAs foram feitas pelo autor, seguindo o resumo delas: Reunião 1 (08/04/2020) - A primeira reunião consistiu em apresentar o DER existente do banco de dados. Alguns questionamentos foram feitos, e alguns possíveis ajustes foram comentados. Acordou-se que durante a semana todas as tabelas do banco seriam criadas, e que posteriormente entrariam em contato para ajustes da estrutura do BD, a nível de servidor. O laboratório também fica responsável de enviar os .CSVs gerados pelos equipamentos, que servirão como entrada no banco (arquivos de exemplo). Nesta mesma reunião definiu-se que o Banco de Dados a ser utilizado é o PostgreSQL (uma escolha interessante para o LABENS, por ser Open Source). Também definiu-se em conjunto qual seria a estrutura dos servidores, de aplicação e do Banco de Dados, com uma explicação de como seria e comprometimento de enviar os acessos com as explicações necessárias. Reunião 2 (15/04/2020) - Nesta reunião foi comentado sobre uma documentação que fora enviada para a contratada, explicando o funcionamento dos StOnes e dos Dataloggers, a mesma havia analisado e entendido o funcionamento, mas questionou em relação ao padrão de nomenclatura das variáveis, se seria a mesma que constava nos arquivos de saída dos dados de exemplo. Mas foi explicado que não, pois uma padronização dos nomes ainda seria elaborada, pois como entre os equipamentos existem diferentes tecnologias, nem todos os arquivos de saída seriam iguais. Mencionou-se da necessidade de quando os

dados forem importados para o banco, que uma verificação de mínimo e máximo no valor fosse feita, com uma maneira de sinalizar, quando o valor passasse desses limites, dado que se isso ocorrer, o dado está errado e precisará de atenção para correção. Também frisou-se a importância de para além de uma importação automática, ter a possibilidade de uma importação em massa de forma manual. Uma preocupação demonstrada para a contratada, foi a questão do desempenho das consultas, pois ao longo do tempo a tendência é crescer o número de registros. Mas a empresa menciona que precisaria estudar melhor o caso, para dar um parecer sobre. Também falou-se da necessidade de criar uma API, para as consultas ao banco, com algumas restrições, onde cada chave de acesso somente APÊNDICE A. Reuniões Realizadas 64 permitiria consultar unicamente o que foi previamente definido. Definiu-se como um possível fluxo de trabalho seguinte, a ativação do banco de dados, em seguida o preenchimento de suas tabelas, para então o desenvolvimento de uma automação de importação dos dados, após isso a criação de uma tela de gestão desses dados (um sistema CRUD), a qual deveria ter políticas de usuários e permissões. Tais políticas deveriam ser muito bem desenhadas e pensadas para não serem tão específicas, para não demandar tantas horas de trabalho do projeto, pois a depender da política, pode ser algo muito custoso. Ficando o pendente da contratada definir exatamente essas políticas e encaminhá-las (inicialmente contendo 4 grupos de perfis). O algoritmo de importação a ser criado, deverá estar preparado para o recebimento de dois padrões de dados (duas possíveis entradas de dados) e em caso de ter variáveis não mapeadas, deverá ignorá-las (visando não importar informações desnecessárias). Reunião 3 (22/04/2020) - Atualizações em relação ao desenvolvimento do projeto foram realizadas, onde a base do sistema já havia sido iniciada e a empresa contratada esperava o envio do arquivo que seria usado para a realização das importações para término de configuração do algoritmo. Arquivos o qual o laboratório iria terminar e providenciar o envio até o final do mês, enquanto isso, a empresa iria iniciar os trabalhos na autenticação dos tokens. Questionou-se a abordagem da utilização do timestamp, se essa seria a melhor abordagem, ou se haveria alguma alternativa, e a empresa retornou que alguns testes seriam necessários, mas que mesmo que tivesse vantagens, seriam poucas. Atualizações sobre o desenvolvimento das tabelas de usuários foram feitas, as quais ainda não haviam sido criadas e que seria importante já ter uma primeira visão de como será a interface CRUD do banco. Também foi alinhada a entrega dos dados de entrada para as tabelas Inventário, Dispositivos, Grandezas e Variáveis, as quais na mesma semana serão enviadas. O assunto sobre o algoritmo de importação foi retomado, reafirmando a importância de que o mesmo possa realizar rotinas de busca dos dados de forma automatizada (buscando no servidor FTP) e manual, em especial porque alguns dos dados a serem importados já são de meses anteriores e a automação irá trabalhar nos meses subsequentes. Também foi

apresentado a empresa um novo padrão de nomenclatura das variáveis, agora com três letras cada, juntamente com a tecnologia e o local em que o dispositivo está instalado. Os dois padrões de arquivos foram apresentados, com suas particularidades (em especial o cabeçalho que estão em ordens distintas). E que seria enviado uma relação com todas as possíveis variáveis de entrada. Um exemplo de como seria a tela de CRUD do sistema foi apresentada pela contratada, e um compromisso de enviar uma foto da primeira versão foi feita. Questionamentos sobre o horário de verão foram feitos, pensando que se a política de não ter horário de verão mudar, pode-se ter comprometimento na importação dos dados de um dado período, pois entrariam horários repetidos, e como o timestamp é parte da chave primária, não seriam criadas, e que posteriormente entrariam em contato para ajustes da estrutura do BD, a nível de servidor. O laboratório também fica responsável de enviar os .CSVs gerados pelos equipamentos, que servirão como entrada no banco (arquivos de exemplo). Nesta mesma reunião definiu-se que o Banco de Dados a ser utilizado é o PostgreSQL (uma escolha interessante para o LABENs, por ser *Open Source*). Também definiu-se em conjunto qual seria a estrutura dos servidores, de aplicação e do Banco de Dados, com uma explicação de como seria e comprometimento de enviar os acessos com as explicações necessárias.

Reunião 2 (15/04/2020) - Nesta reunião foi comentado sobre uma documentação que fora enviada para a contratada, explicando o funcionamento dos StOnes e dos Dataloggers, a mesma havia analisado e entendido o funcionamento, mas questionou em relação ao padrão de nomenclatura das variáveis, se seria a mesma que constava nos arquivos de saída dos dados de exemplo. Mas foi explicado que não, pois uma padronização dos nomes ainda seria elaborada, pois como entre os equipamentos existem diferentes tecnologias, nem todos os arquivos de saída seriam iguais. Mencionou-se da necessidade de quando os dados forem importados para o banco, que uma verificação de mínimo e máximo no valor fosse feita, com uma maneira de sinalizar, quando o valor passasse desses limites, dado que se isso ocorrer, o dado está errado e precisará de atenção para correção. Também frisou-se a importância de para além de uma importação automática, ter a possibilidade de uma importação em massa de forma manual. Uma preocupação demonstrada para a contratada, foi a questão do desempenho das consultas, pois ao longo do tempo a tendência é crescer o número de registros. Mas a empresa menciona que precisaria estudar melhor o caso, para dar um parecer sobre. Também falou-se da necessidade de criar uma API, para as consultas ao banco, com algumas restrições, onde cada chave de acesso somente permitiria consultar unicamente o que foi previamente definido. Definiu-se como um possível fluxo de trabalho seguinte, a ativação do banco de dados, em seguida o preenchimento de suas tabelas, para então o desenvolvimento de uma automação de importação dos dados, após isso a criação de uma tela de gestão desses dados (um

sistema CRUD), a qual deveria ter políticas de usuários e permissões. Tais políticas deveriam ser muito bem desenhadas e pensadas para não serem tão específicas, para não demandar tantas horas de trabalho do projeto, pois a depender da política, pode ser algo muito custoso. Ficando o pendente da contratada definir exatamente essas políticas e encaminhá-las (inicialmente contendo 4 grupos de perfis). O algoritmo de importação a ser criado, deverá estar preparado para o recebimento de dois padrões de dados (duas possíveis entradas de dados) e em caso de ter variáveis não mapeadas, deverá ignorá-las (visando não importar informações desnecessárias).

Reunião 3 (22/04/2020) - Atualizações em relação ao desenvolvimento do projeto foram realizadas, onde a base do sistema já havia sido iniciada e a empresa contratada esperava o envio do arquivo que seria usado para a realização das importações para término de configuração do algoritmo. Arquivos o qual o laboratório iria terminar e providenciar o envio até o final do mês, enquanto isso, a empresa iria iniciar os trabalhos na autenticação dos *tokens*. Questionou-se a abordagem da utilização do *timestamp*, se essa seria a melhor abordagem, ou se haveria alguma alternativa, e a empresa retornou que alguns testes seriam necessários, mas que mesmo que tivesse vantagens, seriam poucas. Atualizações sobre o desenvolvimento das tabelas de usuários foram feitas, as quais ainda não haviam sido criadas e que seria importante já ter uma primeira visão de como será a interface CRUD do banco. Também foi alinhada a entrega dos dados de entrada para as tabelas Inventário, Dispositivos, Grandezas e Variáveis, as quais na mesma semana serão enviadas. O assunto sobre o algoritmo de importação foi retomado, reafirmando a importância de que o mesmo possa realizar rotinas de busca dos dados de forma automatizada (buscando no servidor FTP) e manual, em especial porque alguns dos dados a serem importados já são de meses anteriores e a automação irá trabalhar nos meses subsequentes. Também foi apresentado a empresa um novo padrão de nomenclatura das variáveis, agora com três letras cada, juntamente com a tecnologia e o local em que o dispositivo está instalado. Os dois padrões de arquivos foram apresentados, com suas particularidades (em especial o cabeçalho que estão em ordens distintas). E que seria enviado uma relação com todas as possíveis variáveis de entrada. Um exemplo de como seria a tela de CRUD do sistema foi apresentada pela contratada, e um compromisso de enviar uma foto da primeira versão foi feita. Questionamentos sobre o horário de verão foram feitas, pensando que se a política de não ter horário de verão mudar, pode-se ter comprometimento na importação dos dados de um dado período, pois entrariam horários repetidos, e como o *timestamp* é parte da chave primária, não seria importado. Mas como esse horário não tem sol, na pior das hipóteses (mudança da política) não teria-se grandes perdas. A preocupação quanto a atualização e correção de alguns dados foi feita, dado que em alguns casos os dados podem serem subidos erroneamente. Algumas sugestões foram dadas e a reunião acabou.

Reunião 4 (29/04/2020) - a reunião iniciou-se com a explicação de que na base de dados o timestamp será salvo com UTC-3. Que alguns cálculos com as variáveis de entrada poderão ser feitos, e os mesmos precisarão de uma tabela específica para serem mantidos, a qual deve ser criada. Uma previsão de uma primeira entrega do sistema foi dado, para duas semanas a partir do dia corrente. A política de usuários que ficou pronta foi apresentada a contratada, explicando suas particularidades, como uma possível restrição de acesso por considerando a qual campus o usuário pertence. Mas que para algumas das solicitações, seria necessário revisar o DER atual, prevendo relações compatíveis, ficando acordado que a contratante iria revisar e que um próximo momento seria apresentada as revisões.

Reunião 5 (06/05/2020) - Os arquivos de importação para as tabelas Inventário, Dispositivo, Grandezas e Inventários foram concluídas e encaminhadas para a empresa realizar a importação. Algumas atualizações foram feitas no DER/Modelo Lógico, e atualizadas para a empresa contratada. Foi mencionada a importância de uma padronização das nomenclaturas das chaves estrangeiras, remetendo as suas respectivas tabelas. Algumas tentativas de ajustes no formato de saída dos dados do Datalogger foram feitos, mas sem sucesso, pois esse dispositivo é menos maleável nesses quesitos. Por isso exceções no algoritmo de importação deverão se feitas, e que para identificar qual dispositivo está trabalhando, basta analisar a primeira linha e coluna do arquivo .CSV. Um atributo chamado *datasheet* foi criado para acompanhar os itens de inventário, que basicamente será um texto que conterá o link de acesso desse documento para consulta. E que o sistema em desenvolvimento, tenha em sua interface um botão para baixar esse arquivo, que sempre será um .PDF. Definiu-se alguns valores padrões para alguns atributos e alguns questionamentos sobre encoding foram feitos e a reunião deu-se por encerrada.

Reunião 6 (13/05/2020) - o algoritmo para importação já está parcialmente pronto, onde já possui rotina de verificação das pastas, onde estão localizadas as planilhas com os dados de importação. Também já identificando a diferença entre os inversores e os dataloggers, verifica se as variáveis da planilha encontra-se na tabela de variáveis, onde em caso positivo, importa a coluna de dados, e caso contrário, ignora. Foi então reforçado que a automação deverá sempre buscar no servidor as informações do dia anterior para a importação, mas que os dados dos dois últimos meses serão enviados manualmente para importação. Até então o projeto desenvolvido estava sendo na máquina da empresa contratada, mas que será passado um novo acesso para eles, para a migração. Também foi alinhando que o *timestamp* utilizado no BD foi o simples, dado que os dados importados mantém o padrão de uma mesma zona. Somente parte dos dados enviados anteriormente foram importados, pelo algoritmo de importação, para testes, mas que o restante seria também. Ficando também acordado o envio de algumas telas do sistema CRUD.

Reunião 7 (20/05/2020) - a empresa contratada explica que após a importação dos dados de leitura pelo algoritmo de importação, o mesmo iria mover as pastas para outro arquivo, para não gerar futuro problemas com revalidação de quais pastas já foram ou não importadas. Entretanto, um possível problema foi observado, pois existe um sistema web proprietário do LABENs, que realiza consultas nesses mesmos arquivos, o que iria afetar o funcionamento dele. Ficando combinado uma revisão do que poderia ser feito com a equipe da contratante. Algumas imagens, incluindo a de parceiros, foram enviadas para que fossem adicionadas ao sistema web (o CRUD), para ficar em destaque na tela inicial. A empresa menciona que ainda esta semana termina as telas de CRUD (com algumas exceções), ficando alinhado que em seguida se iniciariam os trabalhos da API. Foi comentado também que se tem aproximadamente 30 gigas de dados para serem importados, o que levará alguns dias para concluir. Sendo necessário subir somente alguns dias, passando por uma validação da equipe da contratante para posteriormente subir os demais períodos. O pedido de verificação dos limites dos valores de leitura foi reforçado, onde em caso que extrapole o limite, uma sinalização deverá ser feita nessa leitura. Um questionamento sobre a possibilidade de uma sinalização em relação a parada de envio de informações pelos dispositivos foi levantada, para que no futuro se retome a conversa.

Reunião 8 (27/05/2020) - na primeira pauta foi mencionado que os dados de maio foram importados para o banco de dados de forma quase automática (o script de importação que importou os dados, mas a indicação dos diretórios foi manual), e que existe a necessidade de verificar se está tudo correto, antes de subir o restante, pois a importação é demorada. Em seguida alguns comentários foram feitos em relação a estrutura do banco de dados, onde é necessário ajustar alguns atributos, sendo o diagrama mais atualizado apresentado e encaminhado a empresa. Também foram identificados um erro em relação a importação, que acontecia devido a uma entrada inesperada de dados. Um questionamento em relação a estrutura de permissões de acessos foi feita, onde apontou-se pela empresa, que não se seguiria o padrão modelado no diagrama atual, uma vez que usam estruturas prontas (*frameworks*) e o mesmo não contempla essa estrutura.

Reunião 9 (03/06/2020) - a empresa contratada iniciou a importação dos dados dos meses de março e abril, avisando que irá demorar alguns dias para concluir. Questionamento sobre uma possível atualização na tabela de grandeza foi feita, onde seriam necessários ajustes em alguns valores subidos previamente. A contratada no momento estava realizando a importação advindo de um diretório a parte ao do FTP (local de onde deveria buscar as informações). E alguns alinhamentos foram feitos, para que essa mudança já fosse realizada, como a criação de uma cópia exata, para que se pudesse trabalhar na original sem riscos de perda. Também definiu-se que o algoritmo de importação iria sempre importar tudo o que estivesse na pasta

definida, para caso houvessem dados de outros períodos, não somente ao anterior do dia, que ele também importasse, e para evitar tentativas repetidas de importação, o que fosse importado seria movido a uma outra pasta chamada "importados". A contratada definiu como próximas entregas, o sistema CRUD parcial, para então início dos trabalhos com a API (além desse ajuste no algoritmo de importação). Um outro problema com a importação foi identificado, onde somente os dados dos inversores estavam sendo importados, mas os dos dataloggers não, ficando a empresa responsável pela verificação.

Reunião 10 (10/06/2020) - no princípio da reunião foi apontado que o problema com a importação dos dados dos dataloggers continuam, e que apesar de estarem corretos, o script está somente movendo o arquivo para a pasta de importados, mas não importando para o Banco de Dados. Foi mencionado que a rotina de importação ainda não estava ativa, pois carecia de validações, mas que iriam corrigir o problema dos dataloggers e mandar subir todos os dados novamente, incluindo os que tinham problemas, para então finalizar e manter uma rotina ativa de importação. O sistema CRUD para a gerência dos dados, via sistema, foi apresentado parcialmente. E terminando as pendências, o próximo passo seria o início do desenvolvimento da API. Algumas dúvidas sobre a rotina de importação foram sanadas, quanto a questão de importações de arquivos antigos, repetidos e que gerem erros. Também foi alinhado que a rotina será configurada no servidor da contratante. Alguns resquícios de atributos que foram solicitados para não serem mais usados ainda constam nas tabelas, atributos os quais foram solicitados para serem retirados.

Reunião 11 (17/06/2020) - foi questionado o que aconteceu, para a rotina de importação não ter importado os dados até o momento, sendo explicado que o problema foi a falta de configuração dessa rotina, após ter sido usada de forma manual. Mas a mesma será ativa. Algumas hipóteses de possíveis problemas quanto a mudança de ano foram levantadas, pois a estrutura de pastas, trabalha com o ano e o mês da geração dos dados, e que da maneira em que está configurada a rotina de importação, na passagem do ano, algumas informações seriam perdidas e que a lógica atual deveria ser modificada para isso não acontecer. A ideia inicial seria para que a rotina verificasse toda as pastas que estivessem no diretório, somente ignorando a de importados, pois tudo que encontrar, deverá importar e mover para a pasta de importados. Alguns apontamentos foram feitos em relação as telas do sistema CRUD, com algumas solicitações de ajustes. E também com a criação da tela inicial contendo algumas imagens de interesse. Alguns ajustes extras quanto ao atributo flag de leitura foram feitos e a reunião foi encerrada.

Reunião 12 (24/06/2020) - um levantamento em relação a problemas com o sistema CRUD foi feito pela contratante, sendo encaminhados e apresentados a contratada para correção, para então avançar para as próximas etapas. Nesse documento explica o

erro e como deveria ser o comportamento correto. A empresa menciona que assim que terminasse o desenvolvimento das permissões de usuários e grupos, iriam trabalhar nessas correções. Tendo esses dois pontos finalizados, iniciariam os trabalhos com a API.

Reunião 13 (01/07/2020) - a importação de dados aparentemente está correta, alguns testes foram feitos por parte da contratante, e foram bem sucedidos. O sistema CRUD já teve algumas correções, dos problemas apontados na última reunião, e também as permissões foram criadas podendo ser editadas para cada um dos quatro grupos existentes. A apresentação do que foi feito foi realizada, podendo a contratada analisar tudo o que foi desenvolvido até o momento, para trabalhar na homologação. Alguns ajustes extras foram solicitados em relação ao sistema. Foi solicitado que uma consulta fosse realizada no banco de dados, dado uma saída em excel, e que todos os registros de log fossem salvos, tanto do uso do sistema, quanto do uso da API que será desenvolvida. Uma visão geral em relação ao que se espera da API foi passada, para início dos trabalhos.

Reunião 14 (08/07/2020) - Nesta reunião, foi comentado que o LABENs precisaria de 10 dias para análise das modificações realizadas no sistema CRUD, para um retorno sobre o que mudou. Explicações sobre como seria a API foram feitas, com ênfase na necessidade de a mesma só permitir acesso de leitura, para garantir que nenhuma modificação nos dados seriam feitas. Também foi comentado que a ideia de ter uma API de consulta, seria para que se pudessem gerar consultas em SQL, as quais iriam gerar chaves de permissão, para a solicitação dessas informações por terceiros, que somente necessitariam do link gerado, para receber a consulta, sem precisar acessar o sistema ou o banco de dados diretamente sendo o retorno em .CSV ou em JSON. Além do mais, comentou-se que o link gerado que requisita a consulta, deveria ter um tempo de vida, uma data onde não mais receberia as requisições, após expirar. Um limite de vezes que se pode realizar a requisição por hora, para evitar sobrecarregar o servidor em caso de erro de uso, ou até mesmo má intenção. E um registro de quem criou a consulta, o IP, o agente do usuário, assim como um nome e sua data de criação. Sendo que somente usuários com tal permissão poderiam criar e gerenciar esses links de requisições.

Reunião 15 (15/07/2020) - A empresa contratada atualiza os presentes em relação aos avanços com os trabalhos na API, onde empregaram esforços na autenticação e no armazenamento das chaves de acesso no banco de dados. Como outro assunto, foi mencionado que na semana seguinte a contratante traria uma devolutiva mais detalhada em relação ao que foi desenvolvido no sistema CRUD, mas que algumas verificações já haviam sido feitas. Para um melhor entendimento em relação ao qual genérica a API precisaria ser, foi entregue a empresa contratada uma lista com exemplos de consultas que os pesquisadores já realizam ou que poderiam realizar.

Ficando como tarefa expandir essas possíveis consultas não somente para a tabela de leitura, onde armazenam os dados dos sistemas solarimétricos e fotovoltaicos, mas para todas as demais tabelas. Algumas solicitações foram feitas, como a de ter a possibilidade de validar o retorno que a consulta SQL está trazendo, antes de salvar e gerar o link com essa consulta para uso via a API. Algumas dúvidas foram sanadas, como a da possibilidade de um usuário ter mais de uma consulta, ou seja, gerar mais de uma chave de acesso atrelada a uma consulta. No final da reunião, levantou-se a hipótese de se expandir o projeto, incluindo funcionalidades que envolvessem o envio de comandos para equipamentos instalados nos campos.

Reunião 16 (22/07/2020) - verificações em relação ao estado atual do sistema CRUD foi realizado e enviado via .PDF para a empresa, para as devidas correções. Em seguida, algumas atualizações foram feitas em relação aos trabalhos desenvolvidos com a API, onde a empresa contratada inseriu uma nova tela para a gerência das chaves geradas. E em meio a análise, menciona-se da necessidade de se ter acesso aos .CSVs com maior facilidade.

Reunião 17 (29/07/2020) - de primeira pauta, pediu-se atualização em relação as correções apontadas no última reunião, mas que a empresa contratada ainda não verificou, pois direcionou os trabalhos no que se relaciona ao desenvolvimento da API. Solicitou-se que o tempo máximo de espera para o retorno ou falha de uma consulta fosse aumentado, pois algumas consultas mais demoradas, chegavam ao seu tempo máximo e não funcionavam. Também foi apresentado uma nova maneira de se usar as chaves geradas usando a API, onde as consultas poderiam ser realizadas pela própria planilha do Google, o que facilitaria o trabalho dos pesquisadores. A contratada também explica de forma conceitual algumas atualizações que realizaram mais ainda não publicaram. Sendo também mencionada a necessidade de ter o funcionamento dessa parte o quanto antes, para que os pesquisadores já pudessem iniciar o uso. E definiu-se o tamanho dos caracteres das chaves geradas para 128 caracteres.

Reunião 18 (05/08/2020) - para o desenvolvimento de um novo módulo de envio de comando para os inversores, foi feita uma rodada de explicações, mostrando como é realizado atualmente, para então uma integração dos envios dos comandos, com interface via o sistema que está em desenvolvimento. Bem como os próximos passos de desenvolvimento dessa tela de comandos, onde em uma primeira versão terá interação manual e em um segundo momento terá um protocolo de segundo nível com confirmação via e-mail. Alguns ajustes na tela de API foram solicitados, como a possibilidade de poder editar uma consulta já criada.

Reunião 19 (12/08/2020) - a principal pauta foi tratar do desenvolvimento do envio de comando para os inversores. Onde já subiu uma tela de comando no sistema WEB, a qual permite gerar as chaves de comando. Uma apresentação do atual uso e uma revisão das tabelas e de seus atributos que compõem essa parte do sistema

foram realizadas. Algumas considerações foram feitas ficando pendente por parte do LABENs o preenchimento das tabelas para os inversores e a designação de um e-mail para a realização do envio dos alertas. Algumas considerações em relação as leituras foram feitas, onde o horário dos registros que estavam sendo importados estavam incorretos, sendo necessário um tratamento antes de subir, bem como uma correção de todo o período que já passou.

Reunião 20 (19/08/2020) - algumas atualizações foram feitas, tal como o desenvolvimento da parte de comunicação com o e-mail (envio de solicitação para mudança nos inversores - um link que leva até a tabela de comandos). Também sendo possível passar comandos via parâmetro aos inversores. Alguns ajustes foram solicitados na interface. Como alguns testes de comunicação com os dispositivos ainda não haviam sido feitos, uma reunião específica para tal foi marcada.

Reunião 21 (26/08/2020) - uma primeira versão funcional do envio de comandos aos inversores foi apresentada, mas com algumas pendências, como a questão do retorno deles. Onde ficou-se acordado que o papel da verificação se o valor passado está dentro do mínimo ou máximo seria realizada pelo próprio ST-One, não sendo necessária uma máscara do sistema para tal (o mesmo verifica e ignora se for o caso). Para dar continuidade com a criação das consultas, uma proposta de estrutura foi apresentada a contratada. Também foi repassado que seria necessário mudar o tempo no banco de dados em relação ao UTC-3, para que os horários mostrados passem a ser compatíveis com a realidade.

Reunião 22 (02/09/2020) - problemas com o envio de comandos para os inversores foram apresentados, para que a empresa desenvolvedora do sistema verificasse e corrigi-se. Também foram encontrados problemas com a importação dos dados de leituras dos dispositivos, onde valores que estavam com erros estavam sendo ignorados e não estavam sendo importados. Uma tentativa de localização e correção do erro foi feita, mas sem sucesso, ficando acordado de uma verificação após a reunião por parte da empresa contratada. Também ficou combinado que poderia haver a necessidade de uma reunião extra na semana com pauta exclusiva relacionada ao desenvolvimento da API.

Reunião 23 (09/09/2020) - algumas funcionalidades solicitadas anteriormente foram feitas e já validadas em relação a edição das chaves de acesso/permissão das consultas feitas via API. Também alguns testes com envios de comandos aos inversores foram feitos com sucesso. Algumas funcionalidades foram discutidas, como a do envio da confirmação da mudança nos dispositivos ser para o administrador do sistema, não necessariamente para quem solicitou. Mudanças na interface também foram solicitadas, e com esses ajustes já se pode dar por finalizada essa etapa. Um debate sobre otimização das consultas foi iniciado, ficando pendente de continuidade em um momento posterior.

Reunião 24 (14/09/2020) - dado o curto tempo para o término do projeto, o LABENs optou por utilizar as horas remanescentes com a contratada para trabalhos de homologação em relação a todas as entregas e treinamento a nível de código do que foi desenvolvido até então. Foi trazida a ideia de se usar *dockers*), para não ser necessário ter que instalar todas as dependências dos projetos. Um repositório no GitHub será criado para a hospedagem de todos os acessos ao sistema. Para os treinamentos, idealizou-se que serão dois encontros por semana, de duas horas cada. Sendo importante que todos os que forem acompanhar, realizem um curso básico de Laravel. Como última pauta, levantou-se a necessidade de se realizar uma rotina de *backup*, a qual o próprio PHP poderia realizar.

Reunião 25 (16/09/2020) - um cronograma de repasse técnico (treinamento) foi elaborado, prevendo sete encontro de duas horas cada. Uma proposta de advinda pelo LABENs, foi de inserir a correção de cerca de onze erros identificados, nesse repasse técnico, para que se pudesse entender na prática a lógica para a manutenção do sistema. O que poderia ser feito para somente alguns itens, dado o tempo curto. Os erros encontrados foram apresentados e a reunião encerrou.

Reunião 26 (14/10/2020) - como pauta única debateu-se sobre o funcionamento do envio de comando para os inversores pelo sistema desenvolvido pela contratada. O mesmo não está cumprindo com os casos de uso acordados no início do desenvolvimento, por isso, um diagrama complementar foi realizado e apresentado. Alguns pontos foram alinhados, como a correção dos retornos dos inversores para então a entrega completa dos requisitos solicitados. Ambas as partes comprometeram em realizar os devidos ajustes e a reunião finalizou.

Neste apêndice estão relacionados os Diagramas de Caso de Uso utilizados no projeto.

APÊNDICE B – Diagramas de Casos de Uso

- C1 - Enviar dados para servidor FTP:** Inversores e Dataloggers enviam os dados coletados para o servidor FTP.
- C2 - Enviar dados para o BD:** Um script de importação busca no servidor FTP os dados coletados e manda para o Banco de Dados.
- C3 - Separar dados importados:** O script de importação separa os dados importados, do FTP para o banco de dados, dos que ainda não foram importados.
- C4 - Cadastrar manutenção:** Um administrador, gestor global ou gestor local cadastra uma ou mais manutenções.
- C5 - Cadastrar contato:** Um administrador, gestor global ou gestor local cadastra um ou mais contatos.
- C6 - Cadastrar dispositivo:** Um usuário administrador ou gestor global cadastra um ou mais dispositivos.
- C7 - Cadastrar dispositivos de inventário:** Um usuário administrador ou gestor global cadastra um ou mais dispositivos de inventário.
- C8 - Cadastrar ocorrência:** Um usuário administrador ou gestor global cadastra uma ou mais ocorrências de manutenções.
- C9 - Cadastrar natureza:** Um usuário administrador ou gestor global cadastra uma ou mais naturezas de manutenções.
- C10 - Cadastrar função:** Um usuário administrador ou gestor global cadastra uma ou mais funções de transferência de um dispositivo.
- C11 - Cadastrar planta:** Um usuário administrador ou gestor global cadastra uma ou mais plantas.
- C12 - Cadastrar ponto de acesso:** Um usuário administrador ou gestor global cadastra um ou mais pontos de acesso.
- C13 - Cadastrar usuário:** Um usuário administrador ou gestor global cadastra um ou mais usuários.
- C14 - Cadastrar variável:** Um usuário administrador ou gestor global cadastra uma ou mais variáveis de dispositivos.
- C15 - Cadastrar grandeza:** Um usuário administrador ou gestor global cadastra uma ou mais grandezas que dispositivos medem.
- C16 - Editar grupos:** Administrador altera as permissões dos grupos Gestor Global, Gestor Local ou Bolsista.
- C17 - Editar permissões:** Um administrador altera as permissões de usuários (Gestor Global, Gestor Local ou Bolsista).
- C18 - Realizar login:** Os usuários logam, através da tela de login, para ter acesso ao sistema.

- C19 - Criar consultas e chaves de consulta de API:** Os administradores e gestores podem criar consultas ao banco de dados, que podem ser realizadas através de uma API, com chaves de acesso, ou seja, somente quem detém as chaves pode consultar.
- C20 - Restringir o uso das chaves de consulta:** Os administradores e gestores que criarem a chave de consulta, devem poder limitar o uso, configurando o máximo de requisições que podem ser feitas por hora, o formato do retorno e prazo de validade. Além disso pode nomear a chave de acesso.
- C21 - Editar chaves de acesso:** Os usuários administradores e gestores podem editar chaves de acesso já existentes, podendo modificar o número de requisições máximas, o formato, a validade, o nome e a consulta SQL.
- C22 - Linguagens e ferramentas:** Será necessário fazer uso de linguagens/ferramentas gratuitas, pois não haverá a possibilidade de pagar por elas após o término do projeto.
- C23 - A infraestrutura deverá ser própria:** Não poderão ser usados serviços de hospedagem pagos, sendo necessário equipamento próprio para tal.

APÊNDICE C – Análise de Requisitos

Neste apêndice estão listados os Requisitos Funcionais e Não Funcionais obtidos durante o desenvolvimento do projeto.

C.1 Requisitos Funcionais

Após o levantamento dos casos de uso, foi possível elaborar os seguintes requisitos funcionais do sistema:

- RF01** - Para usar o sistema web, o usuário precisa fazer login.
- RF02** - O sistema deverá permitir descadastrar usuários.
- RF03** - O sistema deverá armazenar logs de acesso.
- RF04** - O sistema deverá armazenar logs de consultas (importação).
- RF05** - O sistema deverá permitir cadastrar usuários.
- RF06** - O sistema deverá permitir o cadastro de um novo equipamento.
- RF07** - O sistema deverá permitir a retirada de um novo equipamento.
- RF08** - O sistema deverá permitir editar dados de um equipamento, como dados de manutenção.
- RF09** - O sistema deverá permitir a edição de um perfil de usuário.
- RF10** - O sistema deverá permitir cadastrar usuários com diferentes níveis de permissões.
- RF11** - O sistema deverá se comunicar com o Banco de Dados através da API de acesso.
- RF12** - O sistema deverá notificar se parâmetros errados/inválidos foram inseridos.
- RF13** - O sistema deverá realizar registro de log dos usuários e do sistema.
- RF14** - O sistema deverá salvar perfis de consultas com filtragem por usuários.
- RF15** - O sistema deverá solicitar usuário e senha para o acesso a ele.
- RF16** - O sistema deverá pedir como login o CPF do usuário.
- RF17** - O sistema deverá ter tela inicial voltada a apresentação dele.
- RF18** - O sistema deverá ter diferentes permissões de usuários.
- RF19** - O sistema deverá fazer uso de software livre.
- RF20** - O sistema deverá mostrar em todas as telas o qual usuário está logado, e horário de login.
- RF21** - O sistema deverá permitir atribuir mais ou menos permissões aos usuários.
- RF22** - O sistema deverá permitir personalizar as permissões dos grupos de usuários.
- RF23** - O sistema deverá deslogar o usuário após um certo tempo de inatividade ou que estiver logado.
- RF24** - O sistema deverá permitir criar e editar funções de transferência dos dispositivos.
- RF25** - O sistema deverá permitir criar e editar grandezas dos dispositivos.
- RF26** - O sistema deverá permitir criar e editar variáveis dos dispositivos.

- RF27** - O sistema deverá permitir criar e editar itens de inventário.
- RF28** - O sistema deverá permitir criar manutenções de dispositivos.
- RF29** - O sistema deverá permitir atribuir uma natureza para a manutenção realizada.
- RF30** - O sistema deverá permitir criar e editar tipos de natureza de uma manutenção.
- RF31** - O sistema deverá permitir criar e editar tipos de ocorrências de uma manutenção.
- RF32** - O sistema deverá permitir criar e editar plantas.
- RF33** - O sistema deverá permitir criar e editar pontos de acesso.
- RF34** - O sistema deverá permitir criar e editar contatos.
- RF35** - O sistema deverá permitir qualquer usuário poder trocar sua própria senha.
- RF36** - O sistema deverá permitir baixar do próprio sistema as tabelas que apresentar referente aos dispositivos, suas medições e configurações.

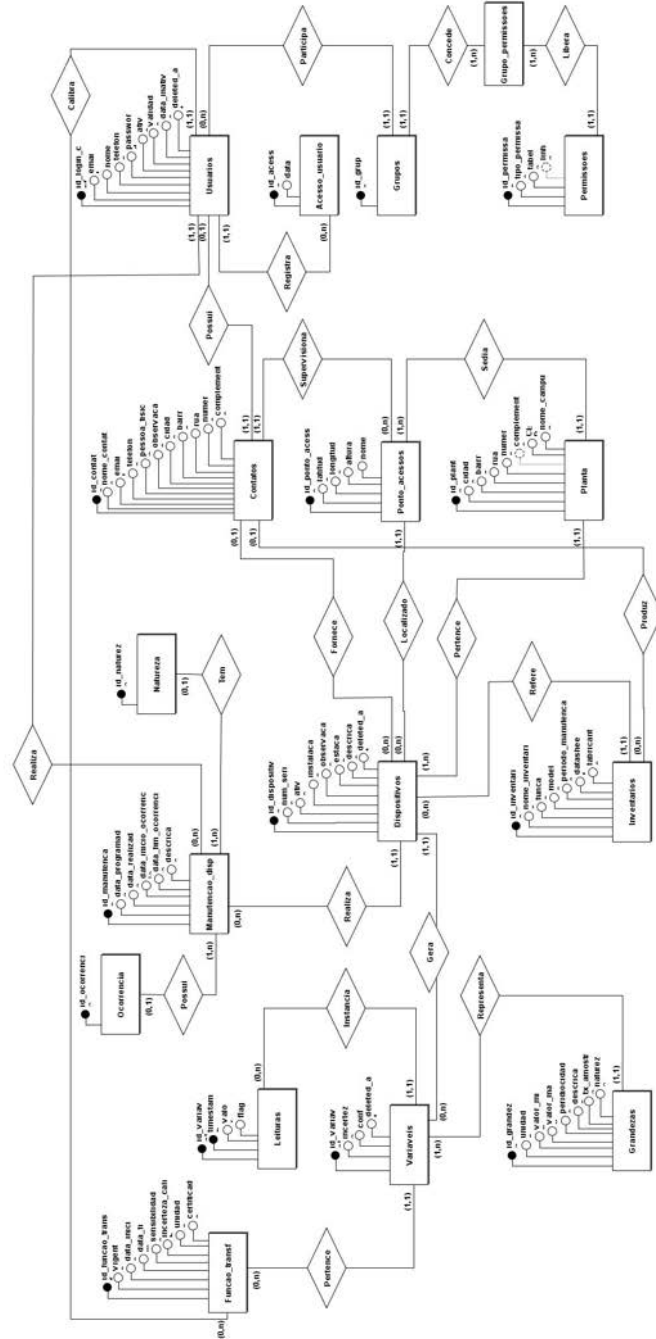
C.2 Requisitos Não Funcionais

Após o levantamento dos casos de uso, foi possível elaborar os seguintes requisitos não funcionais do sistema:

- RNF01** - No sistema deverá conter as logos do LABENS e de seus parceiros.
- RNF02** - Deverá ser um sistema web.
- RNF03** - As linguagens e ferramentas usadas deverão ser de uso gratuito.
- RNF04** - Na tela inicial do sistema, deverá ter a logo do laboratório e dos parceiros.
- RNF05** - Na tela inicial do sistema, deverá ter um texto explicando brevemente sobre o projeto.

APÊNDICE D – Diagrama Entidade-Relacionamento

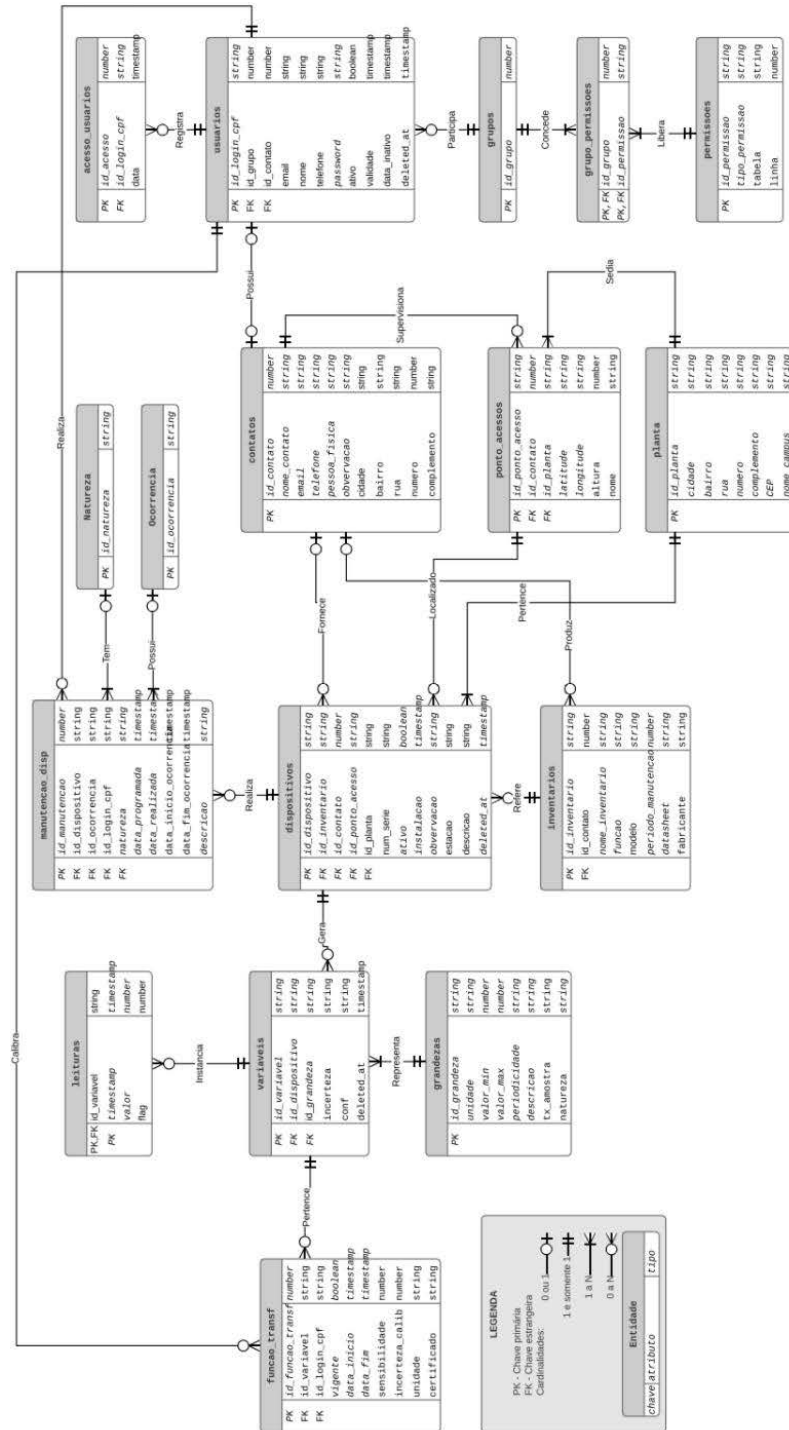
Figura 13 – Diagrama Entidade-Relacionamento



Fonte: LABENS 2020

APÊNDICE E – Modelo Lógico do Banco de Dados

Figura 14 – Diagrama Lógico



Fonte: LABENS 2020