



FACULDADE PROJEÇÃO
Engenharia de Software

Nádio Dib Fernandes Pontes

Análise e correlação de dados de boletins do ONS para predição de incidentes de sobrecarga energética em UHEs de acordo com as resoluções normativas da ANEEL através de frameworks de Data Mining e Web Scraping

Brasília/DF, 01 de Abril de 2024



FACULDADE PROJEÇÃO

Engenharia de Software

Análise e correlação de dados de boletins do ONS para predição de incidentes de sobrecarga energética em UHEs de acordo com as resoluções normativas da ANEEL através de frameworks de Data Mining e Web Scraping

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Software da Faculdade Projeção, como requisito complementar e obrigatório à obtenção do título de Bacharel em Engenharia de Software.

Brasília, 01 de Abril de 2024.

Banca Examinadora:

Professor(a): Fernando Feliu Gonzalez

Orientador(a)

Professor(a):

Professor(a):

Professor Rosa Maria Diekn de Queiroz, Mestre
Coordenadora do curso de Sistemas de Informação

AGRADECIMENTOS

Primeiramente, à minha família, Josiane, Alexandre e Luana, no qual fizeram parte dessa longa jornada durante minha graduação. Também ao meu inspirador, prof. Kauê, no qual me motivou e inspirou a dar início ao curso de Engenharia de Software e continuar resiliente, independente das adversidades nos quais enfrentaria. Meus agradecimentos à minha turma do curso de Engenharia de Software, colegas dos cursos de Sistemas de Informação e Análise e Desenvolvimento de Sistemas, além de meus colegas mais próximos de turma William, Ricardo, Isaac e Yuri.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS.....	7
LISTA DE TABELAS.....	8
RESUMO.....	9
ABSTRACT	10
CAPÍTULO I.....	11
VISÃO INICIAL	11
1. Introdução	11
2. Contextualização.....	11
3. Problemática	12
4. Solução Proposta.....	12
5. Objetivo Geral	13
5.1. Objetivos Específicos	13
6. Estrutura do Projeto	13
CAPÍTULO II.....	15
METODOLOGIA APLICADA.....	15
1. Metodologia e Modelo de Desenvolvimento	15
2. Técnicas de Levantamento de Requisitos	15
2.1 Entrevista.....	15
2.2 Questionário	16
2.3	16
3. Principais tecnologias utilizadas	17
4. Análise de Sistemas Existentes	22
5. Cronogramas	Erro! Indicador não definido.
CAPÍTULO III.....	23
VISÃO DO SISTEMA.....	23
1. Riscos do Projeto	23
2. Restrições do Projeto	23
3. Regras do Negócio	24
4. Requisitos Funcionais	24
5. Requisitos Não Funcionais.....	25
6. Mensagens do Sistema.....	25
7. Lista de Casos de Uso	25
8. Lista de Atores	26
9. Diagrama Geral de Caso de Uso.....	26
10. Diagrama de Classe	26
11. Diagrama de Processo	27
CAPÍTULO IV	28
ESPECIFICAÇÃO DE CASO DE USO	28
UC001 – GERENCIAR PARECER TÉCNICO	28
CAPÍTULO V	31

MODELAGEM DE DADOS	31
1. Diagrama de Entidade Relacional	31
2. Diagrama de Entidade Relacional	31
3.1 Dicionário de Dados	32
3.1.1 Tabela [Nome da Tabela].....	32
CAPÍTULO VI	33
PROTOTIPAÇÃO DO SISTEMA.....	33
1 Manter processos <Identificação de Tela>	33
1.1 Campos da Tela	34
1.2 Comandos da Tela.....	34
CONCLUSÃO	35
REFERÊNCIAS	36
GLOSSÁRIO	Erro! Indicador não definido.
ANEXO A<descrição do anexo>	Erro! Indicador não definido.
APÊNDICE <descrição do apêndice>	Erro! Indicador não definido.

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
SIN	Sistema Integrado Nacional
ONS	Operador Nacional do Sistema Elétrico
UHE	Unidade Hidrelétrica
ER	Entidade Relacional
CAG	Controle Automático de Geração
W3C	World Wide Web Consortium
IDE	Integrated Development Environment
ASGI	Asynchronous Server Gateway Interface
WSGI	Web Server Gateway Interface
SisBIN	Sistema de Boletins Integrado Nacional

LISTA DE FIGURAS

Figura 1: arquitetura simplificada do Selenium WebDriver e suas iterações com os principais navegadores da internet.	18
Figura 2: breve abordagem das principais vantagens de escolher a linguagem Python.	19
Figura 3: principais características da MariaDB.	20
Figura 4: comparação do FastAPI com os principais frameworks de APIs do mercado, onde enfatiza a melhor escolha sendo a intermediária.	21
Figura 5: Uvicorn promove um ganho considerável na performance do servidor ASGI.	22
Figura 6: Diagrama de Caso de Uso do SisBIN para representação das operações diretas à API e afins.	26
Figura 7: Diagrama de Classe.	26
Figura 8: Diagrama de Sequência.	27

LISTA DE TABELAS

Tabela 1: Riscos do Projeto	23
Tabela 2: Restrições do Projeto	23
Tabela 3: Regras de Negócio.	24
Tabela 4: Requisitos Funcionais.	24
Tabela 5: Requisitos Não Funcionais.....	25
Tabela 6: Mensagens do Sistema.....	25
Tabela 7: Lista de Caso de Uso.....	25
Tabela 8: Lista de Atores.	26

RESUMO

Este Trabalho de Conclusão de Curso, tem como objetivo explicitar a necessidade da coleta de dados dos boletins gerados pelo Operador Nacional do Sistema Elétrico (ONS), para fins de predição de incidentes de sobrecarga energética em Unidades Hidrelétricas (UHEs) de acordo com as regulações normativas da Agência Nacional de Energia Elétrica (ANEEL) através de tecnologias de *Data Mining* para mineração de dados extraídos por robôs programáveis, nos quais utilizam *frameworks* de *Web Scraping*. Os possíveis incidentes nas UHEs serão detectáveis por análises correlativas e classificatórias, além de serem norteados pelas normas da ANEEL quanto a sua detecção e prevenção.

Palavras-Chave: ONS; ANEEL; Data Mining; Web Scraping; Web Crawling; prevenção contra incidentes em UHEs; análise correlativa de dados em boletins do ONS; análise classificatória e preditiva de dados em boletins do ONS.

ABSTRACT

The purpose of this Completion of Course Work is, to explicit how important is to collect data from National Electrical Operator System (NOS), to provide an incident foresight to Hydro Power Plants (HPPs) against overwhelm energy overload according standard regulations from National Electrical Energy Agency (NEEA) using technologies and frameworks of Data Mining and Web Scraping. Therefore, possible incidents on HPPs could be found among data correlation, data classification and prediction analysis when applying NEEA standard regulations to detect and mitigate them.

Keywords: NOS; NEEA; Data Mining; Web Scraping; Web Crawling; incident foresight on HPPs; data correlation analysis on NOS bulletins; data classification and prediction analysis on NOS bulletins.

CAPÍTULO I

VISÃO INICIAL

1. Introdução

O Operador Nacional do Sistema Elétrico (ONS) é o órgão responsável pela coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no Sistema Interligado Nacional (SIN) e pelo planejamento da operação dos sistemas isolados do país, sob a fiscalização e regulação da Agência de Energia Elétrica (ANEEL) [1].

Além de ser um órgão que desempenha o papel de monitoração do sistema elétrico brasileiro, também disponibiliza um acervo de boletins, nos quais podem ser consultados via endereço eletrônico, em <http://sdro.ons.org.br/SDRO/DIARIO/index.htm>. Tais boletins são publicados diariamente, desde 03 de agosto de 2013, onde o primeiro registro foi incluído.

Um dos principais propósitos desses boletins é de prover informações quanto ao balanço de energia, produção de energia, carga e demanda, integração de equipamentos e ocorrências no SIN e hidrologia.

Todavia, quando interligadas essas informações podem ser relacionadas para outros fins, como por exemplo, o próprio tema deste trabalho, no qual tem como objetivo correlacionar esses dados que estão isolados em diversos boletins diários e assim possibilitar um modelo de predição de dados unificado para prevenção de possíveis incidentes em UHEs registradas no Brasil.

2. Contextualização

Atualmente, o SIN, que representa o sistema de energia elétrico nacional, nos quais fazem parte os subsistemas Sudeste, Centro-Oeste, Sul, Nordeste e Norte, consta com demandas de carga energética diária para atender os requisitos de abastecimento elétrico nessas regiões.

Entretanto, em certos períodos do ano, o nosso país, Brasil, ocasionalmente recebe tantos eventos nacionais, regionais ou mesmo internacionais que, diretamente, necessitam do fornecimento elétrico para

atender a demanda das residências, indústrias, comércio e outros envolvidos no contexto.

Sendo assim, apesar do SIN ser, em sua grande maioria, automatizado quanto a entrega de cargas nas regiões para atendimento da demanda energética, os procedimentos de monitoração ainda são manuais e contam com o suporte do ONS e normas regulatórias da ANEEL para que sobrecargas desnecessárias não sejam acionadas incorretamente para essas regiões, segundo o Planejamento Anual da Operação Energética para margens de 5 anos [1].

3. Problemática

O principal problema noticiado é a necessidade da unificação destas informações, referente aos boletins diários fornecidos pelo ONS, para análises preditivas de longos períodos, como por exemplo previsões anuais, ao invés de serem realizadas através de observações diárias.

Através deste panorama, será possível contornar alguns tópicos e possibilitar outros benefícios, utilizando tecnologias e mecanismos presentes na atualidade para fornecer ainda mais vantagens para este contexto, por exemplo como a criação de modelos de aprendizagem de máquina nos quais performem tarefas de predição contra eventos e incidentes iminentes em UHEs, baseados nos dados informados pelos boletins diários do ONS.

4. Solução Proposta

Devido a necessidade de coletar dados dos boletins gerados pelo ONS e elaborar artefatos como a criação de relatórios específicos com informações contra possíveis incidentes em UHEs, como por exemplo quanto o levantamento de informações geradas por estes boletins para margens de período, mais amplo, como mensal, semestral ou anual, este problema afeta diretamente os órgãos ONS e ANEEL.

Contudo, o desenvolvimento deste sistema tem como benefício providenciar a criação de uma API pública através de arquitetura RESTful para consulta destes dados relatados anteriormente, nos quais serão obtidos por robôs de extração de dados, que utilizam frameworks para este fim, de modo a

atender estas expectativas do projeto. Além que será providenciado a geração de relatório com margens de previsão de possíveis incidentes no SIN via modelagens de correlação de dados através desta mesma API em formato de dados via JSON.

5. Objetivo Geral

O objetivo deste documento é contribuir para o aprimoramento e aperfeiçoamento da qualidade das informações disponibilizadas além de coletar, analisar e definir necessidades e características gerais do projeto referente à análise preditiva de incidentes em UHEs via metodologias de correlação de dados por frameworks de mineração de dados e robôs para extração de dados.

5.1. Objetivos Específicos

Dado o contexto da análise preditiva de incidentes nos quais envolvem as UHEs, este trabalho tem como visão além de atender as expectativas citadas no objetivo geral, também os seguintes objetivos específicos abaixo:

- Beneficiar a facilitação na coleta de dados para a geração de modelos de aprendizagem de máquina através de demais frameworks de *Machine Learning*; e
- Beneficiar o uso de tecnologias que utilizam frameworks de mineração de dados para criação de modelos preditivos no contexto de predição de incidentes mais complexos de identificar, bem como corroborar no incremento de análises estatísticas mais elaboradas que poderão ser relacionadas com os dados numéricos e classificatórios dos boletins diários do ONS.

6. Estrutura do Projeto

Este projeto de conclusão de curso foi dividido em capítulos nos quais auxiliarão na leitura e compreensão de seus tópicos mais específicos, onde se tentou apresentar de forma mais aprofundada os itens referentes ao tema principal envolvendo este sistema e suas ramificações.

- Capítulo I: é referente à visão inicial e demais relações do projeto em questão, no qual especificam-se tópicos relacionados ao contexto do

projeto, motivações sobre os pontos de melhoria, análises gerais e específicas e a atual estrutura deste documento;

- Capítulo II: é referente à metodologia aplicada no desenvolvimento deste trabalho, no qual enfatiza alguns padrões e tópicos nos quais proporcionaram a evolução e progresso de padrões técnicos aplicados à este projeto, bem como enfoque nas tecnologias, frameworks e demais informações de análises e prazos estipulados que foram aplicados neste trabalho;
- Capítulo III: é referente à visão de todo o sistema desenvolvido, aplicando o conhecimento técnico e profissional no qual foi compartilhado durante o curso de bacharelado de Engenharia de Software de acordo com as melhores práticas e metodologias proporcionadas na formação pela instituição, neste capítulo também é possível obter detalhes sobre o sistema em questão referente a este trabalho;
- Capítulo IV: é referente as especificações técnicas de Casos de Usos proporcionados para o devido andamento de passos essenciais na criação de funcionalidades deste sistema em questão;
- Capítulo V: é referente aos padrões de modelagem de dados utilizados na concepção de bases de dados preferível para persistência das informações geradas pelo sistema, de modo a promover melhor compreensão de sua estruturação e significado;
- Capítulo VI: é referente aos processos de prototipação implementados e testados nos quais comprometeram o sistema em sua evolução nos seus primórdios conceituais.
- Enfim temos os demais componentes deste trabalho como: Conclusão, Referências, Glossário, Anexos e Apêndices.

CAPÍTULO II

METODOLOGIA APLICADA

1. Metodologia e Modelo de Desenvolvimento

A metodologia empregada na implantação deste projeto, para Análise e correlação de dados de boletins do ONS para predição de incidentes de sobrecarga energética em UHEs, é o modelo de desenvolvimento em cascata ou *waterfall*. Esta metodologia tem como principal a ênfase de conceitos e fundamentos sólidos de todo este sistema na qual será desenvolvido.

Segundo Hickey e Davis ^[2], afirmam que é imprescindível que, para o desenvolvimento de software, o conhecimento para a evolução do projeto é fundamental tanto para o embasamento dos principais requerimentos levantados durante a técnicas de elicitação quanto para a construção de documentos de análise, prototipação e demais métodos para obtenção de requisitos utilizados durante o refinamento dos principais requisitos do sistema, nos quais constarão neste Projeto de Conclusão de Curso.

O modelo de desenvolvimento *waterfall* tem como principais vantagens ^[3] a criação dos requerimentos nas fases iniciais, onde são cruciais para a criação do escopo, demais tarefas e design deste projeto; melhoria na utilização de recursos; melhor compreensão durante o desenvolvimento deste projeto; e constante feedback durante todo os processos, quanto a seu andamento.

2. Técnicas de Levantamento de Requisitos

2.1 Brainstorming

Esta técnica é utilizada para realização de discussões informais com o intuito de expressar o livre arbítrio de novas ideias e concepções para o desenvolvimento do sistema.

O Brainstorming ^[4] funciona em duas fases, sendo estas: a fase de geração, na qual as ideias são coletadas sem julgamentos ou critérios; e a fase de evolução, onde através das ideias coletadas também serão discutidas.

O principal intuito para a utilização desta técnica neste projeto vem através de aplicar conceitos de inovação e necessidade de unificar conceitos já abstraídos o mercado para o contexto desta problemática.

O objetivo principal da aplicação desta técnica no sistema foi devido a mescla de experiências prévias tanto com o desenvolvimento de robôs de extração de dados de conteúdo de cunho científico como IEEE, no qual se objetivava a geração de relatórios estatísticos para criação de publicações e apresentações acadêmicas ao departamento de Engenharia Elétrica da Universidade de Brasília, no ano de 2020, no período de agosto até dezembro.

Além de experiências a respeito da criação de robôs nos quais utilizavam frameworks que promovia a extração de informações específicas, também obtive outras experiências no contexto da Engenharia Elétrica nos anos de 2018 a 2019, onde meu antigo professor que ministrava as disciplinas de gerenciamento elétrico, citou a possibilidade de avaliar os boletins do ONS de forma automatizada ao invés de manual.

Sendo assim, esses foram os principais quesitos que possibilitaram a aplicação desta técnica para o avanço tanto de uma Prova de Conceito quanto implementações através de código-fonte e, conseqüentemente, progredir no desenvolvimento deste sistema objetivado neste documento.

2.2 Observação Direta

Esta técnica etnográfica é utilizada através da observação e envolvimento direto do observador na sociedade e geralmente é utilizada em conjunto com outras técnicas a fim de coletar requisitos implícitos que a técnica de entrevista não consegue revelar.

A Observação Direta ^[5] é feita através de dois processos: ativo, no qual a abordagem de questionamento é feita ao usuário durante o processo de observação; e passiva, onde o observador não interage com o usuário durante a observação de sua rotina.

O principal intuito para a utilização desta técnica neste projeto é devido a não existência de projetos ou iniciativas similares em outros projetos acadêmicos, além de prover insumo para o desenvolvimento de outros mecanismos de proteção de incidentes em UHEs.

A técnica de Observação Direta está diretamente interligada a técnica evidenciada anteriormente, Brainstorming, pois mesmo com os protocolos de segurança e demais funções regulatórias de órgãos nacionais como a ANEEL, ainda existe as possibilidades de falha humana no processo de monitoração, que é feito em algumas etapas pelo próprio ONS.

Logo, devido a esta objeção, análises e relatos de profissionais que atuam neste contexto sobre a influência de possíveis incidentes em UHEs, foi evidente a necessidade de um providenciar mecanismo que fornecesse uma melhoria neste processo de segurança.

Por mais calculistas e precavidos contra esses incidentes, ainda existe a possibilidade iminente de acidentes ocorrê-los devido a operação humana em etapas do processo de geração energética dentro da matriz do SIN através das UHEs.

3. Principais tecnologias utilizadas

O sistema será testado em um ambiente computacional com as seguintes características:

- Processador Intel Core i5-4570S CPU @ 2.90GHz;
- Memória virtual (RAM) de 8GB DDR3;
- Disco rígido WDC WD5000AAKX-08ERMA0 HDD;
- Conexão via fibra óptica de 100MB/s para driver de rede Realtek PCIe GbE Family Controller;
- Placa gráfica GPU Intell HD Graphics 4600; e
- Sistemas operacionais: Windows 10 Pro 21H1 e Linux Ubuntu 20.04 LTS.

3.1 Selenium WebDriver

O Selenium WebDriver ^[6] opera o navegador ou *browser* nativamente, bem como o usuário, seja localmente ou ambiente virtual remoto através de um servidor Selenium, esse framework permite a automação de processos dentro do navegador.

Selenium WebDriver Architecture

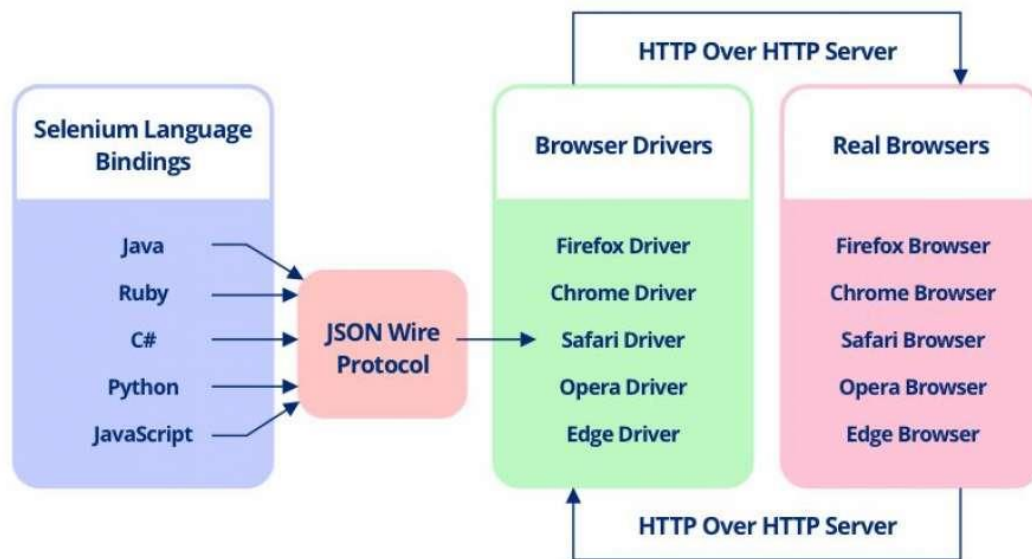


Figura 1: arquitetura simplificada do Selenium WebDriver e suas interações com os principais navegadores da internet.

Este framework refere-se tanto a interligação com a linguagem de programação quanto com a implementação via controle de código partes individuais do *browser*. Popularmente é referido apenas com o termo WebDriver.

O Selenium WebDriver é uma recomendação da W3C [7]:

- O WebDriver foi designado para atuar como uma interface simples e concisa para programação de processos automatizados no navegador;
- O WebDriver fornece uma API compactada orientada à objeto; e
- Permite o controle do navegador de forma eficiente.

3.2 Python 3

O Python [8] é uma linguagem de programação de fácil aprendizagem e ao mesmo tempo poderosa. Ela possui um alto nível de controle sobre as principais estruturas de dados e de forma simplificada atua de modo eficiente na abordagem da programação orientada à objetos. Sendo assim, de forma elegante e dinâmica na sua escrita, junto com seu interpretador de código integrado nativo, promove uma linguagem de programação ideal para a criação de scripts em aplicações que demandam rápido desenvolvimento em diversas áreas e plataformas.

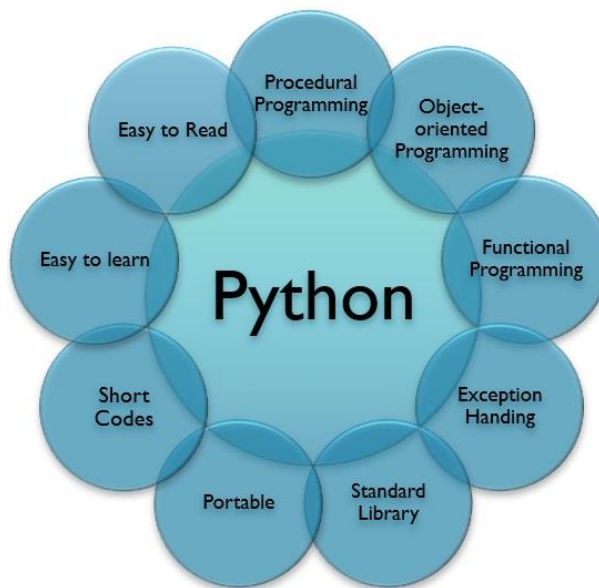


Figura 2: breve abordagem das principais vantagens de escolher a linguagem Python.

O interpretador do Python e sua diversa gama de bibliotecas padronizadas são facilmente disponíveis para obtenção tanto em código fonte original quanto de seus binários para todas as principais plataformas no próprio site da linguagem em <https://www.python.org/>, além de promover o seu uso para diversos propósitos.

Além disso, o interpretador Python permite a extensão de funções e tipos de dados implementados em C ou C++ ^{[9][10]} (ou qualquer outra linguagem de programação oriunda do C nativo). Sendo assim, o Python é comumente escolhido para ser um extensor de linguagens para customização de aplicações.

3.3 MariaDB Server

A MariaDB Server ^[11] é um dos mais populares servidores para banco de dados do mundo. Originalmente foi desenvolvido pelos mesmos criadores do MySQL e se mantém *open source*.

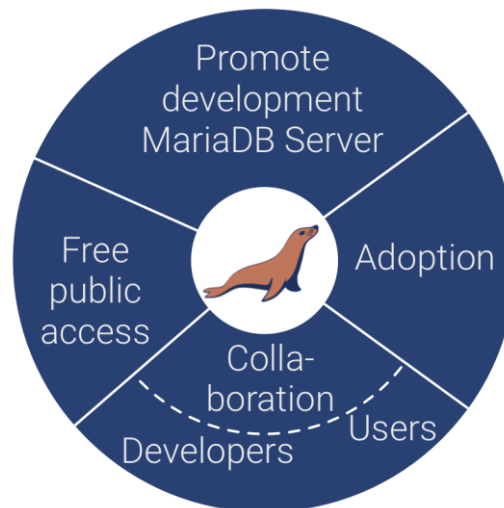


Figura 3: principais características da MariaDB.

MariaDB converte informações e dados de diferentes estruturas em uma grande variedade de aplicações, desde bancos a websites. Foi designada para atuar de forma otimizada e melhorada de modo a substituir o MySQL, onde atuar de maneira rápida, escalável e robusta e preserva um rico ecossistema para persistência de dados dentro das principais *engines*, plugins e demais ferramentas versáteis de banco de dados do mercado e diversas aplicações para diferentes casos de usos.

Além disso, foi desenvolvida como um projeto de software de código aberto e como um banco de dados relacional onde fornece suporte uma interface de interpretação do SQL para acesso aos dados. As últimas versões do MariaDB incluem suporte ao GIS e funcionalidades JSON.

3.4 FastAPI

O FastAPI ^[12] é um moderno framework web de alta performance para construção de APIs utilizando o Python 3.6+ baseado nos principais padrões do Python e recomendações.



Figura 4: comparação do FastAPI com os principais frameworks de APIs do mercado, onde enfatiza a melhor escolha sendo a intermediária.

Fornecer as principais funcionalidades chave como:

- Rapidez: alta performance, quando pareado com aplicações em NodeJS e Go;
- Produtividade: aumento na velocidade de produção de novas funcionalidades em 200% a 300%, segundo estimativas em testes com o time de desenvolvimento ao realizar a criação de aplicações em ambiente de produção;
- Menos bugs: redução de 40% da inclusão de bugs acidentalmente por erro humano, segundo estimativas em testes com o time de desenvolvimento ao realizar a criação de aplicações em ambiente de produção;
- Intuição: fornece amigável integração com IDEs ao prover documentação de funcionalidades internas, complementação de código dinâmica e menor tempo durante as depurações de programa;
- Facilidade: foi desenhada para ser fácil de usar e aprender, requerindo menos tempo para ler documentações;
- Minimalista: redução da criação de código duplicado, além de prover várias funcionalidades para cada declaração de parâmetro, resultando em menos bugs;
- Robusta: obtenha códigos mais rápidos em ambiente de produção com a documentação interativa;
- Baseado nos melhores padrões: baseado e totalmente compatível com os padrões e definições das APIs, como OpenAPI ^[13] (previamente conhecido como Swagger) e JSON Schema ^[14].

3.5 Uvicorn

O Uvicorn ^[15] é um otimizador de performance para implementação de servidores ASGI utilizando bibliotecas Python uvloop ^[16] e httptools ^[17].



Figura 5: Uvicorn promove um ganho considerável na performance do servidor ASGI.

Devido a necessidade de implementação funcionalidades de processamento assíncrono em aplicações Python, o Uvicorn veio para cobrir essa demanda. Com a introdução do ASGI as aplicações permitiram que frameworks com essa necessidade permitissem esse tipo de processamento.

O ASGI deve fornecer suporte no ecossistema Python de aplicações web e seus demais frameworks onde são altamente competitivos contra os de NodeJS e Go em termos de alto tráfego de informações nos seus limites de E/S (Entrada/Saída). Além disso, promove o suporte para HTTP/2 e WebSockets, nos quais podem ser manipulados através de servidores WSGI.

O Uvicorn também fornece suporte para HTTP/1.1 e WebSockets, onde a implementação de suporte HTTP/2 ainda está sendo planejada.

4. Análise de Sistemas Existentes

Segundo o ONS ^[18] e seus sistemas, estes atuam na normatização da operação da distribuição da matriz energética no cenário nacional, além de cuidar da formatação destas normas e instruções operacionais e atuam principalmente nos cenários de pré-operação; operação em tempo real; e pós-operação.

A pré-operação consolida a programação eletroenergética diária através de intervenções nas restrições operativas dos reservatórios junto com o Controle Automático de Geração (CAG).

A operações em tempo real e tem como por objetivo coordenar, supervisionar e controlar o funcionamento operacional da Rede de Operação, além do devido funcionamento das instalações do SIN.

A pós-operação complementam a cadeia da Operação do Sistema com o apuramento dos dados das operações realizadas, bem como análises das ocorrências reportadas, perturbações energéticas e variações nas instalações elétricas do SIN, divulgações de resultados para os agentes do setor como órgãos governamentais, normativos, fiscalizadores e para a sociedade.

Todavia, este projeto tem como enfatizar um único item destes processos de atuação do ONS, que é o cenário de pós-operação relacionado a unificação das análises das ocorrências reportadas pelas UHEs através dos boletins diários via grupamento de dados relacionais e API pública de consulta.

CAPÍTULO III

VISÃO DO SISTEMA

1. Riscos do Projeto

Tabela 1: Riscos do Projeto

Nº	Descrição do Risco	Classificação
R001	Complexidade na implementação dos dados relacionais referentes aos clusters, devido as alterações estruturais dos boletins gerados quanto sua padronização e nomenclaturas oficiais do ONS.	M
R002	Complexidade na implementação de robôs nos quais utilizam frameworks <i>Web Scraping</i> para extração de dados dos boletins diários do ONS, devido a possibilidade de mecanismos <i>anti-crawling</i> utilizados como padrões de segurança cibernética que impedem que essas operações sejam realizadas por operações automatizadas.	A

2. Restrições do Projeto

Tabela 2: Restrições do Projeto

Nº	Descrição da Restrição	Tipo
R001	Implementações de modelos de aprendizagem de máquina.	Tecnológica
R002	Operações com sistemas distribuídos para <i>Data Warehousing</i> .	Tecnológica
R003	Autenticação e autorização por protocolos de segurança, como: <i>OAuth</i> , <i>Bearer Token</i> ou qualquer outra tecnologia de validação de segurança para as consultas direcionadas ao SisBIN provido.	Tecnológica
R004	Técnicas e mecanismos para burlar funcionalidades de segurança durante as operações de extração de dados por robôs automatizados através de frameworks <i>Web Scraping</i> em Python.	Tecnológica

3. Regras do Negócio

Tabela 3: Regras de Negócio.

Código	Nome	Descrição
RN001	Manutenção dos boletins diários do ONS	Apenas o ONS é responsável pela manutenção permanente dos boletins diários gerados através do endereço eletrônico https://sdro.ons.org.br/SDRO/DIARIO/index.htm .
RN002	Persistência de dados obtidos através dos boletins diários	O sistema estabelece a persistência dos dados extraídos pelos relatórios diários de modo a organizar as informações e assim promover uma base uniforme de dados relacionais.
RN003	Atualização sob demanda da extração de dados remotos	Para estabelecer as devidas persistências das informações relevantes extraídos por neste sistema, os robôs automatizados serão acionados manualmente para geração de novos relatórios e atualização da base de dados local envolvida.
RN004	Imutabilidade das fontes ou origens das consultas remotas	O sistema não utilizará outras fontes para consulta de informações de interesse, apenas dentro do domínio do ONS, anteriormente citados na regra de negócio RN001. Sendo assim, a disponibilidade deste serviço é essencial para o devido funcionamento do sistema.
RN005	Padronização dos dados através de entidades relacionais	Os dados emitidos através dos relatórios dos robôs automatizados serão padronizados em formato de entidades relacionais, de modo a auxiliar no desenvolvimento de esquemas para a persistência na base de dados. Logo, promovendo a melhor compreensão de como os dados persistidos serão organizados e estruturados.
RN006	Acionamento via API dos robôs	O sistema não conterá mecanismos de acionamento dos robôs com autenticação ou autorização pelo serviço da API disponibilizada, pois este será para fins educacionais a fim de demonstrar o devido funcionamento de diversas funcionalidades automatizadas e integradas no sistema. Logo, o acionamento será feito através de um <i>endpoint</i> exposto de thread única.

4. Requisitos Funcionais

Tabela 4: Requisitos Funcionais.

Código	Nome	Descrição
RF001	API para consultas via endpoints	Este requisito aborda a criação de uma API para realização de consultas através de endpoints remotos aos usuários do sistema e assim facilitar a integração com demais sistemas adjacentes.
RF002	Base de dados relacional para configuração de	Este requisito aborda a criação e configuração de uma base de dados relacional, na qual suporte a interpretação SQL para manipulação das estruturas de dados relacionais, a fim de persistir os dados

	grandes massas de dados dos boletins diários	gerados pelos robôs automatizados e que serão disponibilizados nas operações de endpoints da API.
RF003	Robôs automatizados para extração de dados gerados pelos boletins do ONS	Este requisito aborda a criação de robôs que desempenhem o papel de extrair dados oriundos dos boletins gerados pelo ONS através de frameworks <i>Web Scraping</i> .
RF004	Operações de correlação de dados relacionais para geração de relatórios preditivos contra incidentes em UHEs	Este requisito aborda a criação de um endpoint específico para chamada de rotinas de cálculo científico e estatístico de modo a providenciar relatórios de predição contra possíveis incidentes em UHEs através de análise dos boletins diários do ONS de acordo com as normas da ANEEL.

5. Requisitos Não Funcionais

Tabela 5: Requisitos Não Funcionais.

Código	Nome	Descrição
RNF001	Arquitetura API REST	Este requisito aborda a utilização da arquitetura REST para a API desenvolvida.
RNF002	Interpretador de dados estruturais através do SQL	Este requisito aborda a utilização do padrão estrutural de Entidade-Relacionamento para manipulação de suas estruturas, persistidas na base de dados, através do SQL.

6. Mensagens do Sistema

Tabela 6: Mensagens do Sistema.

Código	Descrição
MSG001	Dados inválidos.
MSG002	Clusters vazios ou falha no acionamento do robô.
MSG003	Provedor de dados do ONS inativo ou dados inexistentes.

7. Lista de Casos de Uso

Tabela 7: Lista de Caso de Uso.

Código	Nome	Descrição
UC001	Manter Boletim	Este caso de uso tem como finalidade a realização das operações de CRUD no quesito de manipular as informações dos boletins diários do ONS.
UC002	Acionar robô	Este caso de uso tem como finalidade a realização de operações de acionamento em escala de robôs automatizados para extração de dados dos boletins diários do ONS.
UC003	Gerar relatórios preditivos	Este caso de uso tem como finalidade a realização de operações internas dentro da API para cálculo

		preditivo de incidentes em UHEs, dado um período estipulado de análise de acordo com os dados já persistidos pelo sistema.
--	--	----------------------------------------------------------------------------------------------------------------------------

8. Lista de Atores

Tabela 8: Lista de Atores.

Código	Nome	Descrição
001	Administrador	Tem acesso aos principais recursos exclusivos do sistema, como acionamento dos robôs automatizados para extração de dados e manipulação dos dados fornecidos para cada boletim diário do ONS.
002	Usuário comum	Tem acesso somente as análises preditivas de dados para predição de incidentes em UHEs fornecidos pelo sistema.

9. Diagrama Geral de Caso de Uso

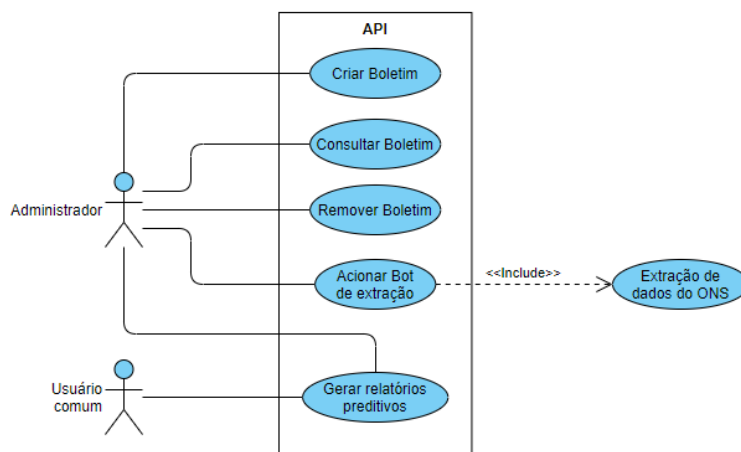


Figura 6: Diagrama de Caso de Uso do SisBIN para representação das operações diretas à API e afins.

10. Diagrama de Classe

Figura 7: Diagrama de Classe.

A ser adicionado.

11. Diagrama de Sequência

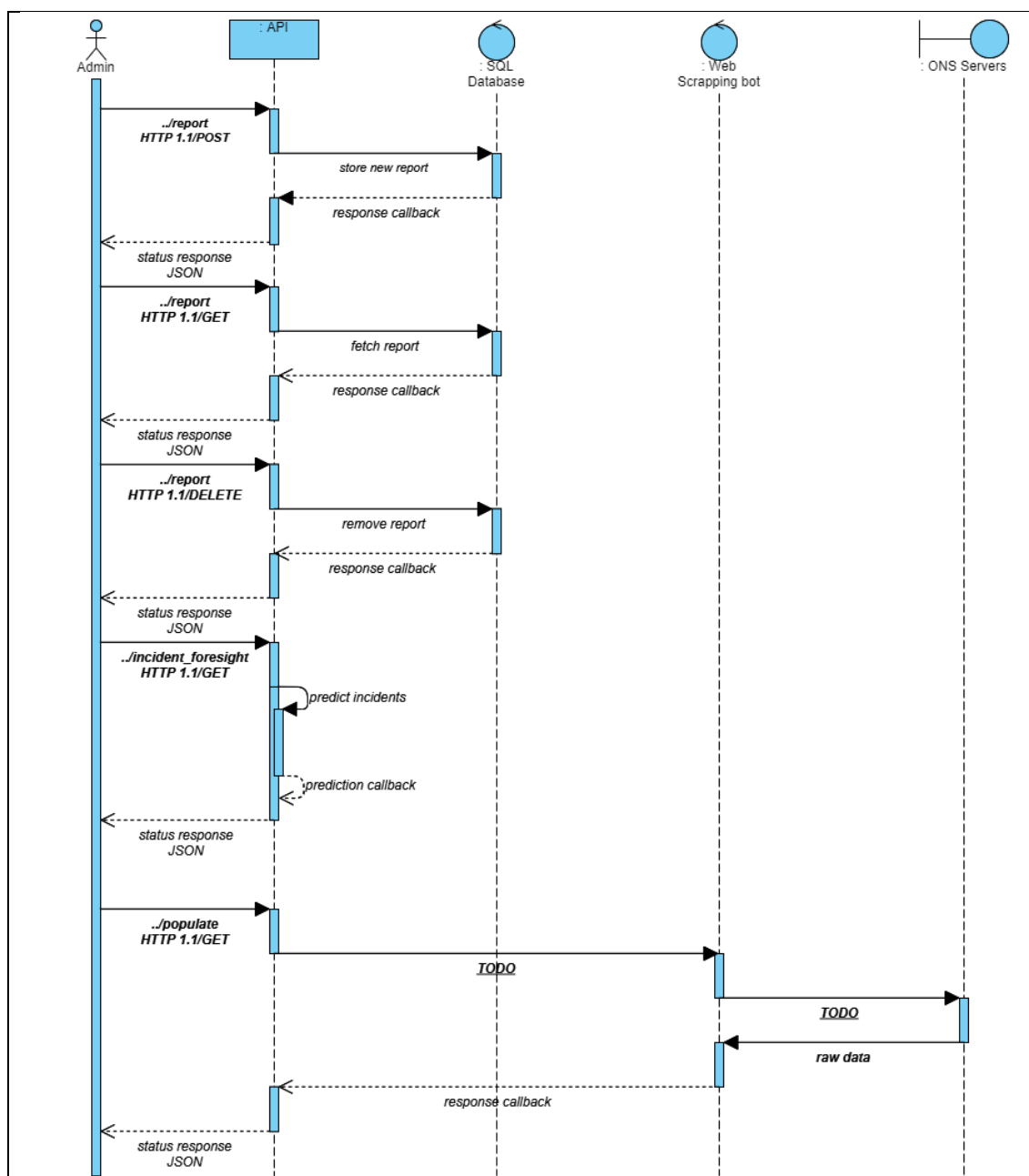


Figura 8: Diagrama de Sequência.

CAPÍTULO IV

ESPECIFICAÇÃO DE CASO DE USO

*[Este capítulo tem como base inserir o aluno na prática e no desenvolvimento de especificações de casos de uso, bem como apresentar de forma simples e direta todos os mecanismos aplicados a escrita de documentos sistêmicos.
Cada especificação de caso de uso deverá iniciar em uma nova página]*

UC001 – GERENCIAR PARECER TÉCNICO

1. Descrição

Este caso de uso tem como finalidade a realização das consultas do Parecer Técnico feitas pelos Coordenadores de Análise dos órgãos do XXXXXX.

2. Pré-condições

- O ator precisa estar logado no sistema SALIC;
- O ator deve ter acesso a esta funcionalidade.

3. Fluxos

3.1 Fluxo Principal – Consultar Parecer Técnico

FP01. O caso de uso se inicia quando o ator seleciona a opção Gerenciar Parecer Técnico no menu principal da aba Análise.

FP02. O sistema apresenta à tela um menu de opções e um campo de pesquisa de PRONAC. [TL001][RN001][RF001]

FP03. O ator preenche o campo com o PRONAC e seleciona a opção Pesquisar. [FA02] [FA03] [FA04] [FA05] [FE02] [FE03][RN002]

FP04. O sistema apresenta na tela uma lista com todos os pareceres técnicos do projeto vinculados a matrícula do usuário. [TL002] [RN001] [RF001]

FP05. O ator seleciona a opção Votar. [FA06][MSG001]

FP06. O caso de uso se encerra.

3.2 Fluxos Alternativos

FA3.2.1. Visualizar Parecer Técnico.

FA3.2.1.1. O ator solicita a opção Visualizar Parecer.

FA3.2.1.2. O sistema traz à tela o parecer técnico. [TL002]

FA3.2.1.3. O ator seleciona a opção Voltar.[RN003]

FA3.2.1.4. O sistema retorna ao passo P04.

FA3.2.2. Imprimir Etiqueta e Processo.

FA3.2.2.1. O ator solicita a opção Imprimir Etiqueta e Processo.

FA3.2.2.2. O sistema abre uma caixa de pesquisa de PRONAC. [TL003]

FA3.2.2.3. O ator preenche o campo com o número PRONAC e seleciona a opção Confirmar. [FE02] [FE03]

FA3.2.2.4. O sistema imprime a Etiqueta e Processo e volta ao passo P06. [RN005]

FA3.2.3. Imprimir Parecer Técnico.

FA3.2.3.1. O ator solicita a opção Imprimir Parecer Técnico.

FA3.2.3.2. O sistema abre uma caixa de pesquisa de PRONAC. [TL004]

FA3.2.3.3. O ator preenche o campo com o número PRONAC e seleciona a opção Confirmar. [FE02] [FE03]

FA3.2.3.4. O sistema imprime o parecer técnico e volta ao passo P06. [RN006]

FA3.2.4. Gerar PDF/Imprimir.

FA3.2.4.1. O ator solicita a opção Imprimir.

FA3.2.4.2. O sistema gera uma página em PDF, abre uma caixa de impressão e retorna ao passo P06.

FA3.2.5. Gerar XLS

FA3.2.5.1. O ator solicita a opção Gerar XLSX.

FA3.2.5.2. O sistema gera um documento em formato XLS e retorna ao passo P06.

FA3.2.6. Voltar

FA3.2.6.1. O ator solicita a opção Voltar.

FA3.2.6.2. O sistema retorna ao último passo que o ele executou.

3.3 Fluxos de Exceção**FE3.1.Tempo de sessão esgotado.**

FE3.1.1. O sistema redireciona o ator para a interface de autenticação do usuário, retorna a mensagem de alerta e volta ao passo P01. [MSG002] [RN004]

FE3.2.Dados inválidos.

FE3.2.1. O sistema exibe a mensagem MSG003 e retorna ao passo anterior.

FE3.3.Dados obrigatórios não informados.

FE3.3.1. O sistema exibe a mensagem MSG004 e retorna ao passo anterior.

4. Pós-condições

- N/A.

5. Pontos de Extensão

- UC002 – Consultar Dados do Projeto.

6. Pontos de Inclusão

- N/A.

[Na especificação de caso de uso será necessário chamar todas as Regras de Negócio; Requisitos Funcionais; Requisitos Não Funcionais; Mensagem e Tela que estejam relacionados. Padronizar a especificação de caso de uso de acordo com o diagrama de caso de uso e diagrama de sequência.]

CAPÍTULO V

MODELAGEM DE DADOS

[Este capítulo deverá propor ao aluno aplicar todo o conhecimento adquirido sobre modelagem de dados]

1. Diagrama de Entidade Relacional

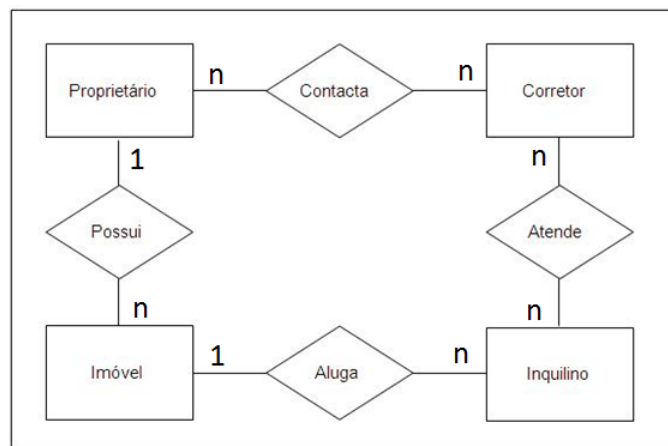


Figura 06 – Diagrama de Entidade Relacional

2. Diagrama de Entidade Relacional

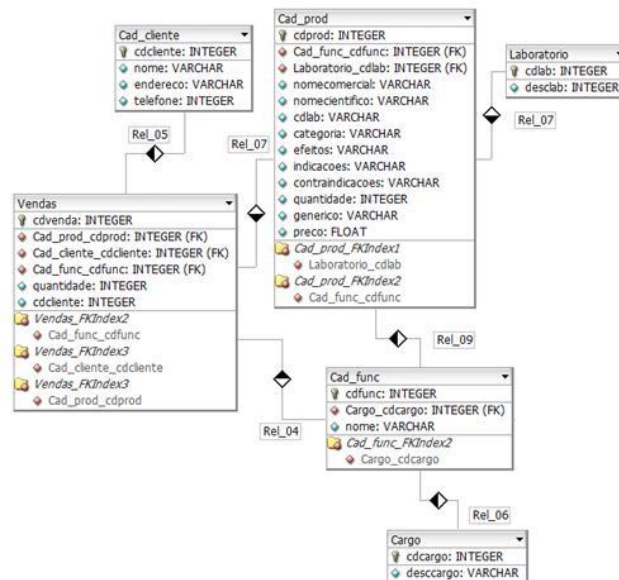


Figura 07 – Diagrama de Banco de Dados: MER

3.1 Dicionário de Dados

3.1.1 Tabela [Nome da Tabela]

Campo	Tipo/Tamanho	Obrigatório	Comentário
[Informar o nome do campo]	[Informar o tipo do campo]	[Informar S para obrigatório e N para não obrigatório]	[Comentar sobre o campo]
RELACIONAMENTOS			
Tabela	Descrição		
[Informar o nome da tabela]	[Realizar uma descrição sobre o campo da tabela que se deseja relacionar]		

[Padronizar o nome das tabelas. Exemplo: DBProcesso]

CAPÍTULO VI

PROTOTIPAÇÃO DO SISTEMA

[Este capítulo deverá auxiliar o aluno na modelagem do sistema a ser desenvolvido]

1 Manter processos <Identificação de Tela>

[Descrever o funcionamento ou proposito da tela no sistema]

The screenshot shows a window titled "Processos" with a standard Windows-style title bar. The form inside is organized into several sections:

- Client Information:** Two text boxes labeled "Cliente" and "Parte Contraria" containing "NOME COMPLETO DO CLIENTE" and "NOME COMPLETO DA PARTE CONTRÁRIA" respectively.
- Process Identification:** Fields for "Processo" (12345678901234567890), "Nº Processo Apenso", "Vara" (65), "Pasta" (8585), and a date (11/12/2010).
- Timeline and Type:** "Entrada" (18/09/1995), "Distribuição" (05/12/1995), and "Tipo" (TRABALHISTA).
- Legal Representation:** "Posição" (Autor), "Advogado" (a blue bar), and "Estado" (SP).
- Location and Court:** "Comarca" (SAO PAULO), "Forum" (BARRA FUNDA), "Cartório" (0), and "Setor" (0).
- Case Details:** "Natureza" (TRABALHISTA), "Instância" (1a. Instância), "Fase" (inicial), "Liminar" (Não Pedida), and "Espécie" (NORMAL).
- Observation:** A large text area labeled "Observação" with a vertical scrollbar.
- Actions:** Four buttons at the bottom: "Baixar Processo", "Gravar", "Excluir", and "Fechar".

Figura 07 – Tela do Sistema: Manter Processos

*[Neste tópico, será informado cada tela do sistema. Para cada tela deverá haver um título padrão. Exemplo: TELA 01 – NOME DA TELA.
Para cada tela, haverá uma descrição de cada campo da tela como também uma descrição dos comandos da tela. Exemplo:]*

1.1 Campos da Tela

Item	Nome do Campo	Tipo	Tam	Máscara	Obrigatório	Valor Padrão	Editável	Domínio	Visível
001	Cliente	Alfanumérico	100	N/A	N	N/A	S	Nome do campo na base de dados	S

1.2 Comandos da Tela

Item	Comando	Ação	Restrições/Observações
01	Gravar	Grava o processo no sistema	Todos os campos obrigatórios devem ser preenchidos

[Sempre alterar entre imagem do prototipo e a descrição da tela]

CONCLUSÃO

[Informar as considerações finais relacionado ao trabalho]

REFERÊNCIAS

- [1]. “O que é ONS – ONS”, acessado em 02 de setembro de 2021 e disponível em <http://www.ons.org.br/paginas/sobre-o-ons/o-que-e-ons> ;
- [2]. “Requirements Elicitation and Elicitation Technique Selection: Model for Two Knowledge – Intensive Software Development Processes.”, Hickey, A. M.; Davis, A. M. 36ª Conferência Internacional Anual do Havaí. Janeiro de 2003;
- [3]. “Waterfall Methodology”, acessado em 15 de outubro de 2021 e disponível em <https://www.sciencedirect.com/topics/computer-science/waterfall-methodology> ;
- [4]. “Brainstorming – Técnicas de Elicitação de Requisitos de Engenharia de Software UFSC”, acessado em 17 de outubro de 2021 e disponível em <https://retraining.inf.ufsc.br/guia/app/classificacoes/tecnicas-de-elicitacao-de-requisitos/entidades/tecnicas-de-elicitacao-de-requisitos-brainstorming> ;
- [5]. “Observação Direta – Técnicas de Elicitação de Requisitos de Engenharia de Software UFSC”, acessado em 17 de outubro de 2021 e disponível em <https://retraining.inf.ufsc.br/guia/app/classificacoes/tecnicas-de-elicitacao-de-requisitos/entidades/tecnicas-de-elicitacao-de-requisitos-brainstorming> ;
- [6]. “WebDriver | Selenium”, acessado em 07 de novembro de 2021 e disponível em <https://www.selenium.dev/documentation/webdriver/> ;
- [7]. “WebDriver – W3C”, acessado em 07 de novembro de 2021 e disponível em <https://www.w3.org/TR/webdriver1/> ;
- [8]. “The Python Tutorial – Python 3.10.0 documentation”, acessado em 07 de novembro de 2021 e disponível em <https://docs.python.org/3/tutorial/> ;
- [9]. “Extending and Embedding the Python Interpreter – Python 3.10.0 documentation”, acessado em 07 de novembro de 2021 e disponível em <https://docs.python.org/3/extending/index.html#extending-index> ;
- [10]. “Python/C API Reference Manual – Python 3.10.0 documentation”, acessado em 07 de novembro de 2021 e disponível em <https://docs.python.org/3/c-api/index.html#c-api-index> ;
- [11]. “About MariaDB Server – MariaDB.org”, acessado em 08 de novembro de 2021 e disponível em <https://mariadb.org/about/> ;
- [12]. “FastAPI”, acessado em 09 de novembro de 2021 e disponível em <https://fastapi.tiangolo.com> ;

- [13]. “OAI/OpenAPI-Specification: The OpenAPI Specification Repository”, acessado em 09 de novembro de 2021 e disponível em <https://github.com/OAI/OpenAPI-Specification> ;
- [14]. “JSON Schema | The home of JSON Schema”, acessado em 09 de novembro de 2021 e disponível em <https://json-schema.org> ;
- [15]. “Uvicorn”, acessado em 09 de novembro de 2021 e disponível em <https://www.uvicorn.org> ;
- [16]. “MagicStack/uvloop: Ultra fast asyncio event loop.”, acessado em 09 de novembro de 2021 e disponível em <https://github.com/MagicStack/uvloop> ;
- [17]. “MagicStack/httpptools: Fast HTTP parser”, acessado em 09 de novembro de 2021 e disponível em <https://github.com/MagicStack/httpptools> ; e
- [18]. “Atuação – Sobre o ONS”, acessado em 21 de Outubro de 2021 e disponível em <http://www.ons.org.br/paginas/sobre-o-ons/atuacao> .