



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Richard Santos Barbosa

**Software para Controle de versões do PI System**

Florianópolis  
2021

Richard Santos Barbosa

## **Software para Controle de versões do PI System**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Leandro Buss Becker, Dr.

Supervisor: Fabrício Costa, Eng.

Florianópolis

2021

### Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<http://portalbu.ufsc.br/ficha>

Richard Santos Barbosa

**Software para Controle de versões do PI System**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 13 de julho de 2021.

Prof. Hector Bessa Silveira, Dr.  
Coordenador do Curso

**Banca Examinadora:**

Prof. Leandro Buss Becker, Dr.  
Orientador  
UFSC/CTC/DAS

Fabício Costa, Eng.  
Supervisor  
Radix - Engenharia e Software

Prof. Felipe Gomes de Oliveira Cabral, Dr.  
Avaliador  
UFSC/CTC/DAS

Prof. Ricardo José Rabelo, Dr.  
Presidente da Banca  
UFSC/CTC/DAS

Dedico este trabalho à Deus o qual me deu forças quando eu não era capaz de continuar sozinho e me ensinou a perseverar em todos os momentos.

## **AGRADECIMENTOS**

Agradeço aos meus pais que me apoiaram sempre nos estudos.

Agradeço ao meu irmão, Alexander, por estar sempre me motivando e ensinando sobre a vida.

Agradeço aos meus avós que me ensinaram a amar e confiar mesmo quando tudo estiver incerto.

Agradeço à Deus por mais uma oportunidade de crescimento.

Agradeço à Radix por me fornecer uma oportunidade incrível de crescimento profissional e pessoal.

Agradeço à Universidade Federal de Santa Catarina por me capacitar a ser um profissional ético.

Agradeço ao meu orientador Leandro Becker o qual me ensinou a ter sensibilidade e respeito com todos.

Agradeço ao meu supervisor na Radix, Fabricio Costa, por ser um exemplo de ser humano e profissional.

*“A persistência é o caminho do êxito.”*  
Charles Chaplin

## RESUMO

Atualmente nos deparamos com indústrias de produção em massa constituídas de diversos tanques, bombas, aquecedores e outros equipamentos os quais necessitam de constante monitoramento, pois, um problema pode acarretar severas consequências de perdas humanas e monetárias. O PI System é uma solução para monitoramento e automação da planta como um todo, a partir desta ferramenta podemos adquirir dados em tempo real e interpretá-las sem a necessidade de estar presente na planta. Este trabalho apresenta uma ferramenta para controle de versões das plantas presentes no PI System, pois no processo de desenvolvimento e mapeamento das tags há a possibilidade, principalmente em times grandes de trabalho, de ações as quais originam resultados indesejados pelo cliente sendo necessário a retomada de pontos anteriores da hierarquia presente no PI System. O projeto conta com a utilização das linguagens de programação Python e VBA para atingir os requisitos estipulados e a construção do software foi elaborada para atuar em um ambiente de máquina virtual com restrições de instalação.

**Palavras-chave:** PI System. Hierarquia. Controle de versões.



## **ABSTRACT**

We are currently faced with mass production industries consisting of several tanks, pumps, heaters, and other equipment that need constant monitoring, as a problem can cause severe consequences of human and monetary losses. The PI System is a solution for monitoring and automation of the plant as a whole, from this tool we can acquire data in real-time and interpret it without the need to be present in the plant. This work presents a tool for version control of the plants present in the PI System, because, in the process of developing and mapping the tags, there is the possibility, especially in large work teams, of actions that originate unwanted results by the client, being necessary the resumption of previous points in the hierarchy present in the PI System. The project relies on the use of Python and VBA programming languages to achieve the stipulated requirements and the construction of the software was designed to work in a virtual machine environment with installation restrictions.

**Keywords:** PI System. Hierarchy. Version control.

## LISTA DE FIGURAS

Figura 1 – Pirâmide da Automação . . . . .	14
Figura 2 – Plano 10 - PDI . . . . .	17
Figura 3 – Exemplo modelo relacional . . . . .	20
Figura 4 – Exemplo banco de dados temporal . . . . .	21
Figura 5 – Banco de dados temporal em conjunto com relacional . . . . .	22
Figura 6 – Hierarquia de elementos no AF . . . . .	27
Figura 7 – Atributos . . . . .	27
Figura 8 – Aba análises de um elemento . . . . .	28
Figura 9 – Template de uma bomba . . . . .	29
Figura 10 – PI SMT . . . . .	30
Figura 11 – Excel com Add-in PI Builder . . . . .	31
Figura 12 – Overview VS Code . . . . .	33
Figura 13 – Estrutura Dataframe . . . . .	35
Figura 14 – VBA Overview . . . . .	37
Figura 15 – VBA - Projetos . . . . .	37
Figura 16 – VBA - Propriedades . . . . .	38
Figura 17 – VBA - Janela de código . . . . .	39
Figura 18 – Planilhas do gerador de Log . . . . .	40
Figura 19 – Casos de uso do sistema . . . . .	43
Figura 20 – Diagrama de atividades . . . . .	44
Figura 21 – Comunicação Python e .NET . . . . .	45
Figura 22 – Extração de elementos e atributos . . . . .	47
Figura 23 – Extração de templates e atributos . . . . .	49
Figura 24 – Criação de arquivo CSV . . . . .	50
Figura 25 – Fluxo do código de transformação . . . . .	51
Figura 26 – Criação de abas . . . . .	52
Figura 27 – Importação de CSVs . . . . .	52
Figura 28 – Definição de Status dos Elementos/Templates . . . . .	53
Figura 29 – Execução do software . . . . .	54
Figura 30 – Arquivo CSV de elementos . . . . .	56
Figura 31 – Arquivo CSV de templates . . . . .	57
Figura 32 – Aba de Elementos . . . . .	58
Figura 33 – Aba de Templates . . . . .	58

## LISTA DE TABELAS

Tabela 1 – Comparativo dos principais historiadores PIMS . . . . .	25
Tabela 2 – Colunas dos Elementos . . . . .	46
Tabela 3 – Colunas dos Templates . . . . .	48
Tabela 4 – Planilha LogSheet . . . . .	59

## LISTA DE ABREVIATURAS E SIGLAS

AF SDK	Asset Framework Software Developer's Kit
BAT	Batch File
BDR	Banco de dados relacional
BDT	Banco de dados temporal
CLP	Controlador Lógico Programável
CLR	Common Language Runtime
CRUD	Create, Read, Update and Delete
KPI	Key Performance Indicator
OCS	OSIsoft Cloud Services
OMF	OSIsoft Message Format
OPC	Object Linking and Embedding for Process Control
PI AF	PI Asset Framework
PIMS	Plant Information Management System
PI SMT	PI System Management Tools
SCADA	Supervisory Control and Data Acquisition
SGBD	Sistema de gerenciamento de banco de dados
SQL	Structured Query Language
VBA	Visual Basic for Application
VBS	Visual Basic Script

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	OBJETIVOS	15
1.2	METODOLOGIA	15
1.3	ESTRUTURA DO DOCUMENTO	16
<b>2</b>	<b>RADIX</b>	<b>17</b>
<b>3</b>	<b>TECNOLOGIAS RELACIONADAS</b>	<b>19</b>
3.1	BANCO DE DADOS	19
<b>3.1.1</b>	<b>Bando de dados relacionais</b>	<b>19</b>
<b>3.1.2</b>	<b>Bando de dados temporais</b>	<b>20</b>
3.2	PIMS	21
<b>3.2.1</b>	<b>Historiador</b>	<b>22</b>
3.2.1.1	Compressão de Dados	22
3.2.1.2	Aquisição de Dados	23
<b>3.2.2</b>	<b>Ferramenta para gerenciamento</b>	<b>23</b>
<b>3.2.3</b>	<b>Ferramenta para desenvolvimento</b>	<b>24</b>
<b>3.2.4</b>	<b>Ferramenta para visualização</b>	<b>24</b>
<b>3.2.5</b>	<b>Comparativo entre os principais fabricantes</b>	<b>25</b>
3.3	OSISOFT PI SYSTEM	25
<b>3.3.1</b>	<b>PI Asset Framework</b>	<b>25</b>
<b>3.3.2</b>	<b>PI System Management Tools</b>	<b>30</b>
<b>3.3.3</b>	<b>PI Builder</b>	<b>31</b>
<b>3.3.4</b>	<b>PI AF SDK</b>	<b>32</b>
3.4	VISUAL STUDIO CODE	32
3.5	PYTHON	33
<b>3.5.1</b>	<b>PyInstaller</b>	<b>34</b>
<b>3.5.2</b>	<b>Python.NET</b>	<b>34</b>
<b>3.5.3</b>	<b>Pandas</b>	<b>34</b>
3.6	JUPYTER NOTEBOOK	35
3.7	EXCEL	36
3.8	VBA	36
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>40</b>
4.1	REQUISITOS DO SISTEMA	41
4.2	CASOS DE USO	42
4.3	ESTRUTURA DO PROJETO	43
4.4	EXTRAÇÃO	44
<b>4.4.1</b>	<b>Extração de Informações de Elementos</b>	<b>45</b>
<b>4.4.2</b>	<b>Extração de Informações de Templates</b>	<b>47</b>

4.4.3	<b>Criação de arquivos CSV</b>	50
4.5	TRANSFORMAÇÃO	50
4.5.1	<b>Criação de abas e configurações no Excel</b>	51
4.5.2	<b>Leitura e carregamento de arquivos CSV</b>	52
4.5.3	<b>Criação de colunas com status das planilhas</b>	53
4.6	EXECUÇÃO	54
4.6.1	<b>Criação de arquivo VBS e BAT</b>	54
5	<b>RESULTADOS</b>	56
5.1	EXTRAÇÃO	56
5.2	TRANSFORMAÇÃO	57
5.3	EXECUÇÃO	57
5.4	TESTES	57
5.5	DISCUSSÃO	59
5.5.1	<b>Pontos positivos</b>	60
5.5.2	<b>Pontos negativos</b>	60
6	<b>CONCLUSÃO</b>	61
6.1	TRABALHOS FUTUROS	61
	<b>REFERÊNCIAS</b>	63

## 1 INTRODUÇÃO

Atualmente, no geral, dentro de indústrias, temos diversos equipamentos, instrumentos, análises, Key Performance Indicator (KPI)s e variáveis sendo atualizadas em tempo real. Este fluxo ocorre diariamente, sendo necessário acompanhamento para se antecipar e evitar possíveis problemas nos diversos componentes. Uma forma de realizar este controle se dá na utilização de um sistema de gerenciamento de informações da planta (Plant Information Management System (PIMS)) por meio do PI System.

Plantas de grande porte podem chegar a estruturas complexas envolvendo diversos dispositivos de chão de fábrica, sendo assim, difícil o acompanhamento. Pode se pensar em um exemplo em que algum erro inesperado compromete o sistema fazendo com que o engenheiro ou semelhante verifique válvula por válvula, controlador por controlador a fim de descobrir o equipamento que está apresentando problemas.

Na empresa encontram-se grupos de trabalho paralelos que estão atualizando a hierarquia dos equipamentos e áreas de processo constantemente podendo causar conflitos ou problemas em que seja necessário recorrer a versões passadas. Em suma surge a necessidade de um controle de versões de forma fácil que não venha atribuir maior quantidade de tempo ao escopo do projeto.

A solução proposta neste trabalho trata de um software com o intuito de automatizar o processo de criação de versões da hierarquia de elementos e templates já criados no PI System pela equipe. Portanto verifica-se se uma redução significativa de tempo destinado a esta tarefa podendo destinar o funcionário para outras tarefas de maior relevância e impacto no escopo do projeto.

O conjunto de ferramentas do PI System desenvolvidas pela OSIsoft se encontra no nível 4 na pirâmide da Automação, para simplificar e encapsular cada item contido nos níveis temos a figura 1.



Figura 1 – Pirâmide da Automação

Fonte: (ANDRADE, 2020)

## 1.1 OBJETIVOS

O objetivo principal do projeto desenvolvido é a criação de um software que desenvolva uma planilha com todas as informações necessárias para caso necessário, recriar a “arvore” ou apenas verificar modificações realizadas de uma hierarquia atual. Necessariamente o software irá realizar todas as tarefas de maneira automática.

## 1.2 METODOLOGIA

As fases de desenvolvimento foram criadas a fim de separar cada ferramenta podendo caracterizar a extração por meio de Python como o tratamento de dados e o Excel como o lado de visualização.

Basicamente tem se três etapas principais do projeto:

- Extração;
- Transformação;
- Execução.

A fase de extração trata, de forma sucinta, da aquisição dos dados do PI System onde para tanto se escolheu a linguagem de programação Python.

A fase de transformação manipula os dados extraídos de forma a organizar a fim de capacitar uma fácil visualização e a possível recriação de elementos pelo plug-in PI Builder.

A fase de Execução trata da ligação entre as fases de extração e transformação via script desenvolvido em VBS e arquivo BAT para execução de todas as fases por meio do clique do usuário.

As principais ferramentas utilizadas ao longo do desenvolvimento foram:

- Linguagem de programação Python
- OSIsoft PI
- Excel
- Visual Basic for Application (VBA)
- Visual Basic Script (VBS)
- Batch File (BAT)

A metodologia de desenvolvimento utilizada foi a de projetos ágil Scrum, a qual fundamenta se em ciclos de desenvolvimento. Em geral, tem-se reuniões diárias no intuito de sanar dúvidas pendências do dia anterior e definir objetivos para o dia atual.



Portanto há um acompanhamento frequente sobre o desenvolvimento do projeto, o que permite, caso necessário, uma mudança de direção ágil, evitando futuros retrabalhos (SIMÕES, 2017).

### 1.3 ESTRUTURA DO DOCUMENTO

No capítulo 2, há a apresentação da empresa Radix a qual forneceu todos os subsídios e requisições para o projeto acontecer.

No capítulo 3, tem-se a exposição de todas as ferramentas utilizadas ao longo do projeto de forma a contextualizar e explicar o funcionamento principal do software desenvolvido. Os principais conceitos estão a ferramenta PIMS, OSIsoft PI, Python e excel, outras ferramentas secundárias como Visual Studio Code e Jupyter Notebook utilizadas ao longo do projeto serão detalhadas devido a serem presentes ao longo do projeto na parte de desenvolvimento do código.

No capítulo 4, trata-se dos detalhes do projeto definindo os casos de uso, requisitos a serem atendidos pelo software desenvolvido e as etapas de implementação: Extração, transformação e execução.

No capítulo 5, expõe-se os resultados adquiridos pelo software, os testes realizados e uma possível aplicação.

No capítulo 6, conclui-se a respeito dos requisitos contemplados e detalhamento dos resultados gerais obtidos pelo desenvolvimento do software.

## 2 RADIX

A Radix é uma empresa de engenharia e software que atua em diversas áreas da tecnologia, desde o meu ingresso nesta, há a preocupação com o estagiário para estimular o crescimento pessoal e profissional por meio de cursos e acompanhamento (RADIX, 2021).

A Universidade Radix é um programa que visa fornecer diversos treinamentos interessantes para seus colaboradores, sendo esta, de grande importância para capacitação dos estagiários.

Há a utilização de ferramentas interessantes para a análise do rendimento atual do colaborador e acompanhamento segundo seus planos pessoais e profissionais. Uma destas ferramentas se chama PDI (Plano de Desenvolvimento Individual). Basicamente, é dividida em sete tópicos que podem ser qualificados seguindo o Modelo de aprendizagem 70/20/10.

A Experiência representa o 70: trata se da aprendizagem em contexto profissional. O Relacionamento representa o 20: Aprendizagem social. Basicamente, vem da integração e aquisição de conhecimento com outras pessoas. A Educação representa o 10: formação estruturada. Retrata o conhecimento adquirido por meio de cursos, formação acadêmica e afins.

Em suma o PDI estipula um possível caminho para alcançar determinado plano. Por exemplo, temos a figura 2 representando um plano de 10.

PLANO	AÇÃO E DESENVOLVIMENTO	COMO	RESULTADO ESPERADO	PRAZO	RESPONSÁVEL
10%	Conclusão da graduação em Engenharia de Controle e Automação	Realização do Estágio Obrigatório e TCC	Graduado em Engenharia de Controle e Automação	mai/20	Richard e UFSC

Figura 2 – Plano 10 - PDI

Fonte: Arquivo Pessoal

Para melhor descrever a empresa, foi retirado um breve trecho publicado pela Radix em que são retratados alguns feitos desta.

Empresa global com sedes no Rio de Janeiro e em Houston (EUA), a Radix atua nas áreas de tecnologia e engenharia, com o objetivo de transformar seu conhecimento técnico-científico em soluções qualificadas e com independência tecnológica para seus clientes. Com mais de 600 colaboradores nas sedes e nos escritórios de São Paulo, Belo Horizonte e Atlanta (EUA), a empresa é eleita desde a sua fundação, em 2010, uma das Melhores Empresas para se Trabalhar no Brasil e na América Latina. (RADIX, 2020)

A empresa atua em diversas áreas. Mais especificamente, o projeto foi desen-

volvido para um cliente com ênfase na área de polímeros e intermediários químicos (O nome da empresa e dados referentes a esta serão preservados). O cliente, possui diversas plantas espalhadas pelo mundo, podendo conter inúmeros equipamentos e instrumentos de chão de fábrica, portanto, é necessário agregar uma ferramenta para maior controle e visualização dos dados provenientes do chão de fábrica.

Toda a documentação foi fornecida. Coube ao cliente disponibilizá-la de forma rápida e organizada para que o trabalho seja entregue na data estipulada previamente apresentada a todos. Há a realização de reuniões semanais com o cliente a fim de estreitar relações e alinhar o projeto ao planejado.

A Radix, para este projeto, utiliza a metodologia ágil SCRUM em diversos pontos, como a realização de daily e acompanhamento constante do projeto, todas as tarefas são transparentes e diretas para todos. Alguns papéis estipulados pelo SCRUM não são claramente definidos no projeto, portanto, pode se inferir que é uma metodologia modificada a fim de alcançar os objetivos de maneira efetiva e ágil sem os papéis predefinidos por esta metodologia.

### 3 TECNOLOGIAS RELACIONADAS

Neste capítulo são tratadas as diversas ferramentas utilizadas ao longo do estágio, podendo ser específicas para chegar à solução ou auxiliares para desempenhar tarefas secundárias.

Os principais tópicos abordados são os tipos de banco de dados utilizados, detalhamento do sistema PIMS, solução PIMS da OSIsoft: PI System, Visual Studio Code como editor de código, a linguagem de programação Python, Jupyter Notebook utilizado para desenvolvimento e testagem do código, Excel como visualizador e editor de planilhas e o VBA para transformação de dados recebidos.

#### 3.1 BANCO DE DADOS

O banco de dados, em linhas gerais, armazena e organiza informações em um domínio específico. Em suma é o agrupamento de dados presentes em um mesmo contexto os quais necessitam serem armazenados no intuito de serem requisitados futuramente ou em âmbito de segurança de dados (SOUZA, 2021).

Os tipos de banco de dados citados nesta seção são utilizados na ferramenta PIMS e especificamente no OSIsoft PI System o qual foi amplamente utilizado neste projeto para aquisição de dados.

##### 3.1.1 Banco de dados relacionais

O banco de dados relacional possui a finalidade de armazenar e fornecer, via requisições, acesso à conjunto de dados relacionados entre si. O modelo relacional é o princípio que norteia os bancos de dados relacionais os quais apresentam uma maneira simples e direta de representar dados presentes em tabelas (ORACLE, 2021).

A linha de um banco de dados relacional possui um registro com uma ID exclusiva chamada chave. Cada coluna da tabela contém atributos dos dados existentes e cada registro, no geral, possui um valor para cada atributo, facilitando, portanto, o estabelecimento das relações entre os pontos de dados.

Geralmente os Banco de dados relacional (BDR) são utilizados para aplicações simples do tipo Create, Read, Update and Delete (CRUD). Para ilustrar pode se citar uma plataforma o qual um usuário pode criar uma reclamação e o dono visualizar, deletar caso não considere válido ou mesmo editar devido a inconsistências como erros ortográficos. Existem diversos cenários possíveis e a grande maioria se aplica às funcionalidades básicas envolvendo CRUD.

A imagem 3 demonstra os principais componentes presentes em um banco de dados relacional e fundamenta se nos conceitos abaixo:

- Tabelas (Entidades ou relações);

- Registros (tuplas ou linhas);
- Colunas (atributos):
- Chaves;
- Relacionamentos.

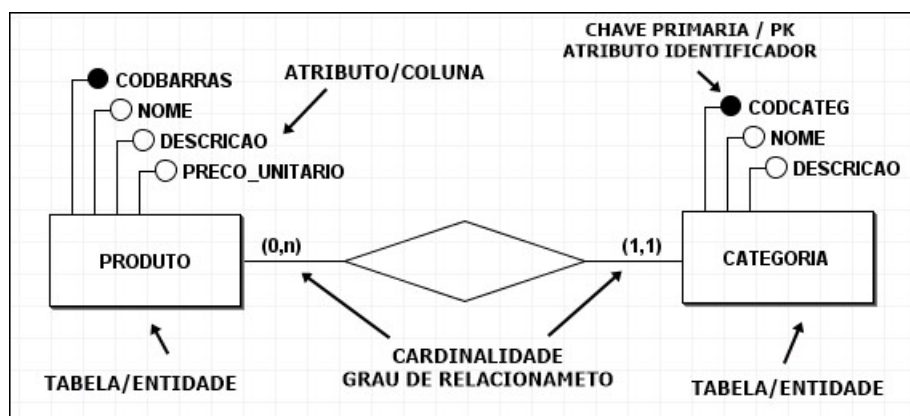


Figura 3 – Exemplo modelo relacional

Fonte: (SANTANA, 2019)

### 3.1.2 Bando de dados temporais

Um Banco de dados temporal (BDT) consiste em manter a evolução de um dado no tempo, portanto, armazena se os estados passado, presente e futuro. Os dados temporais são representados pelos tipos: DATE, TIME, DATETIME, TIMESTAMP, INTERVAL e PERIOD (MACÊDO, 2021).

Os dados possuem um tempo de validade (validade do dado no mundo real que será definido pelo usuário) e em alguns casos podem apresentar um tempo de transação (dado interno controlado pelo Sistema de gerenciamento de banco de dados (SGBD)).

Os dados temporais necessitam de serem avaliados segundo a granularidade temporal o qual faz referência à duração de um determinado dado, por exemplo, um salário poderá assumir uma granularidade mensal (devido a atualização do dado ser mensal) e um cargo profissional granularidade anual (devido ao cargo geralmente ser atualizado anualmente ou levar um grande período de tempo).

A figura 4 exemplifica um banco de dados temporal o qual consiste nas tabelas “Empregado” e “Funcao” com “Emp\_Funcao” exercendo o papel de relacionamento entre elas.

A tabela de relacionamento Emp\_Funcao possui dois atributos “INICIO” e “FIM” os quais determinam o período de tempo em que o empregado estará exercendo determinada função.

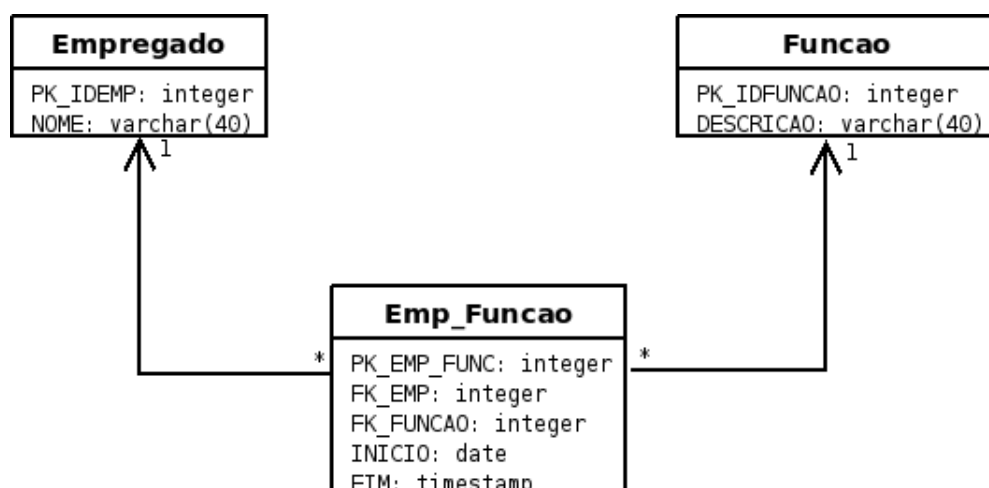


Figura 4 – Exemplo banco de dados temporal

Fonte: (MACÊDO, 2021)

O banco de dados temporal é utilizado no PI System para armazenamento de PI Points os quais são atualizados em períodos de tempo após alguma modificação considerável ocorrer. Os algoritmos existentes para atualização de dados e utilização do BDT serão abordados na seção seguinte.

### 3.2 PIMS

O PIMS é um sistema de gerenciamento de produção focado em armazenar, coletar e posteriormente disponibilizar os dados presentes em uma planta com o intuito de centralizar informações de processo e auxiliar na tomada de decisões de projeto, operacional e manutenção de equipamentos.

Atualmente encontram-se aplicações em diversos tipos de indústrias, em suma, os sistemas PIMS costumam ser utilizados em indústrias cuja necessidade de controlar muitos equipamentos e processos contínuos seja presente.

Através do PIMS, o engenheiro de processo responsável pode consultar, a partir dos componentes da ferramenta PIMS, o estado atual da planta com o estado de dias ou meses atrás, ou ainda mesmo comparar o estado atual com o de um dia considerado excelente, registrado no sistema como “referência” para medições de performances futuras.

Os principais componentes presentes em uma ferramenta PIMS são:

- Historiador;
- Ferramentas para gerenciamento;
- Ferramentas para desenvolvimento;
- Ferramentas para visualização e apresentação de dados.

O banco de dados utilizado em sistemas PIMS são temporais com relacionais como ilustrado na figura 5. O banco de dados temporal exerce o papel de armazenar todos os dados coletados nos sistemas de chão de fábrica.

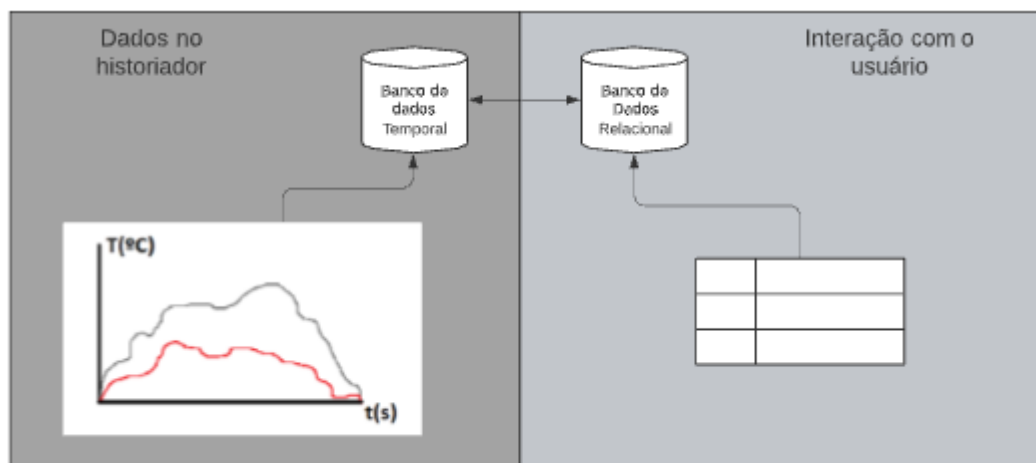


Figura 5 – Banco de dados temporal em conjunto com relacional

Fonte: Arquivo Pessoal

Todos os dados coletados são armazenados em um banco de dados temporal, entretanto este sofre limitações de performance quando envolve buscas complexas sobre grandes períodos de tempo.

Portanto o banco de dados relacional realiza a interação com o usuário, logo, será o banco de visualização em que o usuário possuirá contato importando e exportando dados do banco de dados temporal que encontra se em constante atualização por meio da aquisição de dados no historiador.

### 3.2.1 Historiador

Trata se de um componente cuja finalidade está situada no arquivamento dos dados e recuperação rápida dos dados caso solicitado. Tem se como funções primárias:

- Rápida recuperação dos dados;
- Compressão de dados;
- Aquisição de dados.

#### 3.2.1.1 Compressão de Dados

A compressão de dados se apresenta como um aspecto de grande importância para um sistema PIMS, a partir dela tem se armazenamento de dados da ordem de, por exemplo, anos em um HD tradicional.

O funcionamento da compressão parte da lógica da utilização de algoritmos os quais armazenam apenas dados com mudanças significativas.

Como principais requisitos para um bom algoritmo de compressão têm-se:

- Alta velocidade de compressão e descompressão;
- Alta taxa de compressão;
- Boa reconstrução dos dados;
- Segurança dos dados.

Swinging Doors e Boxcar/Backslope são exemplos de algoritmos utilizados na compressão de dados de historiadores aplicando o método do descarte de pontos fora de uma caixa de compressão.

### 3.2.1.2 Aquisição de Dados

A aquisição de dados é executada via interfaces de integração de dados. As principais interfaces de integração são pelo protocolo Object Linking and Embedding for Process Control (OPC), Supervisory Control and Data Acquisition (SCADA), comunicação direta com Controlador Lógico Programável (CLP), dispositivos de controle ou outras interfaces caso o fabricante do historiador forneça a opção. O historiador da OSIsoft oferece diversas interfaces para integração podendo ser consultado em seu site.

Cada forma de integração de dados possui sua vantagem pode se citar pelas três principais formas de aquisição:

- Protocolo OPC: Fornece dados provenientes de diversos CLPs, Controladores e entre outros dispositivos fornecendo maior centralização de dados, é necessário um servidor OPC para realizar a comunicação como, por exemplo, o Matrikon;
- SCADA: Os dados adquiridos se encontram em unidades de engenharia e, no geral, possuem uma maior quantidade de variáveis que os CLPs;
- Comunicação direta com CLP: Há dados mais precisos devido a maior frequência de aquisição de dados.

### 3.2.2 Ferramenta para gerenciamento

A ferramenta para gerenciamento do sistema PIMS trata de realizar toda a configuração do servidor, cadastro e edição de tags, recuperação de dados, conexões, segurança e licenças.

As PI System Management Tools (PI SMT) podem ser utilizadas como exemplos de ferramentas para gerenciamento de sistema PIMS. As PI SMT são ferramentas cuja



finalidade consiste em administrar os PI Servers a partir de conexões de cliente. A instalação das ferramentas presente no PI System Manager estão no kit incluso em cada PI Server e há a possibilidade de download separado (OSISOFT, 2021c).

### 3.2.3 Ferramenta para desenvolvimento

Trata de disponibilizar uma biblioteca cujo intuito é facilitar a extração de dados de forma direta para softwares terceiros e aplicativos personalizados por meio de linguagens de programação. Como exemplos fornecidos pela OSIsoft pode se citar:

- OSIsoft Message Format (OMF): Consiste em um conjunto de cabeçalhos e corpos de mensagens usados para gerar mensagens compatíveis para a inserção de dados em um PI Server local, Edge Data Store ou OCS (OSISOFT, 2021b);
- Cliente PI SQL: Disponibiliza a extração de dados do PI System por meio consultas Structured Query Language (SQL);
- PI Web API: Realiza a extração de dados para utilização em aplicações Web;
- Asset Framework Software Developer's Kit (AF SDK): Kit de desenvolvimento que fornece a comunicação para aplicativos .NET;
- OSIsoft Cloud Services (OCS): Viabiliza a comunicação do PI System com sistemas em nuvem.

### 3.2.4 Ferramenta para visualização

As ferramentas para visualização fornecem a possibilidade de apresentação de dados de maneira a facilitar a tomada de ações e monitoramento da indústria como um todo por meio de interfaces gráficas, interfaces de gerenciamento de tags, consulta a banco de dados e integração com ferramentas desktop como o Excel.

A partir das ferramentas de visualização há a possibilidade de disponibilização de resumos de lotes, eventos ocorridos e relatórios de KPIs aumentando a aquisição de conhecimentos do processo. Como exemplos fornecidos pela OSIsoft:

- PI DataLink: Add-in do Excel fornece a possibilidade de conexão com o excel para facilitar modificações e geração de relatórios;
- PI Vision: Ferramenta da OSIsoft que possibilita a criação de dashboards para desenvolvimento de apresentações de forma fácil e organizada.

Tabela 1 – Comparativo dos principais historiadores PIMS

Produto	Fabricante	Tipos de Relatório	Preço (R\$)
Infoplus.21	Aspentech	Web e Excel	R\$120000 por ano com suporte
PI System	OSIsoft	Web e Excel	R\$600000 no primeiro ano e posteriormente R\$120000 com suporte
FactoryTalk	Rockwell	Excel com um pacote menor de tags	R\$9945 apenas licença
Wonderware	Wonderware	No próprio software	R\$20995 apenas licença

Fonte: (SOUSA, 2014)

### 3.2.5 Comparativo entre os principais fabricantes

Tem se uma melhor visualização de opções a partir da tabela 1 a qual apresenta comparativo entre as principais fabricantes em termos de preço e funcionalidades.

O projeto foi realizado na ferramenta PIMS da OSIsoft: PI System devido a disponibilidade pela a empresa Radix, além de ser uma das opções mais famosas dentre as outras evidenciadas na tabela 1.

## 3.3 OSISOFT PI SYSTEM

O PI System é um tipo de sistema PIMS utilizada ao longo do projeto o qual fornece uma solução completa para o controle dos equipamentos e áreas de processo presentes.

O software fornecido pela empresa OSIsoft desempenha papel crucial ao longo do projeto. O PI System é dividido em softwares que fornecem subsídios para automação da planta de forma completa (OSISOFT, 2021b).

As ferramentas utilizadas ao longo do projeto são apresentadas a seguir. É importante salientar que as licenças foram todas disponibilizadas pela a empresa o qual o autor se encontra vinculado. Foi realizado um curso de capacitação ao longo do estágio para familiarização com todos os componentes referenciados nas subseções.

### 3.3.1 PI Asset Framework

Esta subseção descreverá as principais funcionalidades da ferramenta e componentes. Inicialmente é importante comentar a respeito de algumas estruturas e tipos de dados existentes deste módulo (OSISOFT, 2021a).

As principais estruturas são:

- PI Point;

- Elementos (Asset);
- Atributos;
- Análises;
- Templates;
- Tabelas.

PI Point: Tipo de dado padrão dentro do PI AF. Contém dados que são atualizados em tempo real, ou seja, que variam segundo a leitura de cada sensor com o dado referente ao equipamento e são recebidos via comunicação OPC com o chão de fábrica.

Elementos (Asset): É a forma que a hierarquia é dividida. Organizam todos os equipamentos em uma estrutura menos complexa e com maior facilidade de encontrar informações.

Atributos: Representam uma propriedade associada com o elemento. Em suma podem referenciar uma fórmula, um PI Point e etc.

Análises: Fornecem subsídios para criar análises que checam a validade de algo. É possível definir funções com “If” e condições em que, por exemplo, avaliamos se algo está dando errado.

Templates: São padrões presentes. Podemos setar para cada elemento seguir um determinado padrão de sua “classe”, por exemplo, uma bomba ou aquecedor que podem possuir marcas diferentes, porém possuem características semelhantes, podendo, portanto, setar atributos em comum e seguirem um template.

Tabelas: Podem seguir um padrão. Há a possibilidade de importar e utilizar funções para referenciar determinado elemento. São necessários códigos SQL para seleção dos dados.

Temos na figura 6 uma hierarquia de elementos exemplo, para ilustrar como o PI Asset Framework (PI AF) é organizado. No exemplo temos um Laboratório de pesquisa com zonas em que cada uma delas possui um soprador (BLR), resfriador (CHL) e aquecedor (HTR).

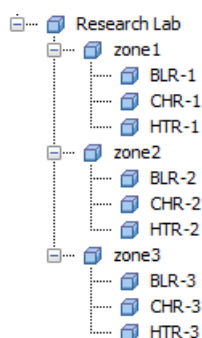


Figura 6 – Hierarquia de elementos no AF

Fonte: Arquivo Pessoal

A árvore de elementos representa a camada superficial dos processos. Podemos notar a facilidade com que conseguimos organizar uma planta qualquer dentro da ferramenta PI AF.

Na figura 7 está representado um elemento “Bomba 1” com os atributos Corrente, Corrente Nominal, Potência, Status e Tensão.

Captura de tela da interface de atributos de um elemento "Bomba1" no PI AF. A aba "Attributes" está selecionada. O filtro está em branco. A tabela de atributos é a seguinte:

Name	Value
Corrente	7,0459 A
Corrente Nominal	10 A
Potencia	1662,7
Status	Ligado
Tensao	220,45 V

Figura 7 – Atributos

Fonte: Arquivo Pessoal

A Corrente possui um PI Point relacionado, portanto está sendo atualizado, em tempo real, segundo sensores no chão de fábrica.

A Corrente Nominal representa uma constante em que é fornecida pelo fornecedor da bomba em questão;

A Potência possui um dado do tipo análise atrelado cuja fórmula é “Tensão x Corrente”;

O Status representa um dado do tipo análise com uma condição “IF” em que valores acima de corrente acima da corrente nominal indicam um status “Desligado”. Para valores abaixo a bomba estará “Ligado”;

A Tensão, por fim, possui um PI Point relacionado à qual é atualizada constantemente seguindo sensores presentes na bomba.

As Análises, representada na figura 8, desempenham um papel fundamental em criar condições e funções de grande importância ao longo dos processos, para exemplificar, temos setados a potência e o Status. Há possibilidade de utilizar funções prontas de complexidade alta e setar o tipo de análise dependendo do que for necessário ao cliente.

A figura 8 trata de um overview do componente de criação de análises.

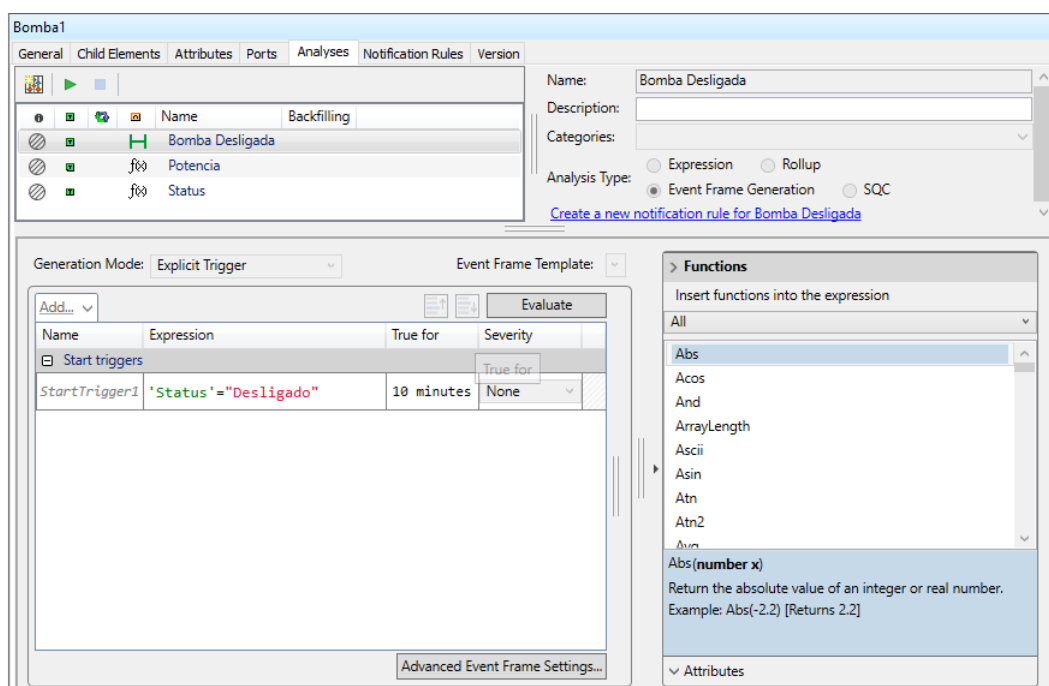
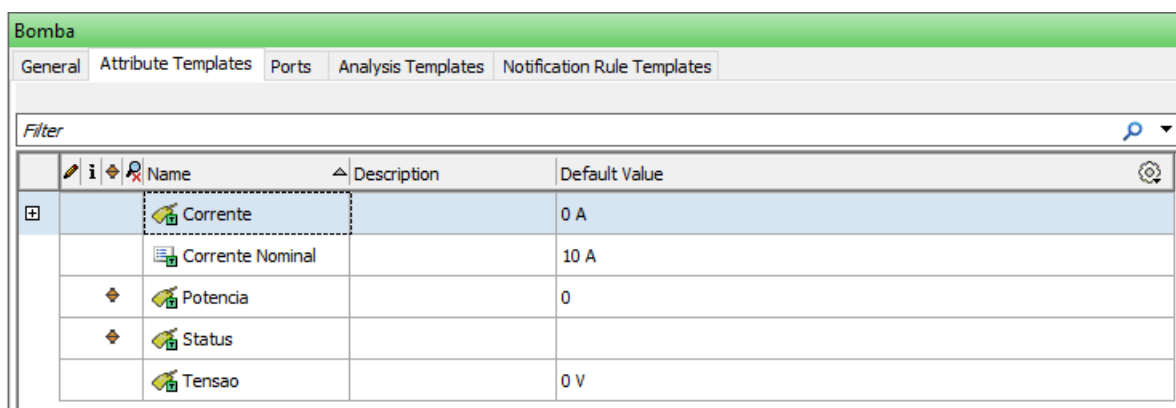


Figura 8 – Aba análises de um elemento

Fonte: Arquivo Pessoal

Há diversas funções possíveis a serem atribuídas a uma expressão, todas elas bem documentadas a respeito da utilização.

Os Templates, representados na figura 9, desempenham um papel de padronização dentro do PI AF, podendo atribuir diversos atributos que serão encontrados em todos os elementos que seguirem o template definido.



Name	Description	Default Value
Corrente		0 A
Corrente Nominal		10 A
Potencia		0
Status		
Tensao		0 V

Figura 9 – Template de uma bomba

Fonte: Arquivo Pessoal

No exemplo da Bomba 1 podemos observar que esta segue o template Bomba e possui os atributos pré-definidos pelo template. É importante salientar que há a possibilidade de criar regras de atualização do nome do PI Point, facilitando, portanto, ainda mais o papel de setar cada atributo dentro do PI AF.

Por exemplo, PI Points com nome “pump 1” tem se um possível padrão a ser atribuído “pump” + o número referente a bomba citada, sendo atribuído sempre o ultimo número presente em uma string com o nome do elemento que segue o template bomba.

Todos os templates desempenham um papel crucial dentro do processo de criação da Hierarquia de elementos, pois evitam possíveis retrabalhos para cada elemento. Por exemplo, todas as bombas necessitam do atributo fabricante. Ao invés de ser criado este atributo em cada uma das bombas, há possibilidade de criar no template referente à bomba. Assim, todas com o template possuirão, após criado com sucesso, o respectivo fornecedor.

Em uma planta complexa existe possibilidade de existência de diversas bombas, aquecedores e sopradores, por exemplo. Portanto, a necessidade de configurar o template de maneira abrangente, clara e direta é de extrema importância. É necessário o entendimento e observação de atributos em comum por tipo de equipamento, pois, caso o template seja configurado erroneamente, será necessário retrabalho e disponibilização de tempo extra para a tarefa de correção do template.

### 3.3.2 PI System Management Tools

O PI SMT é uma interface gráfica do usuário que possibilita a tarefa de administração do PI Data Archive, tais como (OSISOFT, 2021c):

- Configuração dos PI Points;
- Configuração dos Archives;
- Configurar as regras de segurança;
- Visualização de logs do Data Archive.

Em suma, a ferramenta, não será utilizada diretamente pelos estagiários, pois, a configuração e criação dos PI Points será feita pelo cliente. Para melhor explicar a ferramenta temos na figura 10 a visão geral.

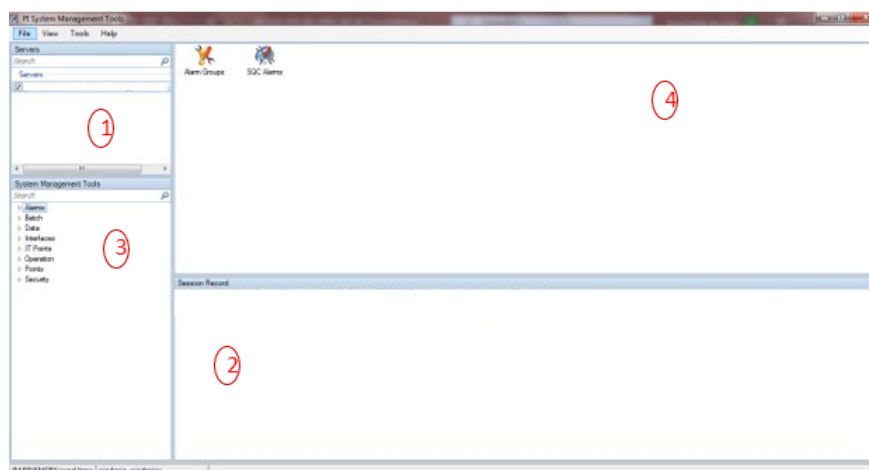


Figura 10 – PI SMT

Fonte: Arquivo Pessoal

As áreas com os números destacados na figura 10 representam:

1. Verifica a lista de servidores e estabelece conexão;
2. Visualização dos logs;
3. Menu de opções;
4. Área para realizar as configurações.

### 3.3.3 PI Builder

Add-in do Excel para Configurações do PI System.

Basicamente, o PI Builder, fornece a capacidade de extração de informações referente aos elementos, PI Points, templates e outros presentes dentro do PI AF, possibilitando a edição de maneira mais simples e rápida no ambiente do Excel.

A tarefa de criar diversos elementos pode acabar sendo cansativa e repetitiva dentro do PI AF, logo, pelo Excel, há flexibilidade fornecendo possibilidade para importação de um elemento qualquer dentro do PI AF, clonagem por meio de planilhas e exportação por meio do PI Builder ao PI AF, estes são apenas exemplos dentro de uma vasta gama de possibilidades de automação e otimização de tarefas repetitivas.

A figura 11 contém um overview de importação, ao Excel, de um PI Point e suas respectivas colunas que podem ser extraídas e exibidas para configuração.

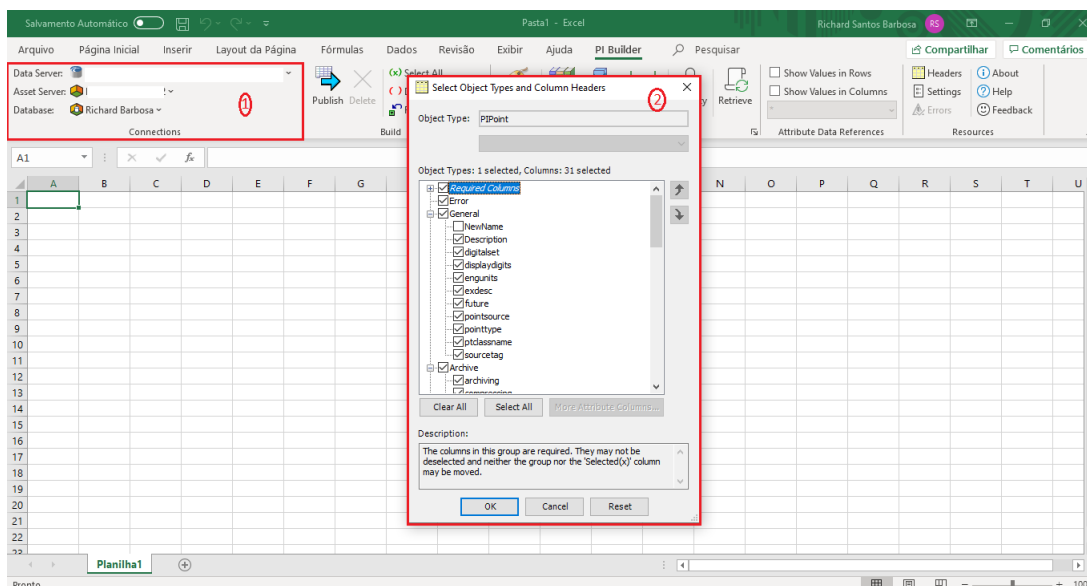


Figura 11 – Excel com Add-in PI Builder

Fonte: Arquivo Pessoal



Portanto, pela figura 11 nota-se que algumas configurações necessitam ser realizadas para extração de dados do PI AF por meio do PI Builder como:

1. Configuração do Servidor e Database a ser utilizada;
2. Seleção de header com diversas informações possíveis.

Em suma o PI Builder simplifica o processo de criação de elementos. Para o caso de plantas grandes é essencial esta ferramenta, pois, a criação de elementos de forma manual, torna-se exaustiva e pouco efetiva possibilitando erros humanos indesejados.

### 3.3.4 PI AF SDK

O AF SDK fornece uma interface programática abrangente baseada em Windows para o PI System.

A abordagem é orientada à objetos para interagir com as estruturas e os dados do PI AF. As principais características do AF SDK são:

- Baseado em .NET Framework;
- Tem mais métodos e opções disponíveis que qualquer outra tecnologia de acesso ao PI AF;
- Está limitado a sistemas operacionais Windows que rodam o .NET Framework.

É possível criar rapidamente hierarquias no PI AF complexas com o AF SDK. Muitos parceiros da OSIsoft preferem criar hierarquias AF dessa maneira, ao invés de utilizar o plugin PI Builder para o Microsoft Excel.

O AF SDK é ideal para desenvolver aplicativos de middleware situados entre o PI System e outros sistemas

O AF SDK provê flexibilidade no modo de integrar outros sistemas ao PI System e fornece a utilização efetiva dos recursos disponíveis no PI AF. Há possibilidade de automatizar a geração e a sincronização de elementos entre o PI AF e outros sistemas.

Por meio da integração da linguagem de programação Python e .NET utiliza-se o AF SDK para automatizar o processo de atualização de Log da estrutura no PI AF que será tratado em capítulos posteriores.

## 3.4 VISUAL STUDIO CODE

O Visual Studio Code é uma IDE com suporte para diversas linguagens. Com uma interface fácil e customizável, se apresenta como uma das principais ferramentas dentro da área de editores de texto. Um recurso interessante trata da integração

com o Git (GIT, 2021) de forma a atualizar constantemente a situação do projeto e sendo disponível a funcionalidade de dar commit sem ter que acessar o Git por fora (MICROSOFT, 2021a).

Uma das vantagens da utilização da IDE é a possibilidade de maior interatividade com o código. Algumas funcionalidades como o fornecimento de informações sobre uma função utilizada possibilitam ao programador maior domínio e conhecimento sobre o código em desenvolvimento

Há uma aba de extensões que permitem uma maior adaptabilidade dependendo da necessidade do usuário. Tem como ponto positivo essa modularidade no editor diferenciando de outros mais completos e mais pesados em quesito de tamanho da instalação e tempo de inicialização do editor.

Na figura 12 há um overview do editor.

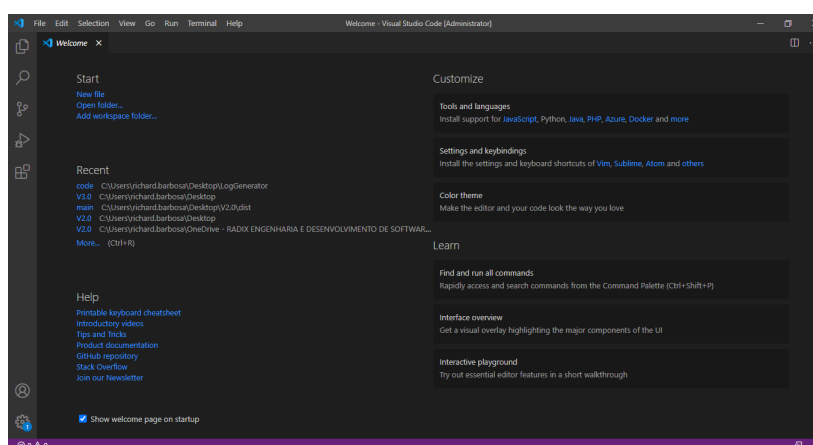


Figura 12 – Overview VS Code

Fonte: Arquivo Pessoal

As abas ao lado esquerdo permitem a seleção de pastas/arquivos, pesquisa de palavras chaves no script em desenvolvimento, controle de versões e administração de extensões.

Como requisitos para o projeto há necessidade de um editor de texto rápido com agregação de elementos do Python, Jupyter Notebook e leitura de arquivos CSV.

A desvantagem trata-se justamente da necessidade de instalação de módulos específicos para outras linguagens, porém, para este projeto, a ferramenta conseguiu agregar todos os requisitos de maneira completa e agradável.

### 3.5 PYTHON

A linguagem de programação Python vem sendo a mais popular dentre todas presente e se apresenta como uma ferramenta indispensável quando o tema relaciona produção de softwares (FERNANDES, 2021).

Python é orientada a objetos o qual como principal aspecto leva em conta a criação de classes, atributos e métodos como principal, mas não única, forma de programação. Possui uma estrutura de dados de alto nível no qual diz respeito a simplificar a experiência do programador para realizar tarefas mais elaboradas.

Há diversos módulos disponíveis para instalação de maneira fácil por meio do Python Package Index (PyPI). Os módulos fornecem meios diretos de utilização de funções prontas facilitando a programação.

Esta ferramenta foi utilizada com maior frequência para tópicos que envolviam automatização e integração de processos, os softwares auxiliares, foram, preferencialmente, desenvolvidos nesta linguagem por ser de fácil utilização, open source e todos estarem mais familiarizados.

### 3.5.1 PyInstaller

A biblioteca PyInstaller unifica um aplicativo Python e todas as suas dependências em um único pacote. A partir da execução desta ferramenta o usuário poderá executar o aplicativo unificado sem a necessidade de realizar a instalação de um interpretador Python ou outros módulos.

O PyInstaller não é um compilador cruzado, portanto, para fazer um aplicativo do Windows há a necessidade de executá-lo no Windows, caso seja necessário um aplicativo para Linux há a necessidade de executar em um computador com o sistema operacional Linux.

O sistema operacional da máquina virtual utilizada ao longo do projeto é o Windows, portanto, o PyInstaller foi executado em um sistema Windows.

### 3.5.2 Python.NET

Python.NET é um pacote que oferece integração da linguagem programação Python com o .NET Common Language Runtime (CLR) e fornece uma ferramenta de script de aplicativo para desenvolvedores .NET. Ele permite que o código Python acesse o CLR e há possibilidade de incorporação do Python em um software .NET (PYTHON.NET, 2021).

A biblioteca Python.NET foi necessária devido ao PI AF SDK ser um kit de desenvolvimento baseado em .NET, portanto para realizar toda a extração de dados do PI AF.

### 3.5.3 Pandas

A biblioteca Pandas está presente, pois proporciona maior facilidade na manipulação de dados e possui uma vasta gama de funções para análise e organização dos

dados além de fornecer a possibilidade de salvar o Dataframe no formato de arquivo CSV o qual será consumido pelo código VBA desenvolvido no Excel.

Pandas é uma biblioteca para manipulação e análise de dados, escrita em Python. Essa é a biblioteca ideal para iniciar análises exploratórias de dados devido ela permitir leitura, manipulação e plotagem de dados de maneira simplificada (RIBEIRO, 2021).

O principal componente da biblioteca Pandas é o dataframe o qual consiste em uma tabela composta por índice, colunas e linhas os quais podem ser facilmente manipulados por funções de manipulação de dados presentes na biblioteca. Tem-se na figura 13 um exemplo de estrutura do dataframe.

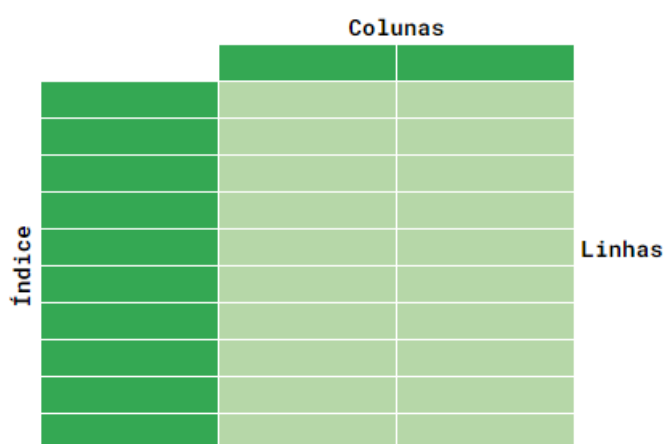


Figura 13 – Estrutura Dataframe

Fonte: (RIBEIRO, 2021)

O dataframe possibilita a utilização de álgebra relacional do tipo join, merge e concat os quais são vastamente utilizados em códigos SQL e serão utilizados na etapa de extração do projeto (PANDAS, 2021).

### 3.6 JUPYTER NOTEBOOK

Esta ferramenta possibilita a utilização de diversos tipos diferentes de linguagem de programação e união entre texto e código de maneira simples e eficiente. O Jupyter Notebook é indispensável para desenvolvimentos de softwares e posteriormente explicação de cada parte do script em um relatório (VIANA, 2021).

A instalação é simples e possui compatibilidade com Visual Studio Code, portanto, há um ambiente de desenvolvimento completo para elaboração do código sem maiores preocupações com debug e elaborações de scripts por outros meios como prompt de comando (MICROSOFT, 2021b).

O Jupyter Notebook foi utilizado com frequência na parte de desenvolvimento de códigos, devido a facilidade de correção de bugs e tratamento direto de módulos do

código, facilitando o debug.

### 3.7 EXCEL

No cotidiano de empresas é nítida a necessidade de ferramentas capazes de facilitar processos específicos e gerais, o Excel se apresenta como uma ferramenta paga de extrema importância devido aos componentes que esta pode nos proporcionar. Basicamente, qualquer pessoa no cenário atual informatizado, já utilizou planilhas para realizar alguma tarefa. Entretanto, nem todos conhecem o potencial do desenvolvedor de planilhas da Microsoft (MICROSOFT, 2021c).

Comparando com algumas concorrentes que não serão descritas, o Excel possui integração ao Teams da Microsoft e este é o principal meio de comunicação da empresa. Todas as licenças das ferramentas da Microsoft foram disponibilizadas pela empresa. Portanto, é mais um motivo para utilizar o Excel ao invés dos concorrentes. O PI Builder é um add-in do Excel, portanto, é necessária a ferramenta da Microsoft para integração com o PI Asset Framework.

Todas as planilhas foram organizadas dentro do Teams a fim de facilitar o acesso dos arquivos pelos empregados da Radix e o cliente. As planilhas são atualizadas em tempo real ao estar no Teams, proporcionando a funcionalidade de trabalho compartilhado (mais de uma pessoa alterando). A funcionalidade de controle de versões do Excel proporciona o controle total do arquivo podendo ser observadas as principais alterações presentes.

O software foi utilizado em todos os processos de produção, pois, o mapeamento dos dados do cliente, por exemplo, foi feito em planilhas, a lista de PI Points criados foi disponibilizado em planilhas e, em suma, no geral, todos os conjuntos de dados possuem formato de planilha do Excel.

### 3.8 VBA

A ferramenta VBA pode ser usada no Excel no intuito de fornecer outras opções para edição e controle de uma planilha. Em suma o VBA, basicamente, é uma linguagem de programação para Excel, pode se criar automatizações das atividades em uma planilha e criação de macros.

Basicamente o VBA no Excel permite que quase todas as ações realizadas a partir de um teclado, mouse ou uma caixa de diálogo possam ser automatizadas e programadas. A maior aplicação de VBA no Excel é para automatização de tarefas repetitivas.

Além da criação de scripts para automatização de tarefas cotidianas, há a possibilidade de desenvolvimento de novas funcionalidades aos aplicativos desenvolvidos seguindo a necessidade das regras de negócio pré-estabelecidas pela empresa, caso

uma regra de negócio, por exemplo, se fundamenta na necessidade de certo documento ser gravado em apenas uma pasta pré-determinada, há a possibilidade de criação de um aviso ao usuário lembrando de alterar o local do documento para a pasta correta. A figura 14 fornece um exemplo da interface do VBA em um Excel a fim de ilustrar, pois esta ferramenta n fica visível e passa despercebido pelos usuários do pacote Office.

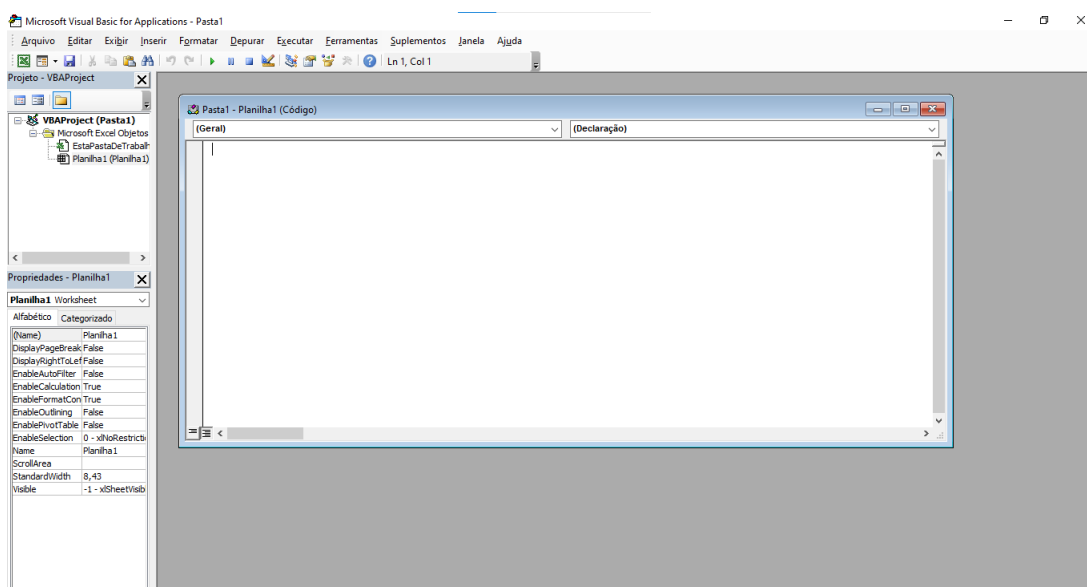


Figura 14 – VBA Overview

Fonte: Arquivo Pessoal

O menu de seleção de projeto está representado pela figura 15 o qual apresenta o espaço de trabalho do projeto desenvolvido. Há a possibilidade de navegação entre os diferentes arquivos que estão sendo modificados.

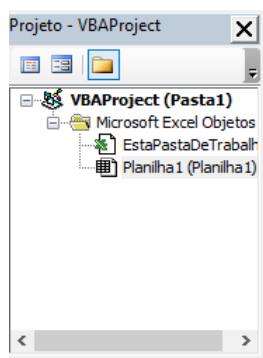


Figura 15 – VBA - Projetos

Fonte: Arquivo Pessoal

A janela Propriedades possui a finalidade de listar todas as suas configurações atuais e propriedades de tempo de design para um objeto selecionado, há a possibili-

dade de edição das propriedades e configurações diretamente da janela (MICROSOFT, 2021d).

Na figura 16 expõe um exemplo de janela Propriedades presente no VBA, é importante a demonstração desta janela devido as diferenças presentes em relação a outros editores como o VS Code (Editor utilizado neste projeto para criação de código em Python).

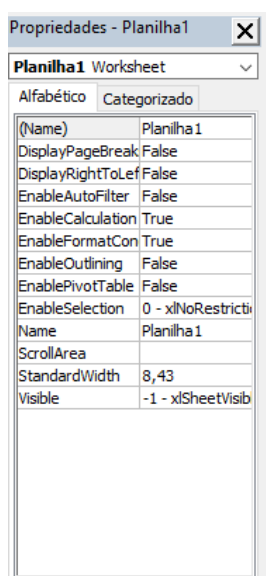


Figura 16 – VBA - Propriedades

Fonte: Arquivo Pessoal

Toda a edição de macros e definições a serem programadas são desenvolvidas na janela de código ilustrado pela figura 17. A caixa da direita superior representa qual função está sendo editada ou criada no momento e a da esquerda superior representa o elemento principal onde estão acontecendo as modificações.

Em suma o VBA realizou papel importante no projeto, pois foi necessário para tratamento dos dados de forma que os dados extraídos por meio do código em Python sejam organizados a fim de melhor visualização.

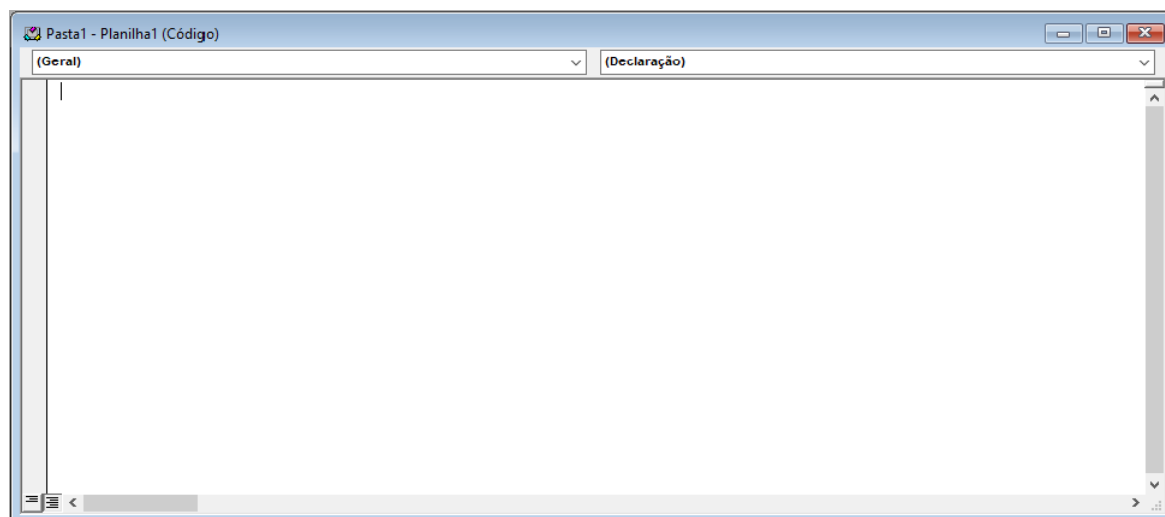


Figura 17 – VBA - Janela de código

Fonte: Arquivo Pessoal



## 4 DESENVOLVIMENTO

Devido à necessidade de um controle de alterações nos elementos e templates presentes no PI AF, surgiu a ideia de criação de um software cuja função principal trata de extrair informações dentro do servidor do cliente verificando-se o status diariamente (entenda como status a situação de modificado ou não alterado e removido). No capítulo de desenvolvimento serão apresentadas todas as etapas de criação de código para obtenção dos resultados que serão evidenciados no próximo capítulo.

A ideia inicial tratava da realização de uma tarefa programada no Windows para acontecer a cada semana sem a necessidade alguma de atuação do funcionário, entretanto, devido a limitações do ambiente de desenvolvimento, não há a permissão para criação de tarefas pelo “TaskScheduler” do Windows ou utilização de softwares de terceiros.

A subseção a seguir tratará dos resultados obtidos pelo código gerador de log cujo objetivo foi realizar um controle de versões da hierarquia de elementos e os templates presentes no PI AF.

O fluxo de planilhas do código está demonstrado na figura 18.

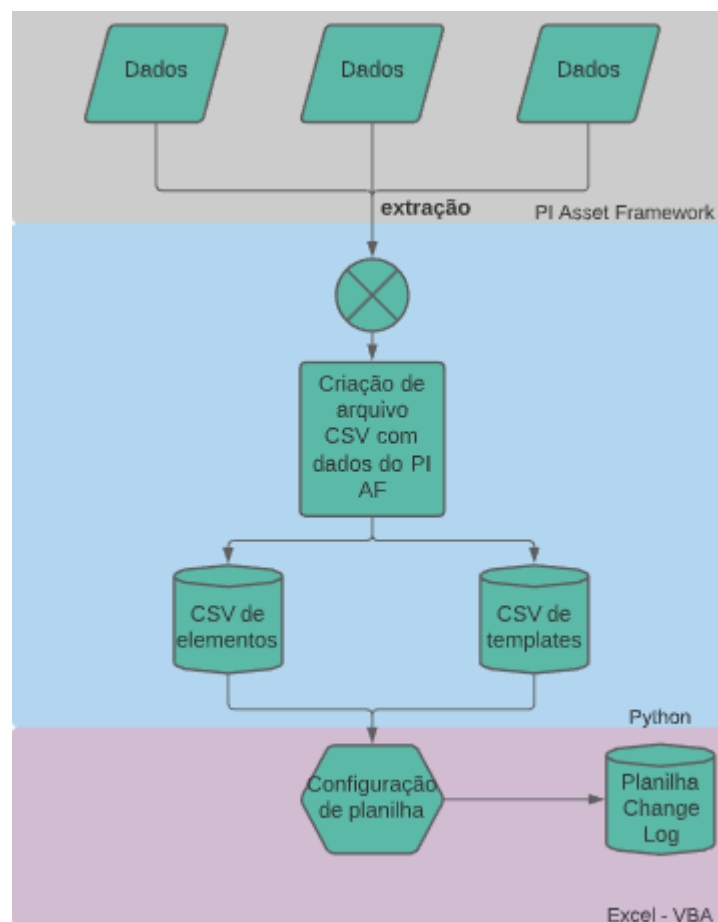


Figura 18 – Planilhas do gerador de Log

Fonte: Arquivo Pessoal

Inicialmente o código desenvolvido em Python recebe como entrada dados provenientes do PI AF. Os dados são processados e distribuídos em colunas que foram previamente solicitadas pelo cliente, posteriormente, são criados dois arquivos CSV:

- Tabela de elementos do PI AF;
- Tabela de templates do PI AF.

Após o processo do Python termos a configuração da planilha excel Change Log por meio de uma macro que receberá os arquivos CSV e tratará os dados a fim de obter abas personalizadas referente aos dias em que foram obtidos os arquivos e colunas status os quais possuem informações da situação do elemento e template em relação ao dia anterior.

Neste capítulo serão abordados os casos de uso presentes no sistema, requisitos, estrutura presente do projeto e detalhamento das etapas de desenvolvimento de projeto extração, transformação e execução.

#### 4.1 REQUISITOS DO SISTEMA

Os requisitos do sistema podem ser divididos em dois grupos: os requisitos funcionais e os requisitos não funcionais. Os primeiros têm como finalidade definir o que o sistema deve ser capaz de executar a fim de cumprir a sua função. Normalmente, estão altamente ligados com as funcionalidades do sistema. Já os requisitos não funcionais estabelecem a forma como serão alcançados os requisitos funcionais, impõem restrições ao sistema, define tecnologias a serem suportadas e/ou utilizadas etc (CANGUÇU, 2021).

Os requisitos do sistema foram definidos com base na necessidade em que o projeto como um todo estava presente, esse contexto trata da aplicação da ferramenta PIMS: PI System para um cliente e foi necessário o controle de versões do Asset Framework ou árvore de elementos presente no sistema do cliente a fim de amenizar e corrigir falhas no mapeamento de elementos, templates e atributos:

1. Fornecer dados atualizados dos elementos e seus atributos;
2. Seguir as pré-definições de colunas necessárias (nome, id, parente por exemplo);
3. Fornecer dados atualizados dos templates e seus atributos;
4. Criar abas referentes à data com que os dados foram requisitados pelo software;
5. Estabelecer status de modificado caso tenha diferenças nos elementos/templates em relação a data passada de atualização da planilha;

6. Definir o caminho do elemento a fim de facilitar navegação e entendimento da hierarquia.

Em relação aos requisitos não funcionais do sistema tem se:

1. Configuração em Excel devido a maior facilidade de visualização para o usuário;
2. Compatibilidade de nomes de colunas a fim de facilitar a utilização do PI Builder;
3. Definir cores com que chame a atenção do usuário para a linha;
4. Criar filtros nas colunas a fim de facilitar manipulação.

## 4.2 CASOS DE USO

Esta seção trata de apresentar os principais casos de uso do software desenvolvido representado pela figura 19. O sistema contará com dois agentes de uso:

- Usuário: agente responsável pela utilização e atualização do software (planilha com dados de cada dia);
- PI System: agente que irá armazenar todos os dados e enviará à planilha por meio de requisições efetuadas pelo software.

Os casos de uso de sistema serão detalhados abaixo a fim de evidenciar as principais funcionalidades fornecidas pelo software em questão.

1. O usuário será capaz de visualizar dados, pré-definidos pelo software, dos elementos e templates além de seus respectivos atributos;
2. Acessar as datas passadas as quais serão as abas presente na planilha;
3. Verificar status dos elementos e templates as quais referem se a indicar se foi modificado, excluído ou não alterado;
4. Retornar a uma versão passada por meio do add-in do Excel PI Builder.

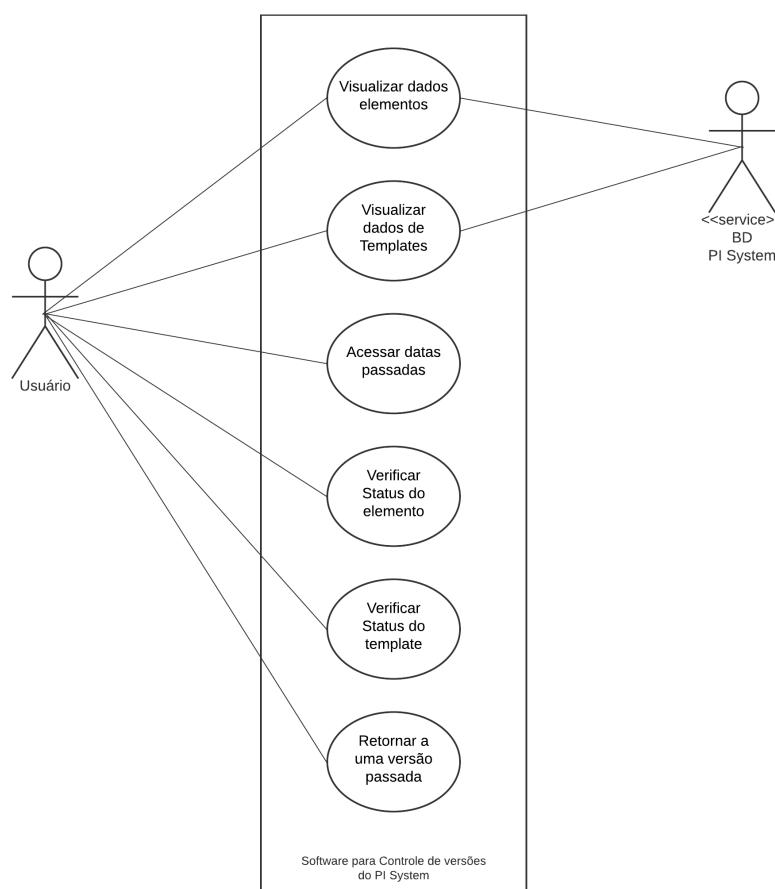


Figura 19 – Casos de uso do sistema

Fonte: Arquivo Pessoal

### 4.3 ESTRUTURA DO PROJETO

Esta seção trata de explicar as etapas de desenvolvimento utilizadas ao longo do projeto de controle de versões. Para a explicação do esqueleto do desenvolvimento foi desenvolvido um diagrama de atividades para explicar os passos na figura 20.

Em suma o processo de criação do software foi dividido em dois grupos, o grupo da extração cujo objetivo principal tratou da geração de arquivos CSV para adaptação no excel e o grupo de Transformação buscou a criação de uma macro na linguagem VBA, dentro da planilha de Logs, para organização e atribuição do status de cada elemento e template.

A Extração de informações de elementos e templates diz respeito à criação de uma tabela com as colunas de maior importância para o controle de versões, após a criação exporta-se para arquivos CSV para posteriormente ser consumida por uma macro desenvolvida pelo grupo de transformação.

O grupo de transformação cria colunas de status e removido, para informar o atual status de cada elemento e template referente à semana anterior além de aplicar

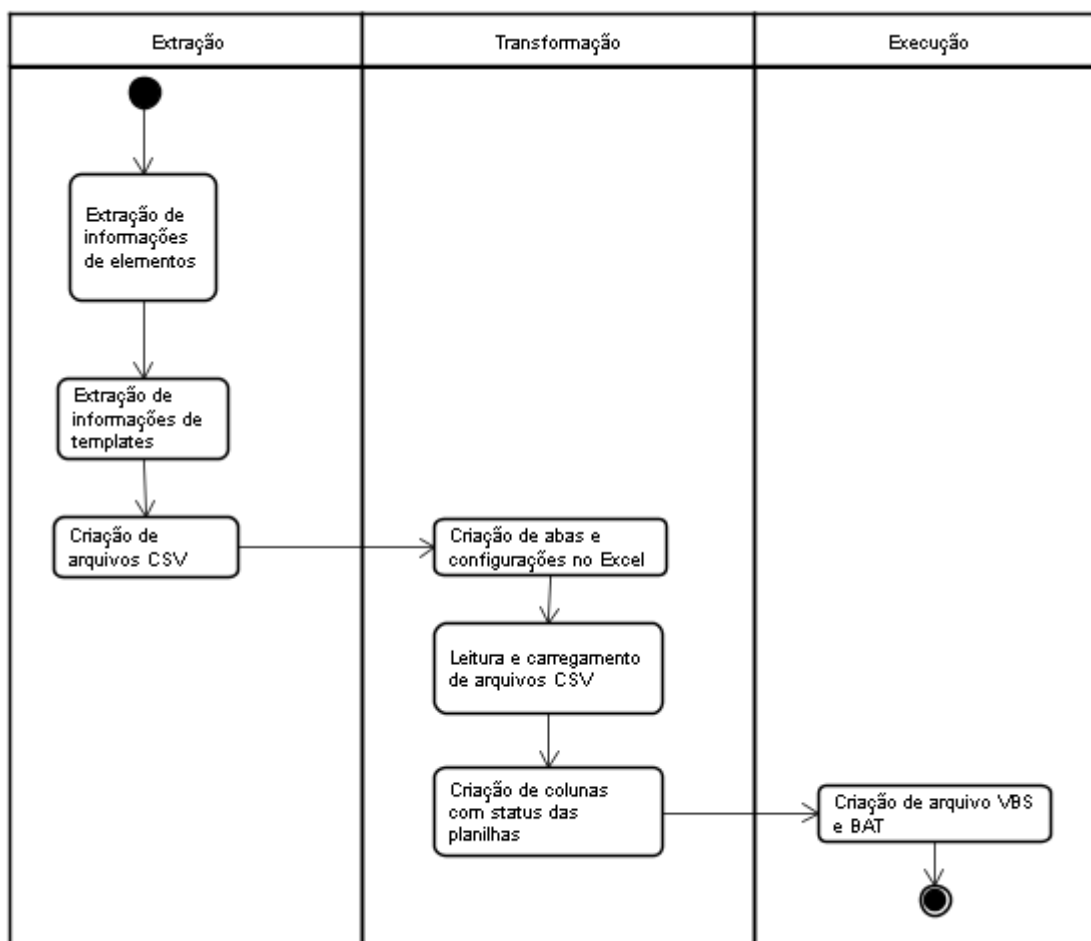


Figura 20 – Diagrama de atividades

Fonte: Arquivo Pessoal

filtros e organizar os dados provenientes da extração.

O arquivo VBS possui a finalidade de executar a macro desenvolvida. O arquivo BAT possui a finalidade de unir as duas vertentes do software (Extração e Transformação) por meio da execução do executável em Python que realiza as tarefas da execução e posteriormente rodar o arquivo VBS que configura e cria duas colunas extras que possuem os status.

Esta planilha é atualizada semanalmente e pode ser checada caso algo de errado tenha ocorrido.

#### 4.4 EXTRAÇÃO

Esta seção possui o intuito de detalhar a construção da etapa de extração do software cujo objetivo será obter dados contidos no banco de dados relacional presente no PI System. O banco de dados relacional presente no PI System, em linhas gerais, possui todos os elementos elaborados no Asset Framework.

Para a extração foi utilizado a linguagem de programação Python devido a

familiaridade do autor e a disponibilidade de diversas fontes de consulta na web. A comunicação do código com o banco de dados foi realizada via o kit de desenvolvimento AFSDK (SQUARE, 2021).

A biblioteca Python.NET possibilitou a integração entre Python e AFSDK a partir do código na figura 21.

```
1
2 import sys
3 import clr
4 import pandas as pd
5
6 sys.path.append(r'C:\Program Files (x86)\PIPC\AF\PublicAssemblies\4.0')
7 clr.AddReference('OSIsoft.AFSDK')
8 clr.AddReference('System.Collections')
9
10 from System.Collections.Generic import List
11 from OSIsoft.AF import *
12 from OSIsoft.AF.PI import *
13 from OSIsoft.AF.Asset import *
14 from OSIsoft.AF.Data import *
15 from OSIsoft.AF.Time import *
16 from OSIsoft.AF.UnitsOfMeasure import *
17 from OSIsoft.AF.Search import *
18 from OSIsoft.AF.Analysis import *
```

Figura 21 – Comunicação Python e .NET

Fonte: Arquivo Pessoal

O código da figura 21 demonstra o tratamento do python de um CLR namespace AFSDK como se fosse um pacote Python. Em suma a biblioteca Python.NET utiliza o CPython runtime (Implementação do Python na linguagem de programação C) para interagir com o .NET, fornecendo a possibilidade de utilização de CLR (PYTHON.NET, 2021).

Posteriormente a integração do Python e .NET por meio das linhas 6 à 8 há o tratamento do OSIsoft AFSDK como pacote do Python possibilitando a importação das classes presentes no kit de desenvolvimento (OSISOFT, 2021c).

#### 4.4.1 Extração de Informações de Elementos

Os nomes das colunas as quais serão utilizadas na planilha foram definidos com base no estabelecido no PI Builder os quais podem ser consultados na documentação da OSIsoft (OSISOFT, 2021b). Na tabela 2 estão listados os nomes de cada coluna dos elementos que serão impressos na planilha final.

A função de cada coluna será apresentada abaixo:

- ObjectType: Apresenta o tipo do elemento podendo ser um atributo, análise, regra de notificação e etc;

Nome da Coluna	Presente no PI Builder	Mandatário para PI Builder
ObjectType	Presente	Sim
Name	Presente	Sim
Description	Presente	Não
UniqueID	Presente	Não
Path	Não	Não
Template	Presente	Não
AttributeDataReference	Presente	Não
AttributeConfigString	Presente	Não
VersionModifier	Presente	Não
CreationDate	Não	Não
ModifyDate	Não	Não

Tabela 2 – Colunas dos Elementos

Fonte: Arquivo Pessoal

- Name: Nome estipulado do elemento
- Description: Descrição definida no Asset Framework;
- UniqueID: Chave unica atribuida a cada elemento presente;
- Path: Caminho referente ao elemento, mostra todos os parentes do elemento;
- Template: Respectivo template sendo utilizado no elemento;
- AttributeDataReference: String contendo o tipo da referência do atributo, como por exemplo um PI Point, String Builder, fórmula e etc;
- AttributeConfigString: String necessária para a configuração do atributo;
- VersionModifier: Representa o usuário que realizou as últimas alterações
- CreationDate: Data de criação do elemento;
- ModifyDate: Data referente à última modificação realizada.

O acesso aos dados foi executado acessando os atributos das classes presentes no AF SDK. Portanto utilizou se a classe de “AFEElement” presente no kit de desenvolvimento como ilustrado pelo código na figura 22.

O search é uma classe AFEElementSearch em que será utilizado o método “findObjects”o qual retorna uma lista de todos os elementos presente no DB (Banco de dados relacional do Asset Framework).

Count foi utilizado para validação interna do total de elementos presente no AF. As linhas 29 até 32 tratam de configurar listas e nomes das colunas que serão utilizados.

```

22 def createList(DB):
23     search = AFElementSearch(DB,"","*")
24     count = search.GetTotalCount()
25     elementsList = []
26     attrList = []
27     ML = []
28     row = []
29     elementsList.append(['ObjectType', 'Name', 'Description', 'UniqueID', 'Path', 'Template', 'AttributeDataReference',
30     'AttributeConfigString', 'VersionModifier', 'CreationDate', 'ModifyDate'])
31     attrList.append(['ObjectType', 'Name', 'Description', 'UniqueID', 'Path', 'Template', 'AttributeDataReference',
32     'AttributeConfigString', 'VersionModifier', 'CreationDate', 'ModifyDate'])
33     for element in search.FindObjects(fullLoad= True):
34         row = []
35         row.append(str(type(element)).replace("<class 'OSIsoft'", '').replace(".AF.Asset.", "").replace(">", ""))[2:])
36         row.append(element.Name)
37         row.append(element.Description)
38         row.append(element.UniqueID)
39         row.append(element.GetPath())
40         row.append(str(element.Template))
41         row.append('')
42         row.append('')
43         row.append(element.Version.Modifier)
44         row.append(str(IAFTransactable(element).CreationDate))
45         row.append(str(IAFTransactable(element).ModifyDate))
46
47         elementsList.append(row)
48
49         for attr in element.Attributes:
50             row = []
51             row.append(str(type(attr)).replace("<class 'OSIsoft'", '').replace(".AF.Asset.", "").replace(">", ""))[2:])
52             row.append(attr.Name)
53             row.append(attr.Description)
54             row.append(attr.UniqueID)
55             row.append(attr.GetPath())
56             row.append(str(attr.Template))
57             row.append(str(attr.DataReferencePlugIn))
58             row.append(attr.ConfigString)
59             row.append('')
60             row.append('')
61             row.append('')
62             attrList.append(row)
63     attrList = pd.DataFrame(attrList[1:], columns=attrList[0])
64     elementsList = pd.DataFrame(elementsList[1:], columns=elementsList[0])

```

Figura 22 – Extração de elementos e atributos

Fonte: Arquivo Pessoal

Os “row”s com string vazia fazem referência, por exemplo, a uma propriedade que um atributo pode possuir e um elemento não. Como o caso da linha 41 o qual a propriedade “AttributeDataReference” não existe para elementos, entretanto, existe para atributos, a documentação da OSIsoft fornece todo o mapeamento de propriedades de cada classe.

Da linha 33 a 47 há a obtenção de dados seguindo as propriedades da classe AFElement os quais serão salvos em uma linha representando um elemento (OSISOFT, 2021a).

As linhas 49 à 62 adquirem os atributos dos elementos e seus respectivos dados seguindo as propriedades presente na classe AFAttribute os quais são salvos em uma linha que representa um atributo presente no Asset Framework.

“attrList” e “elementsList” são as tabelas de atributos e de elementos respectivamente os quais serão salvos juntos posteriormente no CSV de “ElementsList.CSV”.

#### 4.4.2 Extração de Informações de Templates

Os nomes das colunas foram definidos com base no estabelecido no PI Builder os quais podem ser consultados na documentação da OSIsoft. Na tabela 3 estão listados os nomes de cada coluna dos templates que serão impressos na planilha final.



As colunas foram estipuladas pela empresa de realização do estágio com base na frequência de utilização e importância dentro de uma eventual necessidade de retornar a versões anteriores no Asset Framework.

Nome da Coluna	Presente no PI Builder	Mandatário para PI Builder
ObjectType	Presente	Sim
Name	Presente	Sim
Description	Presente	Não
UniqueID	Presente	Não
Parent	Presente	Sim
BaseTemplate	Presente	Não
AttributeDefaultUOM	Presente	Não
AttributeConfigString	Presente	Não
AttributeType	Presente	Não
AnalysisOutputTime	Presente	Não
AnalysisRule	Presente	Não
AnalysisRuleConfigString	Presente	Não
AnalysisRuleVariableMapping	Presente	Não
TimeRule	Presente	Não
TimeRuleConfigString	Presente	Não
VersionModifier	Não	Não
CreationDate	Não	Não
ModifyDate	Não	Não

Tabela 3 – Colunas dos Templates

Fonte: Arquivo Pessoal

Algumas colunas possuem o mesmo significado explicado na subseção anterior. As colunas adicionais presente nos templates são:

- BaseTemplate: Template pai que irá herdar atributos e configurações como descrição e categorias por exemplo;
- AttributeDefaultUOM: UOM padrão do atributo presente no template;
- AttributeType: Tipo do atributo, por exemplo string, int32, double e etc;
- AnalysisOutputTime: Tempo de saída da análise;
- AnalysisRule: Fórmula de cálculo da análise contendo variáveis e funções;
- AnalysisRuleConfigString: Forma com que a regra de análise se comportará;
- AnalysisRuleVariableMapping: Mapeamento de uma variável em uma análise;
- TimeRule: Define a frequência que uma ação se repetirá;

- TimeRuleConfigString: Definições da regra de tempo como periódico, natural e etc.

O código referente a extração dos templates pode ser explicado pelo fluxograma da figura 23. Posteriormente será apresentado um maior detalhamento dos processos envolvidos.

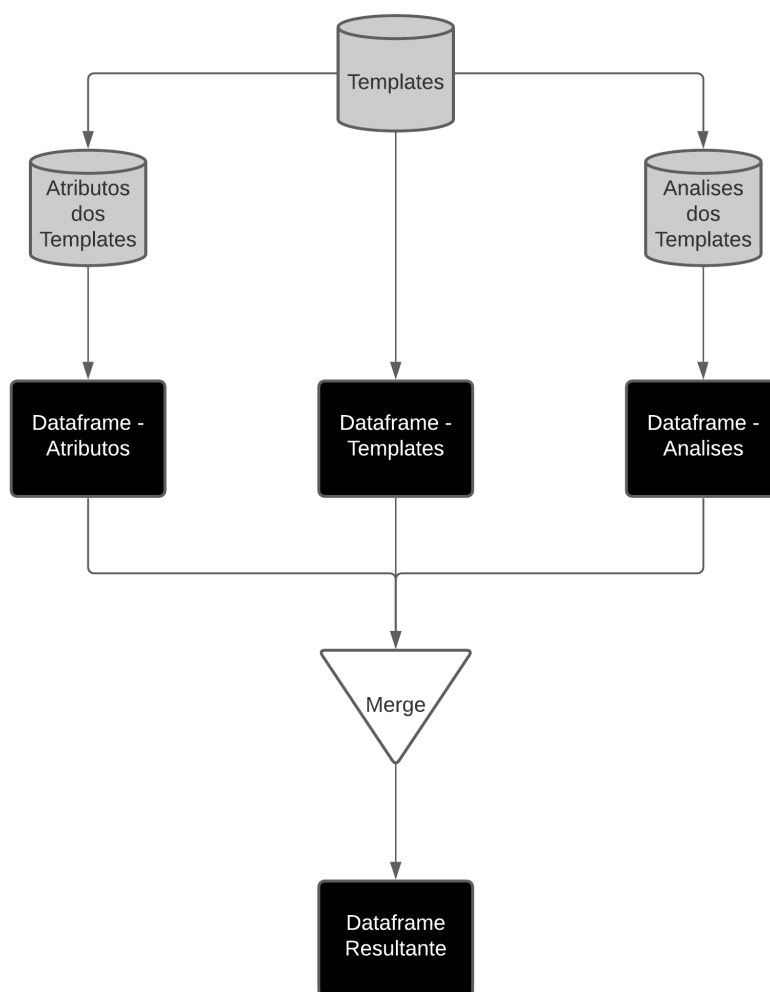


Figura 23 – Extração de templates e atributos

Fonte: Arquivo Pessoal

As atividades executadas pelo código são baseadas na mesma lógica com que foram extraídos os elementos. Portanto, seguindo a documentação fornecida pela OSIsoft, as propriedades presentes nas classes forneceram os dados necessário que foram salvos em uma variável chamada "row" e posteriormente gravada em uma lista que será mesclada no final da função presente no código.

O cilindro "Templates", Atributos dos templates e análises dos templates são respectivamente o conjunto de propriedades provenientes das classes AFElementTemplate, AFAttributeTemplate e AFAnalysisTemplate, os retângulos pretos referem se aos

dataframes criados no Python seguindo a lógica do código presente na imagem 22 nas linhas 63 e 64.

O merge para um dataframe resultante acontecem posteriormente no intuito de criar um arquivo TemplateList.CSV como um dos produtos da etapa de extração.

### 4.4.3 Criação de arquivos CSV

A biblioteca Pandas forneceu de maneira direta uma forma de salvar os dataframes criados em um arquivo CSV. A figura 24 demonstra a forma com que foi configurado no código.

```
167 elementsList = createList(DB)
168 elementsList.to_csv('ElementsList.csv',index=False)
169
170 tempList = createTempList(DB)
171 tempList.to_csv('TemplateList.csv',index=False)
172 txt.close()
```

Figura 24 – Criação de arquivo CSV

Fonte: Arquivo Pessoal

A função createList() presente na linha 167 do código 24 refere se ao código presente na subseção de Extração de Informações de Elementos e retorna um dataframe (com os elementos e seus respectivos atributos) o qual será convertido em CSV (ElementsList.CSV).

A linha 170 possui a função de criar um dataframe, por meio do createTempList referente a subseção de Extração de Informações de Templates, o qual será criado um arquivo CSV com todos os templates, análises de template e atributos de template.

Importante salientar que cada linha do ElementsList contém ou um elemento ou um atributo e possuirão propriedades/dados referentes a existência ou não da propriedade para o tipo citado (elementos, atributos, templates, análises de templates, atributos de templates) caso não exista o dado será uma string vazia.

## 4.5 TRANSFORMAÇÃO

O processo de transformação envolve a formatação dos dados em uma planilha Excel por meio de código VBA. Os processos efetuados pelo código estão apresentados na figura 25.

O VBA foi escolhido por se tratar de uma linguagem de programação utilizada no Excel para criação de macros. O Excel foi definido como requisito não funcional do software sendo obrigatório e estipulado pela empresa.

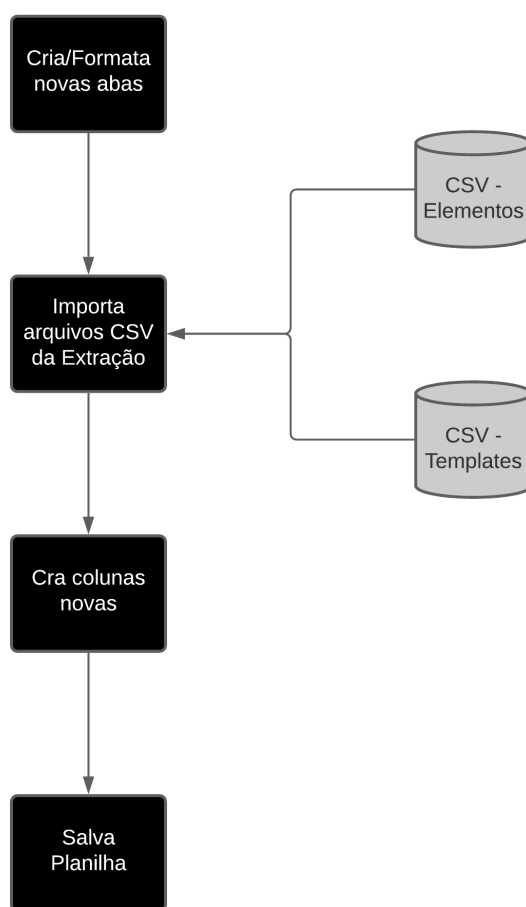


Figura 25 – Fluxo do código de transformação

Fonte: Arquivo Pessoal

O resultado esperado é uma planilha chamada LogSheet a qual trata de unificar todos os dados obtidos na etapa de extração e fornece visualização via Excel para os usuários.

#### 4.5.1 Criação de abas e configurações no Excel

As abas são configuradas a fim de facilitar a distinção pelo usuário dos dados. A figura 26 demonstra o fluxograma simples para criação de abas do código VBA.

Os dados referentes ao mês e dia são extraídos a partir de função do VBA que acessa a data atual do sistema, posteriormente o código irá criar uma aba com um "E" referente a elementos e outra "T" para os template as quais ambas possuirão o mês e dia de utilização do software.

As cores das abas foram configuradas a fim de diferenciar os tipos de dados presentes. A cor azul foi escolhida para a aba dos elementos e a cor verde para os templates.

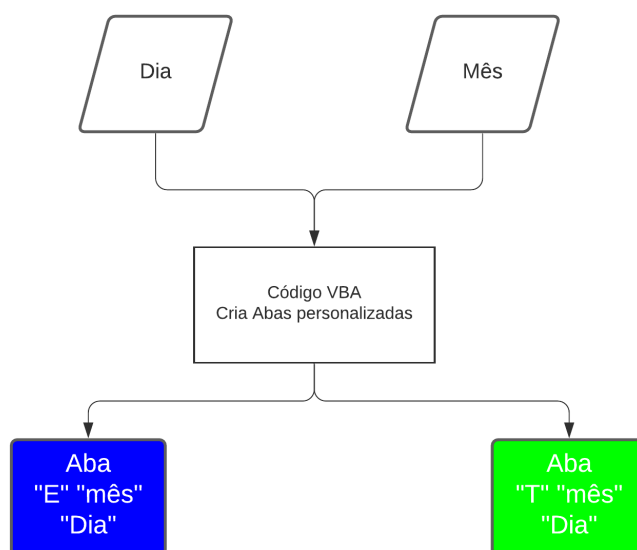


Figura 26 – Criação de abas

Fonte: Arquivo Pessoal

#### 4.5.2 Leitura e carregamento de arquivos CSV

Para a leitura e carregamento de arquivos CSV fornecidas pela etapa de Extração foi utilizado o código da figura 27. As funções utilizadas ao longo do desenvolvimento deste módulo serão detalhadas abaixo.

```
Sub ImportCSV()  
  
    mDate = Date  
    mDay = Day(mDate)  
    mMonth = MonthName(Month(mDate), True)  
    Name1 = "E " & mMonth & " " & mDay  
    Name2 = "T " & mMonth & " " & mDay  
    consultaArquivo "ElementsList.csv", Name1  
    consultaArquivo "TemplateList.csv", Name2  
    updateModificationDate  
    'MsgBox ("CSV imported!")  
  
End Sub
```

Figura 27 – Importação de CSVs

Fonte: Arquivo Pessoal

- `consultaArquivo`: Trata de adicionar uma conexão com arquivo CSV e importa os dados a partir da célula "A1" da planilha;
- `updateModificationDate`: Atualiza a data de modificação presente na aba de "Update Control" a qual possui a data da última alteração nos CSVs.

### 4.5.3 Criação de colunas com status das planilhas

A criação de colunas na planilha LogSheet com status (Modificado, não modificado e excluído) possui o intuito de instruir o usuário diretamente aos elementos e templates modificados e excluídos a fim de reduzir o tempo de procura por modificações do Asset Framework.

A ausência desta coluna faria o operador do software comparar abas linha a linha para encontrar modificações realizadas por algum outro operador no Asset Framework. O código utilizado segue a lógica da figura 28.

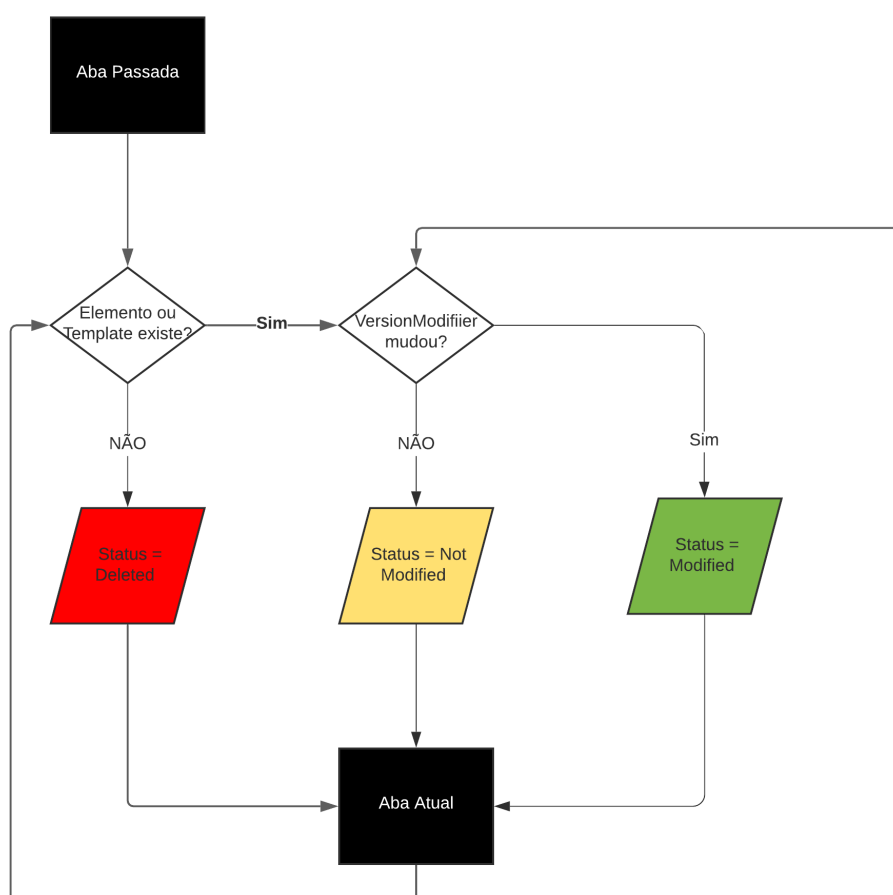


Figura 28 – Definição de Status dos Elementos/Templates

Fonte: Arquivo Pessoal

Inicialmente há a presença da aba passada na planilha a qual será comparada com a atual sendo processada, caso a linha (pode representar um elemento ou template) possua os mesmos campos pode se comparar se a data de modificação presente no VersionModifier são iguais, caso o contrário a célula de status da aba atual receberá "Deleted".

Posteriormente, caso exista a mesma linha em ambas as abas, compara se a coluna de VersionModifier caso a data seja igual em ambas receberá o status de "Not

Modified” caso o contrário receberá ”modified”.

Além da configuração da coluna Status da planilha esta etapa do software irá:

- Adicionar filtros a todas as colunas;
- Definir configurações das células;
- Atribuir cores aos status e ordenar pela data de modificação.

#### 4.6 EXECUÇÃO

A etapa de Execução possui o intuito de unificar o código desenvolvido pela Extração e o da Transformação por meio de arquivo .BAT e VBS.

O código consiste em demandar a tarefa de executar o ”main.exe”, ”auto\_excel.vbs” e posteriormente imprimir uma mensagem indicando que as tarefas foram realizadas com sucesso. Os arquivos citados serão detalhados abaixo:

- main.exe: Executável criado pela biblioteca Pyinstaller, trata apenas de uma conversão do código desenvolvido em Python para executável permitindo a ausência de instalação de qualquer componente ao computador;
- auto\_excel.vbs: Script o qual irá executar a macro desenvolvida em VBA.

##### 4.6.1 Criação de arquivo VBS e BAT

No intuito de detalhar a etapa de Execução foi desenvolvido o diagrama de componentes da imagem 29 a qual menciona quando será utilizado cada arquivo.

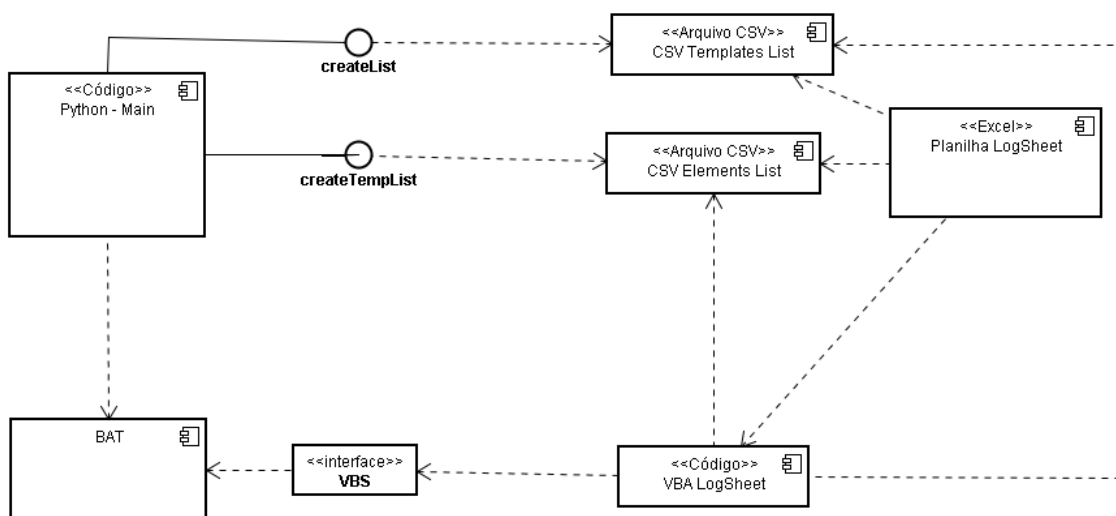


Figura 29 – Execução do software

Fonte: Arquivo Pessoal

Inicialmente o usuário irá clicar no arquivo BAT o qual executará de forma serial cada arquivo. Após a execução do main.exe ser concluída o prompt de comando executará o script auto\_excel.vbs o qual acionará a macro (código VBA) da planilha LogSheet fornecendo ao usuário a visualização por meio do Excel da versão do dia atual acessado pelo usuário e de versões antigas salvas.



## 5 RESULTADOS

Este capítulo tratará dos resultados obtidos bem como os testes realizados seguindo a implementação, análise dos resultados em que serão apresentados pontos positivos, negativos e ganhos obtidos.

Para apresentação dos resultados relacionados a cada etapa o capítulo será separado pelas seções de extração, transformação e execução no intuito de demonstrar ao leitor os principais recursos criados em cada.

### 5.1 EXTRAÇÃO

A seção de extração tratará dos dois principais elementos criados pelo código realizado em Python:

- CSV de elementos;
- CSV de templates.

O objetivo das imagens 30 e 31 é demonstrar a forma com que cada arquivo está disposto no intuito de comparar com os resultados obtidos na seção de transformação.

Object Type	Name	Description	Unique ID	Path	Template	Attribute Data Reference	Attribute Config String	Version Modifier	Creation Date	Modify Date
Element	ABC123_e3e14500-22cb-11eb-896b-5cd935f2a2	Richard Barbosa(Fabrica de Cimento)(Fabrica 1)(Area 2)AB123.Trocador de calor(Fabrica de cimento)...							09/11/2020 19:04:39	09/11/2020 19:05:09
Element	Air Compressor C-3212_907c987a-13c1-11eb-ab86-00035e1300a	Richard Barbosa (Instituto)(Wilmington)(Utilities)(Water Plant)(Air Compressor C-3212,W/L-Compressor)...							21/10/2020 14:24:29	21/10/2020 14:24:29
Element	Area_1_36f956d9-08ae-11eb-8957-5cd935f2a2	Richard Barbosa(Fabrica)Area.None...							07/10/2020 12:06:31	07/10/2020 14:02:56
Element	Area_1_c0b28289-22cb-11eb-896b-5cd935f2a2	Richard Barbosa(Fabrica de Cimento)(Fabrica 1)(Area 1)None...							09/11/2020 19:04:39	09/11/2020 19:04:39
Element	Area_2_1_80c1931e-22cb-11eb-896b-5cd935f2a2	Richard Barbosa(Fabrica de Cimento)(Fabrica 1)(Area 2)None...							09/11/2020 19:04:39	09/11/2020 19:04:39
Element	Area_3_c1f1b51-22cb-11eb-896b-5cd935f2a2	Richard Barbosa(Fabrica de Cimento)(Fabrica 1)(Area 3)None...							09/11/2020 19:04:39	09/11/2020 19:04:39
Element	Area de processo 1_ab0ac205-3887-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 1)None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Area de processo 2_b727c264-3887-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 2)None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Area de processo 3_b727c267-3887-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 3)None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	BLR-1_4129539-0fe4-11eb-8956-5cd935f2a2	Richard Barbosa(Research Lab)(zone1)BLR-1.Blower Template...							16/10/2020 17:01:05	16/10/2020 17:16:09
Element	BLR-2_6c444d15-0fec-11eb-8956-5cd935f2a2	Richard Barbosa(Research Lab)(zone2)BLR-2.Blower Template...							16/10/2020 17:16:09	16/10/2020 17:19:17
Element	BLR-3_6c444d1c-0fec-11eb-8956-5cd935f2a2	Richard Barbosa(Research Lab)(zone3)BLR-3.Blower Template...							16/10/2020 17:16:09	16/10/2020 17:19:17
Element	Bomba1_5bce833-08af-11eb-8957-5cd935f2a2	Richard Barbosa(Fabrica)(Processo)Bomba1,Bomba...							07/10/2020 12:13:37	07/10/2020 17:31:01
Element	Bomba2_0b0777c-08de-11eb-8957-5cd935f2a2	Richard Barbosa(Fabrica)(Area)(Processo)Bomba2,Bomba...							07/10/2020 17:46:19	08/10/2020 12:55:39
Element	Bomba3_0b0777c-08de-11eb-8957-5cd935f2a2	Richard Barbosa(Fabrica)(Area)(Processo)Bomba3,Bomba...							07/10/2020 17:46:19	23/11/2020 12:10:06
Element	Bomba4_60aab8f1-2dfc-11eb-a3e0-e86f38e088c1	Richard Barbosa(Fabrica)(Area)(Processo)Bomba4,Bomba...							23/11/2020 11:59:07	23/11/2020 11:59:07
Element	CHR-1_4c2063c3-0fe4-11eb-8956-5cd935f2a2	Richard Barbosa(Research Lab)(zone1)CHR-1.Chiller Template...							16/10/2020 17:01:05	16/10/2020 17:16:09
Element	CHR-2_6c444d15-0fec-11eb-8956-5cd935f2a2	Richard Barbosa(Research Lab)(zone2)CHR-2.Chiller Template...							16/10/2020 17:16:09	16/10/2020 17:19:17
Element	CHR-3_6c444d1c-0fec-11eb-8956-5cd935f2a2	Richard Barbosa(Research Lab)(zone3)CHR-3.Chiller Template...							16/10/2020 17:16:09	16/10/2020 17:19:17
Element	Coagulate Pump_9b0b972b-0d7c-11eb-895a-5cd935f2a2	Richard Barbosa(Instituto)(Wilmington)(Utilities)(Water Plant)(River)(Coagulate Pump)None...							13/10/2020 15:00:38	13/10/2020 15:00:2
Element	Equipamento 1_1918d050-3888-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 1)(Sub area 1)Equipamento 1,None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Equipamento 2_43b41565-3888-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 1)(Sub area 2)Equipamento 2,None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Equipamento 3_4d3d193b-3888-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 1)(Sub area 3)Equipamento 3,None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Equipamento 4_57066c39-3888-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 2)(Sub area 1)Equipamento 4,None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Equipamento 5_6f2226c4-3888-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 2)(Sub area 2)Equipamento 5,None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Equipamento 6_6f2226c1-3888-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 2)(Sub area 3)Equipamento 6,None...							07/12/2020 09:34:13	07/12/2020 09:34:13
Element	Equipamento 7_65538aad-3888-11eb-a3e2-e86f38e088c2	Richard Barbosa(Planta)(Area de processo 2)(Sub area 1)Equipamento 7,None...							07/12/2020 09:34:13	07/12/2020 09:34:13

Figura 30 – Arquivo CSV de elementos

Fonte: Arquivo Pessoal

Pode se observar nas imagens 30 e 31 que as colunas definidas no capítulo de implementação estão sendo gravadas corretamente no arquivo assim como cada linha representando um elemento todos separados por vírgula seguindo o padrão CSV. No capítulo de transformação há a possibilidade de perceber

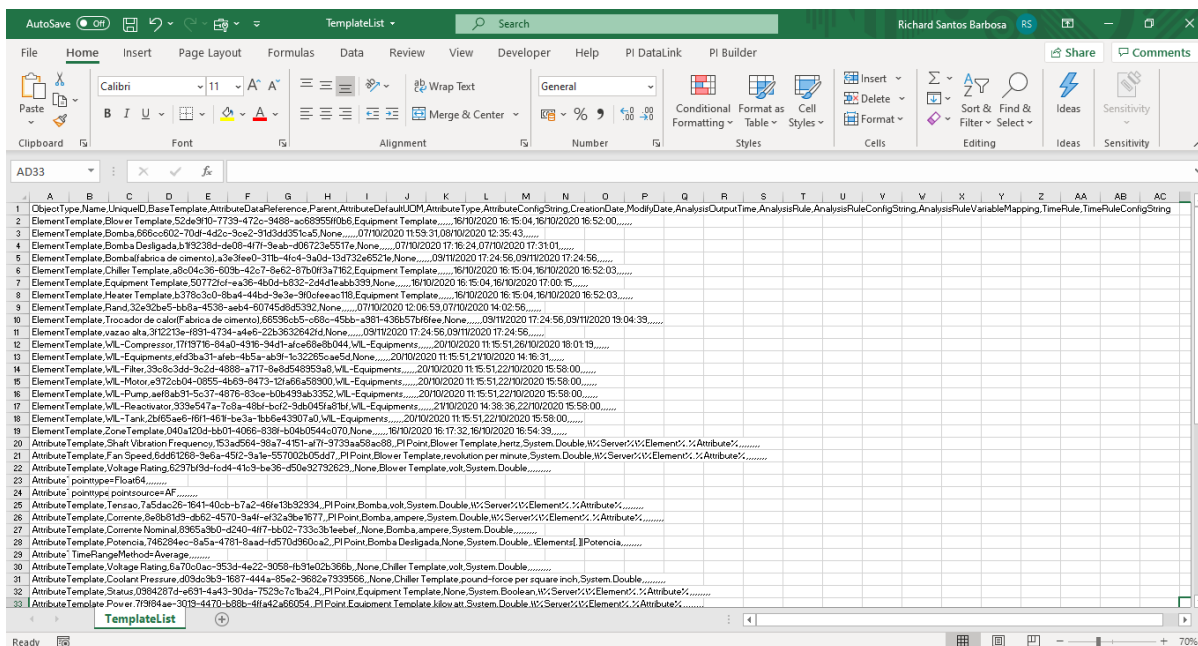


Figura 31 – Arquivo CSV de templates

Fonte: Arquivo Pessoal

## 5.2 TRANSFORMAÇÃO

A seção de transformação possui o objetivo de demonstrar os resultados obtidos em relação aos arquivos CSVs fornecidos pela etapa de extração e, portanto, evidenciar a necessidade desta etapa no contexto do projeto.

As imagens 32 e 33 representam a aba de elementos e Templates respectivamente. Pode se observar a formatação, definição de status e presença de filtros configurados devidamente como programado pelo capítulo de implementação.

## 5.3 EXECUÇÃO

A etapa de execução possui o resultado referente a ligação entre todas as etapas por meio dos arquivos VBS e BAT.

## 5.4 TESTES

Os testes foram realizados a fim de reconhecer bugs e corrigir para garantir robustez ao código.

Resultados obtidos nesta etapa serão tratados por meio de tabelas e imagens a fim de melhor ilustrar cada resultado obtido ao longo do desenvolvimento da solução discutida neste projeto e foram realizados durante o período de aproximadamente 4 meses sem erros.

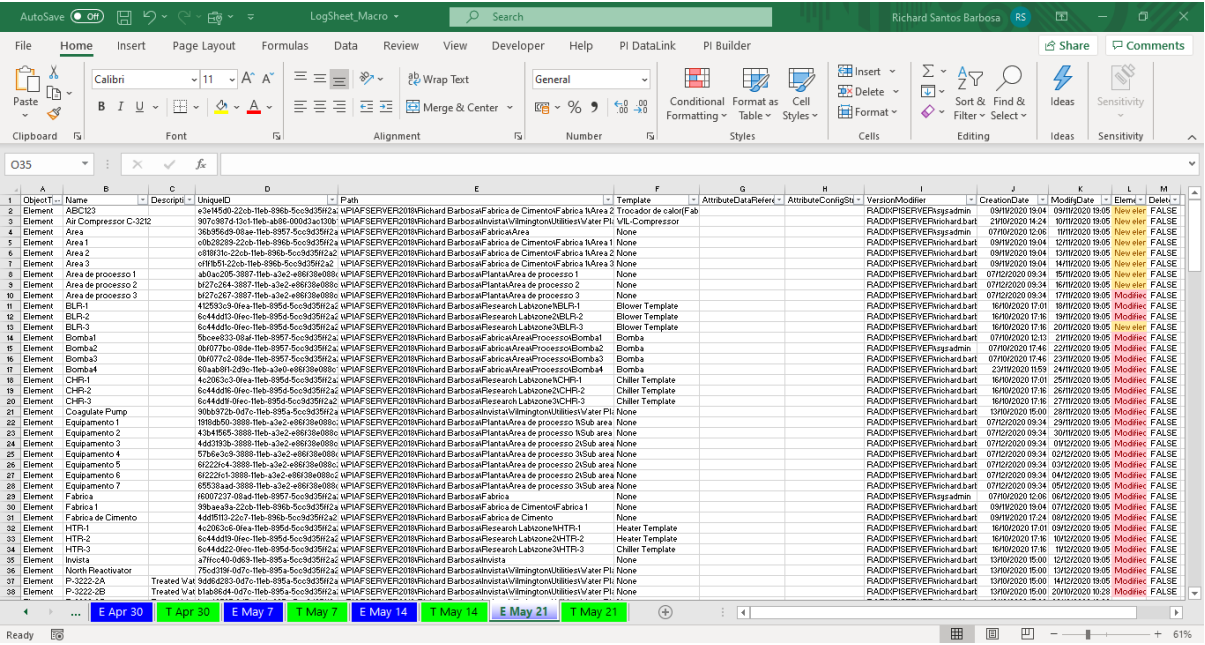


Figura 32 – Aba de Elementos

Fonte: Arquivo Pessoal

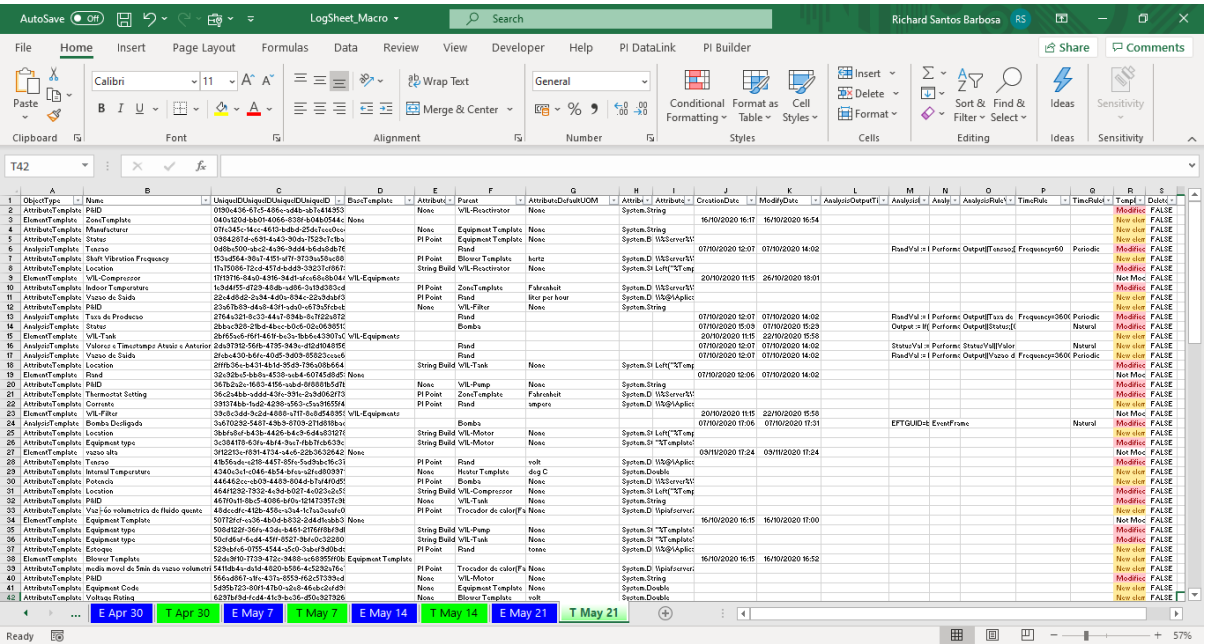


Figura 33 – Aba de Templates

Fonte: Arquivo Pessoal

Na tabela 4 há uma exposição do crescimento da planilha o qual pode ser utilizada para criação de previsões de preço, pois a OSIssoft fornece pacotes com tamanhos definidos. Logo os dados adquiridos podem ser analisados no âmbito de precificação de projetos.

Por exemplo supondo que este projeto atual tenha tido um preço de R\$500000,00

um outro projeto que necessite de 50000 elementos poderá ter uma previsão de R\$1000000,00 caso o preço siga uma função linear.

Data	Quantidade total de Elementos	Quantidade total de Templates	Tamanho LogSheet	Planilha
19/02	22787	1595	16,4 Mb	
26/02	21050	1687	18 Mb	
05/03	19557	1805	24,5 Mb	
12/03	20053	1967	28,5 Mb	
19/03	20451	2123	33,5 Mb	
26/03	21036	2357	38,5 Mb	
05/04	20743	2530	43,3 Mb	
09/04	21573	2771	48,9 Mb	
16/04	22399	3008	55 Mb	
22/04	22473	3266	61,2 Mb	
30/04	23496	3544	69,6 Mb	
07/05	25627	3839	77,4 Mb	
14/05	25060	4202	85,6 Mb	

Tabela 4 – Planilha LogSheet

Fonte: Arquivo Pessoal

As análises de dados em relação a precificação não estão no escopo deste projeto e podem servir como trabalhos futuros.

## 5.5 DISCUSSÃO

A seção de discussão tratará de analisar os resultados obtidos referente aos pontos positivos e negativos, requisitos cumpridos, além dos ganhos para a empresa em realizar as implementações.

Como requisitos da planilha no excel temos:

- Cores nas abas para distinção entre tabela elementos e tabela templates;
- Aba referente as informações de utilização da planilha;
- Coluna Modified em cada Tabela;
- Coluna Removed em cada Tabela.

A coluna modified trata de analisar com o dia anterior se o elemento ou template (dependendo da tabela) foi modificado.

A coluna Removed trata de fornecer informação se o elemento ou template foi removido.

### 5.5.1 Pontos positivos

As documentações das ferramentas OSIsoft são muito qualificadas e fornecem subsídios e treinamentos que capacitam qualquer pessoa a utilizar os componentes do PI System.

O código de geração de versões possibilitou à empresa a simplificação de um processo repetitivo a qual necessitaria de uma pessoa realizar semanalmente a extração e transformação dos dados na planilha dos logs semanais, logo, possibilitando esquecimentos por parte do responsável por esta tarefa. Alguns fatores devem ser considerados como benefícios da geração de versões automática:

- Redução da complexidade da tarefa;
- Menor quantidade de pessoas envolvidas;
- Menos tempo alocado à tarefa.

O log de dados da estrutura a ser desenvolvida no PI AF é valiosa devido aos seguintes fatores:

- Controle de eventos;
- Acompanhamento da situação de elementos e templates.

### 5.5.2 Pontos negativos

A documentação fornecida pelo cliente possui limitações de acesso a um desktop virtual com credenciais previamente fornecidas. O ambiente possui restrições de acesso como:

- Execução de arquivos executáveis;
- Acesso a determinados web sites;
- Acesso ao task scheduler;
- Instalação de softwares.

O acesso ao task scheduler seria de grande utilidade para a criação de tarefas de execução periódica. O escopo inicial do gerador de versões previa a utilização do task scheduler para automatização da criação diária de logs.

As restrições citadas geraram as limitações:

- Execução do código em Python;
- Acesso a sites como OneDrive;
- Automatização de tarefas.

## 6 CONCLUSÃO

Em suma o projeto possui uma administração muito bem feita que possibilita o andamento ágil e flexível, as pessoas envolvidas bem como as ferramentas utilizadas são de alto nível, que facilita a aquisição de resultados satisfatórios por parte do cliente.

Neste relatório houve a limitação referente à exposição de resultados devido a confiabilidade dos dados, portanto, dificuldades em relação ao tratamento de resultados foram encontradas.

Os códigos desenvolvidos atingiram resultados satisfatórios e foram construídos com auxílio de fontes especializadas a fim de gerar resultados satisfatórios. Entretanto, não foi considerado o melhor desempenho ou limpeza do código, focou-se na criação de scripts os quais funcionavam para casos específicos com capacidade de adaptação para outros.

A empresa desde o início mostrou se bastante preocupada com o estagiário fornecendo todos os componentes necessários para o trabalho remoto e capacitação para utilização das ferramentas necessárias ao projeto desenvolvido e descrito neste relatório.

A criação de códigos em Python não era obrigatória, entretanto, devido às experiências prévias com esta linguagem pelo autor, surgiram propostas pelos funcionários da empresa propondo automatizações que foram executadas com sucesso alcançando todos os objetivos propostos.

Em suma pode se citar um grande desafio a realização do Projeto de Fim de Curso de forma remota devido as diversas medidas restritivas adotadas com o objetivo de reduzir a dispersão de Covid-19. Toda a comunicação foi realizada por meio da ferramenta Teams o qual possibilitou toda comunicação entre todos os integrantes da empresa.

O projeto foi desenvolvido durante o mapeamento de elementos, templates e etc para o PI System de um cliente multinacional em um ambiente de desktop virtual sob diversas limitações de usuário.

A relação principal deste projeto com o curso de graduação de Engenharia de Controle e Automação é a automação de processos e atividades gerais, pois o software de geração de versões faz com que uma tarefa de grande importância para o decorrer do mapeamento se torne mais rápida e fácil. Há relação direta com a disciplina de Metodologia para desenvolvimento de sistemas devido a construção do software, os casos de uso levantados, requisitos, diagrama de atividades etc.

### 6.1 TRABALHOS FUTUROS

Esta seção possui o intuito de expor melhorias possíveis após feedbacks recebidos pelos usuários da ferramenta, pode se citar os principais pontos abordados:

- Personalização do Excel no intuito de fornecer caixas de pesquisa e botões para aparecer o dado de forma direta;
- Extração de outros tipos de dados além dos presentes nesta versão como template e elemento;
- Reprodução gráfica da árvore presente no Asset Framework antiga no próprio Excel;
- Criação do código de extração diretamente no .NET;
- Gráfico acompanhando e mapeando a quantidade de elementos x templates criadas.

O projeto em si havia apresentava um problema desafiador em relação as diversas limitações presentes no ambiente do cliente, entretanto foram todas contornadas com êxito e eficácia.

## REFERÊNCIAS

CANGUÇU, Raphael. **O que são Requisitos Funcionais e Requisitos Não Funcionais?** [S.l.]. Disponível em:

<https://codificar.com.br/requisitos-funcionais-nao-funcionais/>. Acesso em: 28 jun. 2021.

FERNANDES, Henrique Marques. **O que é Python e pra que serve?** [S.l.].

Disponível em: <https://marquesfernandes.com/desenvolvimento/o-que-e-python-e-pra-que-serve/>. Acesso em: 28 jun. 2021.

GIT. **Documentation.** [S.l.]. Disponível em: <https://git-scm.com/>. Acesso em: 28 jun. 2021.

MACÊDO, Diego. **Banco de Dados Temporais.** [S.l.]. Disponível em:

<https://www.diegomacedo.com.br/banco-de-dados-temporais/>. Acesso em: 28 jun. 2021.

MICROSOFT. **Git version control in VS Code.** [S.l.]. Disponível em:

<https://code.visualstudio.com/docs/introvideos/versioncontrol>. Acesso em: 28 jun. 2021.

MICROSOFT. **Jupyter Notebooks in VS Code.** [S.l.]. Disponível em:

<https://code.visualstudio.com/docs/datascience/jupyter-notebooks>. Acesso em: 28 jun. 2021.

MICROSOFT. **Microsoft Excel.** [S.l.]. Disponível em:

<https://www.microsoft.com/pt-br/microsoft-365/excel>. Acesso em: 28 jun. 2021.

MICROSOFT. **Usar a janela Propriedades.** [S.l.]. Disponível em:

<https://docs.microsoft.com/pt-br/office/vba/language/reference/user-interface-help/use-the-properties-window>. Acesso em: 28 jun. 2021.

ORACLE. **O Que É um Banco de Dados Relacional.** [S.l.]. Disponível em:

<https://www.oracle.com/br/database/what-is-a-relational-database/>. Acesso em: 28 jun. 2021.



OSISOFT. **AFElement Class**. [S./]. Disponível em:

[https://docs.osisoft.com/bundle/af-sdk/page/html/T\\_OSIssoft\\_AF\\_Asset\\_AFElement.htm](https://docs.osisoft.com/bundle/af-sdk/page/html/T_OSIssoft_AF_Asset_AFElement.htm). Acesso em: 28 jun. 2021.

OSISOFT. **Element columns**. [S./]. Disponível em:

[https://livelibrary.osisoft.com/LiveLibrary/content/en/server-v9/GUID-FD631660-A907-40A2-82D0-143DA1B1F054#addHistory=true&filename=GUID-9F0FA41A-148F-4221-A402-1D59B960DABD.xml&docid=GUID-FD631660-A907-40A2-82D0-143DA1B1F054&inner\\_id=&tid=&query=&scope=&resource=&toc=false&eventType=lcContent.loadDocGUID-FD631660-A907-40A2-82D0-143DA1B1F054](https://livelibrary.osisoft.com/LiveLibrary/content/en/server-v9/GUID-FD631660-A907-40A2-82D0-143DA1B1F054#addHistory=true&filename=GUID-9F0FA41A-148F-4221-A402-1D59B960DABD.xml&docid=GUID-FD631660-A907-40A2-82D0-143DA1B1F054&inner_id=&tid=&query=&scope=&resource=&toc=false&eventType=lcContent.loadDocGUID-FD631660-A907-40A2-82D0-143DA1B1F054). Acesso em: 28 jun. 2021.

OSISOFT. **PI Server**. [S./]. Disponível em:

<https://www.osisoft.pt/pi-system/pi-core/pi-server>. Acesso em: 28 jun. 2021.

OSISOFT. **PI System**. [S./]. Disponível em: <https://www.osisoft.pt/pi-system/>.

Acesso em: 28 jun. 2021.

OSISOFT. **PI System Hierarchy**. [S./]. Disponível em:

<https://docs.osisoft.com/bundle/af-sdk/page/html/pi-system-hierarchy.htm>. Acesso em: 28 jun. 2021.

OSISOFT. **PI System Management Tools (SMT)**. [S./: s.n.]. Disponível em:

[https://livelibrary.osisoft.com/LiveLibrary/content/en/server-v3/GUID-6588E99E-B3D9-4394-9B33-D2659FEA0276#addHistory=true&filename=GUID-F3A0947A-6E3F-4618-93DA-E9E40AF4122D.xml&docid=GUID-6588E99E-B3D9-4394-9B33-D2659FEA0276&inner\\_id=&tid=&query=&scope=&resource=&toc=false&eventType=lcContent.loadDocGUID-6588E99E-B3D9-4394-9B33-D2659FEA0276](https://livelibrary.osisoft.com/LiveLibrary/content/en/server-v3/GUID-6588E99E-B3D9-4394-9B33-D2659FEA0276#addHistory=true&filename=GUID-F3A0947A-6E3F-4618-93DA-E9E40AF4122D.xml&docid=GUID-6588E99E-B3D9-4394-9B33-D2659FEA0276&inner_id=&tid=&query=&scope=&resource=&toc=false&eventType=lcContent.loadDocGUID-6588E99E-B3D9-4394-9B33-D2659FEA0276). Acesso em: 28 jun. 2021.

PANDAS. **Merge, join, concatenate and compare**. [S./]. Disponível em: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/merging.html#merging](https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html#merging).

Acesso em: 28 jun. 2021.

PI SQUARE. **Python 3.6 and AFSDK example (Plthon!)** [S./]. Disponível em:

<https://pisquare.osisoft.com/s/question/OD51I00004UHq9c/python-36-and-afsdk-example-pithon>. Acesso em: 28 jun. 2021.

PYTHON.NET. **Python.NET**. [S.l.]. Disponível em: <http://pythonnet.github.io/>. Acesso em: 28 jun. 2021.

RADIX ENGENHARIA E SOFTWARE S.A. **Missão, Visão e Valores**. [S.l.]. Disponível em: <https://www.radixeng.com.br/sobre#missao>. Acesso em: 28 jun. 2021.

RADIX ENGENHARIA E SOFTWARE S.A. **Radix é eleita Melhor Empresa para se Trabalhar no Brasil**. Rio de Janeiro, 2020. Disponível em: <https://www.radixeng.com.br/noticias/831/radix-e-a-melhor-empresa-para-se-trabalhar-no-rio-de-janeiro-em-2020>. Acesso em: 28 jun. 2021.

RIBEIRO, Lucas. **Introdução a Biblioteca Pandas**. [S.l.]. Disponível em: <https://medium.com/tech-grupozap/introdu%C3%A7%C3%A3o-a-biblioteca-pandas-89fa8ed4fa38>. Acesso em: 28 jun. 2021.

SANTANA, Luiz. **Modelo Relacional**. [S.l.], 2019. Disponível em: <https://consultabd.wordpress.com/2019/08/21/modelo-relacional/>. Acesso em: 28 jun. 2021.

SIMÕES, Caio. **Scrum: O que é?** [S.l.], 2017. Disponível em: <https://uvagpclass.wordpress.com/2017/09/11/scrum-o-que-e/>. Acesso em: 28 jun. 2021.

SOUSA, Silas Martins. **Desenvolvimento de um sistema PIMS com comunicação OPC e acesso por Navegador Web**. 2014. Formiga. Disponível em: <https://formiga.ifmg.edu.br/documents/2017/PublicacoesTCCsBiblioteca/EE/Monografia-TCC-Silas--Martins-Sousa-02-09-2014.pdf>. Acesso em: 28 jun. 2021.

SOUZA, Ivan de. **Banco de dados: saiba o que é, os tipos e a importância para o site da sua empresa**. [S.l.]. Disponível em: <https://rockcontent.com/br/blog/banco-de-dados/>. Acesso em: 28 jun. 2021.

VIANA, Suzana. **Por que usar Jupyter Notebook?** [S.l.]. Disponível em: <https://suzana-svm.medium.com/por-que-usar-jupyter-notebook-77d5a59b42a1>. Acesso em: 28 jun. 2021.