



**UNIVERSIDADE ESTADUAL PAULISTA  
“JÚLIO DE MESQUITA FILHO”  
Faculdade de Engenharia e Ciências de Guaratinguetá**

**JOÃO PEDRO RIBEIRO ROSA**

**Avaliação da resiliência de sistemas de cogeração de hospitais**

Guaratinguetá  
2023

**João Pedro Ribeiro Rosa**

**Avaliação da resiliência de sistemas de cogeração de hospitais**

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Mecânica da Faculdade de Engenharia e Ciências do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Mecânica.

Orientador: Prof. Dr. José Alexandre Matelli  
Coorientador: Dr. Felipe Sartori da Silva

Guaratinguetá  
2023

R788a Rosa, João Pedro Ribeiro  
Avaliação da resiliência de sistemas de cogeração de hospitais / João Pedro Ribeiro Rosa - Guaratinguetá, 2023.  
46 f : il.  
Bibliografia: f. 30-33

Trabalho de Graduação em Engenharia Mecânica –  
Universidade Estadual Paulista, Faculdade de Engenharia e  
Ciências de Guaratinguetá, 2023.  
Orientador: Prof. Dr. José Alexandre Matelli  
Coorientador: Prof. Dr. Felipe Sartori da Silva


1. Energia elétrica e calor - Cogeração. 2. Avaliação de  
riscos de saúde. 3. Monte Carlo, Método de. I. Título.

CDU 536.72

JOÃO PEDRO RIBEIRO ROSA

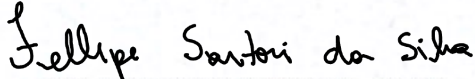
ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE  
DO REQUISITO PARA OBTENÇÃO DO DIPLOMA DE  
"GRADUADO(A) EM ENGENHARIA MECÂNICA"


APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE  
GRADUAÇÃO EM ENGENHARIA MECÂNICA

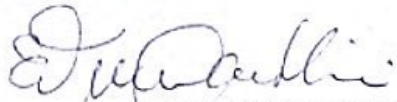
  
Prof. Dr. CELSO EDUARDO TUNA  
Coordenador(a)

BANCA EXAMINADORA:

  
Prof. Dr. JOSÉ ALEXANDRE MATELLI  
Orientador/UNESP-FEG

  
Dr. FÉLICE SARTORI DA SILVA  
Coorientador/UNESP-FEG

  
Prof. Dr. CELSO EDUARDO TUNA  
UNESP-FEG

  
Prof. Dr. ELIANA VIEIRA CANETTI  
UNESP-FEG

Janeiro de 2023

Dedico este trabalho à minha família e a todos que ajudaram na minha formação.

## AGRADECIMENTOS

Gostaria de agradecer primeiramente à minha família. Não posso expressar o quanto me ajudaram durante toda minha vida e o quanto sei que continuarão me ajudando daqui para frente. Em especial à minha mãe Patrícia que sempre foi a pessoa mais forte, gentil, compreensiva e trabalhadora que conheço. Ao meu pai Cláudio que sempre buscou tornar nossas vidas mais fácil, nos apoiou e nos deu ensinamentos para o resto da vida. Ao meu irmão Gabriel que é meu amigo e parceiro desde que me tenho por gente.

Agradeço também às pessoas que marcaram minha graduação: meus amigos da república (vai WCK!), meus amigos do unificado 151 e meus amigos da mecânica. À minha namorada Sarah que conheci durante minha graduação (e está me ajudando a ensaiar para a apresentação deste trabalho, risos), mas que levo para a vida.

Aos amigos que fiz durante minha vida de atleta. Em especial ao meu técnico Digiorgio Fais que me recebeu de braços abertos no Itaguará e proporcionou treinos dolorosos (principalmente de madrugada, risos) e ao mesmo tempo empolgantes. E ao meu técnico Zé Nilton pelos anos juntos e ensinamentos compartilhados.

Aos colegas do meu estágio/primeiro emprego. Por todos ensinamentos, risadas e ajuda que me proporcionaram. Em especial à duas pessoas excelentes(risos) da área de Comercial Excellence – Pedro e Taynná.

Por fim, mas não menos importantes, a todos os professores que tive ao longo da vida. Estes profissionais que se dedicaram e me ensinaram os mais diversos assuntos, sendo o principal: gostar de aprender. Em especial aos meus orientadores Dr. Felipe Sartori e Prof. Dr. José Alexandre Matelli por toda paciência e ensinamentos durante a realização deste trabalho.

“Agir, eis a inteligência verdadeira. Serei o que quiser. Mas tenho que querer o que for. O êxito está em ter êxito, e não em ter condições de êxito. Condições de palácio tem qualquer terra larga, mas onde estará o palácio se não o fizerem ali?”;

Fernando Pessoa

## RESUMO

Ambientes hospitalares recebem inúmeros pacientes por ano, para os mais diversos procedimentos. Sejam cirurgias, diagnósticos, ou apenas a climatização de uma sala de espera, estes ambientes necessitam de energia térmica e/ou elétrica a todo momento. Para suprir esta demanda, a tendência em grandes hospitais é a utilização das plantas de cogeração. Essas plantas, contudo, assim como todos os sistemas da engenharia, estão sujeitas a falhas inesperadas, que podem afetar seu funcionamento, pondo a vida dos pacientes em risco. Portanto, o objetivo deste trabalho é comparar a resiliência de duas plantas de cogeração que simulem a demanda energética de hospitais: eletricidade, água quente, água gelada e vapor. Para isso, foi utilizado um framework previamente proposto baseado em simulações de Monte Carlo que introduz falhas estocásticas aos sistemas para coletar dados dos seus comportamentos. Os resultados evidenciaram uma planta como a mais resiliente e proporcionaram uma análise de possíveis causas que possam justificar essa diferença e melhorias a serem implementadas.

**PALAVRAS-CHAVE:** hospital; resiliência; cogeração; energia; falhas; Monte Carlo.

## **ABSTRACT**

Hospital environments receive numerous patients per year, for the most diverse procedures. Whether surgeries, diagnoses or just the air conditioning of a waiting room, these environments require thermal and/or electrical energy at all times. To meet this demand, the trend in large hospitals is to use cogeneration plants. These plants, however, like all engineering systems, are subject to unexpected failures, which can affect their operation, putting the lives of patients at risk. Therefore, the objective of this work is to compare the resilience of two cogeneration plants that simulate the energy demand of hospitals: electricity, hot water, cold water and steam. For this, a previously proposed framework based on Monte Carlo simulations was used, which introduces stochastic failures to the systems to collect data on their behavior. The results showed a plant as the most resilient and provided an analysis of possible causes that could justify this difference and improvements to be implemented.

**KEYWORDS:** hospital; resilience; cogeneration; subject; energy; fails; Monte Carlo.

## LISTA DE FIGURAS

Figura 1 – Sistema 1- acionador primário: motor de combustão interna .....	15
Figura 2 – Sistema 2- acionador primário: turbina a gás.....	16
Figura 3 – Fluxograma do funcionamento do framework.....	18
Figura 4 – Fluxograma da propagação da falha .....	19
Figura 5– Coeficiente de variação do tempo de operação versus número de simulações....	22
Figura 6– Probabilidade de funcionamento dos sistemas versus tempo .....	23
Figura 7– Comparativo entre as entregas de energia do sistema 1 versus tempo .....	24
Figura 8- Comparativo entre as entregas de energia do sistema 2 versus tempo .....	24
Figura 9 – Comparativo de entrega do percentual médio de água gelada entre o S#1 e o S#2 versus tempo. ....	25
Figura 10– Comparativo de entrega do percentual médio de eletricidade entre o S#1 e o S#2 versus tempo. ....	26
Figura 11– Comparativo de entrega do percentual médio de vapor entre o S#1 e o S#2 versus tempo. ....	26
Figura 12– Comparativo de entrega do percentual médio de água quente entre o S#1 e o S#2 versus tempo.....	27

## LISTA DE TABELAS

Tabela 1– Área embaixo das curvas de percentual de energia específica. ....	27
-----------------------------------------------------------------------------	----

## LISTA DE ABREVIATURAS E SIGLAS

BOILER	Caldeira à gás
CHP	Planta de cogeração
CT	Torre de resfriamento
ES	Energia específica
GEN	Gerador
HEATER	Aquecedor à gás
HEX	Trocador de calor
HRSG	Caldeira de recuperação
HWAC	Sistema de refrigeração por absorção de simples efeito
MDC	Sistema de refrigeração por compressão
P	Bombas
R	Radiador
SAC	Sistema de refrigeração por absorção de duplo efeito
S#1	Sistema de cogeração 1
S#2	Sistema de cogeração 2

## LISTA DE SÍMBOLOS

$pf$	probabilidade de funcionamento do sistema
$t$	tempo de operação
$s$	energia específica
$i$	componente
$e_{s,i}$	entrega da energia específica $s$ do componente $i$
$pb$	probabilidade de falha
$pi$	probabilidade de funcionamento do componente $i$
$j$	componente a ser propagada a falha de $i$
$N$	Número de simulações
$T$	tempo de vida útil do sistema
$cv$	coeficiente de variação
$pe_s$	percentual de entrega média da energia específica $s$

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	13
<b>2</b>	<b>METODOLOGIA</b> .....	15
2.1	PLANTAS DE COGERAÇÃO .....	15
2.2	FRAMEWORK .....	16
2.3	DADOS DE ENTRADA DA SIMULAÇÃO .....	19
2.4	MÉTRICAS .....	20
<b>3</b>	<b>RESULTADOS E DISCUSSÃO</b> .....	22
3.1	COEFICIENTE DE VARIAÇÃO .....	22
3.2	ANÁLISE GRÁFICA .....	22
<b>4</b>	<b>CONCLUSÃO</b> .....	29
4.1	SUGESTÃO PARA TRABALHOS FUTUROS .....	29
	<b>REFERÊNCIAS</b> .....	30
	<b>APÊNDICE A – CÓDIGO ADAPTADO</b> .....	34

## 1 INTRODUÇÃO

Clínicas, postos de saúde e principalmente os hospitais necessitam de energia térmica e elétrica para realizar os mais diversos tipos de procedimentos nos pacientes, sejam eles simples e de rotina, como a utilização de vapor para a esterilização de equipamentos em autoclaves, ou sérios e emergenciais, como a utilização de eletricidade para o funcionamento de máquinas de bombeamento sanguíneas extracorpóreas (LEAOENERGIA, 2023).

Segundo Cappiello (2021), os hospitais e postos de saúde possuem as demandas mais intensas de energia entre as edificações, sendo o segundo setor que mais consome energia nos EUA. Para suprir esta demanda, a maioria dos grandes hospitais vem utilizando as plantas de cogeração (CHP) (TASELI et al, 2016), que são meios mais eficientes de geração de energia do que a queima de combustíveis para a geração de energia térmica e elétrica separadas.

Apesar de gerarem ambas as formas de energia, as CHPs são projetadas de forma a dar prioridade a uma delas; i.e., elas operam em paridade térmica ou elétrica. Caso as demandas de energia térmica e elétrica sejam estáveis, contínuas e estejam pareadas durante o tempo de consumo, como é o caso nos hospitais, o aproveitamento da geração de energia pode ser ainda maior (PAGLIARINI, 2019).

Além disso, por gerarem menos CO<sub>2</sub> para a atmosfera e, portanto, serem mais sustentáveis do que a queima separada, países desenvolvidos oferecem benefícios fiscais para a utilização das CHPs (DFIC, 2016). Esse fato confere a esses sistemas vantagens econômicas quando comparadas com outras formas de obtenção de energia.

O principal componente de uma CHP é o acionador primário, cuja função é converter a energia química de um combustível em energia eletromecânica (VIEIRA, 2019). Os tipos mais comuns de acionadores primários são os motores de combustão interna, turbinas a gás e turbinas a vapor. A escolha do acionador primário é feita em função das demandas energéticas, dos perfis das demandas energéticas e do regime de operação do ciclo de cogeração. Desse modo, a indicação para hospitais é a utilização de motores de combustão interna, que apresentam elevada capacidade de produção eletromecânica comparada a sua produção de energia térmica (MATELLI, 2001).

Diversos estudos buscam otimizar a utilização das CHPs e torná-las mais sustentáveis (AHMADI et al, 2010; SILVEIRA et al, 2012; HAJABDOLLAHI et al, 2017). Contudo, estes e muitos outros estudos possuem como seu maior foco a otimização dos custos da energia gerada pelo sistema, não avaliando qual é o sistema mais seguro para o paciente.

Segundo Mechtenberg et al (2020), falhas no fornecimento de eletricidade para sistemas de saúde causam a morte de 3 a 105 pacientes em cada 1000. Além disso, segundo os autores os custos ocultos (relacionados a possível perda de vida humana) associados a falta de eletricidade pode ser de 10 a 10000 vezes maior que o custo tradicional de eletricidade nesses lugares.

Em busca de maior segurança para as CHPs é utilizado o conceito de confiabilidade na fase de projeto (FRANGOPOULOS et al, 2004). Este conceito é definido como a probabilidade de o sistema operar de forma satisfatória sob condições operacionais preestabelecidas em um período de tempo específico (HAGHIFAM et al, 2011). Esta análise, entretanto, não engloba a ocorrência de eventos inesperados, como desastres naturais, que tem aumentado sua frequência a cada década (MCGREEVY, 2022).

Além do seu aumento de frequência, os desastres naturais também tem afetado um número maior pessoas e provocado perdas econômicas mais significativas com o passar do tempo (EM-DAT, 2020). Segundo Huppert (2006), as causas para estes aumentos provêm de atividades humanas como crescimento populacional, urbanização e mudanças climáticas.

Dentro deste contexto, surge a necessidade da aplicação do conceito de resiliência na fase de projeto de sistemas de cogeração de hospitais. Este conceito tem sido amplamente utilizado nas últimas décadas, nas mais diversas áreas. Na ciência dos materiais, a resiliência é uma propriedade mecânica que mede a absorção da energia do material após a aplicação de uma tensão que o deforme (THURSTON, 1874). Já a psicologia, define a resiliência como a capacidade de certas pessoas de não desenvolverem doenças psíquicas mesmo após terem passado por um trauma (DE SOUZA et al, 2006). Para este trabalho será utilizado o conceito proposto por Haimes (2009), que define a resiliência como a capacidade do sistema de se manter operacional apesar da ocorrência da falha, mesmo que esta seja oriunda de eventos inesperados.

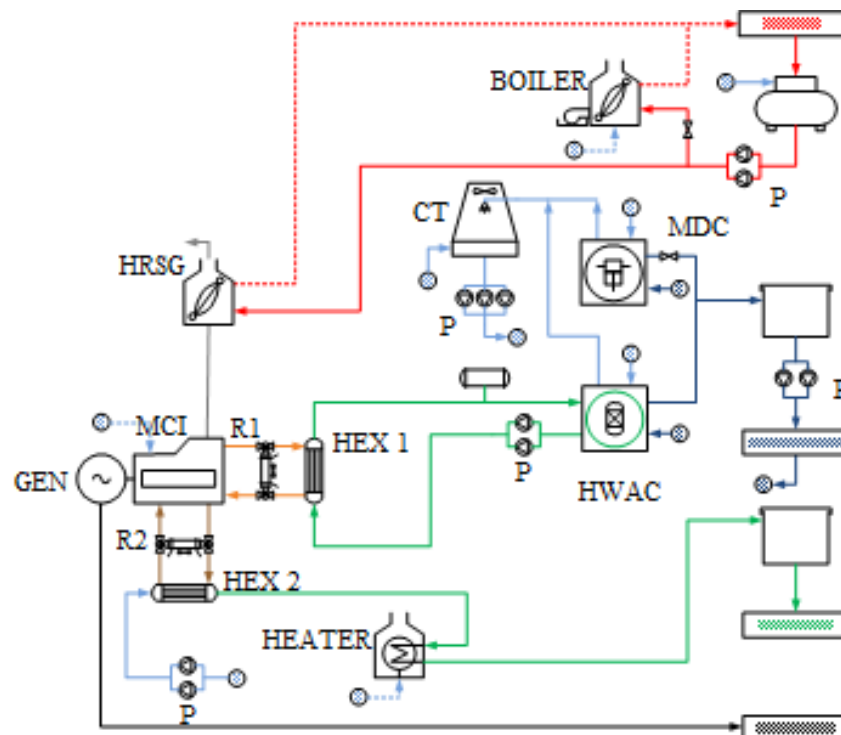
O objetivo desse trabalho é aplicar o framework desenvolvido por Matelli e Goebel (2018) para comparar a resiliência, através da análise gráfica desenvolvida por Lessa (2021), de dois sistemas de cogeração que simulem as demandas energéticas de um hospital: produção de água gelada, água quente, vapor e eletricidade. Este trabalho também tem por objetivo analisar as entregas energéticas de forma separadas afim de identificar oportunidades de melhorias nos sistemas.

## 2 METODOLOGIA

### 2.1 PLANTAS DE COGERAÇÃO

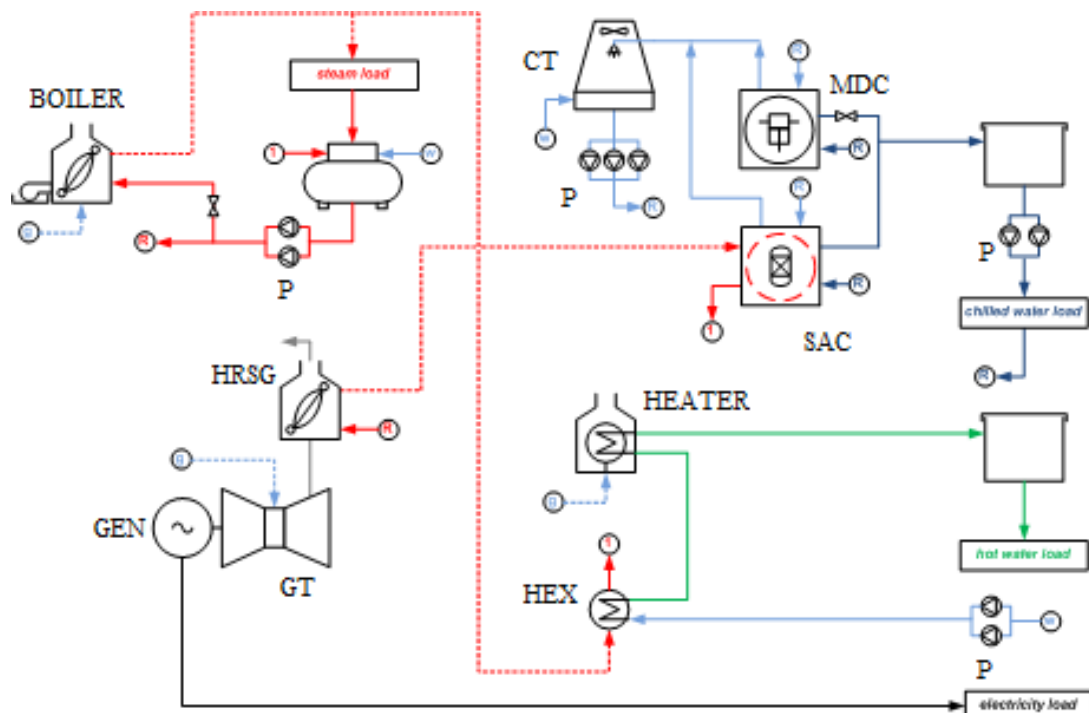
Os dois sistemas de cogeração foram obtidos através do sistema baseado em conhecimento desenvolvido em (MATELLI, 2009). Eles diferem entre si, principalmente, quanto aos acionadores primários. O primeiro sistema (S#1) utiliza um motor de combustão interna enquanto o segundo (S#2) utiliza um conjunto a gás. Os sistemas S#1 e S#2 estão ilustrados nas Figuras 1 e 2 abaixo.

Figura 1 – Sistema 1- acionador primário: motor de combustão interna



Fonte: Adaptado de Matelli (2009)

Figura 2 – Sistema 2- acionador primário: turbina a gás



Fonte: Adaptado de Matelli (2009)

Na Figura 1, o motor de combustão interna (MCI) é conectado aos radiadores (R1 e R2) para o resfriamento do óleo e do ar; à caldeira de recuperação (HRSG) para o fornecimento de vapor; e ao gerador (GEN) para o fornecimento de energia elétrica. Os radiadores são conectados aos trocadores de calor 1 e 2 (HEX 1 e HEX 2) que fornecem água quente para, respectivamente, o sistema de resfriamento por absorção de simples efeito (HWAC) e o aquecedor a gás (HEATER). O HWAC e o sistemas de refrigeração por compressão (MDC), atendem à demanda de água gelada do hospital. Por fim, a demanda de vapor é atendida pela HRSG e pela caldeira (BOILER).

Na Figura 2, o conjunto a gás (GT) é conectado à caldeira de recuperação (HRSG) para a geração de vapor e ao gerador (GEN) para eletricidade. Parte do vapor fornecido pela HRSG e pelo BOILER é utilizado para acionar o sistema de refrigeração de duplo efeito (SAC) que juntamente com o MDC fornece o suprimento de água gelada armazenado no tanque. A outra parte do vapor é fornecida para o trocador de calor (HEX) que gera água quente para o HEATER (o HEATER e o HEX são redundâncias).

## 2.2 FRAMEWORK

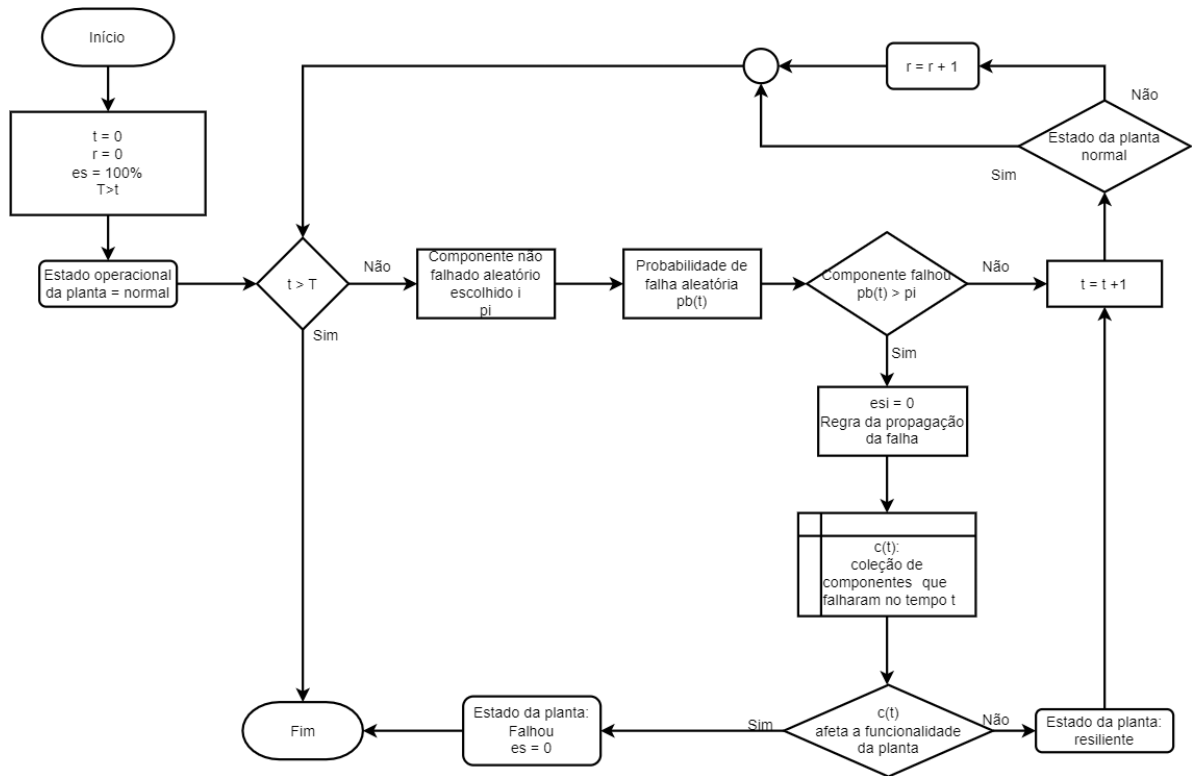
As simulações do framework foram realizadas através da adaptação do código em

Python desenvolvido em (SILVA et al, 2021) para que este também gere dados que permitissem a plotagem do gráfico da probabilidade de funcionamento (pf) versus tempo ( $t$ ) e dos gráficos da produção média de vapor, água quente, água gelada e eletricidade versus tempo.

O código foi executado em um *Jupyter Notebook* (KLUYVER, 2016) utilizando a linguagem *Python 3.9.7* e a biblioteca *Plotly* (PLOTLY TECHNOLOGIES INC, 2015) para a plotagem dos gráficos. O computador utilizado para rodar as simulações foi um Dell 11a geração i7 16GB Ram.

O funcionamento do Framework adaptado está exibido na Figura 3 abaixo. Inicialmente o sistema possui um estado de operação normal, os contadores de tempo estão zerados e a entrega da energia específica  $s$  (vapor, água quente, água gelada e eletricidade) de todos os componentes  $i$  ( $e_{s,i}$ ) está no máximo - este é um parâmetro utilizado para quantificar o percentual de energia específica (ES) produzida pelo componente. Por exemplo, em um sistema que possui duas caldeiras produzindo a mesma quantidade de vapor,  $e_{\text{vapor,caldeira}} = 50\%$ , para cada caldeira.

Figura 3 – Fluxograma do funcionamento do framework



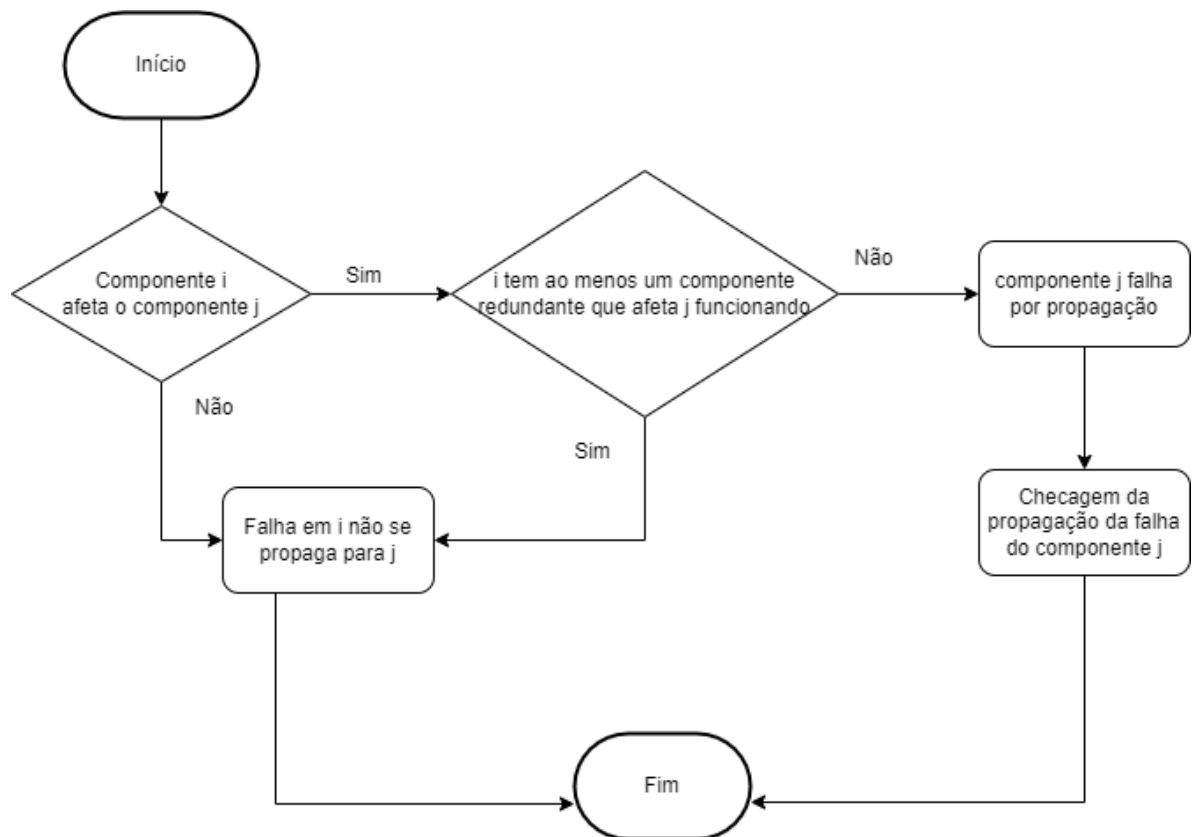
Fonte: Adaptado de Matelli e Goebel (2018)

Em seguida, uma probabilidade aleatória de falha  $pb(t)$  no tempo  $t$  é atribuída a um componente aleatório  $i$ . Se a probabilidade de falha é maior que probabilidade de funcionamento ( $pb(t) > pi$ ), o componente  $i$  falhou, o valor de  $es_i$  é zerado e a propagação da falha é avaliada. Caso contrário, um novo componente é escolhido como candidato a falha até que o tempo de operação seja atingido ( $t > T$ ).

Se uma falha comprometer a funcionalidade da planta, seu estado é alterado para falhado, todos os  $es$  são igualados a zero e a simulação termina. Caso contrário, o estado do sistema é definido como resiliente e os contadores de tempo são atualizados.

A propagação da falha está exibida no Figura 4 abaixo. Se o componente  $i$  falhou e afeta o componente  $j$  e  $i$  não possui nem um componente redundante – componente com a mesma funcionalidade – afetando  $j$ , a falha se propaga para  $j$ . Com a propagação da falha o valor de  $es_j$  é zerado.

Figura 4 – Fluxograma da propagação da falha



Fonte: Adaptado de Matelli e Goebel (2018)

Usualmente, cada incremento de tempo corresponde a 1 hora do tempo de operação do sistema, contudo, a análise da entrega de energia adicionada neste trabalho provou consumir uma quantidade de memória RAM do computador muito alta ocasionando erros de memória. Com isso, o código original desenvolvido em (SILVA, 2021), foi modificado para cada incremento de tempo corresponder a 5 horas (apenas para a análise das entregas percentuais de energia específica).

Uma vez que esta modificação afeta ambos os sistemas, e todas as entregas de energia de maneira idêntica, ela não afetará a análise comparativa entre os sistemas.

### 2.3 DADOS DE ENTRADA DA SIMULAÇÃO

O tempo de operação (T) foi definido como 8760 horas (1 ano de operação ininterrupta). A probabilidade de funcionamento foi definida como 0.9995 para todos os componentes. Conforme verificado em (MATELLI e GOEBEL, 2018), para probabilidades de

funcionamento altas ( $p_i \approx 1$ ), a diferença entre as probabilidades dos componentes não influencia na avaliação da resiliência do sistema. Uma vez que esta é uma propriedade intrínseca ao sistema e sua configuração.

Por se tratar de uma simulação de Monte Carlo, é necessário o cálculo do número de simulações mínimo para o qual os resultados converjam. Portanto, afim de medir a dispersão dos resultados, foi definido um coeficiente de variação ( $cv$ ) na equação 1 abaixo.

$$cv = \frac{\sigma}{\bar{t}} \quad (1)$$

Em que  $\sigma$  é o desvio padrão do tempo de operação do sistema e  $\bar{t}$  é o tempo de operação médio.

## 2.4 MÉTRICAS

As métricas utilizadas para a medição da resiliência e plotagem dos gráficos são descritas abaixo.

Probabilidade de funcionamento ( $pf(T)$ ): avalia o número de simulações ( $c$ ) que completaram o tempo de vida útil ( $T$ ) com um estado funcional ( $t = T$ ), dividido pelo número de simulações totais ( $N$ ). Essa métrica varia de 0 a 1. Quanto maior  $pf(T)$  maior a probabilidade de o sistema não falhar em um tempo de vida  $T$ , i.e., entregar a energia demandada completamente ou parcialmente no tempo de operação  $t$ . A equação 2 exhibe o cálculo de  $pf(T)$ .

$$pf(T) = \frac{\sum_{c=1}^N c(T)}{N} \quad (2)$$

Percentual de entrega médio de energia específica ( $pe_s(t)$ ): Este valor varia de 0 a 100 e avalia o percentual médio de entrega de cada energia específica  $s$  que é gerado pelo sistema no tempo de operação  $t$ , entre as diversas simulações  $c$  (equação 4). A entrega da energia específica é obtida somando a entrega de cada componente  $i$  no tempo  $t$  (equação 3).

$$e_s = \sum_{i=1}^{Ncpn} e_{s,i} \quad (3)$$

$$pe_s = \frac{\sum_{c=1}^N e_{s,c}}{N} \quad (4)$$

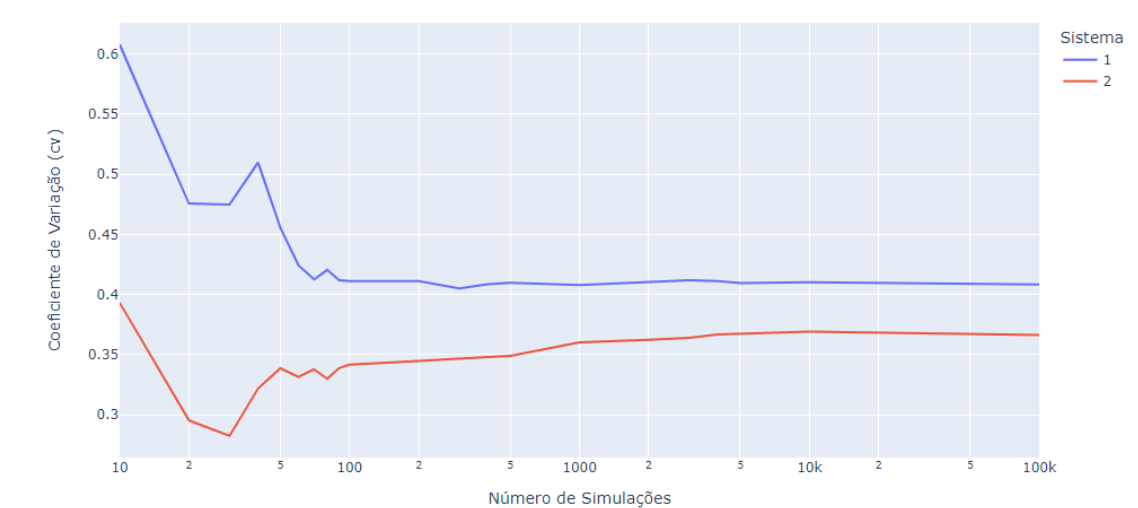
Uma vez obtida  $pe_s(t)$  foi utilizado a biblioteca Scipy(VIRTANEN,2020) para calcular a diferença das áreas abaixo da curva entre os dois sistemas através do método de Simpson e comparar as diferença entre os sistemas.

### 3 RESULTADOS E DISCUSSÃO

#### 3.1 COEFICIENTE DE VARIAÇÃO

Sob as condições propostas, o número de simulações que estabilizou o coeficiente de variação previamente estabelecido foi 5.000 simulações conforme exibido na Figura 5. Este valor é um pouco acima do obtido em (MATELLI e GOEBEL, 2018) (3000 simulações) o que evidencia a importância do cálculo deste coeficiente para cada sistema.

Figura 5– Coeficiente de variação do tempo de operação versus número de simulações



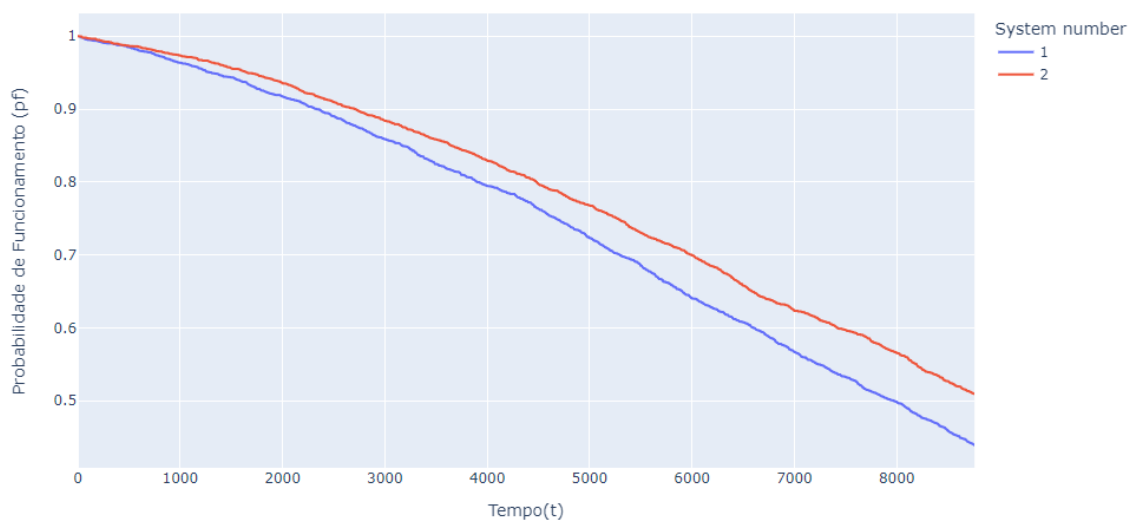
Fonte: Autoria própria (2023)

Uma vez obtido o número de simulações necessárias, os seguintes resultados puderam ser obtidos através do framework.

#### 3.2 ANÁLISE GRÁFICA

O resultado da probabilidade de funcionamento dos sistemas 1 e 2 está exibido na Figura 6.

Figura 6– Probabilidade de funcionamento dos sistemas versus tempo



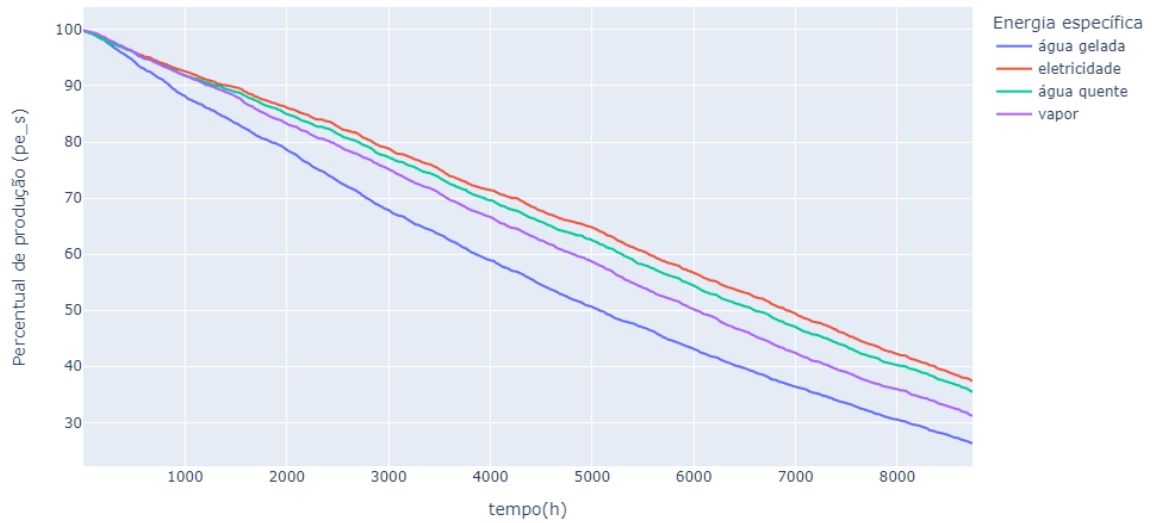
Fonte: Autoria própria (2023)

Através dela percebe-se que o S#2 é mais resiliente que o S#1, uma vez que sua probabilidade de permanecer funcionando completamente ou parcialmente é maior do que a do sistema 1.

Este resultado diverge dos resultados obtidos em (MATELLI e GOEBEL, 2018; SILVA e MATELLI, 2021; LESSA, 2021), nos quais o ciclo que utiliza o motor de combustão como acionador primário é mais resiliente do que aquele com a turbina a gás. Contudo, os sistemas utilizados neste trabalho são diferentes dos sistemas utilizados nos trabalhos previamente citados: o motor de combustão depende de 2 radiadores (não são redundâncias) para seu funcionamento e as demandas de vapor e água quente também são atendidas. Com isso, nota-se que para a avaliação da resiliência não é relevante quais componentes estão sendo utilizados. A configuração do sistema e principalmente a dependência entre componentes impactam de forma mais significativa a análise.

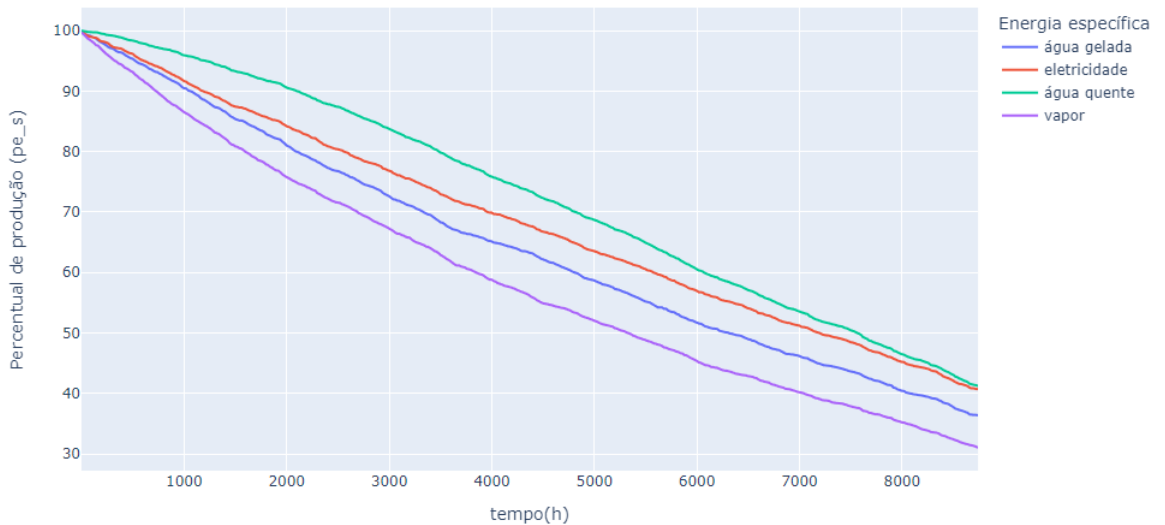
O percentual de entrega de energia para os sistemas S#1 e S#2 estão exibidos, respectivamente, nas Figuras 7 e 8 abaixo. Através da Figura 7 percebe-se que a energia específica (ES) mais propensa a falha no S#1 é a água gelada, enquanto a mais resiliente é a eletricidade. Uma explicação plausível para este resultado é a dependência dos chillers em relação a muitos componentes do sistema. O HWAC depende das bombas, do HEX 1 e da torre de resfriamento (CT); enquanto o MDC depende das bombas, da CT e da eletricidade fornecida pelo barramento (BUS).

Figura 7– Comparativo entre as entregas de energia do sistema 1 versus tempo



Fonte: Autoria própria (2023)

Figura 8- Comparativo entre as entregas de energia do sistema 2 versus tempo



Fonte: Autoria própria (2023)

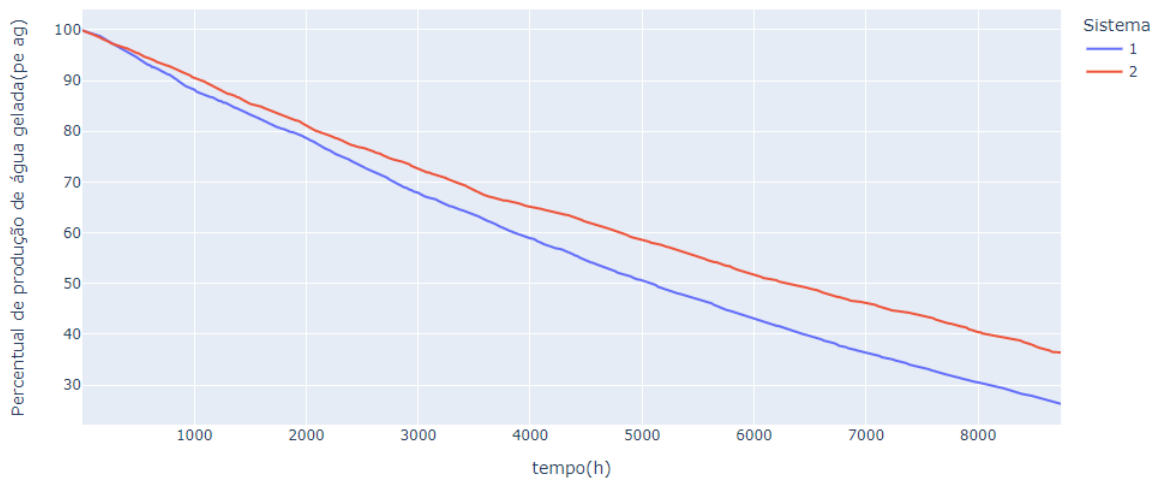
Já para o S#2, percebe-se que a produção de água quente é a entrega de energia mais resiliente, enquanto a produção de vapor é a menos resiliente. Isso pode ser explicado pela redundância do HEX com o HEATER para geração de água quente, com o HEX dependendo apenas do vapor fornecido pelo BOILER e pela HSRG.

A menor resiliência do vapor pode ser explicada pela falha de 50% da produção de vapor quando a turbina falha, além da dependência em relação às bombas e ao tanque de condensado.

Além disso, o cálculo do  $pe_{\text{vapor}}(t)$  também é desfavorecido em relação ao  $pe_{\text{água quente}}(t)$  e ao  $pe_{\text{água gelada}}(t)$  no S#2. Uma vez que caso uma das caldeiras falhe, o SAC e o HEX não terão suas produções de ES reduzidas na propagação da falha devido a presença da redundância. Contudo, em um caso real, a produção de água gelada no SAC e a produção de água quente no HEX sofreria uma redução na sua capacidade. Essa modificação não está no escopo deste trabalho, mas é sugerida para trabalhos futuros.

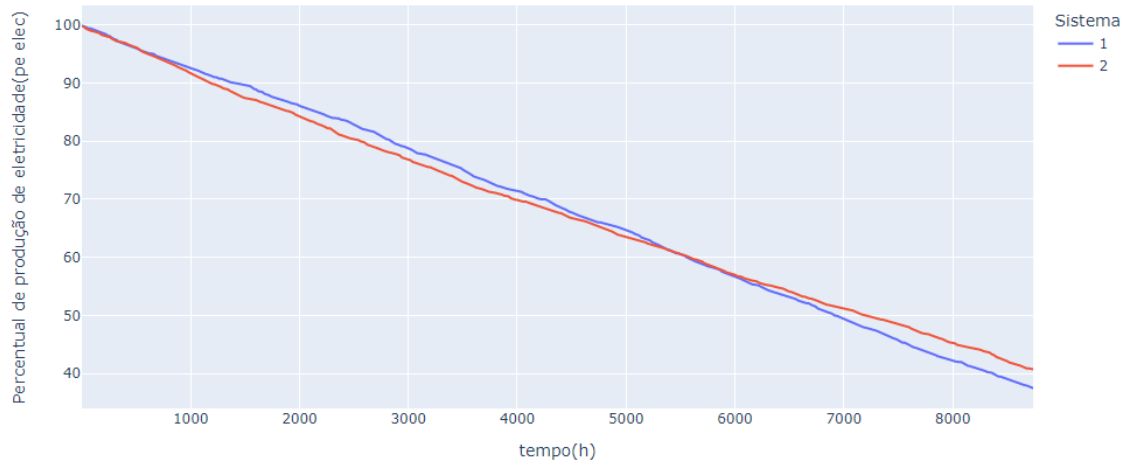
Essa “imprecisão” no cálculo afeta somente a análise comparativa de ES diferentes para um mesmo sistema. A comparação da resiliência de uma mesma ES para sistemas diferentes não é afetada, uma vez que para a medição da resiliência considera-se a entrega parcial da energia como um estado resiliente. Portanto, a comparação de cada ES para os sistemas S#1 e o S#2 está exibida nas Figuras 9 a 12 abaixo.

Figura 9 – Comparativo de entrega do percentual médio de água gelada entre o S#1 e o S#2 versus tempo.



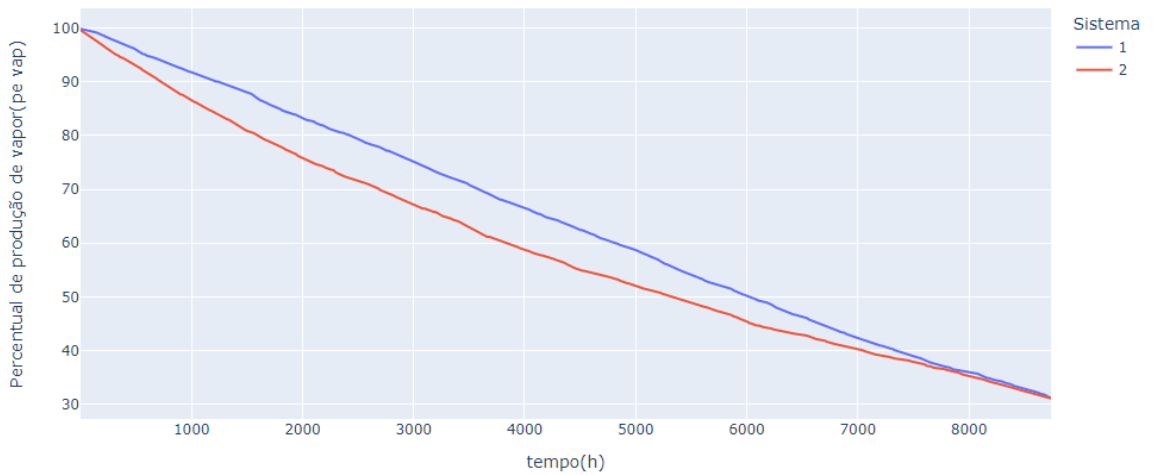
Fonte: Autoria própria (2023)

Figura 10– Comparativo de entrega do percentual médio de eletricidade entre o S#1 e o S#2 versus tempo.



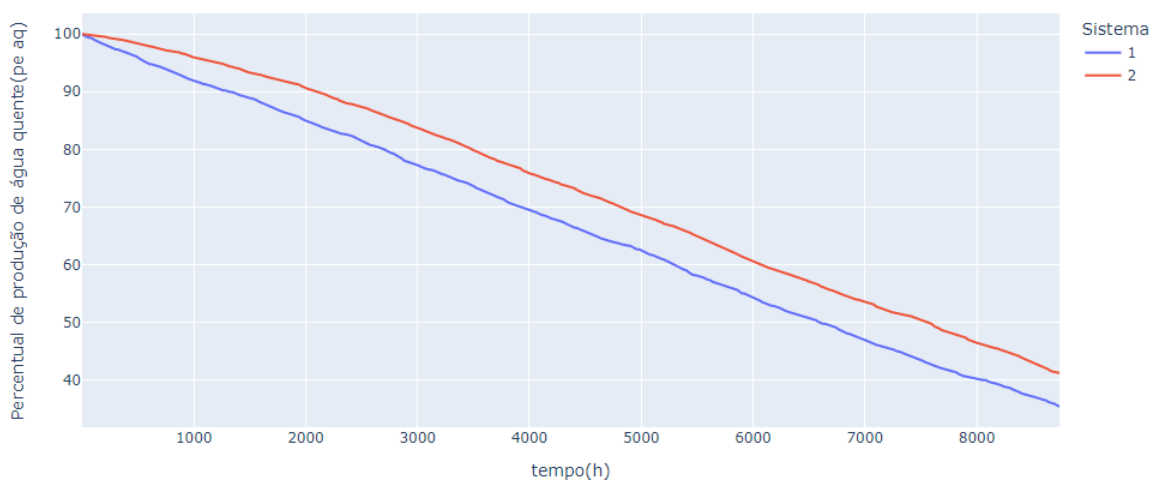
Fonte: Autoria própria (2023)

Figura 11– Comparativo de entrega do percentual médio de vapor entre o S#1 e o S#2 versus tempo.



Fonte: Autoria própria (2023)

Figura 12– Comparativo de entrega do percentual médio de água quente entre o S#1 e o S#2 versus tempo.



Fonte: Autoria própria (2023)

A partir dos gráficos foi utilizado o método de Simpson desenvolvido em (Virtanen,2020) para o cálculo da área abaixo das curvas e facilitar as comparações. Os valores obtidos estão exibidos na Tabela 1 abaixo. Em que o sinal negativo simboliza a resiliência do S#1 sendo maior que a do S#2.

Tabela 1– Área embaixo das curvas de percentual de energia específica.  
Percentual de entrega de energia específica (UA)

Sistema	1	2	
Energia específica	Diferença(S2-S1)		
água gelada	25.482	28.227	2.746
eletricidade	29.994	29.939	- 56
água quente	29.287	31.787	2.500
vapor	28.025	25.887	- 2.138

Fonte: Autoria própria (2023)

A partir das figuras e da Tabela 1, percebe-se que das diversas demandas energéticas específicas do hospital, o S#1 possui a entrega de vapor e a eletricidade mais resiliente que o S#2. Sendo a entrega de vapor a mais impactante. Uma possível melhoria para o S#2, portanto, poderia ser a inclusão de uma caldeira destinada apenas ao fornecimento de vapor para o sistema, ou a inclusão de mais um tanque de condensado.

Também fica claro que as maiores diferenças entre os sistemas estão nas curvas de entrega de água gelada e água quente. Isso pode ser explicado devido a conexão entre os tubos de vapor no S#2. Essa conexão permite que o BOILER e a HRSG sejam redundâncias

na entrega de vapor para o HEATER e para o SAC, garantindo tanto a produção de água quente quanto água gelada. Essa conexão poderia ser implementada para o S#1 no fornecimento de água quente para o HWAC e para o HEATER.

## 4 CONCLUSÃO

Este trabalho se propôs a analisar graficamente duas plantas de cogeração que simulem atender as demandas energéticas de um hospital (i.e., água quente, água gelada, vapor e eletricidade). Para isso, foi utilizado o framework desenvolvido previamente, que gerou dados do comportamento dos sistemas na presença de falhas.

Em posse dos dados, foram utilizadas duas métricas para a comparação dos sistemas. A probabilidade de funcionamento e o percentual médio de entrega de energia específica. Ambas se provaram bastante eficazes. A primeira evidenciou que o sistema 2 é mais resiliente que o sistema 1, abrindo espaço para hipóteses e futuras investigações. Já a segunda facilitou a comparação dos dois sistemas de forma mais específica, bem como seus pontos de melhoria.

Dessa forma, conclui-se que a análise da resiliência na fase de projeto de sistemas de cogeração é grande importância, sobretudo no que diz respeito a sistemas que impactam diretamente a vida humana, como é o caso dos hospitais.

### 4.1 SUGESTÃO PARA TRABALHOS FUTUROS

Sugere-se para trabalhos futuros a melhoria da métrica de entrega de energia específica para que esta reflita melhor os casos reais de propagação da falha e permita a comparação de diferentes entregas de energia para um mesmo sistema, facilitando que melhorias sejam realizadas.

Também é sugerido a inclusão das alterações nas configurações dos sistemas, bem como a avaliação do impacto dessas modificações em contrapartida a outras que possam ser tomadas. Adicionalmente, pode-se analisar o *trade off* entre o custo e a inclusão de redundâncias nos sistemas, afim de obter “o melhor dos dois mundos”.

A falta de memória foi uma dificuldade encontrada neste trabalho que também pode ser solucionada em futuras versões do código ou mediante a utilização de um computador com maior memória RAM.

## REFERÊNCIAS

AHMADI, Pouria; DINCER, Ibrahim. Exergoenvironmental analysis and optimization of a cogeneration plant system using multimodal genetic algorithm (MGA). **Energy**, Oshawa, v. 35, n. 12, p. 5161-5172, 2010. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S0360544210004251?casa\\_token=ydCw1D0R3ooAAAAA:ILly0Gn4NRe1TaAQzbpO2wuhRN5NNf47rNrvRASix4DCnHQSimvRvrHSoFLktVd9QHoc6dQaFw](https://www.sciencedirect.com/science/article/pii/S0360544210004251?casa_token=ydCw1D0R3ooAAAAA:ILly0Gn4NRe1TaAQzbpO2wuhRN5NNf47rNrvRASix4DCnHQSimvRvrHSoFLktVd9QHoc6dQaFw). Acesso em: 13 jan. 2023.

CAPPIELLO, Francesco Liberato; ERHART, Tobias Gabriel. Modular cogeneration for hospitals: A novel control strategy and optimal design. **Energy Conversion and Management**, Naples, v. 237, p. 114131, 2021. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S0196890421003071?casa\\_token=bUGXA1Ntx08AAAAA:T6FgqkwIyqFBQvjLGJNCLwyi6dUyTglu-3oJN2q2wYerGMjDCr9cVtjyh5j4EwxslbJI960\\_lg](https://www.sciencedirect.com/science/article/pii/S0196890421003071?casa_token=bUGXA1Ntx08AAAAA:T6FgqkwIyqFBQvjLGJNCLwyi6dUyTglu-3oJN2q2wYerGMjDCr9cVtjyh5j4EwxslbJI960_lg). Acesso em: 12 jan. 2023.

CENTER, Midwest CHP Application. **Combined heat & power (CHP) resource guide for hospital applications**. 2007. Disponível em:

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=bfa7f1978e5800ba9c426015c0043952da6be307>. Acesso em: 08 fev. 2023.

CENTRE FOR RESEARCH ON THE EPIDEMIOLOGY OF DISASTERS. **The International Disaster Database**. [S.l.], 2020. Disponível em: <https://www.emdat.be>. Acesso em: 08 fev. 2023.

DFIC. **Cogeneration and trigeneration**: how to produce energy efficiently. Disponível em: <https://www.dfic.de/images/pdf/cogeneration-trigeneration-guide.pdf>. Acesso em: 18 jan. 2023

FRANGOPOULOS, Christos; DIMOPOULOS, George. Effect of reliability considerations on the optimal synthesis, design and operation of a cogeneration system. **Energy**, Athens, v. 29, n. 3, p. 309-329, 2004. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S0360544202000312?casa\\_token=V2OspCvh6rkAAAAA:AhuUXWi0WFiheuMLjg3ruz-onQ0AUZSG9asxctnOJTitRUHQiZxEmpznevHlu57ReeDiUcoCA](https://www.sciencedirect.com/science/article/pii/S0360544202000312?casa_token=V2OspCvh6rkAAAAA:AhuUXWi0WFiheuMLjg3ruz-onQ0AUZSG9asxctnOJTitRUHQiZxEmpznevHlu57ReeDiUcoCA). Acesso em: 12 jan. 2023.

HAIMES, Yacov. On the definition of resilience in systems. **Risk Analysis: An International Journal**, Virginia, v. 29, n. 4, p. 498–501, 2009. Disponível em:

[https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1539-6924.2009.01216.x?casa\\_token=V30WUnlxzfsAAAAA:ahyba85VyPEW7Ax4odKyfAQm8vwJ3hVwqQwk4zf7DGjZ0HgjqO8l68Rnyu38AKq1SViUhGQWe3kQpXs](https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1539-6924.2009.01216.x?casa_token=V30WUnlxzfsAAAAA:ahyba85VyPEW7Ax4odKyfAQm8vwJ3hVwqQwk4zf7DGjZ0HgjqO8l68Rnyu38AKq1SViUhGQWe3kQpXs). Acesso em: 10 fev. 2023.

HAJABDOLLAHI, Zahra; FU, Pei-Fang. Multi-objective based configuration optimization of SOFC-GT cogeneration plant. **Applied Thermal Engineering**, Wuhan, v. 112, p. 549-559, 2017. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S1359431116324607?casa\\_token=c3AfgOx1r-UAAAAA:i4BpLjD9RY-zanM5IAto6Zh5pub0myifo63213OYzQxozOlfHOyRbLIoHw7y-ONdYCSusiky\\_A](https://www.sciencedirect.com/science/article/pii/S1359431116324607?casa_token=c3AfgOx1r-UAAAAA:i4BpLjD9RY-zanM5IAto6Zh5pub0myifo63213OYzQxozOlfHOyRbLIoHw7y-ONdYCSusiky_A). Acesso em: 12 fev. 2023.

HUPPERT, Herbert; SPARKS, Stephen. Extreme natural hazards: population growth, globalization and environmental change. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, Cambridge, v. 364, n. 1845, p. 1875-1888, 2006. Disponível em: [https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2006.1803?casa\\_token=1RxO99fPD7kAAAAA:pMBLjbSA2gv\\_Z7PzcoZLysHR5o0hjCpKffbq5J85tOob1\\_YasbTb217fvw-5\\_pSq6llGJeQCzfgZOw](https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2006.1803?casa_token=1RxO99fPD7kAAAAA:pMBLjbSA2gv_Z7PzcoZLysHR5o0hjCpKffbq5J85tOob1_YasbTb217fvw-5_pSq6llGJeQCzfgZOw). Acesso em: 12 jan. 2023.

KLUYVER, Thomas. *et al.* Jupyter Notebooks: a publishing format for reproducible computational workflows. In: KLUYVER, Thomas. *et al.* **Positioning and Power in Academic Publishing: Players, Agents and Agendas**, Amsterdam, pp. 87–90, 2016. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=Lgy3DAAAQBAJ&oi=fnd&pg=PA87&dq=KLUYVER,+Thomas.+et+al.+Jupyter+Notebooks:+a+publishing+format+for+reproducible+computational+workflows.+In+F.+Loizides+%26+B.+Schmidt,+eds.+Positioning+and+Power+in+Academic+Publishing:+Players,+Agents+and+Agendas,+Amsterdam,+pp.+87%E2%80%9390,+2016.&ots=N2FSdSoBeg&sig=Whlv1kLpHAbnugIgxX3KkqAI6RI>. Acesso em: 10 jan. 2023.

LEAOENERGIA. **Entenda qual a importância de possuir em seu hospital um gerador de energia**. Disponível em: <https://www.leaoenergia.com.br/entenda-qual-a-importancia-de-possuir-em-seu-hospital-um-gerador-de-energia/#:~:text=Cl%C3%ADnicas%2C%20postos%20de%20sa%C3%BAde%20e,s%C3%A9rios%20problemas%2C%20incluindo%20a%20morte>. Acesso em: 13 jan. 2023

LESSA, Tiago Magalhães. **Desenvolvimento de análise gráfica para a avaliação da resiliência em sistemas de energia**. 2021. Relatório final de Iniciação científica, Guaratinguetá, 2021.

MATELLI, José Alexandre. *et al.* An expert system prototype for designing natural gas cogeneration plants. **Expert Systems with Applications**, Madrid, v. 36, n. 4, p. 8375-8384, 2009. Disponível em: [https://www.sciencedirect.com/science/article/pii/S0957417408007896?casa\\_token=wKQT6GgK5ZQAAAAA:zsyTzz1YfcoiTHF0lnIrf6ufREv29s9kHDt2BEsXsIkhyKLiMJIEW6jCXgDfjUrmvUrn9syaaw](https://www.sciencedirect.com/science/article/pii/S0957417408007896?casa_token=wKQT6GgK5ZQAAAAA:zsyTzz1YfcoiTHF0lnIrf6ufREv29s9kHDt2BEsXsIkhyKLiMJIEW6jCXgDfjUrmvUrn9syaaw). Acesso em: 10 jan. 2023

MATELLI, José Alexandre. **Sistemas de cogeração baseados em células-combustível aplicados em hospitais**. 2001. Dissertação (Mestrado em Engenharia Mecânica)-Curso de Pós-graduação em Engenharia Mecânica, UFSC, Florianópolis, 2001. Disponível em: <https://repositorio.ufsc.br/handle/123456789/79470>. Acesso em: 20 jan. 2023.

MATELLI, José Alexandre; GOEBEL, Kai. Conceptual design of cogeneration plants under a resilient design perspective: resilience metrics and case study. **Applied energy**, Londres, v. 215, p. 736-750, 2018. Disponível em: [https://www.sciencedirect.com/science/article/pii/S030626191830206X?casa\\_token=V8EtUBYNltIAAAAA:RfjeWz-](https://www.sciencedirect.com/science/article/pii/S030626191830206X?casa_token=V8EtUBYNltIAAAAA:RfjeWz-)

bydRHqGxnDS6KFLSVM6C0AZqnb7fC0FQTN0Gt5REmKxzm2kZVB7AOaiyAZd4xhs  
oCvw. Acesso em: 02 jan. 2023

MCGREEVY, John Ryan; ADRIEN, Elisson. Second impact syndrome: the influence of climate change and increased disaster frequency on livelihoods and adaptive capacity in rural Haiti. **International Journal of Disaster Risk Reduction**, Georgia, p. 103509, 2022. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S2212420922007282?casa\\_token=mHU-c8g5r8AAAAA:nBXOxQOMA4WSUjWDyT8ngUCf5SRa0E\\_O5-P3K7gtz0-qPtFbvq4d4alv5Tun28FqDbGtjvUEZQ](https://www.sciencedirect.com/science/article/pii/S2212420922007282?casa_token=mHU-c8g5r8AAAAA:nBXOxQOMA4WSUjWDyT8ngUCf5SRa0E_O5-P3K7gtz0-qPtFbvq4d4alv5Tun28FqDbGtjvUEZQ). Acesso em: 10. jan 2023.

MECHTENBERG, Abigail *et al.* Health care during electricity failure: the hidden costs. **PloS one**, Notre Dame, v. 15, n. 11, p. e0235760, 2020. Disponível em:

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0235760>. Acesso em: 02 jan. 2023.

PAGLIARINI, Giorgio *et al.* Hospital CHCP system optimization assisted by TRNSYS building energy simulation tool. **Applied Thermal Engineering**, Parma, v. 44, p. 150-158, 2012. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S1359431112002311?casa\\_token=J2FLzEh4xJsAAAAA:Vt7bCK0ktq2YIbCc1mjbsKw5jP1mx\\_0hzQWucqSHjCroRRov5UsxY8HBbxLUobgfwNjYqPcQzw](https://www.sciencedirect.com/science/article/pii/S1359431112002311?casa_token=J2FLzEh4xJsAAAAA:Vt7bCK0ktq2YIbCc1mjbsKw5jP1mx_0hzQWucqSHjCroRRov5UsxY8HBbxLUobgfwNjYqPcQzw). Acesso em: 03 jan. 2023.

SILVA, Fellipe Sartori da. **Energy generation systems under unexpected operational conditions**: consequences in view of components and consideration of resilience in the design phase. Tese (Doutorado em Engenharia Mecânica)- Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2022. Disponível em: <http://hdl.handle.net/11449/217669>. Acesso em: 13 jan. 2023.

SILVA, Fellipe Sartori; MATELLI, José Alexandre. Resilience in cogeneration systems: metrics for evaluation and influence of design aspects. **Reliability Engineering & System Safety**, Guaratinguetá, v. 212, p. 107444, 2021. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S0951832021000132?casa\\_token=rOzgxi ktUusAAAAA:6P6MOOxuYdeKaT3pIT8s7SOHH72Qe25KE4q3E\\_svbdcv2xHmLcqiSO WAVzPQ9kHTPZJgnn3xcQ](https://www.sciencedirect.com/science/article/pii/S0951832021000132?casa_token=rOzgxi ktUusAAAAA:6P6MOOxuYdeKaT3pIT8s7SOHH72Qe25KE4q3E_svbdcv2xHmLcqiSO WAVzPQ9kHTPZJgnn3xcQ). Acesso em: 08 fev. 2023.

SILVA, Fellipe Sartori; MATELLI, José Alexandre. Script for resilience analysis in energy systems: Python programming code and partial associated data of four cogeneration plants. **Data in Brief**, Guaratinguetá, v. 36, p. 106986, 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352340921002705>. Acesso em: 12. jan 2023.

SILVA, Jonny Carlos. *et al.* Development of a knowledge-based system for cogeneration plant design: verification, validation and lessons learned. **Knowledge-Based Systems**, Florianópolis, v. 67, p. 230-243, 2014. Disponível em:

[https://www.sciencedirect.com/science/article/pii/S0950705114001804?casa\\_token=xDBE OhO854gAAAAA:L-77KcP8USCWTgG8fdmee-Hmfl5dUVfyT9RIO\\_w2Z-lOh--F3mSfa-j6nIn0IGPh6QkvRWENcw](https://www.sciencedirect.com/science/article/pii/S0950705114001804?casa_token=xDBE OhO854gAAAAA:L-77KcP8USCWTgG8fdmee-Hmfl5dUVfyT9RIO_w2Z-lOh--F3mSfa-j6nIn0IGPh6QkvRWENcw). Acesso em: 08 fev. 2023.

SILVEIRA, Jose Luz. *et al.* Ecological efficiency and thermoeconomic analysis of a cogeneration system at a hospital. **Renewable and Sustainable Energy Reviews**, Guaratinguetá, v. 16, n. 5, p. 2894-2906, 2012. Disponível em: [https://www.sciencedirect.com/science/article/pii/S1364032112001001?casa\\_token=9YsMpVifOPMAAAAA:T7LMZk4zBNIL3B300\\_F47aFwuXQL4UnyBzPM35bbIWcEVP7sEZ3qQ33No3AonkXj3r7H0YghGA](https://www.sciencedirect.com/science/article/pii/S1364032112001001?casa_token=9YsMpVifOPMAAAAA:T7LMZk4zBNIL3B300_F47aFwuXQL4UnyBzPM35bbIWcEVP7sEZ3qQ33No3AonkXj3r7H0YghGA). Acesso em: 05 jan. 2023.

SOUZA, Marilza Terezinha Soares; OLIVEIRA, Ceneide Maria. Resiliência psicológica: revisão da literatura e análise da produção científica. **Revista Interamericana de Psicología**, Taubaté, v. 40, p. 119-126, 2006. Disponível em: <https://www.redalyc.org/pdf/284/28440113.pdf>. Acesso em: 06 fev. 2023.

THURSTON, Robert Henry. On the strength, elasticity, ductility and resilience of materials of machine construction. **Journal of the Franklin Institute**, Reston, v. 97, n. 5, p. 344-356, 1874. Disponível em: [https://scholar.archive.org/work/odxmndv7jrbwnbv67kdq4wt7g4/access/ia\\_file/crossref-pre-1909-scholarly-works/10.1016%252F0016-0032%252873%252990699-6.zip/10.1016%252F0016-0032%252874%252990403-7.pdf](https://scholar.archive.org/work/odxmndv7jrbwnbv67kdq4wt7g4/access/ia_file/crossref-pre-1909-scholarly-works/10.1016%252F0016-0032%252873%252990699-6.zip/10.1016%252F0016-0032%252874%252990403-7.pdf). Acesso em: 04 jan. 2023.

VIEIRA, Felipe Seabra. **Desenvolvimento de sistema especialista para a geração automática de superestruturas para síntese e otimização de sistemas térmicos**. Dissertação (Mestrado em Engenharia Mecânica)- Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2022. Disponível em: <http://hdl.handle.net/11449/181356>. Acesso em: 13 jan. 2023.

VIRTANEN, Pauli. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. **Nature methods**, Austin, v. 17, n. 3, p. 261-272, 2020. Disponível em: <https://www.nature.com/articles/s41592%2E%80%900019%2E%80%900686%2E%80%902>. Acesso em: 03 jan. 2023.

## APÊNDICE A – CÓDIGO ADAPTADO

```

import pandas as pd #2.0
import time
from numpy import random
import plotly.express as px
import winsound
from scipy.integrate import simpson

## input parameters asked to the user ##
NumSys = input("\nIdentify the systems through their identification number separated by
comma\nWhich system(s) would you like to simulate?\nR: ")
sys_list = NumSys.split(",")
systems = [int(s_s) for s_s in sys_list]

Lifetime = int(input("Specify the system(s) expected lifetime (T, in hours): "))
while Lifetime <=0:
    print("\nPlease, enter a coherent number \n")
    Lifetime = int(input("Specify the system(s) expected lifetime (T, in hours): "))

Simulations = int(input("Specify the number of simulations (N): "))
while Simulations <= 0:
    print("\nPlease, enter a coherent number \n")
    Lifetime = int(input("Specify the system(s) expected lifetime (T): "))

pi = float(input("Specify the probability of component normal operation (pi): "))

initial_time = time.time() # registration of initial time to inform the total simulation time at
the end of the program

## Counters ##
Simulation_Time = 1
Resilient_Time = 0

```

```
Downtime = 0
```

```
Simulation_Counter = 1
```

```
list_SimulationTime = list()
```

```
list_ResilientTime = list ()
```

```
list_Downtime = list()
```

```
list_SimulationCounter = list()
```

```
list_DataFrames_Simulations = list()
```

```
list_specific_functions_probabilities = []
```

```
list_df_components = []
```

```
list_plot = []
```

```
length_systems = len(systems)
```

```
for System_Counter in range(0,length_systems): # a variable going through all the systems
informed by the user
```

```
    ## Making the list with the information of the current simulated system ##
```

```
    SystemNumber = systems[System_Counter]
```

```
    # IMPORTANT: the code in the next line (after the comment ends) opens the archive
with the information about
```

```
    # the system, so it is important to keep this name as the same as the name of the .xlsx file
with the system
```

```
    # information, otherwise python will not find the file and the simulation will fail.
```

```
    file = pd.read_excel(f'S#{SystemNumber}_info_new.xlsx', "Info") #Accessing the file
containing system info ("Info" refers to the worksheet name) #2.0
```

```
    system = []
```

```
    system_info = {'ID': f'system{SystemNumber}', 'OperationalState': 'normal',
'Components': int(file[f'S#{SystemNumber}'][0]),
```

```
                'FailedComponents': 0, 'RepairingComponents': 0,
```

```
                'ElectricityComponents': int(file[f'S#{SystemNumber}'][1]),
```

```
                'ThermalComponents': int(file[f'S#{SystemNumber}'][2])}
```

```
    system.append(system_info)
```

```

list_specific_functions = []
for cpn in range(0, system[0]['Components']):

    ## list of affected components obtained by auxiliary variables ##
    auxiliary1_affects = str(file['Affects'][cpn])
    if '_' in auxiliary1_affects:
        auxiliary2_affects = auxiliary1_affects.split("_")
        affects_list = [int(a_1) for a_1 in auxiliary2_affects]
    else:
        auxiliary5_affects = auxiliary1_affects.split(".")
        auxiliary4_affects = int(auxiliary5_affects[0])
        if auxiliary4_affects == 0:
            affects_list = []
        else:
            affects_list = []
            affects_list.append(auxiliary4_affects)

    ## list of redundancies obtained by auxiliary variables ##
    auxiliary1_redundant = str(file['Redundant'][cpn])
    if '_' in auxiliary1_redundant:
        auxiliary2_redundant = auxiliary1_redundant.split("_")
        redundant_list = [int(r_1) for r_1 in auxiliary2_redundant]
    else:
        auxiliary5_redundant = auxiliary1_redundant.split(".")
        auxiliary4_redundant = int(auxiliary5_redundant[0])
        if auxiliary4_redundant == 0:
            redundant_list = []
        else:
            redundant_list = []
            redundant_list.append(auxiliary4_redundant)

    ## list of components that affect cpn obtained by auxiliary variables ##
    auxiliary1_affectedby = str(file['AffectedBy'][cpn])
    if '_' in auxiliary1_affectedby:

```

```

    auxiliary2_affectedby = auxiliary1_affectedby.split("_")
    affectedby_list = [int(r_1) for r_1 in auxiliary2_affectedby]
else:
    auxiliary5_affectedby = auxiliary1_affectedby.split(".")
    auxiliary4_affectedby = int(auxiliary5_affectedby[0])
    if auxiliary4_affectedby == 0:
        affectedby_list = []
    else:
        affectedby_list = []
        affectedby_list.append(auxiliary4_affectedby)

comp_info = {'ID': int(file['C#'][cpn]), 'Type': str(file['Type'][cpn]),
            'Affects': affects_list, 'Redundant': redundant_list, 'AffectedBy':
affectedby_list,
            'OperationalState': 'normal', 'Function': str(file['Function'][cpn]), 'Repairing':
'no',
            'AverageRepairingTime': int(file['AverageRepairTime'][cpn]),
'RepairingTime': 0, 'TimeSpent': 0,
'SpecificFunction':(file['SpecificFunction'][cpn]),'PercentageProduction':(file['PercentagePr
oduction'][cpn])}
    system.append(comp_info)

# Beginning of the simulation #

while Simulation_Counter <= Simulations: # while the counter of number of simulations
does not reach the total number of simulations given by the user
    while Simulation_Time <= Lifetime: # while the counter of lifetime does not reach the
expected lifetime given by the user

        ## Component randomly choice ##
        if system[0]['OperationalState'] != 'failed':
            comp = random.randint(1, system[0]['Components'] + 1)
            check_loop=0

```

```

while system[comp]['OperationalState'] != 'normal':
    comp = random.randint(1, system[0]['Components'] + 1)
    check_loop+=1
    if check_loop> 5*system[0]['Components']:
        print('Abortar Simulação!')
        break
## Check failure and repair probabilities ##
upper_limit = 10000
pb = random.randint(0, upper_limit + 1) / upper_limit
if pb > pi:
    system[comp]['OperationalState'] = 'failedO'
    system[comp]['PercentageProduction'] = 0
    system[0]['FailedComponents'] = system[0]['FailedComponents'] + 1

## Check failure propagation ##
nowfailed = list()
nowfailed.append(comp)
if len(system[comp]['Affects']) > 0:
    for auxiliary2 in range (0, len(system[comp]['Affects'])):
        affected_component = system[comp]['Affects'][auxiliary2]
        if system[affected_component]['OperationalState'] == 'normal':
            system[affected_component]['OperationalState'] = 'failedP'
            temporary_percentage =
system[affected_component]['PercentageProduction']
            system[affected_component]['PercentageProduction'] = 0
            nowfailed.append(affected_component)
            system[0]['FailedComponents'] = system[0]['FailedComponents'] + 1

## Check functional redundant components (2.0) ##
if len(system[comp]['Redundant']) > 0:
    for redundant_position in range (0, len(system[comp]['Redundant'])):
        redundant = system[comp]['Redundant'][redundant_position]
        if system[redundant]['OperationalState'] == 'normal':
            if len(system[affected_component]['AffectedBy']) > 0:

```

```

        for affectedby_position in range (0,
len(system[affected_component]['AffectedBy'])):
            affectedby =
system[affected_component]['AffectedBy'][affectedby_position]
            if affectedby == redundant:
                if affected_component in nowfailed:
                    system[affected_component]['OperationalState'] =
'normal'
                    system[0]['FailedComponents'] =
system[0]['FailedComponents'] - 1
                    system[affected_component]['PercentageProduction']
= temporary_percentage
                    nowfailed.remove(affected_component)
                nowfailed.remove(comp)
                while len(nowfailed) > 0:
                    auxiliary2_nowchecking = nowfailed[0]
                    if len(system[auxiliary2_nowchecking]['Affects']) > 0:
                        for affectedcomp_position in range (0,
len(system[auxiliary2_nowchecking]['Affects'])):
                            affectedcomp =
system[auxiliary2_nowchecking]['Affects'][affectedcomp_position]
                            if system[affectedcomp]['OperationalState'] == 'normal':
                                system[affectedcomp]['OperationalState'] = 'failedP'
                                temporary_percentage =
system[affectedcomp]['PercentageProduction']
                                system[affected_component]['PercentageProduction'] = 0
                                nowfailed.append(affectedcomp)
                                system[0]['FailedComponents'] = system[0]['FailedComponents']
+ 1
                                if len(system[auxiliary2_nowchecking]['Redundant']) > 0:
                                    for redundant2_position in range (0,
len(system[auxiliary2_nowchecking]['Redundant'])):
                                        redundant2 =
system[auxiliary2_nowchecking]['Redundant'][redundant2_position]

```

```

        if system[redundant2]['OperationalState'] == 'normal':
            if len(system[affectedcomp]['AffectedBy']) > 0:
                for affectedby2_position in range (0,
len(system[affectedcomp]['AffectedBy')):
                    affectedby2 =
system[affectedcomp]['AffectedBy'][affectedby2_position]
                    if affectedby2 == redundant2:
                        if affectedcomp in nowfailed:
                            system[affectedcomp]['OperationalState'] =
'normal'
                            system[0]['FailedComponents'] =
system[0]['FailedComponents'] - 1
                            system[affectedcomp]['PercentageProduction'] =
temporary_percentage
                            nowfailed.remove(affectedcomp)
                            nowfailed.remove(auxiliary2_nowchecking)
                            nowfailed.clear()

df_components =
pd.DataFrame(system[1:]).query("SpecificFunction!='none']").groupby(['SpecificFunction'])
['PercentageProduction'].sum().reset_index()
    ## Check system operational state ##
    if system[0]['FailedComponents'] == 0:
        system[0]['OperationalState'] = 'normal'
    if system[0]['FailedComponents'] > 0:
        producers = system[0]['ThermalComponents'] +
system[0]['ElectricityComponents']
        failed_producers = 0
        for auxiliary4 in range(1, (system[0]['Components'] + 1)):
            if (system[auxiliary4]['Function'] == 'electricity' or
system[auxiliary4]['Function'] == 'thermal') and system[auxiliary4]['OperationalState'] !=
'normal':
                failed_producers = failed_producers + 1
            if failed_producers == producers:

```

```

system[0]['OperationalState'] = 'failed'
df_components['PercentageProduction'] = 0
if system[0]['RepairingComponents'] == 0:
    break
else:
    Downtime = Downtime + 1
else:
    system[0]['OperationalState'] = 'resilient'
    Resilient_Time = Resilient_Time + 1

df_components['Simulation'] = Simulation_Counter
df_components['Time'] = Simulation_Time
list_df_components.append(df_components)
## Time counter ##
Simulation_Time = Simulation_Time + 5

## Writing the information in the associated lists ##
list_SimulationTime.append(Simulation_Time-1)
list_ResilientTime.append(Resilient_Time)
list_Downtime.append(Downtime)
list_SimulationCounter.append(Simulation_Counter)

## Restarting the initial conditions to continue the simulation ##
Simulation_Time = 1
Resilient_Time = 0
Downtime = 0
system[0]['OperationalState'] = 'normal'
system[0]['FailedComponents'] = 0
system[0]['RepairingComponents'] = 0
for auxiliary5 in range(1, system[0]['Components'] + 1):
    system[auxiliary5]['OperationalState'] = 'normal'
    system[auxiliary5]['TimeSpent'] = 0
    system[auxiliary5]['Repairing'] = 'no'

```

```

system[auxiliary5]['PercentageProduction'] =
file['PercentageProduction'][auxiliary5-1]

print('Simulation:\t',Simulation_Counter)
Simulation_Counter = Simulation_Counter + 1
## Saving the opened archive ##
df = pd.DataFrame(
    index = list_SimulationCounter,
    columns = ['Simulation time', 'Resilient time', 'Downtime']
)
df.index.name = 'Simulation Counter'
df['Simulation time'] = list_SimulationTime
df['Resilient time'] = list_ResilientTime
df['Downtime'] = list_Downtime
# IMPORTANT: the next line refers to the name of the archive to be saved. The user is
free to change the name.
df.to_excel(f"[S#{SystemNumber}] pi {pi}_{Simulation_Time} hours.xlsx")

#Builds list with all simulation data to be plotted
df['System']=System_Counter+1
list_DataFrames_Simulations.append(df)

## Restarting the counter ##
full_df_components = pd.concat(list_df_components)
list_df_components.clear()
td_components_by_time =
pd.pivot_table(full_df_components,index=['SpecificFunction','Simulation'],columns=['Time'],fill_value=0,aggfunc='mean').stack().reset_index()
full_df_components = None
df_plot =
td_components_by_time.groupby(['Time','SpecificFunction'])['PercentageProduction'].mean().reset_index()

```

```

df_plot['System'] = System_Counter+1
df_plot['System'] = df_plot['System'].astype('int8')
td_components_by_time = None
list_plot.append(df_plot)
df_plot = None
Simulation_Counter = 1
system.clear()
list_SimulationTime.clear()
list_ResilientTime.clear()
list_Downtime.clear()
list_SimulationCounter.clear()

##Final DataFrame with all simulation data to be plotted
df_simulations = pd.concat(list_DataFrames_Simulations)
final_df_plot = pd.concat(list_plot)
## Calculating the total simulation time and informing the user ##
final_time = time.time()
final_time_hours = (final_time - initial_time) // 3600
final_time_minutes = ((final_time - initial_time) % 3600) // 60
final_time_seconds = ((final_time - initial_time) % 3600) % 60
if final_time_hours > 0:
    print(f"\nThe simulation ended in {final_time_hours} hours, {final_time_minutes}
minutes and {final_time_seconds:.0f} seconds ")
elif final_time_minutes > 0:
    print(f"\nThe simulation ended in {final_time_minutes} minutes and
{final_time_seconds:.0f} seconds ")
else:
    print(f"\nThe simulation ended in {final_time_seconds:.0f} seconds ")

list_pf = []
system_counter=1
for df_simulations in list_DataFrames_Simulations:
    pf_system=pd.DataFrame(range(0,Lifetime+1),columns=["Time"])

```

```

pf_system['N_Simulations_Functional'] = pf_system['Time'].apply(lambda n:
(df_simulations['Simulation time']>=n).sum())
pf_system['System number'] = system_counter
pf_system['Probability Functional State'] = pf_system['N_Simulations_Functional'] /
Simulations
list_pf.append(pf_system)
system_counter+=1

```

```

pf=pd.concat(list_pf)

```

```

pf

```

```

for system in range(1,length_systems+1):
    area_under_curve = simpson(pf.loc[pf['System number']==system,'Probability
Functional State'])#/Lifetime
    print("The area under S#{ } is { } UA.".format(system,int(area_under_curve)))

```

```

fig=px.line(pf,x='Time',y='Probability Functional State',color=pf['System number'])
fig.update_layout(title='Probability of Functional State per System')
fig.show()
with open(r"Probability_Functional_State_Chart.html", 'w') as f:
    f.write(fig.to_html(full_html=False, include_plotlyjs='cdn'))

```

```

fig =
px.line(final_df_plot.query("System==1"),x='Time',y='PercentageProduction',color='Specif
icFunction')
fig.show()
with open(r"S#1_Entrega_energia_específica.html", 'w') as f:
    f.write(fig.to_html(full_html=False, include_plotlyjs='cdn'))

```

```

fig =
px.line(final_df_plot.query("System==2"),x='Time',y='PercentageProduction',color='SpecificFunction')
fig.show()
with open(r"S#2_Entrega_energia_específica.html", 'w') as f:
    f.write(fig.to_html(full_html=False, include_plotlyjs='cdn'))

```

```

fig = px.line(final_df_plot.query("SpecificFunction=='chilled water'"),x='Time',y='PercentageProduction',color='System')
fig.show()
with open(r"Entrega_água_gelada.html", 'w') as f:
    f.write(fig.to_html(full_html=False, include_plotlyjs='cdn'))

```

```

fig = px.line(final_df_plot.query("SpecificFunction=='hot water'"),x='Time',y='PercentageProduction',color='System')
fig.show()
with open(r"Entrega_água_quente.html", 'w') as f:
    f.write(fig.to_html(full_html=False, include_plotlyjs='cdn'))

```

```

fig =
px.line(final_df_plot.query("SpecificFunction=='steam'"),x='Time',y='PercentageProduction',color='System')
fig.show()
with open(r"Entrega_vapor.html", 'w') as f:
    f.write(fig.to_html(full_html=False, include_plotlyjs='cdn'))

```

```

fig =
px.line(final_df_plot.query("SpecificFunction=='electricity'"),x='Time',y='PercentageProduction',color='System')
fig.show()
with open(r"Entrega_eletricidade.html", 'w') as f:
    f.write(fig.to_html(full_html=False, include_plotlyjs='cdn'))

```

```
UA =
final_df_plot.groupby(['System','SpecificFunction'])['PercentageProduction'].apply(simpson
).reset_index()
td_ua = pd.pivot(UA,index=['SpecificFunction'],columns=['System'])
display(td_ua)
## Warning the simulation end ##
# IMPORTANT: if the user does not want to be warned with the sound, please remove the
next line
winsound.Beep(1000, 2000)
```