



# **GUIA DE TESTE DE DESEMPENHO PARA APLICAÇÕES IoT**

**VERSÃO 3.0**

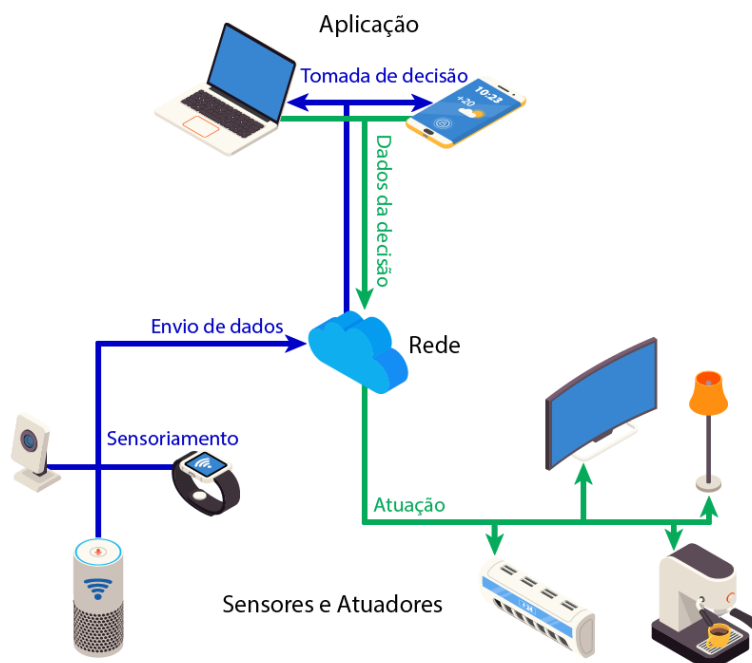
**MARÇO, 2021**

<b>GUIA DE TESTE DE DESEMPENHO PARA APLICAÇÕES IoT</b>	<b>1</b>
INTRODUÇÃO	3
Instruções para o uso do guia	4
Definições de Desempenho	5
Correlação das Características	5
Configuração do Ambiente	7
Comportamento Temporal	8
Definição	8
Contextualização	8
Casos de teste abstratos	9
Medições	13
Utilização de recursos	16
Definição	16
Contextualização	16
Casos de teste abstratos	17
Medições	21
Capacidade	26
Definição	26
Contextualização	27
Casos de teste abstratos	27
Medições	29
Impacto das Subcaracterísticas	30
Custo Benefício	33
Sugestões de Ferramentas	35
Exemplo de uso do Guia	37
Referências	39
APÊNDICE A	40
APÊNDICE B	42

# INTRODUÇÃO

Este guia visa auxiliar nos testes da Característica de Desempenho em aplicações IoT, por meio de seções como Correlação das Características, Casos de Teste Abstratos, Medições, Impacto das Subcaracterísticas, entre outras. Este guia é construído com base nas subcaracterísticas de Desempenho definidas na ISO 25010, Comportamento temporal, Utilização de Recursos e Capacidade, portanto, o guia é construído para cobrir o máximo de cenários de teste de desempenho para IoT, baseado nessas subcaracterísticas. Para cada subcaracterística são apresentadas as definições, propriedades, casos de testes abstratos e medições. Além dessas seções, há mais sete seções para executar o processo de teste dessa característica.

O Guia de Teste de Desempenho foi construído para avaliar aplicações IoT seguindo o contexto do fluxo, apresentada na Figura 0, na qual há o sensoriamento do ambiente em que os dados de sensoriamento são enviados para aplicação, onde são feitas as tomadas de decisão, e a decisão é enviada gerando uma atuação. Ou seja, o guia busca cobrir, em parte, 4 âmbitos na IoT, sensores, atuadores, aplicação e rede que faz a conexão intermediária entre todos.



**Figura 0 - Fluxo de cobertura do guia**

Os Sensores e Atuadores são definidos como objetos conectados, objetos inteligentes, à rede, portadores de sensores e atuadores. Os objetos inteligentes coletam dados, temperatura, luminosidade, som, entre outros e atuam no ambiente.

Na rede vários dados são gerados pelos sensores e através desta camada ocorre o transporte desses dados, nesta etapa é necessário um alto desempenho, para que o sistema esteja pronto para mudanças rápidas de contexto, neste aspecto começa a ficar mais claro o conceito presente na IoT de *thing-thing*, através da interação do envio de mensagens dos sensores através da rede.

Na Aplicação ocorre o processamento de todos os dados transportados, são esses os dados que chegam aos usuários. Nesta camada, todas as decisões são tomadas.

## Instruções para o uso do guia

1 - Compreender as definições de desempenho, apresentadas na seção 1, para o qual o guia foi elaborado, a fim de alinhar o conhecimento sobre a característica abordada.

2 - Com base na aplicação que se pretende testar, utilizar as definições e relações apresentadas na seção 2 de Correlação da característica para listar as características IoT relacionadas com desempenho que são prioritárias para a sua aplicação, podendo ser todas ou algumas, dependendo do seu contexto.

3 - Certificar que o ambiente para os testes de desempenho atende ao necessário apresentado na seção 3.

4 - Compreender as definições e propriedades apresentadas na seção de cada subcaracterística (seções 4, 5 e 6). A partir da compreensão, definir quais propriedades das subcaracterísticas serão avaliadas, podendo ser todas, mas não obrigatoriamente todas as subcaracterísticas do guia precisam ser avaliadas, fica a critério do usuário, a partir da compreensão da necessidade da sua aplicação.

5 - Caso no passo 4 tenha selecionado todas as propriedades pular esse passo. Com o auxílio da seção 7, de Impacto das subcaracterísticas, identificar quais propriedades não selecionadas são influenciadas pelas propriedades selecionadas, definir se essas propriedades também serão avaliadas.

5 - Selecionar quais métricas serão utilizadas. Para essa etapa deve ser utilizada a Tabela presente no Apêndice A. Na tabela é detalhado como pode ser feita a seleção das métricas baseadas nas propriedades. Não é aconselhável fazer seleção de casos de testes, é indicado a realização de todos os casos de testes, a não ser quando serão avaliadas apenas uma ou duas subcaracterísticas, as subcaracterísticas que não forem avaliadas não se faz necessário utilizar os casos de testes que pertence a mesma. Uma subcaracterística está sendo avaliada quando pelo menos uma propriedade da mesma é selecionada no passo 4. Porém ainda que o indicado seja a condução de todos os casos de testes das subcaracterísticas que serão avaliadas, na Tabela presente no Apêndice A também é apresentada a relação dos casos de testes abstratos com as propriedades, para circunstâncias em que se opte pela a seleção dos casos de testes, esta seleção pode ser feita mediante as propriedades selecionadas.

6 - Feito todo o planejamento, calcule o custo benefício que esses testes e medições irão demandar. A seção 8 explica esse processo.

7 - Se desejar utilizar alguma das ferramentas sugeridas no guia, use a Tabela no Apêndice B e veja quais ferramentas podem auxiliar na coleta das métricas selecionadas, na seção 9 são apresentadas as licenças de cada ferramenta e os sites para acessá-las.

8 - Após conduzido todo o planejamento para os testes agora é o momento da execução. Todos os testes que serão utilizados devem ser tornados casos de teste concretos de acordo com contexto de sua aplicação, defina os *thresholds* que serão aceitáveis para as métricas, se atentando a correlação identificada na seção 2 e então execute os testes e as medições utilizando a Tabela do Apêndice B, para que esse processo possa conduzido mais rapidamente.

\* Se houver alguma dúvida sobre a implementação prática deste guia, sugerimos a leitura da seção 10.

# 1. Definições de Desempenho

A partir de estudos lidos na literatura, cinco definições de desempenho foram listadas e apresentadas na Tabela 1. Essas definições ajudam a compreender o Desempenho tratado neste guia.

**Tabela 1 - Definições de Desempenho**

Definição	Autores
O desempenho representa a quantidade de recursos utilizados sob condições estabelecidas.	ISO 25010 (2011)
Desempenho é o tempo gasto para executar um serviço, a taxa na qual o serviço é executado e os recursos consumidos durante a execução do serviço.	R. Jain (1991)
O desempenho está preocupado com a rapidez com que o software "responde quando um evento ocorre"	Baixo, Clements e Kazman (2003)
O desempenho envolve limites de tempo / espaço, como cargas de trabalho, tempo de resposta, taxa de transferência e espaço de armazenamento disponível. Por exemplo, "o sistema deve lidar com 100 transações / segundo".	Chung, de Prado Leite e JCS (2009)
O desempenho está preocupado com a qualidade da resposta do software quando um evento ocorre. Para avaliar se um sistema tem bom desempenho, o tempo entre o evento e a resposta pode ser medido primeiro e depois comparado com uma restrição de tempo determinada anteriormente.	Langsari, Rochimah e Akbar (2018)

## 2. Correlação das Características

Por meio da literatura foi possível estabelecer a correlação de Desempenho com 18 características no total de 27 características encontradas para IoT. Essas correlações podem ajudar a identificar possíveis conflitos nos requisitos da aplicação e, assim, buscar maneiras de mitigá-los. Além de auxiliar na elaboração de *thresholds* para as métricas, visto que dependendo das correlações existentes na aplicação é possível que os resultados esperados das medições se alterem. Um exemplo simples é, para uma aplicação onde a Segurança seja uma característica imprescindível pode-se aceitar valores inferiores nas métricas de desempenho visto que há um maior esforço no aspecto segurança, com algoritmos mais robustos, por exemplo. A partir da ISO 25010 e 25012 foram elencadas as definições de cada característica IoT correlacionada a desempenho, essas definições são apresentadas a seguir.

*Compreensibilidade* - grau em que os dados têm atributos que permitem sua leitura e interpretação pelos usuários, e são expressos em idiomas, símbolos e unidades apropriados em um contexto específico de uso.

*Recuperabilidade* - grau no qual, em caso de interrupção ou falha, um produto ou sistema

pode recuperar os dados diretamente afetados e restabeleça o estado desejado do sistema.

*Manutenibilidade* - grau de eficácia e eficiência com que um produto ou sistema pode ser modificado para melhorá-lo, corrigi-lo ou adaptá-lo às mudanças no ambiente e nas exigências.

*Funcionalidade* - conjunto de funções que satisfaz as necessidades explícitas e implícitas para a finalidade a que se destina o produto.

*Dependabilidade* - qualidade do serviço fornecido por um dado sistema e a confiança depositada no serviço fornecido.

*Confidencialidade* - grau em que os dados têm atributos que garantem que sejam acessíveis e interpretáveis apenas por usuários autorizados em um contexto de uso específico.

*Acurácia* - grau em que os dados têm atributos que representam corretamente o verdadeiro valor do atributo pretendido de um conceito ou evento em um contexto de uso específico.

*Usabilidade* - grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso especificado.

*Segurança* - grau em que um produto ou sistema protege informações e dados para que pessoas ou outros produtos ou sistemas tenham o grau de acesso aos dados apropriado para seus tipos e níveis de autorização.

*Disponibilidade* - grau em que os dados têm atributos que permitem que sejam recuperados por usuários e / ou aplicativos autorizados em um contexto de uso específico.

*Eficiência* - grau em que os dados têm atributos que podem ser processados e fornecem os níveis esperados de desempenho usando as quantidades e tipos de recursos apropriados em um contexto de uso específico.

*Interoperabilidade* - grau em que dois ou mais sistemas, produtos ou componentes podem trocar informações e usar as informações que foram trocadas.

*Modificabilidade* - grau no qual um produto ou sistema pode ser modificado de forma eficaz e eficiente sem introduzir defeitos ou degradar a qualidade do produto existente.

*Portabilidade* - grau em que os dados possuem atributos que permitem que sejam instalados, substituídos ou movidos de um sistema para outro preservando a qualidade existente em um contexto de uso específico.

*Confiabilidade* - grau em que um sistema, produto ou componente executa funções especificadas sob condições especificadas por um período de tempo especificado.

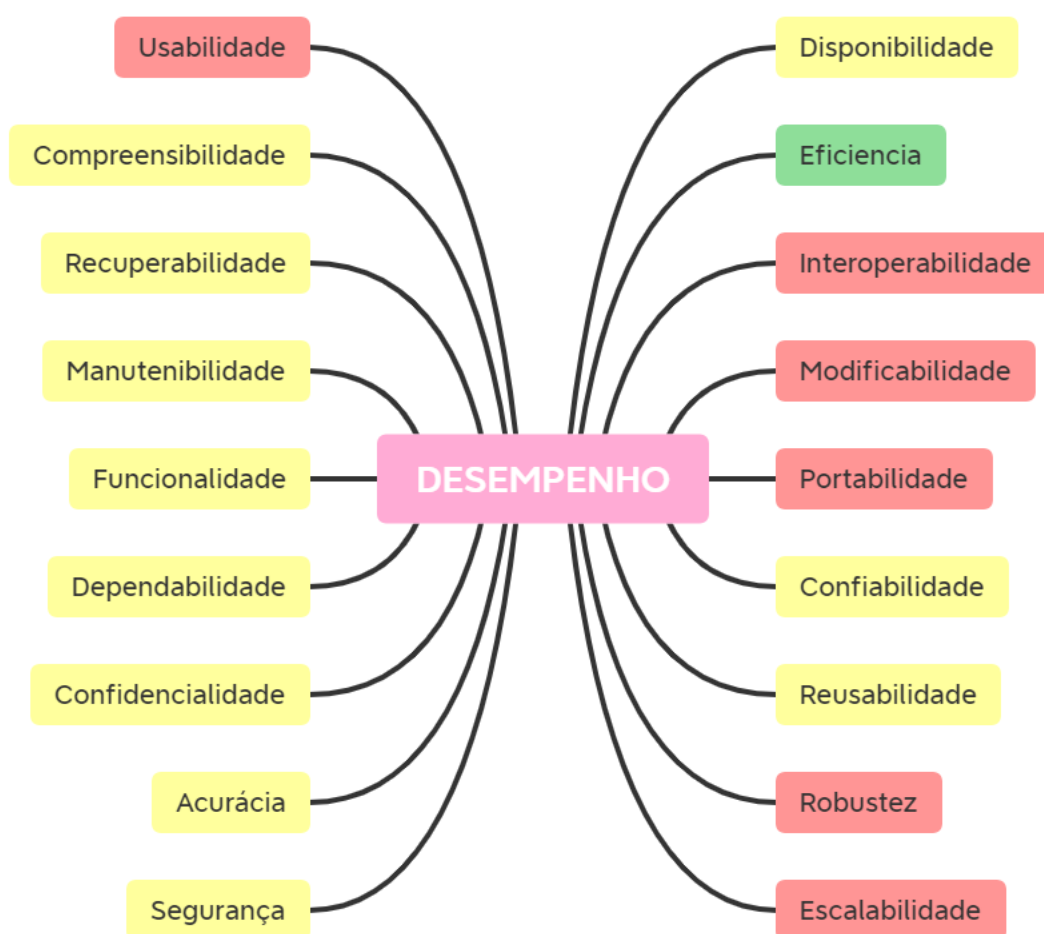
*Reusabilidade* - grau em que um recurso pode ser usado em mais de um sistema, ou na construção de outros recursos.

*Robustez* - habilidade de um sistema, produto ou componente funcionar mesmo em condições anormais.

*Escalabilidade* - facilidade com que uma aplicação ou componente pode ser modificado para expandir suas capacidades existentes, incluindo a capacidade de acomodar grandes volumes de dados.

A Figura 1 mostra as correlações existentes entre essas características e a característica de Desempenho, essas correlações podem ser de 3 tipos: características em vermelho são as que possuem uma influência negativa no Desempenho, por exemplo, usabilidade, em uma

aplicação, quanto maior a quantidade de requisitos relacionados à Usabilidade, maior a dificuldade de aumentar o Desempenho dessa aplicação; características em verde referem-se à influência positiva exercida pela característica em Desempenho, por exemplo, Eficiência, em que quanto maior a Eficiência exigida em uma aplicação, maiores serão os valores alcançados no quesito Desempenho; a última correlação são as características em amarelo, estas correlações dependem do contexto da aplicação, tais como a característica de Disponibilidade que pode ter uma influência positiva, por exemplo, se houver servidores suficientes para satisfazer uma procura, acelerando assim o processo de troca de mensagens, ou uma influência negativa caso contrário.



**Figura 1 - Correlações das características de IoT com o desempenho**

### 3. Configuração do Ambiente

Para a execução deste guia é necessário dispor da seguinte configuração do ambiente:

- Um ou mais sensores;
- Um ou mais atuadores;
- Aplicação que tomará as decisões e enviará os comandos;
- Medidor de energia.

Algumas informações adicionais para a compreensão do guia estão listadas abaixo:

- Todos os testes podem ser realizados localmente;
- As medições podem ser coletadas por meio dos casos de testes;
- O Custo benefício pode ser usado como uma forma de priorização para os testes;
- As ferramentas podem auxiliar durante os testes e medições, mas não são obrigatórias; e
- O impacto das subcaracterísticas ajuda a identificar outras possíveis propriedades que podem ser selecionadas para a realização dos testes.

## 4. Comportamento Temporal

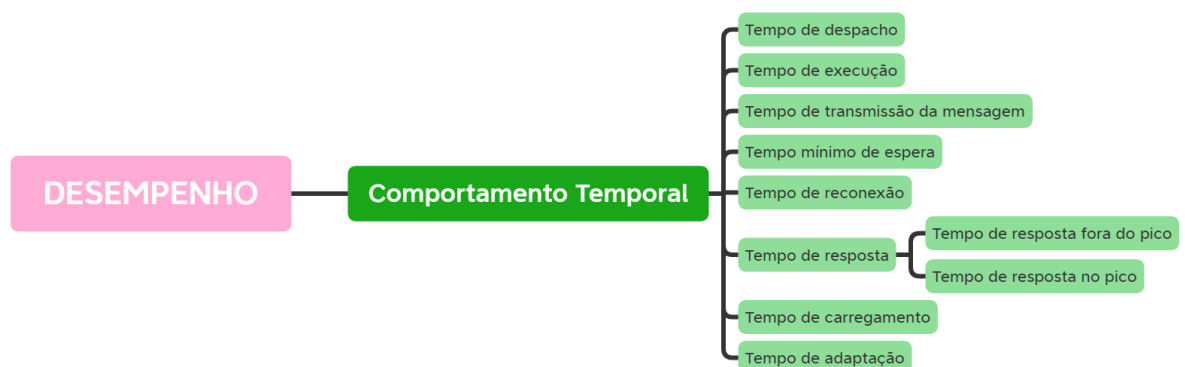
### a. Definição

De acordo com a ISO 25010, comportamento temporal é o nível em que o tempo de resposta e processamento e as taxas de transferência de um produto ou sistema, no desempenho de suas funções, atendem aos requisitos.

### b. Contextualização

Em trabalhos obtidos da literatura, G. Soad, W. Filho e E. Barbosa (2018) e Soares et al., (2014), são apresentadas as relações entre as subcaracterísticas de desempenho e propriedades elencadas nos trabalhos de G. Soad, W. Filho e E. Barbosa (2018) e Uchoa et al., (2017), definindo uma relação baseada na ISO 25010.

A Figura 2 representa essa relação para Comportamento Temporal, as demais subcaracterísticas serão apresentadas em suas respectivas seções. Na figura é apresentado uma divisão definida a partir dos trabalhos supracitados, com base na ISO 25010, nessas oito propriedades estabelecidas para comportamento temporal, foram baseadas as métricas e casos de teste abstratos que serão apresentadas em sequência.



**Figura 2 - Propriedades de comportamento temporal**

Para Comportamento Temporal foram levantadas oito propriedades que caracterizam a subcaracterística em questão, essas propriedades são:



- *P1 - Tempo de Despacho*: tempo até que a tarefa esteja pronta para execução, Sellers (2018) e Nair (2015).
- *P2 - Tempo de Execução*: o tempo necessário para até que uma tarefa seja concluída, Nair (2015).
- *P3 - Tempo de Transmissão da Mensagem*: o tempo de envio de uma mensagem até seu recebimento, Coulouris et al., (2013) e Zhang et al., (2018).
- *P4 - Tempo Mínimo de Espera*: o menor tempo até o recebimento de uma mensagem, K. Govidan, D. Chander e B. Jagyasi (2010).
- *P5 - Tempo de Reconexão*: o tempo em que a rede estabelece uma nova conexão, M. Shavell, A. Schepis e K. Newstadt (2012).
- *P6 - Tempo de Resposta*: o tempo em que é enviada a solicitação e é recebida a resposta, podendo variar quando a aplicação atinge seu pico, Sellers (2018), R. Jain (1991) e Nair (2015).
- *P7 - Tempo de Carregamento*: o tempo que aplicação leva para carregar do ambiente as informações necessárias para seu funcionamento, E. Anderson (2002).
- *P8 - Tempo de Adaptação*: o tempo em que a aplicação se adapta a mudança do estado do ambiente, Lu et al., (2000) e Zhang e Chakrabarty (2004).

### c. Casos de teste abstratos

- **Relacionado a atuadores**

Caso de Teste 01 - TC01	
Título:	Enviar o comando para o atuador
Ambiente de teste:	N atuadores e 1 aplicação
Pré-condições:	Os atuadores devem ser capazes de receber o comando
Passo a passo:	<ol style="list-style-type: none"> <li>1. A aplicação envia o comando desejado</li> <li>2. Verifique o comportamento do atuador</li> </ol>
Pós-condições:	Os atuadores executaram o comando enviado

<b>Caso de Teste 02 - TC02</b>	
Título:	Enviar o comando para o atuador via rede externa
Ambiente de teste:	N atuadores e 1 aplicação
Pré-condições:	O atuador deve ser capaz de receber o comando, a aplicação deve estar em uma rede, fora da rede local
Passo a passo:	<ol style="list-style-type: none"> <li>1. A aplicação envia o comando desejado</li> <li>2. Verifique o comportamento do atuador</li> </ol>
Pós-condições:	Os atuadores executaram o comando enviado

<b>Caso de Teste 03 - TC03</b>	
Título:	Enviar comandos simultaneamente
Ambiente de teste:	N atuadores e 2 aplicações
Pré-condições:	Os atuadores devem ser capazes de receber o comando
Passo a passo:	<ol style="list-style-type: none"> <li>1. Envie um comando simultaneamente das aplicações 1 e 2 para cada atuador</li> <li>2. Verifique o comportamento do atuador</li> </ol>
Pós-condições:	Os atuadores executam os comandos enviados

<b>Caso de Teste 04 - TC04</b>	
Título:	Enviar vários comandos iguais
Ambiente de teste:	1 atuador e N aplicações
Pré-condições:	Os atuadores devem ser capazes de receber o comando
Passo a passo:	<ol style="list-style-type: none"> <li>1. Envie um comando simultaneamente de cada aplicação</li> <li>2. Verifique o comportamento do atuador</li> </ol>
Pós-condições:	O atuador deve ter respondido a todos os comandos enviados

<b>Caso de Teste 05 - TC05</b>	
Título:	Enviar o comando para o atuador no horário de pico
Ambiente de teste:	N atuadores e 1 aplicação

Pré-condições:	Os atuadores devem estar aptos a receber o comando, a aplicação deve estar em seu horário de pico, sabendo que pico é o horário em que há o maior número de acessos à aplicação [Kumar 2000]
Passo a passo:	<ol style="list-style-type: none"> <li>1. Na aplicação envie o comando desejado</li> <li>2. Verifique o comportamento do atuador</li> </ol>
Pós-condições:	Os atuadores executaram o comando enviado

● **Relacionado a sensores**

<b>Caso de Teste 06 - TC06</b>	
Título:	Receber leitura do sensor
Ambiente de teste:	N sensores e N aplicações
Pré-condições:	Os sensores devem estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Os sensores monitoram o ambiente</li> <li>2. A aplicação lê os dados do sensor</li> </ol>
Pós-condições:	A aplicação atualizada em relação ao estado do ambiente

<b>Caso de Teste 07 - TC07</b>	
Título:	Receba a leitura do sensor de uma rede externa
Ambiente de teste:	N sensores e 1 aplicação
Pré-condições:	Os sensores devem estar em perfeito funcionamento, a aplicação deve estar em rede, fora da rede local
Passo a passo:	<ol style="list-style-type: none"> <li>1. Os sensores monitoram o ambiente</li> <li>2. A aplicação lê os dados do sensor</li> </ol>
Pós-condições:	A aplicação atualizada em relação ao estado do ambiente

<b>Caso de Teste 08 - TC08</b>	
Título:	Alterar a configuração do ambiente
Ambiente de teste:	N sensores e 1 aplicação
Pré-condições:	O sensor deve estar monitorando o ambiente

Passo a passo:	<ol style="list-style-type: none"> <li>1. A aplicação lê os dados do sensor</li> <li>2. A configuração do ambiente que o sensor monitora é alterada</li> <li>3. A aplicação faz uma nova leitura dos dados do sensor</li> </ol>
Pós-condições:	A aplicação atualizada em relação ao novo estado do ambiente

<b>Caso de Teste 09 - TC09</b>	
Título:	Status após reconexão
Ambiente de teste:	N sensores e 1 aplicação com acesso ao sensor
Pré-condições:	Os sensores devem estar monitorando o ambiente
Passo a passo:	<ol style="list-style-type: none"> <li>1. A aplicação lê os dados dos sensores</li> <li>2. A conexão entre os sensores e a aplicação é interrompida</li> <li>3. A configuração do ambiente que os sensores monitoram é alterada</li> <li>4. A conexão entre os sensores e a aplicação é restaurada</li> <li>5. A aplicação faz uma nova leitura dos dados do sensor</li> </ol>
Pós-condições:	Aplicação atualizada em relação ao novo estado do ambiente

<b>Caso de teste 10 - TC10</b>	
Título:	Receber a leitura do sensor no horário de pico
Ambiente de teste:	N sensores e N aplicações
Pré-condições:	Os sensores devem estar em perfeito funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Sensores monitoram o ambiente</li> <li>2. A aplicação lê os dados do sensor</li> </ol>
Pós-condições:	Aplicação atualizada em relação ao estado do ambiente

<b>Caso de Teste 11 - TC11</b>	
Título:	Adaptar-se ao novo estado do ambiente

Ambiente de teste:	N sensores e 1 aplicação
Pré-condições:	Os sensores devem estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Sensores monitoram o ambiente</li> <li>2. A aplicação lê os dados do sensor</li> <li>3. A aplicação reage à leitura do dados</li> </ol>
Pós-condições:	Aplicação adaptada ao novo estado do ambiente

#### d. Medições

Tempo de Despacho - M01	
Propósito:	Avaliar o tempo de preparação da tarefa até que esteja pronta para ser enviada.
Método:	Considere o início da preparação da tarefa e compare com o tempo em que está pronta para o envio.
Medida:	$X = t2 - t1$ <p> <math>X</math> = tempo de despacho  <math>t1</math> = tempo de início da preparação da tarefa  <math>t2</math> = tempo em que a tarefa está pronta para enviar </p>
Referência:	Sellers (2018) e Nair (2015)

Tempo de execução - M02	
Propósito:	Avaliar o tempo de execução de uma tarefa.
Método:	Considere o início da execução da tarefa e compare-o com o tempo em que foi concluído.
Medida:	$X = t2 - t1$ <p> <math>X</math> = tempo de execução  <math>t1</math> = tempo de início da execução  <math>t2</math> = tempo em que a tarefa foi concluída </p>
Referência:	Nair (2015)

Tempo de transmissão da mensagem - M03
--

Propósito:	Avaliar o tempo de transferência de uma mensagem.
Método:	Considere o tempo em que a mensagem foi enviada e compare-a com o tempo em que a mensagem foi recebida.
Medida:	$X = t2 - t1$ <p> <math>X</math> = tempo de transmissão da mensagem  <math>t1</math> = tempo em que a mensagem foi enviada.  <math>t2</math> = tempo em que a mensagem foi recebida. </p>
Referência:	Coulouris et al., (2013) e Zhang et al., (2018)

Tempo mínimo de espera - M04	
Propósito:	Avaliar o tempo mínimo de espera até que uma mensagem chegue ao destinatário.
Método:	Realize $n$ experimentos e compare o tempo de transmissão da mensagem em cada um deles.
Medida:	$X = \min(E)$ <p> <math>X</math> = tempo mínimo de espera  <math>E = \{\text{experimento}_1, \text{experimento}_2, \dots \text{experimento}_n\}</math> </p>
Referência:	K. Govidan, D. Chander e B. Jagyasi (2010)

Tempo de reconexão - M05	
Propósito:	Avaliar o tempo que a aplicação pode restabelecer uma nova conexão, quando disponível.
Método:	Contabilize o tempo que a rede ficou disponível e compare com o restabelecimento de uma nova conexão na aplicação.
Medida:	$X = t2 - t1$ <p> <math>X</math> = tempo de reconexão  <math>t1</math> = tempo que uma rede estava disponível  <math>t2</math> = tempo em que uma nova conexão é restabelecido na aplicação </p>
Referência:	M. Shavell, A. Schepis e K. Newstadt (2012)

<b>Tempo de Resposta - M06</b>	
Propósito:	Avaliar o tempo em que a aplicação envia a mensagem e recebe a resposta.
Método:	Contabilize o tempo em que a mensagem é enviada ao destino e compare ao tempo de origem da resposta. Realize a medição até o pico e não pico, sabendo que pico é o momento em que há o maior número de acesso à aplicação [Kumar 2000]
Medida:	$X = t2 - t1$ <p> <math>X</math> = tempo de resposta  <math>t1</math> = tempo em que a mensagem foi enviada ao destino  <math>t2</math> = tempo em que a resposta chegou à origem </p>
Referência:	Sellers (2018), Jain (1991) e Nair (2015)

<b>Tempo de carregamento - M07</b>	
Propósito:	Avaliar o tempo em que a aplicação carrega o estado do ambiente.
Método:	Contabiliza o tempo logo após a retomada da conexão e compare com o tempo após o estado do ambiente ser carregado para a aplicação.
Medida:	$X = t2 - t1$ <p> <math>X</math> = tempo de carregamento  <math>t1</math> = tempo logo após a retomada da conexão  <math>t2</math> = tempo após o carregamento do estado do ambiente na aplicação </p>
Referência:	E. Anderson (2002)

<b>Tempo de adaptação - M08</b>	
Propósito:	Avaliar o tempo em que a aplicação se adapta a um novo estado do ambiente.
Método:	Contabiliza o tempo quando houve uma mudança no estado do ambiente e compare com o tempo após a adaptação da aplicação.

Medida:	$X = t2 - t1$ <p> <math>X</math> = tempo de adaptação  <math>t1</math> = momento da mudança no estado do ambiente  <math>t2</math> = tempo após adaptação da aplicação </p>
Referência:	Lu et al., (2000) e Zhang e Chakrabarty (2004)

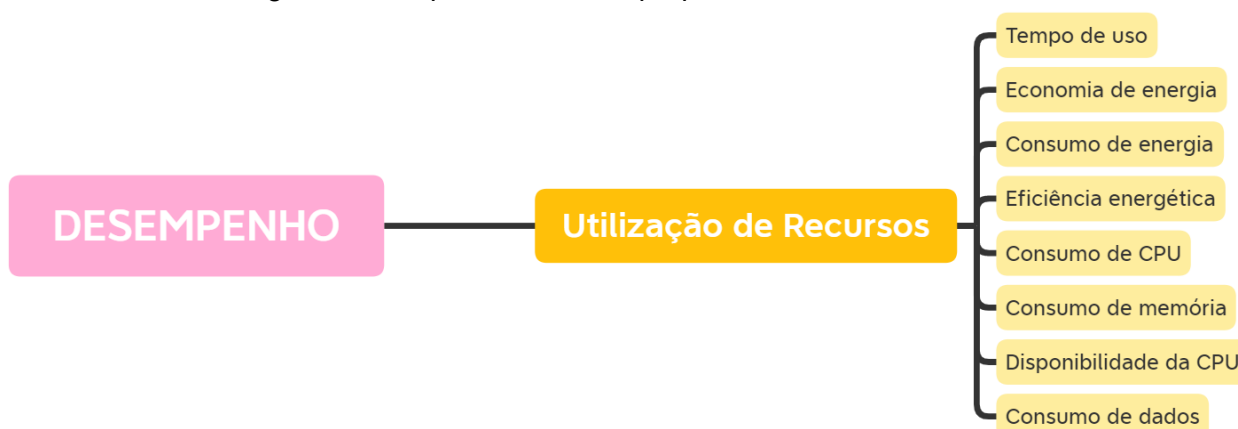
## 5. Utilização de recursos

### a. Definição

De acordo com ISO 25010, é o grau em que as quantidades e os tipos de recursos usados por um produto ou sistema, ao desempenhar suas funções, atendem aos requisitos.

### b. Contextualização

Para a subcaracterística de Utilização de Recursos foram elencados oito propriedades que a caracterizam, essas relações foram baseadas nos estudos de Soares et al., (2014), G. Soad, W. Filho e E. Barbosa (2018) e Uchoa et al., (2017). Na Figura 3 são apresentadas as propriedades.



**Figura 3 - Propriedades de utilização de recursos**

Para Utilização de Recursos foram levantadas oito propriedades que caracterizam a subcaracterística em questão, essas propriedades são:

- *P9 - Disponibilidade da CPU*: porcentagem de tempo em que a CPU está disponível para uso, Thimmannagari (2004) e L. Bonfim (2015).
- *P10 - Consumo da CPU*: quantidade média de uso dos recursos da CPU na aplicação, B. Maytal (2002).
- *P11 - Consumo de Energia*: quantidade média de energia utilizada para o funcionamento de uma aplicação, D. Zhang (2016) e M. Moreno (2014).



- *P12 - Eficiência Energética*: quantidade de energia consumida por uma aplicação comparada a quantidade de energia de fato utilizada para a realizar a função proposta, D. Zhang (2016) e M. Moreno (2014).
- *P13 - Consumo de Memória*: quantidade média de memória utilizada para o funcionamento de uma aplicação, C. Stoeck (2014) e N. Siegmund (2013);
- *P14 - Economia de Energia*: um menor consumo de energia da aplicação, quando comparado ao valor esperado, D. Zhang (2016) e M. Moreno (2014).
- *P15 - Tempo de Uso*: quantidade de tempo que a aplicação permaneceu executando instruções, Thimmannagari (2004) e B. Maytal (2002).
- *P16 - Consumo de dados*: quantidade média de dados utilizados para o funcionamento da aplicação, B. Tiwari (2005) e D. Zhang (2016).

### c. Casos de teste abstratos

#### • Relacionado a atuadores

Caso de Teste 12 - TC12	
Título:	Verificar quanto o sistema consome de energia ao atuar no ambiente
Ambiente de teste:	1 medidor de energia, 1 atuador conectado ao medidor de consumo de energia
Pré-condições:	O atuador deve estar em perfeito estado de funcionamento e conectado ao medidor de consumo de energia
Passo a passo:	<ol style="list-style-type: none"> <li>1. Faça um número padrão de solicitações ao atuador</li> <li>2. Verifique o consumo de energia no medidor</li> <li>3. Faça uma série de solicitações ao atuador no horário de pico do sistema</li> <li>4. Verifique o consumo de energia no medidor novamente</li> </ol>
Pós-condições:	A segunda leitura do medidor foi maior que a primeira

#### Caso de teste 13 - TC13

Título:	Verificar quanto o sistema consome de memória ao atuar no ambiente
Ambiente de teste:	1 atuador
Pré-condições:	O atuador deve estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Execute um número padrão de solicitações ao atuador</li> <li>2. Verifique o consumo de memória</li> <li>3. Execute uma série de solicitações de um momento de pico no sistema para o atuador</li> <li>4. Verifique novamente o consumo</li> </ol>
Pós-condições:	A segunda leitura foi superior à primeira

<b>Caso de teste 14 - TC14</b>	
Título:	Verificar quanto o sistema consome de CPU ao atuar no ambiente
Ambiente de teste:	1 atuador
Pré-condições:	O atuador deve estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Execute um número padrão de solicitações ao atuador</li> <li>2. Verifique o consumo de CPU</li> <li>3. Execute uma série de solicitações desde um momento de pico no sistema até o atuador</li> <li>4. Verifique novamente o consumo de CPU</li> </ol>
Pós-condições:	A segunda leitura foi superior à primeira

<b>Caso de teste 15 - TC15</b>	
Título:	Verificar quanto o sistema consome de dados ao atuar no ambiente
Ambiente de teste:	1 atuador
Pré-condições:	O atuador deve estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Execute um número padrão de solicitações ao atuador</li> <li>2. Verifique o volume de dados trocados na rede</li> <li>3. Execute uma série de solicitações desde um momento de pico no sistema até o atuador</li> <li>4. Verifique o volume de dados trocados na rede</li> </ol>

Pós-condições:	A segunda leitura foi superior à primeira
----------------	---

● **Relacionado ao sensores**

<b>Caso de Teste 16 - TC16</b>	
Título:	Verificar quanto o sistema consome de energia ao detectar o ambiente
Ambiente de teste:	1 medidor de consumo de energia, 1 sensor conectado ao medidor de consumo de energia
Pré-condições:	O sensor deve estar em perfeito estado de funcionamento e conectado ao medidor de consumo de energia
Passo a passo:	<ol style="list-style-type: none"> <li>1. Execute um número padrão de solicitações ao sensor</li> <li>2. Verifique o consumo de energia no medidor</li> <li>3. Execute uma série de solicitações de um momento de pico no sistema para o sensor</li> <li>4. Verifique o consumo de energia no medidor novamente</li> </ol>
Pós-condições:	A segunda leitura do medidor foi superior à primeira

<b>Caso de teste 17 - TC17</b>	
Título:	Verificar quanto o sistema consome de memória ao detectar o ambiente
Ambiente de teste:	1 sensor
Pré-condições:	O sensor deve estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Execute um número padrão de solicitações ao sensor</li> <li>2. Verifique o consumo de memória</li> <li>3. Execute uma série de solicitações de um momento de pico no sistema para o sensor</li> <li>4. Verifique novamente o consumo de memória</li> </ol>
Pós-condições:	A segunda leitura foi superior à primeira

<b>Caso de teste 18 - TC18</b>	
Título:	Verificar quanto o sistema consome CPU ao detectar o ambiente

Ambiente de teste:	1 sensor
Pré-condições:	O sensor deve estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Execute um número padrão de solicitações ao sensor</li> <li>2. Verifique o consumo de CPU da aplicação</li> <li>3. Execute uma série de solicitações desde um momento de pico no sistema até o sensor</li> <li>4. Verifique novamente o consumo de CPU</li> </ol>
Pós-condições:	A segunda leitura foi superior à primeira

Caso de teste 19 - TC19	
Título:	Verificar quanto o sistema consome de dados ao detectar o ambiente
Ambiente de teste:	1 sensor
Pré-condições:	O sensor deve estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Execute um número padrão de solicitações ao sensor</li> <li>2. Verifique o volume de dados trocados na rede</li> <li>3. Execute uma série de solicitações desde um momento de pico no sistema até o sensor</li> <li>4. Verifique novamente o volume de dados trocados na rede</li> </ol>
Pós-condições:	A segunda leitura foi superior à primeira

#### d. Medições

Consumo de CPU em Stand by - M09	
Propósito:	Avaliar quanto o sistema consome da CPU quando em stand by
Método:	Contabilize quanto o sistema consome da CPU apenas por estar ligado, sem realizar nenhuma ação
Medida:	$X = (\sum_{i=1}^n C_n) / n$ <p><math>X</math> = consumo em espera</p>

	<p><math>n</math> = quantidade de leituras de consumo no período de stand by</p> <p><math>C_n</math> = en-ésimo valor de consumo durante o período de stand by</p>
Referência:	Thimmannagari (2004) e L. Bonfim (2015)

Consumo de CPU no pico - M10	
Propósito:	Avaliar quanto o sistema consome da CPU quando atinge o pico
Método:	Contabilize quanto o sistema consome da CPU durante um período de pico
Medida:	$X = (\sum_{i=1}^n C_n)/n$ <p><math>X</math> = consumo em espera</p> <p><math>n</math> = quantidade de leituras de consumo no período de uso em pico</p> <p><math>C_n</math> = en-ésimo valor de consumo durante o período de uso em pico</p>
Referência:	B. Maytal (2002)

Consumo de CPU Médio - M11	
Propósito:	Avaliar o consumo médio de CPU
Método:	Contabilize quanto o sistema consome da CPU durante um período de uso padrão, incluindo momentos de pico e fora de pico, some todos os valores e calcule a média.

Medida:	$X = (\sum_{i=1}^n C_n)/n$ <p><math>X</math> = consumo médio</p> <p><math>n</math> = quantidade de leituras de consumo no período de uso padrão</p> <p><math>C_n</math> = en-ésimo valor de consumo no período de uso padrão</p>
Referência:	L. Bonfim (2015)

Consumo de memória em stand by - M12	
Propósito:	Avaliar o consumo de memória em stand by
Método:	Contabilize quanto o sistema consome de memória somente por estar ligado, sem executar nenhuma ação
Medida:	$X = (\sum_{i=1}^n C_n)/n$ <p><math>X</math> = consumo em espera</p> <p><math>n</math> = número de leituras de consumo no período de stand by</p> <p><math>C_n</math> = en-ésimo valor de consumo durante o período de stand by</p>
Referência:	C. Stoeck (2014) e N. Siegmund (2013)

Consumo de memória no pico - M13	
Propósito:	Avaliar quanto o sistema consome de memória quando atinge o pico
Método:	Contabilize quanto a aplicação consome memória durante um período de pico

Medida:	$X = (\sum_{i=1}^n C_n)/n$ <p><math>X</math>= consumo de pico</p> <p><math>n</math>= número de leituras de consumo no período de uso em pico</p> <p><math>C_n</math> = en-ésimo valor de consumo durante o período de uso em pico</p>
Referência:	C. Stoeck (2014)

Consumo médio de memória - M14	
Propósito:	Avaliar o consumo médio de memória por aplicação
Método:	Contabilize quanto cada aplicação consome da memória durante um período de uso padrão e um período de pico de uso, some todos os valores e calcule a média.
Medida:	$X = (\sum_{i=1}^n M1_n + \sum_{j=1}^m M2_m)/(n + m)$ <p><math>X</math>= consumo médio</p> <p><math>M1</math> = valores de consumo durante o período de uso padrão</p> <p><math>M2</math> = valores de consumo durante o período de pico de uso</p> <p><math>n</math>= número de leituras de consumo no período de uso padrão</p> <p><math>m</math>= número de leituras de consumo no período de pico de uso</p>
Referência:	N. Siegmund (2013)

<b>Consumo de energia em stand by- M15</b>	
Propósito:	Avaliar quanto o sistema consome energia em stand by
Método:	Contabilize, por meio de um medidor, quanto cada componente do sistema (sensores, atuadores, aplicações) consome energia apenas por estar ligado, sem realizar nenhuma ação
Medir:	$X = (\sum_{i=1}^n (\sum_{j=1}^m E_m^n)) / (n * m)$ <p><math>X</math> = consumo de energia em stand by</p> <p><math>n</math>= número de componentes</p> <p><math>m</math>= número de leituras de consumo no período de stand by</p> <p><math>E_m^n</math> = m-ésima leitura do n-ésimo componente</p>
Referência:	D. Zhang (2016) e M. Moreno (2014)

<b>Consumo de energia no pico - M16</b>	
Propósito:	Avaliar quanto o sistema consome energia no período de pico
Método:	Contabilize, por meio de um medidor, quanto cada componente do sistema (sensores, atuadores, aplicações) consome energia durante um período de pico
Medir:	$X = (\sum_{i=1}^n (\sum_{j=1}^m E_m^n)) / (n * m)$ <p><math>X</math>= consumo de pico</p> <p><math>n</math>= número de componentes</p> <p><math>m</math>= número de leituras de consumo no período de uso em pico</p> <p><math>E_m^n</math> = m-ésima leitura do n-ésimo componente</p>
Referência:	D. Zhang (2016)



<b>Consumo Médio de Energia - M17</b>	
Propósito:	Avaliar o consumo médio de energia do sistema
Método:	Contabilize, por meio de um medidor, quanto cada componente do sistema (sensores, atuadores, aplicações) consome energia durante um período de uso padrão e um período de pico de uso, some todos os valores e calcule a média.
Medida:	$X = (\sum_{i=1}^n (\sum_{j=1}^m E_m^n)) / (n * m)$ <p><math>X</math>= consumo médio</p> <p><math>n</math>= número de componentes</p> <p><math>m</math>= número de leituras de consumo no período de uso padrão</p> <p><math>E_m^n</math> = m-ésima leitura do n-ésimo componente</p>
Referência:	D. Zhang (2016) e M. Moreno (2014)

<b>Consumo de dados em stand by - M18</b>	
Propósito:	Avalia o consumo de dados do sistema em stand by
Método:	Contabilize o volume de dados trocados pelo sistema apenas por estar ligado, sem realizar nenhuma ação
Medida:	$X = (\sum_{i=1}^n C_n) / n$ <p><math>X</math>= consumo de dados em stand by</p> <p><math>n</math>= número de leituras de consumo no período de stand by</p> <p><math>C_n</math> = en-ésimo valor de consumo</p>
Referência:	B. Tiwari (2005) e D. Zhang (2016)

<b>Consumo de dados em pico - M19</b>
---------------------------------------

Propósito:	Avaliar o consumo de dados pelo sistema em pico
Método:	Contabilize o volume de dados trocados pelo sistema durante um período de uso em pico, some todos os valores e calcule a média.
Medida:	$X = (\sum_{i=1}^n C_n)/n$ <p><math>X</math>= consumo de dados em pico</p> <p><math>n</math>= número de leituras de consumo no período de uso em pico</p> <p><math>C_n</math> = en-ésimo valor de consumo no período de uso em pico</p>
Referência:	B. Tiwari (2005) e D. Zhang (2016)

Consumo de dados médio - M20	
Propósito:	Avaliar o consumo de dados médio do sistema
Método:	Contabilize o volume de dados trocados pelo sistema durante um período de uso padrão, incluindo momentos de pico e fora de pico, some todos os valores e calcule a média.
Medida:	$X = (\sum_{i=1}^n C_n)/n$ <p><math>X</math>= consumo de dados médio</p> <p><math>n</math>= número de leituras de consumo no período de uso padrão</p> <p><math>C_n</math> = en-ésimo valor de consumo no período de uso padrão</p>
Referência:	B. Tiwari (2005) e D. Zhang (2016)

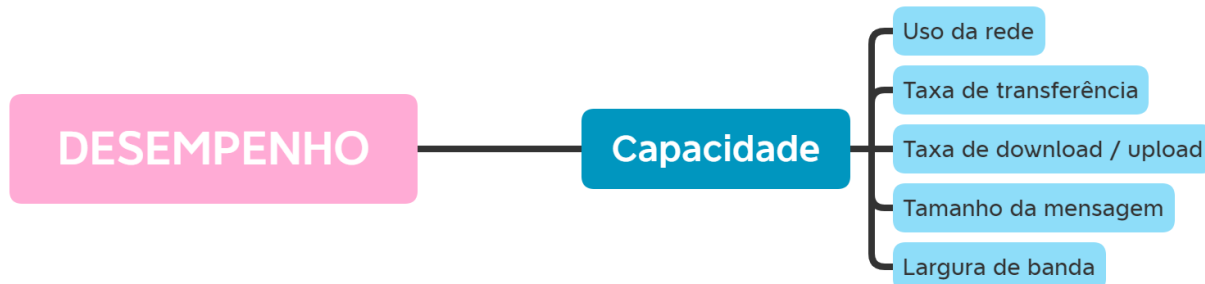
## 6. Capacidade

### a. Definição

De acordo com a ISO 25010, é o grau em que os limites máximos de um produto ou parâmetro do sistema atendem aos requisitos.

## b. Contextualização

Para a subcaracterística de Capacidade foram elencados cinco propriedades que a caracterizam, essas relações estão baseadas nos estudos de Soares et al., (2014), G. Soad, W. Filho e E. Barbosa (2018) e Uchoa et al., (2017). Na Figura 4 são apresentadas as propriedades.



**Figura 4 - Propriedades de capacidade**

- *P17 - Taxa de Download/Upload:* velocidade de recepção de dados de um sistema remoto/velocidade de envio de dados para um sistema remoto, Q. Gao (2016).
- *P18 - Taxa de Transferência:* o número de bits encaminhados por unidade de tempo, R. Jain (1991) e Coulouris et al., (2013).
- *P19 - Tamanho da Mensagem:* o conjunto de bytes presentes em cada pacote para o envio completo da mensagem, L. Peterson e B. Davie (2011) .
- *P20 - Uso da Rede:* porcentagem da rede que está manipulando dados, R. Agrawal e B. Tiwari (2005).
- *P21 - Largura de Banda:* capacidade de transmissão da rede, R. Jain (1991).

## c. Casos de teste abstratos

- **Relacionado a atuadores**

Caso de Teste 20- TC20	
Título:	Estressar um atuador
Ambiente de teste:	1 atuador e N aplicações
Pré-condições:	O atuador deve estar em perfeito estado de funcionamento

Passo a passo:	<ol style="list-style-type: none"> <li>1. Um baixo número de aplicações faz solicitações constantes ao atuador em um intervalo contínuo de tempo</li> <li>2. O número de aplicações é continuamente aumentado até que o atuador não possa responder</li> </ol>
Pós-condições:	O atuador atingiu seu limite

● **Relacionado ao sensor**

<b>Caso de teste 21 - TC21</b>	
Título:	Estressar um sensor
Ambiente de teste:	1 sensor e N aplicações
Pré-condições:	O sensor deve estar em perfeito estado de funcionamento
Passo a passo:	<ol style="list-style-type: none"> <li>1. Um baixo número de aplicações faz requisições constantes ao sensor em um intervalo contínuo de tempo</li> <li>2. O número de aplicações é continuamente aumentado até que o sensor não possa responder</li> </ol>
Pós-condições:	O sensor atingiu seu limite

● **Relacionado ao sistema**

<b>Caso de teste 22 - TC22</b>	
Título:	Download de dados
Ambiente de teste:	Serviço de dados na rede externa ao sistema
Pré-condições:	O sistema requer um arquivo na rede externa para realizar uma ação
Passo a passo:	<ol style="list-style-type: none"> <li>1. O sistema solicita um arquivo de um serviço que está na rede externa</li> <li>2. Um serviço fornece o arquivo solicitado</li> <li>3. O sistema executa a ação</li> </ol>
Pós-condições:	A ação foi realizada

<b>Caso de Teste 23 - TC23</b>	
Título:	Upload de dados
Ambiente de teste:	Serviço de dados na rede externa ao sistema
Pré-condições:	O sistema é capaz de gerar a informação que seja solicitada a você
Passo a passo:	<ol style="list-style-type: none"> <li>1. Um serviço que está na rede externa solicita informações ao sistema</li> <li>2. O sistema gera a informação necessário</li> <li>3. O sistema responde enviando o dados solicitados</li> </ol>
Pós-condições:	Os dados solicitados foram enviados

#### **d. Medições**

<b>Taxa de transferência - M21</b>	
Propósito:	Avaliar a velocidade que um pacote transita pela rede.
Método:	Contabilize o tamanho do pacote e compare com o tempo de envio da mensagem.
Medida:	$X = s/t$ <p><math>X</math>= rendimento</p> <p><math>s</math>= tamanho do pacote</p> <p><math>t</math>= tempo de envio da mensagem</p>
Referência:	R. Jain (1991) e Coulouris et al., (2013)

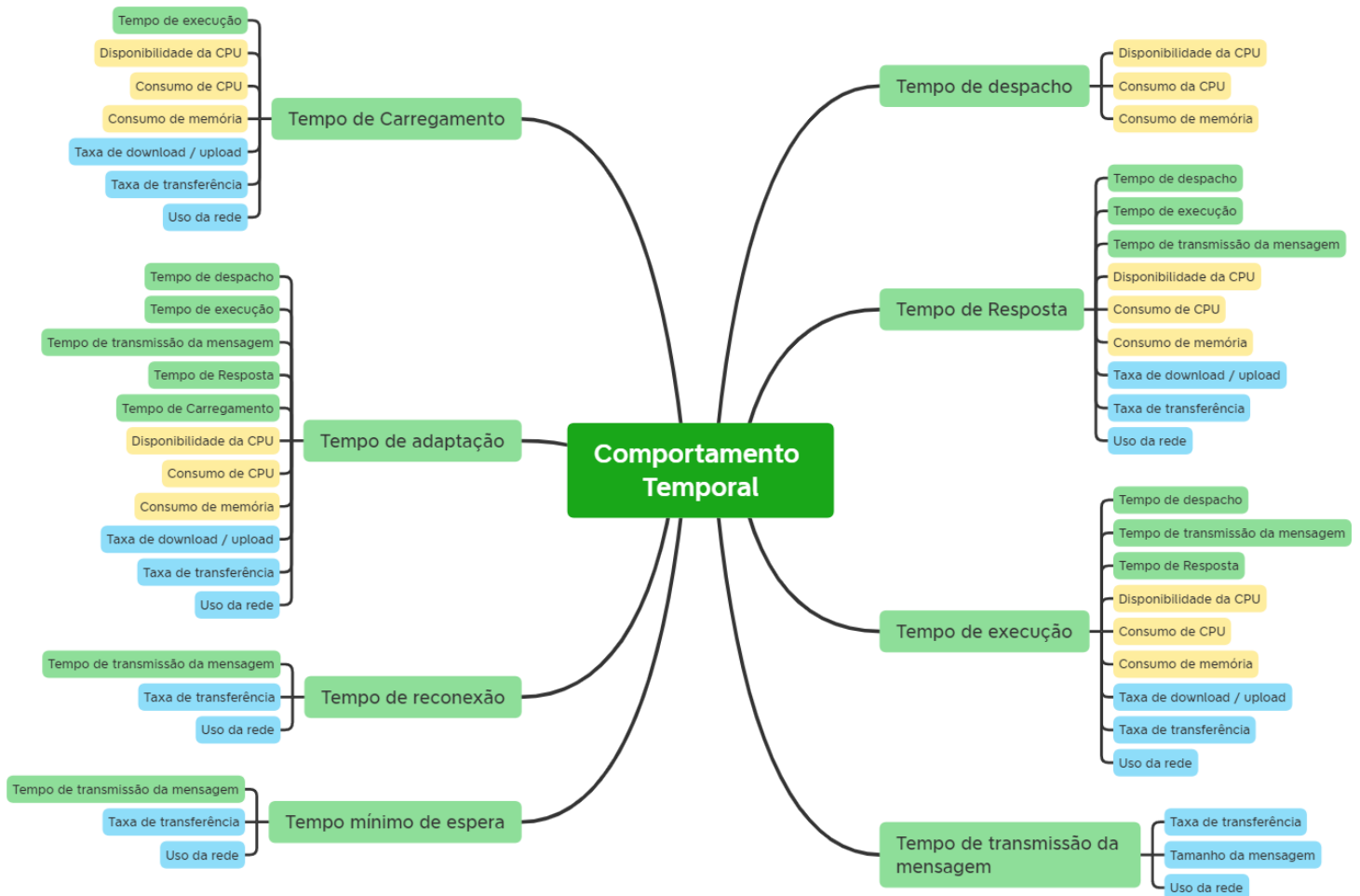
<b>Número Máximo de Clientes Simultâneos - M22</b>	
Propósito:	Avaliar a capacidade do sistema de responder a grandes quantidades de clientes simultâneos.
Método:	Realize um número baixo e constante de solicitações ao sistema em um intervalo contínuo de tempo e aumente esse número continuamente até que o sistema não possa mais responder às requisições.
Medida:	$X = \max(U)$ <p><math>X</math>= número máximo de clientes simultâneos</p>

	$U$ = quantidades de clientes simultâneos
Referência:	R. Agrawal e B. Tiwari (2005)

## 7. Impacto das Subcaracterísticas

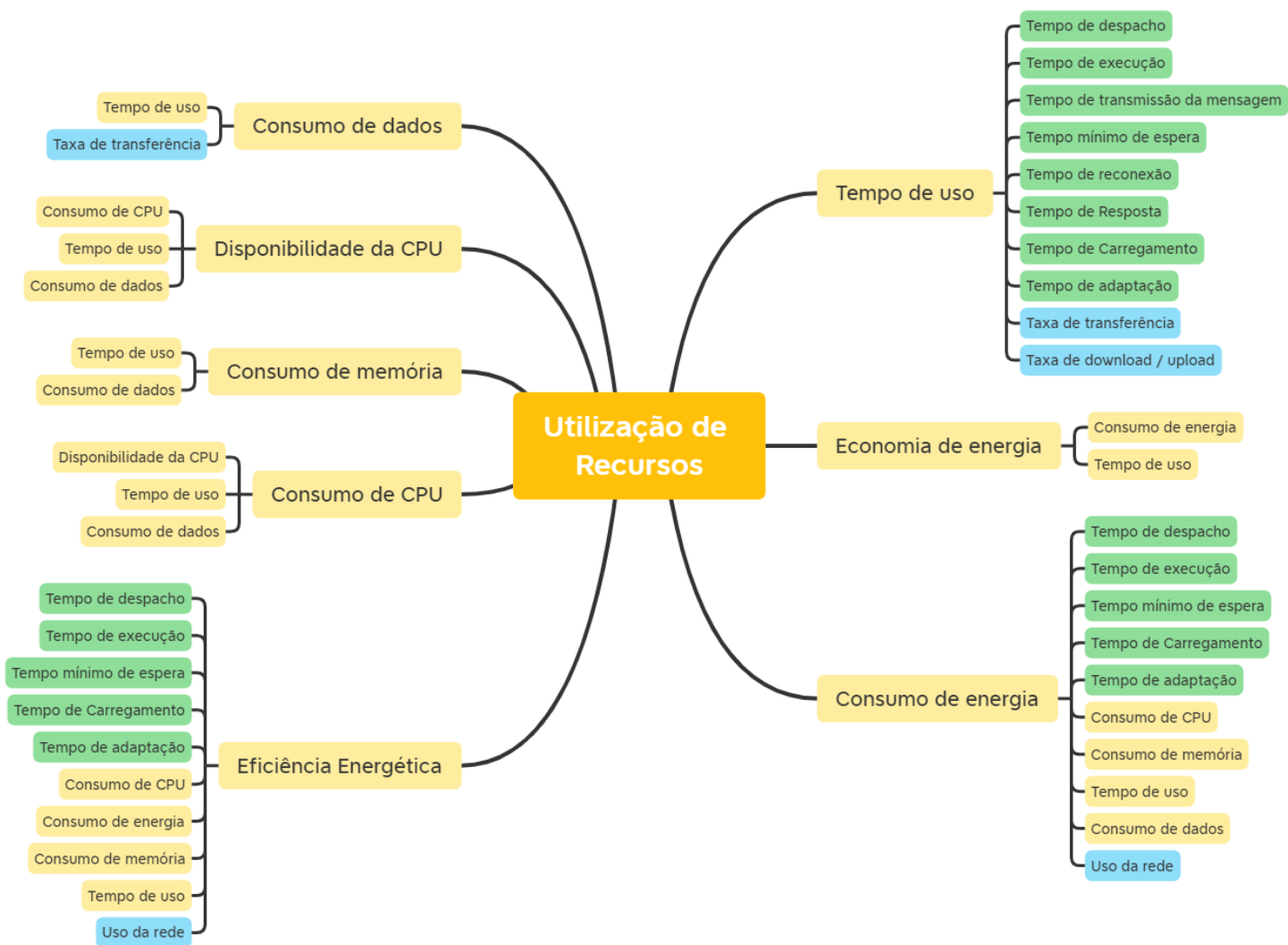
A partir das propriedades apresentadas em cada subcaracterística é possível estabelecer impactos entre as subcaracterísticas. Esses impactos podem ajudar na seleção dos casos de teste e das medições servindo como um indicador de quais propriedades devem ser avaliadas, por exemplo a propriedade de “Tempo de Resposta” da subcaracterística de Comportamento Temporal, sofre impacto da propriedade “Consumo da CPU” da subcaracterística de Utilização de Recursos, pois se o consumo de CPU estiver muito alto, o processamento das informações para ser enviado a resposta será lento. Portanto, isso pode ser utilizado como indicador para em caso de avaliação da propriedade de Tempo de Carregamento também seria importante avaliar o Consumo da CPU, não é obrigatório, mas é indicado. As Figuras 5, 6 e 7 mostram essas relações de impactos para as propriedades de cada subcaracterística, o verde representa as propriedades da subcaracterística do Comportamento Temporal, o amarelo representa as propriedades de Utilização de Recursos e o azul representa as propriedades de Capacidade.

Na Figura 5 é apresentada a relação das propriedades de Comportamento Temporal com as demais propriedades, por exemplo, o “Tempo de Carregamento” propriedade de Comportamento Temporal sofre impacto de “Disponibilidade da CPU”, propriedade de Utilização de Recursos, uma vez que a velocidade do Tempo de Carregamento dependerá da porcentagem de disponibilidade da CPU para carregar rapidamente as informações do ambiente.



**Figura 5 - Relação das propriedades de comportamento temporal com as demais propriedades**

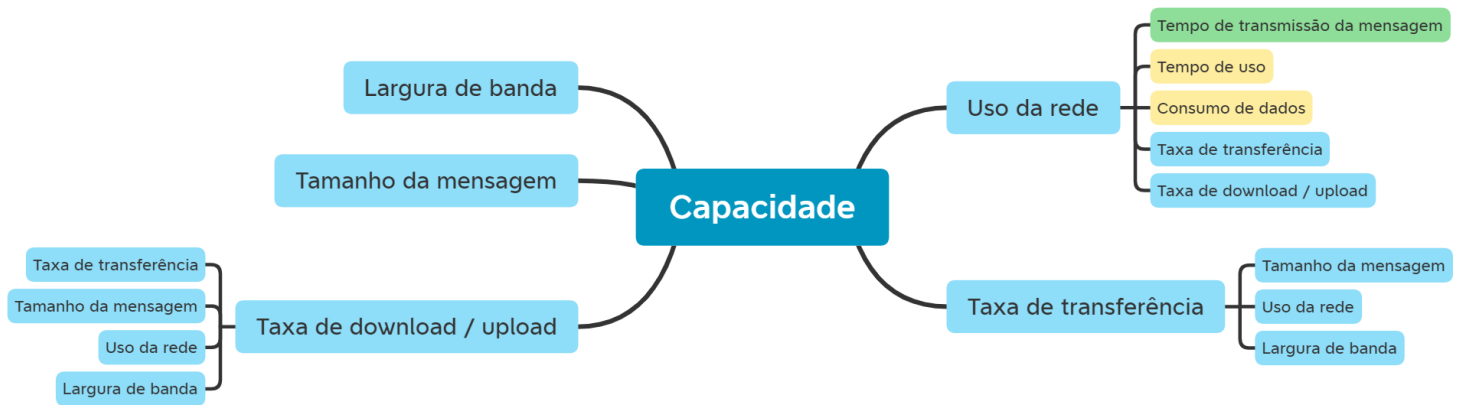
A Figura 6 mostra a relação das propriedades de Utilização de Recursos, em amarelo, com as demais propriedades de Capacidade, em azul, e Comportamento Temporal, em verde. Por exemplo, “Consumo de energia” propriedade de Utilização de Recursos sofre impacto de “Uso da rede”, propriedade da Capacidade, visto que, quão maior o uso da rede maior será o consumo de energia, assim como, quão menor o uso de rede menor será o consumo de energia.



**Figura 6 - Relação das propriedades de Utilização de Recursos com as demais propriedades**

A Figura 7 apresenta a relação das propriedades de Capacidade, em azul, relacionadas com as demais propriedades de Comportamento Temporal, em verde, e Utilização de Recursos, em amarelo. Por exemplo, “Uso de dados”, propriedade de Capacidade, sofre impacto de “Consumo de dados”, pertencente a Utilização de Recursos, apresentando uma relação de proporcionalidade, onde, quanto maior o tempo de uso da aplicação maior será o consumo de dados. As propriedades Largura de banda e Tamanho de mensagem não sofrem impacto de nenhuma outra propriedade, mas impactuam propriedades, como por exemplo, ambas impactuam “Taxa de transferência”.





**Figura 7 - Relação de das propriedades de Capacidade com as demais propriedades**

## 8. Custo Benefício

Com base na correlação de uma característica com as outras, apresentada na seção 2, é possível definir o impacto que essa característica tem nas aplicações. O desempenho, por exemplo, está correlacionado a 18 características, mas nem todas as aplicações priorizarão as 18 características, então dado que uma aplicação prioriza apenas 6 características IoT que se correlaciona com Desempenho, então o coeficiente de impacto (*IC*) é aproximadamente 0,3, ou seja:

*ORC*: número das características correlacionadas ao desempenho priorizadas na aplicação;

*RC*: número total de características relacionadas ao desempenho .

$$IC = ORC/RC$$

Esse impacto (*IC*) quando relacionado ao esforço na execução dos testes e métricas gera o custo benefício que pode auxiliar na priorização dos testes. Para calcular esse esforço, as seguintes fórmulas são aplicadas:

1. Calcula o custo estimado de cada caso de teste com base no tempo médio para o profissional responsável executar um teste e no valor do tempo do profissional responsável pela execução;

$$CT_i = TC_i * VHC_i$$

$CT_i$ : Custo estimado para a realização do caso de teste

$TC_i$ : tempo médio do profissional para realizar um caso de teste

$VHC_i$ : valor do tempo do profissional que realizará o caso teste

2. Após a finalização de todos os CTs, é tomado o valor máximo encontrado;

$$MCT = \max(CT)$$

*MCT: maior custo para a realização de caso de teste*

*CT: todas as estimativas de custo*

3. Normaliza os custos médios dos casos de testes;

$$CCT = (\sum_{i=1}^n CT_i / MCT) / n$$

*CCT: custo médio de casos de teste normalizados*

*CT<sub>i</sub>/MCT: valor de custo estimado do caso de teste i normalizado com o custo mais alto*

*n: número de casos de teste*

4. Repete o processo para as medições e assim obtém o CMD;

*CMT: custo médio das medidas normalizadas;*

5. Desta forma, o Esforço (ESF) é definido por:

$$ESF = (CCT + CMD) / 2$$

Com o Impacto (IC) e Esforço (ESF), através da Figura 8 de custo benefício é possível definir em qual quadrante de priorização está a realização dos testes para desempenho da aplicação. No eixo x é apresentado o impacto (IC), no eixo y o Esforço (ESF). Um Esforço acima de 0,5 é considerado alto, o mesmo para o Impacto. Portanto, em uma aplicação onde o coeficiente de impacto (IC) para o desempenho é de 0,68 a característica de desempenho é considerada uma característica que exerce um alto impacto em para essa aplicação IoT em questão. Os grupos representam:

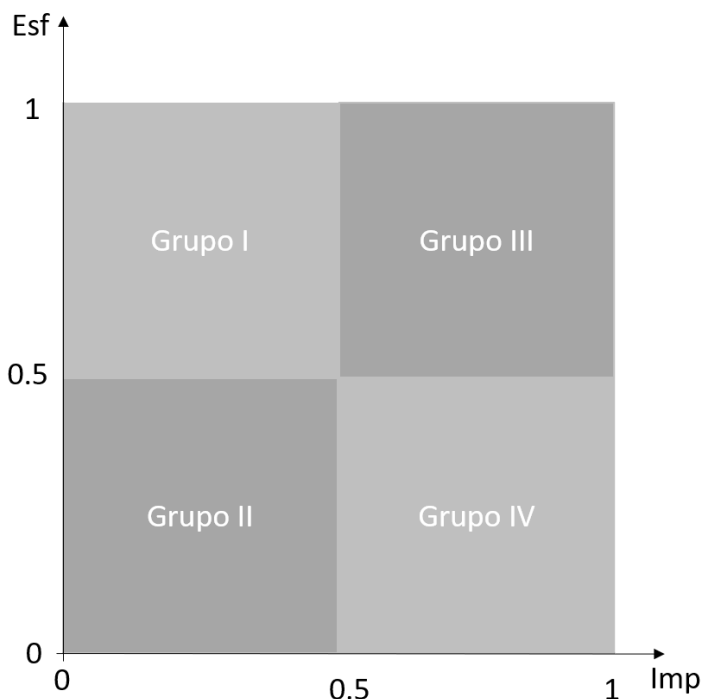
Grupo I: Alto esforço e baixo impacto. Alto custo e baixo benefício = *baixa prioridade* - o esforço para execução dos testes é muito alto e a não execução impacta em poucas características correlacionadas, deve ser avaliado se as poucas características que são impactadas são essências para o sistema, se sim mesmo com o alto esforço devem ser priorizados as condução dos testes e fazer uso das ferramentas e as tabelas de relações para diminuir o esforço durante a execução, caso as características impactadas não sejam essências ao sistema não há necessidade de uma execução imediata dos testes.

Grupo II: Baixo esforço e baixo impacto. Baixo custo e baixo benefício = *média prioridade* - o esforço para execução dos testes é baixo e a não execução impacta em poucas características correlacionadas, deve ser avaliado se as poucas características que são impactadas são essências para o sistema, se sim devem ser priorizados as condução dos testes, caso contrário não há necessidade de uma

*execução imediata dos testes, porém visto que para a execução dos testes é exigido um baixo esforço, em um momento oportuno esses testes podem ser realizados facilmente e de maneira mais completa utilizando a seção de Impacto das subcaracterísticas, adicionando ainda mais propriedades a serem avaliadas.*

Grupo III: Alto esforço e alto impacto. Alto custo, porém, grande benefício = *alta prioridade* - o esforço para execução dos testes é muito alto e a não execução impacta em muitas características correlacionadas, os testes devem ser conduzidos e devem utilizar as estratégias do guia, como ferramentas e as tabelas de relações para diminuir o esforço na execução dos testes.

Grupo IV: Baixo esforço e alto impacto. Baixo custo e grande benefício = *prioridade muito alta* - alta prioridade - o esforço para execução dos testes é baixo e a não execução impacta em muitas características correlacionadas, os testes devem ser conduzidos e por envolver um baixo esforço pode ser, inclusive, adicionadas mais propriedades a serem avaliadas por meio da seção Impacto das subcaracterísticas e caso não esteja sendo feito o uso de todos os casos de testes abstratos é sugerido que sejam adicionados todos.



**Figura 8 - Relação Custo Benefício**

## 9. Sugestões de Ferramentas

A partir dos resultados dos estudos lidos na literatura foi possível elencar 7 ferramentas para auxiliar no teste de aplicações IoT, com foco principal no desempenho. Essas ferramentas estão listadas na Tabela 2, onde é apresentada uma descrição da ferramenta, o método de teste, caixa preta, sem contato com o código-fonte, caixa branca, o teste requer o código-fonte e caixa cinza que mescla os dois métodos. A tabela também mostra o ambiente de teste fornecido pela ferramenta, onde o teste é executado, se a ferramenta é de código aberto ou não e o link para o acesso.

**Tabela 2 - Ferramentas para teste de desempenho de IoT**

<b>Ferramenta</b>	<b>Descrição</b>	<b>Método</b>	<b>Ambiente de Teste</b>	<b>Execução de teste</b>	<b>Licença</b>	<b>Acesso</b>
iFogSim	Simulador para Computação em nuvem. Simula dispositivos e executa medições, por exemplo, de congestionamento da rede.	Caixa cinza	Simulador	Local	Código aberto	<a href="https://github.com/Cloudslab/iFogSim">https://github.com/Cloudslab/iFogSim</a>
Neotys	Plataforma de teste para computação em nuvem. Executa e analisa, testes de carga. Permite que você avalie o tempo de resposta, fornecendo dados de desempenho.	Caixa branca	Plataforma	Remoto	Fechado	<a href="https://www.neotys.com/">https://www.neotys.com/</a>
Wireshark	Um sistema que analisa o tráfego na rede, podendo monitorar a entrada e saída de dados em diferentes protocolos.	Caixa preta	Software Instalável	Local	Código aberto	<a href="https://www.wireshark.org/">https://www.wireshark.org/</a>
LoadUI Pró	Plataforma de teste para computação em nuvem. Executa testes carga, testes funcionais, entre outros. Fornece cenários de teste em que o usuário pode fazer uso de diretamente ou adaptá-lo ao seu aplicação ou até mesmo criar seus próprios casos de teste. Além disso, a ferramenta realiza monitoramento do tráfego na rede.	Caixa branca	Plataforma	Remoto	Fechado	<a href="https://www.soapui.org/professional/loadui-pro/">https://www.soapui.org/professional/loadui-pro/</a>
IoTIFY	Oferece um laboratório virtual, com dispositivos virtuais para o teste de desempenho, de segurança e outros.	Caixa branca	Simulador	Remoto	Fechado	<a href="https://iotify.io/">https://iotify.io/</a>
Tcpdump	Semelhante ao Wireshark, monitora o tráfego de pacotes na	Caixa preta	Software Instalável	Local	Código aberto	<a href="http://www.tcpdump.org/">http://www.tcpdump.org/</a>

	rede. Você pode identificar o tráfego em portas específicas, tráfego de um remetente específico, e destinatários, e outras funções.					<a href="#">p.org /</a>
SOASTA CloudTest	Ferramenta usada no teste de desempenho e teste de carga em Sistemas WEB, permite simular um ambiente com vários dispositivos, com análise em tempo real.	Caixa preta	Software Instalável	Local	Parcialmente Código aberto	<a href="https://community.aka.mai.com/customer/s/s/article/CloudTest-Methodology-and-Assessments?language=en_US">https://community.aka.mai.com/customer/s/s/article/CloudTest-Methodology-and-Assessments?language=en_US</a>

## 10. Exemplo de uso do Guia

Cenário: Uma microempresa de produtos orgânicos monitora a umidade do solo em que é cultivado seus produtos. Para esse monitoramento há 3 sensores que verificam a umidade do solo de tempos em tempos ao ser atingido valores específicos podem ser ligados 4 aspersores inteligentes para umidificar o solo ou podem ser desligados caso a umidade tenha atingido o valor limite. Os testes e as medições dessa aplicação são realizados por um *tester* em que o valor de sua hora de trabalho é de R\$17,00.

A partir dessas informações foram conduzidos os seguintes passos para a execução dos teste de desempenho:

1. Foi lido a seção de desempenho e compreendido o que desempenho significa no contexto do guia.
2. Como base nas definições e nas correlações apresentadas as características correlacionadas de desempenho tidas como prioritárias foram 10: disponibilidade, eficiência, interoperabilidade, modificabilidade, reusabilidade, confiabilidade, funcionalidade, manutenibilidade, recuperabilidade e acurácia, as demais características não foram consideradas relevantes para o contexto da aplicação.
3. No ambiente de teste o sistema é composto por 3 sensores de umidade, 4 aspersores inteligentes como atuadores, 1 aplicação que faz o gerenciamento, além disso para a execução do guia há 1 medidor de energia para os testes que necessitem.
4. A partir da compreensão de cada subcaracterística optou-se por testar todas as seções. As propriedades de cada subcaracterística que serão avaliadas são: Comportamento Temporal - tempo de execução (p2) e tempo de adaptação (p8);

Utilização de Recursos - eficiência energética (p12); Capacidade - taxa de transferência (p19).

5. A partir da seção de impacto das subcaracterísticas foram identificadas mais propriedades que poderiam ser avaliadas, por uma questão de tempo hábil para a execução dos testes foram adicionadas apenas 3: tempo de resposta (p6), tempo mínimo de espera (p4) e economia de energia (p14).
6. Todos os casos de testes serão executados e as métricas selecionadas seguindo a tabela no Apêndice A foram: M03, M08, M15, M17, M18, M06 e M04.
7. Para o próximo passo é calculado o custo benefício sobre o teste:

IC (Índice de impacto) = ORC/RC ->  $IC = 10/18$ , resultado em um **IC = 0,55**, >0,5 então alto impacto.

Foi analisado o tempo médio para execução de cada caso de teste e então calculado o seu CT, por exemplo, para o CT01 foi definido pelo *tester* um tempo de 5 minutos, portanto, para calcular o CT seguiu-se esses passos:

-> primeiro é transformado os minutos em horas visto que o tester recebe 17 reais por hora,  $5/60 = 0,083$ , então:

$CT = 0,083 * 17 = 1,411$ , ou seja, o custo estimado para a realização do caso de teste CT01, nesse contexto, foi de, aproximadamente, R\$1,40.

-> Foram feitos isso para todos os casos de testes e valor máximo encontrado foi de: 1,74, então MCT = 1,74;

-> Feito a normalização dos custos ou seja somados todos os custos estimados e dividido por MCT=1,74 e então dividido pelo total de casos de testes, foi obtido CCT = 0,73.

-> O mesmo processo foi conduzido para as métricas obtendo um CMT = 0,81.

-> Então o esforço é definido por:  $ESF = (CCT + CMT)/2$ , logo  $(0,73 + 0,81)/2$ , resultando em **ESF = 0,77**, >0,5 indicando um alto esforço.

-> Então o Impacto é 0,55 e o Esforço 0,77, essa relação pertence ao grupo 3, indicando uma prioridade alta, logo há uma importância evidente na realização dos testes, apesar do alto esforço, as funções relacionadas a desempenho na aplicação impactam a diversas outras sendo de alta prioridade a condução dos testes dessa característica.

8. Para auxiliar na coleta das métricas M03 e M06, foi selecionado por meio da tabela do Apêndice B a ferramenta IoTify.
9. Visto a real prioridade para os testes, os casos de testes abstratos foram concretizados e definidos os limites das métricas para iniciar a execução dos testes por meio da tabela do Apêndice B, onde, por exemplo, é possível identificar que o caso de teste CT12 pode coletar a métrica M17 e a coleta das métricas M03 e M06 pode ser automatizada por meio da ferramenta IoTify para coletá-la.

# Referências

ISO 25010: 2011 (2011). Qualidade de software e dados. Padrão, Organização Internacional de Padronização, Genebra, CH.

Jain, R. (1991). A arte da análise de desempenho de sistemas de computador - técnicas de projeto experimental, medição, simulação e modelagem. Computação profissional Wiley. Wiley.

Bass, L., Clements, P., & Kazman, R. (2003). Arquitetura de software na prática. Addison-Wesley Professional.

Chung, L., & do Prado Leite, JCS (2009). Sobre requisitos não funcionais em engenharia de software. Em modelagem conceitual: fundamentos e aplicações (pp. 363-379). Springer, Berlim, Heidelberg.

Langsari, K., Rochimah, S., & Akbar, RJ (2018). Medir a eficiência do desempenho do aplicativo aplicando padrões de design e método de refatoração. IPTEK Journal of Proceedings Series, (1), 149-155.

Soad, GW, Filho, NFD e Barbosa, EF (2016). Avaliação da qualidade de aplicativos de aprendizagem móvel. Em 2016 IEEE Frontiers in Education Conference (FIE), páginas 1–8.

Soares, Larissa & Potena, Pasqualina & Machado, Ivan & Crnkovic, Ivica & Almeida, Eduardo. (2014). Análise de propriedades não funcionais em linhas de produtos de software: uma revisão sistemática. Proceedings - 40th Euromicro Conference Series on Software Engineering and Advanced Applications.

Uchoa, AG, Lima, LP, Bezerra, CIM, Monteiro, JM e Andrade, RMC (2017). Dymmer-nfp: Modelagem de propriedades não funcionais e múltiplos cenários de adaptação de contexto em linhas de produtos de software. Em Botterweck, G. e Werner, C., editores, Mastering Scale and Complexity in Software Reuse, páginas 175–183, Cham. Springer International Publishing.

Sellers, N. (2018). Manual da Tecnologia Terahertz. Recursos eletrônicos científicos. 33

Nair, TR (2015). Sistemas avançados de tempo real. Enciclopédia de Ciência e Tecnologia da Informação, Terceira Edição.

Coulouris, G., Dollimore, J., Kindberg, T., E Blair, G. (2013). Distribuídos 5ed Systems: Concepts and Design. Bookman Publishing.

Zhang, K., Leng, S., He, Y., Maharjan, S. e Zhang, Y. (2018). Computação de ponta móvel e rede para Internet das coisas ecológica e de baixa latência. IEEE Communications Magazine, 56: 39–45.

Govindan, K., Chander, D. e Jagyasi, B. (2010). Redes sem fio móveis Multihop. River Publishers Series em Communications Series. River Publishers.

Shavell, M., Schepis, A. e Newstadt, K. (2012). Sistemas e métodos para reduzir a quantidade de tempo necessária para se reconectar a uma rede de computação. Patente

US 8.266.257.

Anderson, EC (2002). Determinar o tempo de carregamento de um sistema operacional. Patente US 6.438.750.

Lu, C., Stankovic, JA, Abdelzaher, TF, Tao, G., Son, SH e Marley, M. (2000). Especificações de desempenho e métricas para sistemas adaptativos em tempo real. Em Proceedings 21st IEEE Real-Time Systems Symposium, pages 13–23. IEEE.

Zhang, Y. e Chakrabarty, K. (2004). Adaptação dinâmica para tolerância a falhas e gerenciamento de energia em sistemas embarcados de tempo real. ACM Transactions on Embedded Computing Systems (TECS), 3 (2): 336–360.

Thimmannagari, C. (2004). Design da CPU: respostas às perguntas mais frequentes. Springer Science & Business Media.

Bomfim, LHDS, Silva, LBD e Salgueiro, RJPDB (2015). Análise de disponibilidade de servidores virtualizados com Markovchains.

Maytal, B. (2002). Patente US Nº 6.457.037. Washington, DC: US Patent and Trademark Office.

Zhang, D., Zhou, Z., Mumtaz, S., Rodriguez, J., & Sato, T. (2016). Uma proposta de eficiência energética integrada para comunicações 5G IoT. IEEE Internet of Things Journal, 3 (6), 1346-1354.

Moreno, M., Úbeda, B., Skarmeta, AF, & Zamora, MA (2014). Como podemos lidar com a eficiência energética em edifícios inteligentes de base iot ?. Sensores, 14.

Stoeck, C., Jolfaei, MA, Buster-Zoeller, A., Kobler, S., Stork, C., & Vielsack, D. (2014). Patente US No. 8.843.526. Washington, DC: US Patent and Trademark Office.

Siegmund, N., Rosenmüller, M., Kästner, C., Giarrusso, PG, Apel, S., & Kolesnikov, SS (2013). Previsão escalável de propriedades não funcionais em linhas de produtos de software: pegada e consumo de memória. Tecnologia de Informação e Software, 55.

Gao, Q., Luby, MG, Mao, Y., & Minder, LC (2016). Patente US No. 9.374.406. Washington, DC: US Patent and Trademark Office.

Peterson, L. & Davie. B. (2011). Redes de computadores. Morgan Kaufmann.

Agrawal, R. & Tiwari, B. (2005). Comunicação de dados e redes de computadores. Vikas Publishing House Unip.



## APÊNDICE A

A Figura 9 refere-se às relações entre as Propriedades, Métricas e casos de testes abstratos e ferramentas. Buscando auxiliar no processo de execução do teste ao selecionar as propriedades. Por meio dessa figura são apresentadas quais métricas devem ser aplicadas, ou seja, é mandatório que as métricas indicadas sejam aplicadas para que as propriedades sejam avaliadas, por exemplo para a avaliar a propriedade P9 é necessário executar as métricas M09, M10 e M11. A relação com os casos de testes abstratos diz respeito ao teste da propriedade em diversos contextos, por isso a importância de realizar todos os testes porque os mesmos abarcam diferentes cenários, por exemplo a propriedade P1 é abarcada pelos casos TC01, TC02, TC03, entre outros, todas em diferentes contextos. Só é indicado selecionar os casos de testes, caso tenham muita *expertise* na área e saiba exatamente o que se deseja testar.

Propriedades		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21
Métricas	M01																					
	M02																					
	M03																					
	M04																					
	M05																					
	M06																					
	M07																					
	M08																					
	M09																					
	M10																					
	M11																					
	M12																					
	M13																					
	M14																					
	M15																					
	M16																					
	M17																					
	M18																					
	M19																					
	M20																					
	M21																					
	M22																					
Casos de Teste	TC01																					
	TC02																					
	TC03																					
	TC04																					
	TC05																					
	TC06																					
	TC07																					
	TC08																					
	TC09																					
	TC10																					
	TC11																					
	TC12																					
	TC13																					
	TC14																					
	TC15																					
	TC15																					
	TC16																					
	TC17																					
	TC18																					
	TC19																					
	TC20																					
	TC21																					
	TC22																					
	TC23																					

Figura 9 - Relação das Propriedades X Métricas X Casos de Teste abstratos

## APÊNDICE B

A Figura 10 refere-se às relações entre Métricas, casos de teste abstratos e ferramentas e busca auxiliar no processo de execução do teste. Por exemplo, o TC01 ao ser executado pode auxiliar na coleta de 8 métricas, entre elas M01, M06 e M18.

A figura também apresenta as ferramentas que podem auxiliar na automatização da coleta das métricas, como a ferramenta Neotys que pode auxiliar na coleta das métricas M03, M06, M18 e M19.

	Métricas	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22
Casos de Teste	TC01																						
	TC02																						
	TC03																						
	TC04																						
	TC05																						
	TC06																						
	TC07																						
	TC08																						
	TC09																						
	TC10																						
	TC11																						
	TC12																						
	TC13																						
	TC14																						
	TC15																						
	TC16																						
	TC17																						
	TC18																						
	TC19																						
	TC20																						
	TC21																						
	TC22																						
	TC23																						
Ferramentas	iFogSim																						
	Neotys																						
	Wireshark																						
	LoadUI Pro																						
	IoTIFY																						
	Tcpdump																						
	SOASTA CloudTest																						

Figura 10 - Relação das Métricas X Casos de Teste abstratos X Ferramentas