

# Plataforma para análise e visualização de dados em tempo real baseada em métodos de Big Data

## *Platform for real-time data analysis and visualization based on Big Data methods*

Gabriel Ferreira  
Instituto Politécnico de Bragança  
Bragança, Portugal  
gabriel97\_felipe@hotmail.com

Paulo Alves  
Centro de Investigação em Digitalização e  
Robótica Inteligente (CeDRI),  
Instituto Politécnico de Bragança  
Bragança, Portugal  
palves@ipb.pt

Simone de Almeida  
Universidade Tecnológica Federal do  
Paraná  
Ponta Grossa, Paraná, Brasil  
simonea@utfpr.edu.br

**Resumo** — Os dados são uma das ciências mais importantes e essenciais para a visualização de gráficos. As análises e técnicas de visualização são muito utilizadas para se criar métricas e previsões a respeito do contexto inserido. O presente trabalho é parte de um projeto da área da engenharia das superfícies, que propõe um modelo de análise e visualização de dados utilizando sensores *IoT* de temperatura e pressão em máquinas injetoras de plástico. O objetivo é monitorar os dados que são coletados no processo de fabricação de moldes de plástico. O resultado foi atingido desenvolvendo um modelo de análise e visualização de dados em tempo real, com sensores que coletam dados de temperatura e pressão, e possibilidade de visualizar o histórico dos dados da aplicação. Também é possível monitorar os alertas que o modelo pode disparar com base nos dados coletados. Através dos dashboards desenvolvido é possível aos gestores de produção terem à sua disposição um sistema de apoio à decisão que agrega informação da produtividade e da necessidade de manutenção preventiva dos equipamentos.

**Palavras Chave** - Visualização de Dados, Sistemas em Tempo Real, Big Data, Internet das Coisas.

**Abstract** — Data is one of the most important and essential sciences for visualizing graphs. Analysis and visualization techniques are widely used to create metrics and predictions about the context. The present work is part of an engineering design of surfaces project, which proposes a model of analysis and visualization of data using temperature and pressure sensors at plastic injection machines. The objective is to monitor the data collected from the manufacture process of plastic molds. The result was achieved by developing a model for analyzing and visualizing data in real time, with sensors that collect temperature and pressure data, and the possibility of viewing the application's data history. It is also possible to monitor the alerts that the model can trigger based on the collected data. Through the developed dashboards, it is possible for production managers to have at their disposal a decision support system that aggregates information of productivity and the need for preventive maintenance of equipment's.

**Keywords** - Data Visualization, Real-Time Systems, Big Data, Internet of Things.

### I. INTRODUÇÃO

Existem diversas técnicas de análise e visualização de dados que incluem algoritmos de Big Data e de Ciência dos Dados, tornando possível automatizar a análise de grandes volumes de dados através da criação de modelos analíticos.

O presente trabalho foi desenvolvido no contexto de um projeto da área da engenharia das superfícies, em que pretende-se desenvolver e aplicar processos de modificação de superfície que promovam soluções avançadas e inovadoras em diferentes sectores de atividades, tais como Automóvel, Aeronáutica, Moldes & Ferramentas, Saúde e Eletrónica [7]. Trata-se de um projeto, que é sediado em parceria com um centro de investigação do Sistema de ID&I nacional, que é uma unidade de investigação interdisciplinar aplicada que agrupa investigadores das áreas de eletrónica, ciência da computação e matemática, localizado no Instituto Politécnico de Bragança.

O projeto é enquadrado nas unidades fabris em máquinas injetoras de plástico, no qual são utilizados sensores para capturar os dados de temperatura e pressão destas máquinas. Esses dados são enviados para uma base de dados local, e então devem ser visualizados em forma de gráficos em tempo real.

Já para consultas posteriores dos dados, também é desenvolvido um *dashboard* para estes dados, dividindo-os por intervalos de tempo, no qual são visualizados os dados máximos, médios, e mínimos, da temperatura e pressão, em forma de gráficos de linhas e circulares. Este trabalho apresenta uma solução para os seguintes objetivos específicos:

1. Desenvolver um servidor para coleta, envio e organização dos dados;
2. Organizar dados de temperatura e pressão coletados pelos sensores;
3. Enviar dados para as interfaces gráficas;
4. Desenvolver um modelo de visualização de dados utilizando os princípios de Big Data;

- Desenvolver uma aplicação para sistemas de placa única de baixo custo para visualização de dados em tempo real na linha de produção.

A solução pretende ser utilizada na linha de produção em sistemas de baixo custo de placa única, como por exemplo RaspberryPI, permitindo a visualização em tempo real, bem como através de uma plataforma web para análise agregada de dados por gestores de produção.

## II. REFERENCIAL TEÓRICO

Nesta secção é apresentado o referencial teórico que foi estudado para a realização do modelo final.

### A. Big Data

O termo *Big Data* refere-se a grandes conjuntos de dados em crescimento. Estes dados podem ser geridos, obtendo-se mais confiabilidade ao serem apresentados, agregando-se dados de diferentes fontes, e codificando os dados para a segurança e privacidade. O objetivo do *Big Data* é garantir que os dados confiáveis sejam facilmente acessíveis, gerenciáveis, armazenados e protegidos [1].

Com previsões diversificadas de dados, como redes de sensores, telescópios, experimentos científicos e instrumentos de alto rendimento, os conjuntos de dados aumentam na taxa exponencial, como demonstrado na Figura 1 [2], chegando a ultrapassar as capacidades computacionais.

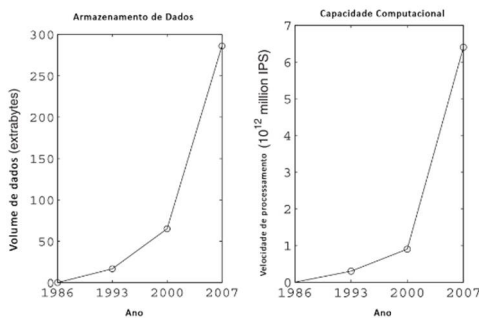


Figura 1. O aumento dos dados superou as capacidades de computação (adaptado de Philip Chen et al. [2])

Enquanto o volume do *Big Data* aumenta, o mesmo acontece com a complexidade e os relacionamentos sob os dados. Então, o *Big Data* teve que se desenvolver para abranger toda esta complexidade. Esta evolução do *Big Data* é definida como "Os três Vs", no qual são: Volume, Velocidade e Variedade [3].

Como é possível ver na Figura 2, os três Vs são vistos como os pilares do *Big Data*, pois quando juntos, se tornam a união de todos os atributos dos mesmos [4].

- Volume:** se refere a quantidade de dados que são armazenados. Antes do crescimento exponencial das plataformas que geram dados, empresas e governos coletavam dados, mas o tempo para armazenar os dados era desafiador.

- Velocidade:** se refere a velocidade na qual os dados são coletados e armazenados. Atualmente, os dados estão sendo coletados em tempo real a taxas muito altas. A velocidade abrange a geração dos dados, como também a frequência da taxa de entrega dos dados.
- Variedade:** se refere aos diferentes tipos de dados que são armazenados. Um dos fatores que torna o *Big Data* realmente grande é a vasta variedade de dados que são armazenados. Muitas das mais recentes são fontes da *web*, incluindo *logs*, cliques em anúncios e redes sociais.

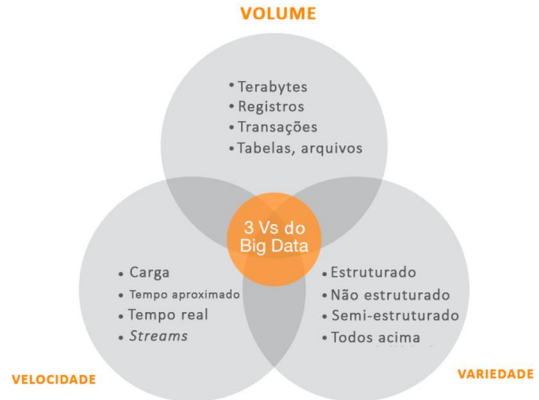


Figura 2. Os três Vs do *Big Data* (adaptado de Russom)

Os dados estão presentes em diversas áreas do conhecimento e são utilizados por diversas tecnologias, devido a vivermos em uma sociedade digital. Seja por mídias, redes sociais, e também por muitas empresas, esses dados são de extrema importância, e devem ser analisados.

Atualmente, existem técnicas de análise de dados, incluindo mineração de dados, análise estatística e aprendizado de máquina. E uma destas técnicas, é a visualização de dados, que permite apresentar os dados importantes e necessários de uma amostra de dados, apresentando de uma maneira clara e concisa [1].

### B. Visualização de Dados

A visualização de dados trata-se de representações visuais de dados quantitativos, representados em sua maioria por gráficos, sejam eles gráficos de "pizza", linhas, barras, áreas, entre outros. A visualização de dados, especificamente, é o processo que utiliza tecnologias computacionais para transformar dados abstratos em modelos visuais [5].

Um sistema em tempo real é um sistema computacional no qual a correção do comportamento do sistema depende não apenas dos resultados lógicos dos cálculos, mas também no instante físico em que esses resultados são produzidos [6]. Significa que estes dados devem ser apresentados com o mínimo de atraso no tempo possível, para garantir a integridade dos dados.

Como apresentado na Figura 3, um sistema em tempo real pode ser decomposto em subsistemas chamados *clusters*, como por exemplo, o objeto controlado, o sistema em tempo real, e o operador humano.

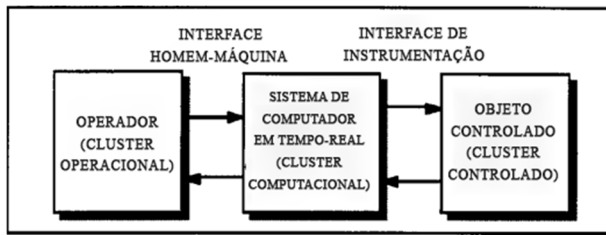


Figura 3. Sistema de tempo real (adaptado de Kopetz et al. [8])

A interface entre o operador humano e o sistema em tempo real é chamada de interface homem-máquina (*Man-Machine interface*) e a interface entre o objeto controlado e o objeto sistema em tempo real é chamado de interface de instrumentação (*Instrumental Interface*).

### C. Internet of Things

*Internet of things*, ou *IoT*, refere-se ao conjunto de dispositivos e sistemas que interconectam sensores e atuadores do mundo real para a internet. Isso inclui vários sistemas:

1. Carros conectados à internet;
2. Dispositivos portáteis, incluindo dispositivos de monitoramento de saúde;
3. Medidores inteligentes;
4. Sistemas de automação residencial e controles de automação;
5. Smartphones;
6. Redes de sensores sem fio.

Incorporando a infraestrutura de computação, uma solução inteligente reúne dois mundos disjuntos. A fusão destes mundos permite sentir e controlar um mundo pelo outro. Um exemplo simples é o ajuste automático de níveis de aquecimento, refrigeração e iluminação [14].

A extração dos dados brutos abrange tanto a detecção de eventos quanto a visualização dos dados brutos, com informações representadas de acordo com as necessidades do usuário final [15].

No paradigma da *IoT*, uma enorme quantidade de redes de sensores são incorporados em vários dispositivos e máquinas no mundo real. Tais sensores implantados em diferentes campos podem coletar vários tipos de dados, como dados de ambiente, dados geográficos, dados astronômicos, etc. Equipamentos móveis, instalações de transporte, instalações públicas e eletrodomésticos podem ser equipamentos de aquisição de dados [16].

### III. TECNOLOGIAS E FERRAMENTAS

Foram utilizadas muitas tecnologias e metodologias para desenvolver a plataforma. Nesta seção, serão listadas as tecnologias, e o desenvolvimento do projeto.

A plataforma de visualização de dados foi desenvolvida utilizando a linguagem de programação *Python* para criar uma *API* que pudesse interagir com os dados em tempo real, e também, criar uma plataforma de visualização com uma interface gráfica. *Python* é uma linguagem de programação que é muito utilizada atualmente. É uma linguagem com uma

sintaxe fácil de se utilizar, e com uma boa curva de aprendizado, principalmente para iniciantes.

A linguagem *Python* também provê muitas bibliotecas que facilitam o esforço para criação de algumas funções [8]. Tais bibliotecas são utilizadas em vários tipos de projetos, e serão citadas abaixo algumas das bibliotecas que foram utilizadas no presente trabalho.

1. **Matplotlib:** *Matplotlib* é uma biblioteca desenvolvida em *Python* que nos permite plotar gráficos 2D para visualização de dados [9].
2. **Flask:** O *Flask* é uma biblioteca de requisições em *Python* para criação de *API's*. Ele tem uma microestrutura, que significa que ele visa manter o núcleo da aplicação simples, mas extensível [10].
3. **Tkinter:** *Tkinter* é um pacote desenvolvido para *Python*, que tem como objetivo desenvolver interfaces gráficas para dispositivos locais [11].
4. **Socketio:** *Socketio* é uma biblioteca que permite trabalhar com sistemas em que exigem uma comunicação cliente-servidor em que não pode haver perda de pacotes [12].

O *MongoDB* é um banco de dados distribuído, baseado em documentos e de propósito geral, orientado para desenvolvedores de aplicativos multiplataforma e para a era da nuvem. Seus dados são armazenados em forma de documentos, o que significa que ele armazena dados em documentos semelhantes a JSON, utilizando o conceito de chave-valor. A sua linguagem tem uma consulta rica e expressiva que permite filtrar e classificar por qualquer campo, independentemente de quanto alinhado possa estar em um documento [13].

## IV. DESENVOLVIMENTO

### A. Estrutura do modelo

Para a execução do modelo, há a necessidade de organizar quais as partes que devem se conectar no projeto, para que se possa fazer a visualização dos dados corretamente. Conforme os dados são captados pelos sensores, estes dados devem ser coletados pelo servidor, e apresentados em uma plataforma que apresenta gráficos em tempo-real.

Como apresentado na Figura 4, há uma maneira de conectar o servidor para receber os dados diretamente dos sensores conectados a um *Raspberry PI*. A primeira ação é quando o dado é enviado para o servidor, para efetuar a coleta dos dados (1), e então, ao receber este dado, o servidor efetua duas ações quase que simultaneamente, no qual se envia o dado atual para todas as interfaces gráficas que estiverem conectadas ao servidor por meio de uma conexão *Broadcast* (2), e insere o dado na base de dados local do *MongoDB* (3).

A partir destes passos, é possível utilizar os dados que estão armazenados na base de dados em consultas futuras, e integrar uma interface web para utilização, como também foi desenvolvido no modelo.

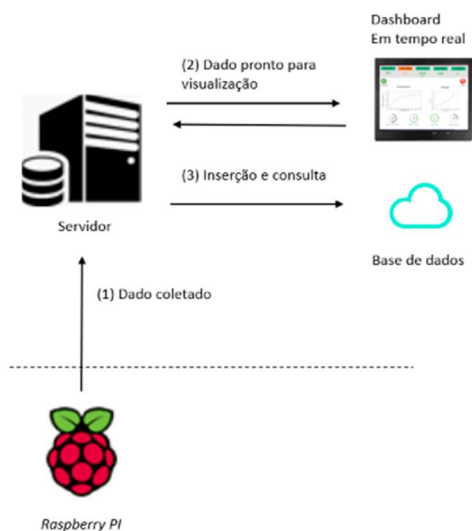


Figura 4. Exemplo de estrutura do modelo

### B. Serviço de Gestão de Dados

Inicialmente, é desenvolvido o serviço para coleta os dados e envia por meio da utilização do *Socketio*, utilizando uma conexão *broadcast* que otimiza o desempenho do envio dos dados.

As tarefas de inserção e envio sempre acontecerão em conjunto, porém, o envio dos dados pode não acontecer, visto que dependerá que tenha algum cliente conectado ao servidor para receber o dado. Já a inserção sempre acontecerá, independente de ter algum cliente conectado ou não, o dado deve ir para a base de dados.

Como se pode notar na Figura 5, após inicializar o servidor e efetuar suas configurações, o servidor aguarda uma requisição para efetuar alguma ação. Caso ele receba uma requisição para coleta de dados, ele primeiramente irá verificar se há algum cliente conectado ao servidor. Caso haja, ele irá enviar esta mensagem em formato *JSON*, que será recebida por uma interface que é apresentada para sua visualização.

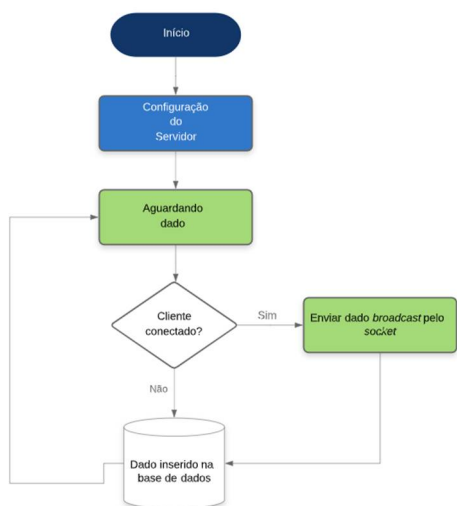


Figura 5. Fluxograma do servidor do modelo

O processo em que o cliente se conecta ao servidor em uma porta e consegue receber todas as mensagens que são enviadas é feito com a ferramenta *Socketio*. Como falado anteriormente, esta ferramenta cria uma porta em que, por meio dela, o servidor e o cliente conseguem se comunicar por meio de mensagens em tempo real, por meio de *sockets*.

Estes *sockets* são como uma referência do endereço do cliente que é criado no servidor, e vice-versa. Por esta razão, o servidor consegue se comunicar com dezenas de clientes ao mesmo tempo, sem precisar que todos eles efetuem uma requisição.

Como pode-se ver na Figura 6, o servidor consegue criar uma conexão unificada onde se percebe todos os diferentes clientes conectados, como um sistema distribuído. Desta forma, se consegue enviar a mesma mensagem ao mesmo tempo para todos os clientes conectados, não permitindo que estes dados se percam.

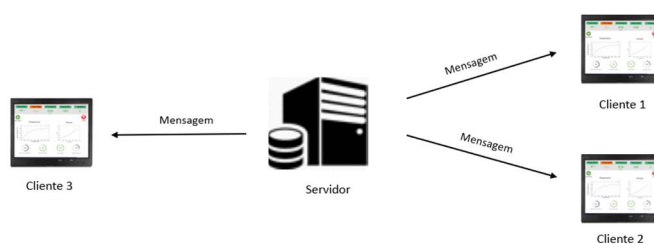


Figura 6. Diagrama do servidor com as conexões utilizando Socketio

### C. Estrutura dos dados

Para poder-se apresentar os dados, na transmissão entre servidor e cliente, deve-se saber o padrão da mensagem, que no caso, é um documento contendo todas as informações de uma leitura dos dados.

Cada leitura é representada por um documento, que é salvo em uma coleção, e posteriormente, consultado na base de dados. Como no presente projeto é utilizado a base de dados em *MongoDB*, então, os dados são estruturados em formato *JSON*. Este padrão de dados facilita a leitura do lado da aplicação, pois facilita a leitura dos dados em objeto.

### D. Sistema de Aquisição de Dados em Tempo Real

Os dados coletados e a apresentação destes dados acontecem quase que simultaneamente, e por isso, é necessário que estes dados sejam tratados e enviados para a interface gráfica o mais eficiente possível.

Pode haver muitas interfaces conectadas ao mesmo tempo, e todas elas recebem os dados em simultâneo. Por esta razão que foi desenvolvido de forma que o cliente efetua a requisição para que possa iniciar a recepção de dados, e a partir daí, o cliente é provido com a apresentação dos mesmos.

O cliente ao anunciar que está conectado significa que o servidor apenas enviará um dado para os clientes quando há algum cliente conectado, não exigindo a necessidade de que os dados sejam enviados quando não há nenhum cliente conectado. Isto facilita na tomada de decisões que o servidor

tem com o cliente, e não o sobrecarrega quando há um alto número de requisições simultâneas.

De todas as formas, para que os dados sejam apresentados com precisão, é necessário que seja feito um tratamento destes dados para caso ocorra algum dado incorreto na aplicação. Este processo de tratamento e análise de dados será abordado no tópico seguinte.

### E. Sistema de Tratamento de Dados

Como a amostra de dados é muito grande, e se necessita que apresente o maior número de informação nas interfaces, deve-se efetuar um tratamento destes dados, antes que eles sejam inseridos na base de dados, e também depois, quando é efetuada uma consulta na base. Por isso, é criado uma média dos dados para serem apresentados, pois como há uma quantidade muito grande de dados sendo recebida, é necessário que se tire a média para se apresentar os gráficos de uma maneira mais limpa.

Na Figura 7 pode-se observar uma agregação do *MongoDB* do tipo *Match*, no qual ele filtra na base de dados todos os dados que estejam em um intervalo. O dado é requisitado com dois parâmetros, o *Timestamp* inicial e o *Timestamp* final da consulta. Este *Timestamp* é um valor numérico que representa a data exata de um intervalo de tempo.

```
{
  "$match": {
    "date": {
      "$gte": float(initial_timestamp),
      "$lte": float(final_timestamp)
    }
  }
}
```

Figura 7. Exemplo de uma consulta em intervalo de tempo

Conforme o intervalo de tempo de agrupamento é menor, mais documentos serão enviados para o usuário. Este intervalo é muito importante para manter a integridade dos dados, pois, caso o intervalo seja muito grande, será considerado uma quantidade muito grande de dados para apenas um agrupamento, então, a média do dado perderá a sua precisão.

### F. Aplicação em tempo real

Esta aplicação local foi desenvolvida em *Python*, utilizando a biblioteca *Tkinter*. A intenção principal é que a interface seja simples e que necessite pouco processamento do hardware, para que possa ser utilizado em ambientes com pouca disponibilidade de componentes, como computadores *Desktop*. A ideia é que seja implementado para ser utilizado diretamente num computador de baixo custo como um *Raspberry PI*, onde são coletados os dados dos sensores.

Foram desenvolvidos dois gráficos que representam, respectivamente, os dados de temperatura e os de pressão. Alguns dos campos apresentam os dados máximos que foram alcançados desde a execução da aplicação, como os dados máximos de temperatura e de pressão, para que seja documentado o pico mais alto dos dados no presente momento.

Quando os dados começam a ser enviados, a interface recebe uma requisição com os dados para serem apresentados nos gráficos. A interface então é atualizada com os gráficos de linha em sua apresentação, como podemos ver na Figura 8. Já também é possível visualizar o número de peças criadas pela máquina injetora de plástico, por meio de uma previsão da temperatura. Sempre que a temperatura chega a um pico no gráfico, é incrementado o número de peças.

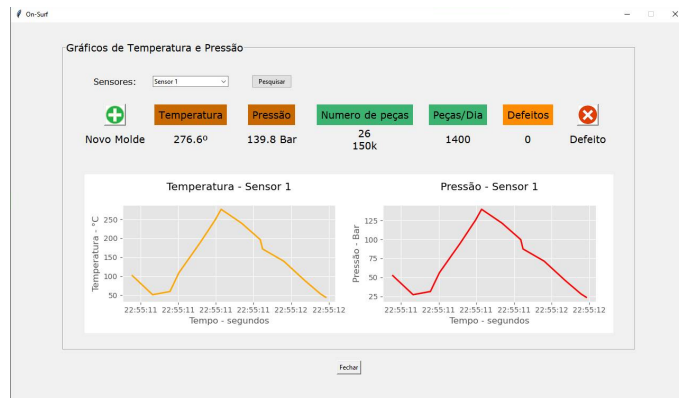


Figura 8. Interface de visualização de dados

Como foram utilizados dados de temperatura e pressão para o modelo, os mesmos são apresentados através dos seus dados máximos em execução da aplicação, representados através de uma escala de cores diferentes para melhor visualização. Quanto mais as cores se aproximam do vermelho, mais significa que os dados aumentaram nas últimas amostras. Os dados que foram projetados são com base no que foi definido pelo projeto, que são os limites da máquina injetora de plástico. Esta mudança de coloração é apresentada na Figura 9.

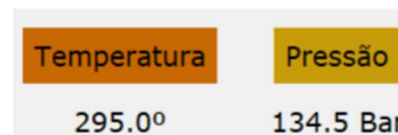


Figura 9. Dados simulados de temperatura e pressão

Esta aplicação não é necessária para que a máquina tenha uma quantidade grande de processamento, e por isso, foi ajustado uma taxa de atualização para que os gráficos sejam atualizados. Os gráficos trabalham numa taxa de atualização de cerca de 1 segundo, pois é o mesmo tempo em que os sensores conseguem trabalhar na taxa de atualização dos dados para a base de dados.

## V. RESULTADOS

Para os resultados, foram utilizados dados de temperatura e pressão simulados por um gerador de onda.

Primeiramente, foi avaliado os resultados em questão de desempenho da aplicação. Os dados das requisições mostraram uma alta velocidade quanto às requisições efetuadas no sistema. O tamanho da amostra apresenta pouca quantidade de *bytes*, facilitando a transferência destes dados para a interface gráfica.

Como é apresentado na Figura 10, percebe-se que as requisições têm um desempenho de cerca de 300

milissegundos, ou 0.3 segundos, para que os dados sejam recebidos. Nos testes, apresentou uma média de 300 *Kilobytes* de tamanho. Estes testes não conseguem apresentar o desempenho da interface renderizar os dados nos gráficos, pois isto depende do *Hardware* utilizado.

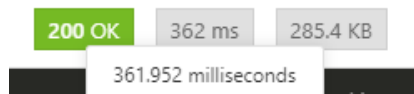


Figura 10. Análise de desempenho em tempo real

Os dados apresentados na Figura 11 foram coletados dos gráficos em tempo real. Após a requisição ser efetuada, a aplicação renderiza os campos e os gráficos, assim, apresentando os dados. Os dados são apresentados com precisão, e em um volume que não interfere na renderização.

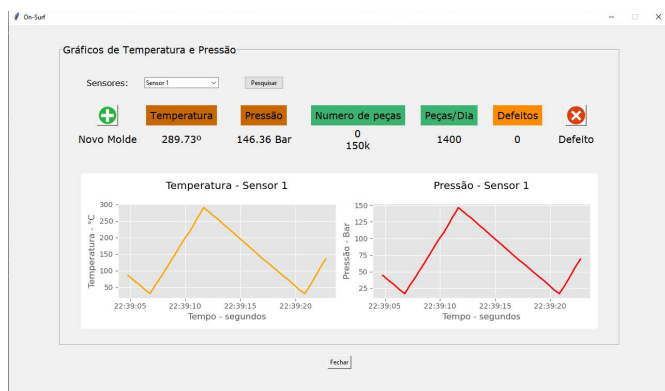


Figura 11. Análise da aplicação em tempo real

Este gráfico de temperatura apresenta todas as contagens de documentos do sensor de temperatura. O eixo Y apresenta o valor da contagem, que no caso, é o valor da temperatura em graus. Já no eixo X, é apresentado o exato momento da leitura. O gráfico consegue apresentar que o maior dado foi perto dos 300 graus, dentro de um intervalo de 20 segundos.

## VI. CONCLUSÃO

Neste trabalho foi apresentado um modelo de análise e visualização de dados em tempo real para variáveis de temperatura e pressão de um sistema de produção industrial. Devido ao grande volume de dados gerados e da velocidade que os dados devem ser coletados, e pela quantidade de abordagens que o *Big Data* abrange, implicou um estudo para desenvolver uma plataforma em que estes dados sejam tratados de uma forma eficiente para serem visualizados em tempo real.

Como pode-se perceber na análise dos dados, e nos testes efetuados ao sistema, a plataforma apresentou um tempo de resposta adequado para a visualização dos dados em tempo real com amostras de 1 segundo. A grande carga de dados foi um problema que foi necessário solucionar, pois quanto maior a carga de dados por segundo, mais o sistema apresenta um atraso na visualização. Porém, com o tratamento destes dados, a aplicação consegue apresentar os dados sem atrasos significativos.

A carga de dados não resultou também em problema nos testes, visto que a visualização dos dados não foi prejudicada por este fator. O modelo apresentado permite assim visualizar dados e analisá-los em tempo real, que era um dos objetivos do projeto.

O modelo pode ser utilizado de maneira considerável em ambientes de produção industrial com sistemas de controlo de qualidade em tempo real, permitindo efetuar previsões de desgaste nos moldes da máquina, e então, facilitando a manutenção preditiva dos equipamentos.

## AGRADECIMENTOS

Este trabalho foi apoiado pela FCT - Fundação para a Ciência e Tecnologia no âmbito do Projeto UIDB/05757/2020.

## REFERÊNCIAS BIBLIOGRÁFICA

- [1] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen e S. Belfkih, "Big Data technologies: A survey", *Journal of King Saud University - Computer and Information Sciences*, n.º30, pp. 431–448, 2018.
- [2] C. P. Chen e C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on 'Big Data'", *Information sciences*, vol. 275, pp. 314–347, 2014.
- [3] D. Laney, "3D Data Management: Controlling Data Volume, Velocity and Variety", *META Group Research Note*, 2001.
- [4] P. Russom, "Big Data Analytics", *TDWI RESEARCH*, 2001.
- [5] D. M. Ribeiro et al., "Visualização de dados na Internet", tese de doutoramento, *Dissertação (Mestrado em Tecnologias da Inteligência e Design Digital)*, 2009.
- [6] H. Kopetz, *Real-time systems: design principles for distributed embedded applications*. Springer Science & Business Media, 2011.
- [7] ON-Surf, <http://onsurf.teandm.pt/>, Accessed: 2020-10-04.
- [8] Python, <https://www.python.org>, Accessed: 2020-07-25.
- [9] J. D. Hunter, "Matplotlib: A 2D graphics environment", *Computing in Science & Engineering*, vol. 9, n.º3, pp. 90–95, 2007. doi:10.1109/MCSE.2007.55.
- [10] Flask, <https://flask.palletsprojects.com/en/1.1.x/>, Accessed: 2020-07-25.
- [11] Tkinter, <https://docs.python.org/3/library/tkinter.html>, Accessed: 2020-07-28.
- [12] Socketio, <https://flask-socketio.readthedocs.io>, Accessed: 2020-07-25.
- [13] MongoDB, <https://www.mongodb.com>, Accessed: 2020-07-27.
- [14] M. Satyanarayanan, "Pervasive computing: Vision and challenges", *IEEE Personal Communications*, vol. 8, n.º4, pp. 10–17, 2001.
- [15] J. Gubbi, R. Buyya, S. Marusic e M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future generation computer systems*, vol. 29, n.º7, pp. 1645–1660, 2013.
- [16] M. Chen, S. Mao e Y. Liu, "Big data: A survey", *Mobile networks and applications*, vol. 19, n.º2, pp. 171–209, 2014.