v7.n12.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

SISTEMA DE MONITORAMENTO DE INDICADORES DE PRODUÇÃO EM LINHA DE ENVASE UTILIZANDO PYTHON, MONGODB E POWER BI: UMA PROVA DE CONCEITO

PRODUCTION INDICATOR MONITORING SYSTEM FOR A FILLING LINE USING PYTHON, MONGODB, AND POWER BI: A PROOF OF CONCEPT

SISTEMA DE MONITOREO DE INDICADORES DE PRODUCCIÓN EN UNA LÍNEA DE ENVASADO UTILIZANDO PYTHON, MONGODB Y POWER BI: UNA PRUEBA DE CONCEPTO

Roberta Rie Flushio Ohata[1], Renata Mirella Farina[2], Fabiana Florian[3]

https://doi.org/10.47820/recima21.v7i12.9879

PUBLICADO: 12/2026

RESUMO

Em linhas de envase de cosméticos com baixo grau de automação, o registro da produção em papel e planilha dificulta a consolidação dos dados, favorece inconsistências e atrasa a análise dos resultados. Este trabalho tem como objetivo desenvolver e avaliar uma Prova de Conceito (PoC) para o monitoramento de indicadores de produção de uma linha de envase, utilizando Python, FastAPI, Pydantic, PyMongo, MongoDB e Power BI. A pesquisa, de natureza aplicada, seguiu a *Design Science Research*: o problema foi identificado em uma linha real e o artefato foi desenvolvido, demonstrado e avaliado com dados simulados. A PoC recebe os registros por uma API REST, valida a estrutura e as regras de domínio, trata eventos duplicados e fora de ordem, armazena eventos e indicadores no MongoDB e calcula rendimento, perdas, disponibilidade, desempenho, qualidade e Eficiência Global dos Equipamentos (*Overall Equipment Effectiveness*, OEE), sinalizando as inconsistências. Na avaliação técnica, os 71 testes automatizados foram aprovados, as 17 requisições HTTP, com 44 asserções, foram executadas sem falhas e os indicadores coincidiram com o cálculo manual. As divisões por zero resultaram em indicadores nulos e sinalizados, e as entradas inválidas foram rejeitadas. O painel no Power BI apresentou os indicadores por lote e valores agregados ponderados dos lotes sem inconsistência. Conclui-se que o artefato atende aos requisitos técnicos definidos. A coleta na linha, a redução do atraso na informação e a utilidade em operação real não foram avaliadas, e por isso não se atribuem ganhos de produtividade à solução.

PALAVRAS-CHAVE: Automação industrial. *Business Intelligence*. Indicadores de produção. Monitoramento da produção. OEE. Sistemas de informação.

ABSTRACT

In cosmetic filling lines with a low degree of automation, recording production on paper and spreadsheets hinders data consolidation, favors inconsistencies, and delays the analysis of results. This study aims to develop and evaluate a Proof of Concept (PoC) for monitoring production indicators on a filling line using Python, FastAPI, Pydantic, PyMongo, MongoDB, and Power BI. This applied research followed Design Science Research: the problem was identified on a real line, and the artifact was developed, demonstrated, and evaluated with simulated data.

[1] Graduanda em Sistemas de Informação pela Universidade de Araraquara (UNIARA), atua profissionalmente na indústria de cosméticos.

[2] Mestre em Engenharia de Produção pela Universidade de São Paulo (EESC-USP), é docente da Universidade de Araraquara (UNIARA).

[3] Doutora em Alimentos e Nutrição pela Universidade Estadual Paulista (FCFAr-UNESP), é docente da Universidade de Araraquara (UNIARA).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

The PoC receives records through a REST API, validates their structure and domain rules, handles duplicate and out-of-order events, stores events and indicators in MongoDB, and calculates yield, losses, availability, performance, quality, and Overall Equipment Effectiveness (OEE), flagging inconsistencies. In the technical evaluation, all 71 automated tests passed, the 17 HTTP requests, with 44 assertions, ran without failures, and the indicators matched the manual calculation. Divisions by zero produced flagged null indicators, and invalid inputs were rejected. The Power BI dashboard displayed the indicators per lot and weighted aggregate values of the lots without inconsistencies. The artifact meets the technical requirements defined. Data collection on the line, the reduction of information delay, and usefulness in real operation were not evaluated; therefore, no productivity gains are attributed to the solution.

KEYWORDS: *Industrial automation. Business Intelligence. Production indicators. Production monitoring. OEE. Information systems.*

RESUMEN

En líneas de envasado de cosméticos con bajo grado de automatización, el registro de la producción en papel y hojas de cálculo dificulta la consolidación de los datos, favorece inconsistencias y retrasa el análisis. Este trabajo tiene como objetivo desarrollar y evaluar una Prueba de Concepto (PoC) para monitorear indicadores de producción de una línea de envasado, con Python, FastAPI, Pydantic, PyMongo, MongoDB y Power BI. La investigación, aplicada, siguió la Design Science Research: el problema se identificó en una línea real y el artefacto se desarrolló, demostró y evaluó con datos simulados. La PoC recibe los registros mediante una API REST, valida su estructura y las reglas de dominio, trata eventos duplicados y fuera de orden, almacena eventos e indicadores en MongoDB y calcula rendimiento, pérdidas, disponibilidad, desempeño, calidad y Eficiencia General de los Equipos (Overall Equipment Effectiveness, OEE), señalizando las inconsistencias. En la evaluación técnica, se aprobaron las 71 pruebas automatizadas, las 17 solicitudes HTTP, con 44 aserciones, se ejecutaron sin fallos y los indicadores coincidieron con el cálculo manual. Las divisiones por cero generaron indicadores nulos señalizados y las entradas inválidas fueron rechazadas. El panel en Power BI mostró los indicadores por lote y valores agregados ponderados de los lotes sin inconsistencias. El artefacto cumple los requisitos técnicos definidos. La recolección en la línea, la reducción del retraso de la información y la utilidad en operación real no se evaluaron, por lo que no se atribuyen ganancias de productividad a la solución.

PALABRAS CLAVE: *Automatización industrial. Business Intelligence. Indicadores de producción. Monitoreo de la producción. OEE. Sistemas de información.*

1. INTRODUÇÃO

A transformação digital tem modificado a forma como as organizações industriais realizam suas atividades produtivas, administrativas e gerenciais, ampliando a capacidade de coleta, armazenamento, processamento e análise das informações produzidas no ambiente operacional. Nesse contexto, a Indústria 4.0 está associada à integração entre sistemas físicos e digitais, à conectividade, à automação e à utilização estruturada dos dados para aperfeiçoar processos e apoiar a tomada de decisão (KAGERMANN; WAHLSTER; HELBIG, 2013). Em ambientes nos quais ocorrem paradas, perdas, variações de produtividade e desvios em relação

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

ao planejamento, a disponibilidade de informações atualizadas pode favorecer a identificação de ocorrências e a análise de suas causas.

Além da disponibilidade dos dados, sua qualidade e rastreabilidade são aspectos relevantes para a utilização dos indicadores industriais. Quando os registros de produção dependem de anotações manuais, planilhas ou transferências sucessivas de informações, podem surgir inconsistências, duplicidades, informações incompletas e atrasos na consolidação dos resultados. Esses problemas aparecem em duas etapas diferentes: no registro do dado, quando o evento acontece, e no seu tratamento posterior, que envolve validar, consolidar e transformar os registros em indicadores. Esta pesquisa trata da segunda etapa.

Em linhas de envase, essa questão é particularmente relevante, porque diferentes eventos precisam ser acompanhados ao longo do processo, como quantidade planejada, quantidade produzida, perdas, tempo de operação e paradas de equipamentos. A associação dessas informações permite calcular indicadores que representam diferentes dimensões do desempenho produtivo, entre os quais se destaca a Eficiência Global dos Equipamentos (*Overall Equipment Effectiveness*, OEE), que considera disponibilidade, desempenho e qualidade em uma avaliação integrada das perdas do processo (NAKAJIMA, 1988; HANSEN, 2001). Para organizar e visualizar esses indicadores, ferramentas de *Business Intelligence* permitem reunir dados de diferentes períodos e facilitar a identificação de variações, desvios e situações que exijam investigação (SHARDA; DELEN; TURBAN, 2018).

Diante desse cenário, esta pesquisa busca desenvolver uma solução computacional de baixa complexidade para o tratamento das informações produtivas. A questão que orienta o trabalho é: como estruturar uma solução computacional que, a partir do registro dos dados de produção, automatize a validação, o cálculo e a visualização dos indicadores de uma linha de envase, centralizando as informações e mantendo a rastreabilidade dos registros?

Parte-se da hipótese de que uma arquitetura computacional simples, formada por uma camada de entrada e validação, uma camada de processamento, um banco de dados orientado a documentos e uma camada de visualização analítica, pode estruturar de maneira consistente o tratamento dos registros de uma linha de envase e permitir o cálculo padronizado dos indicadores definidos na pesquisa. A proposta não busca reproduzir um sistema industrial completo, mas demonstrar, por meio de uma Prova de Conceito (PoC), a viabilidade técnica dessa abordagem.

O objetivo geral deste trabalho é desenvolver e avaliar uma PoC para o monitoramento de indicadores de produção de uma linha de envase da indústria de cosméticos, utilizando Python, FastAPI, Pydantic, PyMongo, MongoDB e Power BI. Como objetivos específicos,

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

pretende-se: estruturar os dados necessários para representar eventos e informações produtivas; definir regras de validação para os dados recebidos; implementar uma API (Interface de Programação de Aplicações) para a entrada dos registros; armazenar eventos e indicadores em banco de dados orientado a documentos; calcular rendimento, perdas, disponibilidade, desempenho, qualidade e OEE; disponibilizar os indicadores em um painel analítico; avaliar o artefato em relação aos requisitos definidos; e analisar as limitações da solução em relação à utilização de dados reais e à operação em ambiente industrial.

A relevância da pesquisa está na aplicação integrada de conceitos de Sistemas de Informação, desenvolvimento de APIs, bancos de dados e *Business Intelligence* a um problema de gestão da produção, com uma solução de escopo reduzido, sem arquiteturas distribuídas ou tecnologias além do necessário. O recorte se limita ao tratamento das informações usadas no monitoramento dos indicadores. Não fazem parte do trabalho a captura automática dos dados, uma interface de registro para o operador, o controle de máquinas, a instrumentação física da linha ou a substituição dos sistemas de automação existentes.

2. REFERENCIAL TEÓRICO

2.1. Indústria 4.0 e transformação digital

A Indústria 4.0 está relacionada à transformação dos processos industriais pela integração entre sistemas físicos, tecnologias de comunicação, sistemas computacionais e análise de dados. Kagermann, Wahlster e Helbig (2013) a apresentam como uma iniciativa de modernização da manufatura. Essa transformação vai além da troca de atividades manuais por ferramentas computacionais, pois envolve mecanismos para coletar dados, processá-los e transformá-los em informações para acompanhamento e decisão, o que Schwab (2016) associa à maior integração entre tecnologias na Quarta Revolução Industrial. Lee, Bagheri e Kao (2015) destacam a integração entre o ambiente físico da fábrica e o espaço computacional nos sistemas ciberfísicos. Neste trabalho, a Indústria 4.0 serve como contexto conceitual para discutir a digitalização e o tratamento dos dados produtivos, sem a pretensão de implementar uma infraestrutura completa de manufatura inteligente.

Para pequenas e médias empresas, contudo, a adoção dessas tecnologias não é imediata. Ghobakhloo *et al.* (2022), em revisão sistemática sobre o tema, observam que essas empresas ainda enfrentam dificuldades nas decisões iniciais de adoção e que fatores como competências de conhecimento e prontidão digital da cadeia de valor favorecem o processo. Em micro e pequenas empresas, Silva *et al.* (2022) identificaram, em 60 artigos analisados, 16

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

tecnologias associadas à Indústria 4.0, nove barreiras à sua implementação e oito resultados de desempenho relatados, e destacam que a escolha das tecnologias a adotar é um desafio para os gestores. Esse cenário reforça o interesse por soluções de menor complexidade, como a proposta neste trabalho.

2.2. Monitoramento de produção

O monitoramento de produção consiste no acompanhamento sistemático das variáveis relevantes para a execução e a avaliação de um processo produtivo. Em uma linha de envase, informações como quantidade produzida, perdas, tempo de operação, paradas e situação do lote caracterizam o desempenho da produção. Ter essas informações em intervalos curtos pode ajudar a identificar desvios, mas o significado de “tempo real” depende do contexto, já que sistemas que controlam equipamentos diretamente têm requisitos diferentes dos sistemas voltados ao acompanhamento gerencial. Neste trabalho, o monitoramento é entendido em sentido analítico e operacional, e nenhuma latência é atribuída à solução sem que tenha sido medida.

2.3. Indicadores de desempenho e OEE

Os indicadores de desempenho são utilizados para representar quantitativamente diferentes características de um processo produtivo, como produtividade, perdas, tempos de parada, eficiência e qualidade. O OEE é uma métrica destinada à avaliação integrada do aproveitamento dos recursos produtivos: Hansen (2001) apresenta o indicador como instrumento para análise da eficiência dos equipamentos, enquanto Nakajima (1988) relaciona sua utilização ao contexto da *Total Productive Maintenance*. Em revisão sistemática de 186 artigos, Ng Corrales *et al.* (2020) descrevem a evolução do OEE e as modificações propostas ao modelo original, além de sua expansão para áreas como logística e serviços.

A composição do OEE considera três fatores: disponibilidade (*Availability*), desempenho (*Performance*) e qualidade (*Quality*). A disponibilidade representa a relação entre o tempo efetivo de operação e o tempo planejado; o desempenho, a relação entre a produção efetiva do equipamento e sua capacidade teórica durante o tempo de operação; e a qualidade, a proporção de produtos considerados bons em relação ao total produzido.

Em sua formulação conceitual, o OEE é apresentado por Nakajima (1988) e Hansen (2001). As expressões utilizadas neste trabalho seguem a formulação operacional consolidada na prática industrial (VORNE INDUSTRIES, 2026), cuja terminologia em inglês foi mantida por

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

corresponder diretamente aos campos implementados na PoC. Inicialmente, determinam-se a quantidade de unidades boas (*Good Count*) e o tempo efetivo de operação (*Run Time*):

$$Good\ Count = Total\ Count - Scrap\ Quantity \quad (1)$$

$$Run\ Time = Planned\ Production\ Time - Stop\ Time \quad (2)$$

A partir dessas grandezas, calculam-se os três fatores e o OEE:

$$Disponibilidade = Run\ Time / Planned\ Production\ Time \quad (3)$$

$$Desempenho = (Ideal\ Cycle\ Time \times Total\ Count) / Run\ Time \quad (4)$$

$$Qualidade = Good\ Count / Total\ Count \quad (5)$$

$$OEE = Disponibilidade \times Desempenho \times Qualidade \quad (6)$$

A separação entre esses fatores evita que indicadores de natureza diferente sejam tratados como equivalentes. Por isso, o rendimento do lote é usado como indicador complementar, e não como substituto do desempenho definido pela equação (4). Como o desempenho compara a produção efetiva com a capacidade teórica dada pelo tempo de ciclo ideal, ele não deveria passar de 100%. Quando passa, em geral o tempo de ciclo ideal foi parametrizado de forma incorreta (VORNE INDUSTRIES, 2026).

2.4. Rendimento e perdas

O rendimento é utilizado neste estudo como indicador de atendimento da produção planejada: a medida relaciona a quantidade produzida ao objetivo previamente estabelecido para o lote, permitindo verificar o nível de atendimento da meta produtiva.

$$Rendimento = (Produced\ Quantity / Expected\ Quantity) \times 100 \quad (7)$$

O percentual de perdas representa a parcela da produção associada a refugos ou a unidades não consideradas boas. Na PoC, ele é calculado pela relação entre a quantidade de perdas e a quantidade produzida:

$$Perdas = (Scrap\ Quantity / Produced\ Quantity) \times 100 \quad (8)$$

Na PoC, a quantidade total (*Total Count*) e a quantidade produzida (*Produced Quantity*) correspondem ao mesmo valor. Assim, o percentual de perdas é o complemento da qualidade, ou seja, Perdas = 100 × (1 − Qualidade). Os dois indicadores foram mantidos porque têm usos diferentes: a qualidade compõe o OEE, e o percentual de perdas mostra diretamente a parcela refugada do lote. Já o rendimento, dado pela equação (7), compara a produção realizada com a quantidade esperada para o lote. Quando a quantidade produzida é zero, os percentuais não são calculados por divisão direta, e o indicador é registrado como nulo e sinalizado.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

2.5. *Business Intelligence* e visualização de dados

O *Business Intelligence* (BI) reúne métodos e tecnologias para organizar, analisar e apresentar informações de apoio à decisão (SHARDA; DELEN; TURBAN, 2018). Em processos industriais, painéis com gráficos, cartões e filtros permitem acompanhar a variação dos indicadores ao longo do tempo e comparar lotes ou períodos. O valor desses painéis, no entanto, depende da qualidade dos dados que os alimentam, pois dados inconsistentes, incompletos ou desatualizados prejudicam a interpretação. Por esse motivo, neste trabalho o Power BI é usado apenas como camada de visualização, depois da validação e do processamento feitos na aplicação em Python.

2.6. Tecnologias de implementação e integração entre componentes

Python é uma linguagem bastante utilizada no processamento e na análise de dados, em razão do grande número de bibliotecas disponíveis (VANDERPLAS, 2022). Neste trabalho, ela é a linguagem da camada de processamento, onde ficam as regras de validação, transformação e cálculo dos indicadores. A entrada dos registros é feita por uma API REST (*Representational State Transfer*) desenvolvida com FastAPI, cuja documentação mostra o uso de modelos Pydantic para representar o corpo das requisições e validar tipos e campos (FASTAPI, [2026?]). Essa validação permite identificar campos ausentes, valores negativos ou estruturas incompatíveis antes que os registros sejam usados nos cálculos.

O MongoDB é um banco de dados orientado a documentos, que organiza as informações em bases, coleções e documentos. O acesso a partir do Python é feito pelo *driver* PyMongo, que oferece operações CRUD (*Create, Read, Update, Delete*), agregações e índices (MONGODB, 2026). Neste trabalho, o armazenamento separa os eventos de origem dos indicadores calculados, o que preserva os registros usados no cálculo e permite recalcular os indicadores quando necessário. A escolha do MongoDB está ligada ao caráter exploratório da PoC e à facilidade de representar eventos e indicadores como documentos, o que não significa que um banco de dados relacional seja inadequado para o problema.

2.7. Trabalhos relacionados

A arquitetura de cinco níveis proposta por Lee, Bagheri e Kao (2015) pressupõe um chão de fábrica instrumentado, em que os dados chegam ao ambiente computacional por meio de sensores instalados nos equipamentos. A principal diferença em relação a esta pesquisa está na origem do dado, que aqui é informado manualmente e enviado à aplicação por uma API, sem captura automática no equipamento.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

A arquitetura CBI4.0 (FAHEEM *et al.*, 2021) trata o problema em outra escala, organizando em múltiplas camadas a coleta de grandes volumes de dados para monitoramento e manutenção. Esta pesquisa situa-se no extremo oposto do espectro de complexidade: um único fluxo de entrada, um banco de dados local e um conjunto reduzido de indicadores.

Em Mishra *et al.* (2022), sensores, processamento de borda, computação em nuvem e aprendizado de máquina são combinados para monitorar e controlar o processo. Além da distância de infraestrutura, há uma distinção de propósito: aquele trabalho atua sobre o processo produtivo, enquanto a solução aqui proposta limita-se a organizar a informação a respeito dele.

Também com foco em baixo custo, Herrero *et al.* (2021) desenvolveram um sistema baseado em Raspberry Pi para medir o OEE em tempo real em linhas da indústria de alimentos, com custo inferior ao de sistemas proprietários semelhantes. Novamente, a diferença em relação a esta pesquisa está na origem do dado, capturado diretamente dos equipamentos.

O trabalho de Raval *et al.* (2026) é o mais próximo desta pesquisa, por adotar o OEE como indicador central e o monitoramento como objetivo. A distância está nos meios: os autores instrumentam uma linha de fabricação de rolamentos com IIoT (*Industrial Internet of Things*) para obter os dados na origem, ao passo que esta pesquisa parte do registro manual já existente e concentra-se no tratamento que ocorre depois dele.

Em conjunto, os trabalhos analisados mostram a relevância do monitoramento dos processos produtivos, da coleta estruturada de dados, do cálculo de indicadores e da visualização analítica, mas pressupõem a instrumentação da linha ou uma infraestrutura mais complexa, enquanto esta pesquisa trata o problema com tecnologias acessíveis. A contribuição do trabalho é discutida na subseção 4.9.

3. METODOLOGIA

3.1. Abordagem metodológica

A pesquisa é de natureza aplicada e foi conduzida segundo a *Design Science Research* (DSR), abordagem voltada à construção e à avaliação de artefatos que resolvem problemas práticos. Na DSR, a utilidade, a qualidade e a eficácia do artefato devem ser demonstradas por métodos de avaliação bem definidos (HEVNER *et al.*, 2004). Foi adotado o processo proposto por Peffers *et al.* (2007), formado por seis atividades: identificação do problema e motivação, definição dos objetivos da solução, projeto e desenvolvimento, demonstração, avaliação e comunicação. O artefato desenvolvido é a PoC descrita na seção 4, ou seja, um sistema implementado e executável.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

3.2. Contexto de aplicação

A pesquisa tem origem no acompanhamento de uma linha de envase de uma indústria de cosméticos de médio porte localizada no interior do estado de São Paulo, na qual a primeira autora atua profissionalmente, ambiente que motivou a definição do problema investigado. A linha apresenta baixo grau de automação e opera em turno único e direto. O fluxo produtivo reúne os postos e as atividades de montagem de caixa, montagem de cartucho, abastecimento e operação de granel, pesagem de frasco com granel, envase, rotulagem, encaixotamento, pesagem e paletização, com acompanhamento direto do operador de produção.

O registro da produção é feito atualmente em planilha e em uma ficha impressa que acompanha o lote, chamada ordem de produção (OP), na qual constam o procedimento e os indicadores acompanhados pela empresa, inclusive a meta de rendimento mínimo de 93%, parâmetro interno definido na própria OP. É nesse formato que está o problema tratado nesta pesquisa: as informações existem, mas ficam dispersas entre o papel e a planilha, sem validação nem consolidação automática. O processo atual também não utiliza o tempo de ciclo ideal de cada produto, necessário para calcular o desempenho do OEE; por isso, nesta etapa, esse valor é tratado como dado de entrada.

3.3. Etapas da pesquisa e natureza dos dados

As atividades da DSR foram conduzidas da seguinte maneira. A identificação do problema partiu da análise da linha de envase (subseção 3.2) e da revisão bibliográfica. Os objetivos da solução foram definidos como requisitos, derivados do problema e dos objetivos específicos, e estão apresentados no Quadro 4 junto com a sua avaliação. O projeto e o desenvolvimento resultaram na PoC descrita nas subseções 4.1 a 4.7. A demonstração consistiu na execução da PoC com cenários simulados e na construção do painel no Power BI Desktop, e a avaliação seguiu o procedimento descrito na subseção 3.4. A comunicação corresponde a este artigo.

Como não há autorização para a utilização de dados reais de produção, a PoC é executada com dados simulados e explicitamente identificados como tais, decisão que preserva a confidencialidade da empresa e mantém os testes reproduzíveis. Os resultados apresentados constituem, portanto, evidências do comportamento da PoC, e não medições do desempenho de uma linha industrial real.

A PoC foi delimitada para demonstrar o fluxo de entrada estruturada, validação, processamento, persistência, cálculo dos indicadores e visualização, sem sensores físicos e sem

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

integração direta com PLC/CLP. Tecnologias como MQTT, *Edge Computing*, Kafka, Kubernetes e arquiteturas distribuídas ficaram fora do escopo, para que a solução pudesse ser implementada, testada e reproduzida no ambiente da pesquisa. A aplicação e o banco de dados são executados em contêineres separados, orquestrados pelo Docker Compose.

3.4. Procedimento de avaliação

A avaliação foi do tipo artificial e somativa, voltada à eficácia técnica do artefato. Venable, Pries-Heje e Baskerville (2016) indicam essa estratégia quando a avaliação com usuários reais, em ambiente real, ainda não é viável ou tem custo elevado. Foram utilizados testes funcionais e estruturais e a demonstração por cenários, métodos previstos por Hevner *et al.* (2004), além de uma avaliação analítica de cada requisito com a respectiva evidência (Quadro 4). A utilidade do artefato em operação, com os usuários da linha, não foi avaliada nesta etapa.

Os cálculos foram verificados pela comparação entre os valores calculados manualmente, de forma independente do código, e os valores produzidos pela aplicação. A comparação incluiu a quantidade de unidades boas, o tempo efetivo de operação, a disponibilidade, o desempenho, a qualidade, o OEE, o rendimento e o percentual de perdas, conforme as equações (1) a (8). Os valores esperados foram arredondados a seis casas decimais, e a tolerância, definida antes dos testes, foi de diferença absoluta de até 0,00001 ou relativa de até 0,01%, prevalecendo a maior, para acomodar diferenças de arredondamento.

A verificação combinou quatro procedimentos: (i) testes automatizados unitários, de API e de integração com o MongoDB, executados com a biblioteca pytest; (ii) carga de cenários simulados (*seed*) pela mesma camada de serviço usada pela API, com conferência automática entre os valores esperados e os calculados; (iii) cenários controlados enviados diretamente à API em execução; e (iv) execução, pela ferramenta newman, de uma coleção de requisições do Postman com asserções sobre os códigos HTTP e os campos das respostas. Os cenários incluíram lote sem ocorrências, rendimento abaixo da meta, perdas, paradas, produção igual a zero, tempo planejado igual a zero, tempo de operação igual a zero, perdas maiores que a produção, valores negativos, eventos duplicados, lote incompleto, tempo de ciclo ideal superestimado, registros parciais recebidos em ordem e fora de ordem e banco de dados indisponível, além do caso de fronteira em que a produção corresponde exatamente à meta de 93%. Também foram verificadas a persistência dos dados após a reinicialização dos contêineres e a carga dos indicadores no painel do Power BI.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4. RESULTADOS E DISCUSSÃO

Os resultados a seguir referem-se à execução da PoC em ambiente local, com contêineres e dados simulados, e não são medições de uma linha industrial real.

4.1. Arquitetura da solução

A arquitetura é composta por quatro camadas: entrada e validação, processamento, persistência e visualização. A camada de entrada é uma API REST desenvolvida com FastAPI, e suas requisições são estruturadas por modelos Pydantic, responsáveis pela validação dos campos e dos tipos de dados. Foram implementados *endpoints* para receber eventos de produção e de parada, consultar os indicadores por lote e por período, consultar a rastreabilidade do lote e verificar a disponibilidade do serviço. Cada tipo de problema identificado gera um código de resposta HTTP (*Hypertext Transfer Protocol*) diferente (subseção 4.3). A documentação interativa desses contratos é gerada automaticamente no padrão OpenAPI (Swagger UI).

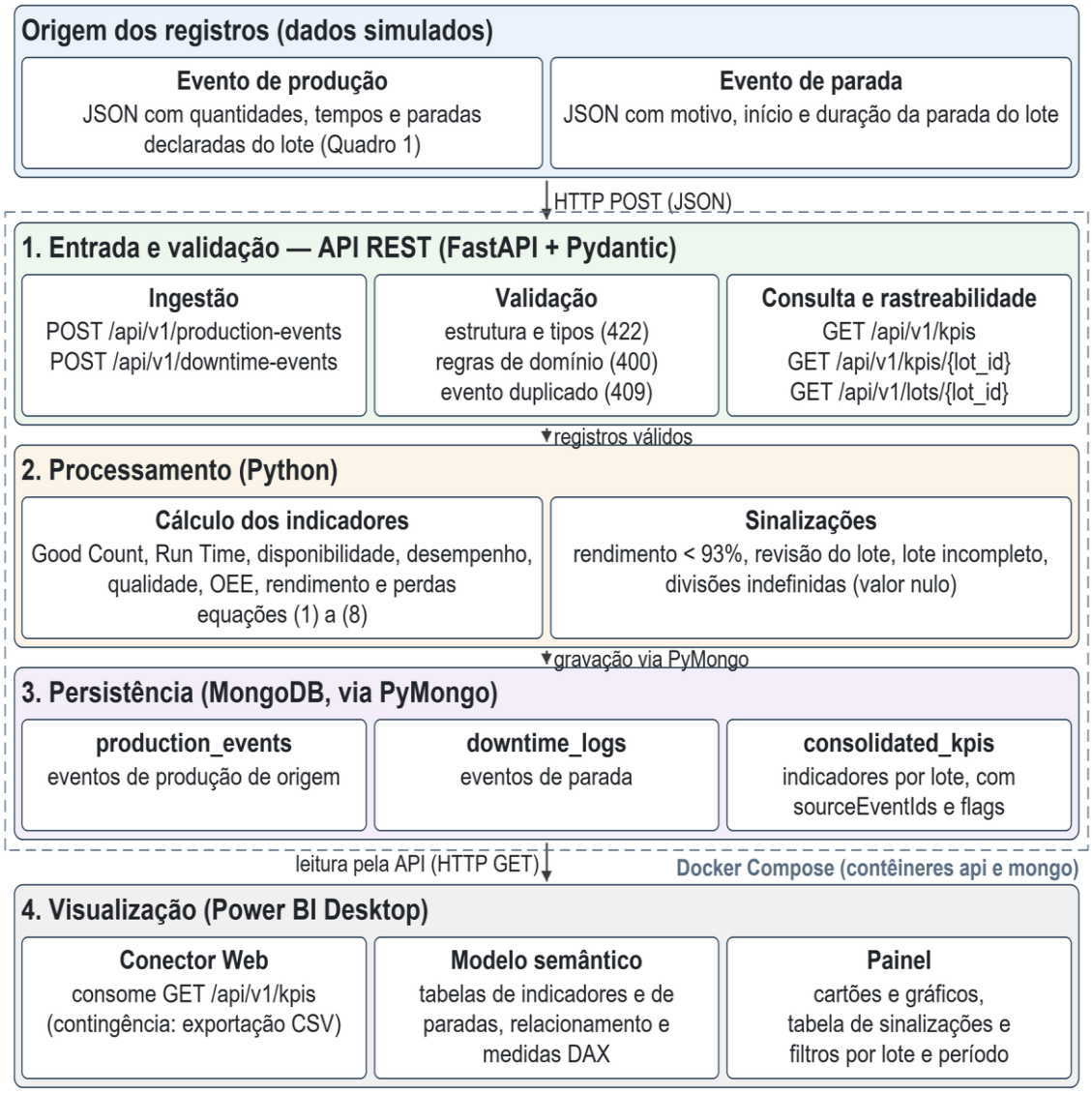
A Figura 1 apresenta a arquitetura geral da solução, com as quatro camadas, seus principais componentes e o fluxo dos registros, desde o envio dos eventos à API até o consumo dos indicadores pelo Power BI.

A camada de processamento, em Python, reúne as regras de transformação dos dados e de cálculo dos indicadores. Essas regras foram implementadas como funções sem estado interno e sem efeitos colaterais, o que facilita testá-las isoladamente. A persistência usa o MongoDB, acessado pelo PyMongo, para armazenar os eventos de produção, os eventos de parada e os indicadores consolidados, e a apresentação dos indicadores fica a cargo do Power BI.

O fluxo começa no recebimento do registro pela API, passa pela validação dos dados e pelo processamento das regras de negócio e dos indicadores, segue para o MongoDB e, depois, fica disponível para o Power BI. Como as regras de negócio ficam concentradas na camada de processamento, separada do armazenamento e da visualização, cada componente tem uma responsabilidade definida, e uma inconsistência identificada na entrada pode ser rastreada até o indicador armazenado.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 1. Arquitetura da solução: camadas, componentes e fluxo dos dados



Fonte: elaboração própria (2026).

Como evidência da implementação dos contratos da API, a Figura 2 apresenta a documentação OpenAPI/Swagger gerada automaticamente pela API, que exibe os seis *endpoints* implementados, agrupados pelas etiquetas infra, ingestao e consulta.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 2. Documentação OpenAPI/Swagger com os endpoints implementados na API (grupos infra, ingestao e consulta)



Fonte: elaboração própria (2026).

4.2. Entrada e estrutura dos dados

A entrada da PoC é formada por registros estruturados em formato JSON (*JavaScript Object Notation*), cada um representando um evento associado a uma linha, lote ou produto. O Quadro 1 apresenta um exemplo de registro utilizado para demonstrar a estrutura da entrada.

Quadro 1. Exemplo de payload JSON para registro de evento de produção

```
{
  "eventId": "evt-001",
  "timestamp": "2026-09-10T18:32:15Z",
  "lineId": "ENV-01",
  "lotId": "LOT-001",
  "productId": "PROD-300",
  "eventType": "lot_snapshot",
  "expectedQty": 1000,
  "totalProducedQty": 960,
  "scrapQty": 20,
  "plannedProductionTimeSec": 3600,
  "stopTimeSec": 480,
  "idealCycleTimeSec": 3.0,
  "stopCount": 2,
  "statusLine": "RUNNING"
}
```

Fonte: elaboração própria (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

O *payload* apresentado é uma estrutura de referência com os dados necessários ao cálculo dos indicadores. Os valores não vêm de um equipamento real: a quantidade esperada e o tempo planejado representam uma ordem de produção de referência, e o tempo de ciclo ideal é um parâmetro do produto, informado junto com o evento. Como a empresa não utiliza esse parâmetro (subseção 3.2), adotou-se o valor de 3,0 s por unidade, escolhido para que o desempenho de todos os lotes simulados válidos ficasse abaixo de 100%, como prevê a definição do indicador. Um cenário específico, com tempo de ciclo superestimado, foi incluído para mostrar como esse erro de parametrização é sinalizado (subseção 4.5).

Essa estrutura corresponde ao contrato implementado no modelo Pydantic ProductionEventIn: a documentação gerada automaticamente pela API para o *endpoint* de ingestão apresenta os mesmos 14 campos do Quadro 1, e a descrição do modelo no código faz referência explícita a esse quadro.

Os dados podem ser divididos em três categorias: os planejados, definidos antes da produção, como a quantidade esperada e o tempo planejado; os observados, registrados durante o processo, como a quantidade produzida, as perdas e as paradas; e os derivados, que são os indicadores calculados. Cada registro do tipo lot_snapshot traz os valores acumulados do lote até o momento do envio. A API aceita vários registros do mesmo lote, por exemplo um a cada hora, e recalcula os indicadores a cada recebimento usando o registro mais recente. Dessa forma, os indicadores podem ser consultados antes do encerramento do lote. A PoC, porém, não inclui uma interface de registro para o operador. Nesta etapa, os registros são enviados à API por requisições HTTP, e a forma de coleta na linha ficou fora do escopo (subseção 4.9).

4.3. Validação e tratamento das informações

Após o recebimento de um registro, a primeira etapa consiste na validação de sua estrutura. O modelo Pydantic associado à requisição verifica tipos, campos obrigatórios e valores com restrições definidas na aplicação, e a estrutura do *payload* é validada de forma estrita, com rejeição de campos não previstos no contrato de entrada.

Além da validação estrutural, são aplicadas regras de domínio. Quantidades e tempos negativos são rejeitados já na validação estrutural (HTTP 422), e a quantidade de perdas não pode ser maior que a quantidade produzida (HTTP 400). Quando a quantidade produzida fica abaixo de 93% da quantidade esperada, meta interna registrada na ordem de produção (subseção 3.2), o lote é sinalizado para revisão operacional. A mesma sinalização é aplicada quando o lote está incompleto ou apresenta inconsistências nos dados (subseção 4.6). Nesta

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

etapa, o campo flags.needsJustification apenas indica que o lote precisa de revisão, e a justificativa em si (texto, autor e decisão) não é armazenada pela API.

Os eventos duplicados são identificados pelo identificador do evento (eventId), cuja unicidade é garantida por um índice único no MongoDB. Os registros recebidos fora da ordem cronológica são mantidos e sinalizados, sem alterar o cálculo, que continua usando o registro mais recente. A estratégia adotada para as anomalias é detectar, sinalizar e registrar as ocorrências. O objetivo não é apagar informações inconsistentes, e sim evitar que elas sejam usadas nos cálculos sem que a situação seja identificada.

Esses tratamentos aparecem no contrato HTTP da API. O *endpoint* de ingestão de eventos de produção responde com códigos diferentes para falhas estruturais do *payload* (422), violações de regras de domínio (400), eventos duplicados (409) e indisponibilidade do MongoDB (503), e todos esses casos foram verificados na avaliação (subseção 4.8).

4.4. Persistência e modelagem dos dados

A modelagem lógica do MongoDB tem três coleções: production_events, com os eventos de produção; downtime_logs, com os eventos de parada, seu motivo, duração e momento; e consolidated_kpis, com os indicadores calculados para cada lote. As coleções se relacionam pelos identificadores lineId e lotId. As coleções de eventos têm índice único sobre eventId, e a de indicadores, sobre a combinação de lotId e lineId, além de índices auxiliares para as consultas por lote, linha e período. O Quadro 2 resume a estrutura dos documentos de consolidated_kpis, que corresponde ao modelo de resposta ConsolidatedKpi documentado pela API.

Quadro 2. Estrutura dos documentos da coleção consolidated_kpis, por grupo de campos

Grupo	Campos	Conteúdo
Identificação	lotId, lineId, productId, timestamp, eventType	Lote, linha, produto, data e hora e tipo do registro de origem mais recente
Quantidades	expectedQty, producedQty, scrapQty, goodQty	Quantidades esperada, produzida, de perdas e de unidades boas
Tempos	plannedProductionTimeSec, stopTimeSec, runTimeSec, idealCycleTimeSec, plannedTimeMin, runTimeMin, stopTimeMin	Tempos planejado, de parada, de operação e de ciclo ideal, em segundos e em minutos
Indicadores	yieldPct, scrapPct, availability, performance, quality, oee	Indicadores das equações (3) a (8), nulos quando indefinidos
Paradas	stopCount, downtimeRecordedCount, downtimeRecordedSec, topStopReasons	Paradas declaradas, eventos de parada detalhados recebidos e principais motivos

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Grupo	Campos	Conteúdo
Tendência	avgLotTimeMin	Média móvel do tempo de operação dos últimos cinco lotes da mesma linha e produto
Rastreabilidade	sourceEventIds, computedAt	Eventos de origem usados no cálculo e momento do recálculo
Sinalizações	flags.under93Pct, flags.needsJustification, flags.incompleteLot, flags.dataIssues	Rendimento abaixo da meta, necessidade de revisão, lote incompleto e lista de inconsistências

Fonte: elaboração própria (2026).

4.5. Processamento dos indicadores de produção

Após a validação, os dados são submetidos às regras de cálculo implementadas na camada de processamento, que seguem as equações (1) a (8) apresentadas nas subseções 2.3 e 2.4. Inicialmente, são determinados a quantidade de unidades boas, pela equação (1), e o tempo efetivo de operação, pela equação (2); a partir deles, calculam-se a disponibilidade, o desempenho, a qualidade e o OEE, pelas equações (3) a (6). O rendimento e o percentual de perdas são tratados separadamente, como indicadores complementares, pelas equações (7) e (8).

O desempenho, definido pela equação (4), não é truncado em 100%, para não alterar a definição matemática do indicador. Como valores acima desse limite indicam tempo de ciclo ideal incorreto (VORNE INDUSTRIES, 2026), a aplicação registra a ocorrência como inconsistência (performance_above_one) e sinaliza o lote para revisão.

Quando o cálculo envolve divisão por zero, situação possível nas equações (3), (4), (5), (7) e (8), a aplicação não executa a divisão. O indicador é registrado como nulo e a ocorrência é sinalizada. Também foram implementadas uma média móvel do tempo de operação dos últimos cinco lotes da mesma linha e produto, como indicador complementar de tendência, e a consolidação das paradas por motivo, que mostra as principais causas de interrupção de cada lote.

4.6. Tratamento de anomalias e rastreabilidade

Além das rejeições descritas na subseção 4.3, a aplicação sinaliza as divisões indefinidas nos indicadores (quando a quantidade produzida, o tempo planejado ou o tempo de operação são iguais a zero), as diferenças entre o tempo de parada informado e a soma dos eventos de parada detalhados, os registros recebidos fora da ordem cronológica e o desempenho acima de 100%. Um lote é considerado incompleto quando o número de paradas declaradas é maior que o número de eventos de parada detalhados recebidos. Esse critério é uma heurística

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

da PoC, definida apenas com os campos do contrato de entrada, e não uma regra do processo industrial.

Os indicadores de registros inconsistentes não são apresentados como resultados válidos sem que a situação seja identificada, e o painel destaca os lotes sinalizados. A rastreabilidade é garantida pela preservação dos registros de origem e pela associação de cada indicador aos identificadores do lote e dos eventos que o originaram (campo sourceEventIds).

4.7. Integração com o Power BI

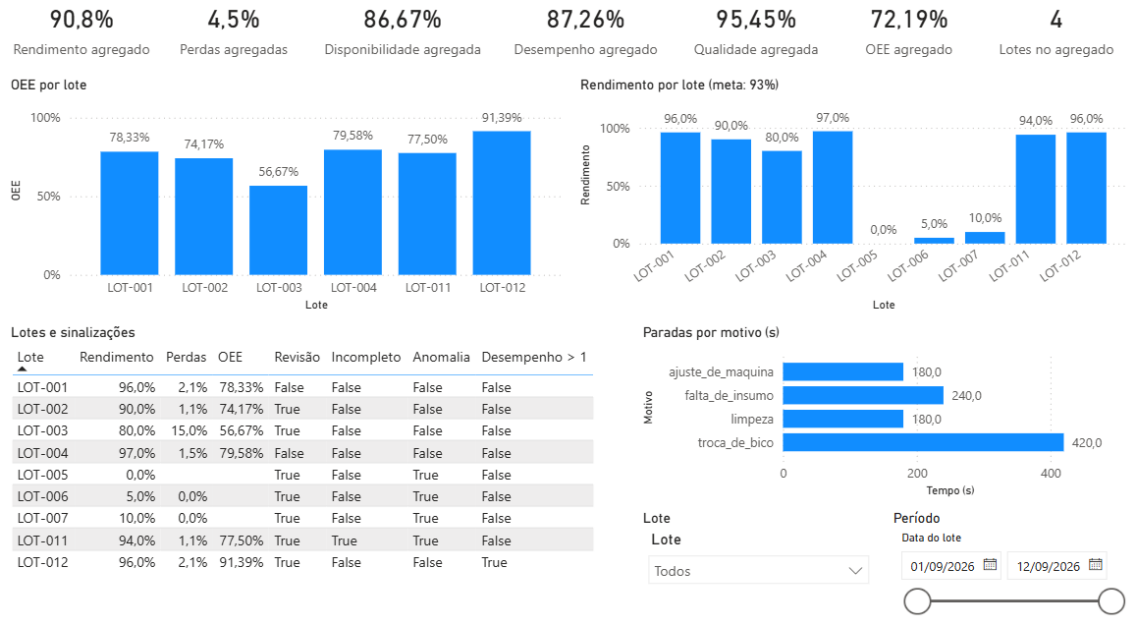
Após o processamento, os indicadores são disponibilizados para a camada de visualização. O conector para MongoDB descrito na documentação da Microsoft, baseado na interface SQL do MongoDB Atlas, exige um banco de dados federado no MongoDB Atlas, que é um serviço em nuvem (MICROSOFT, 2025). Por isso, a integração foi feita por um *endpoint* HTTP que entrega os indicadores em formato compatível com o conector Web do Power BI, com a exportação para arquivo CSV como alternativa de contingência. Nos dois casos, o MongoDB continua sendo a fonte dos dados. O *endpoint* GET /api/v1/kpis permite filtrar por lote, linha, produto e período e paginar os resultados.

Na avaliação, esse *endpoint* respondeu com HTTP 200 e devolveu os nove lotes simulados em formato JSON, e a exportação para CSV também foi executada. A partir dessa fonte, foi criado um projeto do Power BI Desktop (formato PBIP) com um modelo semântico de duas tabelas, uma com os indicadores por lote e outra com as paradas por motivo, um relacionamento entre elas e 26 medidas DAX. Os indicadores de cada lote chegam prontos da API e não são recalculados no Power BI. Depois da atualização dos dados, o modelo carregou os nove lotes e as cinco paradas, e as consultas reproduziram os valores do lote de referência (OEE de 0,783333 e desempenho de 0,923077).

Com esse modelo, o painel foi montado em uma página com 13 visuais (Figura 3): sete cartões com os valores agregados de rendimento, perdas, disponibilidade, desempenho, qualidade e OEE e com o número de lotes considerados; gráficos de colunas com o OEE e o rendimento de cada lote; uma tabela com as sinalizações de revisão, lote incompleto, anomalia de dados e desempenho acima de 100%; um gráfico das paradas por motivo; e filtros por lote e por período.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 3. Painel de monitoramento no Power BI Desktop com os nove lotes simulados, após atualização a partir da API



Lote	Rendimento	Perdas	OEE	Revisão	Incompleto	Anomalia	Desempenho > 1
LOT-001	96,0%	2,1%	78,33%	False	False	False	False
LOT-002	90,0%	1,1%	74,17%	True	False	False	False
LOT-003	80,0%	15,0%	56,67%	True	False	False	False
LOT-004	97,0%	1,5%	79,58%	False	False	False	False
LOT-005	0,0%			True	False	True	False
LOT-006	5,0%	0,0%		True	False	True	False
LOT-007	10,0%	0,0%		True	False	True	False
LOT-011	94,0%	1,1%	77,50%	True	True	True	False
LOT-012	96,0%	2,1%	91,39%	True	False	False	True

Fonte: elaboração própria (2026).

Os cartões não mostram a média simples dos indicadores dos lotes, que daria o mesmo peso a lotes de tamanhos diferentes. Cada valor agregado é calculado pela razão entre as somas das grandezas de base dos lotes; a disponibilidade agregada, por exemplo, é a soma dos tempos de operação dividida pela soma dos tempos planejados. O agregado considera apenas os lotes sem inconsistência de dados, quatro dos nove lotes simulados, e, como todos os fatores usam o mesmo conjunto de lotes, o OEE agregado (72,19%) é igual ao produto da disponibilidade (86,67%), do desempenho (87,26%) e da qualidade (95,45%) agregados. Esses valores coincidem com o cálculo manual. Nos gráficos e na tabela, os lotes com indicador indefinido não aparecem com valor zero. O lote com tempo de ciclo superestimado (LOT-012) aparece com o maior OEE, mas sinalizado para revisão, o que reforça que esse valor não deve ser interpretado isoladamente. O painel foi pensado para permitir a consulta, por lote e por período, do rendimento, das perdas, do OEE e das causas de parada, mas sua utilidade para os usuários da linha não foi avaliada.

Em relação à atualização dos dados, não foram medidas a latência do monitoramento nem a frequência de atualização em produção, que dependeriam da publicação no serviço do Power BI e de um *gateway* de dados local, não testados nesta etapa.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4.8. Avaliação com dados simulados

A PoC foi executada com a API e o MongoDB em contêineres, ambos com verificação de saúde (*healthcheck*) bem-sucedida, e a API só passou a atender depois que o banco ficou disponível. Os documentos gravados (9 eventos de produção, 5 eventos de parada e 9 indicadores consolidados) foram preservados após a reinicialização e a recriação dos contêineres, graças ao volume persistente. A Tabela 1 resume as verificações realizadas.

Tabela 1. Síntese da avaliação da PoC com dados simulados

Verificação	Escopo	Resultado
Testes automatizados (unitários, de API e de integração com o MongoDB)	71 testes, 4 deles de integração: fluxo completo, índice único, recálculo após parada e leitura após reconexão	71 aprovados
Carga de cenários simulados (*seed*)	12 cenários: 9 lotes aceitos e 3 entradas rejeitadas (HTTP 400, 422 e 409)	72 comparações entre cálculo manual e aplicação, sem divergência
Coleção de requisições HTTP (Postman/newman)	17 requisições e 44 asserções	0 falhas
Cenários controlados enviados diretamente à API	13 cenários (Quadro 3)	comportamento esperado em todos
Persistência no MongoDB	reinicialização e recriação dos contêineres	9 + 5 + 9 documentos preservados
Painel no Power BI	modelo com 2 tabelas, 1 relacionamento e 26 medidas; página com 13 visuais	9 lotes e 5 paradas carregados a partir da API; valores conferidos

Fonte: elaboração própria (2026), com base nos registros da avaliação.

A Tabela 2 mostra a conferência do lote de referência descrito no Quadro 1. Os oito valores calculados manualmente coincidem com os calculados pela aplicação e gravados no MongoDB. Nenhuma inconsistência foi registrada para esse lote, que também não foi marcado para revisão, pois o rendimento (96%) supera a meta de 93% e as duas paradas declaradas correspondem aos dois eventos de parada detalhados.

Tabela 2. Conferência do lote de referência (Quadro 1): cálculo manual e valor da aplicação

Indicador	Equação	Cálculo manual	Aplicação
Good Count (unidades)	(1)	940	940
Run Time (s)	(2)	3.120	3.120
Disponibilidade	(3)	0,866667	0,866667
Desempenho	(4)	0,923077	0,923077

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Indicador	Equação	Cálculo manual	Aplicação
Qualidade	(5)	0,979167	0,979167
OEE	(6)	0,783333	0,783333
Rendimento (%)	(7)	96,0	96,0
Perdas (%)	(8)	2,083333	2,083333

Fonte: elaboração própria (2026), com base nos registros da avaliação.

A consulta de rastreabilidade (GET /api/v1/lots/{lot_id}) desse lote retornou o evento de origem, identificado como simulado ("simulated": true), e o indicador consolidado com a referência a esse evento (campo sourceEventIds). A inspeção direta do banco confirmou as três coleções e os índices únicos.

O Quadro 3 sintetiza os cenários controlados enviados diretamente à API. Em nenhum caso de divisão por zero houve erro de execução ou valor zero atribuído silenciosamente: o indicador correspondente foi registrado como nulo e sinalizado. As entradas inconsistentes foram rejeitadas com o código HTTP correspondente à natureza do problema, e o reenvio de um evento já registrado não gerou nova gravação. Na carga de cenários simulados, a entrada com quantidade negativa foi rejeitada na validação estrutural (HTTP 422) e o lote com paradas declaradas sem detalhamento foi sinalizado como incompleto; o caso de fronteira de exatamente 93% foi verificado em teste unitário, sem sinalização indevida.

Quadro 3. Cenários controlados enviados diretamente à API e comportamento observado

Cenário	Entrada (resumo)	Resultado observado (HTTP)
Lote sem ocorrências	produção 980; perdas 10; tempo planejado 3.600 s; parada 300 s	OEE 0,808333; desempenho 0,890909; sem sinalização (201)
Rendimento abaixo de 93%	produção 900; quantidade esperada 1.000	rendimento 90%; lote sinalizado para revisão (201)
Perdas	produção 800; perdas 150	*Good Count* 650; perdas 18,75%; qualidade 0,8125 (201)
Produção zero	produção 0	qualidade, perdas e OEE nulos e sinalizados; desempenho 0 (201)
Tempo planejado zero	tempo planejado 0	disponibilidade, desempenho e OEE nulos e sinalizados (201)
Run Time zero	tempo planejado 600 s; parada 600 s	*Run Time* 0; disponibilidade 0; desempenho e OEE nulos e sinalizados (201)
Perdas maiores que a produção	produção 300; perdas 400	rejeitado por violação de regra de domínio (400)
Evento duplicado	reenvio do mesmo eventId	rejeitado, sem nova gravação (409)
Parada	evento de parada de lote existente	registrado em downtime_logs; indicadores recalculados (201)

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Cenário	Entrada (resumo)	Resultado observado (HTTP)
Tempo de ciclo ideal superestimado	ciclo ideal 3,5 s; produção 960; parada 480 s	desempenho 1,076923, sem truncamento; inconsistência registrada; lote sinalizado para revisão (201)
Registros parciais acumulados	480 unidades às 8h30 e 960 unidades às 9h, no mesmo lote	indicadores recalculados pelo registro mais recente; OEE 0,783333; sem sinalização (201)
Registro fora de ordem	registro das 9h recebido antes do registro das 8h30	cálculo mantido no registro mais recente; evento fora de ordem sinalizado; lote sinalizado para revisão (201)
MongoDB indisponível	banco de dados parado durante a requisição	ingestão rejeitada como indisponibilidade de armazenamento (503); verificação de saúde degradada (503)

Fonte: elaboração própria (2026), com base nos registros da avaliação.

O Quadro 4 apresenta a avaliação analítica dos requisitos definidos a partir do problema (subseção 3.3), com a evidência de cada um. Os requisitos ligados ao tratamento dos registros foram atendidos nos cenários avaliados. O recálculo durante o lote foi atendido apenas em parte, porque depende da frequência com que os registros são enviados. Já a redução do esforço de registro e do atraso entre o evento e a informação não foi avaliada, pois depende de uma interface de coleta e do uso em ambiente real.

Quadro 4. Requisitos do artefato e avaliação com base nas evidências obtidas

Requisito	Evidência	Situação
R1. Receber registros de produção e de parada em formato padronizado	API REST com contratos OpenAPI (Figura 2) e validação por modelos Pydantic	Atendido
R2. Rejeitar entradas inválidas e indicar a natureza da falha	Respostas 422 (estrutura) e 400 (regra de domínio) nos testes e cenários (Quadro 3)	Atendido
R3. Impedir o registro duplicado de eventos	Índice único sobre eventId; resposta 409 sem nova gravação	Atendido
R4. Calcular os indicadores pelas equações (1) a (8) sem atribuir valor a indicadores indefinidos	72 comparações entre cálculo manual e aplicação, sem divergência; divisões por zero registradas como nulas	Atendido
R5. Sinalizar os lotes que exigem revisão	Sinalização de rendimento abaixo da meta, lote incompleto, evento fora de ordem e desempenho acima de 100%	Atendido

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Requisito	Evidência	Situação
R6. Preservar a rastreabilidade entre indicadores e registros de origem	Campo sourceEventIds; consulta de rastreabilidade por lote; dados preservados após a recriação dos contêineres	Atendido
R7. Disponibilizar os indicadores por lote e por período	Painel no Power BI com filtros de lote e de período (Figura 3)	Atendido no ambiente avaliado; não avaliado com usuários
R8. Atualizar os indicadores durante o lote	Recálculo a cada registro parcial acumulado (Quadro 3)	Parcialmente atendido: depende da frequência de registro
R9. Reduzir o esforço de registro e o atraso entre o evento e a informação	Não há interface de coleta para o operador nem uso em ambiente real	Não avaliado

Fonte: elaboração própria (2026).

4.9. Discussão, contribuição e limitações

Quanto à segurança, as credenciais de acesso ao MongoDB não ficam no código-fonte. A aplicação lê a configuração a partir de variáveis de ambiente, e o projeto mantém apenas um arquivo de exemplo, sem credenciais reais. Autenticação, autorização, alta disponibilidade, contingência *offline* e recuperação automática ficaram fora do escopo e precisarão ser incorporadas se a solução for usada em operação.

Em relação à hipótese, os resultados indicam que, no contexto avaliado, a arquitetura de quatro camadas organizou de forma consistente o tratamento dos registros. Os indicadores coincidiram com o cálculo manual, as entradas inconsistentes foram rejeitadas ou sinalizadas, as divisões indefinidas apareceram como valores nulos, a rastreabilidade entre eventos e indicadores foi preservada e o painel apresentou os indicadores e as sinalizações vindos da API. Do ponto de vista da DSR, trata-se de uma avaliação artificial, que demonstra a eficácia técnica do artefato em relação aos requisitos (Quadro 4). A utilidade para a gestão da linha dependeria de uma avaliação em ambiente real, com os usuários, e ainda não foi demonstrada.

Considerando a classificação de Gregor e Hevner (2013), a contribuição da pesquisa está em dois níveis: o próprio artefato e um conjunto de princípios de projeto que podem ser aplicados em outras linhas com registro manual e pouca automação. O primeiro princípio é separar os eventos de origem dos indicadores calculados, para que cada valor possa ser recalculado e rastreado. O segundo é registrar como nulo, e sinalizar, todo indicador que não puder ser calculado, em vez de atribuir zero. O terceiro é validar a entrada em duas etapas,

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

estrutural e de regras de domínio, com códigos de resposta diferentes e controle de duplicidade pelo identificador do evento. O quarto é sinalizar as inconsistências sem corrigi-las automaticamente, como no caso do desempenho acima de 100%, que indica erro de parametrização. O quinto é manter o cálculo dos indicadores na aplicação e deixar para a camada de visualização apenas a agregação dos lotes por razão de somas. Esses princípios não dependem das tecnologias escolhidas e podem orientar soluções semelhantes de baixo custo.

Os resultados se restringem a dados simulados e a um ambiente local. Como a PoC não inclui uma interface de coleta para o operador, a redução do esforço de registro e do atraso entre o evento e a informação, que motivou a pesquisa, não pôde ser avaliada. Também não houve validação com dados industriais reais nem em ambiente produtivo, e o painel não foi avaliado com usuários. Além disso, o tempo de ciclo ideal foi definido pelos autores, por não ser utilizado no processo atual da empresa; o critério de lote incompleto é uma heurística da PoC; e a API não possui autenticação, por estar fora do escopo.

5. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo desenvolver e avaliar uma Prova de Conceito para o monitoramento de indicadores de produção de uma linha de envase da indústria de cosméticos, utilizando Python, FastAPI, Pydantic, PyMongo, MongoDB e Power BI. Seguindo a *Design Science Research*, a pesquisa resultou em um artefato que recebe os registros por uma API, valida a estrutura e as regras de domínio, trata duplicidades e registros fora de ordem, armazena eventos e indicadores de forma rastreável e calcula os indicadores, sinalizando as inconsistências e os lotes que precisam de revisão. Na avaliação com dados simulados, os 71 testes automatizados foram aprovados, as 17 requisições HTTP foram executadas sem falhas, os indicadores coincidiram com o cálculo manual em todos os lotes conferidos e o painel no Power BI apresentou os indicadores por lote e os valores agregados obtidos da API.

Os objetivos específicos foram atendidos quanto à estruturação dos dados, às regras de validação, à API, ao armazenamento, ao cálculo dos indicadores, ao painel e à avaliação do artefato em relação aos requisitos, com a ressalva de que o recálculo durante o lote depende da frequência dos registros. Além do artefato, o trabalho apresenta princípios de projeto que podem ser aplicados em outras linhas com registro manual e pouca automação.

Como os resultados se restringem a dados simulados e a um ambiente local, não se atribuem à solução ganhos de produtividade, redução de erros ou de atraso na informação, nem

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

desempenho em ambiente industrial real. Como trabalhos futuros, sugerem-se o desenvolvimento de uma interface de registro para o operador ou a integração com contadores da linha, a avaliação do painel com os usuários em ambiente real, a medição do atraso entre o evento e a informação, a definição do tempo de ciclo ideal de cada produto, o uso de dados reais, mediante autorização, e a inclusão de autenticação, autorização e registro de justificativas.

FINANCIAMENTO E CONFLITO DE INTERESSES

Esta pesquisa não recebeu financiamento. Os autores declaram não haver conflito de interesses.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Os autores utilizaram a ferramenta de inteligência artificial Claude (Anthropic) como apoio no desenvolvimento do código da Prova de Conceito e dos scripts que geram o painel no Power BI, na redação e na revisão do texto e na adequação às normas da revista, inclusive na etapa de resposta aos pareceres. A definição do problema, dos objetivos, do escopo, dos requisitos e dos cenários de teste, assim como a revisão final do conteúdo, é de responsabilidade dos autores, que assumem integralmente o trabalho apresentado.

REFERÊNCIAS

FAHEEM, Muhammad; BUTT, Rizwan Aslam; ALI, Rashid; RAZA, Basit; NGADI, Md. Asri; GUNGOR, Vehbi Cagri. CBI4.0: A cross-layer approach for big data gathering for active monitoring and maintenance in the manufacturing industry 4.0. **Journal of Industrial Information Integration**, v. 24, 100236, 2021. DOI: https://doi.org/10.1016/j.jii.2021.100236.

FASTAPI. **Request Body**. [S. l.]: FastAPI, [2026?]. Disponível em: https://fastapi.tiangolo.com/tutorial/body/. Acesso em: 10 set. 2026.

GHOBAKHLOO, Morteza; IRANMANESH, Mohammad; VILKAS, Mantas; GRYBAUSKAS, Andrius; AMRAN, Azlan. Drivers and barriers of Industry 4.0 technology adoption among manufacturing SMEs: a systematic review and transformation roadmap. **Journal of Manufacturing Technology Management**, v. 33, n. 6, p. 1029-1058, 2022. DOI: https://doi.org/10.1108/JMTM-12-2021-0505.

GREGOR, Shirley; HEVNER, Alan R. Positioning and presenting design science research for maximum impact. **MIS Quarterly**, v. 37, n. 2, p. 337-355, 2013. DOI: https://doi.org/10.25300/MISQ/2013/37.2.01.

HANSEN, Robert C. **Overall Equipment Effectiveness**: A Powerful Production/Maintenance Tool for Increased Profits. New York: Industrial Press, 2001.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

HERRERO, Angel C.; SANGUESA, Julio A.; MARTINEZ, Francisco J.; GARRIDO, Piedad; CALAFATE, Carlos T. Mitigating electromagnetic noise when using low-cost devices in Industry 4.0. **IEEE Access**, v. 9, p. 63267-63282, 2021. DOI: https://doi.org/10.1109/ACCESS.2021.3074588.

HEVNER, Alan R.; MARCH, Salvatore T.; PARK, Jinsoo; RAM, Sudha. Design science in information systems research. **MIS Quarterly**, v. 28, n. 1, p. 75-106, 2004. DOI: https://doi.org/10.2307/25148625.

KAGERMANN, Henning; WAHLSTER, Wolfgang; HELBIG, Johannes. **Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0**: Final Report of the Industrie 4.0 Working Group. Frankfurt: Forschungsunion, 2013.

LEE, Jay; BAGHERI, Behrad; KAO, Hung-An. A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. **Manufacturing Letters**, v. 3, p. 18-23, 2015. DOI: https://doi.org/10.1016/j.mfglet.2014.12.001.

MICROSOFT. **Conector de interface SQL do Atlas do MongoDB**. [S. l.]: Microsoft Learn, 2025. Disponível em: https://learn.microsoft.com/pt-br/power-query/connectors/mongodb-atlas-sql-interface. Acesso em: 12 nov. 2026.

MISHRA, Debasish; PRIYADARSHI, Ashok; DAS, Sarthak M.; SHREE, Sristi; GUPTA, Abhinav; PAL, Surjya K.; CHAKRAVARTY, Debashish. Industry 4.0 application in manufacturing for real-time monitoring and control. **Journal of Dynamics, Monitoring and Diagnostics**, v. 1, n. 3, p. 176-187, 2022. DOI: https://doi.org/10.37965/jdmd.2022.118.

MONGODB. **MongoDB PyMongo Documentation**. [S. l.]: MongoDB, Inc., c2026. Disponível em: https://www.mongodb.com/docs/languages/python/pymongo-driver/current/. Acesso em: 10 set. 2026.

NAKAJIMA, Seiichi. **Introduction to TPM**: Total Productive Maintenance. Cambridge: Productivity Press, 1988.

NG CORRALES, Lisbeth del Carmen; LAMBÁN, María Pilar; HERNANDEZ KORNER, Mario Enrique; ROYO, Jesús. Overall Equipment Effectiveness: systematic literature review and overview of different approaches. **Applied Sciences**, v. 10, n. 18, 6469, 2020. DOI: https://doi.org/10.3390/app10186469.

PEFFERS, Ken; TUUNANEN, Tuure; ROTHENBERGER, Marcus A.; CHATTERJEE, Samir. A design science research methodology for information systems research. **Journal of Management Information Systems**, v. 24, n. 3, p. 45-77, 2007. DOI: https://doi.org/10.2753/MIS0742-1222240302.

RAVAL, Manan Bhavesh; JOSHI, Hirenkumar Indravadan; KHATRI, Bharat; VALAKI, Janak. Development and implementation of an Industrial IoT-based real-time process monitoring system to improve Overall Equipment Effectiveness (O.E.E.) of bearing manufacturing process. **ITEGAM-JETIA**, v. 12, n. 58, p. 11-22, 2026. DOI: https://doi.org/10.5935/jetia.v12i58.2410.

SCHWAB, Klaus. **A Quarta Revolução Industrial**. São Paulo: Edipro, 2016.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

SHARDA, Ramesh; DELEN, Dursun; TURBAN, Efraim. **Business Intelligence, Analytics, and Data Science**: A Managerial Perspective. 4. ed. Harlow: Pearson, 2018.

SILVA, Nubia Adriane da; ABREU, Jaqueline Lilge; KLINGENBERG, Cristina Orsolin; ANTUNES JUNIOR, José Antônio Valle; LACERDA, Daniel Pacheco. Industry 4.0 and micro and small enterprises: systematic literature review and analysis. **Production & Manufacturing Research**, v. 10, n. 1, p. 696-726, 2022. DOI: https://doi.org/10.1080/21693277.2022.2124466.

VANDERPLAS, Jake. **Python Data Science Handbook**. 2. ed. Sebastopol: O'Reilly Media, 2022.

VENABLE, John; PRIES-HEJE, Jan; BASKERVILLE, Richard. FEDS: a framework for evaluation in design science research. **European Journal of Information Systems**, v. 25, n. 1, p. 77-89, 2016. DOI: https://doi.org/10.1057/ejis.2014.36.

VORNE INDUSTRIES. **OEE calculation**: definitions, formulas, and examples. [S. l.]: OEE.com, c2026. Disponível em: https://www.oee.com/calculating-oee/. Acesso em: 12 nov. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.