



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**MaxVista - Ferramenta de
desenvolvimento de dashboards para
IBM MAXIMO**

Pedro Henrique Nunes de Assis

**TRABALHO DE
CONCLUSÃO DE CURSO**

**ORIENTAÇÃO:
Rafael Frederico Alexandre**

**Fevereiro, 2026
João Monlevade–MG**

Pedro Henrique Nunes de Assis

**MaxVista - Ferramenta de desenvolvimento de
dashboards para IBM MAXIMO**

Orientador: Rafael Frederico Alexandre

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Fevereiro de 2026

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

A848m Assis, Pedro Henrique Nunes de.
MaxVista [manuscrito]: ferramenta de desenvolvimento de
dashboards para IBM MAXIMO. / Pedro Henrique Nunes de Assis. - 2026.
81 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Rafael Frederico Alexandre.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Sistemas de
Informação .

1. Administração - Ativos (Contabilidade). 2. Aplicações Web -
Desenvolvimento. 3. Dashboards (Sistemas de informação gerencial). 4.
Fábricas - Manutenção. 5. Inteligência competitiva (Administração). 6.
Negócios - Processamento de dados. 7. Planejamento empresarial. I.
Alexandre, Rafael Frederico. II. Universidade Federal de Ouro Preto. III.
Título.

CDU 005.932:004.775

Bibliotecário(a) Responsável: Flavia Reis - CRB6/2431



FOLHA DE APROVAÇÃO

Pedro Henrique Nunes de Assis

**MaxVista - Ferramenta de
desenvolvimento de dashboards para
IBM MAXIMO**

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 24 de fevereiro de 2026

Membros da banca

Doutor - Rafael Frederico Alexandre - Orientador (Universidade Federal de Ouro Preto)
Doutor - Fernando Bernardes de Oliveira - (Universidade Federal de Ouro Preto)
Doutor - Luiz Carlos bambirra Torres - (Universidade Federal de Ouro Preto)
Especialista - Leonardo Lima - (it-eam)

Rafael Frederico Alexandre, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 01/04/2026



Documento assinado eletronicamente por **Rafael Frederico Alexandre, PROFESSOR DE MAGISTERIO SUPERIOR**, em 01/04/2026, às 11:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1083358** e o código CRC **27B8A258**.

“What gets measured gets managed.”

— Peter F. Drucker (1909 – 2005).

Resumo

Este trabalho apresenta o MaxVista, uma ferramenta web para desenvolvimento e disponibilização de *dashboards* para o IBM Maximo e para o IBM Maximo Application Suite Manage (MAS Manage), com o objetivo de reduzir o tempo entre dados operacionais e indicadores acionáveis em ambientes industriais. Embora o ecossistema Maximo ofereça relatórios embutidos e suporte a iniciativas de Inteligência de Negócios (BI), organizações frequentemente enfrentam alto esforço de implantação, dependências de infraestrutura e baixa agilidade para evoluir painéis operacionais de manutenção e gestão de ativos. Como alternativa, o MaxVista é proposto em abordagem *plug and play*: a solução requer essencialmente a parametrização de conexão ao banco de dados, automatiza a descoberta e a preparação dos dados, aplica padronização semântica e publica *dashboards* configuráveis com baixo atrito de adoção.

A arquitetura separa a camada de consulta e tratamento dos dados da camada de visualização. O *backend* executa consultas em Linguagem de Consulta Estruturada (SQL) controladas, normaliza e estrutura o retorno em um contrato estável de dados em Notação de Objetos JavaScript (JSON), consumido pelo *frontend* para renderização interativa. A aplicação incorpora autenticação, organização por grupos e controle de acesso, além de proteção de credenciais por *hash* para senhas de usuários e criptografia simétrica para segredos de conexão, mantendo persistência interna desacoplada do repositório transacional do cliente. Os resultados demonstram a viabilidade funcional por meio de painéis temáticos e indicadores implementados, bem como evidenciam características técnicas de governança, segurança e separação de responsabilidades, destacando cenários em que rapidez, padronização e baixa fricção são decisivos.

Palavras-chave: IBM Maximo; MAS Manage; *dashboards*; BI industrial; *plug and play*.

Abstract

This monograph presents MaxVista, a web-based dashboard development and delivery tool designed to work alongside IBM Maximo and IBM MAS Manage, aiming to shorten the path from operational data to actionable industrial KPIs. While the Maximo ecosystem supports embedded reporting and enterprise BI initiatives, organizations often face significant implementation effort, infrastructure dependencies, and limited agility to evolve maintenance-oriented operational dashboards. As an alternative, MaxVista follows a plug-and-play approach: it primarily requires database connection parameterization, automates data discovery and preparation, applies semantic standardization, and publishes configurable dashboards with reduced time-to-value.

The architecture enforces a clear separation between data querying and processing and the visualization layer. The backend executes controlled SQL queries, normalizes and structures results into a stable JSON data contract, which is then consumed by the frontend for interactive rendering. The solution includes authentication, group-based authorization and access control, and credential protection through password hashing and symmetric encryption for connection secrets, keeping an internal persistence layer isolated from the customer's transactional repository. The results demonstrate functional feasibility through implemented thematic panels and indicators, and provide technical evidence regarding governance, security, and responsibility separation, emphasizing industrial scenarios where speed, standardization, and low adoption friction are critical.

Keywords: IBM Maximo; MAS Manage; dashboards; industrial BI; plug and play.

Lista de ilustrações

Figura 1 – Arquitetura de referência do MAS em OpenShift.	24
Figura 2 – Arquitetura de referência básica do IBM Maximo.	28
Figura 3 – Arquitetura de referência avançada do IBM Maximo.	29
Figura 4 – Visão geral de funcionamento do MaxVista	39
Figura 5 – Pipeline MaxVista	40
Figura 6 – Diagrama entidade-relacionamento (ER) do sistema. A relação N:M entre usuários (DBCon) e grupos (Group) é modelada pela tabela associativa GroupUser.	42
Figura 7 – Script de automação para abertura do MaxVista	44
Figura 8 – Ação associada ao objeto para abertura do MaxVista	44
Figura 9 – Conexão do MaxVista com MAS	46
Figura 10 – Conexão do MaxVista com Maximo 7.x	46
Figura 11 – Painel de <i>login</i> do MaxVista	49
Figura 12 – Painel de registro de usuários do MaxVista	50
Figura 13 – Painel de recuperação de senha do MaxVista	50
Figura 14 – Página de perfil do usuário	51
Figura 15 – Painel de conexão com banco de dados	51
Figura 16 – Painel de gerenciamento de grupos	52
Figura 17 – Painel de gerenciamento de integrantes	52
Figura 18 – Painel de criação de grupos	53
Figura 19 – Painel de visualização anual de ordens de serviço	53
Figura 20 – Painel de nuvem de palavras em <i>log</i> de serviço	54
Figura 21 – Tipo de serviço por solicitação	54
Figura 22 – Painel de visualização de manutenções preventivas ativas por site	55
Figura 23 – Painel de ordens de serviço corretivas por site	55
Figura 24 – Painel de ordens de serviço por manutenção preventiva completas	55
Figura 25 – Painel de relação de grupos de itens	56
Figura 26 – Painel de ordens de serviço aguardando material por ano	57
Figura 27 – Painel de tempo de espera de material por ordem de serviço	57
Figura 28 – Tempo de exibição de cada <i>dashboard</i>	58
Figura 29 – Ação de acesso ao MaxVista	59
Figura 30 – Painel principal: número de ordens de serviço por site	71
Figura 31 – Painel principal: ordens concluídas a partir de manutenção preventiva	71
Figura 32 – Painel principal: ordens de serviço concluídas por mês	72
Figura 33 – Painel principal: sazonalidade de ordens de serviço por ano e mês	72
Figura 34 – Painel principal: ordens concluídas no prazo e vencidas por período	72

Figura 35 – Painel principal: origem das ordens de serviço	73
Figura 36 – Painel principal: relação por tipo e site	73
Figura 37 – Painel principal: distribuição de status das ordens de serviço	74
Figura 38 – Painel principal: tempo de espera em aprovação	74
Figura 39 – Painel principal: tempo médio de falha por ativo	74
Figura 40 – Painel principal: composição de ativos e locais por manutenção preventiva	75
Figura 41 – Painel principal: tipos de solicitações de serviço	75
Figura 42 – Painel principal: tipos de trabalho por site	76
Figura 43 – Painel principal: nuvem de termos associada a logs de serviço das ordens de serviço	76
Figura 44 – Painel de Manutenção: ordens corretivas por site	77
Figura 45 – Painel de Manutenção: concentração de corretivas por ativo	77
Figura 46 – Painel de Manutenção: concentração de corretivas por local	78
Figura 47 – Painel de Manutenção: manutenção preventiva ativa por site	78
Figura 48 – Painel de Manutenção: ordens associadas a MPs completas	78
Figura 49 – Painel de Almoxarifado: ordens de serviço aguardando material	79
Figura 50 – Painel de Almoxarifado: tempo de espera associado a ordens aguardando material	79
Figura 51 – Painel de Almoxarifado: completude cadastral por código de mercadoria	80
Figura 52 – Painel de Almoxarifado: completude cadastral por grupo de mercadoria	80

Lista de tabelas

Tabela 1 – Comparação sintética entre BIRT, Cognos Analytics e MaxVista	35
Tabela 2 – Tecnologias identificadas na implementação do MaxVista	45
Tabela 3 – Contadores globais do DB2 — Teste 1 (antes/depois) e variação. . . .	61
Tabela 4 – Contadores globais do DB2 — Teste de carga sustentada com alta concorrência (antes/depois) e variação.	61

Lista de abreviaturas e siglas

API	Interface de Programação de Aplicações (<i>Application Programming Interface</i>)
BI	Inteligência de Negócios (<i>Business Intelligence</i>)
BIRT	Ferramenta de Relatórios e Inteligência de Negócios (<i>Business Intelligence and Reporting Tools</i>)
BROS	Servidor Exclusivo de Relatórios BIRT (<i>BIRT Report Only Server</i>)
DB2	Sistema Gerenciador de Banco de Dados IBM Db2
DSR	Pesquisa em Ciência do Design (<i>Design Science Research</i>)
DSRM	Metodologia de Pesquisa em Ciência do Design (<i>Design Science Research Methodology</i>)
EAM	Gestão de Ativos Empresariais (<i>Enterprise Asset Management</i>)
EN	Norma Europeia (<i>European Norm</i>)
ERP	Planejamento de Recursos Empresariais (<i>Enterprise Resource Planning</i>)
ETL	Extraair, Carregar, Transformar
HxGN	Hexagon (marca associada à solução HxGN EAM)
IA	Inteligência Artificial
IBM	International Business Machines (IBM)
IDC	International Data Corporation (IDC)
IFS	Industrial and Financial Systems (IFS)
ISO	International Organization for Standardization (ISO)
JSON	Notação de Objetos JavaScript (<i>JavaScript Object Notation</i>)
KPI	Indicador-Chave de Desempenho (<i>Key Performance Indicator</i>)
MAS	Suíte de Aplicações Maximo (<i>Maximo Application Suite</i>)
MDN	Mozilla Developer Network (MDN)
MP	Manutenção Preventiva

OIDC	OpenID Connect
OS	Ordem de Serviço
RBAC	Role-Based Access Control
RCA	Análise de Causa Raiz (<i>Root Cause Analysis</i>)
SaaS	Software como Serviço (<i>Software as a Service</i>)
SAML	Security Assertion Markup Language
SAP	Systems, Applications and Products in Data Processing (SAP)
SCM	Gestão da Cadeia de Suprimentos (<i>Supply Chain Management</i>)
SLA	Acordo de Nível de Serviço (<i>Service Level Agreement</i>)
SQL	Linguagem de Consulta Estruturada (<i>Structured Query Language</i>)
SSO	Single Sign-On
TEI	Impacto Econômico Total (<i>Total Economic Impact</i>)
TfL	Transport for London (TfL)
VM	Máquina Virtual
WMATL	Status “Aguardando Material” no Maximo (<i>Waiting on Material</i>)

Sumário

1	INTRODUÇÃO	15
1.1	Justificativa	15
1.2	Objetivos	17
1.3	Metodologia	18
1.4	Organização do trabalho	19
1.5	Considerações finais do capítulo	19
2	REVISÃO BIBLIOGRÁFICA	20
2.1	Gestão de ativos, EAM e o papel do IBM Maximo	20
2.2	Infraestrutura do IBM Maximo Application Suite (MAS)	21
2.2.1	Plataforma de execução e camadas do OpenShift	21
2.2.2	Serviços de dados e persistência	22
2.2.3	Serviços transversais do MAS: identidade, certificados e licenciamento	22
2.2.4	Integração, mensageria e serviços analíticos	22
2.2.5	Observabilidade e operação do ambiente	23
2.2.6	Cenários de aplicação da infraestrutura do MAS	23
2.3	Infraestrutura do IBM Maximo Asset Management 7.6	25
2.3.1	Componentes de infraestrutura e visão geral	25
2.3.2	Servidor de aplicação e <i>middleware</i> : WebSphere e WebLogic	25
2.3.3	Comunicação com o banco de dados e persistência	26
2.3.4	Integração com sistemas externos e mensageria	26
2.3.5	Escalabilidade, disponibilidade e cenários de aplicação	26
2.4	Módulos do Maximo e sua contribuição para a gestão de ativos	30
2.4.1	Cadastro de ativos e localizações	30
2.4.2	Ordens de serviço, solicitações e ciclo de execução	30
2.4.3	Planejamento, manutenção preventiva e confiabilidade	30
2.4.4	Suprimentos, inventário, compras e contratos	31
2.4.5	Administração, segurança e governança	31
2.4.6	Configurações técnicas, banco de dados e customizações no ecossistema Maximo	31
2.4.7	Maximo Application Suite e ampliação de capacidades	32
2.5	Soluções EAM concorrentes e tendências de mercado	32
2.5.1	Principais soluções concorrentes	33
2.5.2	Tendências de mercado e implicações para visualização	33

2.6	Comparação entre abordagens de visualização: BIRT, Cognos Analytics e MaxVista	33
2.6.1	BIRT no Maximo: relatórios embutidos e foco operacional	33
2.6.2	Cognos Analytics: BI governado, escalabilidade e diversidade de artefatos	34
2.6.3	MaxVista: foco em dashboards industriais e redução de esforço de disponibilização	34
2.7	Considerações finais do capítulo	35
3	DESENVOLVIMENTO	36
3.1	Abordagens existentes no ecossistema Maximo	36
3.2	Visão geral do MaxVista	36
3.2.1	Objetivo e escopo da solução	36
3.2.2	Fluxo lógico de funcionamento	37
3.2.3	Pipeline de dados do MaxVista	39
3.3	Modelo de dados	41
3.4	Requisitos e considerações do desenvolvimento	42
3.5	Tecnologias utilizadas	42
3.6	Utilização de dados do Maximo	45
3.7	Considerações finais do capítulo	46
4	RESULTADOS	48
4.1	Ambiente e condições de validação	48
4.2	Resultados: funcionalidades, painéis e indicadores disponibilizados	48
4.2.1	Gerência de usuários	49
4.2.2	Painel geral	53
4.2.3	Painel de manutenção: manutenção corretiva e concentração por ativo/local	54
4.2.4	Painel de almoxarifado: materiais, inventário e impacto do status WMATL	56
4.2.5	Modo apresentação	57
4.2.6	Integração com o Maximo	58
4.3	Resultados técnicos: conectividade, segurança e separação de responsabilidades	59
4.4	Testes de <i>performance</i> e impacto no banco de dados do Maximo	60
4.4.1	Procedimento de teste e métricas coletadas	60
4.4.2	Teste de execução de referência	60
4.4.3	Teste de carga sustentada com alta concorrência	61
4.4.4	Síntese e implicações para uso em cenários com maior volume de dados	62
4.5	Considerações finais do capítulo	62
5	CONCLUSÃO	63
5.1	Trabalhos futuros	64

REFERÊNCIAS	66
APÊNDICES	70
APÊNDICE A – IMAGENS DO PAINEL PRINCIPAL	71
APÊNDICE B – IMAGENS DO PAINEL DE MANUTENÇÃO . . .	77
APÊNDICE C – IMAGENS DO PAINEL DE ALMOXARIFADO . .	79

1 Introdução

Em organizações industriais intensivas em ativos, o desempenho operacional está diretamente associado à confiabilidade, disponibilidade e segurança de equipamentos e sistemas críticos. Nesse contexto, as práticas de gestão de ativos e manutenção dependem de registros consistentes e rastreáveis, capazes de sustentar planejamento, execução, auditoria e conformidade. Soluções de *Enterprise Asset Management* (EAM), como o IBM Maximo e o MAS Manage, consolidam esses dados em uma base transacional que reúne informações de ativos, ordens de serviço, históricos, materiais, localizações e regras operacionais, viabilizando governança e controle de processos.

Apesar da disponibilidade de dados, a geração de valor analítico contínuo permanece um desafio recorrente. Em muitos cenários, as organizações dependem de relatórios estáticos, extrações manuais ou iniciativas de BI, que demandam esforço significativo de implantação, modelagem, integração e manutenção para produzir benefícios mensuráveis. No ecossistema Maximo, mecanismos como o Business Intelligence and Reporting Tools (BIRT) atendem principalmente à produção de relatórios embutidos e formatados, enquanto plataformas como o IBM Cognos Analytics oferecem recursos corporativos avançados de BI governado, mas frequentemente implicam custos, complexidade de implantação e dependências de infraestrutura. Assim, há espaço para soluções que reduzam a fricção de adoção e acelerem a disponibilização de *dashboards* operacionais, preservando requisitos mínimos de segurança e governança.

Diante desse cenário, este trabalho apresenta o MaxVista, uma ferramenta desenvolvida para geração e disponibilização de *dashboards* a partir de dados do IBM Maximo (nomenclatura utilizada até a versão 7.x) e do MAS Manage (nomenclatura utilizada a partir da versão 8.x). O MaxVista opera mediante parametrização de conexão com o banco de dados e execução de consultas estruturadas para alimentar painéis voltados a indicadores de manutenção, gestão de ativos e almoxarifado, além de permitir evolução para outras áreas que demandem acompanhamento por métricas operacionais. A proposta busca reduzir o esforço inicial de implementar o sistema de *dashboards* e reduzir o tempo entre a coleta de dados transacionais e a obtenção de informação gerencial acionável.

1.1 Justificativa

A competitividade de organizações intensivas em ativos está diretamente associada à capacidade de manter elevados níveis de disponibilidade, confiabilidade e segurança operacional, ao mesmo tempo em que se controlam custos e riscos. Nesse contexto, a gestão de ativos é tratada como um conjunto coordenado de atividades para obtenção de

valor a partir de ativos, apoiada por princípios e benefícios descritos na série International Organization for Standardization (ISO) 55000 ([INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2014](#)). Complementarmente, a ISO 55001 estabelece requisitos para a implementação e melhoria de um sistema de gestão de ativos, reforçando a necessidade de processos, responsabilidades e mecanismos de governança que sustentem a tomada de decisão ao longo do ciclo de vida do ativo ([International Organization for Standardization, 2014](#)).

Sistemas EAM consolidam cadastros, hierarquias, ordens de serviço, planos e históricos de manutenção em uma base única, reduzindo assimetrias de informação e ampliando rastreabilidade e auditabilidade. No IBM Maximo, por exemplo, um ativo é representado por registros que descrevem equipamentos e materiais serializados, incluindo hierarquia, localização e atributos necessários para suportar processos operacionais ([IBM, s.d.f](#)). No MAS Manage, a plataforma é descrita como capaz de fornecer visão abrangente de ativos e processos de trabalho associados, sustentando planejamento, controle e conformidade ([IBM, s.d.e](#)).

Entretanto, ainda que a consolidação transacional seja essencial, a geração de valor contínuo depende da transformação de dados operacionais em indicadores gerenciais e operacionais consistentes. Para esse propósito, é recorrente a necessidade de padronização e seleção de *Key Performance Indicators* (KPIs) e de instrumentos de visualização que reduzam o tempo entre evento, diagnóstico e ação. Normas como a *European Norm* (EN) 15341 apresentam diretrizes e um conjunto de KPIs aplicáveis à função manutenção ([British Standards Institution, 2019](#)). Em paralelo, estudos destacam que a escolha de indicadores e a maneira de apresentá-los influenciam a efetividade da gestão, sobretudo quando se busca alinhamento entre desempenho, custo e risco ([STENSTRÖM, 2013](#)). Em ambientes industriais, o uso de painéis (*dashboards*) tende a favorecer comunicação, monitoramento contínuo e priorização, desde que as métricas sejam semanticamente coerentes e operacionalmente acionáveis ([TOKOLA, 2016](#)).

A relevância do Maximo para a indústria também é evidenciada por materiais públicos que associam sua adoção a ganhos de desempenho operacional. Em documento de valor de negócio patrocinado pela IBM, a International Data Corporation (IDC) reporta benefícios potenciais como redução de *downtime* não planejado, aumento da vida útil média de ativos e melhoria de produtividade de equipes técnicas ([IDC Research, Inc., 2024](#)). Ademais, em estudo de caso sobre o *Transport for London* (TfL), é reportada a expectativa de economia ao longo de horizonte plurianual por meio de padronização e centralização de processos de manutenção e dados de ativos ([IBM, 2024b](#)).

Na prática, porém, a extração de valor analítico pode ser limitada por dependência de relatórios estáticos ou por soluções de BI que exigem esforço elevado de modelagem, desenvolvimento, testes e sustentação. No ecossistema Maximo, uma alternativa recorrente

é o BIRT, integrado ao produto para relatórios embutidos, cuja customização demanda ambiente de desenvolvimento e processo de importação e publicação no Maximo (IBM, s.d.b). Outra abordagem frequente é a adoção de plataformas corporativas de BI, como o IBM Cognos Analytics, que oferece capacidades de BI governado, com recursos de administração e controles para distribuição de informação em escala (IBM, s.d.d). Estudos de impacto econômico indicam que programas com BI corporativo podem envolver custos relevantes de implantação e recorrência, variando conforme contexto e modelo adotado (Forrester Consulting, 2022). Além disso, no contexto Maximo, há condições específicas de licenciamento (*entitlement*) e limitações de uso conforme versão e modalidade, o que pode impactar a estratégia de adoção (IBM, s.d.g; IBM, s.d.a).

Diante desse cenário, justifica-se o desenvolvimento do MaxVista como alternativa orientada a reduzir o esforço inicial de disponibilização de *dashboards* e acelerar o uso gerencial de dados operacionais do IBM Maximo/MAS Manage. A proposta parte de uma integração baseada em conexão direta ao banco de dados do Maximo, com camadas de consulta e visualização voltadas à entrega de indicadores industriais, buscando reduzir fricções típicas de iniciativas extensas de BI e viabilizar evolução incremental de painéis conforme necessidades organizacionais.

1.2 Objetivos

O objetivo geral deste trabalho é apresentar, fundamentar e avaliar o MaxVista como uma ferramenta de desenvolvimento e disponibilização de *dashboards* a partir de dados do IBM Maximo ou do MAS Manage, estabelecendo um paralelo crítico com abordagens usuais — BIRT e IBM Cognos Analytics — em termos de esforço de implantação, agilidade de entrega e aderência ao contexto industrial.

Como objetivos específicos, destacam-se:

- Caracterizar o contexto de EAM e a relevância do IBM Maximo para operações industriais, incluindo evidências públicas de valor;
- Descrever a arquitetura e o fluxo de dados do MaxVista, evidenciando decisões de projeto e componentes;
- Implementar e ilustrar painéis e indicadores representativos para manutenção, almoxarifado e visão gerencial;
- Avaliar aspectos técnicos essenciais, como conectividade, controle de acesso e proteção de credenciais, e discutir requisitos para evolução a uso em produção;
- Comparar o MaxVista com BIRT e IBM Cognos Analytics, discutindo pontos fortes, limitações e cenários de adoção.

1.3 Metodologia

A pesquisa possui natureza aplicada, com foco no projeto e desenvolvimento de um artefato de software e na avaliação de sua utilidade em um contexto realista de dados e processos de EAM. O trabalho adota princípios de *Design Science Research* (DSR), nos quais o conhecimento é produzido por meio da construção e avaliação de artefatos voltados à resolução de problemas organizacionais (HEVNER et al., 2004). Como referência operacional, utiliza-se também a *Design Science Research Methodology* (DSRM), que orienta etapas desde a identificação do problema e definição de objetivos até desenvolvimento, demonstração e avaliação (PEFFERS et al., 2007).

A condução do estudo seguiu as diretrizes centrais da DSR no que se refere à relevância do problema, à construção de um artefato com utilidade prática e à avaliação baseada em evidências. Nesse enquadramento, o MaxVista é tratado como artefato de software cujo propósito é reduzir fricção na disponibilização de *dashboards* operacionais no ecossistema Maximo, respondendo a uma lacuna recorrente entre a existência do dado transacional e sua conversão em informação acionável para decisão. A contribuição do trabalho é apresentada tanto na maneira do artefato implementado quanto na explicitação de suas decisões arquiteturais e implicações de uso em cenário industrial, permitindo reprodutibilidade e extensão por trabalhos futuros (HEVNER et al., 2004; PEFFERS et al., 2007).

Na prática, o método foi conduzido em ciclos: (i) levantamento de requisitos a partir de necessidades típicas de indicadores de manutenção e gestão de ativos, com definição de escopo de painéis e seleção de métricas alinhadas a referenciais consolidados de desempenho em manutenção, incluindo diretrizes normativas e de terminologia para indicadores (British Standards Institution, 2019; STENSTRÖM, 2013); (ii) desenho arquitetural e implementação do MaxVista, incluindo persistência interna, controle de acesso e *endpoints* de dados, com ênfase em separação de responsabilidades entre camada de consulta e tratamento e camada de visualização; (iii) demonstração por meio da execução do MaxVista conectado a uma base do Maximo, com painéis alimentados por consultas em SQL e retorno em JSON, evidenciando o funcionamento ponta a ponta do fluxo de dados e a geração de visualizações em *dashboards* (TOKOLA, 2016); (iv) avaliação técnica e operacional, contemplando análise de segurança e governança (proteção de credenciais, separação de responsabilidades e isolamento da persistência interna) e verificação de desempenho por testes de carga e observação de métricas do banco de dados, buscando avaliar a viabilidade de execução sob concorrência e identificar pontos de atenção para escalabilidade; e (v) consolidação comparativa com BIRT e IBM Cognos Analytics, com base em documentação e evidências públicas, discutindo esforço de implantação, governança e aplicabilidade, além de elementos de impacto econômico associados a iniciativas de BI corporativo (Forrester Consulting, 2022).

Complementarmente, a revisão bibliográfica e documental utilizada para fundamentação do contexto de gestão de ativos e requisitos de governança foi orientada por normas internacionais de gestão de ativos, que definem conceitos, princípios e requisitos para sistemas de gestão, e sustentam a interpretação do papel do EAM no ciclo de vida do ativo ([INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2014](#); [International Organization for Standardization, 2014](#)). Essas referências foram utilizadas como base conceitual para justificar a necessidade de rastreabilidade, padronização e consistência semântica na construção de indicadores e painéis operacionais em ambientes intensivos em ativos.

1.4 Organização do trabalho

Para além deste capítulo introdutório, a estrutura deste documento está organizada conforme descrito a seguir. O [Capítulo 2](#) apresenta a revisão bibliográfica e documental, examinando fundamentos de gestão de ativos, indicadores de manutenção e abordagens de visualização de dados, além de situar soluções usuais do ecossistema Maximo. O [Capítulo 3](#) detalha a concepção e o desenvolvimento do MaxVista, incluindo arquitetura, fluxo de dados, tecnologias utilizadas e decisões de implementação de *backend* e *frontend*, bem como aspectos de segurança e organização modular. O [Capítulo 4](#) apresenta os resultados e análises obtidos com a execução do sistema, discutindo as evidências funcionais (painéis e indicadores), os resultados técnicos (conectividade e segurança) e observações operacionais relevantes para escalabilidade e uso em cenários com maior volume de dados. Por fim, o [Capítulo 5](#) sintetiza as conclusões do trabalho, reafirma as contribuições do MaxVista, discute limitações identificadas e propõe direções para evoluções futuras e trabalhos posteriores.

1.5 Considerações finais do capítulo

Este capítulo apresentou a motivação e o contexto do problema, destacando a lacuna entre a disponibilidade de dados transacionais no ecossistema Maximo e a conversão desses dados em informação gerencial acionável com baixa fricção. Foram delimitados os objetivos do trabalho e a abordagem metodológica adotada, estabelecendo o enquadramento do MaxVista como artefato orientado à entrega ágil de *dashboards* operacionais com requisitos mínimos de governança e segurança. A partir desse enquadramento, o próximo capítulo aprofunda os fundamentos conceituais e documentais de gestão de ativos, indicadores de manutenção e soluções usuais de visualização, de modo a sustentar o posicionamento do MaxVista e orientar as decisões de projeto discutidas adiante.

2 Revisão bibliográfica

Este capítulo apresenta a revisão bibliográfica e documental que fundamenta o problema abordado e contextualiza a proposta do MaxVista no ecossistema de gestão de ativos. Inicialmente, são discutidos os conceitos de gestão de ativos e EAM, destacando o papel dos sistemas transacionais na padronização de processos, na rastreabilidade e na conformidade em ambientes industriais. Em seguida, o IBM Maximo é caracterizado como solução de EAM, com ênfase em sua organização por módulos e na relevância de seus registros para a construção de indicadores e análises de desempenho. Na sequência, são examinadas tendências e soluções concorrentes no mercado de EAM, ressaltando como aspectos como adoção de nuvem, integração corporativa e maturidade de governança influenciam a implementação e o uso dessas plataformas. Por fim, a revisão aborda caminhos para transformar dados do Maximo em informação gerencial, comparando abordagens baseadas em relatórios embutidos (BIRT), plataformas de BI corporativo (IBM Cognos Analytics) e o posicionamento do MaxVista como ferramenta voltada à disponibilização ágil de *dashboards* operacionais.

2.1 Gestão de ativos, EAM e o papel do IBM Maximo

As organizações industriais têm sua competitividade diretamente relacionada à confiabilidade, disponibilidade e segurança de seus ativos. Conforme a norma ISO 55000, um ativo pode ser entendido como um item, coisa ou entidade que possui valor potencial ou real para uma organização ([INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2014](#)). Em ambientes industriais intensivos em ativos, a gestão sistematizada desses elementos envolve não apenas o registro patrimonial, mas também a disciplina de manutenção, o planejamento de intervenções, o controle de recursos, a conformidade e a rastreabilidade das decisões.

Nesse contexto, soluções de EAM assumem papel estratégico ao consolidar processos e registros, parametrizar regras operacionais e sustentar auditoria e governança sobre a operação. No IBM Maximo, por exemplo, o conceito de ativo é materializado por meio de registros que descrevem equipamentos ou materiais serializados, com informações de localização, fornecedor, custos de manutenção e hierarquias (ativo pai e ativos filhos) ([IBM, s.d.f](#)). Esse conjunto de dados, quando preenchido e mantido com consistência, forma a base para análises operacionais e gerenciais.

O IBM Maximo Manage, componente do MAS Manage, é descrito como uma aplicação voltada a oferecer visão abrangente dos ativos, suas condições e localizações, bem como dos processos de trabalho associados, apoiando planejamento, controle, audi-

toria e conformidade (IBM, s.d.e). Adicionalmente, a base de dados do sistema organiza atributos e relacionamentos físicos e lógicos relevantes para o ciclo de vida do ativo, possibilitando correlacionar eventos de manutenção, falhas, consumos de materiais e desempenho operacional.

A relevância do Maximo em ambientes industriais também é sustentada por comunicações e estudos de caso públicos. Em materiais associados ao produto, a IBM referencia resultados de pesquisa da IDC apontando potenciais ganhos com redução de *downtime* não planejado, aumento da vida útil média dos ativos e melhoria de produtividade técnica, entre outros (IDC Research, Inc., 2024). Em estudo de caso sobre o *Transport for London* (TfL), a IBM descreve projeções de economia ao longo de horizonte plurianual associadas à padronização e centralização do processo de manutenção com suporte do Maximo (IBM, 2024b). Esses elementos reforçam que, quando adequadamente explorados, os dados consolidados no EAM deixam de ser apenas registros transacionais e passam a viabilizar decisões que afetam custo, risco e desempenho (IBM, 2024b).

2.2 Infraestrutura do IBM Maximo Application Suite (MAS)

O MAS adota uma arquitetura *cloud-native*, baseada em *containers* e orquestração em plataforma Kubernetes, sendo operacionalizado tipicamente sobre o Red Hat OpenShift Container Platform (OCP). Nesse modelo, o MAS é implantado como um conjunto de componentes e serviços (operadores, *workloads* e dependências) que suportam as aplicações da suíte, além do Manage, como Monitor, Predict e Visual Inspection, bem como serviços transversais de identidade, licenciamento, persistência e observabilidade.

2.2.1 Plataforma de execução e camadas do OpenShift

A base de execução do MAS é um *cluster* OpenShift composto por um *control plane* — nós mestres responsáveis por serviços de controle, como *scheduler*, *controllers* e repositório de estado do *cluster* — e por nós de trabalho (*worker nodes*), responsáveis pela execução dos *pods* das aplicações e serviços. Alternativamente, pode-se adotar um *single node*, que consiste na instalação do *cluster* com apenas um nó. A organização em camadas (controle e dados) permite separar responsabilidades de coordenação do *cluster* e de processamento de carga de aplicação, o que é relevante para requisitos de disponibilidade e escalabilidade.

Do ponto de vista de infraestrutura, a implantação do MAS depende de elementos de conectividade e exposição de serviços, como balanceadores de carga, resolução de nomes e políticas de rede, além de mecanismos de autenticação integrados ao OpenShift. Em cenários corporativos, essa base é complementada por requisitos de administração e acesso

controlado ao ambiente — por exemplo, por meio de *bastion host* — conforme a política de segurança e segregação de ambientes adotada.

2.2.2 Serviços de dados e persistência

A arquitetura do MAS envolve múltiplos tipos de persistência, de acordo com as necessidades de cada serviço e aplicação. Observa-se o uso de bancos relacionais e não relacionais, além de armazenamento persistente para *volumes* de aplicação. Em implantações que utilizam serviços de dados internos ao *cluster*, soluções de armazenamento distribuído no ecossistema OpenShift (como *storage* para *persistent volumes*) viabilizam a persistência de componentes de aplicação, registros e serviços compartilhados.

Além disso, conforme o perfil de uso e as escolhas de implantação, o MAS pode integrar bancos de dados analíticos, como *data warehouse*, para suportar cargas de análise e uso de dados por aplicações que exigem agregações e históricos em maior escala. Esse aspecto é particularmente relevante quando se considera o consumo de dados para indicadores, painéis e análises de desempenho.

2.2.3 Serviços transversais do MAS: identidade, certificados e licenciamento

A operação do MAS depende de serviços comuns (*cross-cutting services*) que sustentam segurança e governança na plataforma. Entre esses serviços, destacam-se:

- **Identidade e autenticação:** integração com provedor de identidade para autenticação no OpenShift e, quando aplicável, federação com SSO corporativo (por exemplo, via SAML/OIDC), de modo a alinhar o acesso às políticas organizacionais.
- **Gestão de certificados:** mecanismos para emissão, distribuição e rotação de certificados utilizados por rotas, serviços internos e integrações, contribuindo para a proteção de tráfego em trânsito e para a padronização de configuração por ambiente.
- **Licenciamento:** serviços responsáveis pelo controle de consumo e validação de licenças da suíte, integrados ao ciclo de vida das aplicações e à administração do ambiente.

Esses serviços impactam diretamente iniciativas de disponibilização de *dashboards* e integrações, pois condicionam como autenticação, autorização, criptografia e governança são implementadas de ponta a ponta no ecossistema.

2.2.4 Integração, mensageria e serviços analíticos

Em cenários corporativos, a arquitetura do MAS pode incorporar serviços de integração e mensageria para viabilizar comunicação entre aplicações, ingestão de dados

e integração com sistemas externos. Essa camada tende a ser composta por *brokers* e conectores, permitindo integração com *ERPs*, sistemas legados, fontes operacionais e serviços de dados.

Adicionalmente, o MAS pode ser complementado por serviços analíticos e de ciência de dados — como plataformas de dados e capacidades de *machine learning* — para aplicações que demandam classificação, predição ou enriquecimento de informação. A presença desses componentes reforça que a infraestrutura do MAS não se limita ao EAM transacional, mas pode abranger uma cadeia mais ampla de consumo e processamento de dados.

2.2.5 Observabilidade e operação do ambiente

A implantação do MAS em OpenShift normalmente envolve mecanismos de observabilidade para apoiar operação e sustentação, incluindo coleta de *logs*, métricas e eventos do *cluster* e das aplicações. Esses recursos são relevantes para administração do ambiente, diagnóstico de falhas, acompanhamento de desempenho e atendimento a requisitos de auditoria operacional.

2.2.6 Cenários de aplicação da infraestrutura do MAS

A infraestrutura do MAS pode ser aplicada em diferentes cenários organizacionais, conforme requisitos de governança, conectividade e estratégia de nuvem:

- **Implantação *on-premises* ou em nuvem privada:** adequada quando existem restrições regulatórias, requisitos de soberania de dados ou necessidade de integração estreita com redes industriais e sistemas internos.
- **Implantação em nuvem pública (ou modelo *Software as a Service*):** aplicável quando se busca padronização operacional, elasticidade e redução de esforço de administração de infraestrutura, mantendo mecanismos corporativos de governança e segurança.
- **Cenários híbridos:** utilizados quando parte dos dados e integrações permanece em ambiente local, enquanto serviços e aplicações operam com recursos escaláveis em nuvem, exigindo atenção a conectividade, segurança e latência.

Em todos os casos, a arquitetura em *containers* e a padronização por OpenShift oferecem um referencial para implantação e operação de aplicações associadas à gestão de ativos, incluindo componentes transacionais e analíticos. Para iniciativas de visualização e disponibilização de *dashboards* — como as discutidas neste trabalho —, a infraestrutura

do MAS constitui o contexto no qual requisitos de conectividade, governança, identidade e desempenho devem ser endereçados de maneira consistente.

O MAS adota uma arquitetura *cloud-native*, baseada em *containers* e orquestração em plataforma Kubernetes, sendo operacionalizado tipicamente sobre o OCP, conforme Figura 1. Nesse modelo, o MAS é implantado como um conjunto de componentes e serviços (operadores, *workloads* e dependências) que suportam as aplicações da suíte, além do Manage, como Monitor, Predict e Visual Inspection, bem como serviços transversais de identidade, licenciamento, persistência e observabilidade.

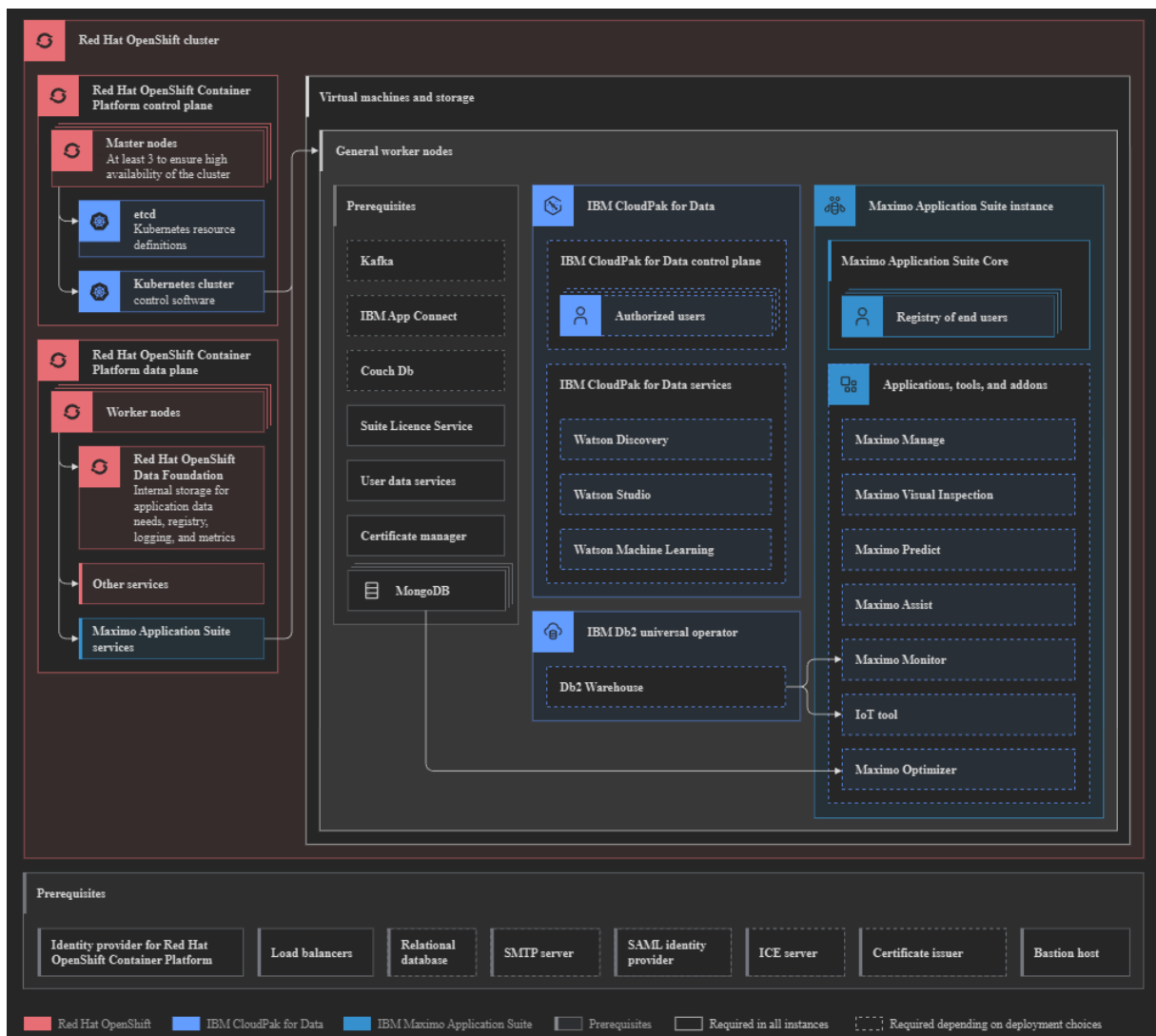


Figura 1 – Arquitetura de referência do MAS em OpenShift.

Fonte: IBM (s.d.). Disponível em: <<https://www.ibm.com/docs/en/masv-and-l/cd?topic=models-maximo-application-suite-architecture>>. Acesso em: 28 jan. 2026.

2.3 Infraestrutura do IBM Maximo Asset Management 7.6

O IBM Maximo Asset Management 7.6 adota uma arquitetura corporativa baseada em três camadas, na qual a interface é acessada via navegador e a lógica de negócio é executada em um servidor de aplicação, que por sua vez mantém conectividade com um servidor de banco de dados relacional. Em uma configuração básica, o Maximo opera em um único servidor de aplicação conectado a uma instância única de banco de dados, podendo também estabelecer conexão com um servidor de relatórios quando necessário (IBM, 2020). Em arquiteturas mais avançadas, podem existir diversos servidores de aplicação, recebendo funções diferentes ou compartilhando as mesmas funções para redundância e confiabilidade. Essa organização facilita a separação entre apresentação, processamento e persistência e permite evolução incremental para topologias de maior escala.

2.3.1 Componentes de infraestrutura e visão geral

Do ponto de vista de infraestrutura, o desempenho do Maximo é condicionado não apenas pela aplicação e pelo banco de dados, mas também por aspectos de rede, capacidade de hardware e serviços de software — servidor de aplicação, servidor de relatórios e servidor de banco de dados — (IBM, 2020). Em função dessa dependência entre camadas, recomenda-se que decisões de dimensionamento e arquitetura considerem simultaneamente latência de rede, concorrência de usuários, volume transacional e rotinas em *background* — por exemplo, *cron tasks* e processamento de filas.

2.3.2 Servidor de aplicação e *middleware*: WebSphere e WebLogic

No Maximo 7.6, a camada de *middleware* é representada pelo servidor de aplicação Java EE, responsável por hospedar a aplicação, gerir recursos de execução — como *threads*, memória e *pools* — e controlar conectividade e segurança em conjunto com os componentes corporativos. Em cenários de implantação, é comum a presença de um servidor HTTP na borda — ou balanceador — para encaminhar requisições ao servidor de aplicação, sobretudo quando se busca distribuir carga e prover alta disponibilidade.

A literatura técnica da IBM para Maximo 7.6.x descreve que a arquitetura de implantação pode evoluir de uma configuração básica — um servidor de aplicação — para topologias com *clustering* horizontal e/ou vertical, com múltiplas JVMs e separação de cargas, como transações interativas em um *cluster* e processos de filas e *cron tasks* em outro *cluster*, visando melhorar escalabilidade e experiência do usuário (IBM, 2020). O mesmo material ressalta que customizações extensas podem aumentar sobrecarga de processamento e reduzir a capacidade de usuários por JVM, reforçando a relação entre decisões de extensão e dimensionamento (IBM, 2020).

Em relação ao Oracle WebLogic, recomendações específicas indicam que, nesse cenário, não se utiliza o IBM HTTP Server, sendo comum a adoção do Oracle HTTP Server ou do Apache HTTP Server como componente frontal, conforme as práticas do ecossistema WebLogic (IBM, 2020). Dessa forma, a comunicação entre cliente e aplicação tende a seguir o encadeamento navegador → servidor HTTP/balanceamento → servidor de aplicação (WebSphere/WebLogic) → serviços internos (filas/relatórios) e banco de dados.

2.3.3 Comunicação com o banco de dados e persistência

A persistência do Maximo 7.6 é realizada em banco de dados relacional, e o servidor de aplicação realiza a comunicação com a base por meio de conectividade JDBC, usando recursos de *pooling* e gerenciamento de conexões típicos da plataforma de aplicação. Em termos de suporte e implantação, são recorrentes cenários com IBM Db2, Oracle Database e Microsoft SQL Server, com recomendações de configuração e ajuste de desempenho voltadas ao perfil de carga e ao volume de dados históricos (IBM, 2020).

Do ponto de vista de desempenho, materiais técnicos enfatizam que a capacidade do banco de dados e o ajuste de consultas influenciam diretamente a experiência de uso, especialmente em bases com grandes históricos e em ambientes com alta concorrência (IBM, 2020). Nessa linha, são apontadas práticas como revisão de índices, análise de *plans* de execução e parametrização adequada do ambiente, de modo a mitigar gargalos na camada de persistência.

2.3.4 Integração com sistemas externos e mensageria

Além do uso transacional, implantações corporativas frequentemente demandam integração com sistemas externos — como *ERPs*, sistemas legados e plataformas de dados. Para esses cenários, a documentação técnica descreve que, quando o *Integration Framework* é configurado, torna-se necessária a adoção de filas de mensageria adicionais, usadas para enviar dados do Maximo a sistemas externos e para receber dados externos destinados ao Maximo (IBM, 2020). Esse desenho suporta integrações síncronas e assíncronas e tende a ser combinado com mecanismos de controle operacional — monitoramento de filas, reprocessamento e auditoria de mensagens — conforme criticidade do processo.

2.3.5 Escalabilidade, disponibilidade e cenários de aplicação

A aplicação do Maximo 7.6 varia conforme porte organizacional e criticidade operacional. Para ambientes de desenvolvimento e cenários produtivos de menor escala, a própria referência técnica considera adequada a configuração básica com um servidor de aplicação conectado a uma instância única de banco de dados (IBM, 2020). Em contrapartida, para

ambientes corporativos com grande número de usuários e processos automatizados intensivos, recomenda-se arquitetura com *clustering* e separação de cargas entre componentes interativos e componentes de processamento em segundo plano, incluindo filas e rotinas agendadas (IBM, 2020).

Em termos de aplicação organizacional, esse modelo de infraestrutura é compatível com implantações *on-premises* e também em *datacenters* corporativos/privados, especialmente quando há requisitos de integração com redes internas, controle de acesso centralizado e proximidade com o banco de dados e sistemas legados. Em todos os cenários, a definição da topologia — servidores, *clusters*, mensageria e relatórios — deve ser coerente com: (i) perfil de uso (concorrência e transações), (ii) volume e crescimento do histórico, (iii) estratégia de integração e (iv) requisitos de disponibilidade e governança (IBM, 2020).

O IBM Maximo Asset Management 7.6 adota uma arquitetura corporativa baseada em três camadas, na qual a interface é acessada via navegador e a lógica de negócio é executada em um servidor de aplicação, que por sua vez mantém conectividade com um servidor de banco de dados relacional. Em uma configuração básica, o Maximo opera em um único servidor de aplicação conectado a uma instância única de banco de dados, podendo também estabelecer conexão com um servidor de relatórios quando necessário (IBM, 2020), como pode ser visto na Figura 2. Em arquiteturas mais avançadas, podem existir diversos servidores de aplicação, recebendo funções diferentes ou compartilhando as mesmas funções para redundância e confiabilidade, como demonstrado pela Figura 3. Essa organização facilita a separação entre apresentação, processamento e persistência e permite evolução incremental para topologias de maior escala.

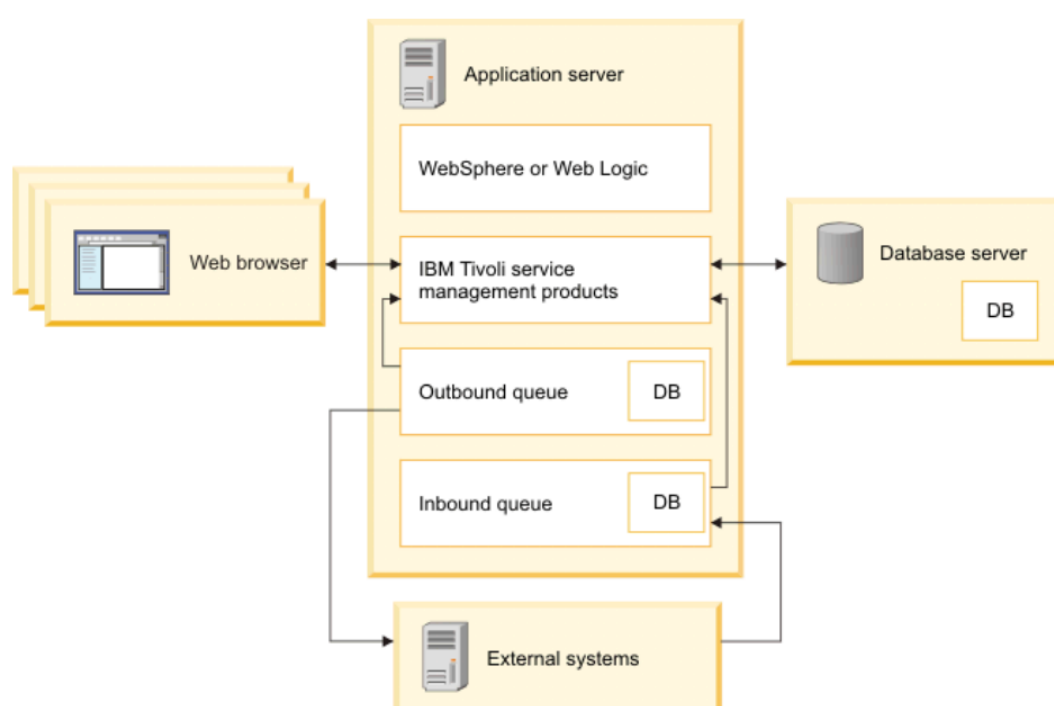


Figura 2 – Arquitetura de referência básica do IBM Maximo.

Fonte: IBM (s.d.). Disponível em: <<https://community.ibm.com/community/user/viewdocument/best-practices-for-system-performan?CommunityKey=3d7261ae-48f7-481d-b675-a40eb407e0fd&hlmlt=VT&tab=librarydocuments>>. Acesso em: 28 jan. 2026.

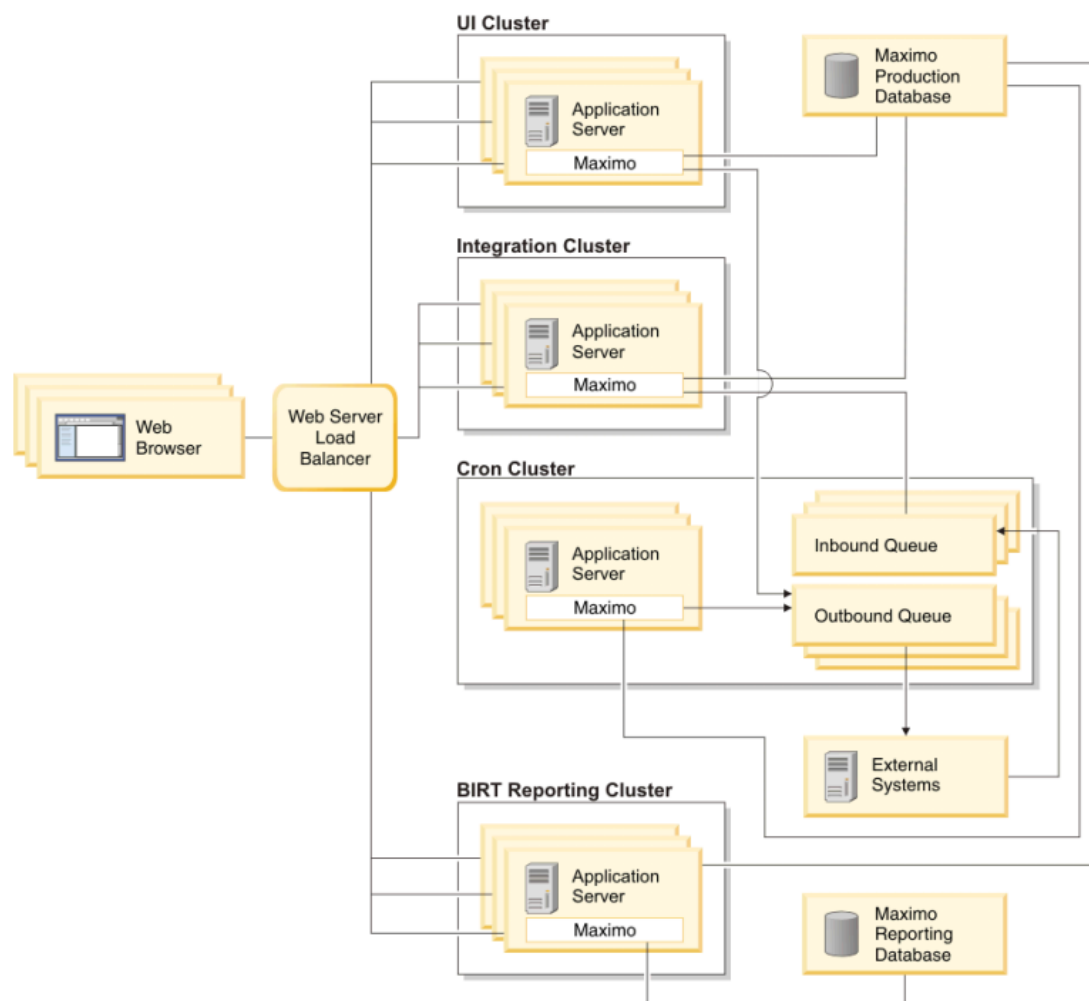


Figura 3 – Arquitetura de referência avançada do IBM Maximo.

Fonte: IBM (s.d.). Disponível em: <<https://community.ibm.com/community/user/viewdocument/best-practices-for-system-performan?CommunityKey=3d7261ae-48f7-481d-b675-a40eb407e0fd&hlmt=VT&tab=librarydocuments>>. Acesso em: 28 jan. 2026.

2.4 Módulos do Maximo e sua contribuição para a gestão de ativos

O Maximo organiza suas funcionalidades em módulos e aplicações, cobrindo o ciclo completo de gestão do ativo: cadastro, planejamento, execução da manutenção, suprimentos, custos e governança. A documentação do Maximo Manage apresenta a estrutura por módulos e suas respectivas aplicações, evidenciando que a solução foi desenhada para apoiar processos de ponta a ponta, do planejamento à execução e ao controle (IBM, 2024a). No contexto do MAS, além do Manage, a IBM disponibiliza capacidades adicionais — como monitoramento e análises de condição — como parte da proposta de valor do ecossistema (IBM, 2023b).

Do ponto de vista da gestão de ativos, a contribuição desses módulos se materializa em dois eixos: (i) padronização e rastreabilidade do processo (o que foi feito, por quem, quando e com qual resultado) e (ii) estruturação de dados operacionais confiáveis, condição necessária para indicadores e *dashboards* consistentes.

2.4.1 Cadastro de ativos e localizações

O cadastro de ativos e localizações constitui o núcleo informacional do EAM, pois define *o que* está sendo gerenciado, no qual está instalado e como se relaciona hierarquicamente com outros elementos, como sistemas, subsistemas e componentes. Essas informações sustentam análises como criticidade, concentração de falhas por classe de ativo, custo acumulado por equipamento e comparações por planta/unidade. Em termos práticos, a qualidade desse cadastro influencia diretamente a capacidade de produzir indicadores confiáveis, pois diversas métricas dependem de hierarquias e classificações consistentes (IBM, s.d.f).

2.4.2 Ordens de serviço, solicitações e ciclo de execução

O módulo de ordens de serviço estrutura a execução da manutenção ao registrar o ciclo de vida das intervenções: abertura, planejamento, programação, execução, apontamentos e encerramento. Esse encadeamento oferece rastreabilidade operacional e fornece insumo para indicadores como *backlog*, *lead time*, cumprimento de prazos, distribuição por tipo de manutenção — corretiva, preventiva, preditiva — e reincidência por ativo. Quando associado a regras e status padronizados, o conjunto de registros permite monitorar gargalos e sustentar governança do processo, sobretudo em ambientes com auditoria e conformidade.

2.4.3 Planejamento, manutenção preventiva e confiabilidade

A manutenção preventiva e o planejamento associado — planos, rotinas e gatilhos por tempo ou medidores — representam um mecanismo essencial de transição da manuten-

ção reativa para um modelo mais controlado. Em termos de gestão, esse módulo contribui para previsibilidade de carga de trabalho, consumo de recursos e melhoria do desempenho do ativo ao longo do tempo. Do ponto de vista analítico, permite medir aderência ao plano preventivo, efetividade das preventivas — por exemplo, redução de corretivas — desempenho por família de ativo e tendências sazonais.

2.4.4 Suprimentos, inventário, compras e contratos

Os módulos de inventário e compras conectam execução de manutenção a materiais, fornecedores e custos, constituindo dimensão crítica para entender atrasos, itens gargalo e fragilidades cadastrais. A relação entre ordens de serviço e materiais permite construir indicadores de impacto logístico sobre o atendimento de manutenção. Além disso, itens e classificações viabilizam análises de padronização e qualidade cadastral, relevantes para governança e redução de variação indevida entre unidades.

2.4.5 Administração, segurança e governança

Em ambientes corporativos, a gestão do EAM exige controle de acesso, segregação de funções e rastreabilidade de alterações. Nesse sentido, as capacidades administrativas do Maximo — gestão de usuários, perfis e permissões — e a disciplina de governança dos dados operacionais constituem pilar para auditoria e conformidade. Isso se reflete diretamente em iniciativas de visualização: *dashboards* confiáveis demandam não apenas consultas corretas, mas também garantia de que o dado foi produzido sob regras estáveis e com acesso controlado.

2.4.6 Configurações técnicas, banco de dados e customizações no ecossistema Maximo

Além dos módulos funcionais, o ecossistema IBM Maximo inclui camadas técnicas e mecanismos de customização determinantes para adequação do EAM às regras de negócio de cada organização. Na prática, a implantação envolve configurações de ambiente, ajustes de modelo de dados, integrações e personalizações controladas, de modo a refletir processos, nomenclaturas, fluxos de aprovação e requisitos de auditoria específicos.

Do ponto de vista de persistência e infraestrutura, o MAS Manage depende de componentes e configurações de banco de dados e conectividade que influenciam diretamente a operação transacional e, por extensão, o desempenho de consultas e extrações analíticas (IBM, 2025a). Em iniciativas de visualização, esse aspecto é relevante porque estratégias de consulta, índices e políticas de acesso podem impactar latência, concorrência e estabilidade do ambiente.

No âmbito de customização, o Maximo disponibiliza mecanismos formais para alterar comportamento e automatizar regras sem modificar diretamente o núcleo do produto. Destacam-se os *automation scripts*, utilizados para implementar validações, cálculos, preenchimentos automáticos e regras associadas a eventos do negócio. A documentação de boas práticas descreve recomendações de organização, legibilidade e segurança para desenvolvimento de *scripts*, enquanto guias operacionais detalham o processo de teste e os tipos de *scripts* suportados em diferentes contextos de execução (IBM, 2025d; IBM, 2025e).

Adicionalmente, a plataforma dispõe de recursos de configuração e extensão que abrangem domínios, *lookups*, regras de segurança, *workflows*, rotinas periódicas e integrações com sistemas externos. No MAS, a documentação descreve recursos de integração e extensibilidade e caracteriza o *Maximo integration framework* como base para definir conjuntos de dados, importação e exportação, além de outros mecanismos de integração (IBM, 2025c; IBM, 2025b). Em conjunto, essas capacidades sustentam a adaptação do Maximo a diferentes indústrias, ao mesmo tempo em que influenciam diretamente a qualidade e a consistência dos dados disponíveis para indicadores e *dashboards*. Portanto, ao discutir visualização e geração de valor analítico, é necessário considerar que a maturidade de cadastro e a disciplina de configuração e customização impactam o dado consumido por relatórios embutidos (BIRT), por BI corporativo (IBM Cognos Analytics) e por ferramentas dedicadas, como a proposta do MaxVista.

2.4.7 Maximo Application Suite e ampliação de capacidades

No MAS, o Manage pode ser complementado por aplicações que ampliam o uso de dados operacionais e condições do ativo, como ferramentas de monitoramento, saúde do ativo e predição. Essa composição é apresentada pela IBM como parte do ecossistema do MAS, indicando uma direção na qual o EAM se integra a camadas analíticas e de monitoramento para suportar decisões mais rápidas e orientadas por dados (IBM, 2023b). Para trabalhos focados em *dashboards* operacionais, como o MaxVista, isso reforça a importância de padronizar o acesso ao dado do Manage e estruturar indicadores diretamente acionáveis para manutenção e gestão de ativos.

2.5 Soluções EAM concorrentes e tendências de mercado

O mercado de EAM é impulsionado por necessidades convergentes: reduzir paradas não planejadas, aumentar confiabilidade, controlar custos e elevar a maturidade de governança sobre ativos físicos. Relatórios de mercado indicam crescimento contínuo do setor, com destaque para iniciativas de digitalização, conectividade e migração para modelos em nuvem (Grand View Research, 2024; Technavio, 2024). Em paralelo, a adoção de

arquiteturas híbridas e *Software as a Service* aparece como tendência recorrente, sobretudo pela busca de escalabilidade, redução de custos de infraestrutura e maior agilidade de implantação (PR Newswire, 2024).

2.5.1 Principais soluções concorrentes

Entre plataformas amplamente adotadas, destacam-se soluções integradas a grandes ecossistemas de ERP e cadeia de suprimentos e plataformas especializadas em EAM: SAP (SAP, 2024), Oracle (Oracle, 2024), IFS (IFS, 2024) e Infor/Hexagon (HxGN EAM) (Infor, 2024). Em termos práticos, a escolha entre soluções tende a considerar aderência ao ecossistema existente, capacidade de governança, facilidade de integração, custos, maturidade funcional e suporte a arquiteturas de nuvem.

2.5.2 Tendências de mercado e implicações para visualização

As tendências observadas no mercado têm efeito direto sobre como *dashboards* e camadas analíticas são implementados: (a) crescimento de ofertas *Software as a Service* e implantação híbrida, pressionando por conectividade padronizada, segurança e governança (PR Newswire, 2024); (b) maior integração entre EAM e camadas analíticas, incluindo IA aplicada à identificação de padrões e suporte à decisão (Grand View Research, 2024); e (c) valorização de dados operacionais confiáveis, com foco em indicadores consistentes e comparáveis entre unidades, reforçando a importância de semântica e padronização.

2.6 Comparação entre abordagens de visualização: BIRT, Cognos Analytics e MaxVista

A conversão do dado transacional do EAM em informação de gestão depende de mecanismos de consulta, organização e visualização. No ecossistema Maximo, duas abordagens são especialmente recorrentes: relatórios embutidos via BIRT e plataformas corporativas de BI como IBM Cognos Analytics. O MaxVista se posiciona como proposta alternativa, com foco em acelerar a entrega de *dashboards* operacionais conectados ao repositório do Maximo.

2.6.1 BIRT no Maximo: relatórios embutidos e foco operacional

O BIRT é uma tecnologia de relatórios baseada no ecossistema Eclipse, utilizada como mecanismo de *reporting* embutido em aplicações Java (Eclipse Foundation, 2013). No Maximo, seu uso é orientado a relatórios operacionais e formatados, associados a documentação, auditoria e emissão de saídas em formatos como PDF e planilhas. A documentação descreve o processo de obtenção e instalação do *BIRT Report Designer*

e registra a versão utilizada no contexto do produto (IBM, 2023a). Em Maximo 7.6, a IBM também descreve melhorias e a versão utilizada na linha, com ênfase em desempenho e administração (IBM, 2018). Em cenários de carga, há recomendações como o uso do *BIRT Report Only Server* (BROS) para balancear processamento de relatórios (IBM, 2022). Apesar dessas capacidades, o BIRT é menos alinhado ao paradigma de *dashboards* interativos de monitoramento contínuo, pois seu desenho privilegia relatórios com *layout* e saída definidos.

2.6.2 Cognos Analytics: BI governado, escalabilidade e diversidade de artefatos

O IBM Cognos Analytics se posiciona como plataforma de BI governado, com recursos de *dashboards* interativos, relatórios corporativos, governança centralizada, trilhas de auditoria e controle de acesso granular, aplicável em nuvem, *on-premises* ou híbrido (IBM, s.d.d). A documentação de administração e segurança descreve elementos centrais do produto, como gerenciamento de fontes de dados, políticas de segurança, usuários, grupos e papéis, além de recursos associados a atividades e agendamentos (IBM, 2019). Em projetos corporativos, isso permite escalar a distribuição de análises e estabelecer governança; em contrapartida, tende a demandar maior esforço de implantação — além de competências especializadas.

2.6.3 MaxVista: foco em dashboards industriais e redução de esforço de disponibilização

O MaxVista é proposto com foco em disponibilização de *dashboards* orientados à manutenção e gestão de ativos, a partir de conexão direta ao banco do Maximo. Diferentemente de iniciativas tradicionais de BI, busca reduzir esforço inicial de modelagem extensa e acelerar o ciclo de entrega ao concentrar autenticação, gerenciamento de conexões e publicação de painéis em um único sistema. Nesse sentido, sua contribuição se dá principalmente pela padronização de consultas e pela entrega de painéis operacionais acionáveis, sem depender de uma cadeia completa de BI corporativo. Para consolidar essa comparação, a Tabela 1 apresenta uma síntese dos critérios discutidos nesta seção.

Tabela 1 – Comparação sintética entre BIRT, Cognos Analytics e MaxVista

Critério	BIRT (no Maximo)	Cognos Analytics	MaxVista
Finalidade predominante	Relatórios embutidos e formatados	BI corporativo governado e <i>dashboards</i>	<i>Dashboards</i> operacionais focados em Maximo
Tipo de entrega	Artefatos de relatório (saída definida)	<i>Dashboards</i> e relatórios em escala corporativa	Painéis temáticos e indicadores operacionais
Integração e esforço inicial	Integração nativa; customização requer <i>designer</i> e importação (IBM, 2023a)	Integração via fontes/modelos; usualmente exige governança (IBM, 2019)	Integração via parametrização de conexão e consultas
Principais forças	Padronização e documentação operacional	Governança, auditoria, escala e variedade de artefatos	Agilidade para disponibilizar painéis operacionais
Principais limitações	Menor aderência a monitoramento contínuo interativo	Pode demandar alto esforço de implantação e especialização	Exige maturidade de KPIs e atenção a segurança/ <i>performance</i>

Fonte: elaborado pelo autor.

Por fim, este capítulo fundamenta o problema e o contexto no qual o MaxVista se insere: apesar da riqueza do repositório transacional do Maximo, a conversão contínua desses dados em informação gerencial pode se tornar onerosa quando depende exclusivamente de relatórios estáticos ou de programas completos de BI corporativo. Assim, o trabalho se orienta a apresentar e discutir o MaxVista como ferramenta voltada a reduzir esforço de disponibilização e acelerar a geração de valor analítico a partir do EAM.

2.7 Considerações finais do capítulo

A revisão bibliográfica e documental consolidou os conceitos de gestão de ativos e EAM, situou o papel do IBM Maximo e MAS Manage como repositório transacional e discutiu abordagens recorrentes para transformar dados operacionais em informação de gestão. Também foram examinadas tendências e soluções concorrentes, bem como a comparação entre relatórios embutidos (BIRT), BI corporativo (IBM Cognos Analytics) e a necessidade de abordagens que reduzam esforço inicial e acelerem ciclos de evolução de painéis. Com essa base, o capítulo seguinte descreve o desenvolvimento do MaxVista, detalhando suas decisões arquiteturais e o fluxo de dados adotado para equilibrar padronização, desempenho e segurança no consumo de informações do Maximo.

3 Desenvolvimento

Organizações industriais intensivas em ativos demandam rastreabilidade, padronização e controle dos processos de manutenção e gestão de ativos. Esses processos impactam diretamente aspectos fundamentais, como disponibilidade, confiabilidade, segurança e custo da operação. Nesses casos, soluções de EAM são utilizadas para centralizar informações na forma de registros, possibilitando controle preciso da operação e auditoria. No entanto, em ambientes industriais com alta complexidade operacional, a existência do dado transacional, por si só, não garante ganhos operacionais. O valor da ferramenta depende da capacidade de transformar registros em indicadores consistentes para suportar decisões. Na área de EAM, a análise de dados costuma enfrentar desafios relacionados a regras de negócio, interpretações distintas de registros entre organizações, dependência de extrações manuais, relatórios desconectados entre áreas e dificuldade de rastreabilidade. Esse cenário reduz o potencial de impacto da ferramenta e afeta a tomada de decisão, fazendo com que decisões sejam tomadas tardiamente ou com dados de baixa confiabilidade. Nessa perspectiva, os *dashboards* devem ser entendidos como parte importante do mecanismo de gestão, pois evidenciam tendências, apoiam a priorização e reduzem o tempo de resposta. Para isso, a solução precisa equilibrar consistência semântica, desempenho, segurança de acesso e facilidade de evolução.

3.1 Abordagens existentes no ecossistema Maximo

Como citado anteriormente, no ecossistema Maximo é comum o uso de BIRT e, em alguns casos, também são adotadas soluções de BI, em geral o IBM Cognos Analytics (IBM, s.d.b; IBM, s.d.d). Ferramentas de relatórios embutidos, com desenvolvimento de novos relatórios para cada organização, atendem bem a demandas de documentação, auditoria e saídas formatadas. Soluções de BI atendem demandas de governança e distribuição ampla de dados para múltiplas fontes e áreas. Na prática, contudo, essas abordagens podem não otimizar o tempo de entrega de *dashboards* operacionais quando a organização necessita de painéis específicos de manutenção e ativos com rápida disponibilização e evolução contínua.

3.2 Visão geral do MaxVista

3.2.1 Objetivo e escopo da solução

O MaxVista é proposto como ferramenta orientada à disponibilização de *dashboards* a partir do repositório de dados do Maximo ou do MAS Manage. Sua operação baseia-se

na configuração direta de parâmetros de conexão ao banco de dados e, a partir disso, no fornecimento de *dashboards* que consomem consultas estruturadas e retornam dados no formato adequado para alimentação dos gráficos. Dessa maneira, o MaxVista visa reduzir o esforço para disponibilização de *dashboards* ao usuário final, ao concentrar em um único sistema: (i) gestão de usuários e permissões; (ii) cadastro de conexões com bancos do ambiente Maximo; e (iii) *dashboards* com gráficos e filtros, alimentados via *endpoints* (API) que consultam o banco e retornam dados estruturados. O funcionamento do MaxVista pode ser descrito por um fluxo lógico composto por quatro etapas principais.

3.2.2 Fluxo lógico de funcionamento

No MaxVista, o mecanismo de conexão foi tratado como o núcleo estruturante do projeto, pois é ele que transforma fontes heterogêneas do ecossistema Maximo em dados consistentes, rastreáveis e prontos para consumo. Em vez de incorporar regras específicas em cada painel, a solução consolida, em um componente genérico e parametrizável, a lógica de como os dados são acessados e preparados. Nesse modelo, os elementos variáveis ficam explicitados em configurações: tipo de banco — Oracle ou DB2 —, *driver*, *host*, porta, base ou *schema*, credenciais, limites e políticas de execução, além de opções operacionais como *timeouts* e validações. Essa parametrização é sustentada por uma persistência interna propositalmente enxuta e isolada do repositório transacional do Maximo, composta por entidades de controle e segurança: *DBCon* (usuários, sessão e perfis, com credenciais em *hash* e atributos de privilégio); *DBInfo* (catálogo de conexões externas associado 1 : 1 ao usuário, contendo parâmetros de conexão e credenciais protegidas); além de *Group* e *GroupUser*, para governança por grupos e associação $N : M$, com unicidade de vínculos. Nesse modelo, as credenciais de banco não são tratadas como detalhe periférico: elas permanecem cifradas via Fernet e só são disponibilizadas no instante da execução, reforçando o princípio de reduzir exposição e preservar segregação entre controle interno e dados operacionais do cliente.

A partir do momento em que uma conexão é selecionada, o *backend* executa a consulta em SQL de modo controlada e aplica um *pipeline* padronizado de processamento: coleta do *result set*, normalização de tipos e formatos — como números, datas e valores nulos —, sanitização, estruturação do retorno e geração de um *payload* JSON consistente, publicado via API. Esse *payload* funciona como um contrato de dados estável entre o *backend* e a camada de apresentação, removendo acoplamentos entre “origem” e “visualização” e permitindo que a evolução da interface seja guiada por composição, não por reescrita.

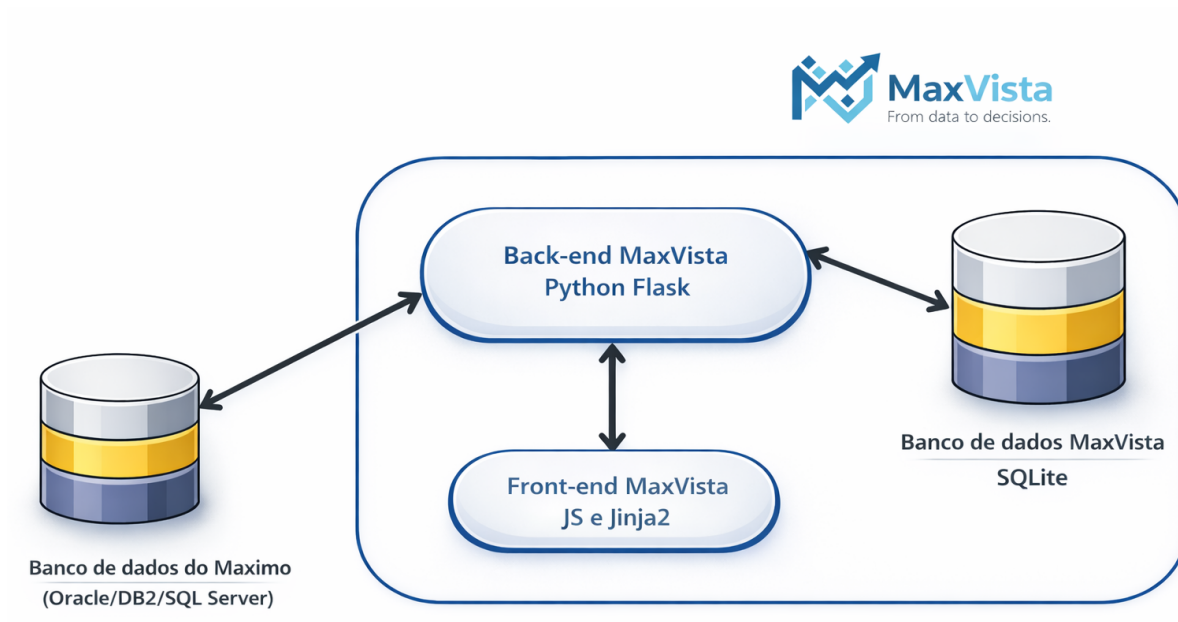
Por isso, embora o *frontend* atual utilize uma abordagem web com HTML/Jinja2 e gráficos com bibliotecas como Carbon Charts/D3.js, o projeto permanece aberto à substituição ou coexistência com outros *frameworks* e consumidores: qualquer tecnologia capaz de ler a API pode aproveitar os dados já estruturados — desde novas bibliotecas

de *dashboards* até integrações com ferramentas de BI, rotinas de auditoria, exportações automatizadas e serviços externos. Além disso, a arquitetura contempla a possibilidade de integração direta com o próprio Maximo, por exemplo via *automation scripts* para abrir o MaxVista dentro do sistema, o que amplia o valor do mecanismo de conexão: ele não se limita ao acesso e à extração de dados; consolida um ponto de integração entre ambientes, padroniza o contrato de entrega de informação e estabelece uma base arquitetural para extensões, integrações e componentes derivados do mesmo núcleo de tratamento.

1. Cadastro e autenticação do usuário: o sistema controla sessão e acesso por autenticação, restringindo a navegação a usuários autenticados ([Flask-Login](#), s.d.).
2. Parametrização de conexão ao banco: o usuário cadastra informações de conexão (nome, *driver*, *host*, porta, base e credenciais) para que o sistema consiga estabelecer comunicação com o banco do Maximo da organização.
3. Consulta e preparação do dado: ao requisitar um gráfico, o *backend* executa consultas em SQL, monta o conjunto de dados em estrutura de listas e dicionários e retorna em JSON por meio de uma chamada de API.
4. Renderização no *frontend*: o navegador consome a resposta da API e desenha os gráficos por biblioteca de visualização, aplicando filtros de interface, por exemplo seleção de registro por *site* ([MDN Web Docs](#), s.d.; [Carbon Charts](#), s.d.).

Esse fluxo separa com clareza a camada de consulta e a camada de visualização (Figura 4), facilitando a evolução da ferramenta sem exigir alterações estruturais nos mecanismos de autenticação e configuração.

Figura 4 – Visão geral de funcionamento do MaxVista



Fonte: elaborado pelo autor (2026).

3.2.3 Pipeline de dados do MaxVista

O *pipeline* de dados do MaxVista foi estruturado para transformar requisições analíticas em entregas padronizadas de informação, mantendo separação entre a camada de controle do produto e as fontes operacionais do cliente. O fluxo inicia-se no *frontend*, quando o usuário autenticado seleciona um painel e dispara uma requisição para um *endpoint* de dados. Essa requisição carrega o contexto mínimo de execução — identificador do painel/rota, parâmetros de filtro e, quando aplicável, a referência da conexão escolhida — sendo imediatamente associada à sessão ativa e às permissões derivadas dos vínculos de grupo.

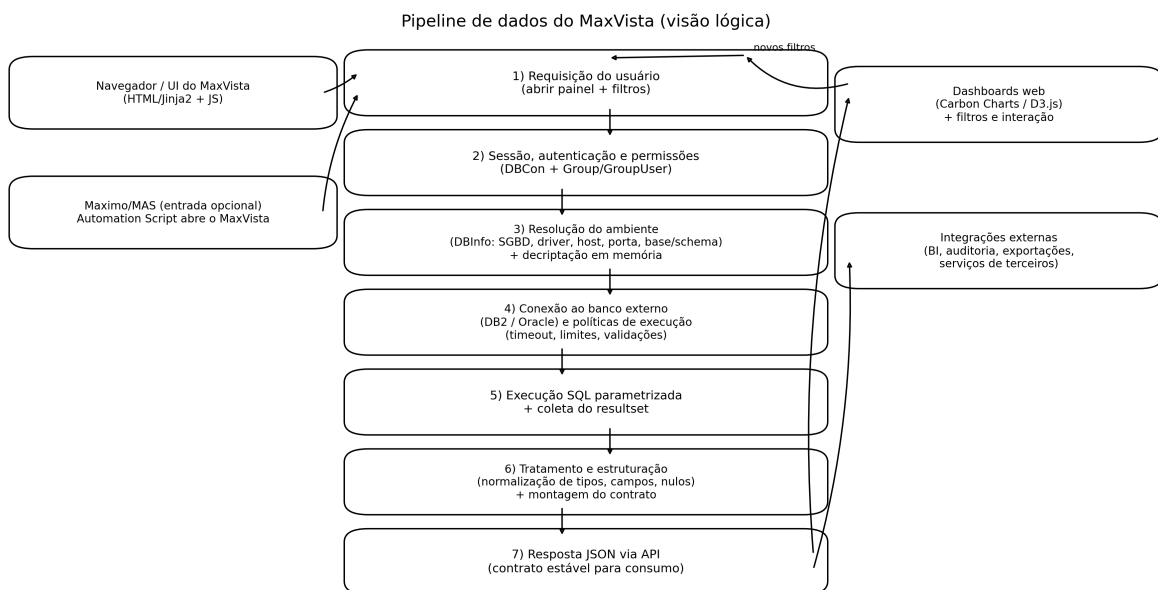
Na etapa seguinte, o *backend* resolve o contexto de execução consultando a persistência interna: a identidade do usuário é obtida a partir do *DBCon*, enquanto a conexão-alvo é selecionada em *DBInfo*, respeitando a relação 1:1 que permite uma única conexão por usuário. Em paralelo, as regras de autorização são aplicadas por meio das entidades *Group* e *GroupUser*, garantindo que apenas usuários autorizados acessem determinados painéis, fontes e conjuntos de informações. Com a conexão definida, o motor parametrizável compõe os parâmetros técnicos necessários para acesso, recupera as credenciais protegidas e as disponibiliza somente em memória no instante de execução, preservando o isolamento do repositório transacional do Maximo e reduzindo a superfície de exposição de segredos.

Com o contexto validado, o *pipeline* entra na fase de consulta. Nessa etapa, o *backend* executa instruções em SQL de modo controlado, aplicando os filtros recebidos e respeitando políticas operacionais como limites, *timeouts* e validações. O retorno bruto da consulta é então submetido a um *pipeline* padronizado de tratamento, no qual são normalizados tipos e formatos, ajustados aspectos de consistência — como padronização de chaves e campos — e estruturado um *payload* coerente para consumo. Esse processamento converte o resultado em um contrato de dados estável, desenhado para minimizar dependências entre a origem e a camada de visualização.

Na fase de entrega, o *payload* é exposto por meio de uma API HTTP no *backend*, retornando um JSON consistente que pode ser consumido tanto pelo *frontend* web quanto por consumidores externos. O *frontend*, por sua vez, recebe o JSON e o converte em elementos visuais, aplicando bibliotecas Carbon Charts/D3.js e refletindo interações do usuário em novas requisições com parâmetros atualizados, fechando o ciclo de exploração.

Por fim, a padronização do contrato e a separação entre integração e apresentação estendem o *pipeline* para além do uso em *dashboards*. Como a entrega é feita via API com dados previamente estruturados, o mesmo fluxo pode alimentar integrações externas, como ferramentas de BI, rotinas automatizadas de auditoria, exportações programáticas e serviços de terceiros. Adicionalmente, a arquitetura suporta integração com o próprio Maximo/MAS, por exemplo por meio de *automation scripts* que abrem o MaxVista a partir do sistema, permitindo que a exploração e a visualização ocorram de maneira contextualizada, sem alterar o repositório transacional (Figura 5).

Figura 5 – Pipeline MaxVista



Fonte: elaborado pelo autor (2026).

3.3 Modelo de dados

O modelo de dados do MaxVista foi projetado para sustentar as funções essenciais da aplicação sem interferir no repositório transacional do Maximo ou do MAS. Para isso, a persistência interna mantém apenas informações de controle e configuração, agrupadas em quatro entidades principais. A entidade *DBCon* representa os usuários (contas de acesso ao sistema), armazenando identificadores, credenciais em formato de *hash* e atributos de privilégio, o que viabiliza autenticação, controle de sessão e diferenciação de perfis, como o superusuário. Associada a ela, a entidade *DBInfo* armazena os parâmetros necessários para conexão aos bancos externos do ambiente Maximo, como *driver*, *host*, porta, base e credenciais. Essa relação é modelada como 1 : 1, permitindo que um usuário tenha apenas uma conexão cadastrada, preservando a separação entre dados da plataforma e dados operacionais do cliente; adicionalmente, as credenciais de conexão são persistidas de modo protegido, sendo descritografadas apenas no momento da execução das consultas. Usuários administrativos em nível de aplicação são os únicos capazes de acessar e visualizar todas as conexões cadastradas no sistema.

A gestão de permissões por agrupamento é representada pela entidade *Group*, que define conjuntos lógicos de acesso, e pela entidade associativa *GroupUser*, responsável por materializar a participação de usuários em grupos. Essa abordagem implementa a relação $N : M$ entre usuários e grupos, assegurando flexibilidade de administração e suporte à governança de acesso aos recursos do sistema. Para evitar duplicidade de vínculos, a tabela *GroupUser* impõe unicidade no par (`group_id`, `user_id`), garantindo que um usuário não seja associado mais de uma vez ao mesmo grupo. Essa parametrização de usuários por grupo é inspirada na estrutura do Maximo, que segue padrão equivalente. Em síntese, o diagrama evidencia um núcleo de persistência voltado à governança — usuários, grupos e permissões — e à parametrização segura de conexões, compondo a base necessária para que os *dashboards* sejam disponibilizados por meio de consultas executadas sob controle de acesso e rastreabilidade, a partir do mesmo núcleo de integração e tratamento (Figura 6).

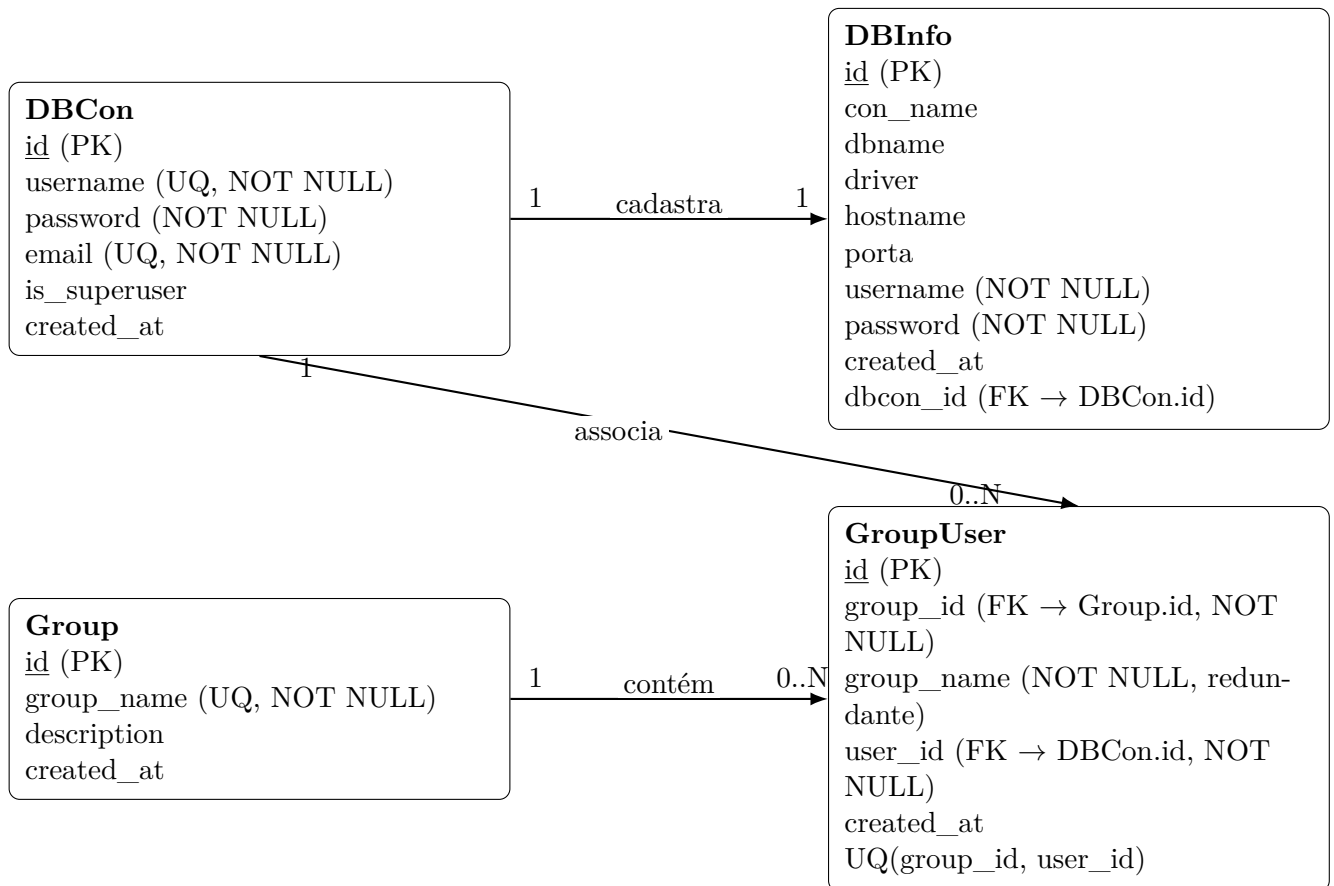


Figura 6 – Diagrama entidade-relacionamento (ER) do sistema. A relação N:M entre usuários (DBCon) e grupos (Group) é modelada pela tabela associativa GroupUser.

3.4 Requisitos e considerações do desenvolvimento

Quanto aos requisitos funcionais, estão implementados: autenticação de usuários, com gestão de conta e recuperação de senha; cadastro e atualização de parâmetros de conexão, incluindo *driver* e credenciais; controle de acesso por usuário, com possibilidade de um superusuário visualizar conexões globais; e rotas de *dashboards* que retornam dados para gráficos e painéis em páginas distintas de análise. Em relação aos requisitos não funcionais relevantes para o ambiente industrial, destacam-se: segurança, com proteção de credenciais e controle de acesso; desempenho, com consultas eficientes, especialmente em grandes históricos; auditabilidade, com rastreabilidade de acessos e alterações de configuração; e manutenibilidade, com estrutura modular que facilita a adição de novos *dashboards* e consultas, utilizando rotas e *blueprints* (Pallets Projects, s.d.d).

3.5 Tecnologias utilizadas

A Tabela 2 sintetiza as tecnologias adotadas na implementação do MaxVista por camada e sua finalidade, servindo como referência para a descrição detalhada apresentada a seguir.

Backend: A implementação foi feita em Python, utilizando o *microframework* Flask para suportar a aplicação web (Pallets Projects, s.d.a). O projeto está separado entre inicialização da aplicação, modelos de dados e rotas, com o uso de *blueprints* para modularização das funcionalidades, como *login*, configurações de banco, *dashboards* e grupos (Pallets Projects, s.d.d). O controle de autenticação e sessão foi feito utilizando Flask-Login, permitindo restringir acesso a rotas e carregar usuários a partir de um identificador persistente (Flask-Login, s.d.). Como ferramenta de persistência interna, foi utilizado SQLAlchemy com banco de dados SQLite, armazenando informações da aplicação (SQLAlchemy, s.d.; Flask-SQLAlchemy, s.d.).

Persistência e modelos de dados: A solução mantém um banco interno para armazenar as contas de usuário, cadastro de conexões e grupos com suas respectivas relações. Com essa abordagem, os dados da plataforma ficam completamente separados dos dados operacionais do Maximo, que permanecem sem alterações no banco de dados do cliente.

Segurança: São utilizados dois mecanismos complementares: (i) *hash* de senha do usuário, realizado com funções de Werkzeug, reduzindo o risco de exposição de senha em texto puro (Pallets Projects, s.d.e); e (ii) criptografia simétrica das senhas de conexão, em que as credenciais do banco são armazenadas e criptografadas por Fernet, com chave definida em variável de ambiente e carregada via *python-dotenv* (PyCA Cryptography, s.d.; python-dotenv, s.d.). Na execução das consultas, a senha é descriptografada e a conexão é estabelecida.

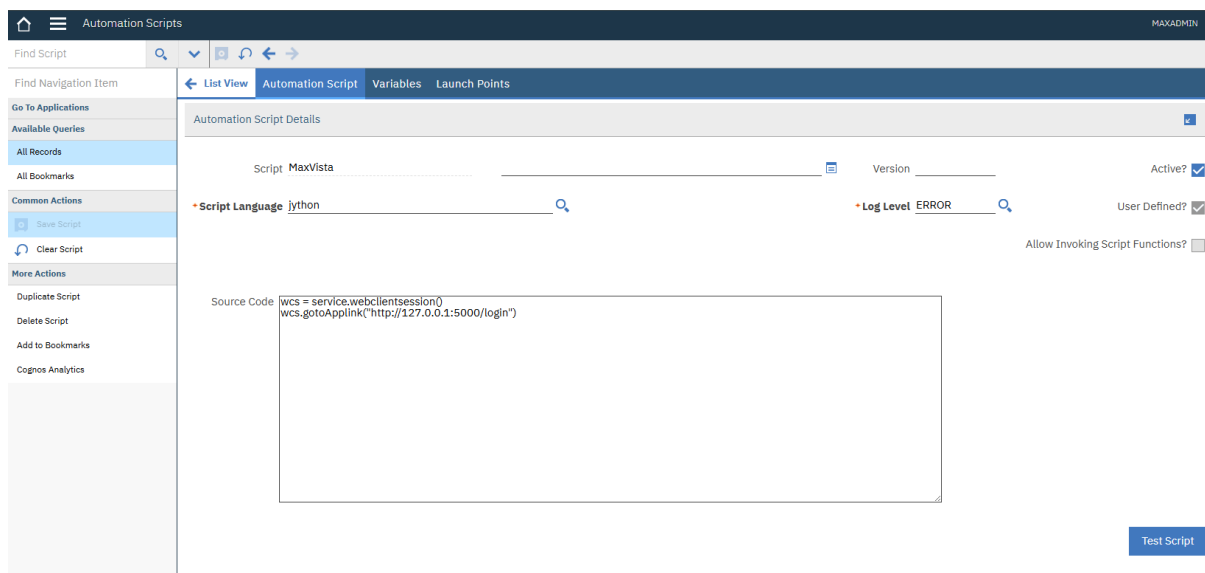
Conexão com banco do Maximo: Para o acesso ao banco de dados do Maximo, o software contempla os *drivers* de conexão dos bancos prioritariamente utilizados: IBM DB2 e Oracle. No DB2, a conexão é feita pela biblioteca `ibm_db` e requer o IBM Data Server Driver no ambiente (IBM, s.d.c; IBM, s.d.h). No Oracle, a conexão é realizada via *driver* `oracledb`, iniciando pelo carregamento do Oracle Instant Client, que aponta para a instalação no diretório local (Oracle, s.d.).

Frontend: Como base, o *frontend* utiliza *templates* HTML com Jinja2 (nativo do Flask), arquivos CSS e JavaScript por página para estilização, consumo de dados e renderização dos gráficos (Pallets Projects, s.d.b; Pallets Projects, s.d.c). Foram utilizados o IBM Carbon Design System para estilização e o Carbon Charts para renderização dos gráficos (Carbon Design System, s.d.; Carbon Charts, s.d.). O padrão utilizado para atualização dos gráficos é a solicitação de dados via API, recebendo JSON com os dados solicitados e instanciando os gráficos a partir dessas informações (MDN Web Docs, s.d.).

Integração com o Maximo: Considerando os recursos de customização disponibilizados pelo IBM Maximo, é possível implementar integrações que incorporem o acesso ao MaxVista ao próprio ambiente de operação do sistema. Para este trabalho, foi desenvolvido um *automation script* e realizado seu acoplamento por navegação independente (Figuras 7

e 8), viabilizando a abertura do MaxVista a partir do Maximo e reduzindo a necessidade de acesso externo por navegação independente.

Figura 7 – Script de automação para abertura do MaxVista



Fonte: elaborado pelo autor (2026).

Figura 8 – Ação associada ao objeto para abertura do MaxVista

Launch Point	Description	Launch Point Type	Object	Active?
MAXVISTA		ACTION	WORKORDER	<input checked="" type="checkbox"/>

Fonte: elaborado pelo autor (2026).

Tabela 2 – Tecnologias identificadas na implementação do MaxVista

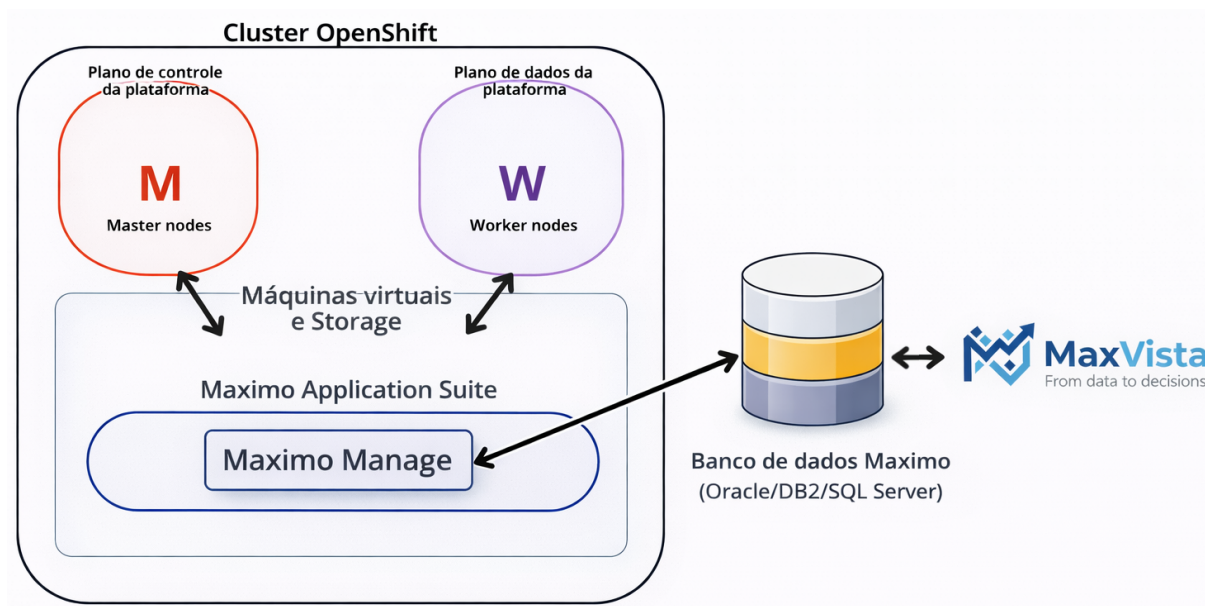
Camada	Tecnologia	Finalidade no MaxVista
Backend	Python + Flask	Servidor web, rotas, templates e organização modular.
Autenticação	Flask-Login	Controle de sessão e restrição de acesso.
Persistência interna	Flask-SQLAlchemy/SQLAlchemy + SQLite	Armazenamento de usuários, grupos e configurações.
Segurança de usuário	Werkzeug (hash)	Hash de senha de usuário.
Segurança de dados de conexão	cryptography (Fernet) + python-dotenv	Criptografia de credenciais e carregamento de chave por ambiente.
Integração DB2	ibm_db + IBM DB2 ODBC Driver	Conexão e consultas ao DB2 do Maximo.
Integração Oracle	Estrutura com oracledb + Instant Client	Base para conexão Oracle (a consolidar).
Frontend	HTML/Jinja2 + CSS/JS	Interface e navegação.
Visualização	Carbon Styles + Carbon Charts + D3.js	Componentes visuais e gráficos interativos.
Integração frontend-backend	Fetch + JSON	Consumo dos dados das APIs para montar gráficos.

Fonte: elaborado pelo autor.

3.6 Utilização de dados do Maximo

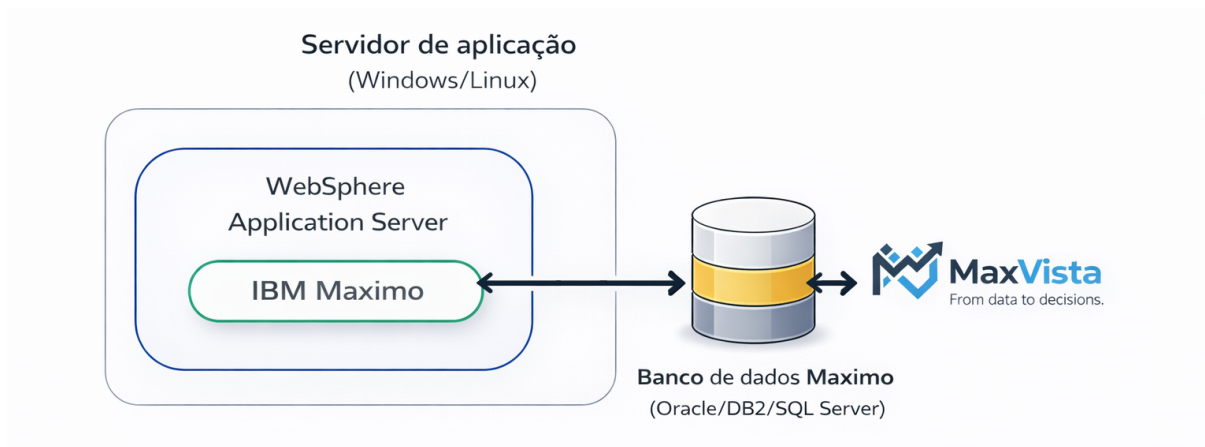
O MaxVista se conecta ao Maximo como uma aplicação externa e desacoplada do ambiente transacional, consumindo dados por conexão em SQL com credenciais dedicadas. Na prática, o MaxVista mantém um cadastro de conexões por ambiente — *host*, porta, SGBD, base e usuário —, armazena as credenciais de modo protegida e abre a sessão apenas no momento de execução das consultas, encerrando-a após as ações, priorizando operações de leitura e filtragens que evitem impacto no processamento operacional. Esse desenho permite alternar entre múltiplos ambientes — desenvolvimento, homologação e produção —, aplicar regras de permissão por grupo e manter rastreabilidade das execuções, sem exigir customização no Maximo e preservando a separação entre a camada analítica do MaxVista e o repositório transacional do EAM. Dessa forma, obtém-se aderência rápida à ferramenta, permitindo que o MaxVista seja implantado por organização com baixo esforço operacional. Por operar de maneira desacoplada do ambiente transacional, sua entrada em produção tende a ocorrer sem exigir janelas de indisponibilidade do Maximo e sem demandar alocação significativa de recursos adicionais, reduzindo riscos e acelerando a adoção pelos usuários. Como ilustrado nas Figuras 9 e 10, o MaxVista adota a mesma lógica de integração tanto no MAS quanto no Maximo 7.x, preservando um modelo consistente de acesso entre as duas plataformas.

Figura 9 – Conexão do MaxVista com MAS



Fonte: elaborado pelo autor (2026).

Figura 10 – Conexão do MaxVista com Maximo 7.x



Fonte: elaborado pelo autor (2026).

3.7 Considerações finais do capítulo

Este capítulo detalhou a concepção e o desenvolvimento do MaxVista, com ênfase na separação entre camada de consulta e preparação de dados e camada de visualização, na persistência interna voltada a controle e governança e no contrato de dados em JSON exposto por APIs. As decisões descritas buscam reduzir acoplamentos entre fonte e interface, permitindo evolução incremental de *dashboards* e integração com diferentes consumidores, ao mesmo tempo em que mantêm proteção de credenciais e controles de acesso compatíveis com o contexto industrial. No capítulo seguinte, essas escolhas são confrontadas com evidências de funcionamento e avaliação técnica, por meio da apresentação dos painéis

implementados e de testes operacionais que observam comportamento, desempenho e impacto no banco de dados do Maximo.

4 Resultados

Este capítulo apresenta os resultados obtidos com o desenvolvimento do MaxVista e sua aplicação na geração e disponibilização de *dashboards* a partir de dados do IBM Maximo ou do MAS Manage. A análise foi estruturada para ilustrar, de forma objetiva, o que foi entregue, como a solução se comporta tecnicamente e quais evidências sustentam sua utilidade no contexto industrial. Para isso, os resultados são discutidos em três frentes: (i) verificação funcional, baseada nos painéis implementados e nos indicadores disponibilizados; (ii) avaliação técnica, com ênfase em conectividade, controle de acesso e proteção de credenciais; e (iii) observações operacionais relacionadas a desempenho, estabilidade e requisitos para uso em cenários com maior volume de dados.

4.1 Ambiente e condições de validação

A validação do MaxVista foi conduzida por meio da execução da aplicação em ambiente de desenvolvimento e conexão a uma base do Maximo, utilizando os conectores implementados. O objetivo desta validação foi comprovar a capacidade do sistema de: (a) estabelecer conexão parametrizada ao banco; (b) executar consultas em SQL para extração e agregação de dados; e (c) renderizar *dashboards* interativos no *frontend* a partir de respostas em JSON retornadas por *endpoints*.

Para padronizar a apresentação dos resultados, foi utilizado o mesmo ambiente durante todo o processo, com as seguintes especificações:

- Banco do Maximo: IBM DB2;
- Infraestrutura de execução do MaxVista (máquina/VM): servidor local;
- Navegador utilizado para visualização: Google Chrome.

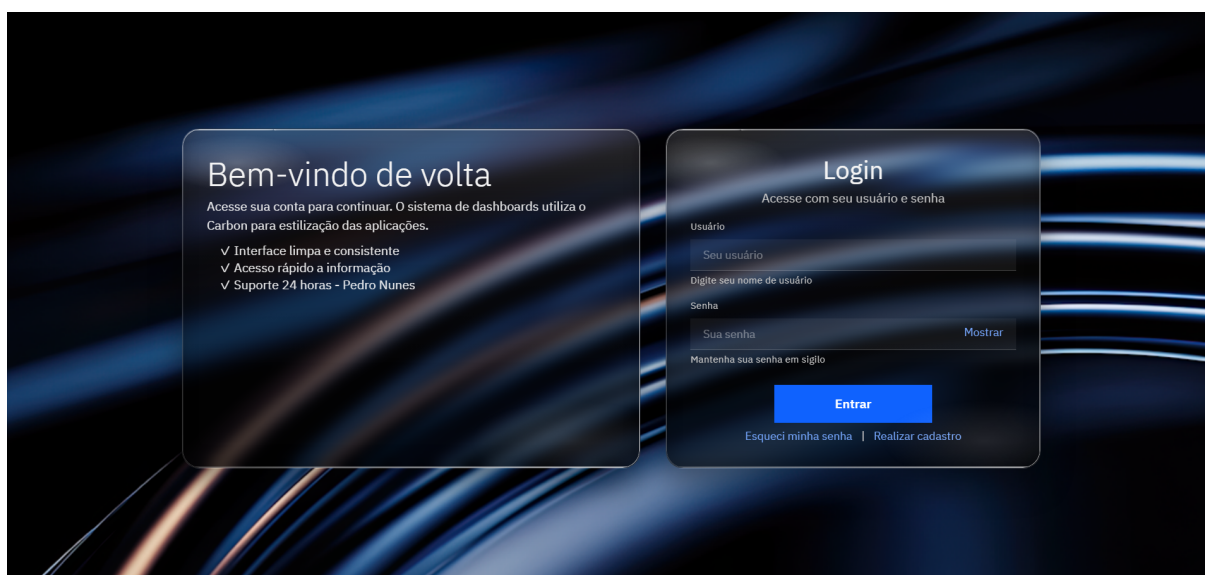
4.2 Resultados: funcionalidades, painéis e indicadores disponibilizados

Os resultados funcionais do MaxVista são evidenciados pela disponibilização de painéis temáticos e pela renderização de gráficos alimentados por consultas ao banco do Maximo. Em termos práticos, o sistema demonstrou: autenticação, seleção de conexão cadastrada, consumo de *endpoints* de dados e montagem de visualizações no padrão definido de interface.

4.2.1 Gerência de usuários

A gerência de usuários no MaxVista foi concebida como um componente central para viabilizar a utilização da ferramenta em ambientes corporativos, nos quais é necessário controlar acesso, segmentar permissões e garantir rastreabilidade das ações realizadas no sistema. Assim, foram implementadas funcionalidades de autenticação, recuperação de credenciais e administração de perfis, além de mecanismos de associação a grupos e gerenciamento de conexões com bancos de dados do IBM Maximo/MAS Manage. As Figuras 11, 12 e 13 apresentam as telas responsáveis pelas ações gerais do usuário no MaxVista: autenticação no sistema, registro de nova conta e recuperação de senha. Em conjunto, essas interfaces evidenciam o fluxo básico de acesso à aplicação e o suporte à gestão de credenciais.

Figura 11 – Painel de *login* do MaxVista



Fonte: elaborado pelo autor (2026).

Figura 12 – Painel de registro de usuários do MaxVista

Bem-vindo ao nosso sistema

Ao se cadastrar, você terá acesso a uma plataforma segura e personalizada. Seus dados serão utilizados apenas para fins de autenticação e comunicação com nossa equipe.

- Armazenamento seguro dos seus dados
- Acesso à área do usuário com painel interativo
- Tema escuro otimizado para contraste

* Todos os campos são obrigatórios

Cadastro de novo usuário
Preencha os dados abaixo para criar sua conta

Usuário
Digite seu nome de usuário

E-mail
Digite seu e-mail

Senha
Crie uma senha forte

Realizar cadastro Voltar

Ao cadastrar-se, você concorda com nossos [Termos de Uso](#) e [Política de Privacidade](#).

Fonte: elaborado pelo autor (2026).

Figura 13 – Painel de recuperação de senha do MaxVista

Redefina sua senha com segurança

Informe o e-mail cadastrado, escolha uma nova senha e confirme a senha para garantir que você tenha controle seguro sobre sua conta. Evite reutilizar senhas antigas ou fracas.

- Utilize uma senha com no mínimo 8 caracteres
- Combine letras, números e símbolos
- Evite dados pessoais ou senhas previsíveis

Recuperação de Senha
Preencha os campos abaixo para redefinir sua senha

E-mail cadastrado
Digite seu e-mail

Enviamos um código de verificação

Nova senha
Digite a nova senha

Use letras, números e símbolos

Confirmação da nova senha
Confirme a nova senha

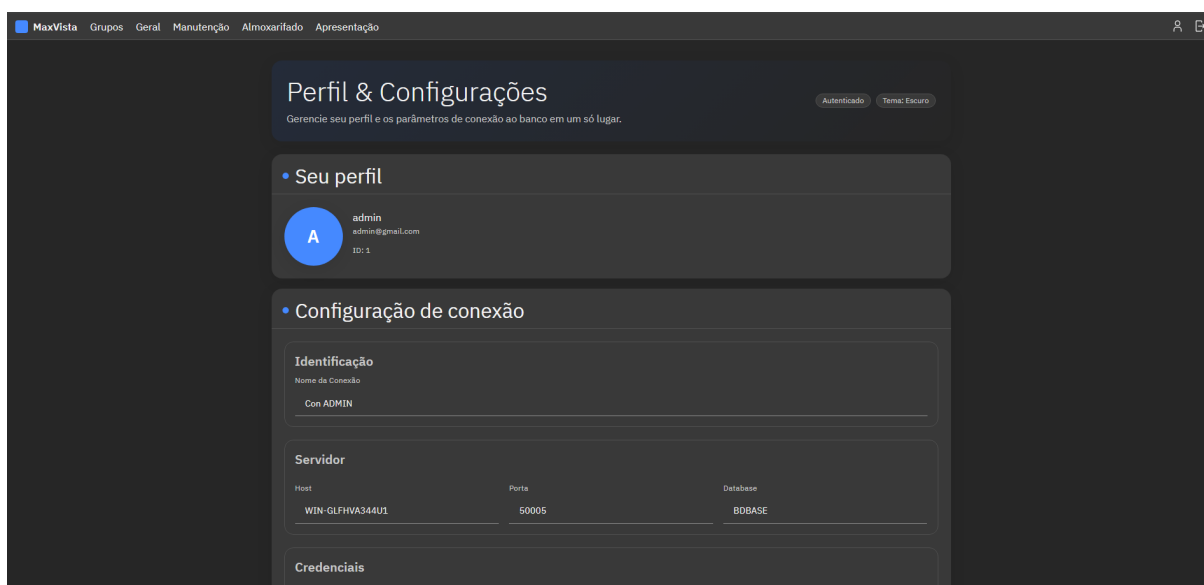
Recuperar minha senha Voltar

Se não lembrar o e-mail, entre em contato com o suporte.

Fonte: elaborado pelo autor (2026).

Outro aspecto relevante da gerência de usuários (Figura 14) é a possibilidade de cadastro e manutenção de conexões a bancos de dados (Figura 15). Cada usuário pode registrar parâmetros de conexão com um único banco de dados — exceto usuários administrativos, que podem acessar todas as conexões registradas no sistema — permitindo que o MaxVista execute consultas e alimente os *dashboards* a partir da instância correspondente do Maximo. Para mitigar riscos, as credenciais de conexão são tratadas de forma protegida, utilizando criptografia simétrica para armazenamento, de modo que o sistema não mantenha senhas em texto puro.

Figura 14 – Página de perfil do usuário



Fonte: elaborado pelo autor (2026).

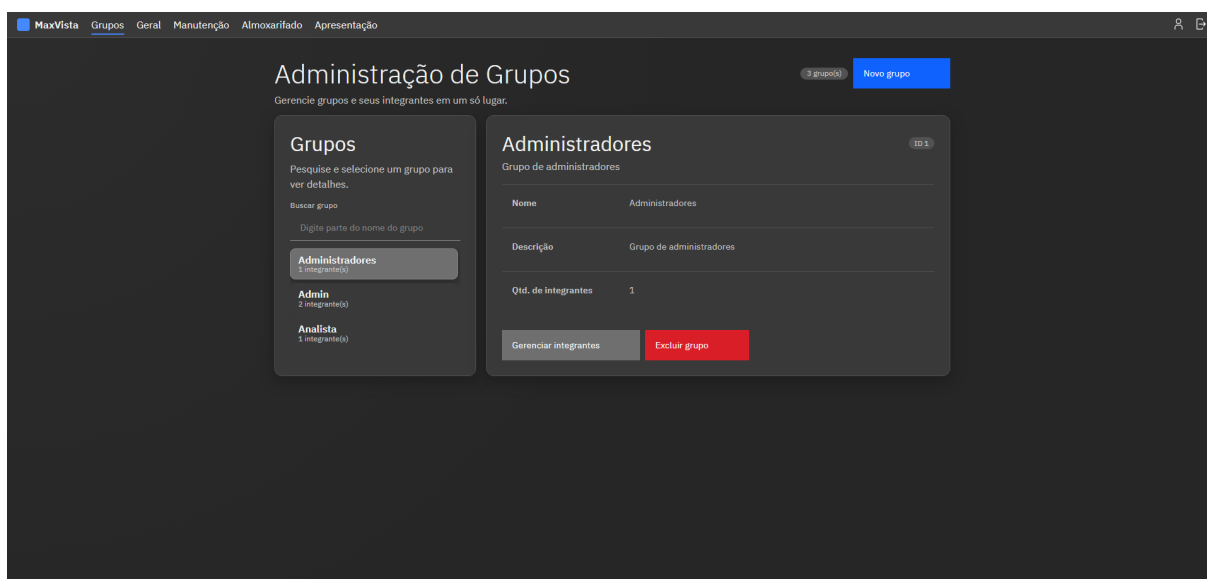
Figura 15 – Painel de conexão com banco de dados



Fonte: elaborado pelo autor (2026).

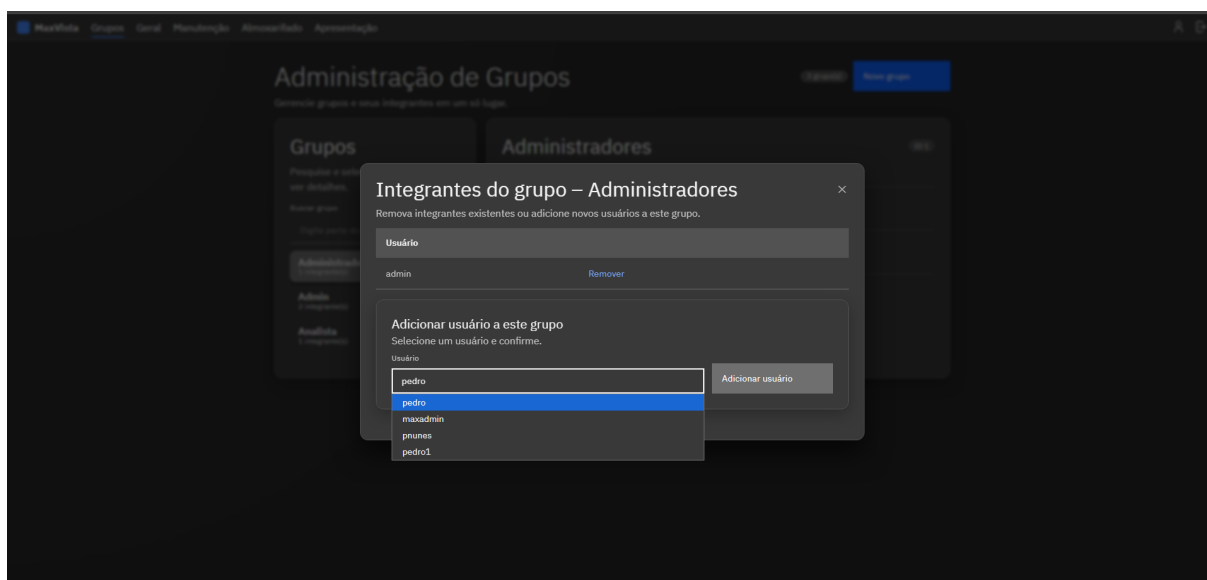
A solução também contempla organização de usuários por grupos, permitindo estruturar permissões e acesso a recursos de forma alinhada ao que foi contratado por cada organização e com base em aspectos de organização interna, como grupos por unidade, contrato, área funcional ou perfil de atuação. A associação de usuários a grupos e a administração dessas relações são realizadas por um usuário administrativo, responsável por criar grupos, atribuir usuários e controlar permissões relacionadas ao uso das conexões e dos painéis, conforme ilustrado nas Figuras 16, 17 e 18.

Figura 16 – Painel de gerenciamento de grupos



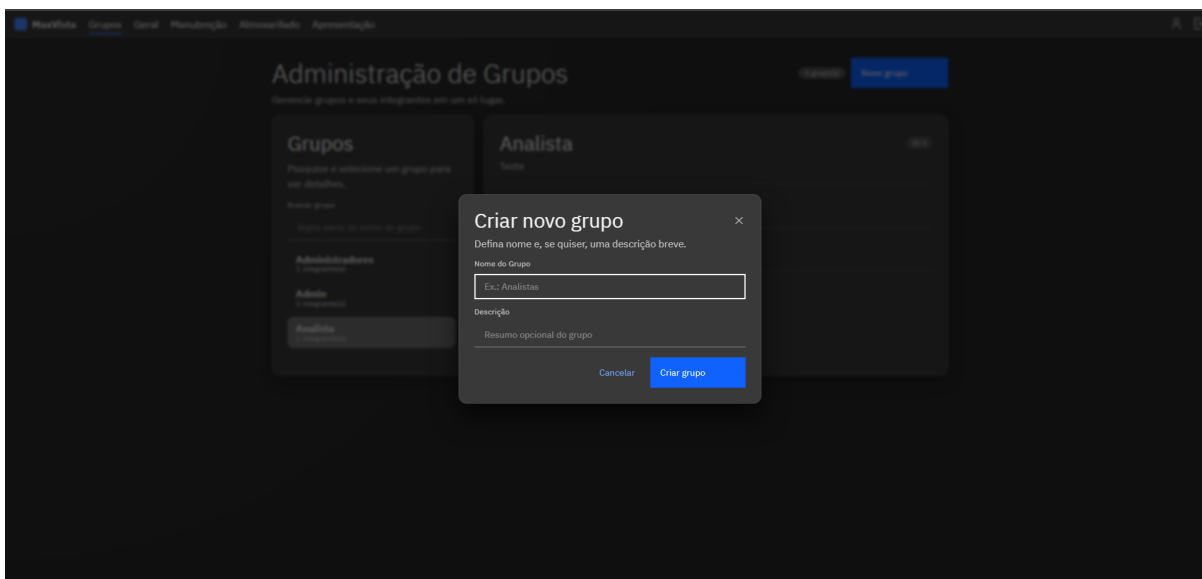
Fonte: elaborado pelo autor (2026).

Figura 17 – Painel de gerenciamento de integrantes



Fonte: elaborado pelo autor (2026).

Figura 18 – Painel de criação de grupos

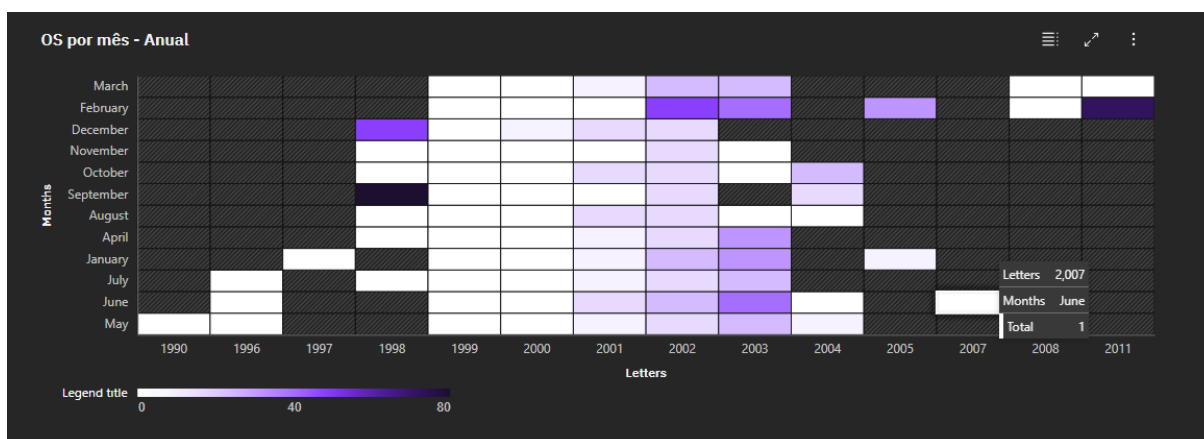


Fonte: elaborado pelo autor (2026).

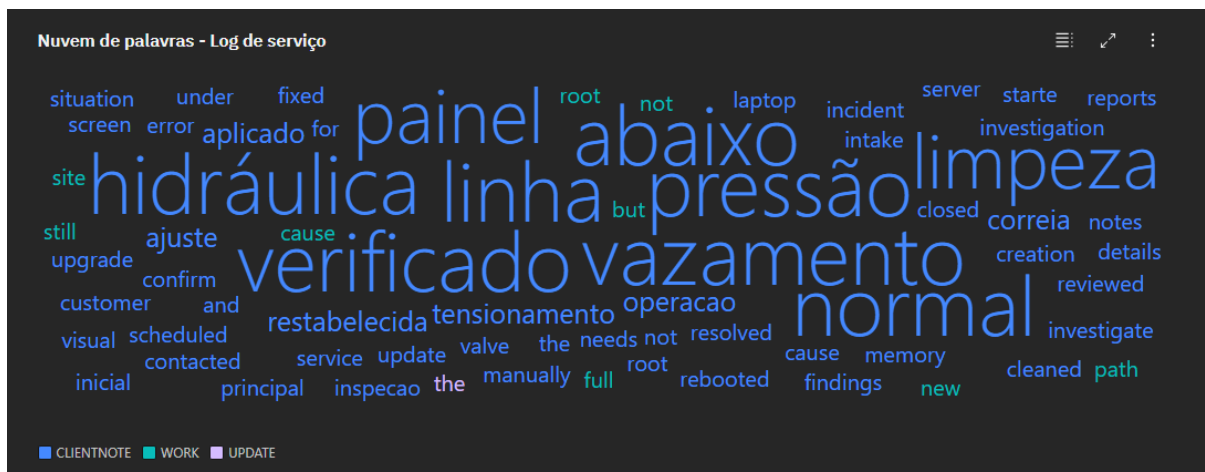
4.2.2 Painel geral

O painel geral consolida indicadores de volume, tendências e distribuição, com o objetivo de apoiar acompanhamento gerencial e operacional. Os gráficos demonstram a capacidade do MaxVista de apresentar recortes por site, agregações temporais e composição por categorias do processo. Esse painel foi elaborado para indicar a construção de diferentes tipos de *dashboards*, conforme ilustrado nas Figuras 19, 20 e 21.

Figura 19 – Painel de visualização anual de ordens de serviço

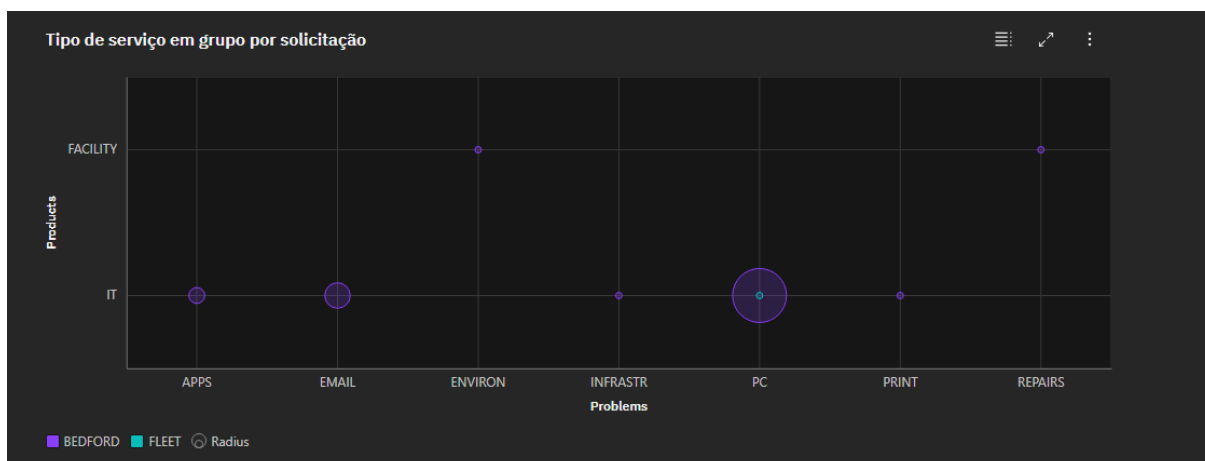


Fonte: elaborado pelo autor (2026).

Figura 20 – Painel de nuvem de palavras em *log* de serviço

Fonte: elaborado pelo autor (2026).

Figura 21 – Tipo de serviço por solicitação



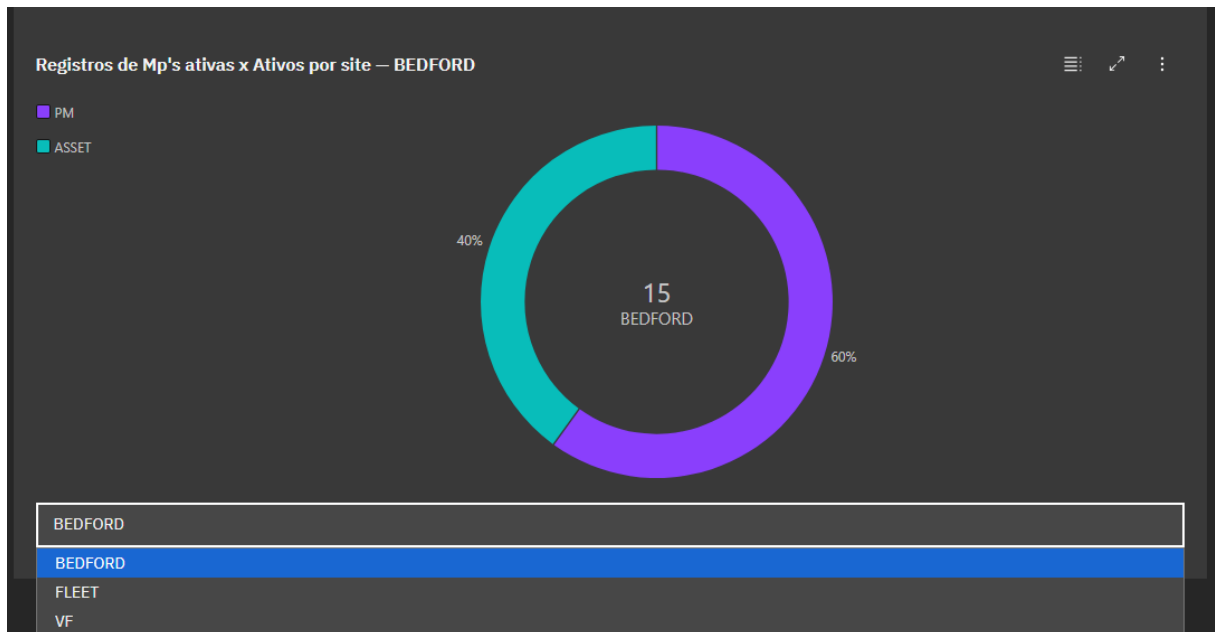
Fonte: elaborado pelo autor (2026).

Como resultado, observa-se que o painel geral atende ao objetivo de prover uma visão consolidada do comportamento de execução e do volume de ordens, permitindo identificar sazonalidade, variações por unidade e possíveis gargalos, servindo como base para priorização e análise de desvios.

4.2.3 Painel de manutenção: manutenção corretiva e concentração por ativo/local

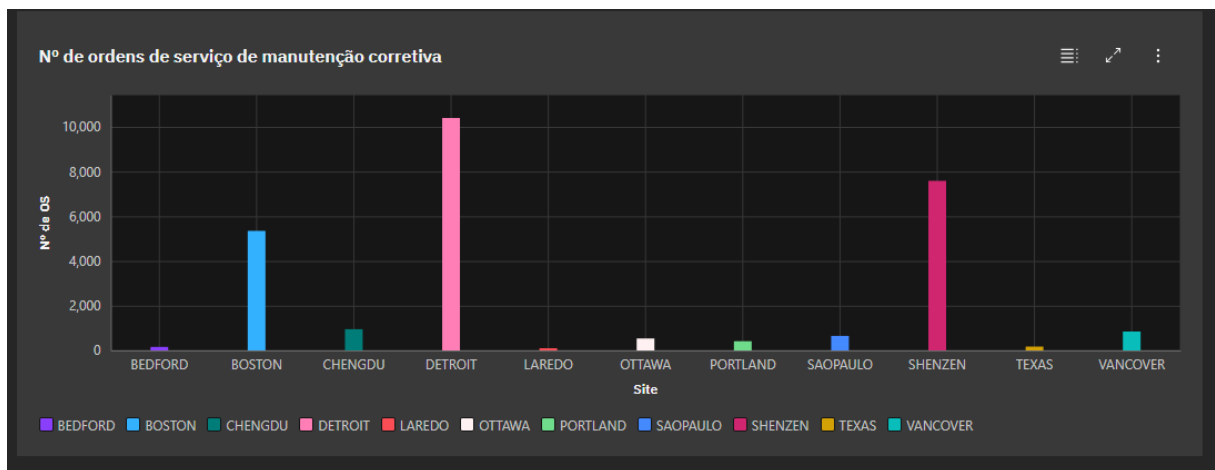
O painel de manutenção evidencia a exploração de corretivas e a identificação de concentração por ativo e por localização, tarefa relevante para atividades de confiabilidade e engenharia de manutenção. Ao concentrar a visão em ordens corretivas, o painel facilita o rastreamento de recorrência e direciona ações para ativos e áreas com maior incidência, apoiando estratégias de manutenção preditiva e preventiva, conforme ilustrado nas Figuras 22, 23 e 24.

Figura 22 – Painel de visualização de manutenções preventivas ativas por site



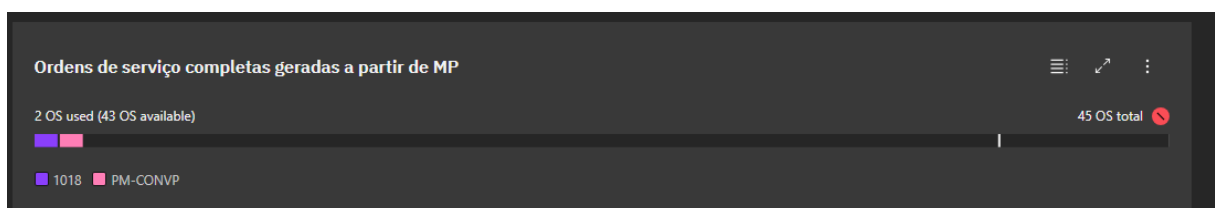
Fonte: elaborado pelo autor (2026).

Figura 23 – Painel de ordens de serviço corretivas por site



Fonte: elaborado pelo autor (2026).

Figura 24 – Painel de ordens de serviço por manutenção preventiva completas



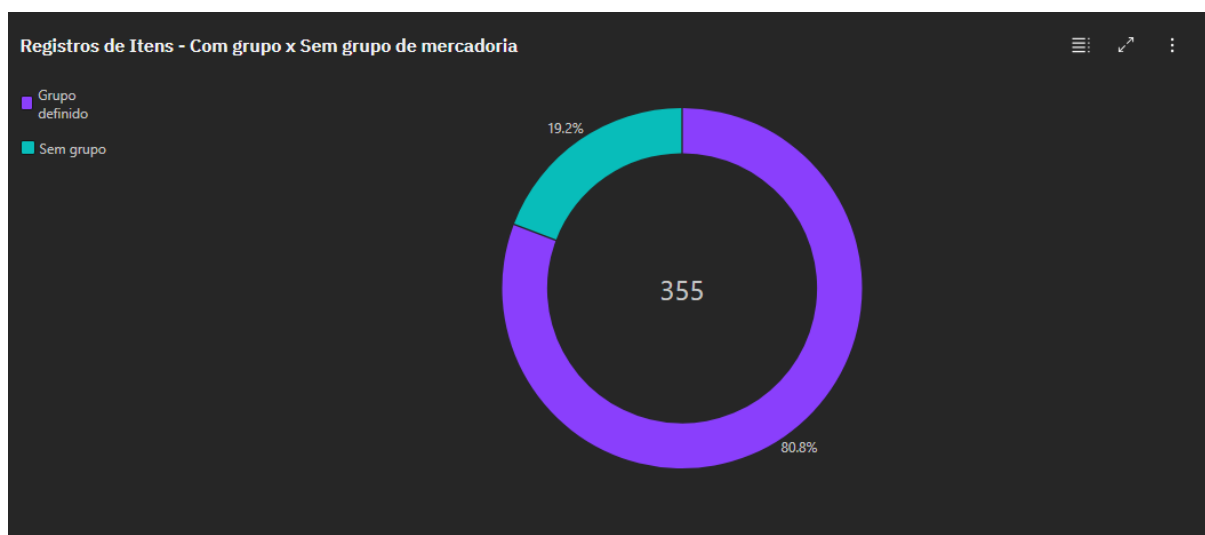
Fonte: elaborado pelo autor (2026).

O resultado funcional central do painel de manutenção é ilustrar que o MaxVista consegue executar agregações, ordenar resultados e limitar o conjunto retornado, fornecendo uma visualização diretamente acionável para priorização de RCA e revisão de estratégia de manutenção.

4.2.4 Painel de almoxarifado: materiais, inventário e impacto do status WMATL

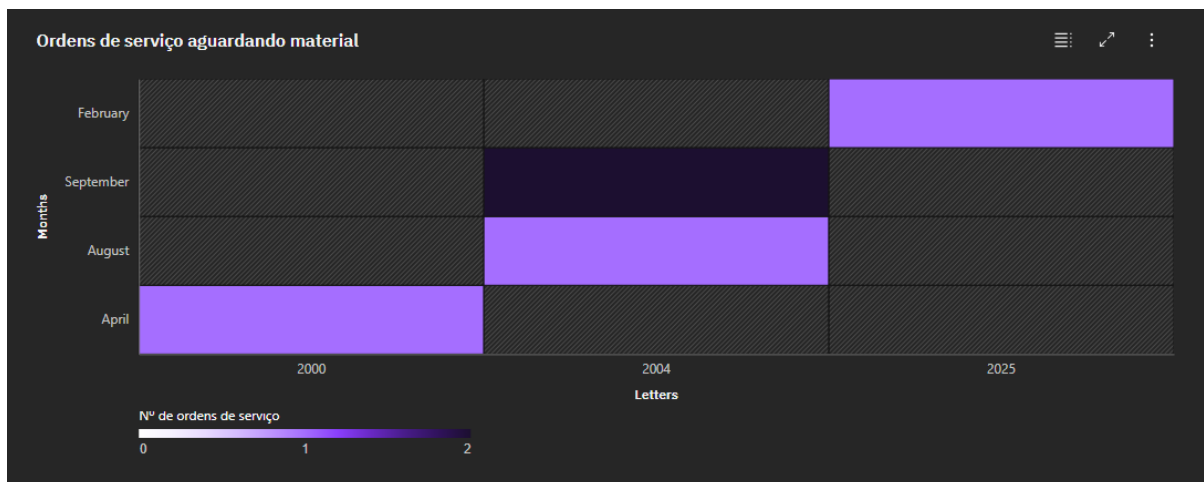
O painel de almoxarifado foi validado como mecanismo de leitura de ordens aguardando material e de verificação de completude cadastral do inventário, em campos associados a *commodity/commodity group*. Esses elementos são particularmente importantes em organizações nas quais a indisponibilidade de material impacta diretamente o *backlog* e a execução do plano de manutenção, conforme ilustrado nas Figuras 25, 26 e 27.

Figura 25 – Painel de relação de grupos de itens



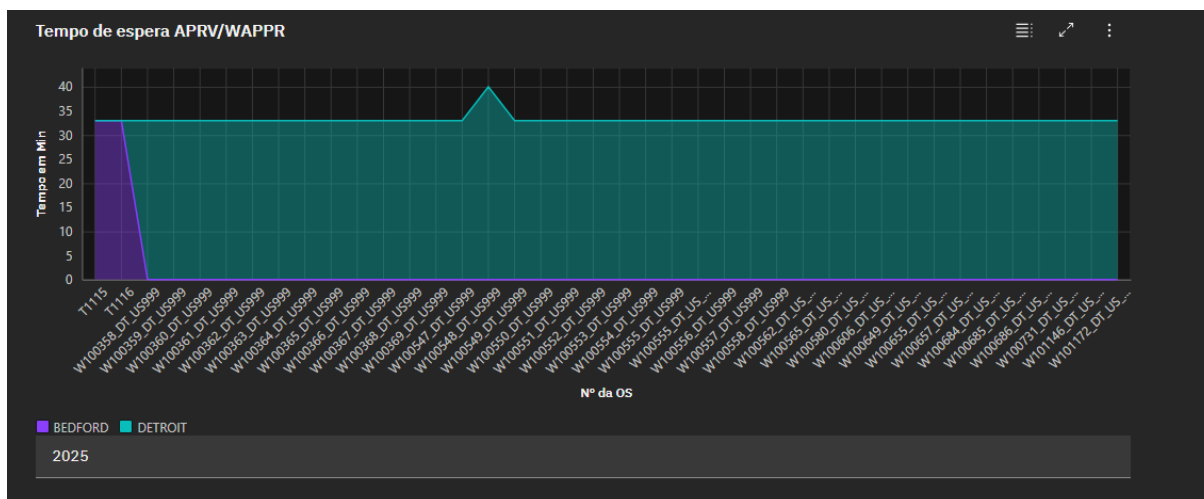
Fonte: elaborado pelo autor (2026).

Figura 26 – Painel de ordens de serviço aguardando material por ano



Fonte: elaborado pelo autor (2026).

Figura 27 – Painel de tempo de espera de material por ordem de serviço



Fonte: elaborado pelo autor (2026).

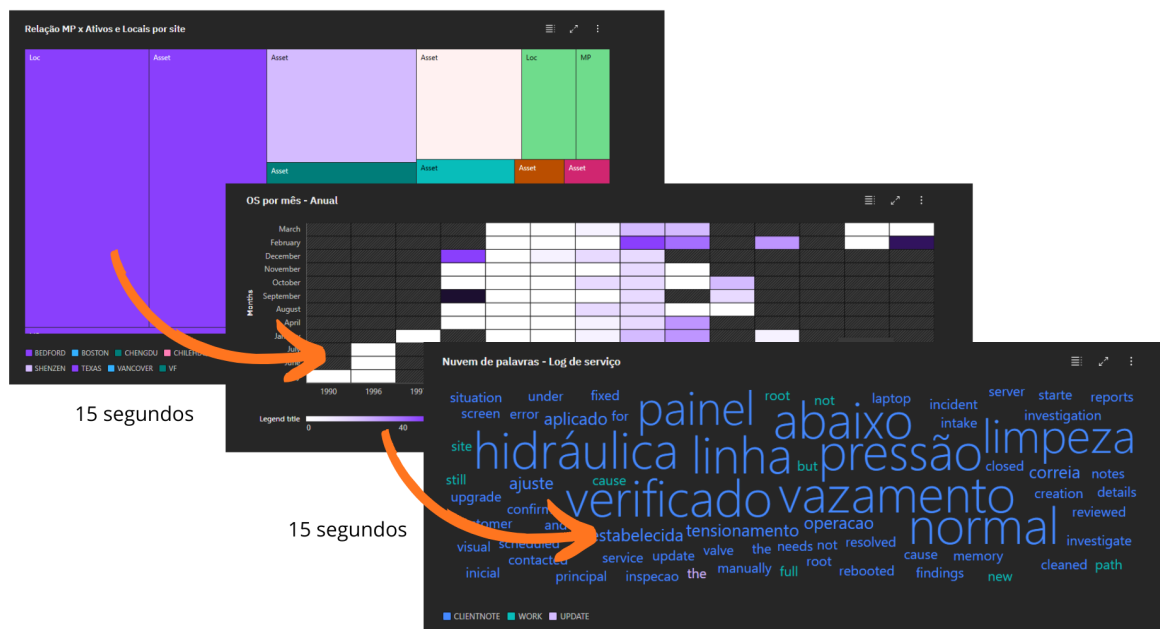
Os resultados do painel de almoxarifado reforçam a aplicabilidade do MaxVista para visões interáreas (manutenção e suprimentos), evidenciando que a mesma base de dados do EAM pode sustentar indicadores que correlacionam execução de ordens de serviço com disponibilidade de recursos e qualidade de cadastro.

4.2.5 Modo apresentação

Também foi implementado um modo de apresentação voltado à exibição dos *dashboards* em formato adequado para acompanhamento contínuo em ambientes operacionais e gerenciais, como salas de controle, reuniões de rotina e painéis de gestão à vista. Nesse modo, a navegação é simplificada e elementos de interface menos relevantes para análise imediata são reduzidos, priorizando a área útil dos gráficos, a legibilidade dos indicadores

e a comparação rápida entre métricas. Essa funcionalidade exibe todos os *dashboards* da aplicação por uma janela de tempo predefinida, conforme ilustrado na Figura 28.

Figura 28 – Tempo de exibição de cada *dashboard*



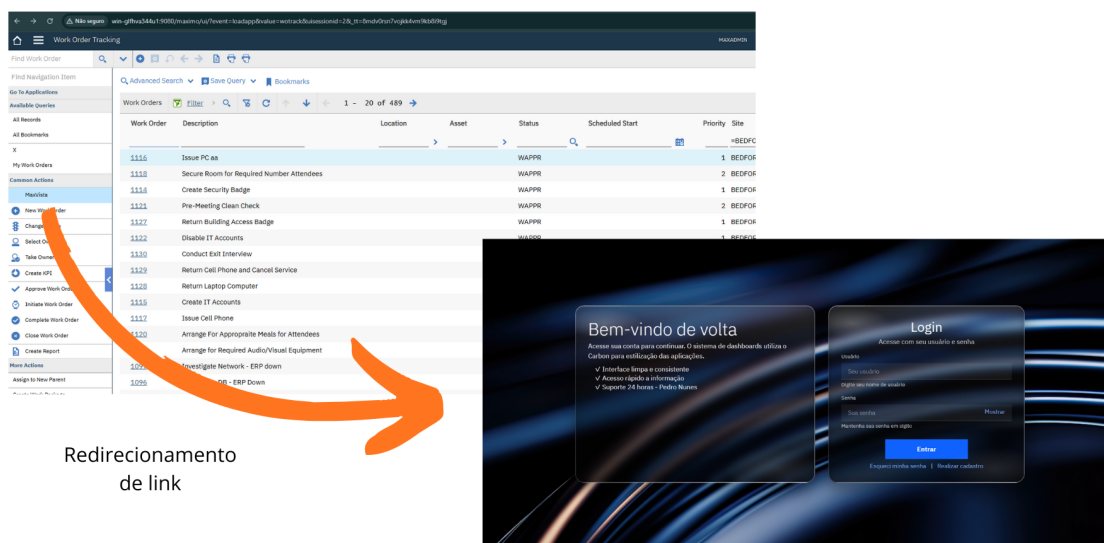
Fonte: elaborado pelo autor (2026).

4.2.6 Integração com o Maximo

Também foi implementada a abertura do MaxVista a partir do IBM Maximo, por meio de um recurso de integração incorporado à própria interface do EAM. Essa funcionalidade pode ser aplicada em diferentes ambientes Maximo e consiste na inclusão de um botão em uma aplicação escolhida, o qual aciona uma ação customizada associada a um *automation script*. Dessa forma, o usuário consegue acessar o MaxVista diretamente a partir do Maximo, sem depender de navegação externa, conforme ilustrado na Figura 29.

Além disso, a solução pode ser estendida para suportar *login* automático no MaxVista, aproveitando informações de autenticação disponíveis no Maximo. Nesse cenário, os dados necessários ao acesso são repassados de forma controlada a partir do contexto do usuário logado, reduzindo atrito operacional e evitando que o usuário precise inserir novamente suas credenciais.

Figura 29 – Ação de acesso ao MaxVista



Redirecionamento
de link

Fonte: elaborado pelo autor (2026).

4.3 Resultados técnicos: conectividade, segurança e separação de responsabilidades

Em termos técnicos, os resultados demonstram:

- Separção de responsabilidades (*backend* × *frontend*): os dados são preparados no *backend* e transmitidos em JSON para renderização no *frontend*, reduzindo acoplamento entre consulta e visualização e favorecendo evolução incremental de *dashboards*.
- Persistência interna desacoplada do Maximo: o MaxVista mantém banco interno (usuários, grupos e conexões), preservando os dados operacionais do Maximo sem alteração, o que favorece governança e reduz risco de interferência no EAM.
- Proteção de credenciais: as credenciais de conexão são armazenadas de forma protegida por criptografia simétrica (Fernet), e as senhas de usuários são persistidas com *hash*. O resultado técnico é a redução de risco de exposição direta de segredos em texto puro, embora permaneçam recomendações para endurecimento de gestão de segredos em produção (por exemplo, rotação de chaves e armazenamento seguro).

4.4 Testes de *performance* e impacto no banco de dados do Maximo

Além da validação funcional e dos aspectos de segurança, foi realizada uma verificação operacional com foco em desempenho e, principalmente, no impacto do MaxVista sobre o banco de dados do Maximo. A motivação deste teste é prática: como os *dashboards* são alimentados por consultas ao DB2, é necessário indicar que a execução dessas extrações, sob concorrência, não induz comportamento indesejado no banco do cliente, como contenção por *locks* e degradação por espera excessiva.

4.4.1 Procedimento de teste e métricas coletadas

Os testes foram executados em ambiente controlado utilizando o *Apache JMeter*, disparando um conjunto fixo de consultas em SQL com base nos *dashboards* do MaxVista, todas direcionadas a objetos do *schema* MAXIMO. Os resultados foram coletados em duas camadas:

- a) Executor — JMeter: registro do comportamento de execução (tempo por amostra, quantidade de requisições e *status* das execuções), permitindo observar estabilidade sob concorrência.
- b) Banco — DB2: coleta de *snapshots* antes e após cada execução por meio do monitor do DB2, utilizando as funções `MON_GET_DATABASE` (contadores globais) e `MON_GET_PKG_CACHE_STMT` (evidência de concentração de custo por *statement*). Nesta análise, os valores de CPU e espera são interpretados como contadores do monitor; portanto, a discussão é fundamentada em deltas e proporções (tendência e relação entre leitura/retorno, CPU/espera) e em *locks*.

4.4.2 Teste de execução de referência

Este teste teve como finalidade validar o conjunto de consultas e observar o comportamento do DB2 sob baixa concorrência, com os seguintes parâmetros:

- *Threads*/usuários simultâneos: 5;
- *Ramp-up*: 30 segundos;
- Execução curta, com duas execuções de cada caso.

A Tabela 3 apresenta os contadores globais do DB2 coletados antes e após o teste.

Tabela 3 – Contadores globais do DB2 — Teste 1 (antes/depois) e variação.

Métrica	Antes	Depois	Varição (Δ)
RQSTS_COMPLETED_TOTAL	20.544	50.864	+30.320
ROWS_READ	768.314	96.998.965	+96.230.651
ROWS_RETURNED	92.037	948.088	+856.051
TOTAL_CPU_TIME	1.875.000	36.156.250	+34.281.250
TOTAL_WAIT_TIME	1.271	1.458	+187
LOCK_WAIT_TIME	0	0	0

Como resultado, observa-se que não houve contenção por *locks*, evidenciada pela variação nula em LOCK_WAIT_TIME, sugerindo que o *workload* de leitura não induziu bloqueios relevantes no banco. Por outro lado, a relação entre leitura e retorno indica custo associado a varredura/agregação: no período, o banco leu 96.230.651 linhas para retornar 856.051 linhas, resultando em aproximadamente 112 linhas lidas por linha retornada. Esse comportamento é compatível com consultas analíticas e reforça a necessidade de observar impacto sob concorrência mais elevada.

4.4.3 Teste de carga sustentada com alta concorrência

O segundo teste foi conduzido para representar um cenário mais exigente, com concorrência sustentada:

- *Threads*/usuários simultâneos: 20;
- *Ramp-up*: 30 segundos;
- Duração nominal: 5 minutos.

A Tabela 4 apresenta os contadores globais do DB2 coletados antes e após o teste.

Tabela 4 – Contadores globais do DB2 — Teste de carga sustentada com alta concorrência (antes/depois) e variação.

Métrica	Antes	Depois	Varição (Δ)
RQSTS_COMPLETED_TOTAL	50.883	2.821.576	+2.770.693
ROWS_READ	96.998.973	1.834.438.641	+1.737.439.668
ROWS_RETURNED	948.170	14.855.733	+13.907.563
TOTAL_CPU_TIME	36.156.250	2.010.859.375	+1.974.703.125
TOTAL_WAIT_TIME	1.471	8.667.999	+8.666.528
LOCK_WAIT_TIME	0	0	0

No cenário de maior concorrência, novamente não foi observada contenção por *locks*, o que reforça que o MaxVista, ao operar com consultas predominantemente de leitura, não tende a onerar o DB2 por bloqueios transacionais. Em contrapartida, o custo do período concentrou-se em processamento/leitura: o banco leu 1.737.439.668 linhas e

retornou 13.907.563 linhas, resultando em aproximadamente 125 linhas lidas por linha retornada. Assim, ainda que o banco suporte o *workload* sem bloqueios, o fator de maior relevância para escalabilidade é a amplificação de leitura, a qual pode aumentar consumo de CPU e I/O conforme cresce o volume de dados e a concorrência do ambiente produtivo.

4.4.4 Síntese e implicações para uso em cenários com maior volume de dados

Os testes indicam dois pontos centrais. Primeiro, sob as condições avaliadas, a execução das consultas do MaxVista não gerou evidência de oneração por *locks*, mesmo no cenário de maior concorrência, o que é desejável em aplicações de *dashboards*, por reduzir o risco de contenção transacional no banco do Maximo. Segundo, o impacto observado está predominantemente associado a leitura e agregação, evidenciado pela razão ROWS_READ/ROWS_RETURNED elevada em ambos os cenários (aproximadamente 112:1 no teste leve e 125:1 no teste sustentado), o que caracteriza amplificação de leitura típica de consultas analíticas e pode representar maior consumo de CPU e I/O conforme o volume de dados cresce.

Cabe ressaltar que, em operação, essa carga tende a ser distribuída entre instâncias distintas, uma vez que cada cliente associa o MaxVista ao seu próprio banco de dados, diferindo do ambiente de teste aplicado. Ainda assim, em cenários com bases maiores ou com concorrência adicional proveniente do próprio Maximo, os resultados sugerem que a evolução do MaxVista deve priorizar refinamentos que reduzam varreduras, como adoção de filtros temporais mais seletivos, revisão de consultas com alta leitura e, quando aplicável, estratégias operacionais, por exemplo janelas de atualização e limitação de concorrência. Em síntese, a solução apresentou métricas consistentes para o contexto avaliado e não indicou degradação relevante do banco a ponto de comprometer a aplicação, mantendo como principal ponto de atenção a eficiência das consultas em cenários de escala.

4.5 Considerações finais do capítulo

Os resultados apresentados evidenciaram a viabilidade funcional do MaxVista por meio de painéis temáticos e indicadores alimentados por consultas ao banco do Maximo, bem como indicaram aspectos técnicos relacionados a conectividade, controle de acesso, proteção de credenciais e separação de responsabilidades. A avaliação operacional indicou que, nas condições analisadas, o comportamento do sistema é compatível com uso voltado a *dashboards* e que os principais pontos de atenção para evolução concentram-se na eficiência das consultas e em estratégias de escalabilidade em cenários com grandes históricos e maior concorrência. A partir dessas evidências, o capítulo final sintetiza as contribuições do trabalho, discute limitações e consolida direções de evolução para fortalecer a solução em ambientes produtivos.

5 Conclusão

A partir desse cenário, conclui-se que a geração de valor analítico de maneira contínua a partir dos dados do MAS Manage ainda representa um desafio recorrente em ambientes industriais, especialmente quando as organizações dependem de relatórios estáticos ou de iniciativas tradicionais de BI que demandam alto esforço de modelagem, infraestrutura e implantação. Nesse contexto, o MaxVista oferece uma proposta coerente para reduzir fricção e acelerar o processo de transformação de registros do Maximo em indicadores gerenciais, ao concentrar em uma única aplicação um controle apartado, com visualização e implantação simplificadas em novas organizações.

Do ponto de vista técnico, a arquitetura adotada está alinhada às necessidades práticas de operação, ao separar claramente a camada de consulta e preparação dos dados da camada de renderização no *frontend*. A implementação contempla requisitos funcionais essenciais, como autenticação, recuperação de senha, controle de acesso e rotas de *dashboards*, e inclui requisitos não funcionais relevantes, como criptografia de credenciais, estrutura modular e uso de componentes de interface consistentes. Em paralelo às abordagens tradicionais, o MaxVista se posiciona de modo complementar: o BIRT permanece adequado para relatórios embutidos e saídas formatadas, enquanto o IBM Cognos Analytics se destaca em governança corporativa e distribuição ampla de dados; o MaxVista, por sua vez, prioriza rapidez de adoção, padronização e entrega ágil de painéis operacionais orientados à manutenção e à gestão de ativos, além de possibilitar customização eficiente conforme necessidades de cada organização.

Ainda assim, a análise evidencia que a solução, na maneira atual, apresenta características típicas de protótipo que precisam ser fortalecidas para utilização em produção, principalmente no que se refere à segurança operacional, gestão de segredos, auditabilidade e estratégia de *deploy*. Para evoluir com robustez em ambientes industriais, é importante endurecer controles de acesso e trilhas de auditoria, padronizar práticas de configuração por ambiente, ampliar testes automatizados e otimizar consultas em cenários com grandes históricos, garantindo previsibilidade de desempenho. Também é relevante amadurecer a camada semântica e o conjunto de indicadores, de modo a reduzir divergências de interpretação entre unidades e aumentar a comparabilidade de resultados.

Quanto aos objetivos propostos pelo trabalho: (i) o contexto de EAM e a relevância do IBM Maximo foram caracterizados por meio da revisão bibliográfica e documental, incluindo normas e evidências públicas de valor; (ii) a arquitetura e o fluxo de dados do MaxVista foram descritos com base na separação entre *backend* e *frontend*, na persistência interna e no mecanismo de conexão/consulta; (iii) foram implementados e demonstrados

painéis e indicadores representativos para manutenção, almoxarifado e visão gerencial, sustentando a verificação funcional; (iv) aspectos técnicos essenciais foram avaliados, com discussão sobre conectividade, controle de acesso e proteção de credenciais, além de recomendações para evolução a ambiente produtivo; e (v) foi realizado o paralelo crítico com BIRT e IBM Cognos Analytics, destacando diferenças de finalidade, esforço de implantação, governança e cenários de adoção.

Em suma, é possível afirmar que o MaxVista é uma alternativa promissora para encurtar o tempo entre dado transacional e decisão, ofertando uma via direta para disponibilização de *dashboards* no ecossistema Maximo, com menor custo de entrada e maior agilidade de evolução. A adoção de uma arquitetura simples e modular faz com que a ferramenta se posicione como um acelerador de valor para cenários industriais nos quais velocidade de entrega, padronização de indicadores, confiabilidade e baixa complexidade de implantação são pontos determinantes.

5.1 Trabalhos futuros

Com base nos resultados obtidos e nas limitações observadas na implementação e validação do MaxVista, indicam-se como direções para trabalhos futuros evoluções técnicas e metodológicas voltadas à maturação da solução para uso em ambiente produtivo, bem como ao aprofundamento de sua avaliação em contexto organizacional.

- **Segurança, identidade e governança:** Ampliar os mecanismos de segurança e governança, incluindo a adoção de autenticação corporativa com SSO por protocolos como SAML ou OIDC, quando aplicável, e o refinamento do controle de acesso baseado em papéis (RBAC), com granularidade por painel, por *site*, por grupo e por escopo de dados. Adicionalmente, é relevante fortalecer a gestão de segredos em produção por meio de cofres de segredos, rotação de chaves e separação de chaves por ambiente, reduzindo riscos de exposição de credenciais. Como linha complementar, recomenda-se implementar trilhas de auditoria, registrando eventos de autenticação, acessos, execuções de consultas, alterações administrativas e exportações, com políticas de retenção e mecanismos de consulta.
- **Hospedagem acoplada ao OpenShift:** Adaptar o MaxVista para implantação em ambiente OpenShift, de modo a viabilizar hospedagem padronizada da aplicação em cenários corporativos e, quando necessário, em ambientes com restrições de conectividade. Essa abordagem permitiria operacionalizar o MaxVista por meio de *containers*, com políticas de escalabilidade, resiliência e gerenciamento centralizado de configuração. Adicionalmente, pode-se implementar um serviço dedicado de extração e preparação de dados (*ETL/ELT*), responsável por consultar o banco do Maximo

em janelas definidas, aplicar tratamentos e agregações e disponibilizar os resultados ao *frontend* do MaxVista. Com isso, parte do processamento deixaria de ocorrer sob demanda, reduzindo a carga no banco transacional e melhorando a previsibilidade de desempenho. Em cenários *offline* ou de conectividade intermitente, essa estratégia favorece a manutenção de um repositório local de dados tratados, permitindo a continuidade do consumo de *dashboards* conforme a política de atualização adotada.

- **Desempenho, escalabilidade e otimização de consultas:** Para suportar cenários com grandes históricos e maior volume transacional, adotar estratégias de otimização de consultas em SQL, considerando planos de execução, índices e filtros por *site* e período, além de paginação e limitação de janelas temporais. Sugere-se avaliar mecanismos de *cache* e pré-agregação de indicadores recorrentes, com invalidação controlada, de modo a reduzir carga no banco de dados do Maximo. Adicionalmente, testes de carga e definição de métricas operacionais podem contribuir para estabelecer limites de uso, previsibilidade de desempenho e requisitos mínimos de infraestrutura.
- **Integração com o ecossistema Maximo:** Aprofundar a integração do MaxVista com o IBM Maximo por meio de mecanismos incorporados à interface do EAM, como ações e *automation scripts*, contemplando controles adicionais de segurança (por exemplo, parâmetros assinados e regras de redirecionamento). Também é pertinente investigar alternativas de integração além do acesso direto ao banco, quando requerido por governança, como consumo de APIs e serviços disponíveis ou replicação controlada para repositórios analíticos. Em cenários de MAS, pode-se explorar padronizações de implantação em *containers*, com *templates* e guias de *deploy* para adoção em múltiplas organizações.

Essas sugestões representam evoluções de grande impacto e ampliam as possibilidades de uso do MaxVista, fortalecendo-o como uma solução de *dashboards* para uso junto ao IBM Maximo.

Referências

- British Standards Institution. *BS EN 15341:2019 Maintenance — Maintenance Key Performance Indicators*. 2019. Acesso em: 19 jan. 2026. Disponível em: <<https://knowledge.bsigroup.com/products/maintenance-maintenance-key-performance-indicators>>. Citado 2 vezes nas páginas 16 e 18.
- Carbon Charts. *Carbon Charts*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://charts.carbondesignsystem.com/>>. Citado 2 vezes nas páginas 38 e 43.
- Carbon Design System. *Carbon Design System*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://carbondesignsystem.com/>>. Citado na página 43.
- Eclipse Foundation. *Eclipse BIRT (Business Intelligence and Reporting Tools)*. 2013. Acesso em: 19 jan. 2026. Disponível em: <<https://projects.eclipse.org/projects/technology.birt>>. Citado na página 33.
- Flask-Login. *Flask-Login Documentation*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://flask-login.readthedocs.io/>>. Citado 2 vezes nas páginas 38 e 43.
- Flask-SQLAlchemy. *Flask-SQLAlchemy Documentation*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://flask-sqlalchemy.readthedocs.io/en/stable/>>. Citado na página 43.
- Forrester Consulting. *The Total Economic Impact™ of IBM Cognos Analytics with Watson*. [S.l.], 2022. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/downloads/cas/3KXKQ0QK>>. Citado 2 vezes nas páginas 17 e 18.
- Grand View Research. *Enterprise Asset Management Market Size, Share & Trends Analysis Report*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.grandviewresearch.com/industry-analysis/enterprise-asset-management-eam-market-report>>. Citado 2 vezes nas páginas 32 e 33.
- HEVNER, A. R. et al. Design science in information systems research. *MIS Quarterly*, v. 28, n. 1, p. 75–105, 2004. Citado na página 18.
- IBM. *New in BIRT reporting (Maximo Asset Management 7.6.1)*. 2018. Acesso em: 19 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/mam/7.6.1?topic=intelligence-birt-reporting>>. Citado na página 34.
- IBM. *IBM Cognos Analytics Version 11.1: Administration and Security Guide*. 2019. Acesso em: 19 jan. 2026. Disponível em: <https://www.ibm.com/docs/en/SSEP7J_11.1.0/com.ibm.swg.ba.cognos.ug_cra.doc/ug_cra.pdf>. Citado 2 vezes nas páginas 34 e 35.
- IBM. *Best Practices for System Performance: Maximo 7.6.x Versions*. 2020. IBM TechXchange Community. Acesso em: 28 jan. 2026. Documento com PDF anexo: *Maximo Best Practices for System Performance 7.6.x.pdf*. Disponível em: <<https://community.ibm.com/community/user/viewdocument/best-practices-for-system-performan?CommunityKey=3d7261ae-48f7-481d-b675-a40eb407e0fd&hlmlt=VT&tab=librarydocuments>>. Citado 3 vezes nas páginas 25, 26 e 27.

- IBM. *10 minute Tip - Birt Report Only Server (BROS)*. 2022. Acesso em: 19 jan. 2026. Disponível em: <<https://www.ibm.com/support/pages/10-minute-tip-birt-report-only-server-bros>>. Citado na página 34.
- IBM. *Downloading and installing BIRT Report Designer (Maximo Manage)*. 2023. Acesso em: 19 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/masv-and-l/maximo-manage/cd?topic=environment-downloading-installing-birt-report-designer>>. Citado 2 vezes nas páginas 34 e 35.
- IBM. *IBM Maximo Application Suite: Product Description Guide*. 2023. Acesso em: 19 jan. 2026. Disponível em: <<https://www.pacmug.com/wp-content/uploads/2023/05/IBM-Maximo-Application-Suite-Product-Description-Guide.pdf>>. Citado 2 vezes nas páginas 30 e 32.
- IBM. *Maximo Manage: Modules and applications*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/masv-and-l/maximo-manage/cd?topic=maximo-manage-modules-applications>>. Citado na página 30.
- IBM. *Transport for London: Cutting carbon from the commute*. 2024. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/case-studies/transport-for-london>>. Citado 2 vezes nas páginas 16 e 21.
- IBM. *Configuring database instances for Maximo Manage*. 2025. IBM Documentation. Acesso em: 20 jan. 2026. Citado na página 31.
- IBM. *Data synchronization using the Maximo integration framework*. 2025. IBM Documentation. Acesso em: 20 jan. 2026. Citado na página 32.
- IBM. *Integration extensibility resources*. 2025. IBM Documentation. Acesso em: 20 jan. 2026. Citado na página 32.
- IBM. *Scripting best practices*. 2025. IBM Documentation. Acesso em: 20 jan. 2026. Citado na página 32.
- IBM. *Testing automation scripts*. 2025. IBM Documentation. Acesso em: 20 jan. 2026. Citado na página 32.
- IBM. *Cognos Analytics entitlement for Maximo Application Suite*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/masv-and-l/maximo-manage/cd?topic=analytics-cognos-entitlement-maximo-application-suite>>. Citado na página 17.
- IBM. *Configuring the BIRT report development environment*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/masv-and-l/maximo-manage/cd?topic=development-configuring-birt-report-environment>>. Citado 2 vezes nas páginas 17 e 36.
- IBM. *Connecting to an IBM database server in Python (ibm_db)*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/db2-warehouse?topic=db-connecting-database-server>>. Citado na página 43.
- IBM. *IBM Cognos Analytics*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/products/cognos-analytics>>. Citado 3 vezes nas páginas 17, 34 e 36.

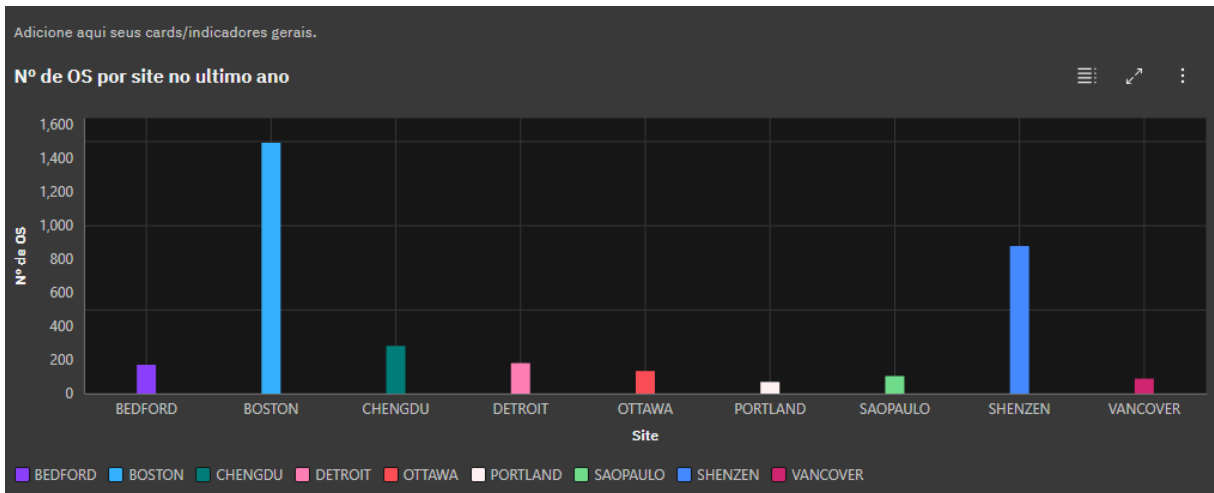
- IBM. *IBM Maximo Manage*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/masv-and-l/cd?topic=applications-maximo-manage>>. Citado 2 vezes nas páginas 16 e 21.
- IBM. *Managing assets*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/docs/en/mhs-and-em/cd?topic=managing-assets>>. Citado 3 vezes nas páginas 16, 20 e 30.
- IBM. *Maximo 7.6.1 and Cognos Analytics*. s.d. Acesso em: 2 jan. 2026. Disponível em: <<https://www.ibm.com/support/pages/bilog-maximo-761-and-cognos-analytics>>. Citado na página 17.
- IBM. *python-ibmdb (ibm_db) repository*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://github.com/ibmdb/python-ibmdb>>. Citado na página 43.
- IDC Research, Inc. *The Business Value of IBM Maximo: Business Value Snapshot*. 2024. Acesso em: 2 jan. 2026. Disponível em: <https://login.structuredweb.com/sw/app/ContentManager/PublicDownloadPDF.aspx?TacticID=M7IPyM9UHuu_hCIHDloH8g2&acctid=79016&csid=0KJKt9YeIeic8AfQbmr0uA2>. Citado 2 vezes nas páginas 16 e 21.
- IFS. *Enterprise Asset Management Software Solutions*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.ifs.com/en/products/alm/eam>>. Citado na página 33.
- Infor. *HxGN EAM (formerly Infor EAM)*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.infor.com/products/eam>>. Citado na página 33.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO 55000:2014 Asset management — Overview, principles and terminology*. Geneva, 2014. Acesso em: 2 jan. 2026. Disponível em: <<https://www.iso.org/standard/55088.html>>. Citado 3 vezes nas páginas 16, 19 e 20.
- International Organization for Standardization. *ISO 55001:2014 Asset management — Management systems — Requirements*. 2014. Acesso em: 19 jan. 2026. Disponível em: <<https://www.iso.org/standard/55089.html>>. Citado 2 vezes nas páginas 16 e 19.
- MDN Web Docs. *Fetch API*. s.d. Acesso em: 12 jan. 2026. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API>. Citado 2 vezes nas páginas 38 e 43.
- Oracle. *Oracle Maintenance (Oracle Fusion Cloud Maintenance)*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.oracle.com/scm/maintenance/>>. Citado na página 33.
- Oracle. *Initializing python-oracledb*. s.d. Acesso em: 12 jan. 2026. Disponível em: <https://python-oracledb.readthedocs.io/en/latest/user_guide/initialization.html>. Citado na página 43.
- Pallets Projects. *Flask Documentation*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://flask.palletsprojects.com/>>. Citado na página 43.
- Pallets Projects. *Flask Documentation: Templates*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://flask.palletsprojects.com/en/stable/templating/>>. Citado na página 43.

- Pallets Projects. *Jinja Documentation*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://jinja.palletsprojects.com/>>. Citado na página 43.
- Pallets Projects. *Modular Applications with Blueprints*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://flask.palletsprojects.com/en/stable/blueprints/>>. Citado 2 vezes nas páginas 42 e 43.
- Pallets Projects. *Werkzeug Documentation: Security Helpers*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://werkzeug.palletsprojects.com/en/stable/utils/>>. Citado na página 43.
- PEFFERS, K. et al. A design science research methodology for information systems research. *Journal of Management Information Systems*, v. 24, n. 3, p. 45–77, 2007. Citado na página 18.
- PR Newswire. *Enterprise Asset Management Market report (press release/summary)*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.prnewswire.com/news-releases/enterprise-asset-management-market-report-2024-forecast-and-trends-301941055.html>>. Citado na página 33.
- PyCA Cryptography. *Fernet (symmetric encryption)*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://cryptography.io/en/latest/fernet/>>. Citado na página 43.
- python-dotenv. *python-dotenv*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://pypi.org/project/python-dotenv/>>. Citado na página 43.
- SAP. *Enterprise Asset Management (EAM) Software Solutions*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.sap.com/products/scm/asset-management-eam.html>>. Citado na página 33.
- SQLAlchemy. *SQLAlchemy Documentation*. s.d. Acesso em: 12 jan. 2026. Disponível em: <<https://docs.sqlalchemy.org/>>. Citado na página 43.
- STENSTRÖM, C. *Performance indicators and terminology for value driven maintenance*. 2013. Acesso em: 19 jan. 2026. Disponível em: <<https://www.diva-portal.org/smash/get/diva2:1822165/FULLTEXT01.pdf>>. Citado 2 vezes nas páginas 16 e 18.
- Technavio. *Enterprise Asset Management Market Analysis (Report summary)*. 2024. Acesso em: 19 jan. 2026. Disponível em: <<https://www.technavio.com/report/enterprise-asset-management-market-industry-analysis>>. Citado na página 32.
- TOKOLA, H. *Dashboard in manufacturing performance management (thesis / report)*. 2016. Acesso em: 19 jan. 2026. Disponível em: <https://www.theseus.fi/bitstream/handle/10024/118965/Tokola_Heikki.pdf>. Citado 2 vezes nas páginas 16 e 18.

Apêndices

APÊNDICE A – Imagens do painel principal

Figura 30 – Painel principal: número de ordens de serviço por site



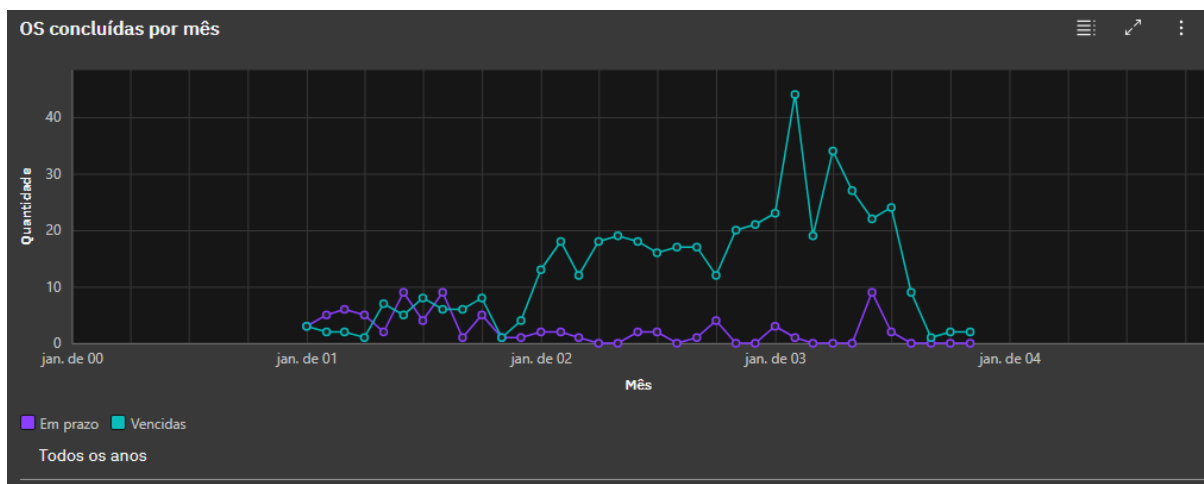
Fonte: elaborado pelo autor (2026).

Figura 31 – Painel principal: ordens concluídas a partir de manutenção preventiva



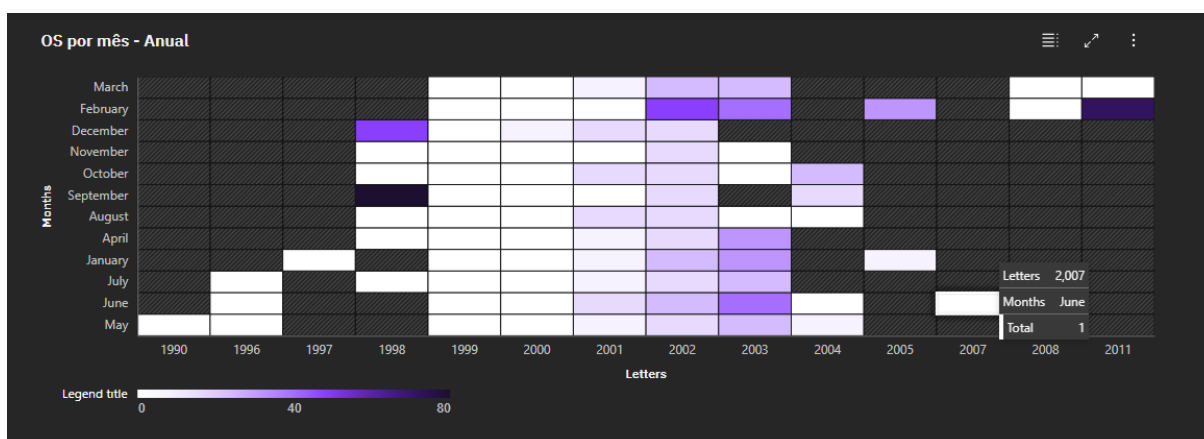
Fonte: elaborado pelo autor (2026).

Figura 32 – Painel principal: ordens de serviço concluídas por mês



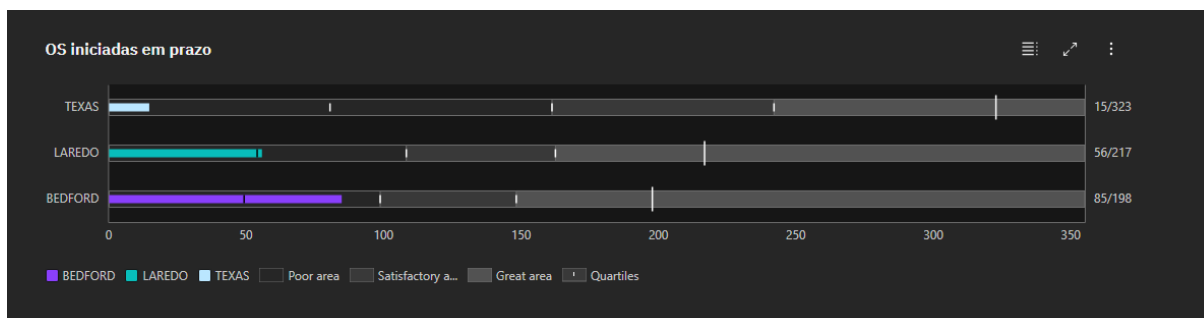
Fonte: elaborado pelo autor (2026).

Figura 33 – Painel principal: sazonalidade de ordens de serviço por ano e mês



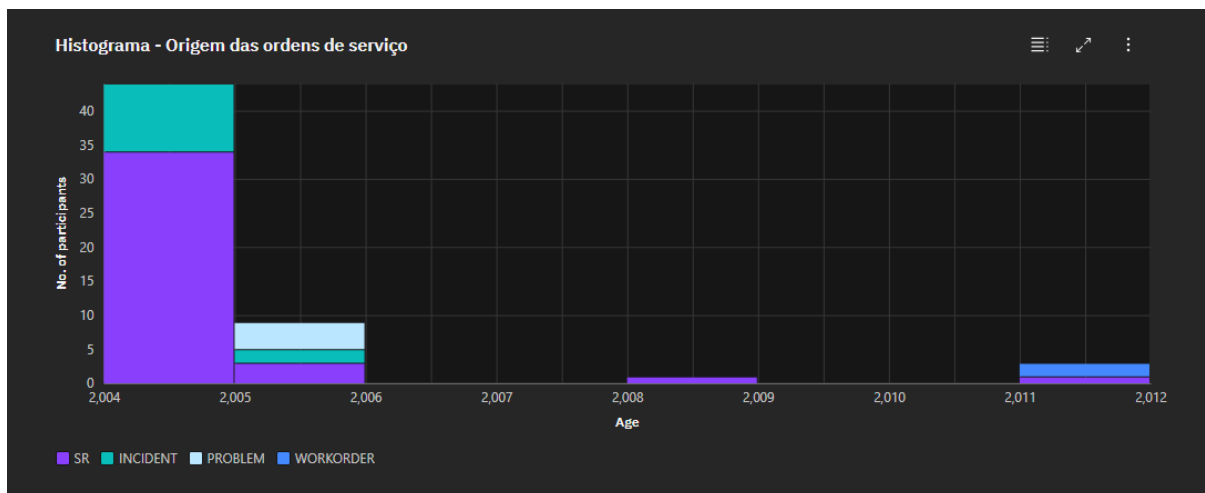
Fonte: elaborado pelo autor (2026).

Figura 34 – Painel principal: ordens concluídas no prazo e vencidas por período



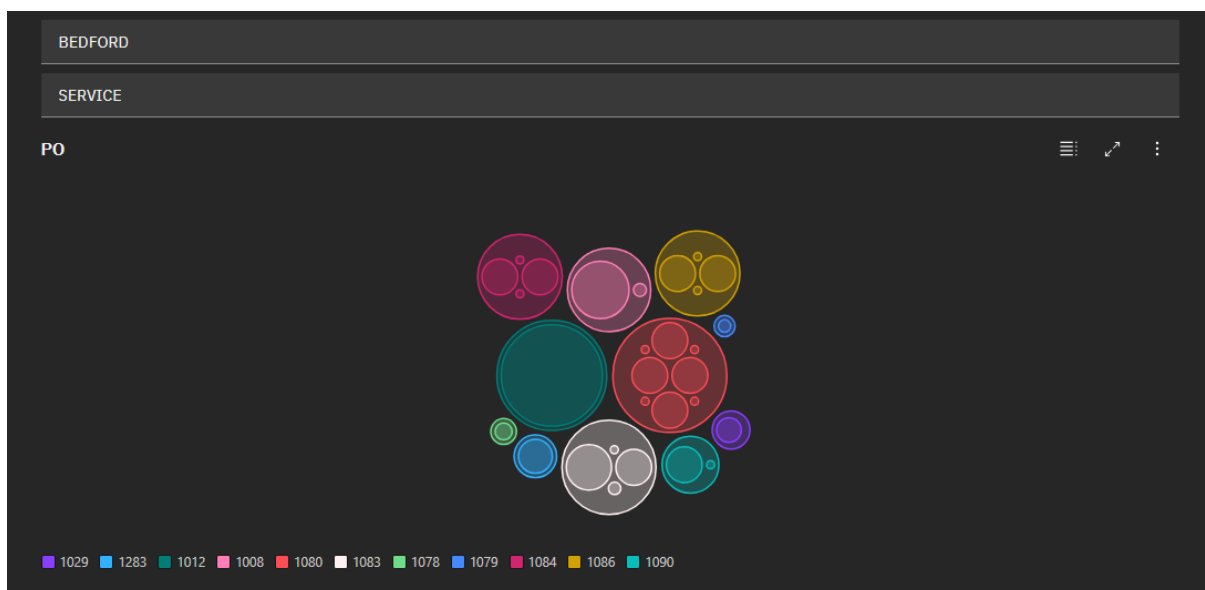
Fonte: elaborado pelo autor (2026).

Figura 35 – Painel principal: origem das ordens de serviço



Fonte: elaborado pelo autor (2026).

Figura 36 – Painel principal: relação por tipo e site



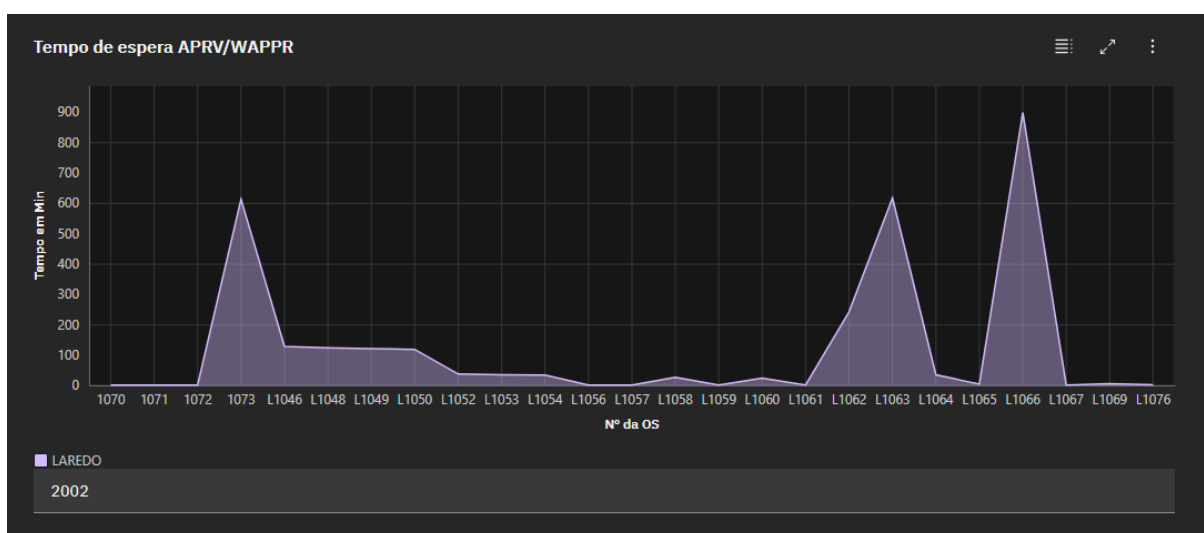
Fonte: elaborado pelo autor (2026).

Figura 37 – Painel principal: distribuição de status das ordens de serviço



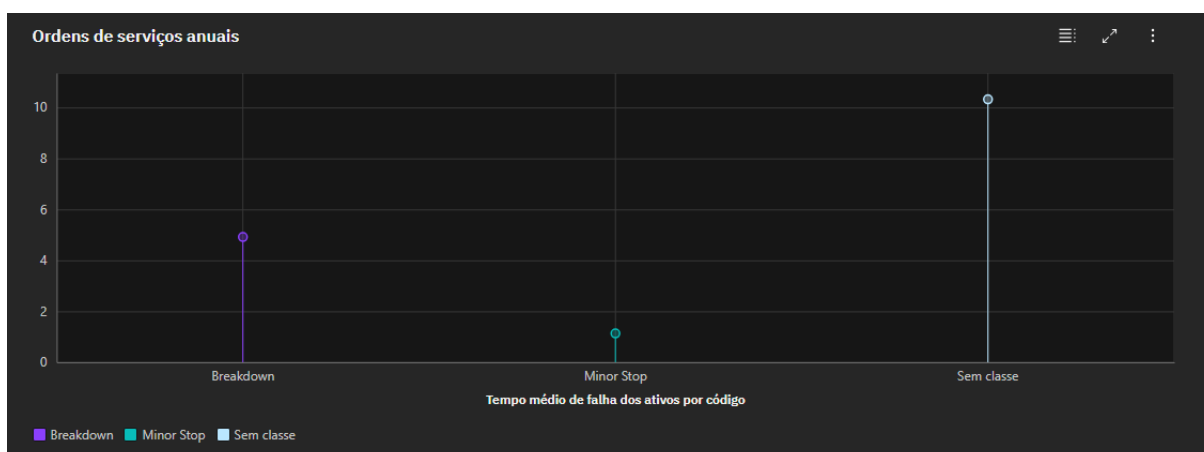
Fonte: elaborado pelo autor (2026).

Figura 38 – Painel principal: tempo de espera em aprovação



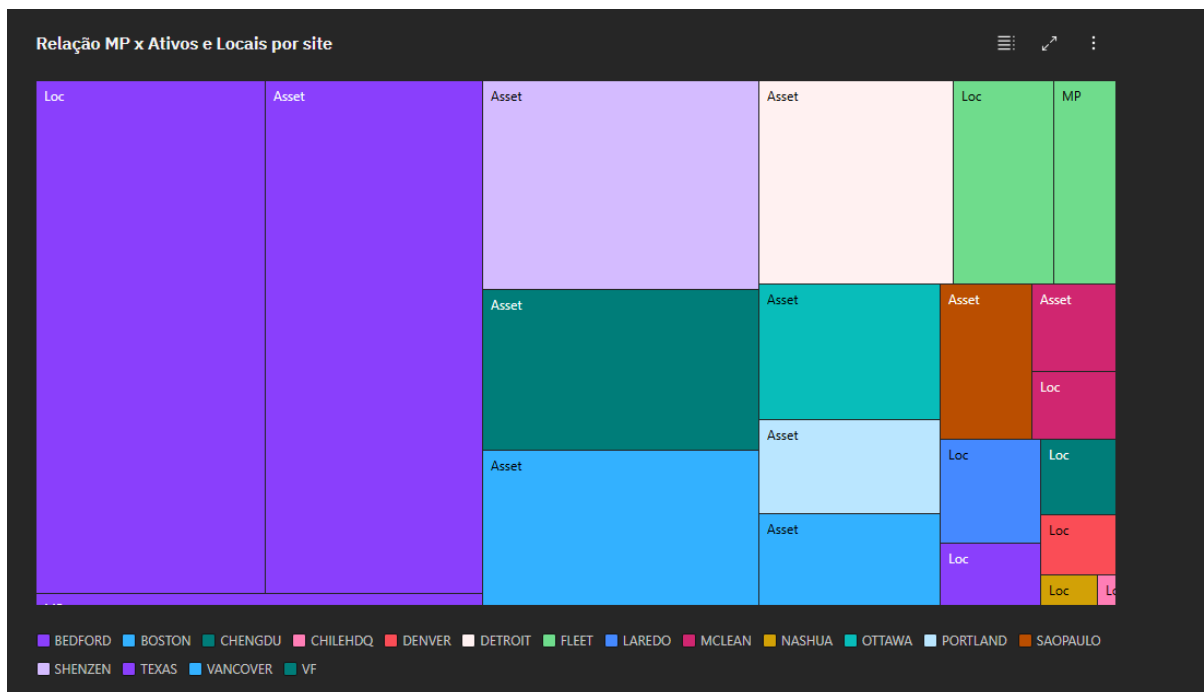
Fonte: elaborado pelo autor (2026).

Figura 39 – Painel principal: tempo médio de falha por ativo



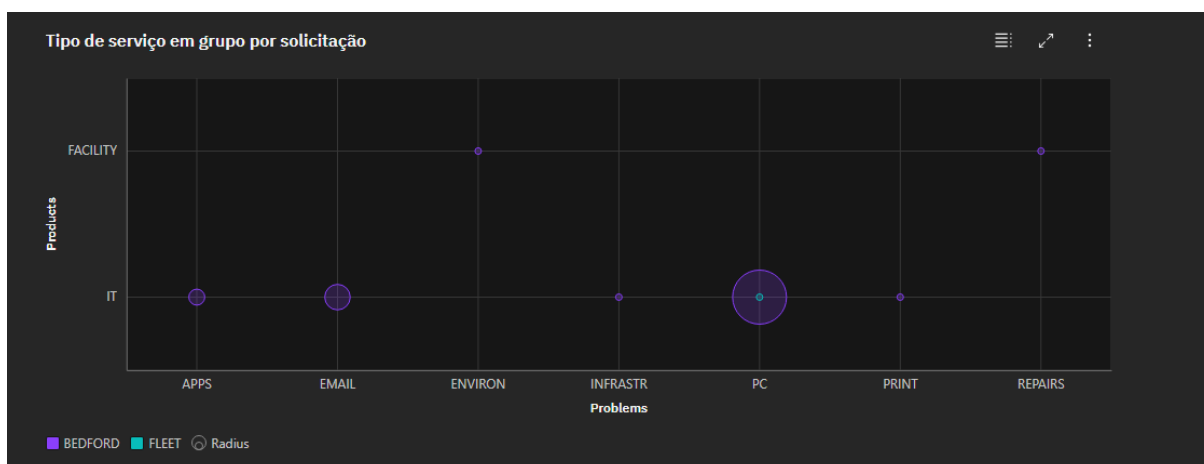
Fonte: elaborado pelo autor (2026).

Figura 40 – Painel principal: composição de ativos e locais por manutenção preventiva



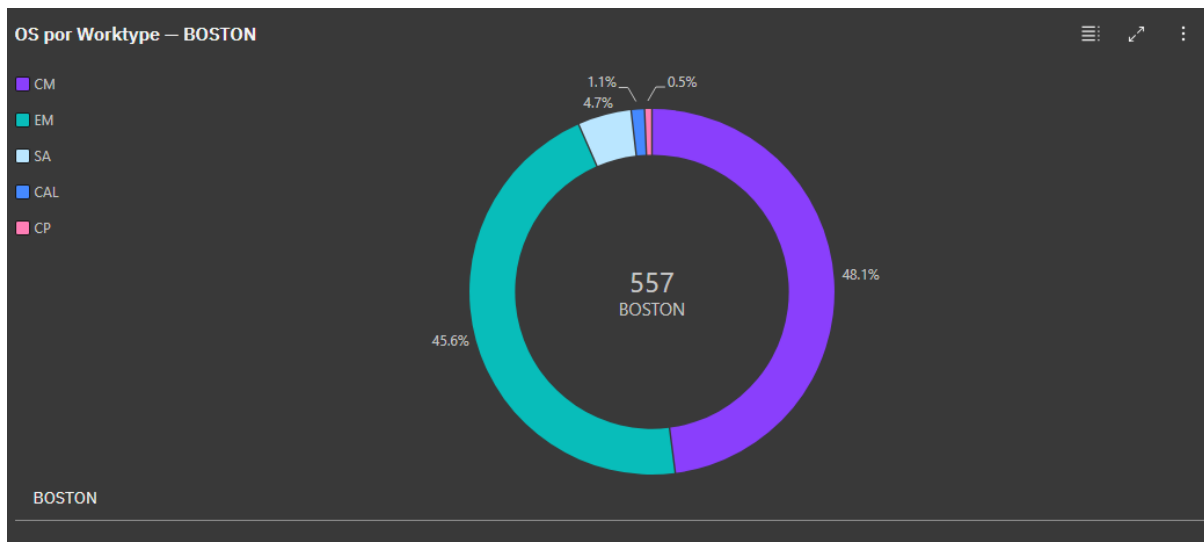
Fonte: elaborado pelo autor (2026).

Figura 41 – Painel principal: tipos de solicitações de serviço



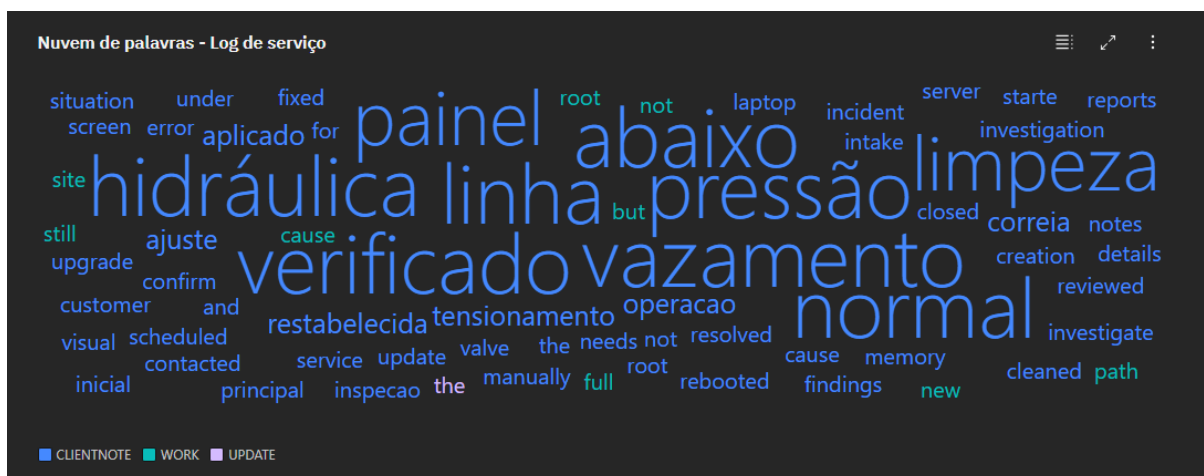
Fonte: elaborado pelo autor (2026).

Figura 42 – Painel principal: tipos de trabalho por site



Fonte: elaborado pelo autor (2026).

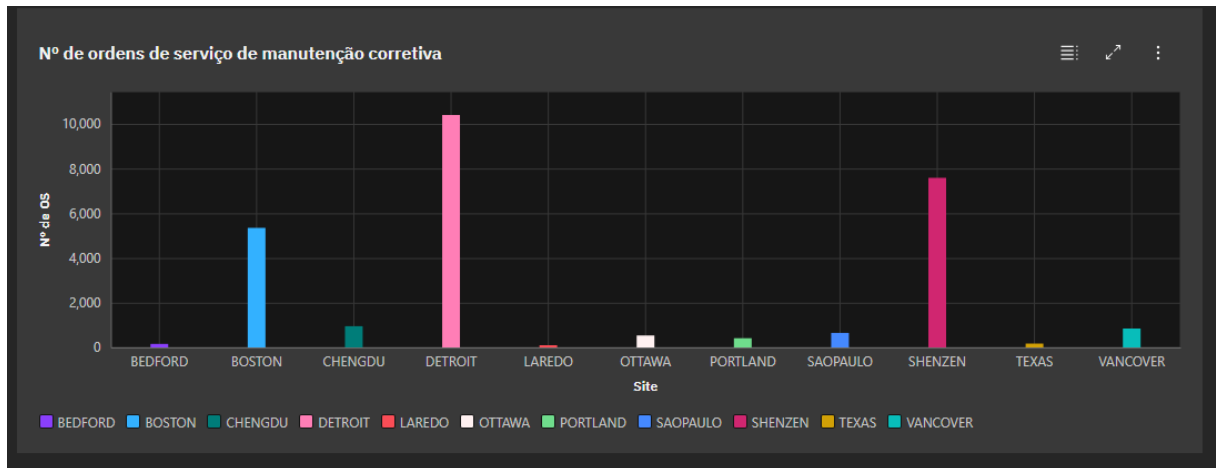
Figura 43 – Painel principal: nuvem de termos associada a logs de serviço das ordens de serviço



Fonte: elaborado pelo autor (2026).

APÊNDICE B – Imagens do painel de Manutenção

Figura 44 – Painel de Manutenção: ordens corretivas por site



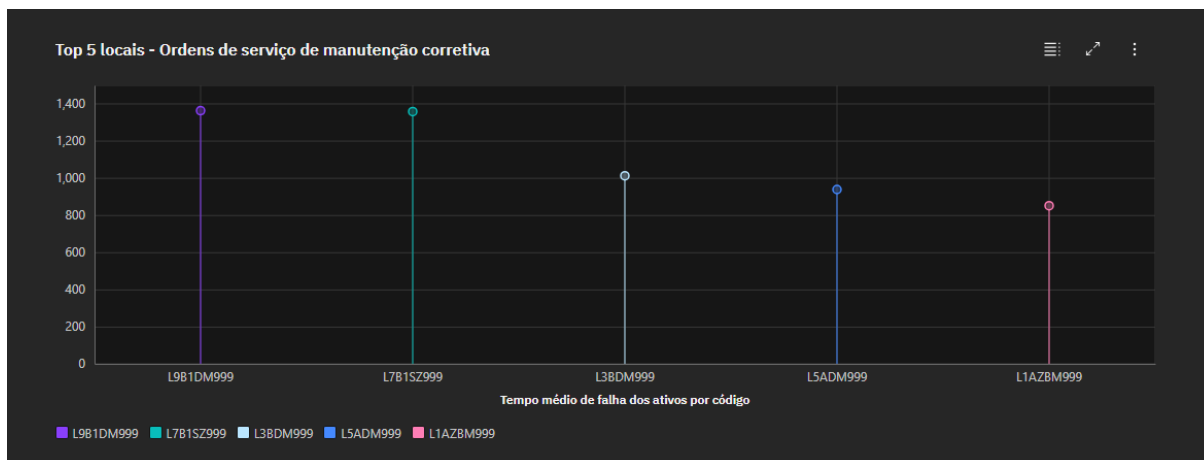
Fonte: elaborado pelo autor (2026).

Figura 45 – Painel de Manutenção: concentração de corretivas por ativo



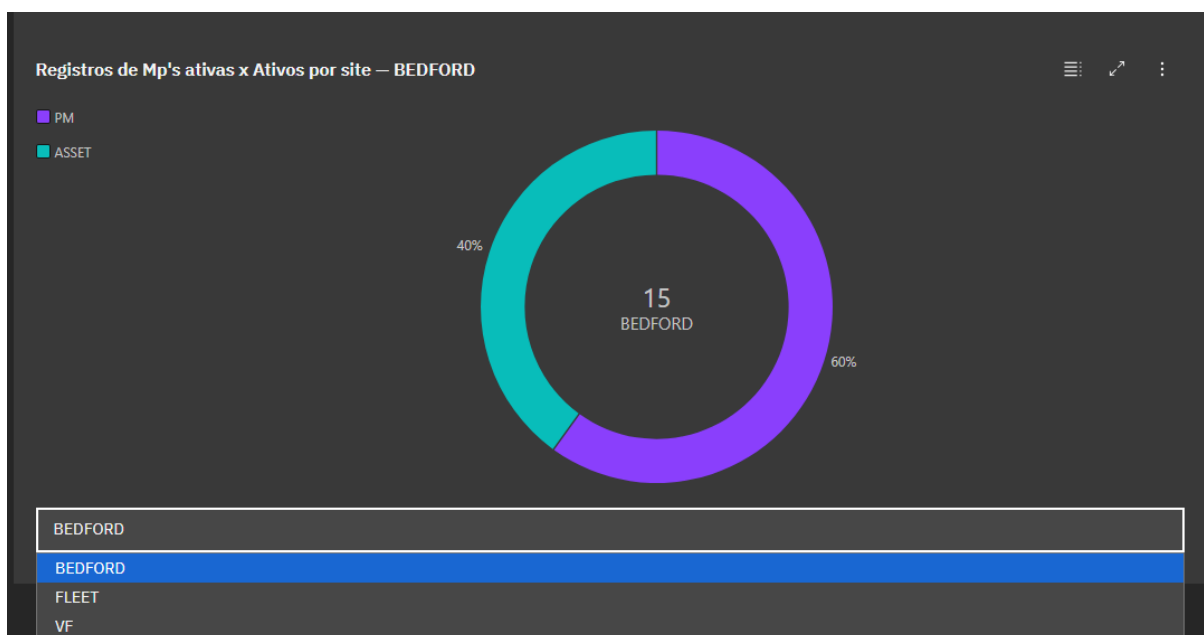
Fonte: elaborado pelo autor (2026).

Figura 46 – Painel de Manutenção: concentração de corretivas por local



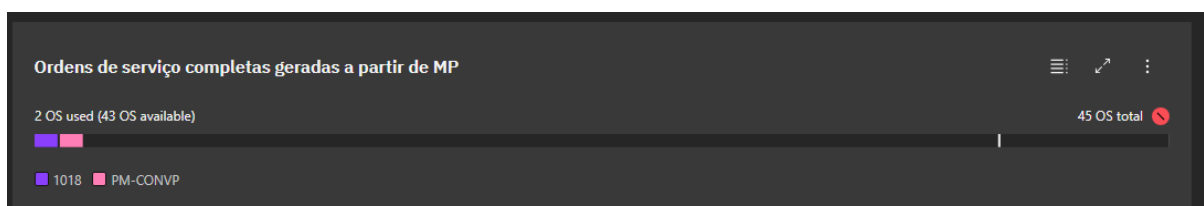
Fonte: elaborado pelo autor (2026).

Figura 47 – Painel de Manutenção: manutenção preventiva ativa por site



Fonte: elaborado pelo autor (2026).

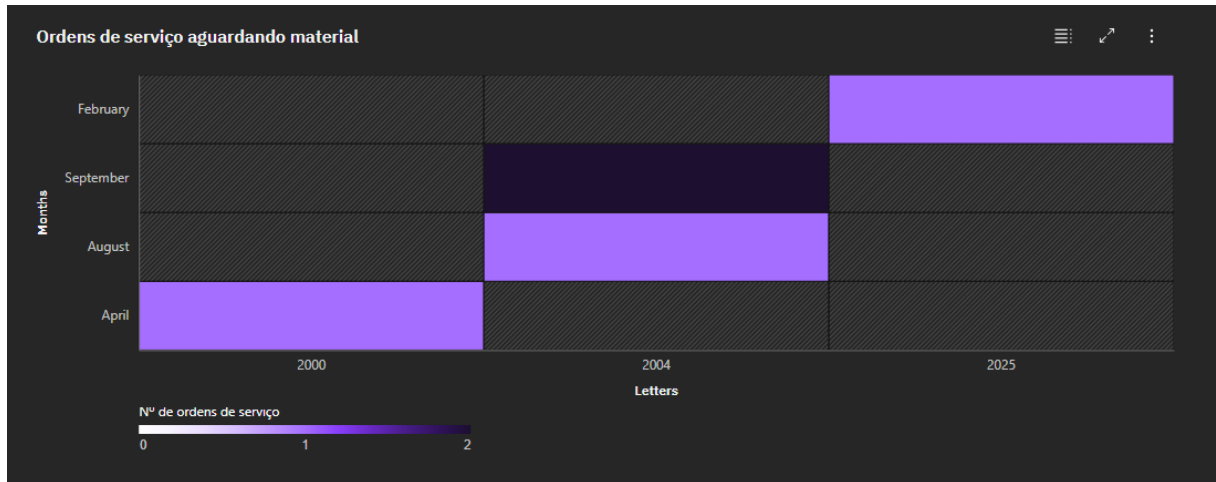
Figura 48 – Painel de Manutenção: ordens associadas a MPs completas



Fonte: elaborado pelo autor (2026).

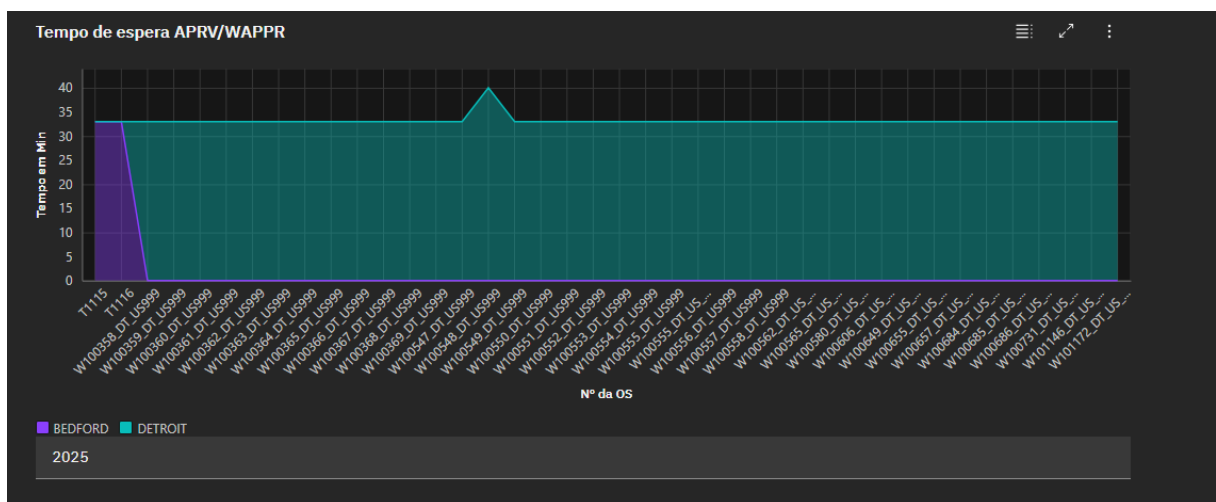
APÊNDICE C – Imagens do painel de Almojarifado

Figura 49 – Painel de Almojarifado: ordens de serviço aguardando material



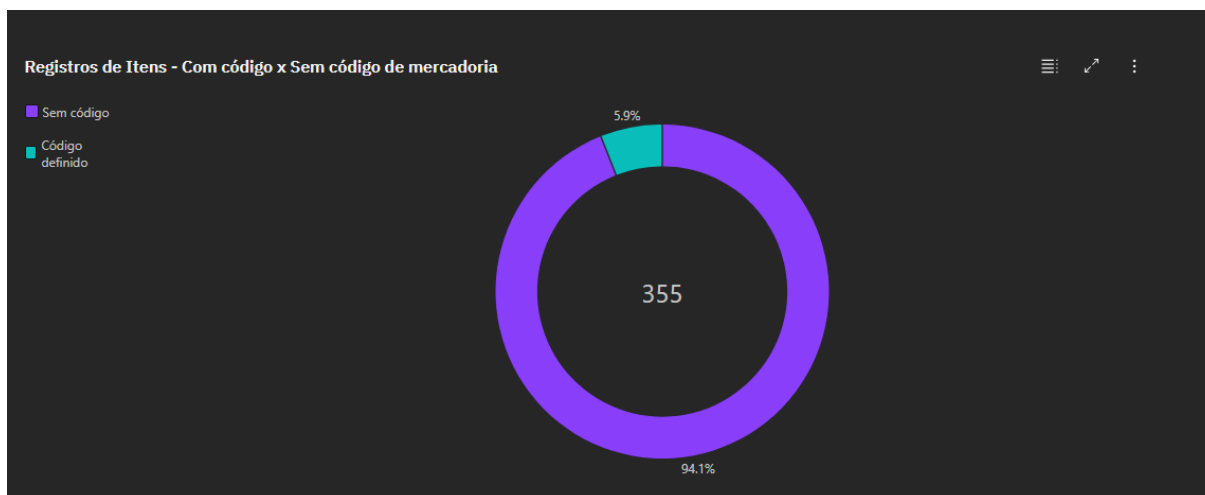
Fonte: elaborado pelo autor (2026).

Figura 50 – Painel de Almojarifado: tempo de espera associado a ordens aguardando material



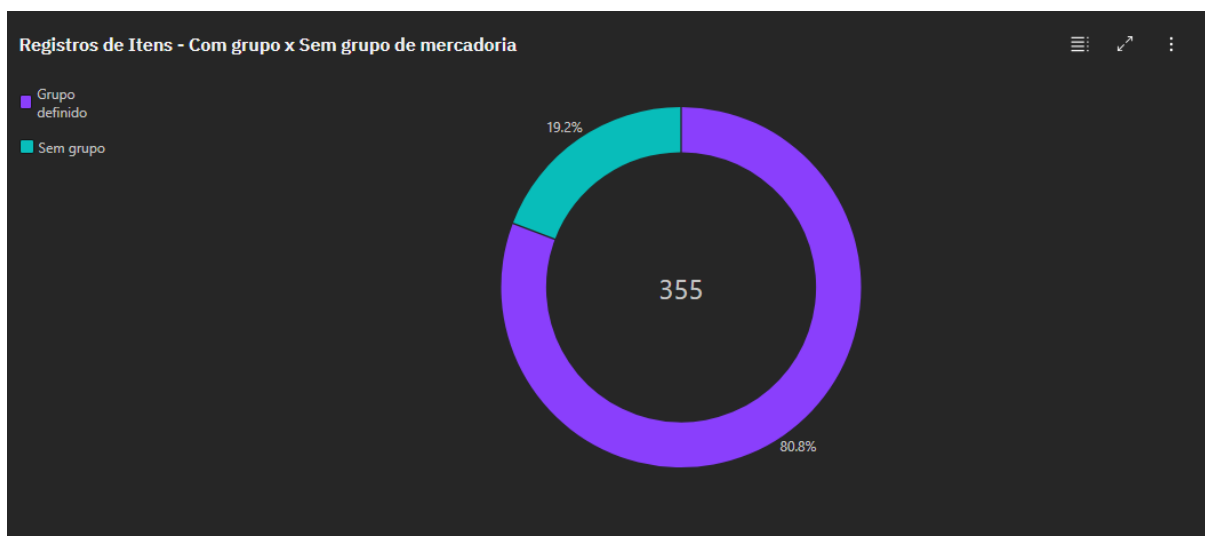
Fonte: elaborado pelo autor (2026).

Figura 51 – Painel de Almoxarifado: completude cadastral por código de mercadoria



Fonte: elaborado pelo autor (2026).

Figura 52 – Painel de Almoxarifado: completude cadastral por grupo de mercadoria



Fonte: elaborado pelo autor (2026).