 

**Especificação de Requisitos e Mapeamento de Dados – Fase 1, Sistema Integrado de Proteção de Dutos (SIPD)**

**T2M Test to Market LTDA**

**Rua Afrânio de Melo Franco, 333 – Quitandinha – 25651-000, Petrópolis, RJ**

[**www.t2mlab.com**](https://www.t2mlab.com/)

**Informações de Controle do Documento**

|  |  |
| --- | --- |
| Nome do Documento | SIPD Fase 1 - Especificação de  Requisitos e Mapeamento de Dados |
| Informações do Projeto | IT-0114-24-016 – DRVCLN Ciência de  Dados para Inteligência Artificial SIPD  Inteligência Operacional |
| Nome da Aplicação | Sistema Integrado de Proteção de Dutos |
| Sigla | SIPD |
| Cliente | Transpetro |
| Autores do Documento | Samuel Pacheco |
| Versão do Documento | 1.0 |
| Status do Documento | Em Produção |

**Histórico de Edição**

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Data | Adições/Modificações | Elaborado/Revisado por |
| 1.0 | 24/02/2025 | Criação do documento | Samuel Pacheco |

**Especificação de Requisitos e Mapeamento de Dados – SIPD Fase 1**

**1. Introdução**

Este documento tem como objetivo detalhar os requisitos funcionais da fase 1 do projeto SIPD, abordando a automação da integração entre o Data Lake e o Power BI, além de fornecer um mapeamento detalhado dos dados armazenados no MongoDB que serão utilizados nos dashboards do Power BI.

**2. Visão Geral da Solução**

A solução proposta tem como finalidade otimizar o fluxo de dados entre o MongoDB, o Data Lake e o Power BI, garantindo uma extração eficiente, armazenamento seguro e disponibilização adequada das informações para análise e tomada de decisão.

O fluxo de dados será estruturado da seguinte forma:

1. **Extração de Dados:**  
   1. O sistema realizará a extração dos dados diretamente do MongoDB por meio de consultas estruturadas, garantindo a obtenção das informações necessárias sem impactar o desempenho da base de produção.
   2. A extração será realizada em intervalos configuráveis, assegurando a atualização contínua dos dados.
2. **Armazenamento no Data Lake:**
   1. Os dados extraídos do MongoDB serão transformados e armazenados no Azure Data Lake, seguindo um modelo de particionamento que facilite a organização e otimização de consultas.
   2. O armazenamento será estruturado por categorias, tais como alertas, rondas e verificações, permitindo um acesso eficiente aos diferentes tipos de informações
3. **Processamento e Transformação:**  
   1. O sistema realizará a padronização e normalização dos dados, garantindo consistência entre os diferentes registros.
   2. Serão aplicadas validações para identificação e tratamento de inconsistências nos dados antes de serem disponibilizados para consumo.
4. **Integração com Power BI:**
   1. O Power BI se conectará diretamente ao Data Lake para consumo dos dados processados.
   2. A disponibilização das informações no Power BI seguirá um modelo otimizado, utilizando tabelas dimensionais e de fatos para garantir a eficiência na análise e visualização.

Com essa abordagem, a solução visa eliminar a necessidade de processos manuais de extração e transformação de dados, proporcionando maior confiabilidade e eficiência na análise operacional e de desempenho do SIPD.

**3. Requisitos Funcionais**

|  |  |  |
| --- | --- | --- |
| **ID** | **Requisito** | **Descrição** |
| RF01 | Extração de Alertas | O sistema deve extrair dados de alertas gerados no SIPD e armazená-los no Data Lake. Essa extração deve ocorrer em intervalos configuráveis e incluir dados de geolocalização, timestamps e classificação de alerta. |
| RF02 | Extração de Rondas | O sistema deve consolidar informações sobre as rondas realizadas, incluindo status, tempo de execução, equipe responsável e rota percorrida. Deve permitir a recuperação de históricos e correlacionar com alertas próximos. |
| RF03 | Extração de Verificações | Dados de verificação devem ser extraídos e consolidados para análise de eficiência das equipes. Deve incluir detalhes como motivo da verificação, status final, ações tomadas e tempo médio para resolução. |
| RF04 | Integração com Power BI | Os dados armazenados no Data Lake devem estar disponíveis para análise no Power BI. A integração deve ser feita via pipelines automatizados, garantindo a atualização periódica dos dados. |
| RF05 | Transformação e Padronização | O sistema deve realizar transformações nos dados extraídos para garantir consistência, como padronização de timestamps, normalização de status e unificação de registros duplicados. |

**4. Mapeamento de Dados (MongoDB x Power BI)**

A tabela abaixo apresenta o mapeamento de onde os dados utilizados no Power BI estão localizados no MongoDB.

|  |  |  |
| --- | --- | --- |
| **Indicador no Power BI** | **Collection no MongoDB** | **Campo(s) Relacionado(s)** |
| Filtro de Sistemas | **CCPD-REGISTRATIONS** | * **Ccpd-registrations.gasducts** * **Ccpd-registratrions.states** |
| Filtro de Período da Patrulha | **CCPD-PATROL** | * **Ccpd-patrol.patrol\_repetion** * **Ccpd-patrol.patrols** * **Ccpd-patrol.patrolteams** |
| Filtro Tipo de alerta | **CCPD-ALERTS** | * Ccpd-alerts.alerts |
| Filtro Status de Alerta | **CCPD-ALERTS** | * Ccpd-alerts.alerts |
| Gráfico de barras: Qtd alertas por sistemas | **CCPD-REGISTRATIONS** | * Ccpd-registrations.opticalProcessingSystem * Ccpd-registrations.gasducts * Ccpd-registrations.inspection\_points |
| Gráfico de barras: Tipos de Alerta (qtd) | **CCPD-REGISTRATIONS** | * Ccpd-registrations.procedures |
| Gráfico de rosca: Status (verificados) | **Ccpd-alerts** | * Ccpd-alerts.alerts |
| Gráfico de barras: KP mais alarmados | **CCPD-EVENT-VERIFICATION** | * Ccpd-event-verification.verifications |
| Gráfice barras: Quantidade de Alertas | **CCPD-ALERTS** | * Ccpd-alerts.alerts |
| Gráfico de barras: Quantidade de Patrulhas por período | **CCPD-PATROLS** | * **Ccpd-patrol.patrol\_repetion** * **Ccpd-patrol.patrols** * **Ccpd-patrol.patrolteams** |
| Card: Tempo Médio de ruído | **CCPD-EVENT-VERIFICATION e CCPD-ALERTS** | * Ccpd-event-verification.events * Ccpd-event-verification.events\_verification |
| Card: Tempo médio por alerta | **CCPD-EVENT-VERIFICATION** | * Ccpd-event-verification.events |
| Card: Qtd de alertas | **CCPD-ALERTS** | * Ccpd-alerts.alerts * Ccpd-alerts.sequence |
| Canais mais alarmados | **CCPD – ALERTS, CCPD-EVENT-VERIFICATION e CCPD-REGISTRATIONS** | * Ccpd-alerts.alerts.event-verification * Ccpd-event-verification.event\_verifications * Ccpd-registrations.channels * Ccpd-registrations.channels * Ccpd-registrations.companies * Ccpd-registrations.companies |
| TACCPD: tempo de atendimento entre o início do alerta e a solicitação para verificação/inspeção PD. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * taccpd = sentDate - eventCreatedAt |
| TACS: tempo entre a criação do registro e a solicitação para verificação/inspeção PD. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * tacs = sentDate - createdAt |
| TAPD: tempo de atendimento para verificação/inspeção PD. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * tapd = dateSendEndVerification - sentDate |
| TTA: tempo total de atendimento. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * tta = dateSendEndVerification - eventCreatedAt |
| TERF: tempo de entrega do relatório final. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * terf = dateSendEndVerification - dateManagementReport |
| TVI: tempo de verificação/inspeção. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * tvi = endDateInspection - startDateInspection |
| TSI: tempo entre solicitação e início da Inspeção. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * tsi = startDateInspection - sentDate |
| TRIVI: tempo de resposta para início da verificação/inspeção. | Database: CCPD-EVENT-VERIFICATIONS  Collection: verifications | * trivi = startDateMovement – sentDate |
| Card: Qtd de verificação | **CCPPD-EVENT-VERIFICATION** | * Ccpd-event-verifications.events * Ccpd-event-verifications.event\_verification * Ccpd-event-verifications.event\_main\_actions |

**Resumo dos locais a serem extraídos os dados**

**Banco de Dados: CCPD-REGISTRATIONS**

* **Ccpd-registrations.gasducts**
* **Ccpd-registrations.states**
* **Ccpd-registrations.opticalProcessingSystem**
* **Ccpd-registrations.inspection\_points**
* **Ccpd-registrations.procedures**
* **Ccpd-registrations.channels**
* **Ccpd-registrations.companies**

### **Banco de Dados: CCPD-PATROL**

* **Ccpd-patrol.patrol\_repetion**
* **Ccpd-patrol.patrols**
* **Ccpd-patrol.patrolteams**

### **Banco de Dados: CCPD-ALERTS**

* **Ccpd-alerts.alerts**
* **Ccpd-alerts.sequence**

### **Banco de Dados: CCPD-EVENT-VERIFICATION**

* **Ccpd-event-verification.verifications**
* **Ccpd-event-verification.events**
* **Ccpd-event-verification.events\_verification**
* **Ccpd-event-verification.event\_verifications**
* **Ccpd-event-verification.event\_main\_actions**

**5. Arquitetura e Fluxo de Dados**

* Os dados são extraídos do MongoDB por meio de consultas estruturadas nas collections listadas acima.
* Após a extração, os dados são armazenados no Azure Data Lake, organizados por data e categoria.
* O Power BI se conecta ao Data Lake para acessar os dados processados e gerar dashboards.

**6. Considerações Finais**

Este documento servirá como referência para o DBA responsável pela extração de dados e para a equipe de BI na definição de relatórios e dashboards. Ajustes adicionais podem ser realizados conforme a necessidade do projeto.