



Data lake para agregação de dados de produção e ferramentas de visualização na indústria de estampagem

Leonardo Leite Meira dos Santos - 54363

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Paulo Alves

Prof. Kécia Marques

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2022-2023



Data lake para agregação de dados de produção e ferramentas de visualização na indústria de estampagem

Leonardo Leite Meira dos Santos - 54363

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Paulo Alves

Prof. Kécia Marques

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2022-2023

Dedicatória

(Facultativo) Dedico este trabalho a ...

Agradecimentos

(Facultativo) Agradeço a ...

Resumo

O resumo (no máximo com 250 palavras), permite a avaliação do interesse de um documento e facilita a sua identificação na pesquisa bibliográfica em bases de dados onde o documento se encontre referenciado.

É recomendável que o resumo aborde, de forma sumária:

- Objetivos principais e tema ou motivações para o trabalho;
- Metodologia usada (quando necessário para a compreensão do relatório);
- Resultados, analisados de um ponto de vista global;
- Conclusões e consequências dos resultados, e ligação aos objetivos do trabalho.

Como este modelo de relatório se dirige a trabalhos cujo foco incide, maioritariamente, no desenvolvimento de software, algumas destas componentes podem ser menos enfatizadas, e acrescentada informação sobre análise, projeto e implementação do trabalho.

O resumo não deve conter referências bibliográficas.

Palavras-chave: termos (no máximo 4), que descrevem o trabalho.

Abstract

Direct translation (maximum of 250 words) to English of the section “Resumo”.

Keywords: direct translation of “Palavras-chave”

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	2
1.3	Estrutura do Documento	2
2	Revisão literaria	5
2.1	Nome do artigo	5
3	Methodology	7
3.1	Definição dos requisitos	7
3.1.1	Requisitos Funcionais	7
3.1.2	Requisitos Não Funcionais	11
3.2	Método de coleta e armazenamento de dados	13
3.3	Processo de desenvolvimento do software	15
3.4	Tecnologias	15
3.5	Gestão das atividades do projeto	15
3.6	Desafios para obtenção dos dados	15
4	Arquitetura do sistema	17
4.1	Diagrama do sistema	17
4.2	Princípios do SOLID	17
4.3	Arquitetura do backend	17

4.4	Arquitetura do frontend	18
4.5	Containers	18
5	Implementação	19
5.0.1	Implementação do banco de dados	19
5.0.2	Implementação da API	19
5.0.3	Interpretação dos Bytes	20
5.0.4	Decodificação da Mensagem	20
5.0.5	Implementação do módulo de processamento de dados	22
5.0.6	Implementação do frontend	22
5.0.7	Adaptando a implementação para outros contextos	22
6	Características do Sistema do ponto de vista funcional	23
6.0.1	Monitoramento em tempo real	23
6.0.2	Sistema de alertas e nnotificações	23
6.0.3	Dados historicos	23
6.0.4	Interface do usuário	23
7	Resultados e Avaliação	24
8	Conclusão e Trabalhos Futuros	25
8.0.1	Resumo	25
8.0.2	Limitações do sistema	25
8.0.3	Sugestões para trabalhos futuros	25
A	Proposta Original do Projeto	A1
B	Outro(s) Apêndice(s)	B1

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

1.1 Enquadramento

No cenário industrial contemporâneo, a busca constante por eficiência e inovação tornou-se um pilar essencial para a competitividade e sustentabilidade financeira das empresas. À medida que a tecnologia avança, a produção industrial sofre uma evolução para se manter competitiva perante ao mercado. Nesse contexto, o monitoramento e a otimização das máquinas em linhas de produção tornam-se cruciais para garantir um funcionamento eficaz e prevenir potenciais paralisações ou falhas operacionais.

No entanto, a tradição das práticas industriais muitas vezes é caracterizada por inspeções manuais e sistemas de monitoramento desatualizados, que não conseguem fornecer informações em tempo real ou análises aprofundadas sobre o desempenho das máquinas. Esse descompasso tecnológico pode resultar em perdas significativas, em termos de produção, de recursos financeiros e manutenção de máquinas.

Além disso, com a crescente integração de sistemas de IoT e a proliferação de sensores avançados, existe uma quantidade imensa de dados sendo gerada continuamente. No entanto, sem a infraestrutura adequada para coletar, armazenar e analisar esses dados, as empresas podem se encontrar sobrecarregadas, incapazes de extrair insights significativos que poderiam informar decisões estratégicas e operacionais.

1.2 Objetivos

Dado o enquadramento anterior, a necessidade identificada é de desenvolver um sistema robusto que possa receber dados de sensores que coletam dados das máquinas em tempo real, armazená-los de maneira eficiente em um data lake e apresentá-los através de um dashboard, transformando a maneira como as empresas monitoram e otimizam suas linhas de produção, garantindo não apenas eficiência, mas também uma abordagem proativa à manutenção e gestão industrial. Esse sistema não apenas forneceria informações em tempo real sobre o status e o desempenho das máquinas, mas também permitiria análises históricas, ajudando gestores e técnicos a identificar tendências e falhas, além de otimizar a produção, minimizando perdas na produção e otimizando os ganhos financeiros.

1.3 Estrutura do Documento

Esta dissertação de mestrado está organizada começando pela Introdução, onde o problema e o escopo são apresentados. Esta seção contextualiza a necessidade de modernização industrial, destacando a importância do monitoramento e otimização de máquinas na indústria. A relevância do estudo é então justificada, com foco na crescente demanda por análise de dados no mercado.

Segue-se com a Revisão Literária, que apresenta uma análise sobre trabalhos e conceitos relacionados ao projeto. Esta seção engloba desde trabalhos semelhantes já realizados, métodos de análise de dados, até discussões sobre visualização de dados e importância da gestão de dados no mercado atual.

A Metodologia discute a escolha de tecnologias específicas usadas no projeto, detalhando o método de armazenamento de dados, e o processo adotado para o desenvolvimento do software. Também são descritas as estratégias de gestão das atividades do projeto e os desafios enfrentados durante a coleta de dados.

No capítulo sobre a Arquitetura do Sistema, é discutido sobre as especificidades técnicas do sistema proposto. Desde o diagrama geral do sistema até a implementação de

princípios de codificação, como o SOLID, esta seção visa elucidar o design e a funcionalidade do software criado.

A seção de Implementação aprofunda-se nos aspectos técnicos do sistema. Aqui, cada componente do banco de dados, da API e do módulo de processamento de dados é detalhadamente descrito. Também se discute a reutilização potencial do sistema em outros contextos, e o que seria necessário para isso.

No que diz respeito às Características do Sistema do ponto de vista funcional, a dissertação tem uma seção que foca em mostrar a aplicação prática do software, utilizando capturas de tela e diagramas para demonstrar as funcionalidades.

Em Resultados e Avaliação, são apresentados os resultados obtidos durante o projeto, destacando as principais realizações e benefícios observados na implementação do sistema proposto. A Conclusão e Trabalhos Futuros resume os pontos chave do projeto, identifica as limitações do sistema atual e sugere possíveis direções para continuidade e implementações futuras.

Capítulo 2

Revisão literaria

In this chapter it is expected to have a generic description of the problem and area: scope, concepts and technology and/or a literature review (state-of-the-art). In case of a practical project, there should also be described the tools and the justification for their use.

Usually, this chapter is divided in multiple sections, to complement the topics.

trabalhos semelhantes já realizados, métodos de análise de dados, até discussões sobre visualização de dados e importância da gestão de dados no mercado atual

2.1 Nome do artigo

Texto...

Capítulo 3

Methodology

3.1 Definição dos requisitos

A definição precisa dos requisitos é fundamental para garantir que o sistema desenvolvido atenda às necessidades e objetivos. Os requisitos desse projeto foram classificados nas categorias funcionais e não funcionais, para garantir uma compreensão completa do que é esperado do sistema.

3.1.1 Requisitos Funcionais

Os requisitos funcionais desempenham um papel fundamental no desenvolvimento de sistemas, definindo as funções que um sistema ou componente de software deve ser capaz de executar. Em essência, eles fornecem uma descrição das interações que o sistema terá com seus usuários ou com outros sistemas, especificando os serviços que o sistema deve fornecer.

Para garantir eficácia, os requisitos funcionais devem ser claramente definidos, sem ambiguidades, e serem mensuráveis, rastreáveis, completos e consistentes. Além disso, eles devem ser definidos levando em consideração as necessidades e objetivos do projeto, garantindo que o sistema desenvolvido seja não apenas tecnicamente sólido, mas também útil e relevante para seus usuários finais.

Dentro do contexto do sistema desenvolvido, está listado abaixo os requisitos funcionais do sistema e sua descrição, de forma a deixar claro, de forma funcional, o que o sistema deve fazer.

O sistema deve permitir que um usuário acesse o sistema de forma segura com um email e senha

Dado que o sistema é para a visualização dos dados de operação de uma indústria de estampagem, as informações disponibilizadas devem ser acessadas apenas por usuários previamente autorizados.

O sistema deve permitir que um usuário veja as suas informações pessoais que são armazenadas no sistema

Cada usuário que tiver acesso ao sistema terá registrado nele alguns de seus dados pessoais, como e-mail, cargo e tipo de acesso. Portanto, cada usuário deve ter acesso a suas informações pessoais que estão salvas no sistema.

O sistema deve exibir em tempo real os valores lidos pelos sensores em cada uma das máquinas

Ao receber os dados enviados pelos sensores, o sistema deve exibir em tela os valores lidos, separado por tipo de sensor e máquina.

O sistema deve armazenar um valor máximo ideal para cada tipo de sensor utilizado

Cada sensor deverá ter um valor máximo ideal para o funcionamento. Ele servirá de parâmetro para entender se o valor lido pelo sensor indica um bom ou mal funcionamento da máquina.

O sistema deve identificar sempre que um valor lido pelo sensor não estiver abaixo do valor ideal

Esse requisito refere-se à capacidade do sistema de detectar, de forma automática, toda vez que o sensor indicar um valor que não esteja abaixo do limite pré definido. Isto é, se o valor ideal for X , e o sensor ler um valor maior ou igual a X , o sistema reconhecerá esta situação.

O sistema deve registrar sempre que um valor lido pelo sensor não estiver de acordo com o valor ideal

Esse requisito implica que o sistema deve manter um registro de todos os momentos em que o valor detectado pelo sensor não estiver abaixo do valor ideal armazenado.

O sistema deve mostrar em tela quando um valor lido pelo sensor não estiver abaixo do ideal

Sempre que o sensor detectar um valor abaixo do padrão ideal, o sistema deverá exibir um alerta na interface de forma que fique sempre visível para o usuário.

O sistema deve mostrar no formato de notificação os registros de não funcionamento abaixo do valor ideal

Este requisito estabelece que o sistema deve apresentar aos usuários na forma de notificações quando o sensor ler um valor acima do ideal, para possibilitar que os usuários sejam informados, mesmo que posteriormente, sempre que um alerta for identificado.

O sistema deve permitir que o usuário marque uma notificação como lida, de forma que ela não apareça novamente

Após ser notificado, os usuários deverão ter a capacidade de marcar essa notificação como "lida", garantindo que a mesma informação não continue a ser exibida repetidamente.

O sistema deve exibir gráficos mostrando os valores lidos pelos sensores nos dias anteriores de forma agregada, separando por máquinas

Esse requisito assegura que os usuários possam visualizar, por meio de representações gráficas, as leituras dos sensores de dias anteriores de forma agregada. Estes gráficos devem ser categorizados por máquina, proporcionando uma análise detalhada do desempenho de cada equipamento ao longo do tempo.

O sistema deve exibir nos gráficos uma análise estatística do funcionamento das máquinas, junto com o valor máximo de funcionamento ideal

Os gráficos devem oferecer uma análise estatística, mostrando os indicadores estatísticos dos dados agregados média, mediana, percentil 75 e média removendo os outliers. Juntamente com isso, o gráfico também mostrará o valor ideal, servindo como uma referência para avaliar o desempenho.

O sistema deve permitir filtrar as informações exibidas em tela por máquinas

Os usuários devem ter a flexibilidade de selecionar e visualizar informações específicas para determinadas máquinas, permitindo que eles se concentrem em equipamentos específicos conforme a necessidade.

O sistema deve permitir filtrar os gráficos exibidas em tela por data

O sistema deve oferecer a capacidade de os usuários filtrarem as exibições gráficas por datas específicas, permitindo análises temporais detalhadas e comparações entre diferentes períodos.

O sistema deve exibir os gráficos de parada das máquinas de forma a exemplificar a exibição desses dados

O sistema deve mostrar as paradas de máquina de acordo com os dados passados pelas planilhas com os dados. Dessa forma poderá ser exemplificado como ficaria as informações

de parada das máquinas caso o sistema recebesse essas informações.

3.1.2 Requisitos Não Funcionais

Requisitos não funcionais são especificações que determinam as características de desempenho, usabilidade, confiabilidade e outras propriedades que o sistema deve possuir, ao invés de comportamentos específicos que ele deve demonstrar. Enquanto os requisitos funcionais descrevem o que um sistema deve fazer, os requisitos não funcionais especificam como o sistema deve realizar essas funções.

Estes requisitos são cruciais para garantir a satisfação do usuário e a eficácia operacional do sistema, desempenhando um papel fundamental na qualidade e na operação geral de um produto de software.

Os requisitos não funcionais podem ser de vários tipos, como usabilidade, desempenho, segurança, disponibilidade, manutenção e confiabilidade.

Dentro do contexto do sistema desenvolvido, está listado abaixo os requisitos não funcionais do sistema e sua descrição, de forma a deixar claro o que foi levado em consideração no momento de desenvolver cada uma das funcionalidades do sistema.

Disponibilidade

O sistema deve possuir mecanismos de reconexão automática que se ativam quando problemas de conexão ou recebimento de dados dos sensores forem detectados, garantindo assim a continuidade no recebimento dos dados.

Segurança no acesso

O sistema deve implementar controles de acesso para que somente colaboradores autorizados tenham permissão para acessar os dados e funcionalidades pertinentes ao seu papel.

Segurança na rede

Para garantir a segurança da transmissão de dados, a conexão ao sistema deve ser estabelecida utilizando o protocolo HTTPS, que incorpora a camada de segurança TLS, protegendo assim os dados contra interceptações e alterações.

Transmissão em tempo real

O sistema deve processar e transmitir os dados enviados pelos sensores em uma arquitetura baseada em streaming. O atraso entre o envio do dado pelo sensor e sua visualização pelo usuário final deve ser inferior a três segundos.

Modularidade

A arquitetura do sistema deve ser modular, possibilitando a integração e a adição de novos componentes ou funcionalidades de maneira eficiente e sem comprometer o funcionamento das partes já existentes.

Manutenibilidade

Priorizando a longevidade e facilidade de manutenção, o sistema deve ser desenvolvido seguindo boas práticas de programação e os princípios do SOLID. Isso facilitará futuras modificações, expansões e a correção de eventuais problemas.

Escalabilidade de sensores e máquinas

O design do sistema deve ser capaz de lidar com um crescente volume de sensores e máquinas, garantindo que não haja degradação de performance ou falhas quando a demanda por recursos aumentar.

Portabilidade

O sistema deve garantir compatibilidade com os principais navegadores web disponíveis no mercado. Além disso, a interface de usuário deve se adaptar bem em telas maiores

como televisões, permitindo que o dashboard seja visualizado de forma clara em diferentes ambientes da fábrica.

Usabilidade

A interface do sistema e seus componentes devem ser projetados considerando princípios fundamentais de design de interação, garantindo que os usuários possam compreender e interagir com o sistema de maneira intuitiva e eficiente.

3.2 Método de coleta e armazenamento de dados

como que seria feito o recebimento e o armazenamento dos dados - Recebimento dos dados pelo protocolo feito pelo outro professor - protocolo esse que decodifico as informações e disponibilizo para uso

Dentro do contexto do projeto, a forma como os dados dos sensores são coletados e armazenados influenciam muito no funcionamento do sistema, pois é a partir deles que todo o sistema é estruturado. Dessa forma foi utilizado como base um protocolo desenvolvido em outro projeto, que transmite todas as informações necessárias para o contexto desse projeto. Dentro do sistema em questão, ficou a responsabilidade de implementar o decodificador para o determinado protocolo.

O protocolo utilizado foi desenvolvido dentro do mesmo contexto desse projeto, dentro do IPB, para atender uma demanda da mesma empresa, portanto ele se tornou a opção mais ideal dentre as possibilidades, otimizando a comunicação entre os sensores e o sistema. O formato é estruturado de forma a representar distintamente as informações pertinentes à máquina, ao tipo de comunicação, ao sensor e ao significado dos dados transmitidos.

Machine ID (2 bytes)

O campo *Machine ID* é responsável por identificar a máquina em questão e está dividido em dois subcampos:

- **High (bytes de ordem superior):** Representa o tipo da máquina. Exemplos de valores possíveis são: prensa, torno, robot, tapete, entre outros.
- **Low (bytes de ordem inferior):** Identifica o número da máquina.

Type (1 byte)

O campo *Type* indica o tipo de mensagem e pode assumir os seguintes valores:

1. Publish
2. Request to publish

Sensor ID (2 bytes)

O campo *Sensor ID* fornece detalhes sobre o sensor que está transmitindo os dados:

- **High (bytes de ordem superior):** Representa a quantidade física sendo medida, como temperatura, velocidade, pressão, força, entre outros.
- **Low (bytes de ordem inferior):** Indica o número do sensor.

Meaning of Data (2 bytes)

O campo *Meaning of Data* fornece informações sobre o tipo e significado dos dados:

- **High (bytes de ordem superior):** Tipo dos dados:
 1. Not defined
 2. Normal
 3. Raw data
 4. Alarm
- **Low (bytes de ordem inferior):** Significado dos dados, que varia de acordo com o equipamento. Exemplos incluem:

- Oil critical temperature
- Check oil temperature
- Oil pressure

Length (2 bytes)

O campo *Length* indica o número de bytes subsequentes no pacote.

Data

Este campo representa os dados transmitidos pelo sensor. A especificação exata do que os dados representam deve ser definida e normalizada, conforme indicado pela notação (*).

3.3 Processo de desenvolvimento do software

- Definição do escopo - Comparar a ideia inicial com o escopo entregue (?) - Organização das tarefas em um kanban - Notion para documentação - Github - Reuniões semanais com o professor para discutir sobre o andamento

3.4 Tecnologias

Desenho arquitetural e e uma seção para cada tecnologia da stack

3.5 Gestão das atividades do projeto

- Historias de usuario - Listagem do que deveria ser feito no kanban no notion

3.6 Desafios para obtenção dos dados

Descrever mais sobre como foi complexo ter uma modelo de dados bem definido, como as empresas tem políticas que podem dificultar o acesso a informações que podem ser

necessárias para o sucesso de um projeto como esse

Capítulo 4

Arquitetura do sistema

In this chapter, you should describe the implementation, highlighting the most important aspects, the difficulties and the technical solutions that were followed. In particular, if code from others was used (available as open-source), should be easily identified.

4.1 Diagrama do sistema

texto e a imagem....

4.2 Princípios do SOLID

Topicos explicando cada uma das letras do SOLID e relacionando com exemplos concretos do sistema

4.3 Arquitetura backend

API

- Camada de recebimento das requisições - Regras de negocio e modelos do sistema -
Camada de infraestrutura

Python Scripts

4.4 Arquitetura do frontend

- Next JS e estrutura pre definida de pastas para as paginas do sistema - Componentes e layouts - Context API e seu uso - Acesso a camada externa (API)

4.5 Containers

Capítulo 5

Implementação

This chapter presents and describes the tests that were developed to check if the project fulfills the objectives and solves the problem described in Analysis/Methodology.

To better understand, the results of each test should be preceded by a description of the test and the expected results.

The work results are commented, including:

- What can be learned from the results?
- What could be done differently?
- What was beyond initial objectives?
- What are the objectives there were not met and why?

5.0.1 Implementação do banco de dados

- Databases - Collections

5.0.2 Implementação da API

- Uvicorn usado pelo FastAPI e sua forma assíncrona - Biblioteca Motor usada para acesso ao MongoDB - Biblioteca Pydantic para criação dos modelos e tipos - Web socket

para envio de notificações - Biblioteca Jose para autenticação - Comunicação entre as camadas com a classe Result - Contratos de interfaces - Tratamento de erros com classes personalizadas

Para interpretar e extrair informações da mensagem recebida do multicast, é crucial decodificar adequadamente a mensagem de acordo com o protocolo definido anteriormente. A implementação dessa decodificação é feita pela função `__parse_multicast_message`.

5.0.3 Interpretação dos Bytes

A primeira parte da decodificação envolve interpretar os bytes da mensagem. A função auxiliar `__parse_bytes` é utilizada para essa tarefa. Dada uma sequência de bytes, esta função interpreta os bytes utilizando a ordem big-endian (onde os bytes mais significativos vêm primeiro):

```
def __parse_bytes(self, bytes):
    data = int.from_bytes(bytes, byteorder='big')
    high_data = (data >> 8) & 0xFF
    low_data = data & 0xFF

    return (high_data, low_data)
```

Aqui, `data` contém o valor inteiro dos bytes fornecidos. O byte de ordem superior (High) é extraído deslocando o valor 8 bits para a direita e aplicando uma operação "E"(&). O byte de ordem inferior (Low) é simplesmente obtido aplicando a operação "E" com 0xFF.

5.0.4 Decodificação da Mensagem

Com a capacidade de interpretar os bytes, a função principal `__parse_multicast_message` pode começar a decodificação:

- Primeiro, ela extrai o tipo de máquina e o número da máquina dos dois primeiros bytes da mensagem.
- O terceiro byte da mensagem é então interpretado como o tipo da mensagem. Se o tipo da mensagem for 2, a função retornará diretamente uma solicitação para publicar.
- Os bytes 4 e 5 são interpretados como o ID do sensor, que contém a quantidade física sendo medida e o número do sensor.
- Os bytes 6 e 7 são usados para extrair o tipo de dados e seu significado.
- Finalmente, os bytes 8 e 9 são usados para determinar o comprimento dos dados que seguem.

A informação extraída é então organizada em um dicionário para representação clara e fácil acesso aos componentes individualmente:

```
message_dict = {
    'Machine': {
        ...
    },
    'Type': ...,
    'Sensor': {
        ...
    },
    'MeaningOfData': {
        ...
    }
}
```

Esta estrutura permite uma representação clara e modular da mensagem decodificada, tornando fácil a integração e utilização em outras partes do sistema.

5.0.5 Implementação do módulo de processamento de dados

- Leitura dos dados de acordo com o processamento realizado anteriormente - Uso da biblioteca pandas - Funcionamento do BoxPlot para realizar análise estatística

5.0.6 Implementação do frontend

- Criação de paginas componentes de layouts - Recharts - Material UI - Days JS - Criação da camada de dados com o Context API - Acesso externo a API - Axios e fetch

5.0.7 Adaptando a implementação para outros contextos

Discussão sobre a reutilização do sistema para outros contextos.... - como fazer - alterações necessarias

Capítulo 6

Características do Sistema do ponto de vista funcional

Adicionar capturas de tela e diagramas para exemplificar o uso

6.0.1 Monitoramento em tempo real

6.0.2 Sistema de alertas e nnotificações

e como isso ajuda a empresa

6.0.3 Dados historicos

6.0.4 Interface do usuário

- Exibição no computador - Exibição na televisão

Capítulo 7

Resultados e Avaliação

Apresentação dos resultados obtidos - Discutir as principais realizações do projeto, incluindo a capacidade de monitorar em tempo real as máquinas da planta e a análise histórica do funcionamento das máquinas.

- Sistema que ajuda no monitoramento das maquinas da planta - Exibir para os funcionários como as maquinas estão funcionando - Gerar insights a partir dos dados gerados

Capítulo 8

Conclusão e Trabalhos Futuros

The conclusions should synthesize and provide a single view to the work developed. It can be done a brief reference to similar work of others and to the knowledge that emerged from it, as well as future work suggestions. The consistency of the document implies that the conclusions should be coherent with the main ideas in the introduction.

8.0.1 Resumo

8.0.2 Limitações do sistema

(Adicionar novas camadas de dados e paginas no dashboard pode ser demorado)?

8.0.3 Sugestões para trabalhos futuros

- melhorias de código - Monitoramento do funcionamento do sistema e logs - Log para recebimento dos dados e quando para de receber - Log para a analise estatistica - Log para erros no recebimento dos dados - Generalização para outros contextos - destacando a tendência crescente de coleta e análise de dados na indústria - Melhoria da performance para grande numero de maquinas e graficos exibidos na tela ao mesmo tempo. - Upgrade para next 13 e 14. Pois com isso podemos ter o beneficio dos server componentes, o que melhoraia a performace da aplicação - Os graficos podem ser carregados como server

componentes com cache equivalente ao intervalo que o modulo de processamento roda, otimizando assim o frontend da aplicação

Apêndice A

Proposta Original do Projeto



Curso de Licenciatura em Engenharia Informática
Projeto 3º Ano - Ano letivo de 2016/2017

<Título do projeto>

Orientador: <Nome do orientador>

Coorientador: <Nome do coorientador>

1 Objetivo

<Objetivo do projeto>

2 Detalhes

<Detalhes que julguem ser necessários>

3 Metodologia de trabalho

<Eventual metodologia de trabalho>

Dimensão da equipa:

Recursos necessários:

Apêndice B

Outro(s) Apêndice(s)

Listagens de código fonte, texto/imagens produzidos por testes complementares, etc.