

PESQUISA E INOVAÇÃO

1ºCCO

Monitoramento da temperatura em motores a jato equipados em frotas de Companhias Aéreas

Contexto:

No contexto brasileiro, a aviação civil é uma das mais confiáveis do mundo, tendo aparecido múltiplas vezes em quarto ou quinto lugar no ranking de segurança operacional, segundo dados da ICAO(International Civil Aviation Organization; Em tradução livre, Organização da Aviação Civil Internacional) em 2017, 2018 e 2020, contudo, tais níveis de segurança exigem dos órgãos reguladores como a ANAC(Agência Nacional de Aviação Civil) um alto grau de regulação sobre as manutenções das aeronaves, estipulando de maneira simples a manutenção por ciclos em motores, essas revisões também são estipuladas por ciclos, que são basicamente pousos e decolagens completadas (um voo completo), dependendo do tipo de motor e tipo de operação, vigente sobre diversos RBAC (Regulamentos Brasileiros da Aviação Civil), os ciclos para revisão podem ser mais distantes ou mais próximos, podendo ser desde de 25 ciclos a 150 ciclos.

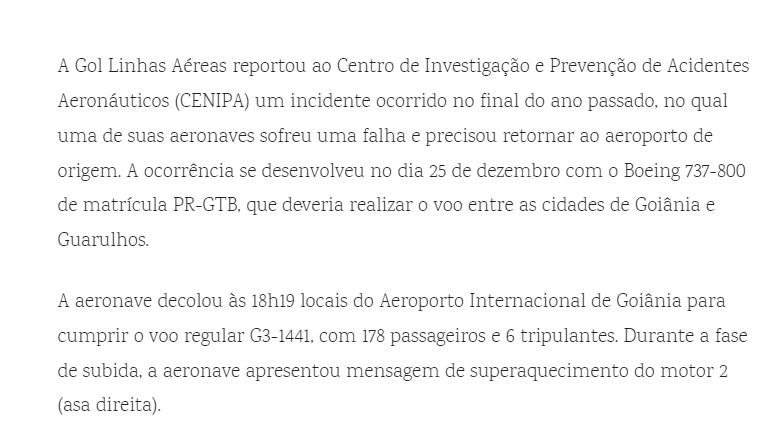
Sendo assim, por causa da distância de ciclos muitas vezes longe, a companhia acaba optando por manter as manutenções nas revisões esperadas preditivas, sendo que quando analisadas nesses estados, muitas das vezes os motores já estão em condições que exigem várias trocas de componentes na qual custam muito ao capital das empresas, principalmente as que operam sobre a regra “low-cost” que determina que o custo seja baixo, reduzindo todas as despesas operacionais desnecessárias, este é um dos motivos na qual recentemente a GOL Linhas Aéreas pediu recuperação judicial, pois as manutenções estão dando mais despesa do que lucro.

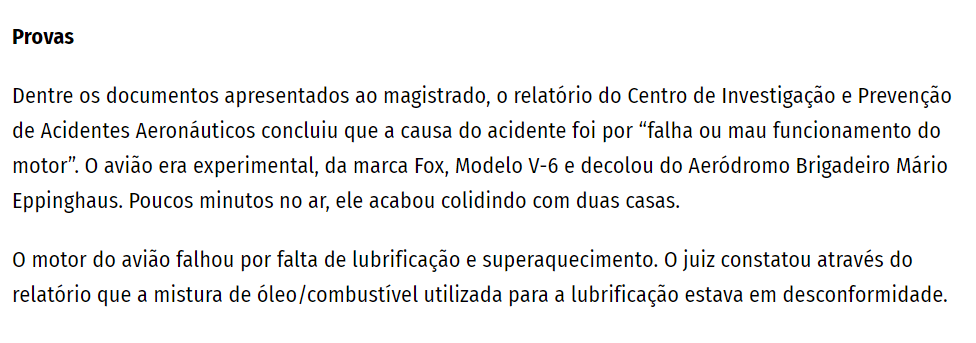
Aliado ao elevado índice de impostos sobre peças exportadas, combustível e licenças, isso se torna um problema para as empresas. Deixar as manutenções para as datas preditivas pode resultar na perda momentânea de uma aeronave, a qual não poderá ser utilizada por vários dias (entre 5 e 35 dias), acarretando custos milionários para as empresas. Isso se deve ao preço relacionado ao custo de manutenção, somado ao imposto aplicado em peças exportadas.

  
Infográfico representando os custos e receitas de serviços aéreos, mostrando prejuízo de 15,2 bilhões de reais, como se pode observar, os dois maiores fatores são Segurança, Arrendamento e Manutenção de Aeronaves e Combustível.

Um dos fatores extremamente prejudiciais à uma aeronave, são problemas relacionados a superaquecimento dos motores, podendo gerar desgaste nos componentes, e como consequência, falhas mecânicas graves que afetam o desempenho da aeronave como um todo. Para tentar contornar o problema foi criado um sistema de acompanhamento por satélite/ACARS (Aircraft Communication Addressing and Reporting System), mas a implantação deste sistema é opcional(pago) para as empresas aéreas e a extração dos logs é demorada, podendo ser cíclicas (com uma certa frequência) ou após pouso, mas nunca em tempo real, estudos mostram que um sistema via satélite em tempo real custaria as empresas cerca de 300 milhões de dólares por ano.

Abaixo estão alguns exemplos de incidentes em voo provocados pelo superaquecimento dos motores:

  
Fonte:  https://aeroin.net - Um dos casos em que o superaquecimento do motor deu grande prejuízo a companhia Gol Linhas Aéreas. O voo teve que ser cancelado e o avião ficou parado por um dia para manutenção.

  
Fonte:  https://opopular.com.br - Queda de avião particular em Goiânia mata bebê. O motivo da queda foi devido ao superaquecimento dos motores.

Além dos casos acima, de acordo com a fonte https://aeroin.net, no curto período de janeiro até março de 2020, foram registrados dois incidentes por um falso aviso de superaquecimento nos motores, o que demonstra que o sistema de monitoramento está muito suscetível a falhas.

Todos esses incidentes poderiam ter sido evitados caso existisse um monitoramento em tempo real por parte das companhias aéreas, utilizando como exemplo o voo da Gol que precisou aterrissar por superaquecimento no motor, este problema poderia ter sido evitado caso as estatísticas de outros voos tivessem sido monitoradas, pois seria possível perceber que os níveis de calor estavam ficando acima do normal, cada vez mais rápido.

Objetivos:

* Uma solução IoT para monitoramento de temperatura dos motores em frotas de aviões comerciais a jato (TurboFan), pelo tempo operacional útil do motor.
* Website institucional com disponibilidade dos dados e estatísticas adquiridos dos motores.
* Um dashboard de monitoramento de dados para companhias clientes.

Justificativa:

Economia de até 300 milhões de dólares anuais com monitoria de temperatura dos motores.

Escopo

A proposta da solução é a implementação de um sistema IoT (Internet of things) que possibilitará uma rápida visualização da condição de temperatura dos motores nas frotas, facilitando tomadas de decisão por parte da equipe de manutenção, isso seria viável através de um processo/caminho lógico resumido a seguir:

1. Captura das temperaturas

Os dados serão capturados com um sensor de temperatura instalado em cada motor das aeronaves. Para protótipo, utilizamos o Arduino Uno com o sensor LM35.

2. Servidor / API

Todas as temperaturas, capturadas pelo sensor, serão enviadas a um servidor através de uma API, que será responsável pela troca de informações entre o sensor de temperatura, banco de dados e interface web.

Outra API será utilizada para controle de usuários e autenticação.

3. Banco de dados

Após as informações serem enviadas ao servidor, tradadas e validadas, os registros ficarão armazenados em um banco de dados.

4. Interface web (Site)

Após todo o fluxo entre sensores, servidor e banco de dados, a aplicação web será responsável por requisitar todos os registros através das APIs e exibir na página em forma de gráficos e tabelas ao usuário final.

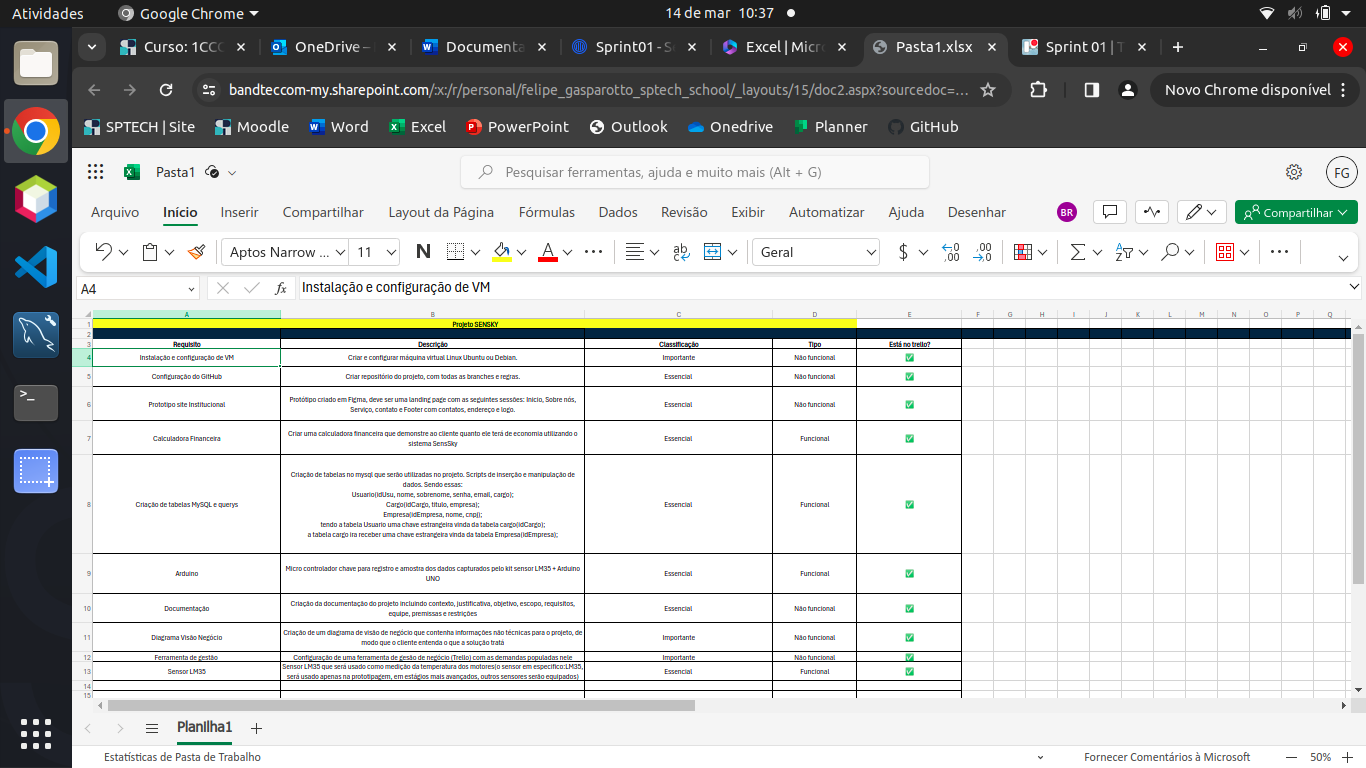
Premissas

Para viabilidade do desenvolvimento e operação do projeto, partimos das seguintes premissas:

* O cliente arcará com todo o custo necessário para o desenvolvimento do projeto (Licenças, computadores, Arduino e componentes), além de disponibilizar um local onde a equipe de desenvolvimento possa se reunir. O cliente será responsável por manter toda a infraestrutura e limpeza deste local.
* O cliente arcará com todo o custo relacionado a treinamento de equipes para cuidar da operação e manutenção da plataforma.
* O cliente arcará com todo custo relacionado a infraestrutura, como hospedagem das APIs, banco de dados e site web.
* O nosso sistema irá funcionar apenas nos motores da TurboFan.

Requisitos

Para o sucesso da solução SensSky, a aplicação deve atender os seguintes requisitos:



Restrições

Algumas restrições são importantes:

1. O sistema é compatível apenas com aeronaves do tipo TurboFan e só faremos a instalação em aeronaves com esse motor;
2. Caso ocorra a queda da conexão por parte da aeronave os dados voltarão a ser enviados somente com a volta da conexão, tendo em vista que, todos os dados adquiridos durante o período offline serão perdidos;
3. Garantimos compatibilidade da interface web apenas com os navegadores: Microsoft Edge, Opera, OperaGX e Google Chrome;
4. Não nos responsabilizamos por troca do sensor por utilização de forma indevida.

Banco de Dados

O banco de dados da aplicação foi construído utilizando o SGBD MySql, por meio da IDE Mysql Workbench, contendo 3 tabelas, estando sujeitas a modificação dos campos e adição ou remoção de demais tabelas.

Os registros abaixo são apenas para carater ilustrativo:

Tabela Usuario:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Usuario | | | | | |
| idUsu | nome | sobrenome | senha | email | cargo |
| 1 | Bryan | Rocha | \*\*\*\*\*\*\*\* | [bryan@gmail.com](mailto:brayan@gmail.com) | 2 |
| 2 | Gustavo | Amorim | \*\*\*\*\*\*\*\*\* | [gustavo@gmail.com](mailto:gustavo@gmail.com) | 1 |
| 3 | Lucas | Matos | \*\*\*\*\*\*\*\*\* | [lucas@gmail.com](mailto:lucas@gmail.com) | 4 |
| 4 | Alexandre | Vasconcelos | \*\*\*\*\*\*\*\*\* | [alexandre@gmail.com](mailto:alexandre@gmail.com) | 3 |

Atributos:

* idUsu:
  + Tipo INT;
  + Chave Primária;
  + Irá receber o código de identificação do registro dentro da tabela.
* nome:
  + Tipo VARCHAR tamanho 30 caracteres;
  + Ira receber o nome do usuário do sistema.
  + Não pode ser NULO
* sobrenome:
  + Tipo VARCHAR tamanho 60 caracteres;
  + Ira receber o sobrenome do usuário do sistema.
  + Não pode ser NULO
* senha:
  + Tipo VARCHAR tamanho 60 caracteres;
  + Ira receber a senha criptografada do usuário.
  + Não pode ser NULO
* email:
  + Tipo VARCHAR tamanho 60 caracteres;
  + Chave única;
  + Possui uma chave de checagem à qual irá verificar se o registro possui um ‘@’ e um ‘.’ não permitindo que ambos estejam encostados um no outro e exigindo que eles estejam presentes no registro.
  + Não pode ser NULO
* cargo:
  + Tipo INT;
  + Chave estrangeira vinda da tabela "Cargo”;
  + Ira receber o código referenciando o campo idCargo da tabela Cargo;
  + Não pode ser NULO

Tabela Empresa:

|  |  |  |
| --- | --- | --- |
| Empresa | | |
| idEmpresa | nome | cnpj |
| 1 | Empresa 1 | 12345678932165 |
| 2 | Empresa 2 | 98765432103254 |
| 3 | Empresa 3 | 45821384835884 |

Atributos:

* IdEmpresa
  + Tipo INT;
  + Chave primária;
  + Ira receber o código de identificação do registro dentro da tabela.
* nome:
  + Tipo VARCHAR tamanho 30 caracteres;
  + Ira receber o nome da empresa.
  + Não pode ser NULO
* cnpj
  + Tipo CHAR tamanho 14 caracteres;
  + Chave única;
  + Ira receber o CNPJ da empresa.
  + Não pode ser NULO

Tabela Cargo:

|  |  |  |
| --- | --- | --- |
| Cargo | | |
| idCargo | titulo | empresa |
| 1 | Supervisor | 1 |
| 2 | Operador | 3 |

Atributos:

* IdCargo
  + Tipo INT;
  + Chave Primária;
  + Ira receber o código de identificação do registro dentro da tabela.
  + Não pode ser NULO
* titulo
  + Tipo VARCHAR tamanho 30 caracteres;
  + Ira receber o nome do cargo.
  + Não pode ser NULO
* empresa
  + Tipo INT;
  + Chave estrangeira da tabela Empresa;
  + Irá receber o código referenciando o campo idEmpresa da tabela Empresa
  + Não pode ser NULO

Prototipagem:

A prototipagem da solução será feita através de um Arduino modelo uno e com o sensor de temperatura LM35. O sensor de temperatura será conectado ao Arduino e este será conectado ao computador e se comunicará com nossa API localmente durante o desenvolvimento.

Durante todo o desenvolvimento do projeto a prototipagem será realizada localmente, simulando um cenário real e todos os testes necessários para o funcionamento do projeto em produção.

Interface Web:

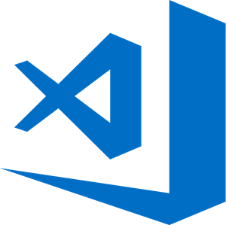
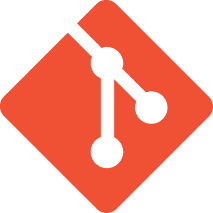
A interface Web será um site dinâmico, responsável por renderizar todas as informações enviadas pela API da plataforma. As tecnologias utilizadas serão:

1. HTML
2. CSS
3. JavaScript

As informações estatísticas contidas nos gráficos e tabelas gerados pela plataforma, serão atualizadas a cada 3 minutos.

Todo o desenvolvimento da interface web será realizada dentro da IDE Visual Studio Code.

A interface será testada nos navegadores OperaGX, Opera e Google Chrome. Não nos responsabilizaremos pelo mal funcionamento da aplicação em navegadores não especificados nesta documentação.

Ambiente de Desenvolvimento:

Versionamento:

O versionamento da aplicação foi realizado com uso do da ferramenta Git em conjunto do GitHub para versionamento remoto.

No qual foi feito a seguinte estruturação:

Cada integrante da equipe de desenvolvimento possui um Commit próprio na qual deverá subir suas atualizações e modificações do sistema.

Após as modificações serem enviadas, passarão por análise e aprovação da equipe,

Caso as atualizações sejam aprovadas elas serão mandadas por Commit de Testes onde a aplicação será testada junto das novas funcionalidades adicionadas.

Após todos os testes terem sido realizados e não haja ocorrido nenhum problema, as atualizações serão enviadas para Branch main/master de produção na qual só podem ser enviadas funcionalidades que já passaram por análise e aprovadas por toda a equipe do projeto.