

## Relatório Final

Título do projeto de pesquisa: Desenvolvimento de um aplicativo para visualização de dados de telemetria baseada em Ethernet

Bolsista: Douwe De Jong

Orientador(a): Alison de Oliveira Moraes

Período a que se refere o relatório: Fevereiro de 2018 a Julho de 2018

### Resumo

O IAE está desenvolvendo um hardware e firmware de um decomutador de telemetria para ser utilizado em seus veículos de sondagem. Este projeto teve como objetivo desenvolver um aplicativo para ser usado em testes com este decomutador que está em desenvolvimento pelo Laboratório de Aquisição e Processamento de Dados (APDT) do Instituto de Aeronáutica e Espaço (IAE). Com este aplicativo será possível realizar testes para visualização de dados de telemetria em tempo real para que cientistas e pesquisadores do IAE possam acompanhar testes e ensaios de cargas úteis. Entre os desafios deste projeto vale citar a implementação de uma interface visual configurável para que cada usuário possa ajustá-la de maneira desejada de acordo com suas necessidades. O desenvolvimento deste aplicativo foi feito em Javascript junto com suas diversas ferramentas e frameworks online; uma vez que a interpretação dos dados está sendo usada em linguagem C. Ao término deste trabalho, foi possível obter uma primeira versão deste aplicativo que está funcional.

### 1. Introdução

Telemetria engloba o processo no qual medições realizadas por sensores embarcados são transmitidas para uma estação distante em solo, a Figura 1 ilustra a etapa de aquisição de dados de telemetria a bordo de foguetes. Na estação receptora em solo, os dados são recebidos, gravados, decomutados e analisados, como mostra a Figura 2. É neste conjunto de solo, que o decomutador opera, sendo este responsável pela extração dos dados de telemetria que foram transmitidos de forma multiplexada. A aplicação dessa técnica é essencial para o

desenvolvimento de foguetes no IAE, já que é necessário obter dados críticos do foguete em tempo real e a uma distância segura durante ensaios e em operações de lançamento.

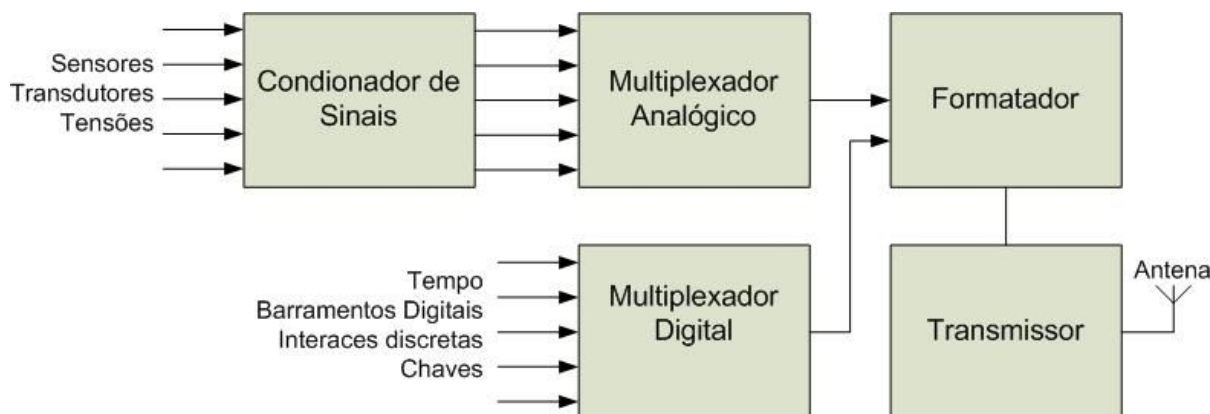


Figura 1: Arquitetura de uma rede de telemetria de bordo.

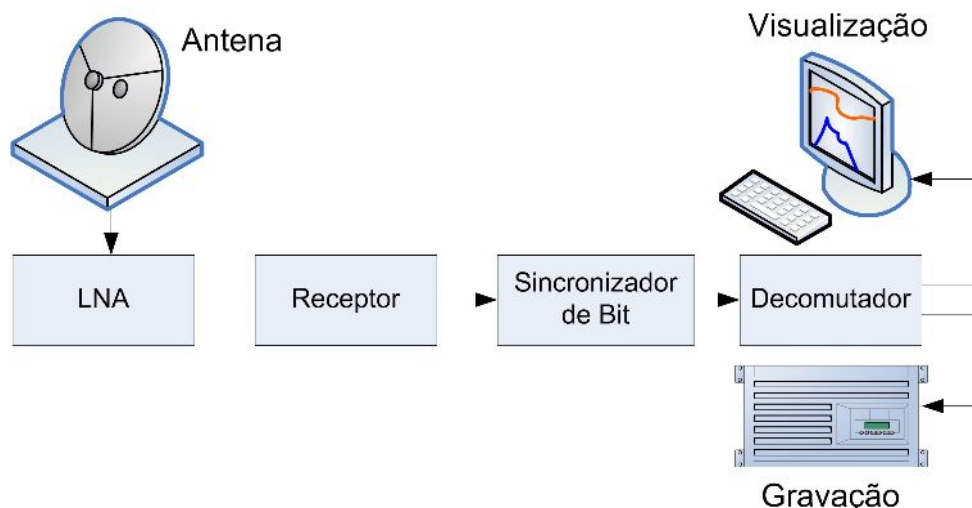


Figura 2: Arquitetura da estação de recepção de telemetria.

O projeto agora passou pela fase de desenvolvimento da parte de mostração de dados. A interface, embora funcional ainda não está "bonita", mas já consegue cumprir seu objetivo. Ser um programa com interface customizável e rápida o bastante para suportar uma quantidade grande de dados. O próximo passo do desenvolvimento seria projetar o sistema que recebe os dados e manda para a interface.

Os assuntos abordados neste trabalho possibilitarão melhorias no projeto de cargas úteis de foguetes do IAE, economizando espaço e massa principalmente que são aspectos essenciais em projetos de lançadores principalmente.

## 2. Materiais

O aplicativo desenvolvido deverá atuar em conjunto com o decomutador PCM, com interface de saída Ethernet que esta em desenvolvimento no IAE. A Figura 3 a seguir ilustra um diagrama de funcionamento do aplicativo desenvolvido (em rosa) integrado ao decomutador. o decomutador receberá os dados de telemetria codificados em PCM. Este dispositivo vai extrair os dados do formato transmitido, identifica-los e transmiti-los via Ethernet.

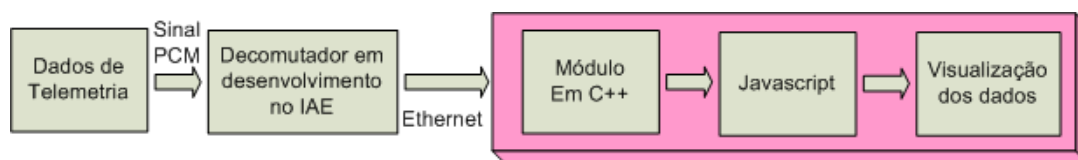


Figura 3: Diagrama da visão geral do funcionamento do aplicativo, com as partes envolvidas.

Analisando a Figura 3, no módulo em C++ acontece o processamento mais pesado do aplicativo. Os dados entram por uma porta especificada pelo usuário, a partir de um protocolo especificado, e são identificados e organizados pelo programa fazendo os processamentos de dados necessário para a próxima etapa. Na etapa seguinte do programa, é feita a exibição dos dados. Antes de mostrados os dados, o usuário deve criar seu layout de exibição e endereçar a informações que ele deseja ver, aqui chamadas de pipelines. Feito isso, na última etapa do programa, os dados são inseridos nos formatos gráficos pré-selecionados para que possam ser visualizados. Com relação a parte de programação, o aplicativo desenvolvido tem como plataforma principal as bibliotecas de Javascript chamadas de Reactjs, Electronjs e Canvasjs. Além disso, ao longo do desenvolvimento foram usados alguns pacotes para auxiliar o desenvolvimento, entre estes vale citar: babel-core, babel-loader, babel-polyfill, babel-preset-es2015, babel-preset-react, babel-preset-react-hmre, concurrently, css-loader, express, node-sass, webpack, webpack-dev-middleware, webpack-hot middleware, after, bootstrap, keygenerator, react-bootstrap, react-dom, react-jss, react-modal, react-rnd, react-router, react-router-dom, react-table.

### 3. Métodos

Como o desenvolvimento do aplicativo ocorreu em paralelo com o desenvolvimento do hardware e firmware do decomutador, foi necessário criar um simulador virtual do decomutador. Esta simulação foi implementada através da criação de um servidor em Javascript com protocolo TCP que enviava dados de telemetria. A Figura 4 ilustra um diagrama do simulador integrado a aplicativo de interface gráfica.

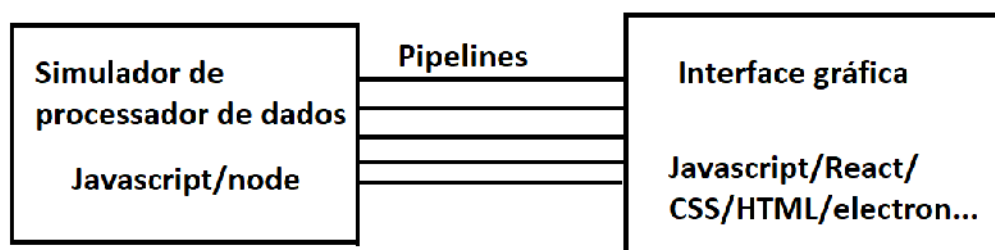


Figura 4: Diagrama do simulador de dados de telemetria via Ethernet desenvolvido.

Este simulador envia os dados para a interface gráfica que atualiza os componentes ativos. Nesta abordagem foi utilizada a função pipe faz parte da do conjunto de rotinas (API) Stream do node.js. Desta forma o servidor envia bytes de dados que são recebidos pela interface gráfica. No caso o pipe transforma algo que seria lido pelo decomutador para algo que será processado no aplicativo, ou seja, ele transforma um fluxo legível para um fluxo de gravação. A Figura 5 ilustra o diagrama de funcionamento da interface gráfica. A interface gráfica foi construída de tal maneira que o usuário tenha 3 opções: a opção de carregar um layout de gráficos já produzidos, a opção de criar um layout novo ou a opção de carregar uma versão criada anteriormente. O layout de gráficos é como os gráficos devem ser definidos e as pipelines são os dados que serão recebidos por cada gráfico dado o layout.

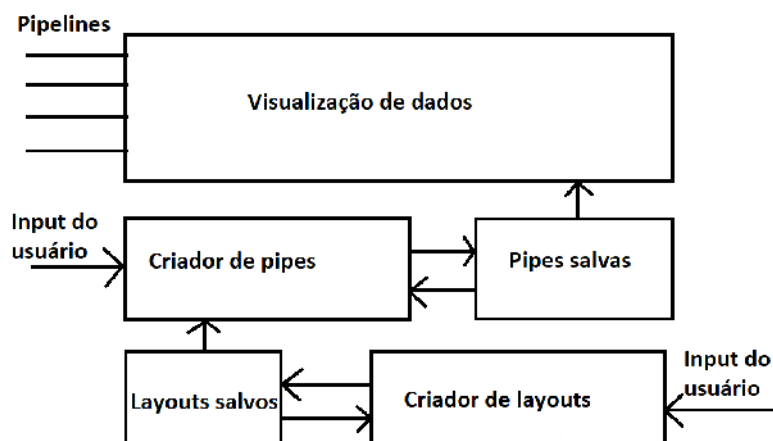


Figura 5: Diagrama de funcionamento da interface gráfica.

#### 4. Resultados

Nesta seção serão apresentadas as interfaces do aplicativo desenvolvido. Iniciando a apresentação das interfaces, a Figura 6 ilustra um exemplo da etapa de montagem dos layouts que o usuário vai escolher para visualização dos dados de telemetria. Como é possível observar, o usuário pode posicionar, mudar o tamanho dos componentes e os classificar como tabelas, botões ou gráficos. Desta forma a interface pode atender a necessidade de qualquer teste que possa ocorrer.

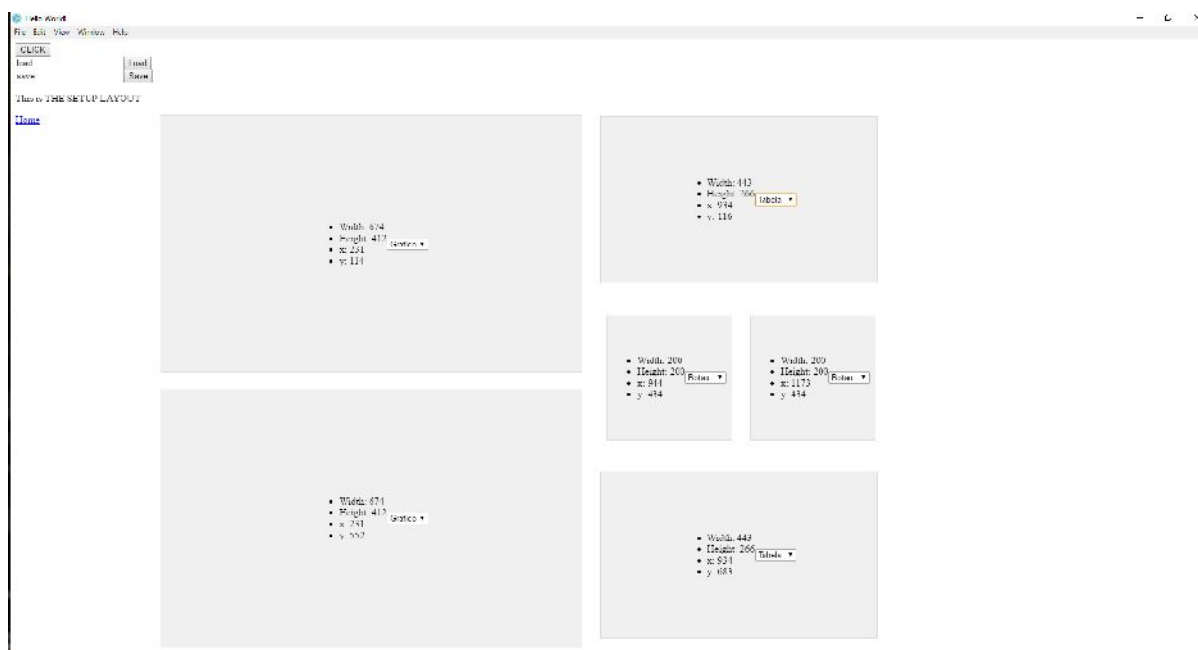


Figura 6: Interface configurável de visualização dos dados.

Na segunda interface do aplicativo é possível observar a montagem realizada na interface anterior da Figura 6, conforme mostra a Figura 7. Nesta 'tela' são apresentadas as informações de telemetria que foram selecionadas para ser mostrada nas suas respectivas interfaces, neste caso, neste caso, dois campos de plots, duas tabelas e dois botões de indicação de estado por troca de cor. Esta 'tela' é um exemplo claro de uma interface que pode ser utilizada em ensaios de foguetes do IAE. Este dados serão imediatamente mostrados quando o decomutador transmiti-los via Ethernet. Esta abordagem é muito vantajosa pois um único hardware poderá ser utilizado por múltiplos usuários simultaneamente e cada usuário vai ter sua visualização configurável de modo particular a suas necessidades. Um resumo do trabalho desenvolvido e descrito nesta seção é explicado em detalhes no link: <https://youtu.be/p-5QwgB8sao>

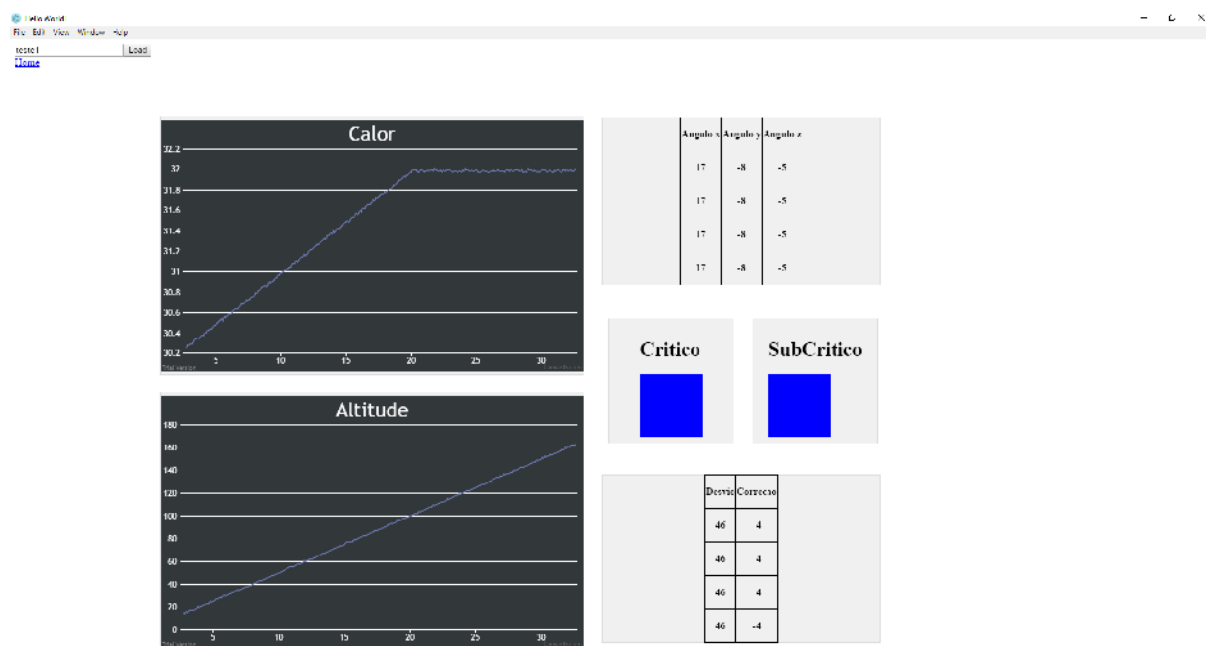


Figura 7: Interface configurada pelo usuário para visualização de dados de telemetria.

## 5. Conclusões e próximas etapas

Neste trabalho foi desenvolvida uma interface funcional e rápida para visualização de dados de telemetria de foguetes distribuídos por um decomutador PCM com interface Ethernet. Neste trabalho foram utilizadas as linguagens HTML, C++ e Javascript. O uso de diversas

linguagens de programação em conjunto foi um desafio mas proporcionou uma experiência construtiva que agregou a formação profissional do aluno. Além disto foi provado o conceito do uso deste tipo de aplicativo que adquire dados de telemetria via Ethernet. Portanto os objetivos do trabalho foram atingidos de forma satisfatória. Apesar disto ao longo do desenvolvimento surgiram sugestões de melhoria que pretende-se implementas como trabalhos futuros. Como exemplo vale citar a interface de visualização dos dados que está funcional conforme mostrado na Figura 7 ainda pode ser melhorada com mais opções de visualizações, além da inclusão de alertas e medidas estatísticas. Estas melhorias podem fazer com que esta ferramenta seja ainda mais útil para o pesquisadores que venham a usar este aplicativo.

## Referências

Electronjs.org. (2018). *Electron / Build cross platform desktop apps with JavaScript, HTML, and CSS.* [online] Available at: <https://electronjs.org/> [Acessado 25 Jan. 2018].

Reactjs.org. (2018). *React - A JavaScript library for building user interfaces.* [online] Available at: <https://reactjs.org/> [Acessado 25 Jan. 2018].

CanvasJS. (2018). *Beautiful HTML5 JavaScript Charts / CanvasJS.* [online] Available at: <https://canvasjs.com/> [Acessado 25 Jan. 2018].

Npmjs.com. (2018). *npm.* [online] Available at: <https://www.npmjs.com/> [Acessado 25 Jan. 2018].