1. ARQUITETURA

A melhor arquitetura para a criação de um software modular.

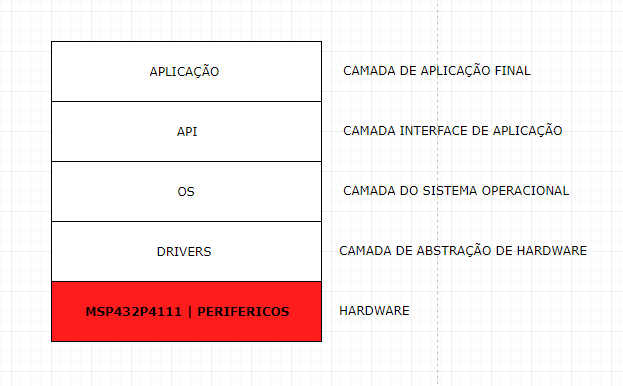
* Possuindo princípios organizacionais claros e explícitos, expressos como arquitetura de produtos de software.
* Seja hierarquicamente organizado em camadas.
* Que cada camada consiste em um conjunto modular de componentes, cada um com sua própria função ou finalidade, e cada um oferecendo uma única interface robusta para outros módulos acessarem essa funcionalidade.

Analisando a primeira definição, se torna um pouco difícil atingi-la, uma vez que sistemas embarcados possuem diferenças muito grandes de uma aplicação para outra. Por exemplo, se o software de voo da Missão X utiliza um computador de bordo com um processador ARM, esse software possivelmente não será compatível com um computador embarcado que tenha como processador AVR. Com isso, essa definição torna-se útil para aplicação em desktop, onde há um sistema operacional fazendo uma interface com o hardware, fazendo com que um software seja utilizado em computares de diferentes configurações.

Já a segunda e a terceira definição é cabível de execução. Dividir a implementação em camadas é um dos .... . E a última, dividindo cada camada em componentes, tornará fácil a fase de implementação e testes. Cumprir a segunda e terceira definição está claro qual é o significado.

O autor Eickhoff (2012) nomeia esse tipo de arquitetura como *static architecture*, pois ela fica estática Dããa.

Sendo assim, para a aplicação do software embarcado para o OBC, será utilizado a seguinte arquitetura:



Abaixo, seguirá uma aplicação sucinta sobre cada camada e sua aplicação.

**HARDWARE:**

A camada de hardware se refere a parte física do computador de bordo. Essa parte foi tratada na seção XXX. Essa camada abrange o microcontrolador, sensores, periféricos, etc;

**CAMADA DE ABSTRAÇÃO DE HARDWARE**

A camada de abstração de hardware, ou HAL (em inglês, *Hardware Abstraction Layer*), é uma camada que permitirá a comunicação entre o sistema operacional e a parte física do sistema. O HAL inclui o *Bootloader,* pacotes oferecidos pelo processador, etc.

Uma das vantagens do microcontrolador MSP432 é o uso da biblioteca *DriverLib* da Texas Instruments. Esse *API* faz a interface entre os registradores, fornecendo um usabilidade boa (Texas Instruments, 2018).

**SISTEMA OPERACIONAL**

Essa camada é composta pelo sistema operacional, que gerenciará as camadas acima. De acordo com o Autor da nasa fodao cubesat e o APendice X, a maioria das missões utilizava como sistema operacional o Linux ou o FreeRTOS.

Como o MSP432P4111 não possui uma capacidade alta para rodar o Linux, escolheu-se o FreeRTOS.

**API**

Essa camada será a interface de alto nível que a arquitetura possuirá. Será implementado as funcionalidades mais primordiais para um OBC. Essas funcionalidades serão extraídas de requisitos iniciais e de algumas missões CubeSat passadas.

**APLICAÇÃO FINAL**

Essa camada será destinada a versões futuras do sistema. Quando os requisitos e o escopo estiver delimitado, novos componentes e rotinas poderão ser implementados.

MEYER, Marc H. ; WEBB, Peter H. . Modular, layered architecture: the necessary foundation for effective mass customisation in software. **International Journal of Mass Customisation**, Boston, USA, 2005. 14-36, p. 23. Disponível em: <http://web.mit.edu/deweck/Public/ESD39/Readings/MatlabCaseStudy.Meyer-Webb-2005.pdf>. Acesso em: 02 jun. 2018.