**ETS Hub**

**Definição de Escopo**

O ETS Hub é uma arquitetura de ecossistema de diversos módulos (outros sistemas), com o objetivo de agregar e combinar diversos sistemas e outras funcionalidades referentes a ETS, além de servir como “playground” de aprendizado para novas turmas de Desenvolvimento de Sistemas ou turmas relacionadas, onde as mesmas podem criar e adicionar módulos (ou incrementar e/ou alterar módulos já existentes) constituídos de Interface Web, API e Banco de Dados ou algum outro formato de projeto equivalente.

**Análise de Requisitos**

RF 001. Todo módulo do ecossistema deve seguir uma das seguintes arquiteturas de módulo:

RF 001.1. Um módulo pode ser constituído de Interface Web, API, Banco de Dados e Sistema Embarcado.

RF 001.2. Um módulo pode ser constituído de Interface Web, API, Banco de Dados.

RF 001.3. Um módulo pode ser constituído de Interface Web e API.

RF 001.4. Um módulo pode ser constituído de API e Banco de Dados.

RF 001.5. Um módulo pode ser constituído de Interface Web.

RF 001.6. Um módulo pode ser constituído de API.

RF 001.7. Um módulo pode ser constituído de Interface Web, API e Sistema Embarcado.

RF 001.8. Um módulo pode ser constituído de API e Sistema Embarcado.

RF 001.9. Um módulo pode ser constituído de API, Banco de Dados e Sistema Embarcado.

RF 001.10. Um módulo pode ser constituído de API e Sistema Embarcado.

RF 002. Para o uso de tecnologias de desenvolvimento, diferentes das usadas por padrão na Bosch, deve-se atender três requisitos principais:

RF 002.1. A Bosch já possui uma licença da tecnologia desejada, se for necessária uma licença.

RF 002.2. Um aval de uso da tecnologia por parte da ICO.

RF 002.3. E uma justificativa válida do porque as tecnologias já existentes não conseguem cumprir com os requisitos do módulo e como a nova tecnologia irá cumprir com os mesmos.

RF 003. Todo modelo de módulo que contenha uma Interface Web, pode conter uma ou mais Interfaces Web utilizando qualquer tecnologia desejada (Angular, React, Blazor, etc...), vide a RF 002.

RF 004. Todo modelo de módulo que contenha uma API, pode conter uma ou mais APIs, se e somente se, as APIs sejam de tecnologias diferentes (C# WebApi, Flask, Django, etc...), vide a RF 002.

RF 005. Todo modelo de módulo que contenha um Banco de Dados, pode conter um ou mais Banco de Dados, se e somente se, os Banco de Dados sejam de tecnologias diferentes (Microsoft Sql Server, MySQL, Kafka, MongoDB, Elastic Search, Radis, etc...), vide a RF 002.

RF 006. Todo modelo de módulo que contenha um Sistema Embarcado, pode conter um ou mais Sistemas Embarcados (Arduino, Esp32, RaspBerry PI, etc...), vide a RF 002.

RF 008. Nenhum Banco de Dados dentro do ecossistema pode ser acessado diretamente por qualquer aplicação, todo acesso a qualquer Banco de Dados dentro do sistema deve ser feito através de uma API, e apenas aquela API terá acesso à aquele Banco de Dados.

RF 009. Apesar da RF 008 limitar o acesso de Banco de Dados a uma única API, a API em si pode acessar qualquer Banco de Dados dentro do mesmo módulo ao qual ela pertence.

RF 010. Todo request entre APIs dentro do ecossistema, inclusive APIs dentro de um mesmo módulo, deve ocorrer através da “API Broker”, uma API com o objetivo de reencaminhar requests entre APIs, para uma maior abstração entre as APIs do ecossistema, facilitando a substituição de qualquer API dentro do ecossistema.

RF 011. Toda API deve possuir testes unitários, com um nível de cobertura de código superior a 90%.

RF 012.

NF 001. Todo módulo do Sistema, incluindo o Hub (Core), deve rodar dentro de um container Docker no servidor.

NF 002. O módulo Core será responsável por manejar as informações e verificações de login do sistema, sendo assim, todos os outros módulos devem contatar o módulo Core para a realização de login ou busca de informações relacionadas aos usuários do sistema ou do usuário logado.

NF 003. Todo módulo do sistema, mais especificamente as APIs destes módulos não podem se comunicar diretamente umas com as outras, sendo necessário a comunicação através de um intermediário Kafka.

NF 004. Toda a API de um módulo deve acessar ou ter acesso a apenas um Banco de Dados, e toda a interação com aquele Banco de Dados deve ocorrer por intermédio da API. Tendo como única exceção à regra são Bancos Externos ao Sistema, como o SAP, LENEL e outros Bancos Bosch.

NF 005. O módulo Core deve possuir um sistema Token semelhante ao JWT, implementado internamente, que será usado para validar todo request feito ao módulo, exceto os requests de login.

NF 006.