**Ecossistema 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, Banco de Dados, Sistemas Embarcados ou algum outro formato de projeto equivalente.

**Análise de Requisitos**

**Regras de Negócio**

RN 001. Toda aplicação dentro do ecossistema, sendo elas Interface Web, API ou Banco de Dados, deve estar contido dentro de um container Docker.

RN 002. Um container Docker deve conter uma, e somente uma, aplicação rodando dentro do mesmo.

RN 003. Todo módulo do ecossistema deve seguir uma das seguintes arquiteturas de módulo:

RN 004.1. Um módulo pode ser constituído de Interface Web, API, Banco de Dados e Sistema Embarcado.

RN 004.2. Um módulo pode ser constituído de Interface Web, API, Banco de Dados.

RN 004.3. Um módulo pode ser constituído de Interface Web e API.

RN 004.4. Um módulo pode ser constituído de API e Banco de Dados.

RN 004.5. Um módulo pode ser constituído de Interface Web.

RN 004.6. Um módulo pode ser constituído de API.

RN 004.7. Um módulo pode ser constituído de Interface Web, API e Sistema Embarcado.

RN 004.8. Um módulo pode ser constituído de API e Sistema Embarcado.

RN 004.9. Um módulo pode ser constituído de Sistema Embarcado.

RN 004.10. Um módulo pode ser constituído de API, Banco de Dados e Sistema Embarcado.

RN 005. Para o uso de tecnologias de desenvolvimento, diferentes das usadas por padrão na Bosch, deve-se atender três requisitos principais:

RN 005.1. A Bosch já possui uma licença da tecnologia desejada, se for necessária uma licença.

RN 005.2. Um aval de uso da tecnologia por parte da ICO.

RN 005.3. 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.

RN 006. 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.

RN 007. 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# Web Api, Flask, Django, etc.), vide a RF 002.

RN 008. 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, Oracle, Kafka, etc.), vide a RF 002.

RN 009. 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.

RN 010. 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.

RN 011. Nenhum Banco de Dados dentro do ecossistema deve ter mais do que uma API que tenha acesso aos dados deste Banco de Dados.

RN 012. Uma API pode acessar qualquer Banco de Dados dentro do mesmo módulo no qual ela se encontra.

RN 013. 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.

RN 014. Toda API deve possuir testes unitários, com um nível de cobertura de código igual ou superior a 90%.