

1 Relatório técnico descritivo sobre OpenIoT

Este trabalho visa a pesquisa e o estudo de um projeto *open source* real e no caso, o projeto escolhido foi o OpenIoT. Serão apresentados dados relevantes como o desenvolvimento do projeto, sua licença, governança e manutenção.

1.1 Descrição do projeto

OpenIoT é um *middleware* de código livre que visa integração de sensores, atuadores e usuários dado o avanço do que é chamado *Internet of Things*, ou seja, dispositivos conectados a internet que podem se comunicar não apenas com usuários mas também com outros dispositivos.

O projeto OpenIoT é, portanto, uma solução *open source* que visa ser uma expansão da computação em nuvem e assim ser a base para a implementação da Internet das Coisas, coletando e processando sensores ao redor do mundo, dado que o termo sensor no ambiente de OpenIoT refere-se tanto aos sensores físicos quanto aos virtuais, ou seja, mídias sociais ou *feed* de notícias. O *middleware* disponibiliza um ambiente de implementação de baixo esforço para um *driver* de um novo sensor. Além disso, o sistema prove métodos de *streaming* dos dados em uma infraestrutura de computação em nuvem, fazer buscas por sensores dinamicamente por meio desta infraestrutura assim como a visualização dos dados obtidos.

2 Caracterização do projeto OpenIoT

2.1 Desenvolvimento

- Existe um local dedicado para o desenvolvimento?

Atualmente é recomendado que o projeto seja feito em uma plataforma Linux, porém as informações necessárias para a realização em ambiente Windows está em construção. Não há um local dedicado específico para o desenvolvimento, no entanto há várias dependências que o desenvolvedor precisa ter para conseguir fazer um módulo para o OpenIoT, por exemplo o *Java Development Kit* assim como os *middlewares* GSN e LSM e suas respectivas dependências. Contudo, a própria página do projeto no GitHub influencia o uso do ambiente de desenvolvimento Eclipse para poder integrar diferentes dependências, como o JBOSS e Maven, com o GSN e LSM em um ambiente já conhecido por muitos desenvolvedores.

- é possível extrair o atual código fonte a partir de um repositório público de código fonte?

A última versão do código está disponível no GitHub

- Quão grande é o código?

São no total 610,474 linhas dado que 420,312 são de código e o restante comentários.

- Quais são as principais linguagens de programação?

Java e XML entre 11 linguagens.

- A utilização do pacote depende de algum outro software proprietário ou de código fonte aberto?

O *middleware* depende de dois pacotes de código aberto, o GNS(*Global Sensor Network*) que é uma infraestrutura para rápida implementação e integração de sensores em uma rede *wireless*. O outro pacote é o LSM(*Linked Stream Middleware*) que possibilita o armazenamento em nuvem dos dados coletados pelos sensores.

- é possível calcular o número de *downloads* ou usuários de uma versão em particular?

Não é possível ter o número de *downloads* do projeto.

2.2 Licença Software Livre

- Quem são os patrocinadores que contribuem para a sustentabilidade do projeto?

O projeto foi idealizado e desenvolvido pelo consórcio OpenIoT, que inclui sete organizações europeias e um centro de pesquisa australiano.

- Quem detém os direitos autorais do código?

O projeto está sob a licença LGPL versão 3.0, como mencionado abaixo, portanto os próprios desenvolvedores são os detentores dos direitos autorais.

- O projeto está sob qual tipo de licença de código aberto?

O projeto está sob a versão LGPL versão 3.0.

- Por que os responsáveis pelo projeto escolheram a licença de código aberto?

A licença LGPL foi escolhida por ser bem vista não apenas pelos desenvolvedores de código aberto, mas também por aqueles no meio empresarial e assim encorajar o uso de verses do *middleware* em plataformas comerciais

2.3 Governança

- Existem quantos desenvolvedores alocados para o projeto?

São 17 desenvolvedores, todos das 8 instituições envolvidas no projeto

- Quantos *committers*, também conhecidos por desenvolvedores que podem realizar mudanças propostas, o projeto possui?

São 25 *committers*

- O que você pode dizer sobre o modelo de governança de código fonte aberto?

O código fonte por ser desenvolvido por uma comunidade apresentam princípios mais democráticos na forma em que são geridos em comparação com os softwares corporativos. Há discussões, eleições e outros mecanismos para escolha de lideranças, delegar tarefas e definir métricas de projeto.

2.4 Manutenção

- Gerenciamento de *releases*: Qual o número e frequência de *releases*?

Há três *releases*.

- Comunicação: Existe um canal de comunicação útil e ativo para a comunidade / suporte ao usuário?

O projeto utiliza dois grupos do Google como listas de email, uma para usuários e outra para desenvolvedores

- Existe um *bugtracker* (rastreamento de bugs) com uma lista de bugs corrigidos/pendentes de correção?

O próprio GitHub possui um controle de bugs reportados pelos *committers*

- Existe um plano de metas para planos futuros? Existe evidência que o plano de metas foi utilizado no passado?

Na página *Web* do projeto, há uma área de *Deliverables and Results* que reporta o que foi feito desde 2012 e o que ainda falta ser feito até o final de 2014

- Existe consultoria comercial, treinamento ou consulta disponível para o projeto? A partir de múltiplos prestadores de serviços?

Não foi possível obter informações que sejam relevantes para responder de maneira concreta.