

Tópicos Avançados em Cidades Inteligentes

Aula 2 – Visão Geral do InterSCity

Pablo Teófilo Durans

Orientado por: Francisco José da Silva e Silva

Laboratório de Sistemas Distribuídos Inteligentes (LSDi)

Universidade Federal do Maranhão (UFMA)

<http://www.lsd.ufma.br>

Agosto de 2018





Sumário

- 1 Introdução
- 2 Instalação
- 3 Desenvolvimento para Plataforma



Sumário

- 1 Introdução
- 2 Instalação
- 3 Desenvolvimento para Plataforma



INCT da Internet do Futuro para Cidades Inteligentes

- **Objetivo:** estabelecimento de uma rede de cooperação em pesquisa e desenvolvimento tecnológico voltada para a temática da Internet do Futuro e Cidades Inteligentes;



INCT da Internet do Futuro para Cidades Inteligentes

- **Objetivo:** estabelecimento de uma rede de cooperação em pesquisa e desenvolvimento tecnológico voltada para a temática da Internet do Futuro e Cidades Inteligentes;
- **Participantes:** USP, Unicamp, UFRJ, PUC-Rio, UFG, UFMS, UFMA, Unifesp e UFABC;



INCT da Internet do Futuro para Cidades Inteligentes

- **Objetivo:** estabelecimento de uma rede de cooperação em pesquisa e desenvolvimento tecnológico voltada para a temática da Internet do Futuro e Cidades Inteligentes;
- **Participantes:** USP, Unicamp, UFRJ, PUC-Rio, UFG, UFMS, UFMA, Unifesp e UFABC;
- InterSCity:
 - InterSCity Plataforma;
 - InterSCSimulator;
 - entre outros.



InterSCity Plataforma

- Prover serviços e API de alto nível para desenvolvimento de aplicações para Cidades Inteligentes
- Baseado em Microserviços
- Restful + Json = Consumidor Independente de Plataforma
- Mozilla Public License 2.0



InterSCity Plataforma - Princípios do Design

- Modularidade via Serviços
- Modelos e Dados Distribuídos
- Evolução Descentralizada
- Reuso de Projetos Open Source
- Adoção de Padrões Abertos
- Assíncrono vs. Síncrono
- Serviços Sem Estado

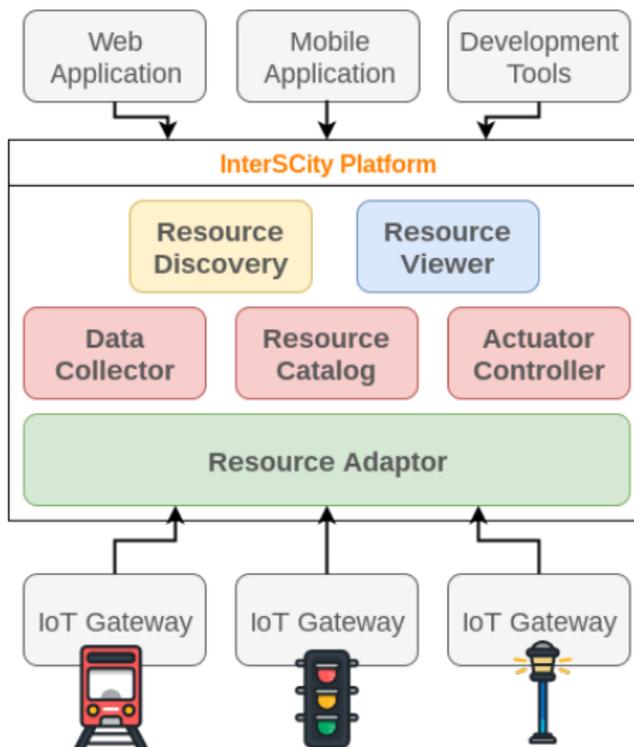


Tecnologias Associadas

- **IoT:** Disponibiliza uma interface para que Gateways IoT forneçam dados de sensores e recebam comandos de atuação;
- **Big Data:** Para estar preparada para receber grande volumes de dados, plataforma utiliza banco de dados NoSQL (MongoDB) e requisições Assíncronas;
- **Cloud:** A arquitetura baseada em Microserviços, utilização de Containers e comunicação via webservice (Restful) tornam o InterSCity Platform adequado para ser executado na nuvem



InterSCity Platform - Arquitetura



InterSCity Plataforma - Arquitetura

- **Resource Adaptor:** Ponto único de contato para os Gateways IoT;
- **Resource Catalog:** Mantém os dados estáticos e localização dos recursos e as capacidades;
- **Data Collector:** Gerencia os dados de contexto;
- **Actuator Controller:** Gerencia os comandos de atuação;
- **Resource Discovery:** Provê filtros para que recursos sejam descobertos



InterSCity Platform - Clientes da Plataforma

Através do desenho da arquitetura, percebe-se que existem 2 dimensões de clientes da plataforma:

- **Gateways IoT:** Responsáveis por coletar dados e realizar atuações nos recursos da cidade;
- **Aplicações:** Implementa funcionalidades ao povo da cidade.



Sumário

1 Introdução

2 Instalação

3 Desenvolvimento para Plataforma



Prerequisitos

- Sistema Operacional Linux
 - Subsistema linux no windows ??
- Git
- Docker (versão 1.13 ou maior)
- Docker Compose (versão 1.10 ou maior)



Download do Projeto

- Fazer clone do dev-env do projeto

\$ git clone

<http://gitlab.com/intercity/intercity-platform/dev-env.git>

- Inicializar os submodulos (microserviços)

\$ git submodule init

- Baixar os submodulos (microserviços)

\$ git submodule update



Configurando e Inicializando a Plataforma

Toda a tarefa de configurar e inicializar a plataforma é feita de forma bastante simples por scripts:

- Entre na pasta dev-env
\$ cd dev-env/
- Construa os Containers Docker
\$./project setup
- Crie a rede platform no docker
\$ docker network create platform
- Inicie os Microserviços
\$./project start



Controlando os Microserviços

- It's works? – Acessar no navegador:
`http://localhost:8000/catalog/capabilities`
- Os Microserviços podem ser parados com o comando `./project stop`
- Cada Microserviço possui um executável `script/development` utilizado para iniciar e parar microserviços individualmente.
Ex: `.../resource-adaptor/scripts/development start`
- Os containers que estão em execução podem ser listados com:
`$ docker ps`



Sumário

- 1 Introdução
- 2 Instalação
- 3 Desenvolvimento para Plataforma



Documentação

- Site do INCP: <http://intercity.org/>
- Repositório do gitlab:
<https://gitlab.com/intercity/intercity-platform/>
 - **docs**: Informações Gerais sobre a plataforma;
 - **dev-env**: Instalação do InterSCity Platform para desenvolvimento;
 - No diretório de cada microserviço, possui uma pequena descrição no arquivo README.md
- API: <http://playground.intercity.org/>



Bibliotecas

Pode ser utilizada qualquer plataforma e/ou linguagem para desenvolver aplicações e gateways IoT para o Interscity, mas é necessário bibliotecas para:

- **Cliente Restful**

Java: Unirest

Ruby: gem rest-client

Android: Retrofit

- **Json Parser**

Java: JSON-java

Ruby: gem json

Android: Gson



Ferramentas Auxiliares

- REST client para debug: Insomnia
- Formatador de Json:
<https://jsonformatter.curiousconcept.com/>
- Controlar Container Docker: Eclipse Docker Tooling

