



Desenvolvimento de Software Dirigido a Modelos - 2021/2

Eduardo Oliveira – Arquitetura IoT RILA*

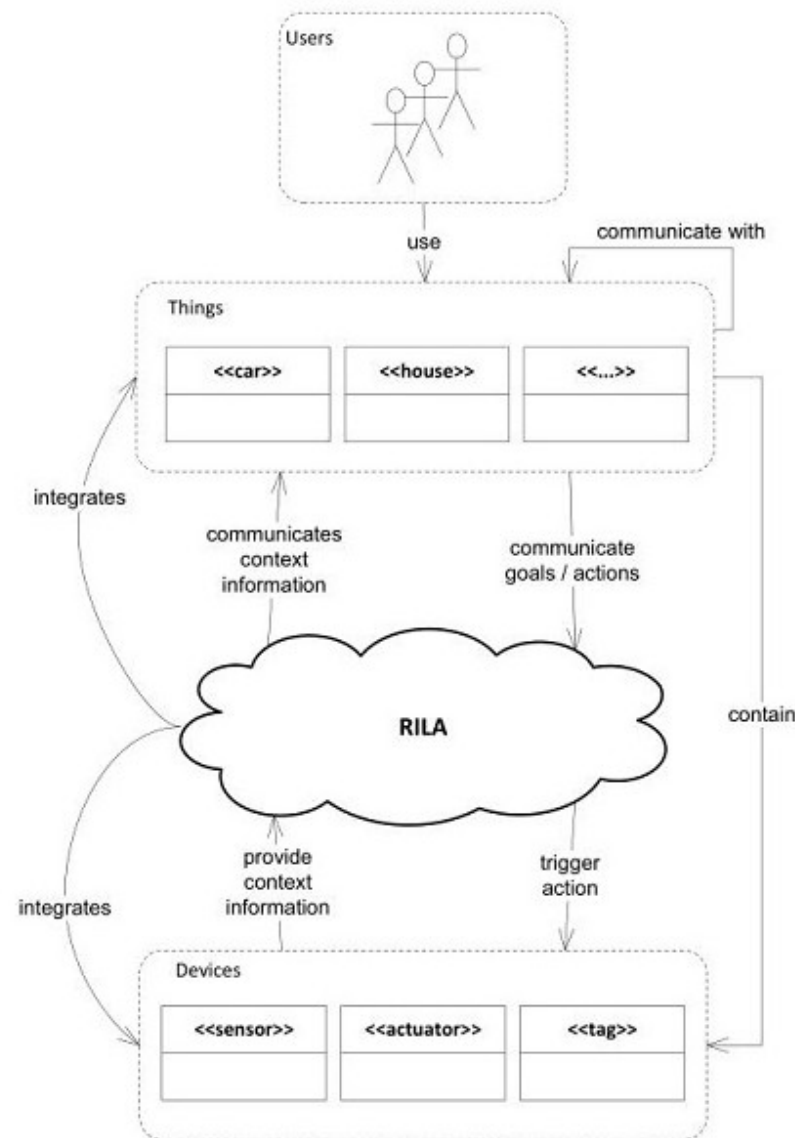
** Reference IoT Layered Architecture*



Agenda

- Domínio escolhido
- Definição da Linguagem Concreta
- Definição do PIM
- Definição do PSM
- XMI do PIM
- M2M (PIM \rightarrow PSM) - ATL
- XMI do PSM
- M2T - Acceleo
- Conclusões
- Trabalhos Futuros

Domínio Escolhido – RILA

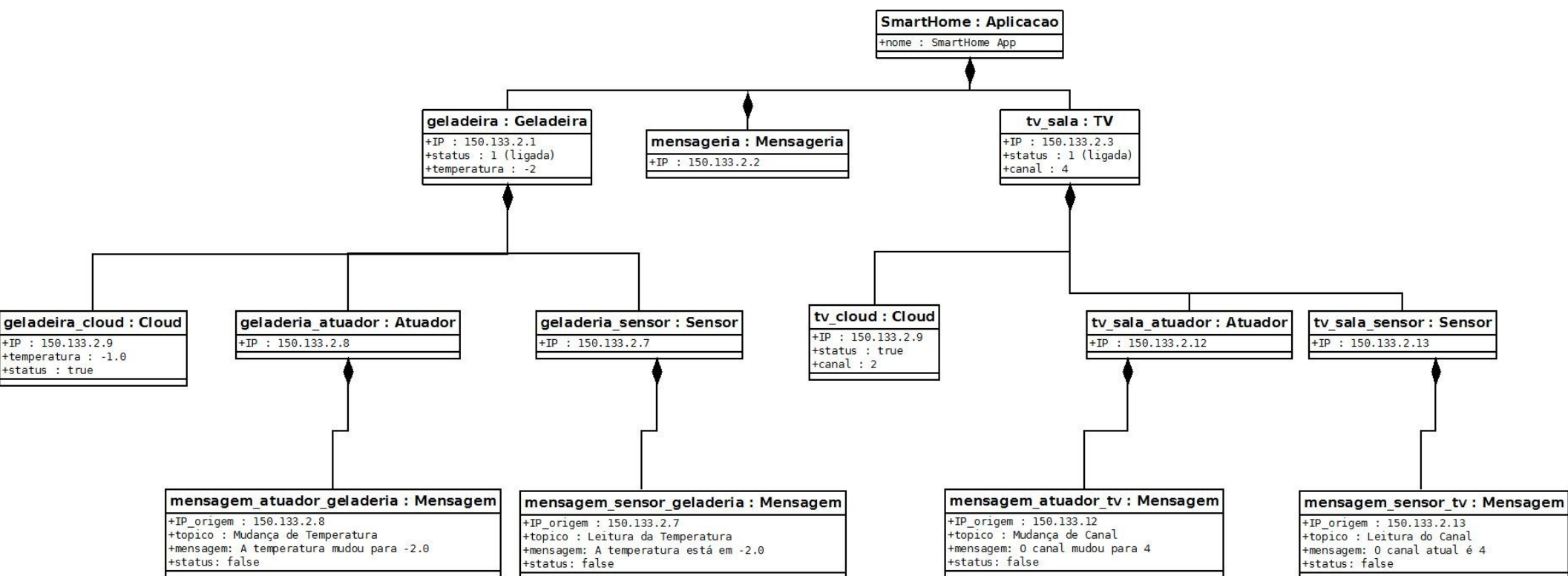


"Reference IoT Layered Architecture"

<https://architecturez.net/pst/az-cf-176871-1454077830>

Conceitos extraídos

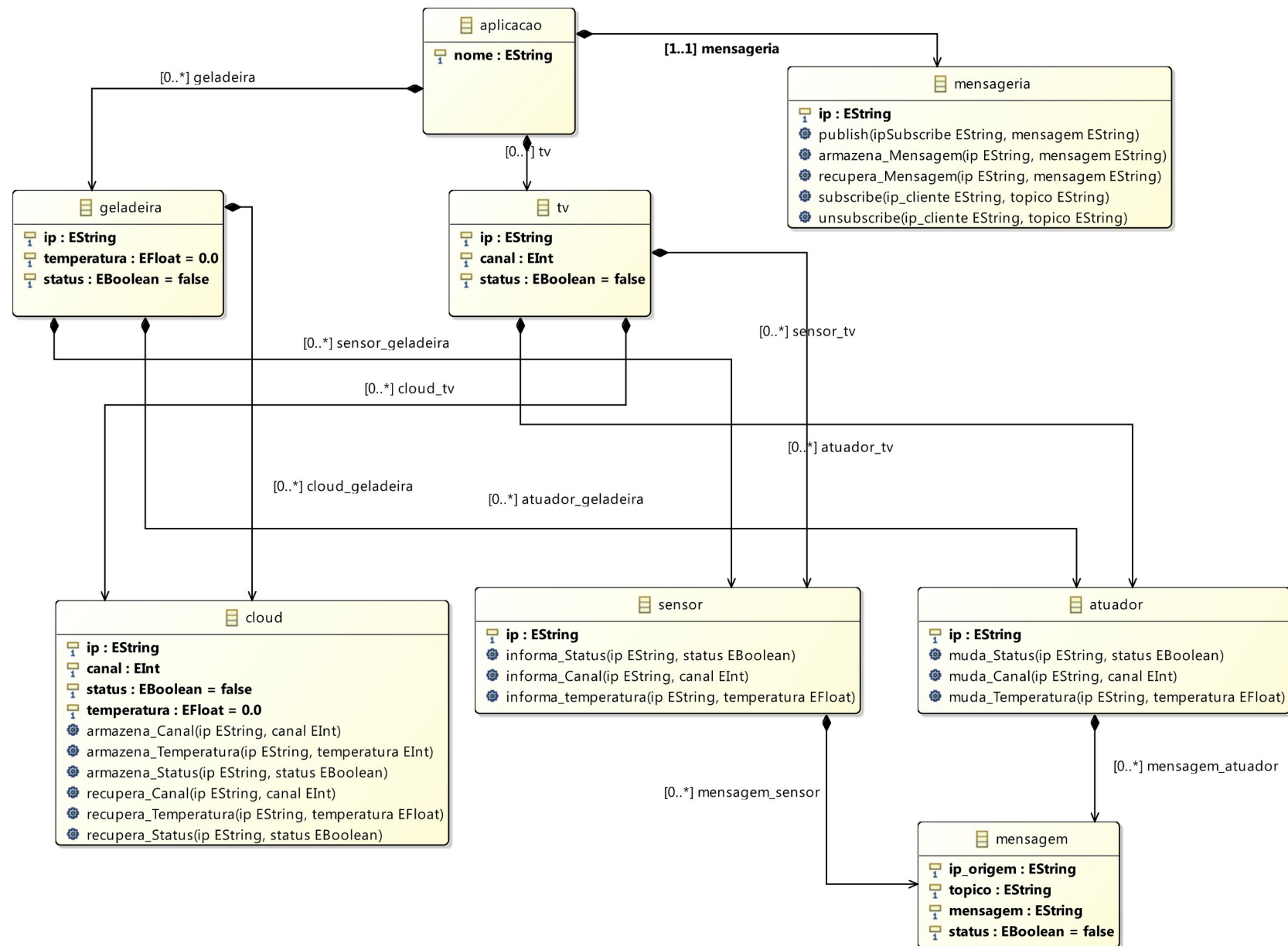
Diagrama de Objetos



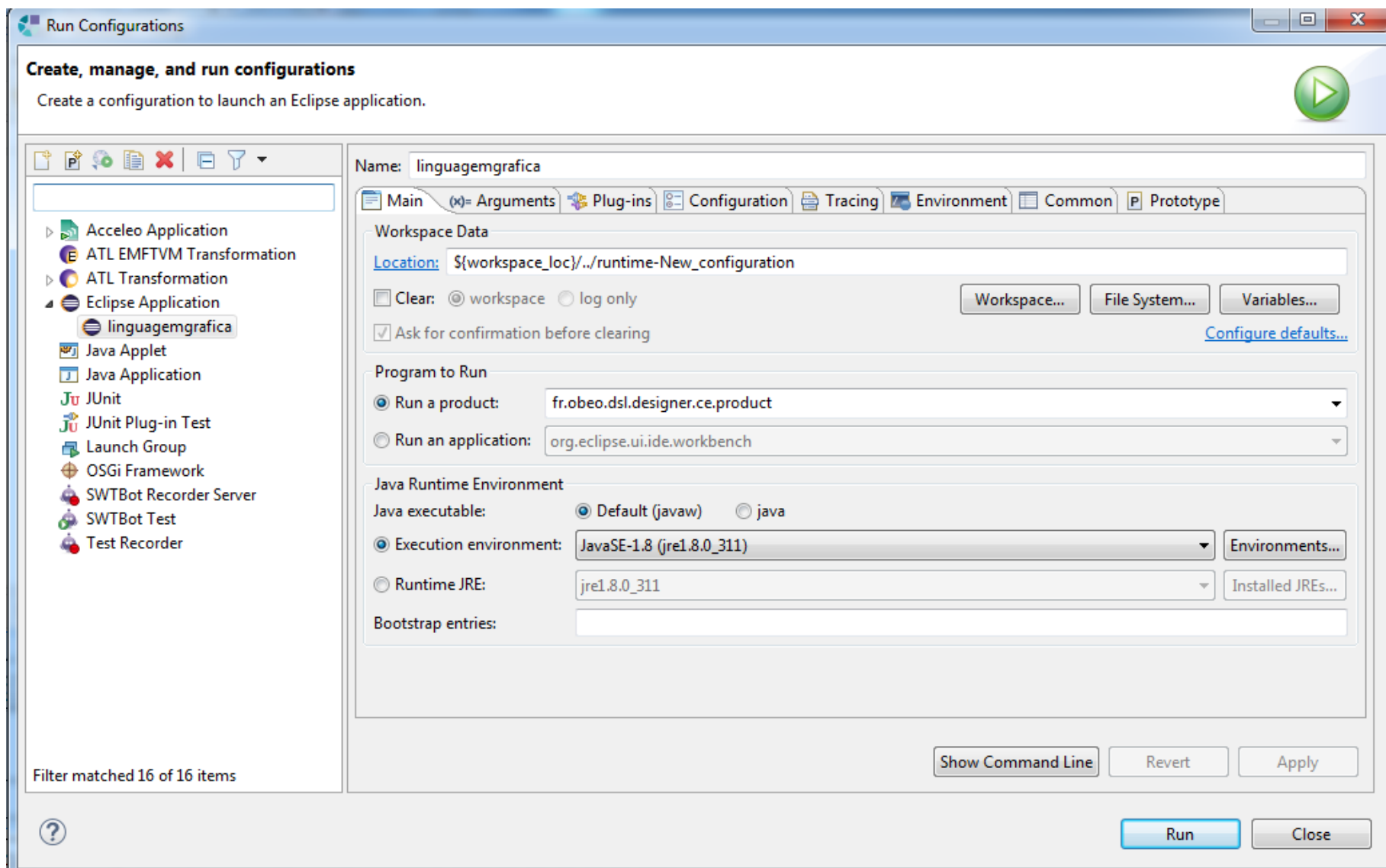
Conceitos extraídos

Conceito (classes)	Propriedades Intrínseca (atributos)	Propriedade Extrínseca (associações entre as classes)
Aplicação	Nome: String	
Geladeira	Ip: String Status: Boolean Temperatura: Float	Associada a classe aplicação.
TV	Ip: String Status: Boolean Canal: Int	Associada a classe aplicação.
Mensageria	Ip: String	Associada a classe aplicação.
Cloud	Ip: String Status: Boolean Temperatura: Float Canal: Int	Associada as classes Tv e Geladeira.
Sensor	Ip: String	Associada as classes Tv e Geladeira.
Atuador	Ip: String	Associada as classes Tv e Geladeira.
Mensagem	Ip_origem: String Topico: String Mensagem: String Status: Boolean	Associada as classes Sensor e Atuador.

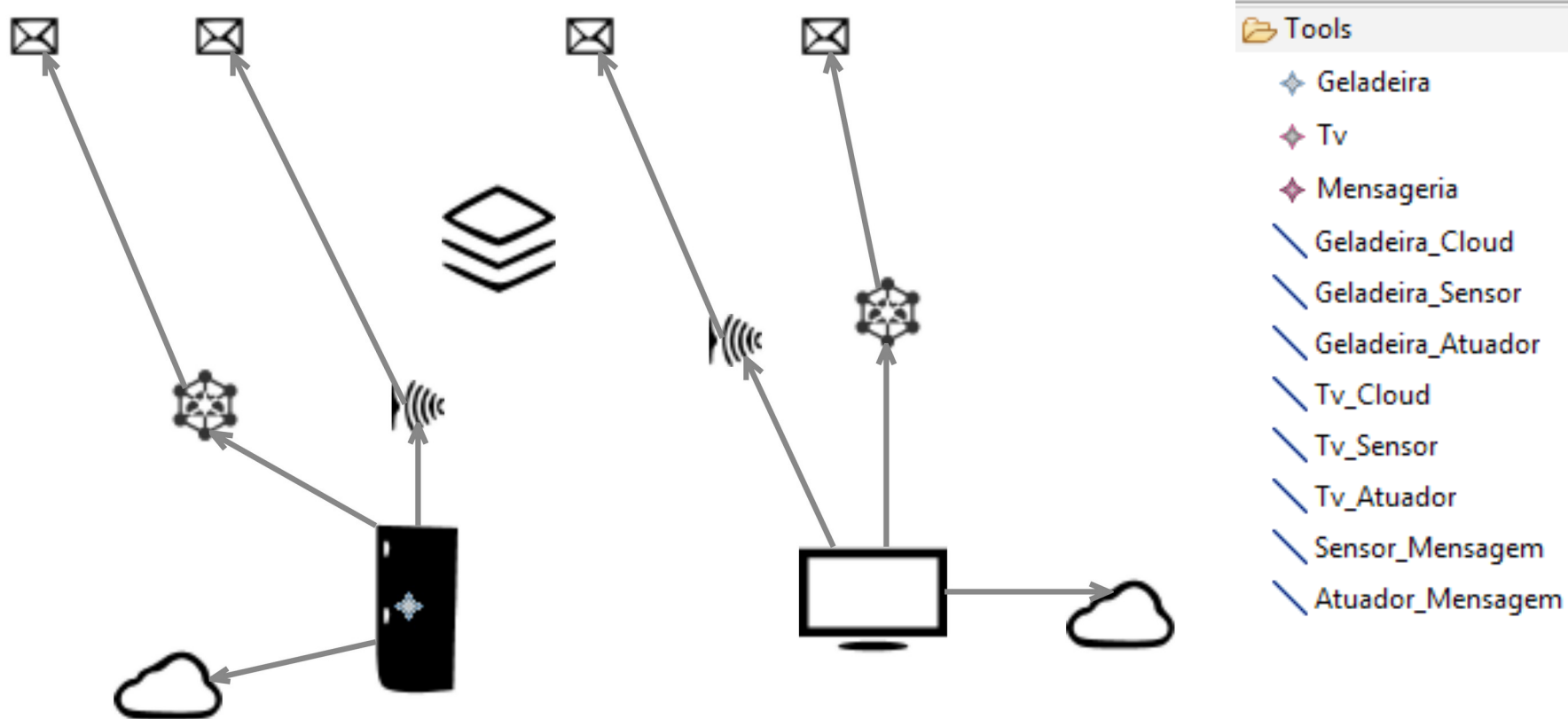
Diagrama de Classes da DSML



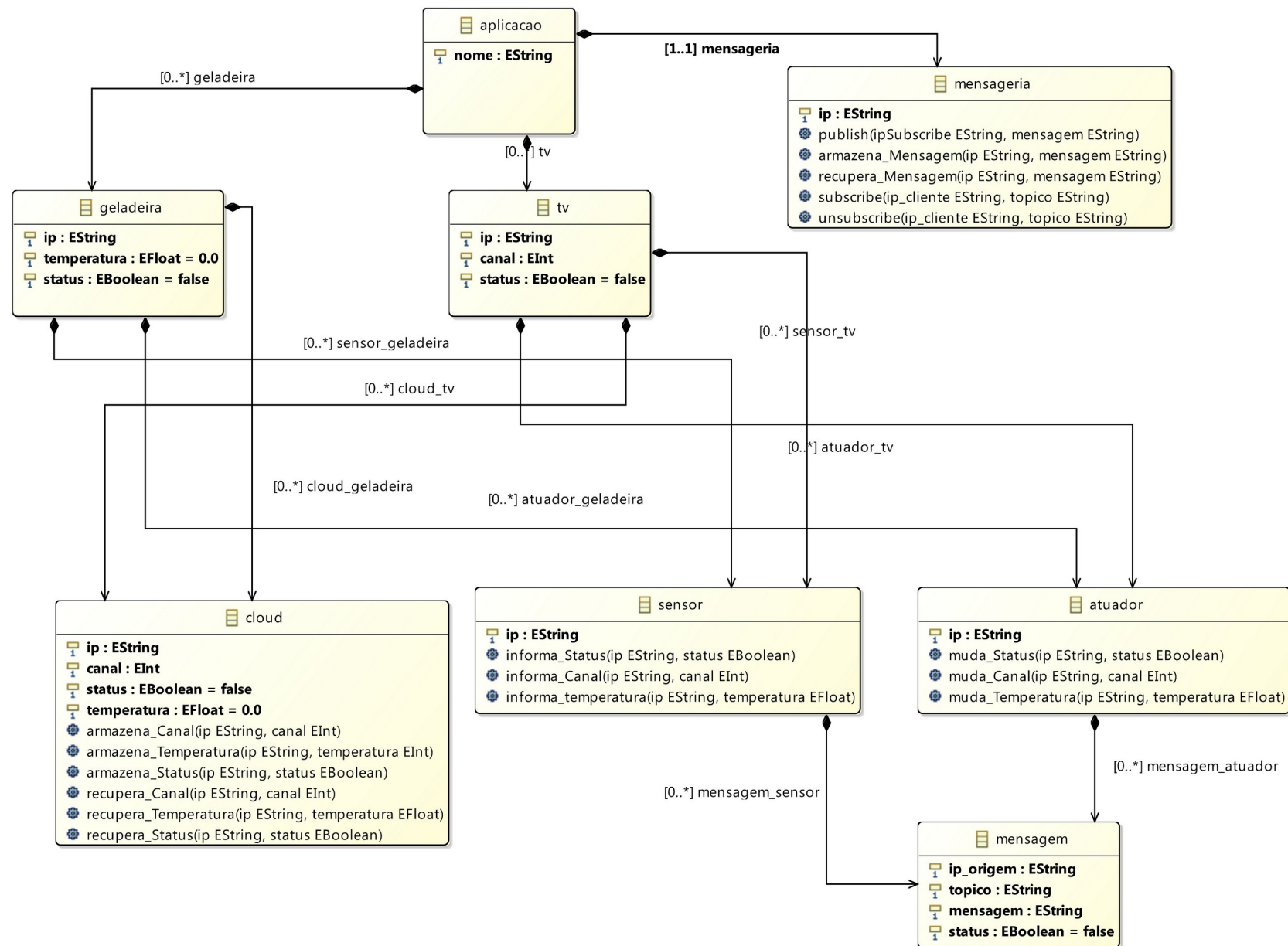
Configuração – Linguagem Concreta



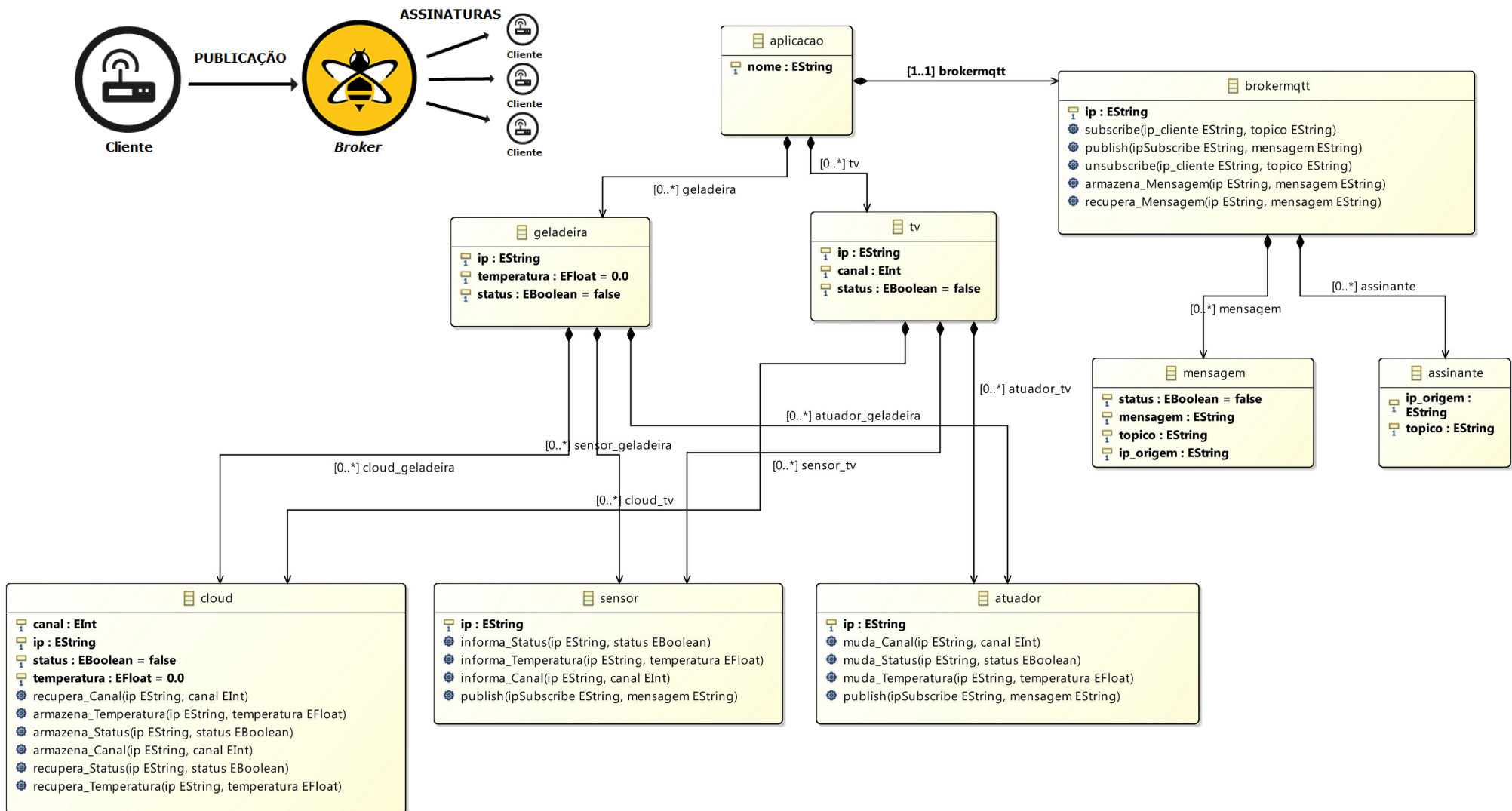
Definição da Linguagem Concreta



Definição do PIM – Metamodelo



Definição do PSM – Metamodelo com protocolo MQTT



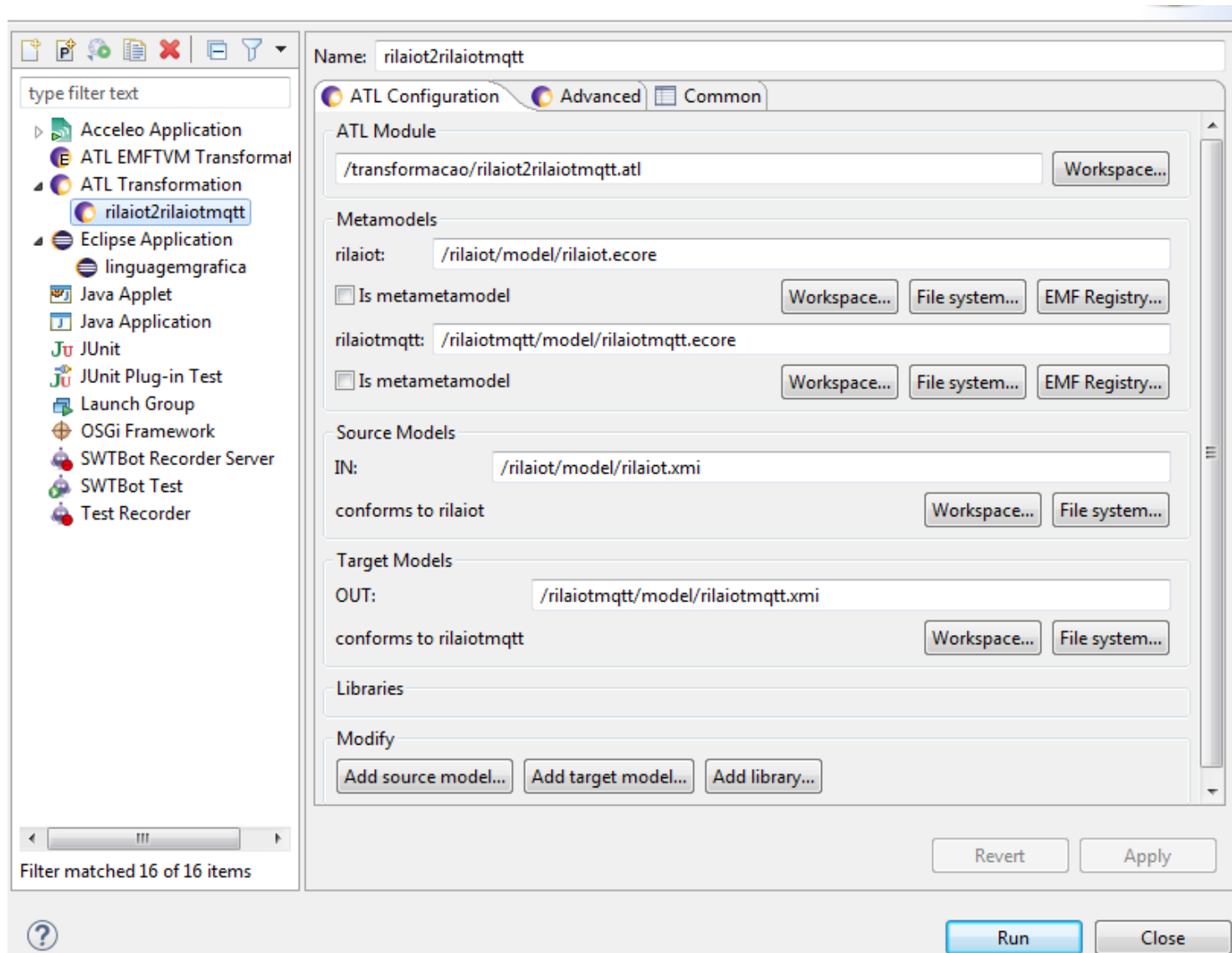
XMI do PIM

platform:/resource/rilaiot/model/rilaiot.xmi

- ◆ Aplicacao SmartHome
 - ◆ Geladeira 123.123.1
 - ◆ Cloud 123.123.4
 - ◆ Atuador 123.123.5
 - ◆ Mensagem 123.123.5
 - ◆ Sensor 123.123.6
 - ◆ Mensagem 123.123.6
 - ◆ Tv 123.123.2
 - ◆ Cloud 123.123.7
 - ◆ Atuador 123.123.8
 - ◆ Mensagem 123.123.8
 - ◆ Sensor 123.123.9
 - ◆ Mensagem 123.123.9
 - ◆ Mensageria 123.123.10

```
<?xml version="1.0" encoding="UTF-8"?>
<rilaiot:aplicacao
  xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:rilaiot="http://www.example.org/rilaiot"
  nome="SmartHome">
  <geladeira
    ip="123.123.1">
    <cloud_geladeira
      ip="123.123.4"/>
    <atuador_geladeira
      ip="123.123.5">
      <mensagem_atuador
        ip_origem="123.123.5"
        topico="Mudança de temperatura"
        mensagem="Mudando a temperatura para 1.0"/>
      </atuador_geladeira>
    <sensor_geladeira
      ip="123.123.6">
      <mensagem_sensor
        ip_origem="123.123.6"
        topico="Leitura da temperatura"
        mensagem="Lendo a temperatura em 0.0"/>
      </sensor_geladeira>
    </geladeira>
  <tv ip="123.123.2"
    canal="2">
    <cloud_tv
      ip="123.123.7"
      canal="2"/>
    <atuador_tv
      ip="123.123.8">
      <mensagem_atuador
        ip_origem="123.123.8"
        topico="Mudança de canal"
        mensagem="Mudando para o canal 8"/>
      </atuador_tv>
    <sensor_tv
      ip="123.123.9">
      <mensagem_sensor
        ip_origem="123.123.9"
        topico="Leitura do canal"
        mensagem="Lendo o canal 2"/>
      </sensor_tv>
    </tv>
  <mensageria
    ip="123.123.10"/>
  </rilaiot:aplicacao>
```

Configuração – ATL



Regras da ATL - M2M

```
-- @path rilaiot=/rilaiot/model/rilaiot.ecore
-- @path rilaiotmqtt=/rilaiotmqtt/model/rilaiotmqtt.ecore

module rilaiot2rilaiotmqtt;
create OUT : rilaiotmqtt from IN : rilaiot;

helper def: getMensagens(rila:rilaiot!mensagem) :
Sequence(rilaiotmqtt!mensagem)
=
rilaiotmqtt!mensagem.allInstances()->iterate(men;
conjunto:
Sequence(rilaiotmqtt!mensagem) = Sequence{}|
conjunto -> append (men)

);

helper def: getAssinantes(rila:rilaiot!mensagem) :
Sequence(rilaiotmqtt!assinante)
=
rilaiotmqtt!assinante.allInstances()->iterate(ass;
conjunto:
Sequence(rilaiotmqtt!assinante) = Sequence{}|
conjunto -> append (ass)

);
```

```
rule aplicacao2aplicacao {
from
a : rilaiot!aplicacao
to
a1 : rilaiotmqtt!aplicacao (
nome <- a.nome,
tv <- a.tv,
geladeira <- a.geladeira,
brokermqtt <- a.mensageria
)
}

rule tv2tv {
from
a : rilaiot!tv
to
a1 : rilaiotmqtt!tv (
ip <- a.ip,
canal <- a.canal,
status <- a.status,
sensor_tv <- a.sensor_tv,
atuador_tv <- a.atuador_tv,
cloud_tv <- a.cloud_tv
)
}
```

Regras da ATL - M2M

```
rule sensortv2sensortv {
  from
  a : rilaiot!sensor
  to
  a1 : rilaiotmqtt!sensor (
    ip <- a.ip
  )
}

rule atuadortv2atuadortv {
  from
  a : rilaiot!atuador
  to
  a1 : rilaiotmqtt!atuador (
    ip <- a.ip
  )
}

rule cloudtv2cloudtv {
  from
  a : rilaiot!cloud
  to
  a1 : rilaiotmqtt!cloud (
    ip <- a.ip,
    canal <- a.canal,
    temperatura <- a.temperatura,
    status <- a.status
  )
}
```

```
rule geladeira2geladeira {
  from
  a : rilaiot!geladeira
  to
  a1 : rilaiotmqtt!geladeira (
    ip <- a.ip,
    temperatura <- a.temperatura,
    status <- a.status,
    sensor_geladeira <- a.sensor_geladeira,
    atuador_geladeira <- a.atuador_geladeira,
    cloud_geladeira <- a.cloud_geladeira
  )
}

rule aplicacao2brokermqtt {
  from
  a : rilaiot!mensageria
  to
  a1 : rilaiotmqtt!brokermqtt (
    ip <- a.ip,
    mensagem <- thisModule.getMensagens(a),
    assinante <- thisModule.getAssinantes(a)
  )
}

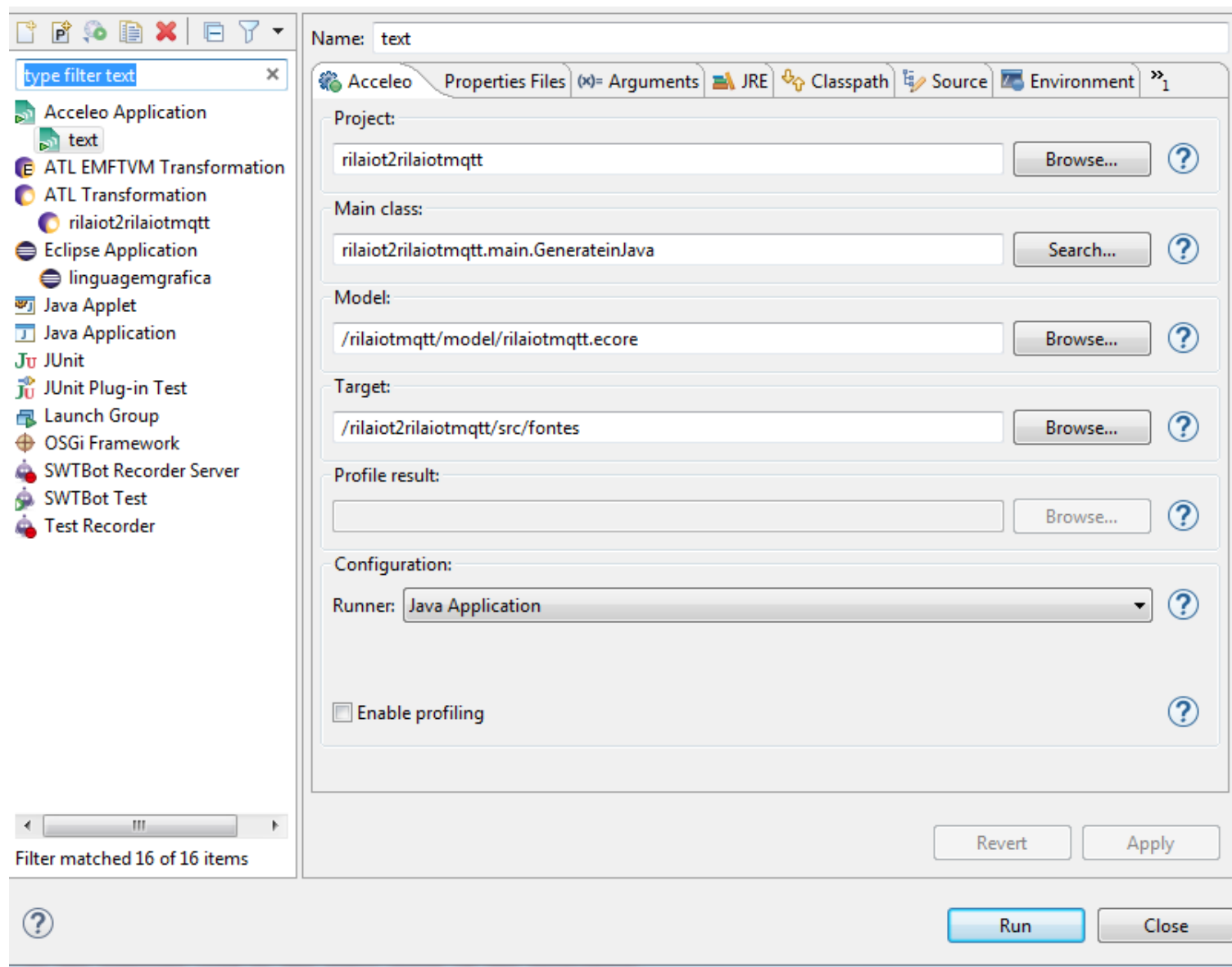
rule mensagem2mensagem {
  from
  a : rilaiot!mensagem
  to
  a1 : rilaiotmqtt!mensagem (
    ip_origem <- a.ip_origem,
    topico <- a.topico,
    mensagem <- a.mensagem,
    status <- a.status
  ),
  a2: rilaiotmqtt!assinante (
    ip_origem <- a.ip_origem,
    topico <- a.topico
  )
}
```

XMI do PSM

- platform:/resource/rilaiotmqtt/model/rilaiotmqtt.xmi
 - Aplicacao SmartHome
 - Brokermqtt 123.123.10
 - Mensagem Mudando a temperatura para 1.0
 - Mensagem Lendo a temperatura em 0.0
 - Mensagem Mudando para o canal 8
 - Mensagem Lendo o canal 2
 - Assinante 123.123.5
 - Assinante 123.123.6
 - Assinante 123.123.8
 - Assinante 123.123.9
 - Tv 123.123.2
 - Cloud 123.123.7
 - Sensor 123.123.9
 - Atuador 123.123.8
 - Geladeira 123.123.1
 - Cloud 123.123.4
 - Sensor 123.123.6
 - Atuador 123.123.5

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rilaiotmqtt:aplicacao xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:rilaiotmqtt="http://www.example.org/rilaiotmqtt"
nome="SmartHome">
  <brokermqtt ip="123.123.10">
    <mensagem mensagem="Mudando a temperatura para 1.0"
topico="Mudança de temperatura" ip_origem="123.123.5"/>
    <mensagem mensagem="Lendo a temperatura em 0.0"
topico="Leitura da temperatura" ip_origem="123.123.6"/>
    <mensagem mensagem="Mudando para o canal 8"
topico="Mudança de canal" ip_origem="123.123.8"/>
    <mensagem mensagem="Lendo o canal 2" topico="Leitura
do canal" ip_origem="123.123.9"/>
    <assinante ip_origem="123.123.5" topico="Mudança de
temperatura"/>
    <assinante ip_origem="123.123.6" topico="Leitura da
temperatura"/>
    <assinante ip_origem="123.123.8" topico="Mudança de
canal"/>
    <assinante ip_origem="123.123.9" topico="Leitura do
canal"/>
  </brokermqtt>
  <tv ip="123.123.2" canal="2">
    <cloud_tv canal="2" ip="123.123.7"/>
    <sensor_tv ip="123.123.9"/>
    <atuador_tv ip="123.123.8"/>
  </tv>
  <geladeira ip="123.123.1">
    <cloud_geladeira ip="123.123.4"/>
    <sensor_geladeira ip="123.123.6"/>
    <atuador_geladeira ip="123.123.5"/>
  </geladeira>
</rilaiotmqtt:aplicacao>
```

Configuração Acceleo



Acceleo – geração do código Java

```
[comment encoding = UTF-8 /]
[module generateinJava('http://www.example.org/rilaiotmqtt')]

[template public generateElement(raiz : rilaiotmqtt::aplicacao)]
[comment @main/]

[for (i : rilaiotmqtt::geladeira | raiz.geladeira)]
[file ('fontes/'+i.ip+'.java', false, 'UTF-8')]
    package fontes;
    public class [i.ip/]{
[/file]
[file ('fontes/'+i.status+'.java', false, 'UTF-8')]
    package fontes;
    public class [i.status/]{
[/file]
[file ('fontes/'+i.temperatura+'.java', false, 'UTF-8')]
    package fontes;
    public class [i.temperatura/]{
[/file]
[/for]

[/template]
```

Acceleo – erro para rodar o código Acceleo

The generation failed to generate any file because there are no model elements that matches at least the type of the first parameter of one of your main templates. The problem may be caused by a problem with the registration of your metamodel, please see the method named "registerPackages" in the Java launcher of your generator. It could also come from a missing [comment @main/]

in the template used as the entry point of the generation.

```
/**
 * This can be used to update the resource set's package registry with all needed EPackages.
 *
 * @param resourceSet
 *      The resource set which registry has to be updated.
 * @generated NOT
 */
@Override
public void registerPackages(ResourceSet resourceSet) {
    super.registerPackages(resourceSet);

    if (!isInWorkspace(rilaiotmqtt.RilaiotmqttPackage.class)) {
        resourceSet.getPackageRegistry().put(rilaiotmqtt.RilaiotmqttPackage.eINSTANCE.getNsURI(),
rilaiotmqtt.RilaiotmqttPackage.eINSTANCE);
    }

/**
 * This can be used to update the resource set's resource factory registry with all needed factories.
 *
 * @param resourceSet
 *      The resource set which registry has to be updated.
 * @generated NOT
 */
@Override
public void registerResourceFactories(ResourceSet resourceSet) {
    super.registerResourceFactories(resourceSet);

    if (!isInWorkspace(rilaiotmqtt.RilaiotmqttPackage.class)) {
        resourceSet.getPackageRegistry().put(rilaiotmqtt.RilaiotmqttPackage.eINSTANCE.getNsURI(),
rilaiotmqtt.RilaiotmqttPackage.eINSTANCE);
    }
}
```



Conclusões

- Muita dificuldade para usar o Sirius;
- Falta de experiência prévia no ambiente Eclipse e linguagem Java dificultaram bastante o andamento do trabalho;
- Muito tempo gasto com retrabalho na reconstrução dos metamodelos e reinstalação do ambiente (problemas de infraestrutura);
- Pouca documentação/tutoriais disponíveis na internet.



Trabalhos Futuros

- Aprimorar o conhecimento no Sirius;
- Obter mais experiência no ambiente Eclipse e na linguagem Java;
- Evoluir o metamodelo Rila para atender a mais requisitos da arquitetura.