

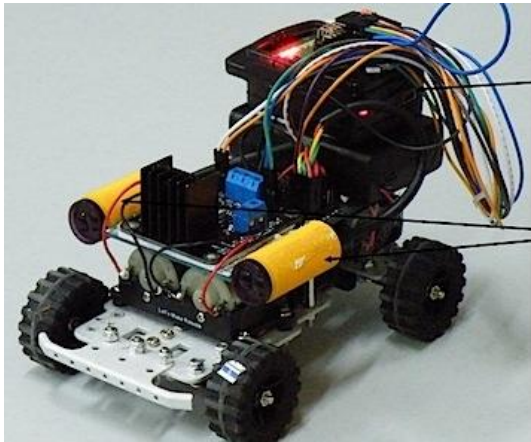
Mãos a Obra

Integrando dispositivos Inteligentes à Web



Introdução a Plataforma Sun SPOT

- **Plataforma de sensores desenvolvida pela Sun Microsystems/Oracle**
 - Diversas possibilidades de uso
 - Ex: Controle de robôs, sensoriamento de eventos físicos, etc.



Introdução a Plataforma Sun SPOT

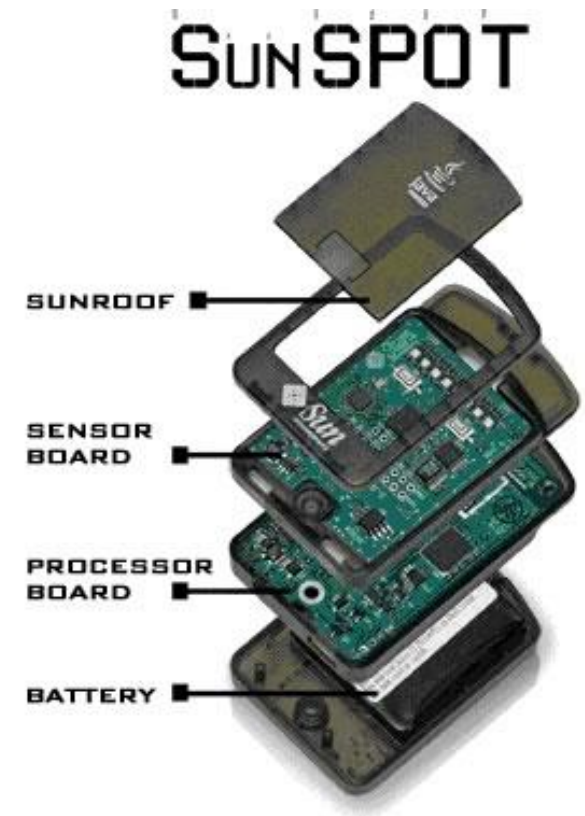
- **Existem 2 tipos de dispositivos:**
 1. *Free-range* SPOT (ou simplesmente SPOT)
 2. Estação base (EB)



Introdução a Plataforma Sun SPOT

- ***Free-range SPOT***

- Bateria recarregável
- Interface USB
- Sensores de:
 - Temperatura,
 - Luminosidade e
 - Acelerômetro nos três eixos
- 3 botões
 - Um botão para ligar, desligar e reiniciar os SPOTs
 - Dois botões que podem ser utilizados para gerar entradas nas aplicações ocasionadas pelo evento clique



Introdução a Plataforma Sun SPOT

- ***Free-range SPOT*** (continuação)
 - 8 leds de três cores (vermelho, verde e azul)
 - 1 power led e 1 activity led
 - 4 pinos genéricos de entrada e saída
 - 4 pinos de saída de alta tensão
 - Processador AMR 32 bits
 - 1MB de RAM
 - 8MB de *flash*



Introdução a Plataforma Sun SPOT

- ***Free-range SPOT*** (continuação)
 - Comunicação IEEE 802.15.4
 - Protocolo de roteamento LQRP
 - Programado em Java
 - Máquina Virtual chamada *Squawk Machine*
 - O código é executado sem a necessidade de Sistema Operacional



Introdução a Plataforma Sun SPOT

- **Estação Base**

- Comunicação via rádio padrão IEEE 802.15.4
- Interface USB
- Funciona ligada a outro dispositivo com maior poder computacional
 - Ex: Um computador

O termo “estação base” pode ser utilizado para o SPOT que não é *free-range* ou para o conjunto SPOT + computador.



Introdução a Plataforma Sun SPOT

- **Desenvolvendo para Sun SPOT**
 - Requisitos
 - JRE, Ant (opcional) e SDK instalados com as variáveis de ambiente do sistema operacional devidamente configuradas
 - Maiores detalhes e download em www.sunspotworld.com
 - Utilizando a IDE Netbeans
 - Mais fácil de escrever e de implantar código nos SPOTs
 - Pode ser baixada juntamente com o SDK

Introdução a Plataforma Sun SPOT

- **Desenvolvendo para Sun SPOT (continuação)**
 - A classe principal deve estender *Middlet*
 - Três métodos abstratos devem ser sobrescritos: *startApp()*, *pauseApp()* e *destroyApp(boolean unconditional)*
 - Aplicações de demonstração
 - Podem ser usadas como base para o desenvolvimento das suas aplicações
 - Várias aplicações mostrando como usar todos os recursos dos SPOTs
 - Acompanham o SDK baixado da página do Sun SPOT

Introdução a Plataforma Sun SPOT

- **Desenvolvendo para Sun SPOT (continuação)**

```
/**imports*/
public class Exemplo extends MIDlet {
    public static DatagramConnection conn = null;
    public static int maxlen = 0;

    protected void startApp() throws MIDletStateChangeException {
        try {
            conn = (UDPConnection) Connector.open("udp://:" + 100);
            int response = 0;
            ILightSensor lightSensor = (ILightSensor)
                Resources.lookup(ILightSensor.class);
            UDPDatagram dg = (UDPDatagram) conn.newDatagram(conn.getMaximumLength());
            response = lightSensor.getValue();
            dg.write(response);
            conn.send(dg);
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    protected void pauseApp() {}
    protected void destroyApp(boolean unconditional) throws MIDletStateChangeException {}
} //fim da classe Exemplo
```

Integrando Dispositivos a Web

Exemplos práticos utilizando a
plataforma Sun SPOT



Integrando RSSF a Web

A integração de RSSF com a Web é feita de duas formas:

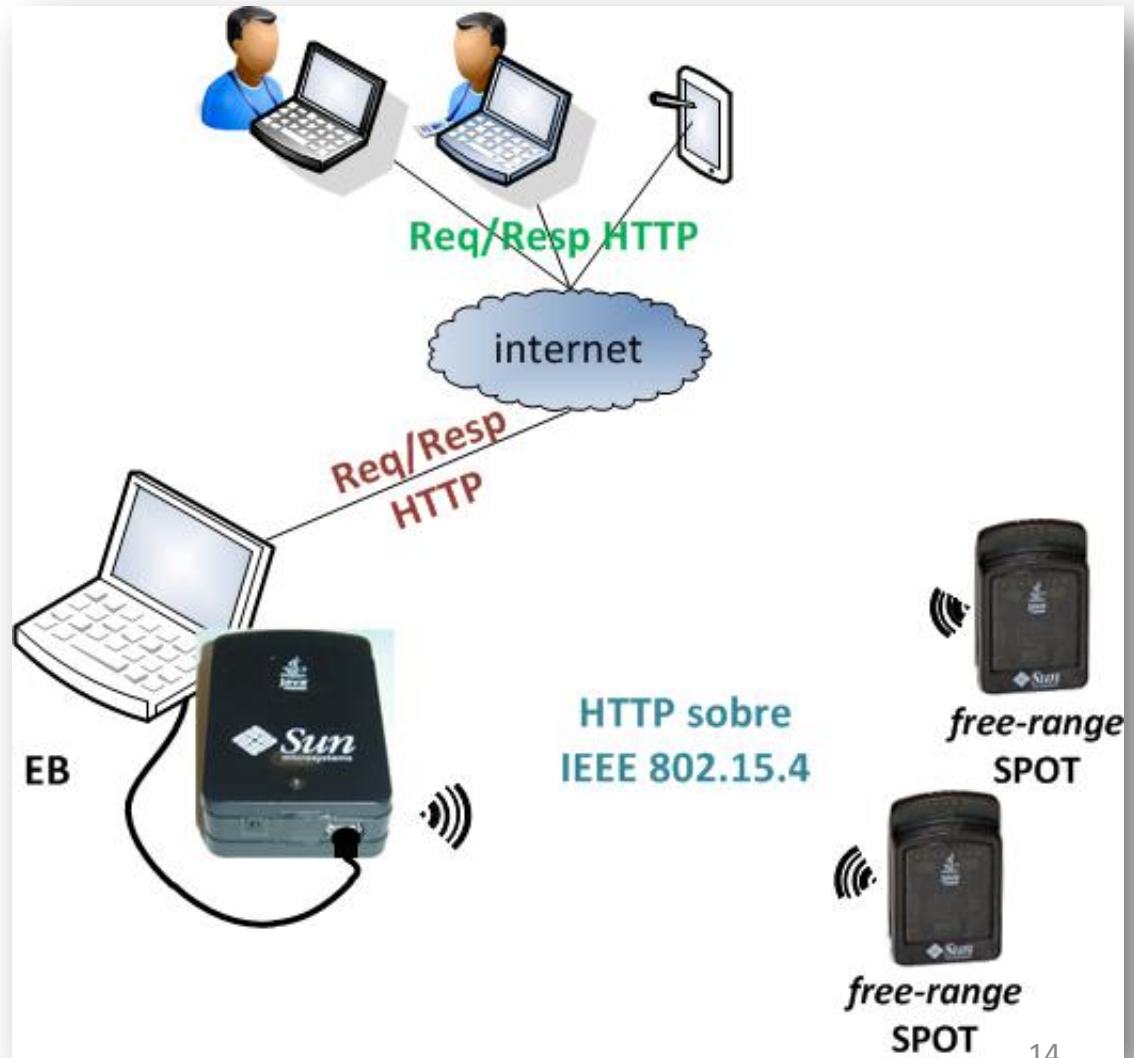
1. Empregando um **Gateway (proxy) + servidor embarcado** no dispositivo o qual fornece uma interface RESTful para integrá-lo a Web, ou
2. Utilizando um **Smart Gateway** (*instalado na EB*) que se comunica com os dispositivos através de suas APIs e protocolos fornecendo uma interface RESTful de acesso as funcionalidades desses dispositivos

Primeira Abordagem de Integração de Dispositivos de RSSF

Utilizando um Gateway + servidor embarcado nos SPOTs

Cenário da Infraestrutura de Comunicação

- A EB recebe as requisições HTTP da Web e as repassa para o SPOT
- Os SPOTs devem tratar as requisições HTTP, enviando a resposta para EB que envia essa resposta para o cliente que enviou a requisição



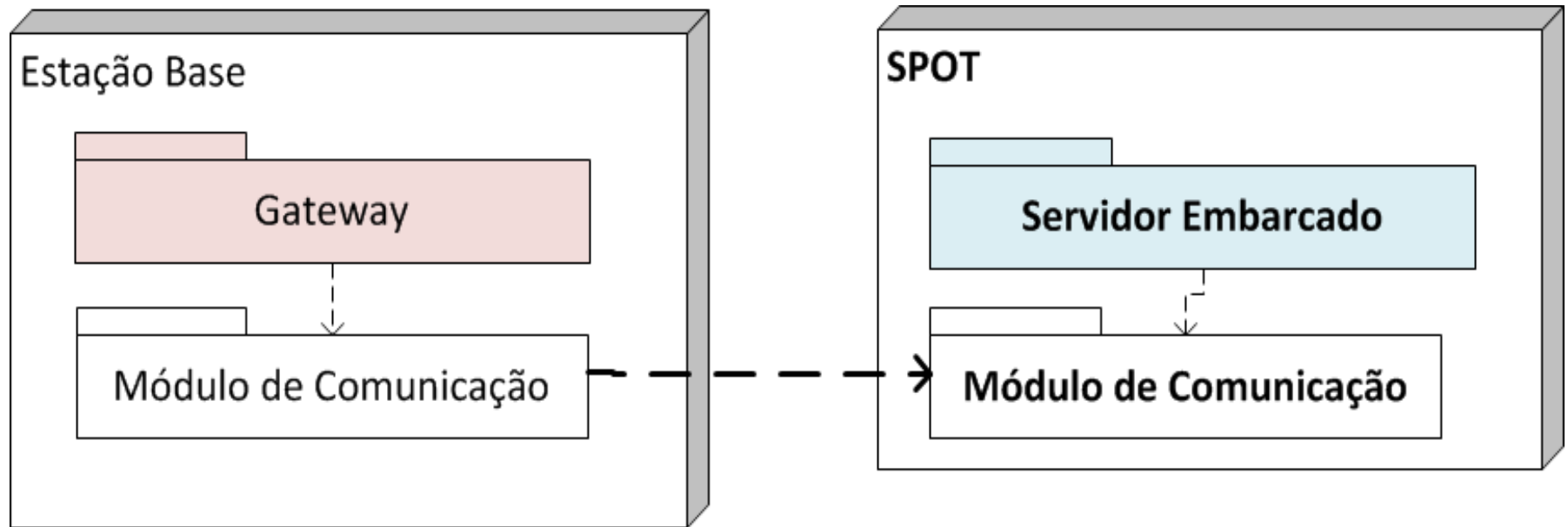
Serviços fornecidos pela EB e pelo SPOT

- **A EB atua como um *gateway* entre a Internet e a rede IEEE 802.15.4**
 - A tarefa fundamental do gateway é repassar a requisição HTTP para o SPOT
- **Os recursos do SPOT são disponibilizados através de uma interface RESTful**
 - O dispositivo tem um servidor embarcado
 - O qual expõe a interface RESTful
 - O SPOT trata as requisições HTTP e gera respostas adequadas para essas requisições

Projeto *WebOfThings* que acompanha a nova versão do SDK

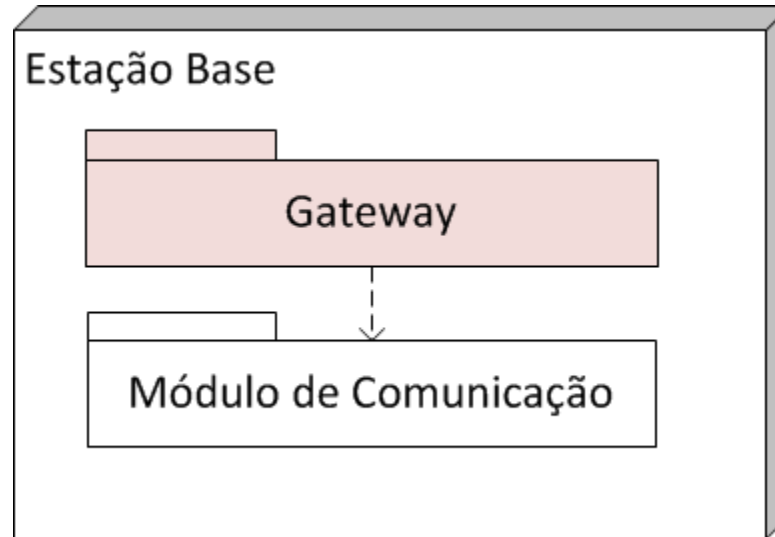
- **A Sun/Oracle disponibilizou uma aplicação de integração do SPOTs à Web segundo os preceitos da WoT**
 - Essa aplicação possui uma infraestrutura de comunicação HTTP entre a EB e os SPOTs
- **O projeto Inclui:**
 - Um *gateway* que é executado na EB e tem como função básica repassar as requisições Web advindas da Internet para os SPOTs através da rede ZigBee
 - Os SPOTs possuem um servidor HTTP e as suas funcionalidades são acessadas através de requisições HTTP conforme os princípios REST

Componentes físicos e os respectivos subsistemas utilizados no projeto *WebOfThings*



O Módulo de Comunicação representa uma API para o protocolo de comunicação rádio subjacente

Focando no *Gateway*



Funções básicas:

- (i) prover serviço de registro e descoberta de recursos
- (ii) encaminhar as requisições HTTP da Web para os SPOTs e entregar as respostas devolvidas para os clientes que enviaram as requisições.

Mas o gateway oferece outras funcionalidades...

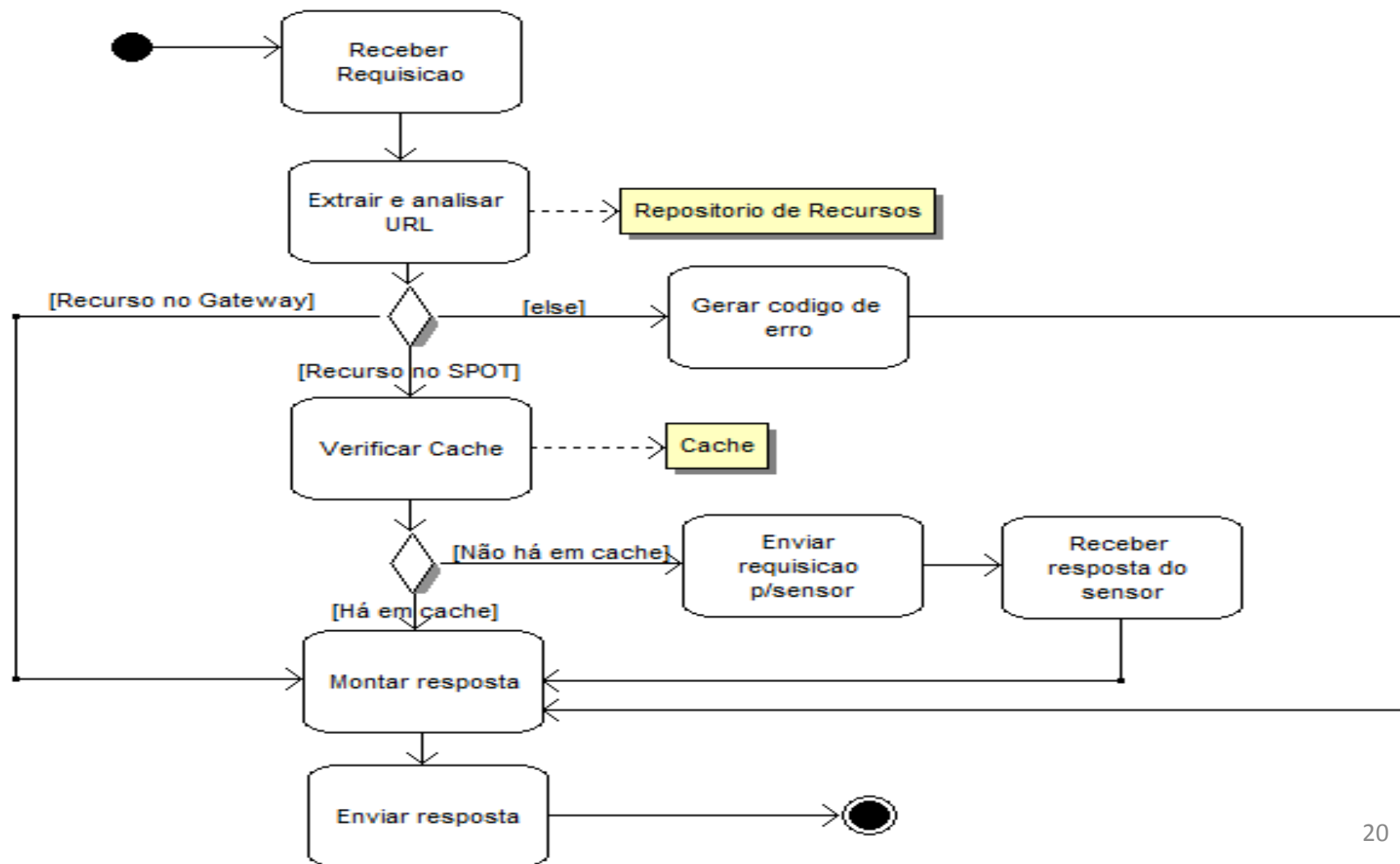
Funções do *Gateway* (1)

- **Fornece um serviço de registro/descoberta de SPOTs**
 - Novos SPOTs podem se registrar no *gateway* para ter suas funcionalidades disponibilizadas na Web
 - SPOTs descobertos se tornam subrecursos do *gateway*
 - Verifica periodicamente se os SPOTs descobertos ainda estão disponíveis

O Serviço de Descoberta será apresentado mais detalhadamente quando estivermos falando de *Smart Gateway

Funções do *Gateway* (2):

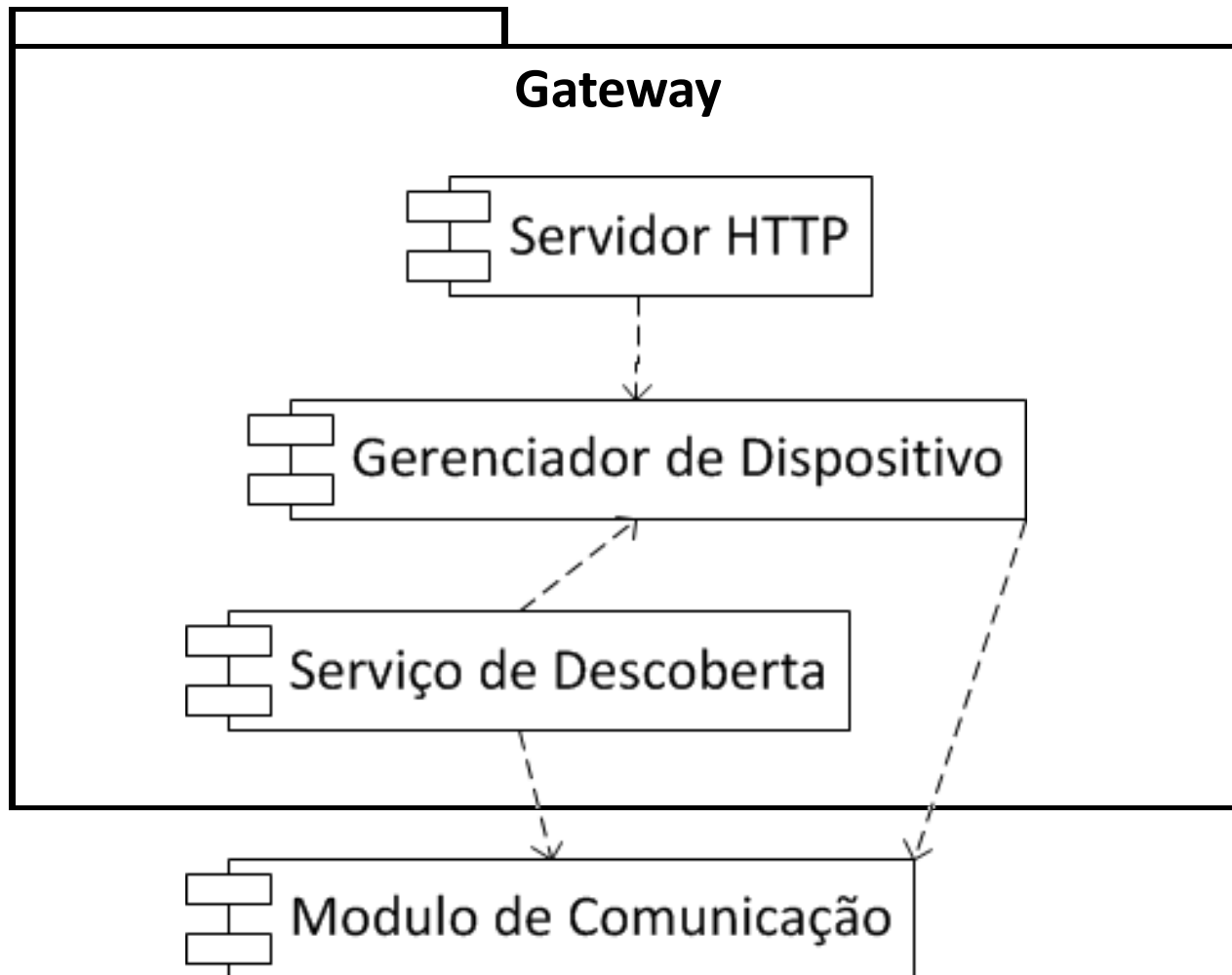
Atividades realizadas pelo *Gateway* ao receber uma requisição



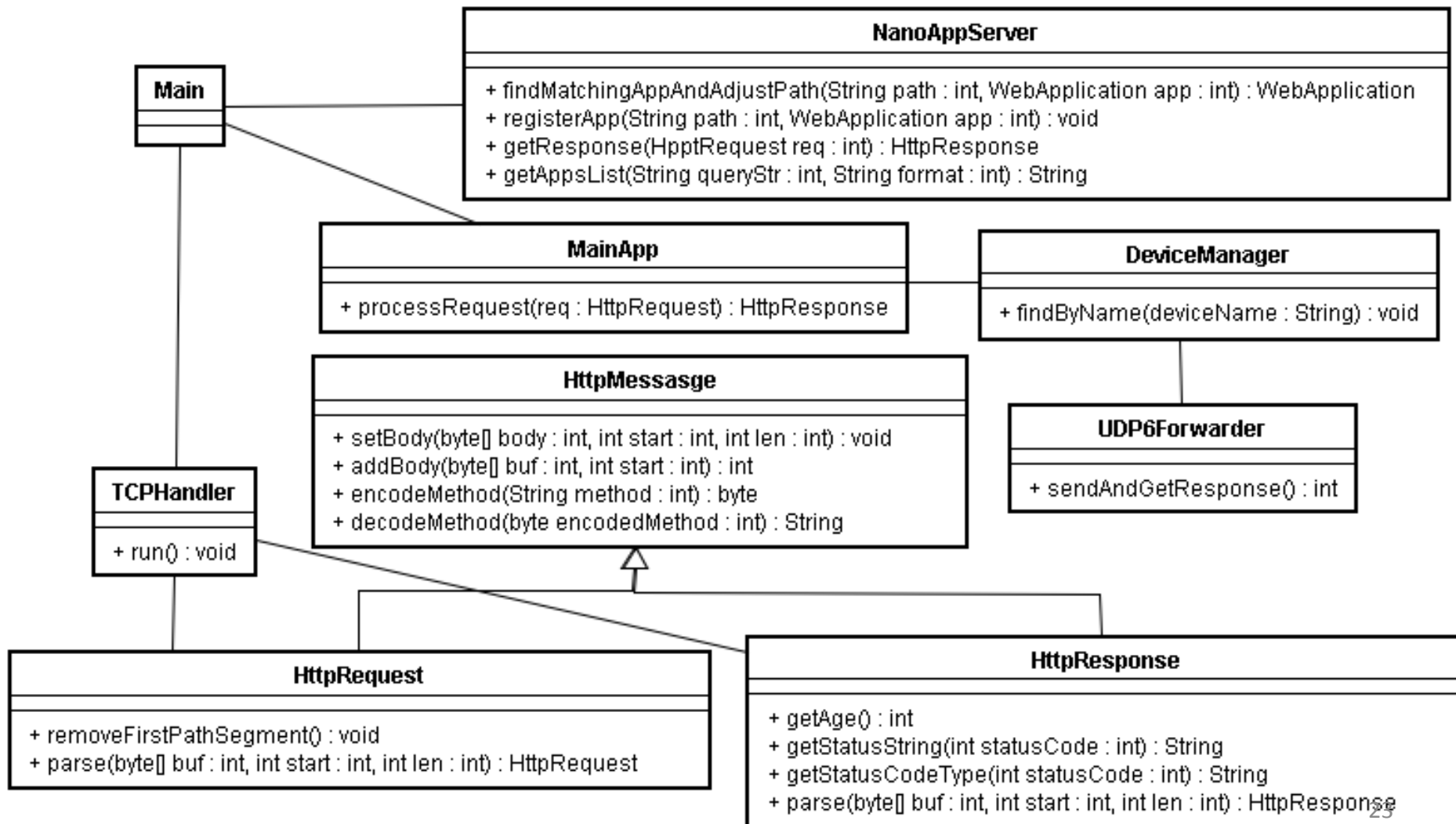
Funções adicionais do *Gateway*

- **Mantém os dados dos SPOTs em *cache***
 - Esses dados são utilizados para responder requisições direcionadas ao mesmo SPOT-recurso
 - Evita múltiplos acessos ao SPOT
 - importante principalmente para economia de energia dos dispositivos
- **Fornece uma representação HTML com *hiperlinks* para os SPOTs conectados ao gateway**
- **Compacta as requisições HTTP antes de passá-las para os SPOTs**

Componentes do *Gateway*

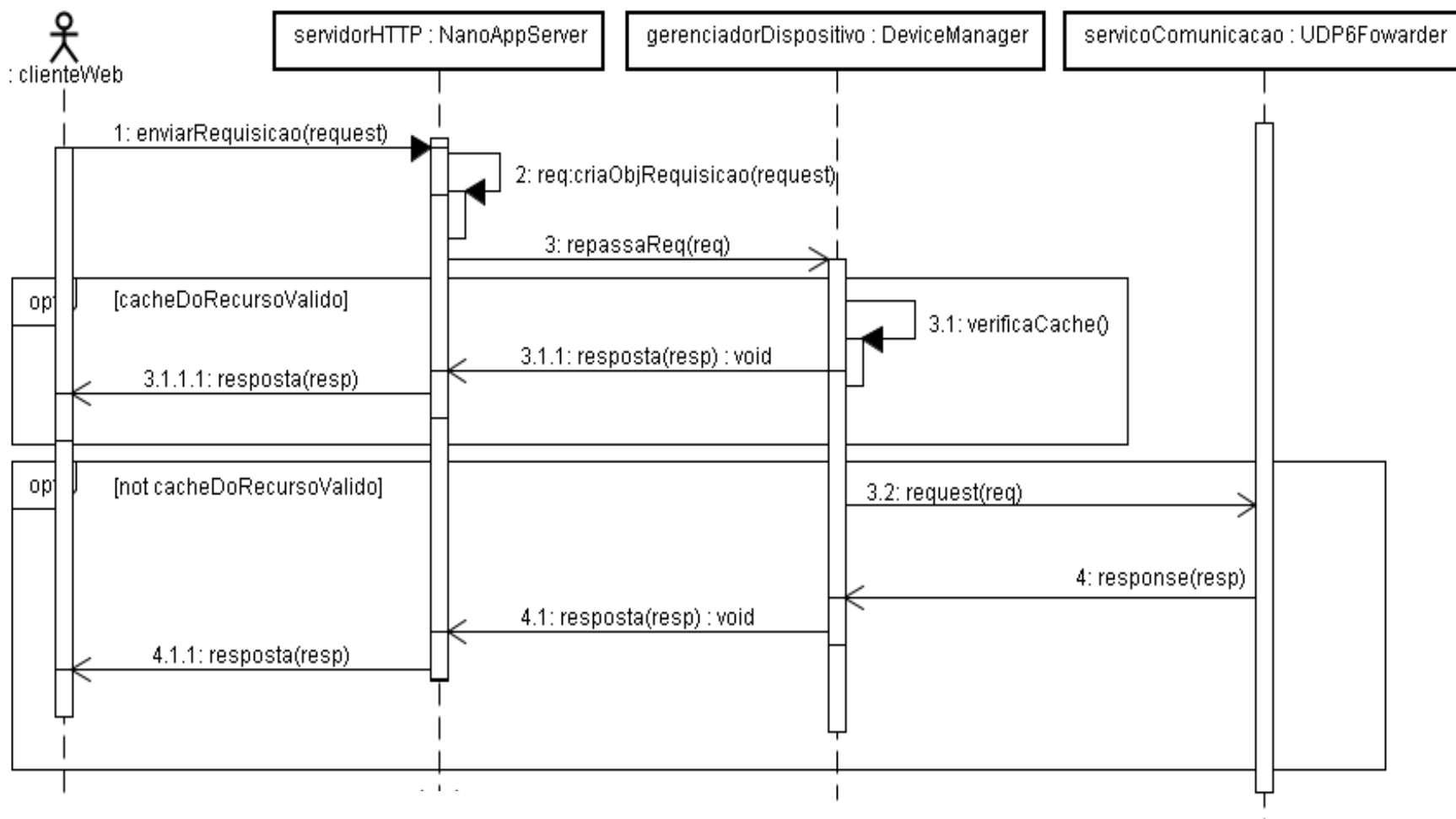


Classes do Gateway utilizadas para repassar requisições para os SPOTs

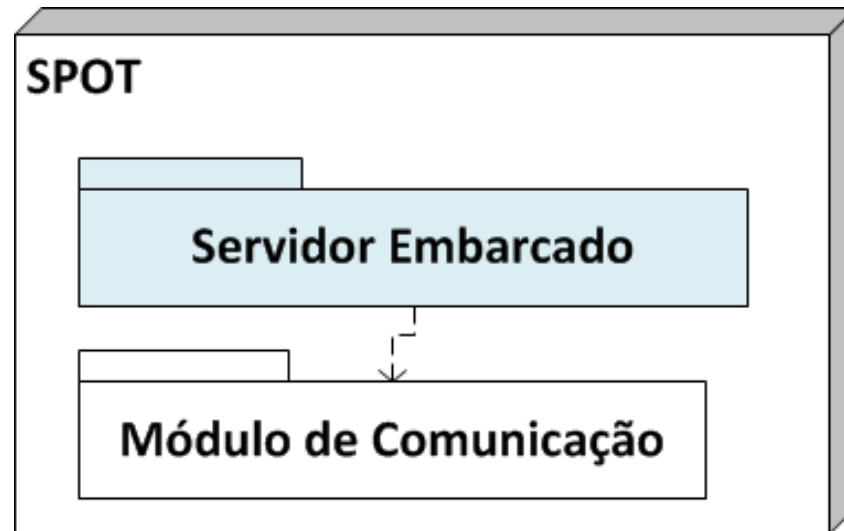


Sequência do tratamento de requisições no *gateway*

sd Diagrama de Sequencia do tratamento de requisição realizado no gateway



Focando no SPOT *free-range*



Esses dispositivos e suas funcionalidades são o alvo das requisições dos clientes. Os SPOTs realizam uma série de funções...

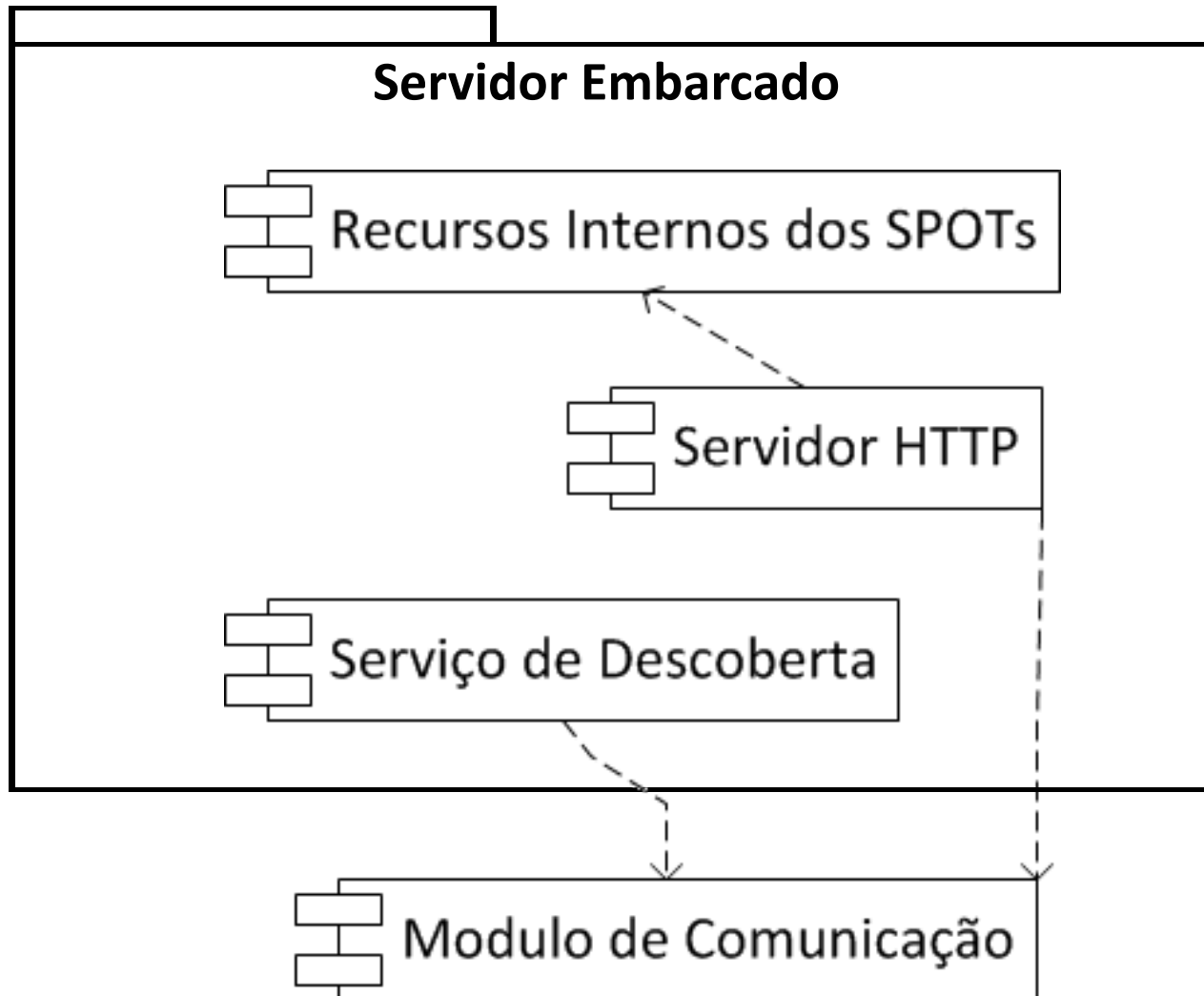
Funções do SPOT *free-range*

- **Fornecer seus serviços na forma de recursos RESTful**
- **Responder as requisições direcionadas aos seus recursos**
 - Fornecer uma representação com *hiperlinks* para todos os seus recursos
 - Realizar a tarefa especificada na requisição e retornar uma representação com código HTTP de resposta

Funções do SPOT *free-range*

- **Responder a mensagens de anúncio enviadas pelo gateway para fins de registro**
- **Descompactar requisições HTTP e compactar as respostas para diminuir o tamanho da mensagem transferida na rede IEEE 802.15.4**

Componentes do SPOTs



Classes executadas nos SPOTs para que seus recursos sejam acessados através de uma interface RESTful

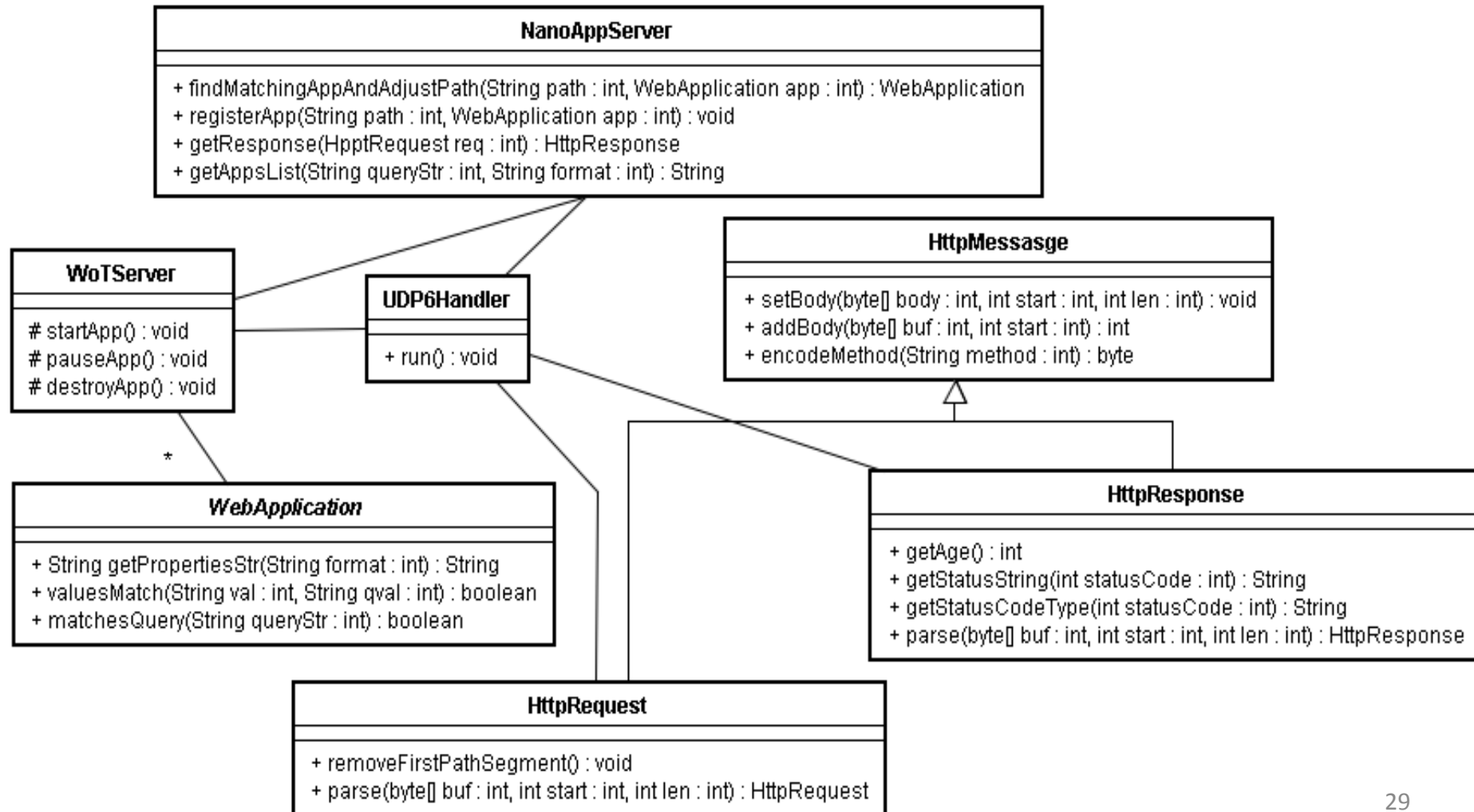
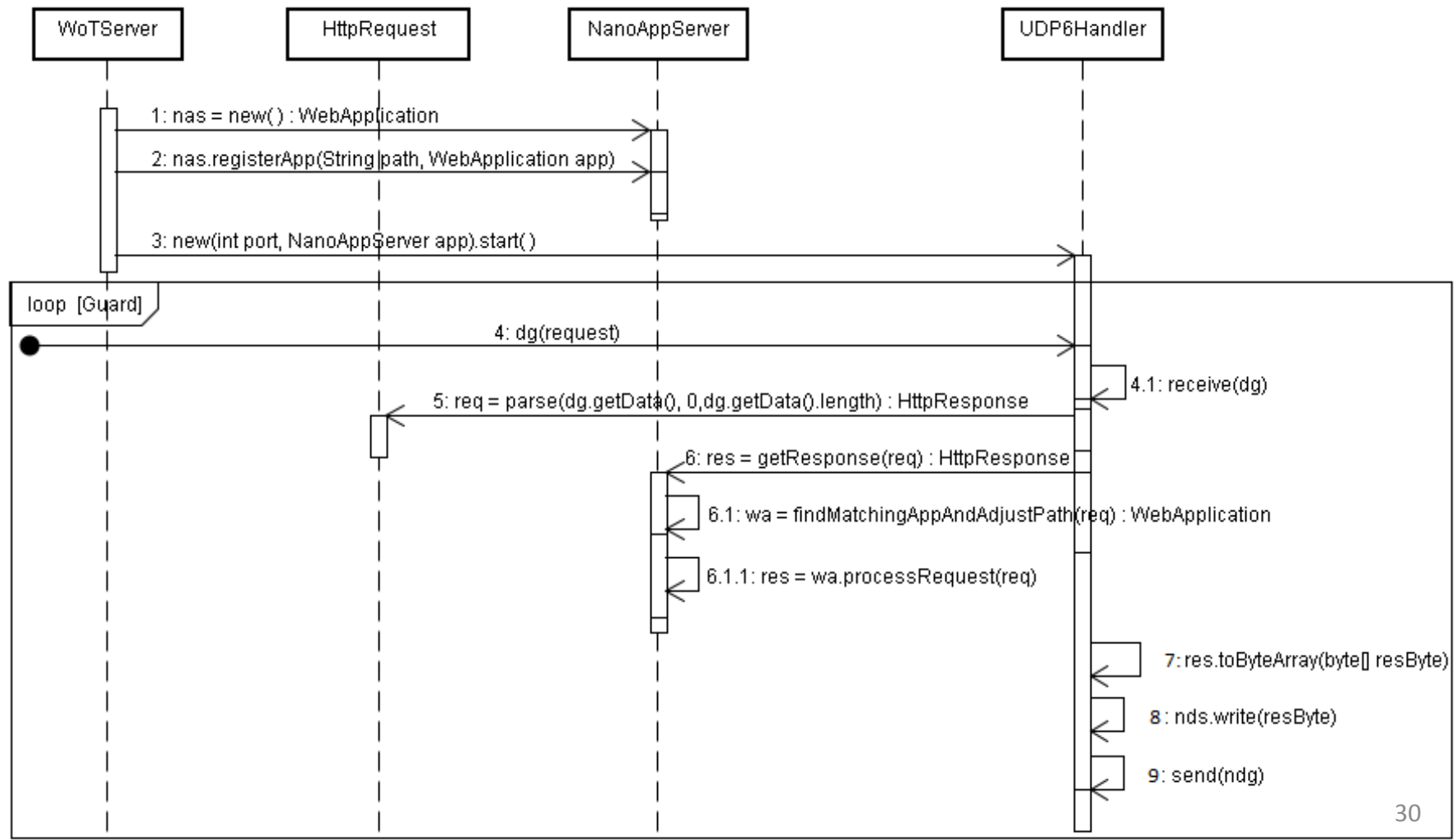


Diagrama de Seqüência das mensagens trocadas durante a execução da aplicação embarcada nos SPOTs



Exemplo de Aplicação criada sobre a Infraestrutura de comunicação HTTP

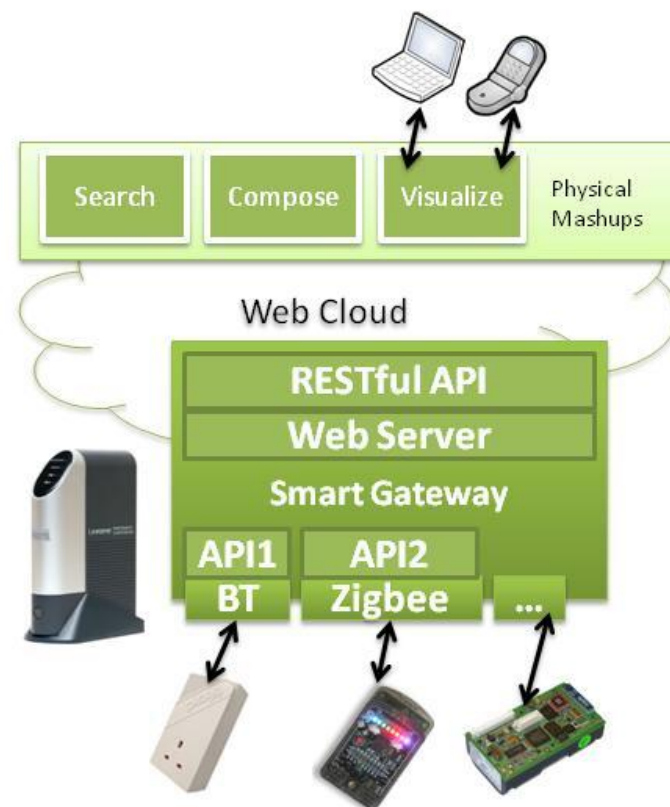
- **Requisitos da aplicação de exemplo :**
 - Obter dados de luminosidade
 - Obter status da bateria
 - Interagir com leds do nó (simulando uma operação de *atuação*)
- **A aplicação de exemplo rodando nos SPOTs permitirá:**
 - Retornar uma representação com *hiperlinks* para todos os recursos RESTful disponíveis no SPOT
 - Retornar uma representação da luminosidade capturada pelo dispositivo de sensoramento do SPOT
 - Interagir com os leds alterando a sua cor ou os fazendo piscar
 - Retornar uma representação contendo o nível da bateria

Executando a aplicação

- **Implantando código no SPOT**
 - Conectar o SPOT ao computador através da interface USB
 - Fazer a implantação da aplicação
 - Desconectar o SPOT e reiniciá-lo
- **Executando o *gateway***
 - Conectar a EB ao computador através da interface USB
 - Lembrando que quando nos referimos ao SPOT e usamos o termo EB, estamos falando do dispositivo que não é um SPOT *free-range*.
 - Executar o *gateway* na EB (computador com SPOT-EB conectado)

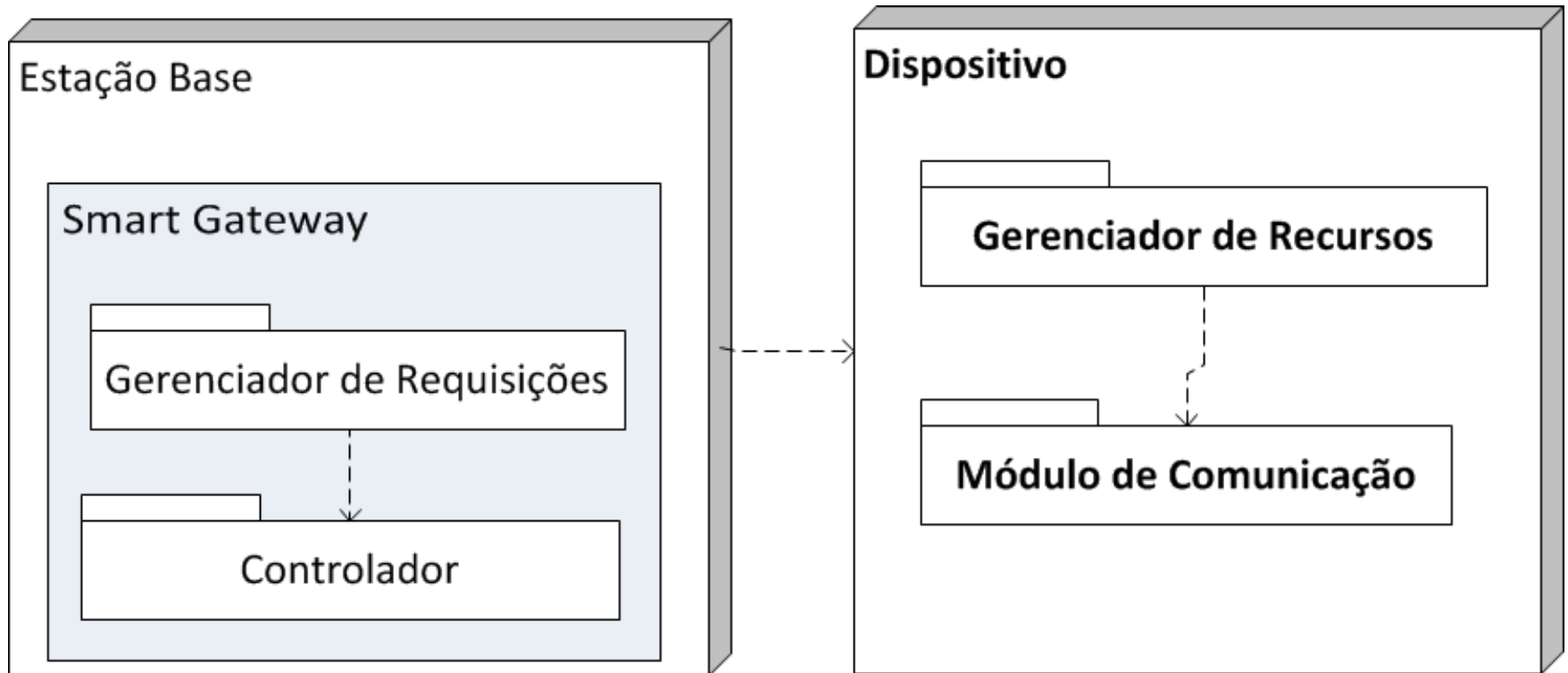
Segunda Abordagem de Integração de Dispositivo

Smart Gateway:
utilizando o conceito
de controlador para se
comunicar com
diferentes dispositivos

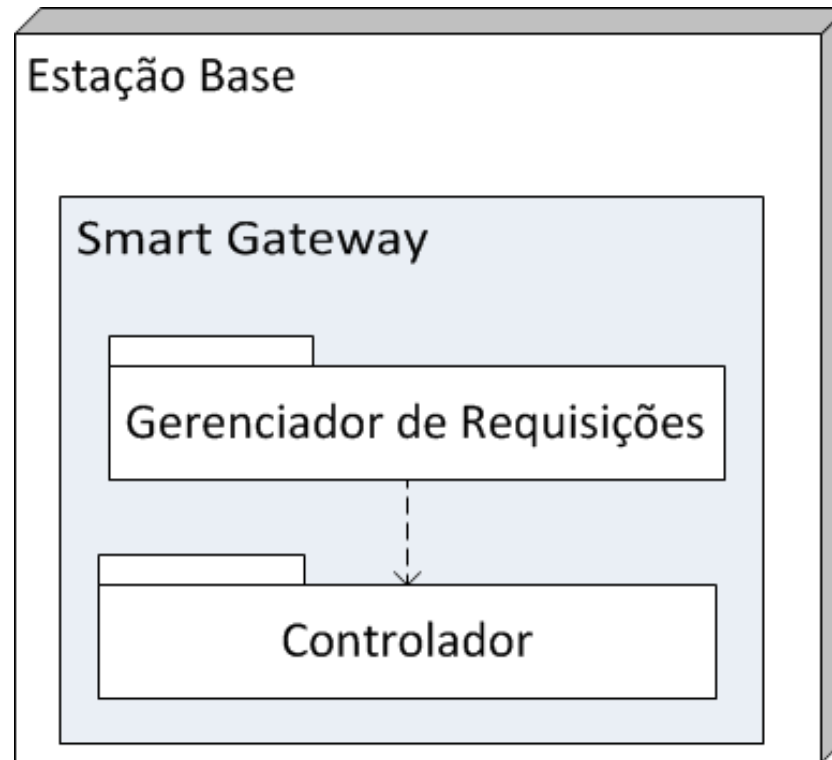


Fonte: Vlad Trifa, Samuel Wieland, Dominique Guinard, Thomas Michael Bohnert: Design and Implementation of a Gateway for Web-based Interaction and Management of Embedded Devices. Proceedings of the 2nd International Workshop on Sensor Network Engineering (IWSNE 09). Marina del Rey, CA, USA, June 2009

Componentes físicos e subsistemas presentes na integração de dispositivos a Web utilizando *Smart Gateway*



Focando no *Smart Gateway*

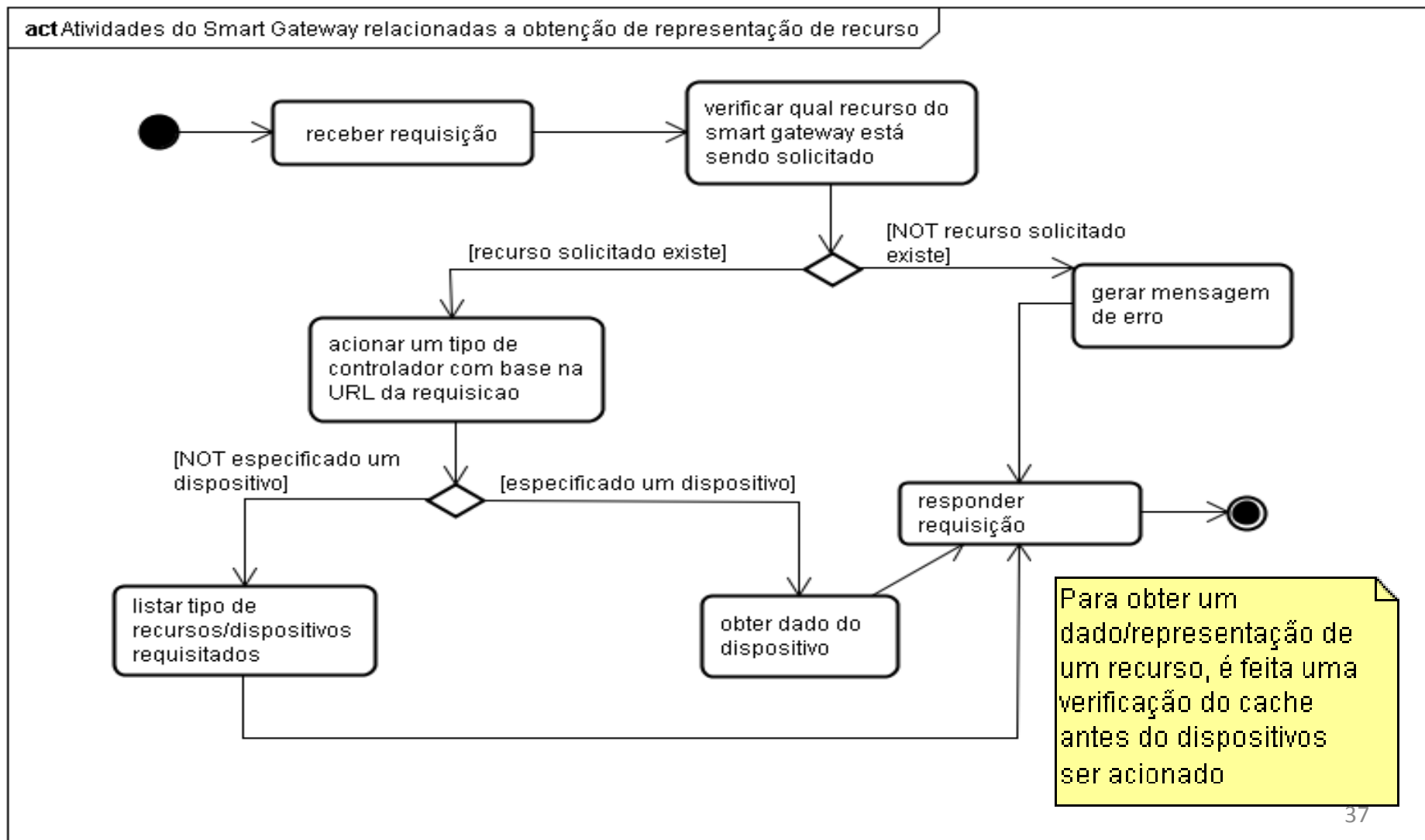


O Smart Gateway desenvolvido é uma adaptação do gateway do projeto WebOfThings da Sun/Oracle

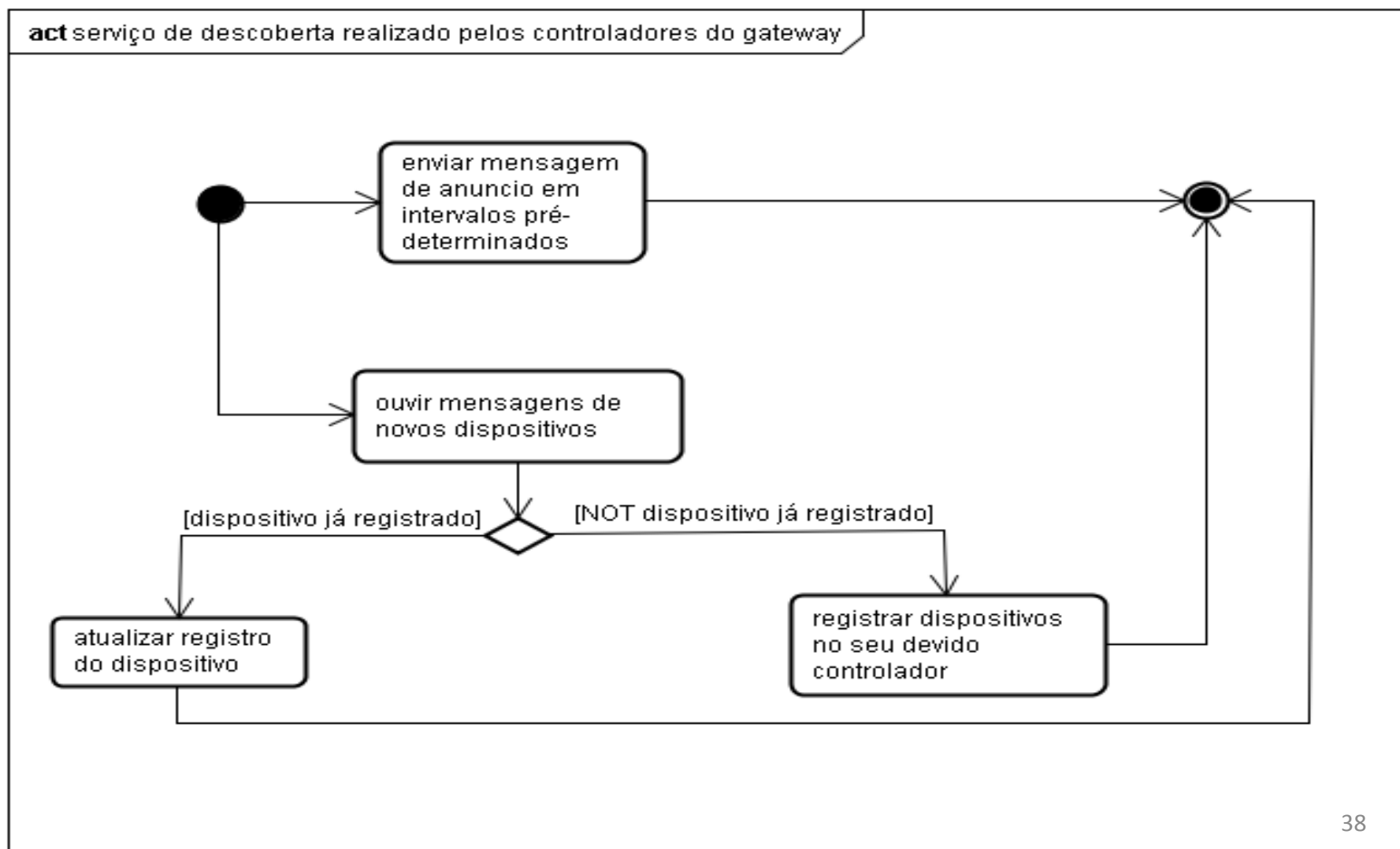
Funções do *Smart Gateway*

- **Todas de um gateway “normal”**
- **Permitir que dispositivos com recurso de hardware insuficiente para executar um servidor embarcado sejam acessados via Web**
 - Oferece uma interface Web RESTful para acesso as funcionalidades dos dispositivos
 - Retorna uma representação contendo *hiperlinks* para todos os dispositivos conectados ao smart gateway
 - Uso de controladores para comunicação com os dispositivos por meio da suas API e protocolos específicos

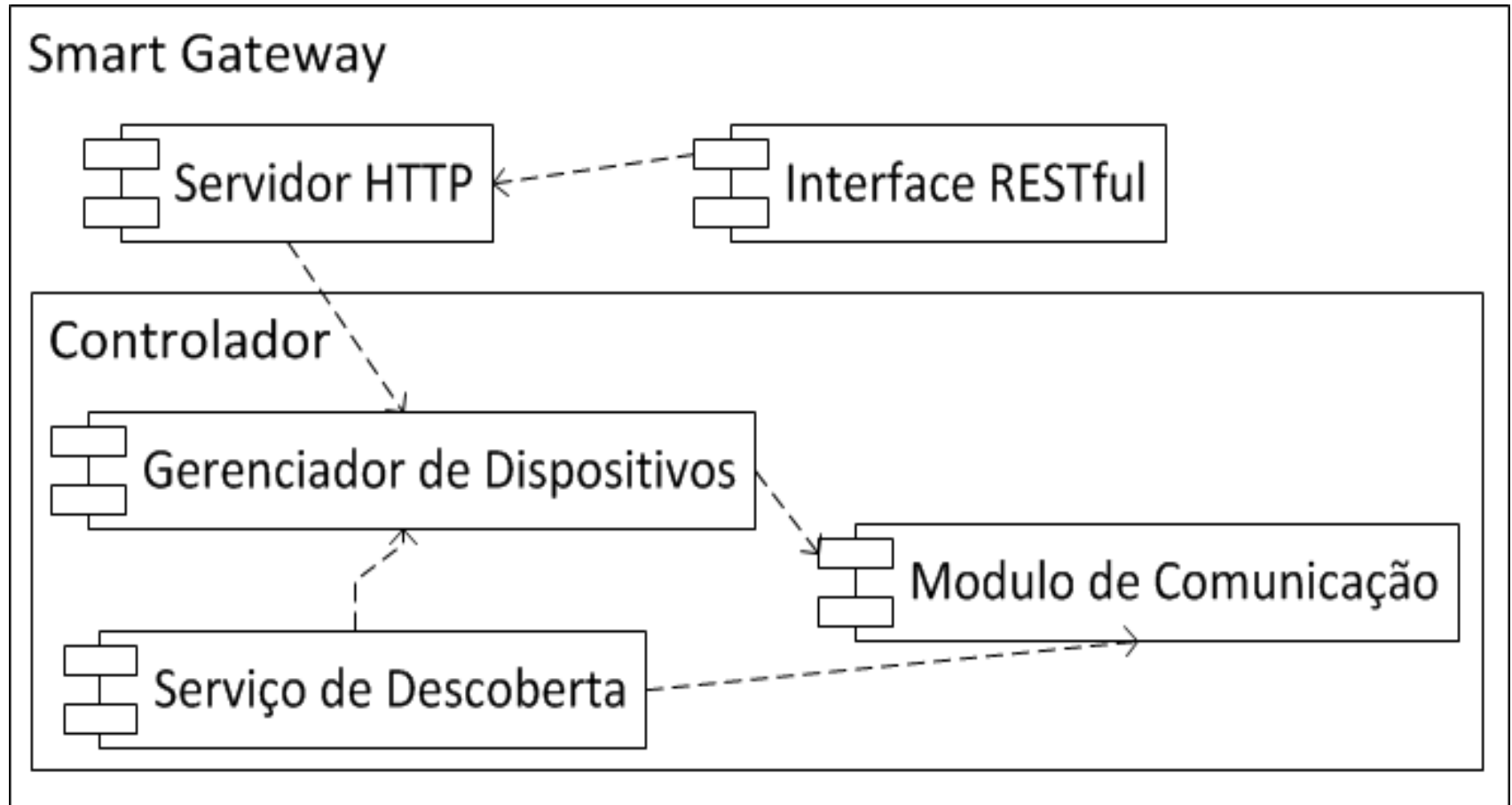
Atividades do *Smart Gateway* relacionadas a obtenção de representações de recursos



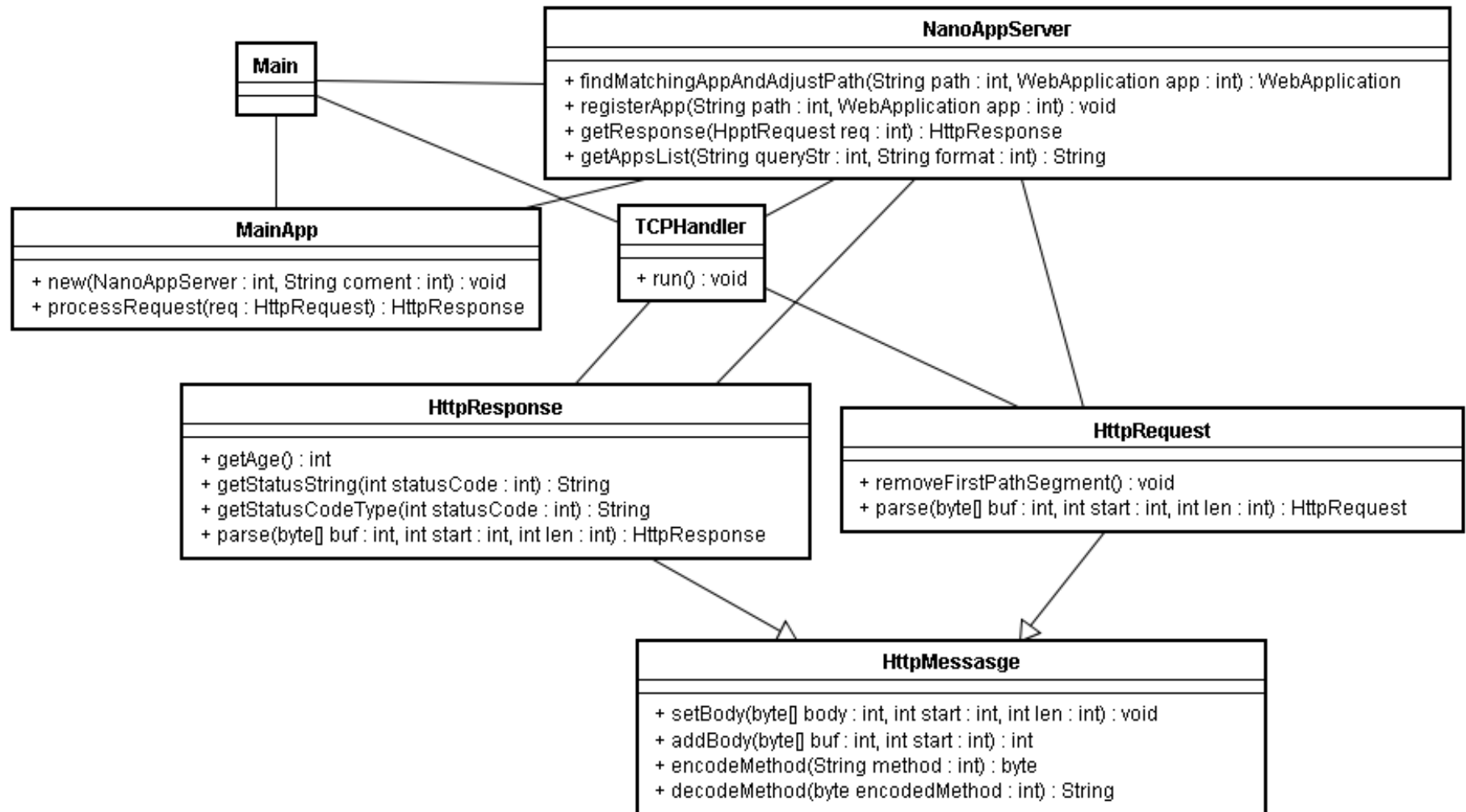
Atividades do *Smart Gateway* relacionadas a descoberta de recursos



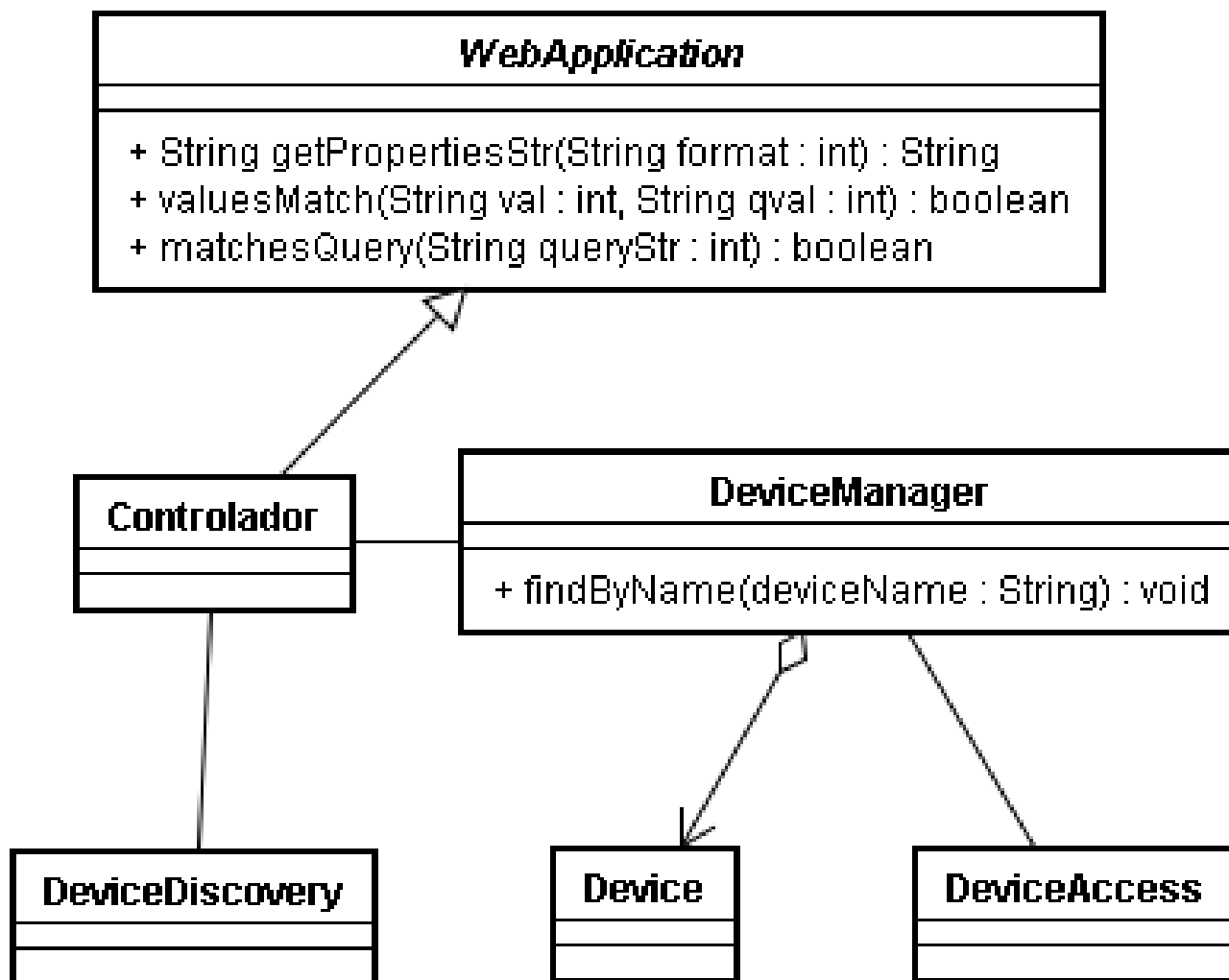
Componentes do *Smart Gateway* - *Detalhamento*



Classes Utilizadas para oferecer a interface RESTful de acesso aos recursos dos dispositivos

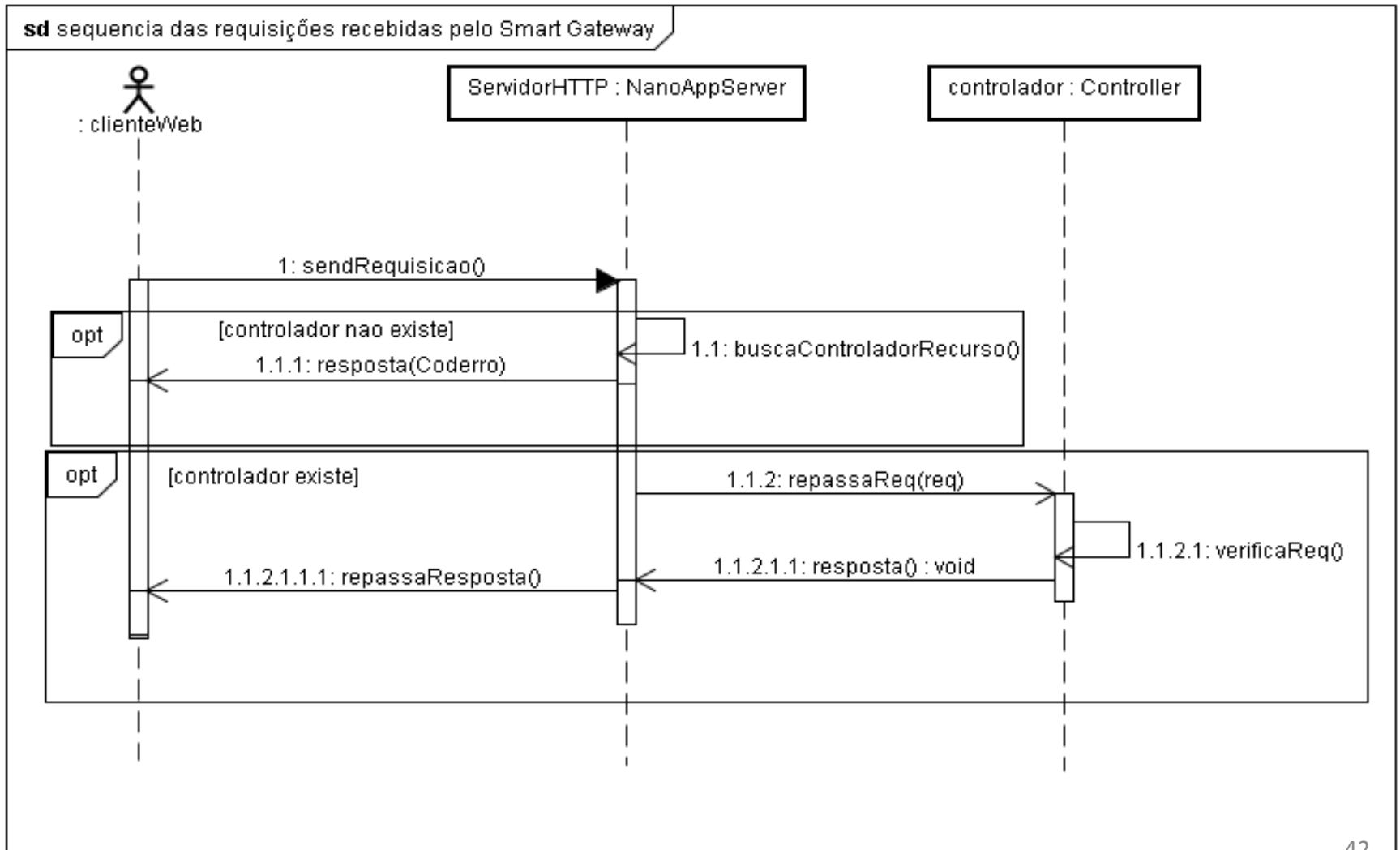


Principais Classes dos Controladores

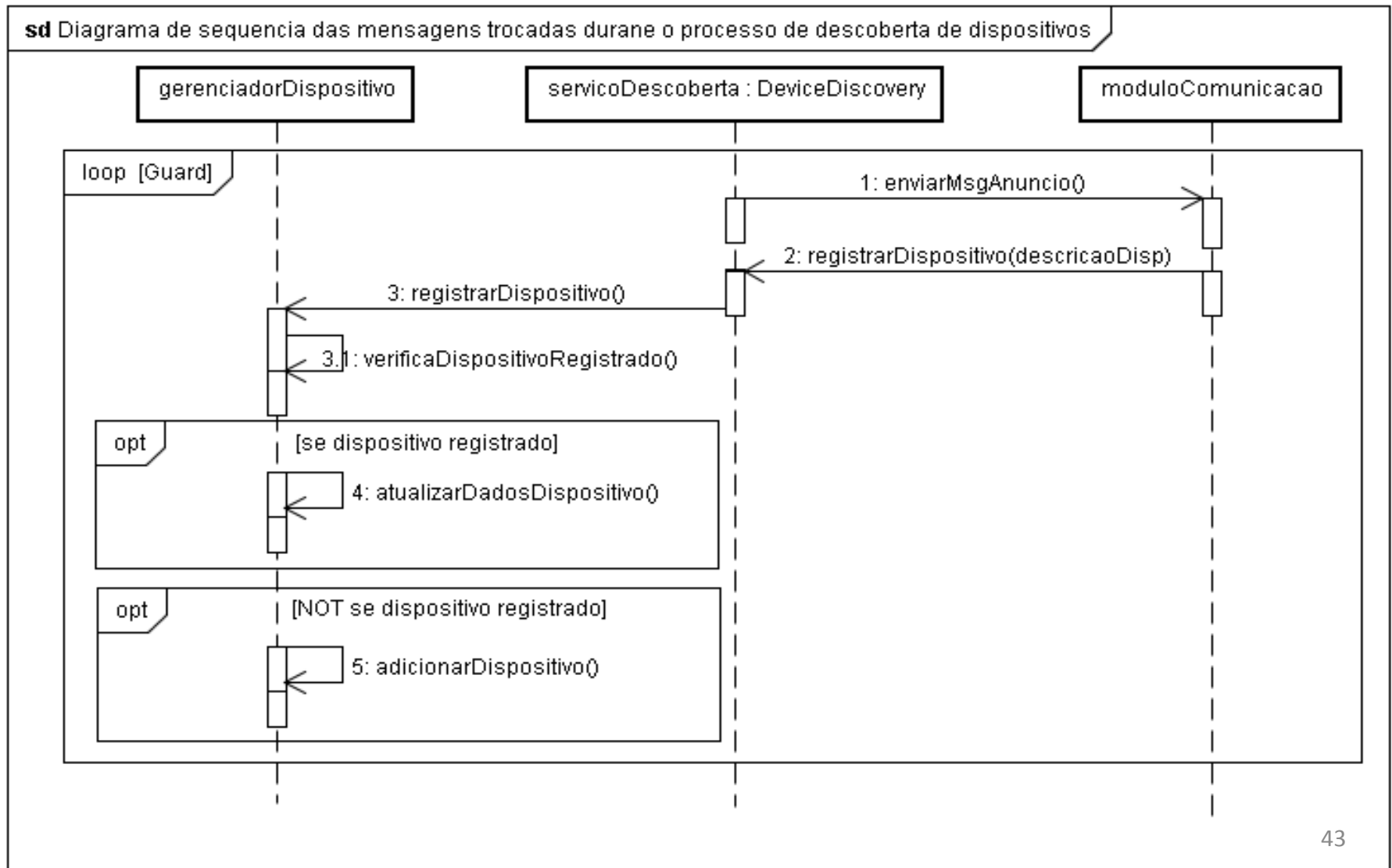


Principais classe dos controladores dos dispositivos

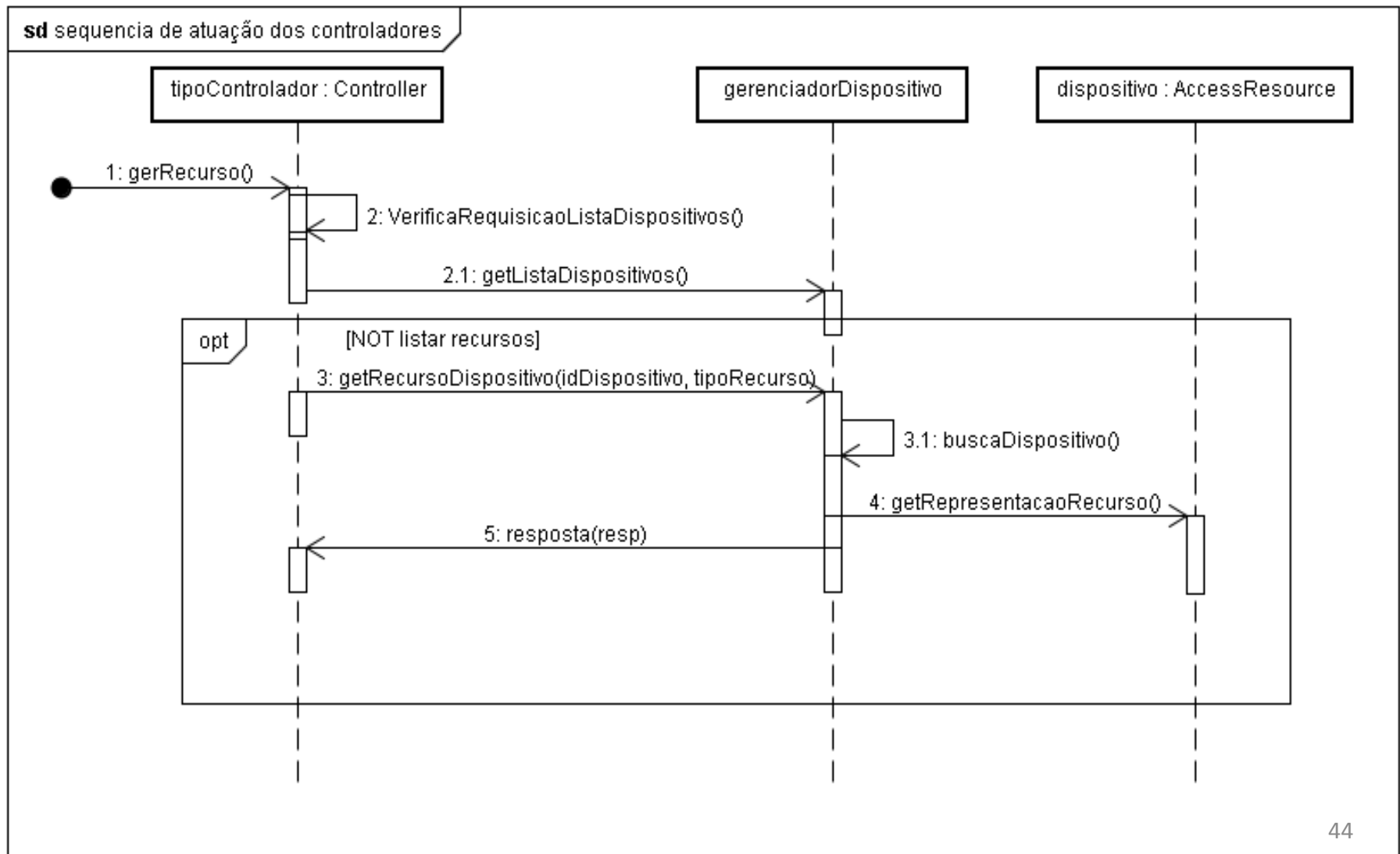
Sequência das mensagens trocadas pelo *Smart Gateway* ao receber uma requisição



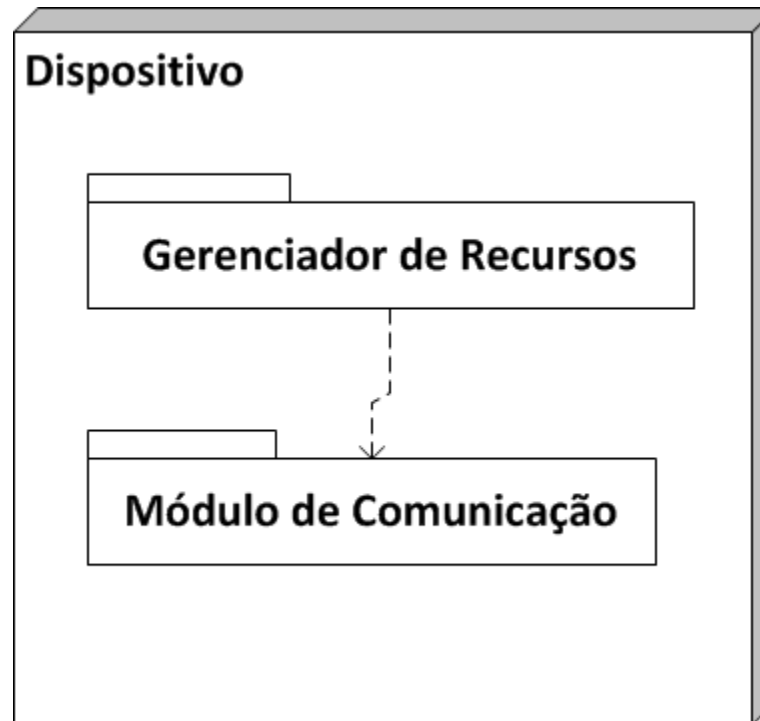
Sequência das mensagens trocadas durante o processo de descoberta



Sequência das mensagens trocadas por um controlador



Analizando o papel dos dispositivos



A comunicação entre o dispositivo e o *smart gateway* é realizada pelo controlador. Os controladores devem conhecer e acessar os dispositivos e seus recursos

Dispositivos Utilizados no Exemplo de Integração ao *Smart Gateway*

- **Nesse exemplo vamos explorar a heterogeneidade de recursos**
 - Gerenciada pelo Smart Gateway
- **Além de um SPOT sem servidor embarcado, vamos considerar:**
 - Sensor de plataforma diferente da SUN
 - Micaz sem servidor embarcado
 - SPOT com servidor embarcado

Dispositivos Utilizados no Exemplo de Integração ao *Smart Gateway*

- **SPOT com servidor embarcado**
- **SPOT sem servidor embarcado**
- **Micaz sem servidor embarcado**



Web Of Things



SPOT API



Micaz
47

Sun SPOT com servidor embarcado

- **Um controlador que recebe a requisição HTTP da rede TCP/IP e a repassa para o SPOT utilizando a rede IEEE 802.15.4**
- **Mesma aplicação utilizado no primeiro exemplo de integração**
 - Mesma forma de integração do projeto WebOfThings da Sun SPOT World

SPOT sem servidor embarcado

- **O controlador se comunica com o SPOT utilizando a API deste**
 - A abstração de comunicação é oferecida pelo controlador
 - O controlador realiza o processo de descoberta enviando mensagens em *broadcast* na porta 201
 - Mensagem JSON de anúncio enviada pelo controlador dos SPOTs que não possuem servidor embarcado

```
{ "gateway_id" : "MAC_gateway" , "timestamp" : "timestamp" }
```

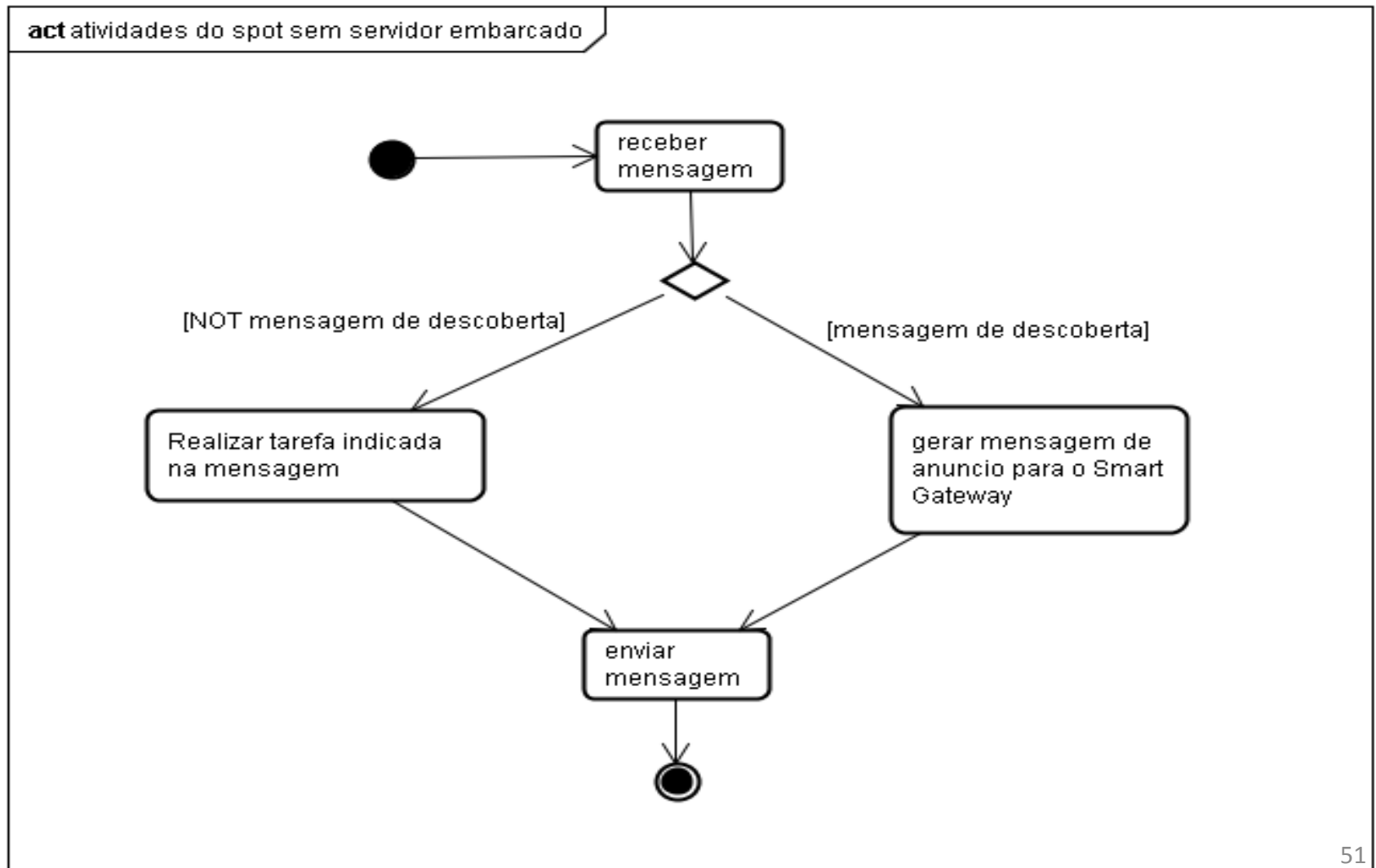
SPOT sem servidor embarcado

- Mensagem JSON de registro enviada pelo SPOT sem servidor embarcado para o seu controlador no *Smart Gateway*

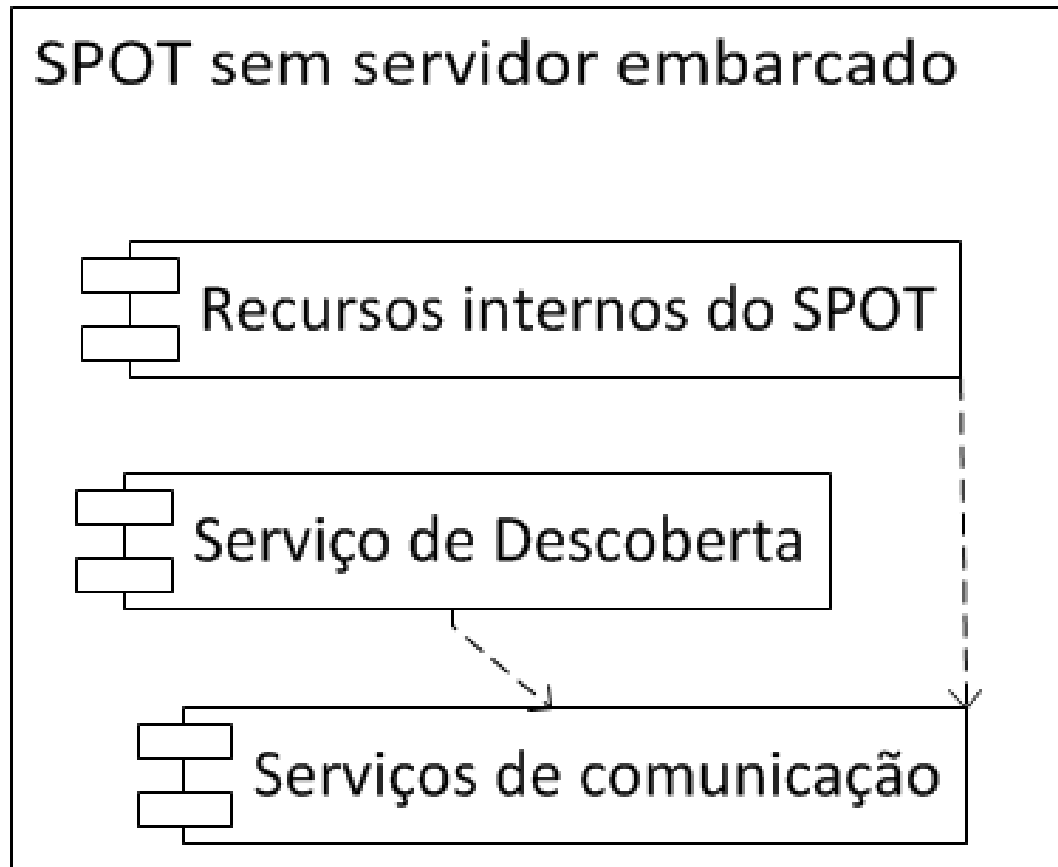
```
{  
  "MAC" : "0014.4F01.0000.0F20" ,  
  "timestamp" : " 1300223762980" ,  
  "recursos" : {  
    "temperatura" : "/temp" ,  
    "luminosidade" : "/light"  
  }  
}
```

- O controlador mantém as representações dos recursos em *cache*

Atividades do SPOT sem servidor embarcado



Componentes do SPOT sem servidor embarcado



Micaz

- **Micaz sem servidor embarcado**
 - Plataforma de sensor da família crossbow
 - Possui pouquíssimo recurso de hardware comparado aos SPOTs
 - API diferente da que é utilizada pelos SPOTs
 - Utiliza um sistema operacional chamado tinyOS
 - Programado com uma linguagem específica chamada nesC



Micaz

- **Micaz sem servidor embarcado** (continuação)
 - Seu controlador acessa seus recursos utilizando a API que ele oferece
 - Controlador semelhante ao utilizado para o SPOT sem servidor embarcado
 - Não entraremos em detalhes
 - A aplicação desenvolvida oferece dados de temperatura e luminosidade
 - Foram utilizadas duas aplicações de demonstração que acompanham o tinyos 2.x: RadioSenseToLeds e TestSerial

Implantando a aplicação nos SPOTs

- **Implantando código no SPOT**
 - Para SPOTs a implantação é semelhante a que foi realizada para o SPOT com servidor embarcado
 - Implantar código no SPOT com servidor embarcado
 - Implantar código no SPOT sem servidor embarcado
 - Conectar EB ao computador e executar software do *Smart Gateway*

Implantando a aplicação no Micaz

- **Implantando código no Micaz**

- Instalando RadioSenseToLeds em um micaz e atribuindo-lhe ID 1
 - Comando: `SENSORBOARD=mda100 make micaz install.1 mib510,/dev/ttyUSB0`
- Instalando TestSerial no micaz que vai ser utilizado na EB e atribuindo-lhe ID 0
 - Comando: `make micaz install.0 mib510, /dev/ttyUSB0`
- Executando a classe java que se comunica com o micaz
 - `Java TestSerial -comm serial@/dev/ttyUSB1:micaz`

Para maiores informações sobre o micaz acessar
<http://www.dcc.ufrj.br/~silvana/>

Integrando Dispositivos a Web: As URIs de Acesso

- **A URI localiza o gateway, identifica o controlador e em seguida o dispositivo**
 - Recursos são identificados de forma única e global
 - Dispositivos são subrecursos do gateway
 - As funcionalidades do dispositivo são subrecursos do mesmo

Exemplos de URIs:

<http://localhost:8888/spotapi/spot-1265/light>

URI de um SPOT com servidor embarcado

<http://localhost:8888/micaz/micaz-1/temp>

URI de um SPOT sem servidor embarcado

Integrando Dispositivos a Web: **Representação de Estado**

- **Os dispositivos fornecem dois tipos de representação: HTML e JSON**

1. HTML

- Simplifica a interação entre clientes (humanos) e os recursos dos dispositivos
- *Hiperlinks* para os recursos podem ser facilmente utilizados pelas pessoas em seus navegadores Web
- Retornado pelo SPOT que possui servidor embarcado
- Retornado pelo *gateway* com uma listagem de todos os dispositivos conectados a ele

Integrando Dispositivos a Web: Representação de Estado

2. JSON

- Esse tipo de representação é enviada por todos os dispositivos
- Utilizado pelo SPOT que não possui servidor embarcado para retornar uma representação de um recurso ou a lista de seus subrecursos
- Utilizado pelo SPOT que possui servidor embarcado quando o mesmo recebe uma requisição especificando este tipo de representação

Integrando Dispositivos a Web: **Representação de Estado**

Recursos Disponíveis

/luminosidade

/temperatura

Listagem dos recursos disponíveis em um SPOT com servidor embarcado

```
{"resource":{"typeOfResource":"temperature","scale":"Celsius","value","32","timestamp":"1300223762980","deviceId":"temperatura","link":"/temperatura"}}
```

Representação JSON do recurso temperatura

Web das Coisas: Indo além da integração de dos dispositivos físicos a Web

- **Devemos nos lembrar que:**
 - Os dados brutos dos dispositivos podem não possuir valor para o usuário final
 - Os dados desses dispositivos podem ser integrados em aplicações Web
 - Os *mashups* são aplicações Web que combinam dados de mais de uma fonte em uma única ferramenta integradora
 - Os conteúdos usados são obtidos a partir de terceiros via uma interface Web pública

Mashups Físicos

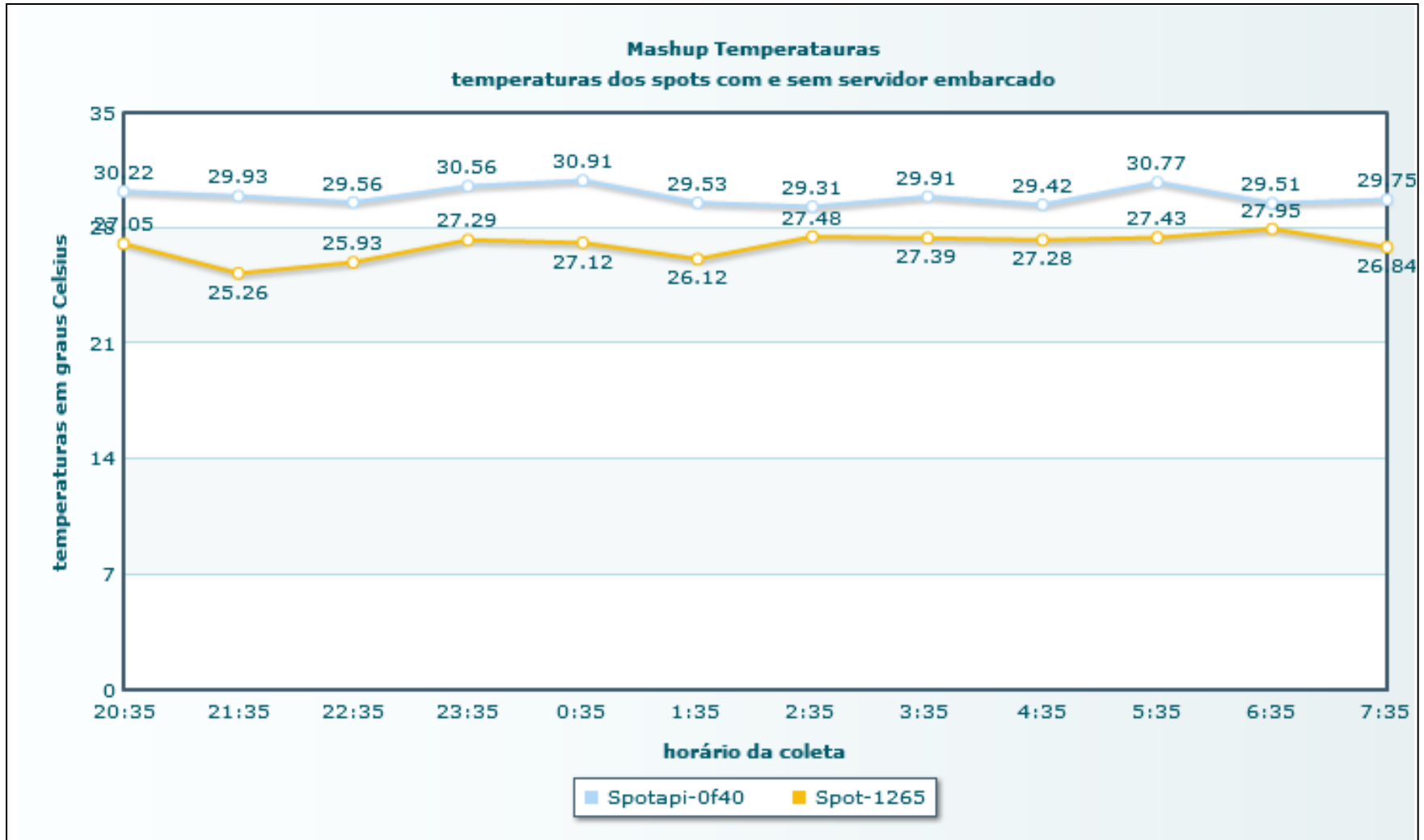


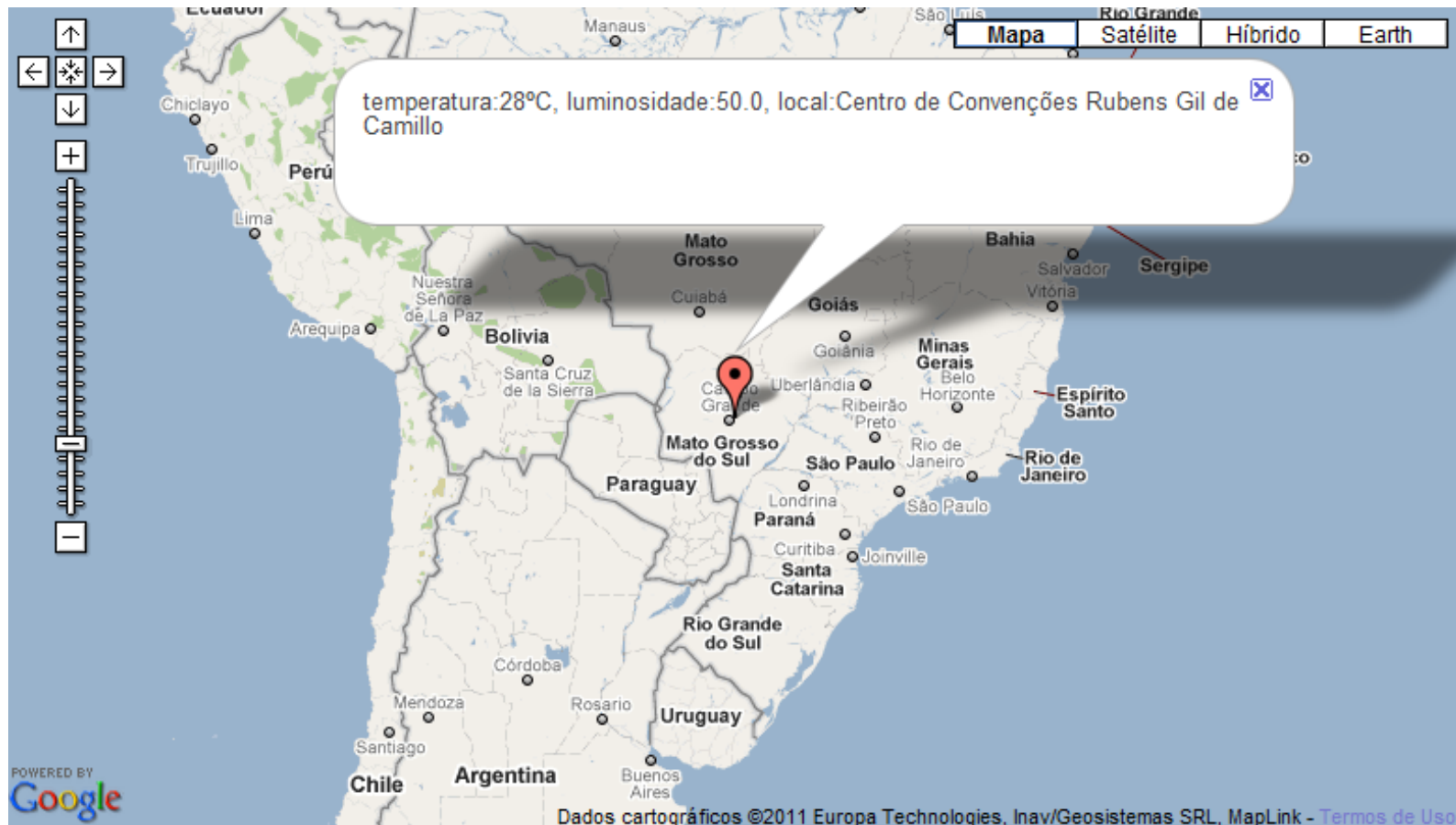
Gráfico de Temperaturas coletadas pelos SPOTs colocados em diferentes ambientes em um intervalo de 12 horas

Mashups Físicos



Mashup físico que apresenta em uma mapa da Web a localização de um SPOT e o valor da temperatura local obtida através do acesso Web ao dispositivo de sensoriamento desse SPOT

Mashups Físicos



Mashup físico que apresenta em uma mapa a temperatura obtida de um SPOT e a luminosidade obtida de um Micaz

Conclusão

- **Nesse minicurso nós vimos:**
 - Os conceitos básicos da Web das Coisas
 - Uma arquitetura para WoT baseada em REST e ROA
 - A implementação de alguns componentes da WoT
 - A aplicação dos padrões Web para integração dos dispositivos tem se mostrado uma solução promissora

Conclusão

- **Nesse minicurso nós vimos (continuação):**
 - A Web possibilita a criação de uma nova gama de aplicações que conterão informações do mundo real
 - Os padrões Web tornam a programação dos dispositivos mais simples
 - A Web é simples e aberta
 - A Web como um terreno comum para dispositivos e aplicações pode alavancar a ubiquidade e versatilidade da mesma
 - A comunidade de desenvolvedores Web são potenciais desenvolvedores da WoT

Dúvidas





Web das Coisas

Conectando Dispositivos Físicos ao Mundo Digital

Autores

e -mais: Tiago Cruz de França, tcruz.franca@gmail.com

Paulo F. Pires, paulo.f.pires@gmail.com

Luci Pirmez, luci.pirmez@gmail.com

Flávia C. Delicato, fdelicato@gmail.com

Claudio Farias, cmicelifarias@gmail.com