

## 5 Estudos de Caso

Com o intuito de utilizar o *framework* Shine e demonstrar sua viabilidade na instanciação de sistemas multi-agentes distribuídos que interagem com a infra-estrutura RFID existente, foram desenvolvidos dois estudos de caso que solucionam dois problemas distintos. Este capítulo apresenta os problemas e as soluções implementadas.

O primeiro problema é o da realização de inventários de estoques. Este é um problema clássico de armazéns, centros de distribuição e grandes estoques que armazenam grandes quantidades de produtos distintos. Inventário em tempo real é uma das diversas aplicações vislumbradas com a utilização da tecnologia de RFID, e promete melhorar muito o processo de inventário utilizado hoje. A seção 5.1 apresenta com mais detalhes o problema.

O segundo problema é um problema ainda não conhecido pelas empresas que participam de uma cadeia de suprimentos, que ainda não implantaram a utilização de RFID: é o problema de reconciliação de EPC fantasma. A seção 5.2 apresenta com mais detalhes o problema.

Com estes dois estudos de caso, pretende-se mostrar que o *framework* Shine é útil para a instanciação de aplicações direcionadas tanto a problemas clássicos de cadeias de suprimentos, como inventário, como a novos problemas surgidos a partir da implantação da tecnologia RFID.

### 5.1. Inventário

Esta seção apresenta o problema da realização de inventários em armazéns, centros de distribuição e grandes estoques.

#### 5.1.1. O Problema

A tarefa de realizar inventários em armazéns, centros de distribuição e grandes estoques tem o objetivo de inventariar todos os produtos presentes no local em questão em determinado momento, ou seja, saber exatamente quais

produtos existem em estoque, a quantidade presente de cada produto, e eventualmente sua exata localização.

Como parte do processo de entendimento do problema, o estudo de caso incluiu a visita a um estoque de uma fábrica de centrais telefônicas na cidade de Santana de Parnaíba, SP. A fábrica em questão é de porte médio, produzindo aproximadamente 200 centrais por mês. O estoque da fábrica é composto, em sua maioria, por produtos de matéria-prima e semi-acabados. Os produtos acabados são produzidos sob demanda e pouco permanecem em estoque.

Foi observado e relatado por funcionários da empresa que recorrentemente há perdas de eficiência e margem financeira na produção das centrais devido a erros de inventário, pois muitas vezes acredita-se que há determinado componente em estoque e, quando precisa-se dele para produzir uma central, percebe-se que não há. Isto acarreta em compra emergencial do componente, transporte rápido e mais caro, e atraso na entrega final. Todos esses fatores contribuem para o aumento de custo de produção além de prejudicar a imagem da empresa.

Outro fato observado é a própria realização dos inventários. O estudo realizado nesta fábrica teve a oportunidade de presenciar a realização de um. O processo de realização de inventário é extremamente custoso para a empresa, pois necessita de paralisação de parte do quadro de funcionários, além de ser naturalmente sujeito a erros humanos. Todo o processo é manual: há uma separação do material a ser inventariado; a contagem física é realizada no mínimo duas vezes; caso as contagens não coincidam mais contagens são realizadas; e nunca o resultado final é 100% confiável. Além desses problemas com o inventário em si, poucos dias depois, o mesmo pode deixar de ser real caso erros na contabilização de material recebido ou expedido aconteçam [3].

### **5.1.2. Inventários em tempo real com RFID**

A utilização de RFID para ajudar no processo de inventários é uma das aplicações mais proeminentes desta tecnologia. Tendo todos os produtos existentes em estoque possuindo etiquetas RFID emitindo seus códigos EPC, e possuindo leitoras espalhadas pelo armazém e infra-estrutura de software coordenando as leituras e processando os dados, é possível realizar inventários em tempo real, a qualquer momento e sem intervenção humana. Com essa

estrutura, é possível saber exatamente todos os produtos existentes em estoque, suas quantidades, locais de armazenamento, e ainda outros dados como temperatura ambiente, produtos com prazo de validade vencido, etc.

Atualmente, grandes empresas que enfrentam desafios para armazenagem, recebimento e expedição de produtos provenientes de diferentes fornecedores (centros de distribuição de grandes redes de varejo, por exemplo) investem muito tempo na tarefa de recebimento para atingir 99% de precisão nos dados de inventário. Com a utilização da infra-estrutura de RFID espera-se atingir quase 100% de precisão [3].

## **5.2.**

### **Reconciliação de EPC fantasma**

Esta seção apresenta o problema de reconciliação de EPCs fantasmas. Na seção seguinte, a solução para os dois estudos de caso é apresentada em conjunto.

#### **5.2.1.**

##### **O problema**

Novas tecnologias sempre oferecem novos desafios. Com sua criação e principalmente com os testes de utilização e instalações realizadas, problemas previstos são finalmente encontrados e enfrentados, e problemas não previstos aparecem e precisam ser gerenciados.

Com a tecnologia de RFID/EPC também não foi diferente. Por se tratar de uma tecnologia que faz uso de rádio frequência, diversos problemas florescem. Um exemplo é o problema de colisão de leitora.

Um número considerável de leitoras deve ser instalado num armazém ou centro de distribuição, já que as leitoras RFID possuem um raio finito de alcance (nos padrões de hoje, alguns metros) de capacidade de leitura de etiquetas RFID. Quando mais de uma leitora é instalada lado a lado, esta configuração enfrenta o problema de colisão de leitora (*reader collision problem* [11]). Este problema ocorre quando a área coberta por uma leitora possui uma interseção com a área coberta por outra leitora. Quando há esta interseção, as leitoras podem interferir uma com a outra, chegando ao ponto de impossibilitar o correto funcionamento da outra. Este problema está sendo enfrentado por desenvolvedores de leitoras, e não é o foco deste estudo de caso.

Outro problema provindo da utilização de RFID é o problema de reconciliação de etiquetas fantasmas. Existem cenários nos quais este problema pode acontecer. Os próximos dois parágrafos apresentam dois cenários que ajudam na explicação do problema apresentado no parágrafo seguinte.

Um primeiro exemplo é um centro de distribuição que requer que determinado produto tenha um fluxo pré-determinado em seu estabelecimento. Para que o fluxo do produto seja observado e comparado com o fluxo que ele deveria seguir, diversas leituras devem ser feitas e sistemas devem processar estes dados e coordenar que leitoras devem procurar por este produto. Caso o fluxo não seja seguido, o sistema deve tomar alguma providência. Neste cenário ideal, todas as etiquetas são lidas a qualquer momento.

Outro cenário em que o problema de reconciliação de etiquetas fantasmas ocorre é como se segue: imagina-se duas empresas parceiras, um fornecedor e um cliente, que possuem infra-estrutura RFID e certo nível de integração para envio prévio de informações de entregas do fornecedor para o cliente. Quando o fornecedor prepara seu carregamento e faz a expedição do mesmo, uma notificação prévia de carregamento (*Advanced Shipping Notice – ASN*) é enviada para o cliente contendo a informação exata de quais etiquetas RFID/EPC estão presentes no carregamento em questão, o dia esperado de chegada, entre outras. Com esta informação, o cliente pode efetuar uma verificação automática no momento de recebimento da mercadoria. Mais uma vez, neste cenário ideal todas as etiquetas são lidas a qualquer momento.

Porém, em se tratando de rádio frequência, interferências podem acontecer, embalagens podem absorver radiação, assim como conteúdo líquido, ou a radiação pode sofrer desvios. Motivos não faltam para que determinada etiqueta não consiga ser lida em determinado momento. Portanto, quando o cliente do segundo cenário descrito acima tentar ler as etiquetas do carregamento sendo recebido e conseguir ler apenas algumas das etiquetas, o que deve ser feito? Deve-se considerar que o produto está faltando? Deve-se assumir que a etiqueta está com problemas? Ou assume-se que ela está presente, mas que apenas não se conseguiu ter uma leitura naquele momento.

Quando se espera ler uma etiqueta em um dado momento por uma dada leitora, caso não se consiga, pode-se assumir que problemas de leitura aconteceram, mas que provavelmente o produto se encontra naquele lugar. No segundo cenário, os produtos são recebidos na área de recebimento, podem ser averiguados, e posteriormente são encaminhados para o local apropriado de estocagem. É provável que o produto, uma vez em outro local, devidamente

armazenado, seja lido por outra leitora. Como a leitura no local de recebimento não teve sucesso, e conseqüentemente não se pode persistir a informação de que o produto foi recebido, é preciso que a segunda leitora divulgue a informação de que aquele produto foi lido por ela, e que ele não é mais um produto faltante, não é mais uma etiqueta fantasma.

Portanto, quando não se consegue ler um produto com uma etiqueta RFID/EPC que se espera ler em determinado local e data, diz-se que este produto (EPC) é um EPC fantasma.

No momento que se tenta ler um produto e verifica-se que a leitura não pôde ser feita, alguma ação precisa ser tomada. Uma delas é fazer com que demais leitoras tenham conhecimento da falta deste produto para que, caso uma delas faça uma leitura com sucesso deste produto, o mesmo seja reconciliado com o sistema (ele não é mais um produto faltante e foi visto em determinado local e hora).

### **5.3. Solução dos estudos de caso**

Para apresentar a solução proposta para os estudos de caso descritos na seção anterior, a próxima seção mostra os casos de uso, seguidos de suas descrições e posteriormente os diagramas de classe.

#### **5.3.1. Casos de Uso**

A Figura 16 mostra o digrama de casos de uso para os estudos de caso.

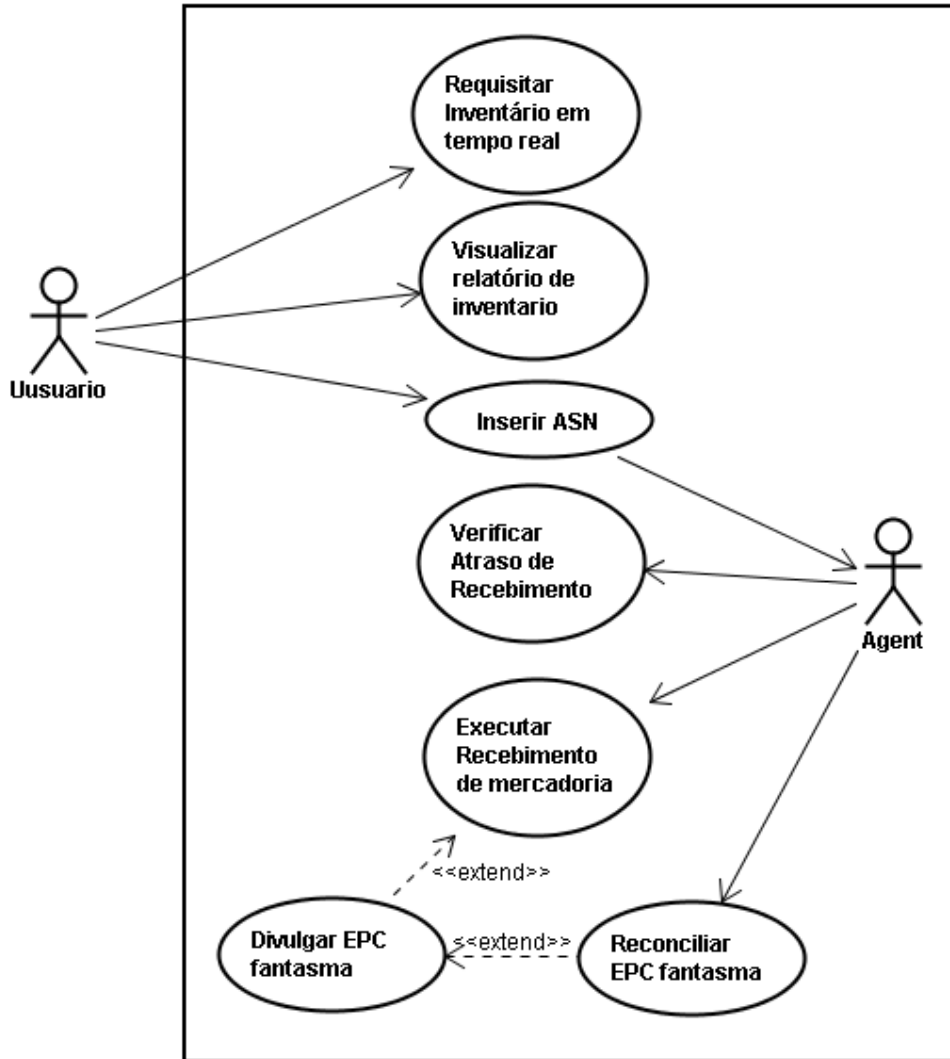


Figura 16: diagrama de casos de uso dos estudos de casos

#### Descrição dos Casos de Uso:

##### 1) Requisar Inventário em tempo real.

- Descrição: com a aplicação Shine sendo executada, o usuário deve clicar no botão que inicia o processo de inventário. O sistema deve requisitar a todas as leitoras disponíveis uma leitura de todas as etiquetas presentes naquele momento. Com a informação desta leitura, o sistema deve gerar um relatório para ser visualizado pelo usuário.
- Pré-requisitos: todos os Nodes devem estar em execução no momento da requisição do inventário.
- Fluxo de eventos:

- Usuário requisita inventário.
- O sistema, composto por diversos Nodes distribuídos, faz uma leitura das etiquetas sendo “visualizadas” naquele momento.
- O sistema gera relatório de inventário.

## 2) Visualizar relatório de inventário

- Descrição: Este caso de uso deve apenas apresentar o relatório gerado pelo caso de uso anterior.
- Pré-requisitos: o relatório deve estar disponível para ser apresentado (deve ter sido gerado pelo caso de uso anterior).

## 3) Inserir ASN

- Descrição: este caso de uso tem o objetivo de fornecer ao sistema uma Advanced Shipping Notice (ASN) para que a aplicação possa esperar o recebimento da mercadoria descrita na ASN. O usuário deve poder inserir a ASN no sistema.
- Fluxo de eventos: o usuário acessa a aplicação Shine, seleciona a opção de inserção de ASN e insere a mesma.

## 4) Verificar Atraso de recebimento

- Descrição: o sistema deve verificar, a partir dos dados da ASN inserida pelo caso de uso anterior, se há alguma mercadoria que deveria ser entregue, porém se encontra com atraso de entrega. Em caso positivo, o sistema deve notificar o usuário.
- Pré-requisitos: deve existir alguma ASN previamente inserida no sistema.
- Fluxo de eventos:
  - O sistema verifica constantemente se há atraso de entrega de mercadorias para as ASNs presentes no sistema.
  - Caso haja atraso de entrega, o sistema deve notificar o usuário. A forma de notificação pode ser por envio de e-mail, por exemplo.

## 5) Executar Recebimento de Mercadoria

- Descrição: o usuário deve poder realizar uma ação no sistema de “recebimento de mercadoria”. Ao executar esta função, o sistema deve realizar uma leitura das etiquetas presentes na localização de recebimento.
- Pré-requisitos:
  - Deve existir um arquivo de configuração de ECspec específico para recebimento de mercadorias, que irá definir quais são as leitoras localizadas na região de recebimento.
  - A ASN deve estar inserida no sistema.
- Fluxo de eventos:
  - Após a inserção da ASN, o sistema está preparado para esperar o recebimento da mercadoria descrita nela.
  - Sempre que o usuário desejar, ele deve poder iniciar o processo de recebimento, clicando no comando correspondente no sistema.
  - Ao executar esta função, o sistema deve realizar uma leitura das etiquetas presentes na região coberta pelas leitoras definidas no ECspec de recebimento.
  - Caso haja algum EPC sendo lido que pertença aos EPCs listados na ASN, o sistema deve informar que a ASN correspondente está sendo recebida.

#### 6) Divulgar EPC fantasma

- Descrição: durante o processo de recebimento descrito acima, ao se verificar que determinada ASN está sendo recebida, deve-se verificar se todos os EPCs sendo esperados estão sendo lidos no momento da leitura do recebimento. Caso esteja faltando algum EPC, deve-se divulgar esta informação para os demais Nodes do sistema.
- Pré-requisitos: é preciso haver ao menos um EPC sendo esperado (presente na ASN) que não esteja sendo lido pelas leitoras para que o processo de divulgação deste “EPC fantasma” se inicie.
- Fluxo de eventos:
  - Percebendo que determinada ASN está sendo recebida (final do caso de uso anterior), o sistema deve verificar se todos os EPCs presentes na ASN estão sendo lidos.



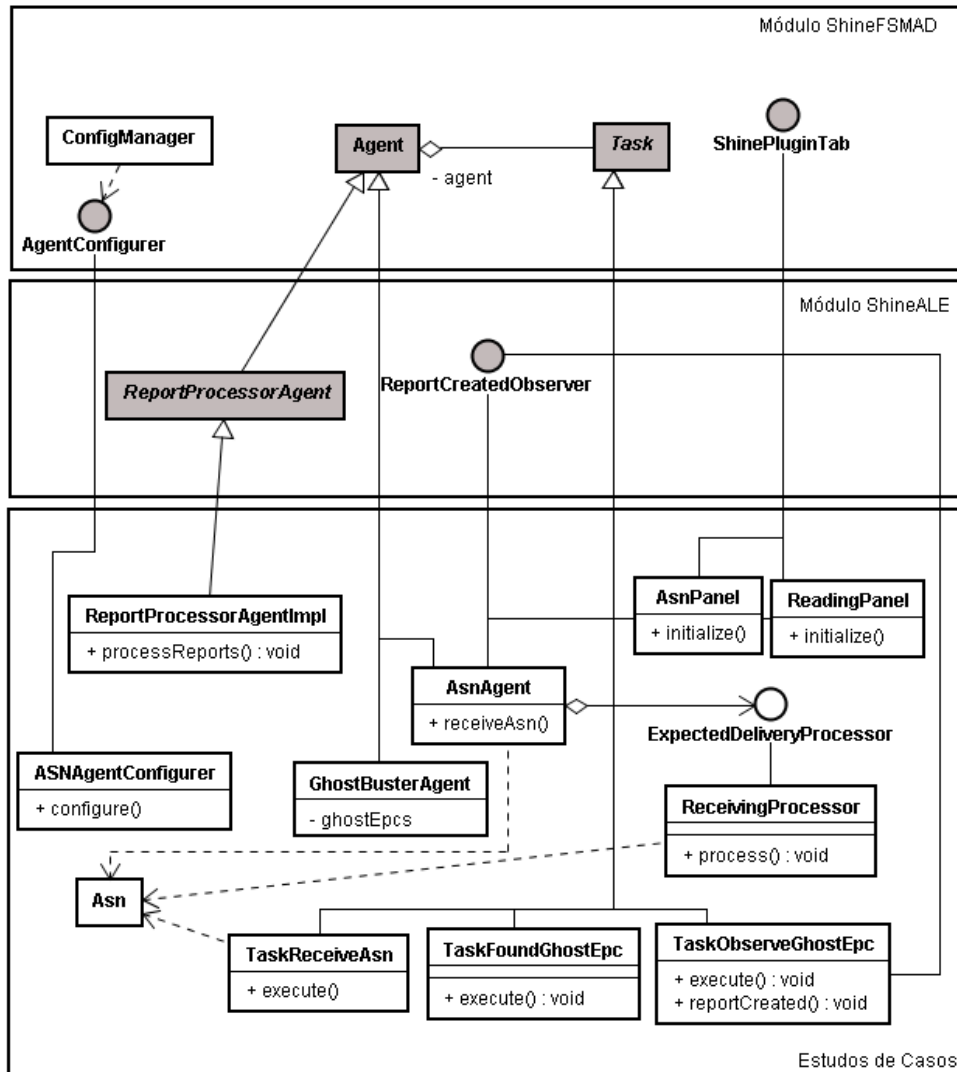
- Caso algum EPC não esteja presente na lista de EPCs lidos, o Node efetuando o processo de recebimento deve divulgar a lista de EPCs fantasmas: esperados porém não sendo lidos.
- O sistema envia mensagem contendo a lista de EPCs fantasmas para todos os demais Nodes.
- Os demais Nodes, recebendo esta mensagem, ficam monitorando, a cada leitura realizada, para verificar se algum dos EPCs fantasmas estão presentes na lista de EPCs sendo lidos em algum momento.

#### 7) Reconciliar EPC fantasma

- Descrição: quando um Node realiza uma leitura qualquer e percebe que um EPC fantasma está presente na lista de EPCs sendo lidos, é preciso enviar mensagem a todos os outros Nodes com o objetivo de informar que o determinado EPC não é mais um fantasma, que foi encontrado e se encontra em determinado local (reconciliação).
- Fluxo de eventos:
  - Um Node, realizando uma leitura em algum momento, e tendo uma lista de EPCs fantasmas, verifica se os EPCs sendo lidos contém algum EPC que é um “fantasma”.
  - Caso haja algum EPC sendo lido pertencente a lista de EPCs fantasmas deste Node, o sistema deve realizar a reconciliação.

### 5.3.2. Projeto e Implementação dos Estudos de Caso

Os dois estudos de casos descritos neste capítulo foram implementados, naturalmente, estendendo o *framework* Shine. Os *hot spots* implementados são apresentados na Figura 17, destacados em cinza. Esta figura apresenta todas as classes implementadas para a realização dos estudos de casos. Isso mostra o quanto o *framework* facilitou sua implementação e mostra como foi possível reutilizar toda a infra-estrutura fornecida por ele. As entidades destacadas em cinza são os *hot spots* que foram implementados pelos estudos de casos.



**Figura 17: diagrama de classes dos estudos de casos**

Além dessas classes apresentadas na figura, foi preciso criar arquivos XML de ECspecs (especificação de ciclos de eventos) e de configuração dos Nodes (foram criados dois Nodes). Nos estudos de caso em questão, foi levado em consideração que existiam duas leitoras lógicas RFID (podendo representar n leitoras físicas na realidade). Portanto, para que se tivesse um inventário, era preciso requisitar uma leitura de um ciclo de evento a essas duas leitoras, como mostra a Tabela 3. Esta tabela, apresenta o ECspec para o caso de inventário. Além disso, foi preciso criar os arquivos de configuração dos dois Nodes usados nos estudos de casos. Os arquivos XML de configuração destes dois Nodes podem ser encontrados nos ANEXOS II e III.

Para o estudo de caso em questão, foi criada uma ASN simples contendo algumas informações de produtos sendo entregues. A ASN pode ser vista no ANEXO IV, enquanto que a especificação de ASNs que podem ser usadas com este estudo de caso é apresentada em forma de XML Schema no ANEXO V.

```
<?xml version="1.0" encoding="UTF-8"?>
<ale:ECSpec xmlns:ale="urn:epcglobal:ale:xsd:1"
  xmlns:epcglobal="urn:epcglobal:xsd:1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:epcglobal:ale:xsd:1 file:/C:/ ale.xsd"
  schemaVersion="1.0" >
  <logicalReaders>
    <logicalReader>ReceivingDockDoorReader</logicalReader>

    <logicalReader>Node2Reader</logicalReader>
  </logicalReaders>
  <boundarySpec>
    <repeatPeriod unit="MS">20000</repeatPeriod>
    <stableSetInterval unit="MS">5000</stableSetInterval>
  </boundarySpec>
  <reportSpecs>
    <reportSpec reportName="InventoryReport">
      <reportSet set="CURRENT"/>
      <output includeList="true"/>
    </reportSpec>
  </reportSpecs>
</ale:ECSpec>
```

**Tabela 3: ECSpec de inventário para as leitoras existentes no cenário do estudo de caso.**

A seção a seguir apresenta em detalhes as classes implementadas para os estudos de casos.

### 5.3.2.1. Descrição das classes implementadas

Seguindo as classes apresentadas na Figura 17, abaixo são apresentados detalhes das mesmas a fim de se mostrar como o framework ajudou na solução dos estudos de casos.

#### (I) Agentes Implementados:

- 1) ReportProcessorAgentImpl: esta classe implementa o *hot spot* ReportProcessorAgent do módulo ShineALE. Este *hot spot* é uma classe abstrata e define o método abstrato `processesReports()`. A classe implementa este método de forma a apenas informar o usuário que os relatórios estão sendo processados. Nenhum processamento a fim de agrupar EPCs, ou quantificar classes de EPCs, por exemplo, foi

implementado, mas poderia ser, facilmente. Esta classe serviu apenas como prova da aplicação do *hot spot*.

- 2) `ASNAgent`: esta classe implementa o agente responsável por receber e processar uma ASN (*Advanced Shipping Notice*). A forma implementada para o envio da ASN para este agente foi através da interface com o usuário: o usuário informa a ASN pela interface, que por sua vez envia uma mensagem para este agente repassando a ASN em questão. O agente, então, se adiciona como um observer [18] da criação de relatórios para poder ficar verificar se em cada relatório criado há EPCs sendo lidos que pertençam ao carregamento sendo aguardado, definido pela ASN. Caso haja algum EPC especificado na ASN sendo lido no relatório, este agente considera a ASN como recebida e verifica se todos os EPCs esperados foram recebidos. Caso algum EPC esperado não tenha sido recebido (não esteja presente no relatório de leitura), este agente executa os `ExpectedDeliveryProcessors` que ele conhece (esta entidade será detalhada abaixo).
- 3) `GhostBusterAgent`: a principal característica deste agente é o seu conhecimento a respeito de EPCs fantasmas que ele deve observar, “caçar”. Ele tem o controle sobre a lista de EPCs fantasmas sendo procurados pela aplicação. Além disso, ele é um observer da geração de relatórios, e a cada relatório criado, ele verifica se algum EPC fantasma foi encontrado. Caso tenha sido encontrado, ele retira este EPC da lista de fantasmas e envia mensagem para os demais `GhostBusterAgents` de outros Nodes, para que estes também parem de procurar por este EPC encontrado (reconciliação do EPC).

#### (II) Tarefas implementadas:

- 1) `TaskReceiveAsn`: esta tarefa é executada pelo `ASNAgent`, e tem a função de criar um objeto do tipo `Asn` a partir da descrição de ASN, em XML, recebida pela interface com o usuário. Após criar este objeto, esta tarefa executa o método de recebimento de ASN do agente `ASNAgent`.
- 2) `TaskObserveGhostEpc`: esta tarefa é responsável por adicionar os EPCs fantasmas à lista de fantasmas do `GhostBusterAgent`. Além disso, ela coloca o `GhostBusterAgent` como um Observer da

geração de relatórios, para que ele possa observar os relatórios sendo criados e procurar por EPCs fantasmas, verificando se eles foram achados.

- 3) `TaskFoundGhostEpc`: esta tarefa é executada após o recebimento da mensagem provinda do agente `GhostBusterAgent`, que contém informação sobre EPCs fantasmas que foram encontrados. Ela é executada por outros agentes `GhostBusterAgent` de outros Nodes. Seu objetivo é retirar os EPCs recebidos das listas de EPCs fantasmas, afinal eles foram encontrados.

### (III) Demais entidades:

- 1) `ExpectedDeliveryProcessor`: esta interface define processadores de entregas esperadas. Quando o `ASNAgent` encontra EPCs que eram esperados na entrega, porém não foram encontrados no relatório de leitura (no caso de algum EPC da ASN ter sido encontrado), ele executa todas as classes que implementam esta interface presentes na sua lista de `ExpectedDeliveryProcessors`. Esta lista é alimentada no momento de configuração do agente, pela classe `ASNAgentConfigurer`.
- 2) `ReceivingProcessor`: esta classe implementa a interface `ExpectedDeliveryProcessor`. Ela verifica se todos os EPCs esperados da ASN estão presentes na coleção de EPCs lidos. Caso falte algum, este é um EPC fantasma e precisa ser divulgado para os agentes caçadores de fantasmas (`GhostBusterAgent`).
- 3) `ASNAgentConfigurer`: esta classe implementa a interface `AgentConfigurer` que é um *hot spot* do Shine (módulo `ShineFSMAD`). Ela é responsável por configurar o `ASNAgent` no momento de inicialização do sistema. Ela sabe ler as informações pertinentes a este agente presentes no arquivo XML de configuração do Node e atribuir estes valores ao agente.
- 4) `Asn`: esta é a entidade do domínio de negócio de ASN. Ela representa uma ASN real como um objeto.
- 5) `AsnPanel`: esta classe implementa o *hot spot* `ShinePluginTab` do *framework* Shine. Ela é responsável pela interface com usuário (GUI) para a inserção de ASN no sistema.

- 6) `ReadingPanel`: esta classe implementa o *hot spot* `ShinePluginTab` do *framework* Shine. Ela é responsável pela interface com usuário (GUI) para apresentação dos EPCs sendo lidos. Ela também é responsável por iniciar a execução de um ciclo de evento (`EventCycle`) oferecendo ao usuário uma lista de possíveis `ECspecs` a serem executadas e enviando mensagem para o agente `RequestsManagerAgent` (módulo ShineALE) para que este gerencie a execução deste ciclo de evento. Esta é a única classe implementada para o estudo de caso de inventário em tempo real.

Percebe-se pelas descrições das classes e interfaces implementadas que a maioria delas foi implementada visando o estudo de caso de EPC fantasma. Para o estudo de caso de inventário, precisou-se implementar apenas a classe de interface `ReadingPanel` (descrita acima), e os arquivos XML de especificação de ciclo de evento (`ECspec`).

As figuras apresentadas a seguir (Figura 18 e Figura 19) apresentam os diagramas de seqüências dos casos de uso relacionados ao estudo de caso de inventário em tempo real. A seqüência da Figura 19 se inicia a partir do recebimento da mensagem enviada no diagrama da Figura 18. A seqüência de envio de mensagem não é apresentada nestas figuras, porém já foi mostrada na Figura 11, da seção 4.6.1.2.

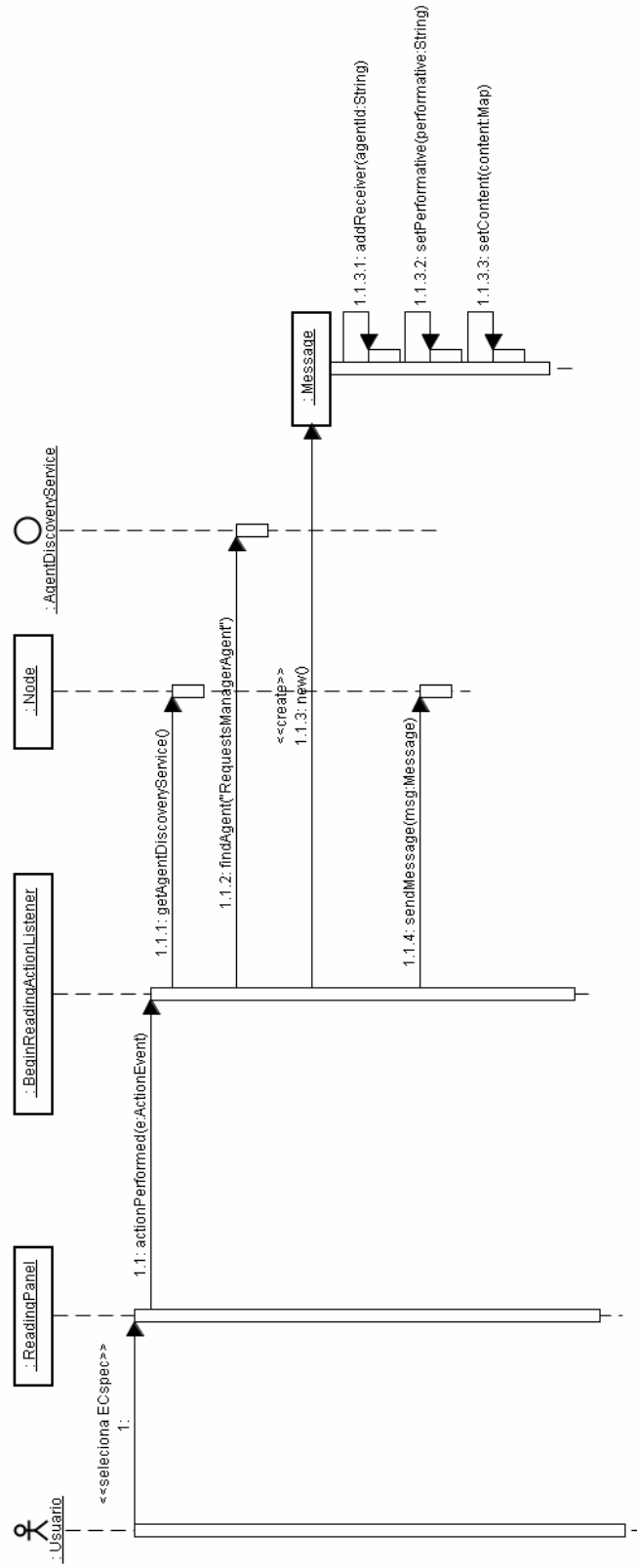


Figura 18: diagrama de seqüências do inventário (parte 1)

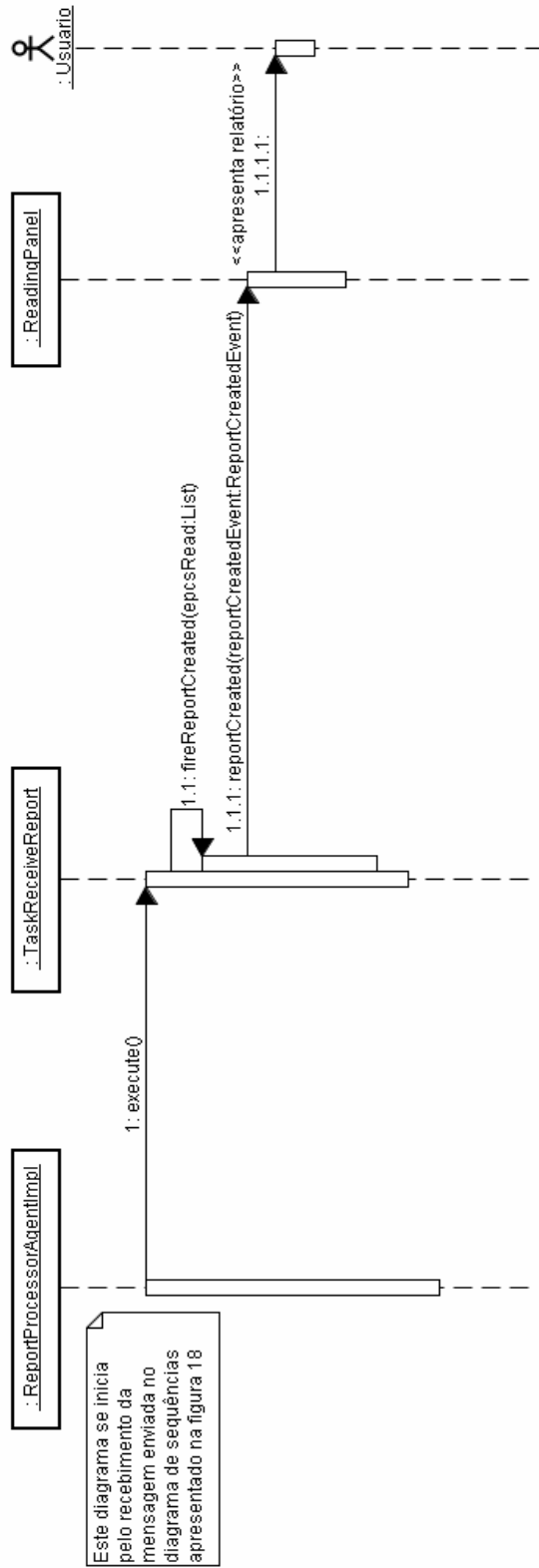
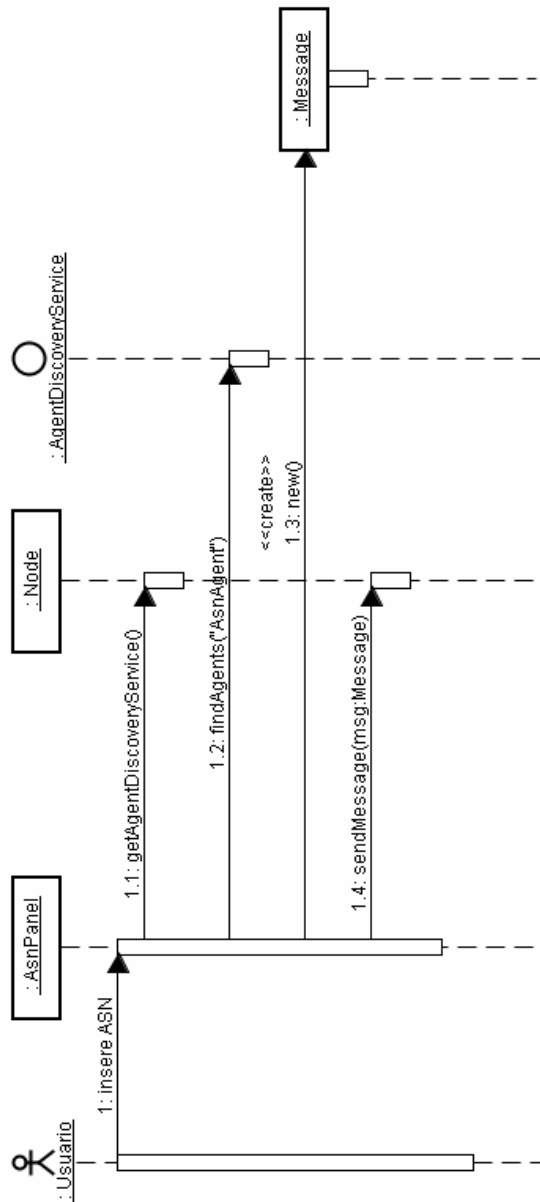


Figura 19: diagrama de seqüências do inventário (parte 2)



Abaixo são apresentadas as figuras referentes aos casos de uso para o estudo de caso de EPCs fantasmas. A Figura 20 apresenta o diagrama de seqüências da interação do usuário com o sistema para a inserção de uma ASN.



**Figura 20: diagrama de seqüências do estudo de caso de EPCs fantasmas (parte 1)**

A Figura 21 apresenta o diagrama de seqüências que é continuação do diagrama da Figura 20. Ele se inicia através do recebimento da mensagem

enviada na Figura 20 para o `AsnAgent`. O último método apresentado neste diagrama de seqüências da Figura 21 verifica se há algum EPC esperado que não foi recebido. Se sim, envia mensagem para os agentes `GhostBusterAgent`, informando o EPC fantasma. A continuação desta seqüência é apresentada na Figura 22.

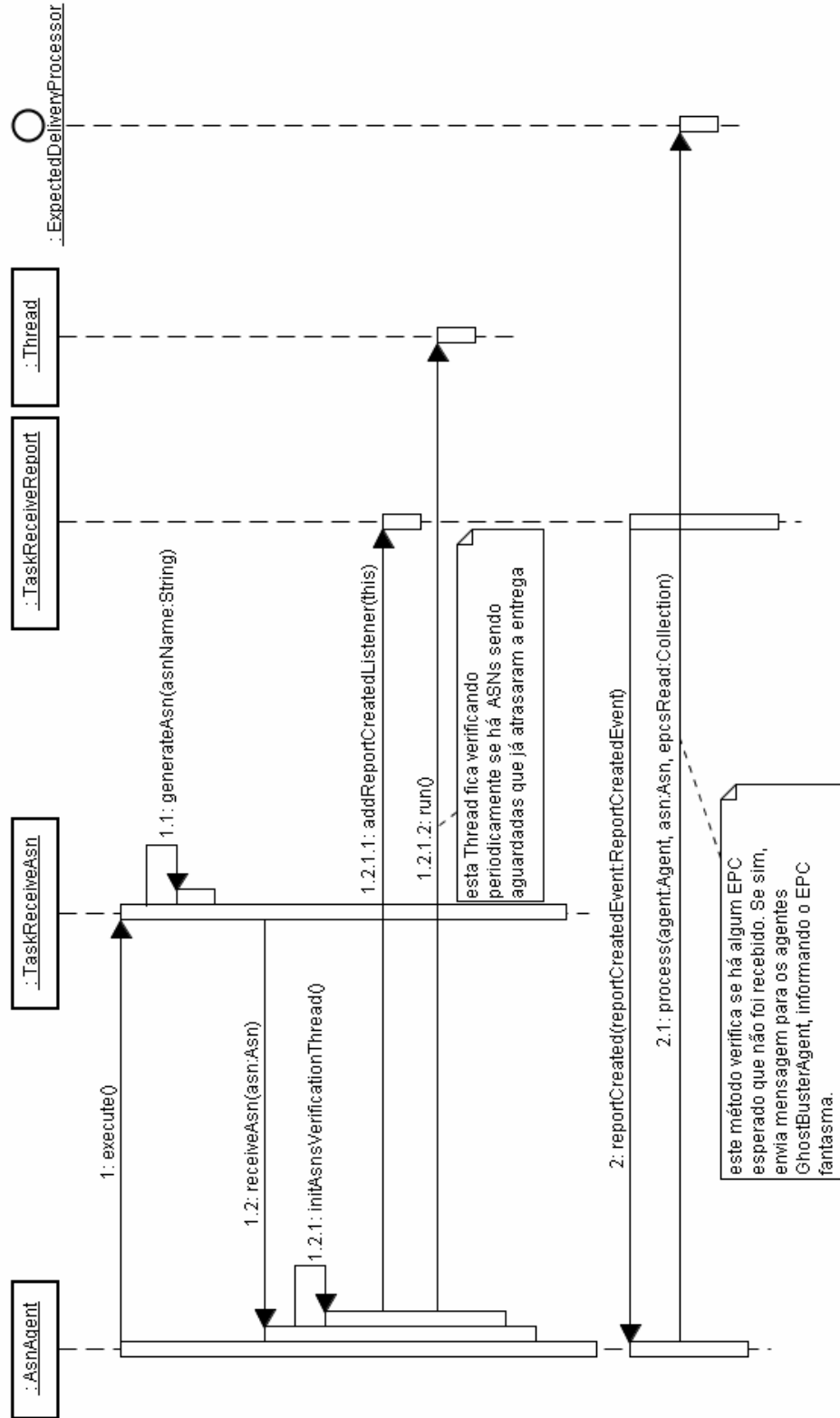


Figura 21: diagrama de seqüências do estudo de caso de EPCs fantasmas (parte 2)

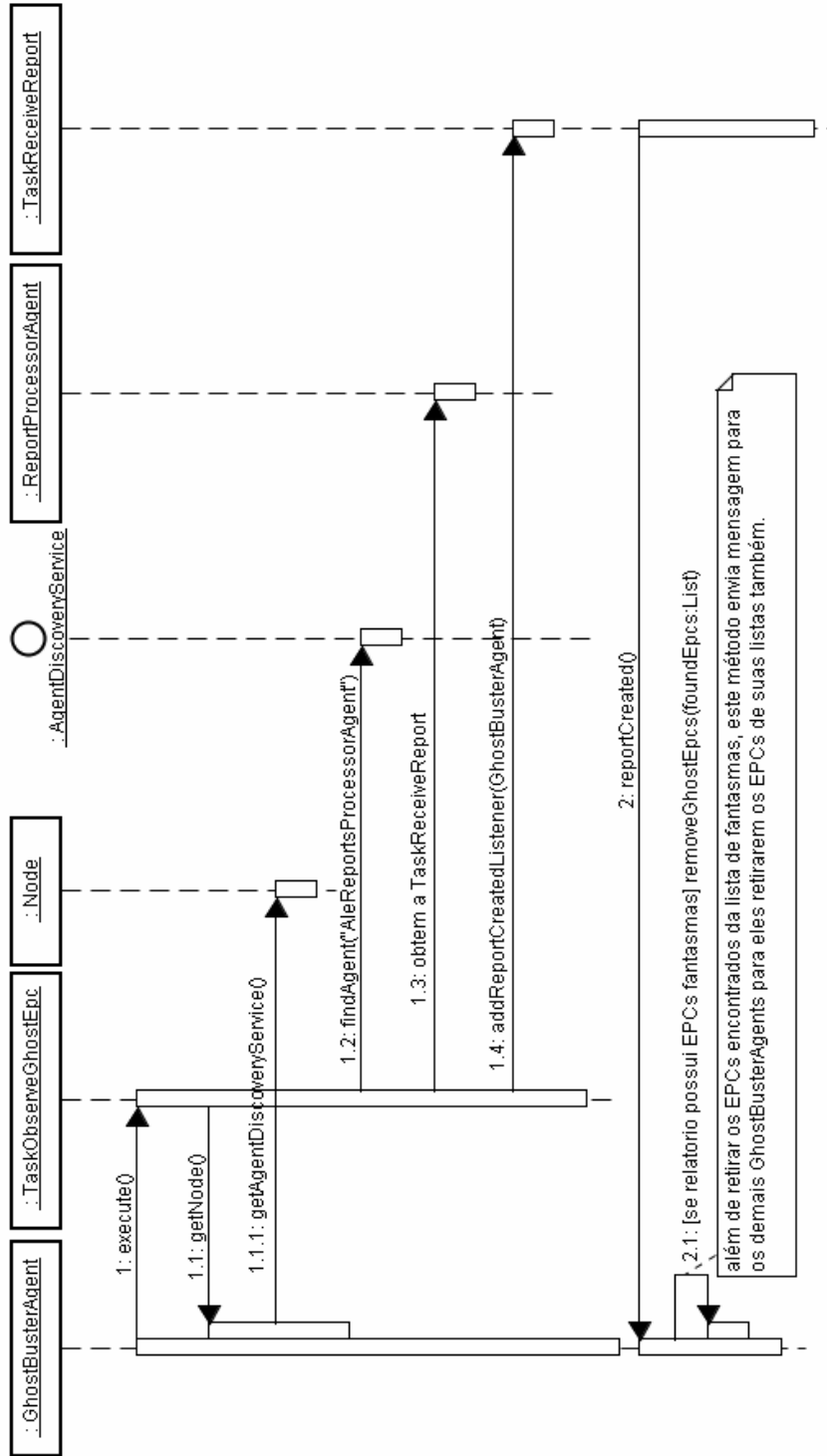


Figura 22: diagrama de seqüências do estudo de caso de EPCs fantasmas (parte 3)

Portanto, com a implementação das classes apresentadas na Figura 17 e com a definição do ECspec apresentado na Tabela 3, conseguiu-se, com pouco esforço de codificação, instanciar o *framework* Shine com o intuito de implementar uma aplicação que executa os casos de uso descritos na seção 5.3.1 deste capítulo. Foi possível, portanto, mostrar a aplicação eficiente de sistemas multi-agentes para o domínio de RFID, além de mostrar a implementação de um sistema multi-agentes usando o *framework* aqui proposto.