

SISTEMA PARA SIMULAÇÕES DE AMBIENTE RFID

Vinicius José Schmidt

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
vjschmidt@faccat.br

Marcelo Cunha de Azambuja

Professor Orientador
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
azambuja@faccat.br

Resumo

Este artigo tem por objetivo apresentar o desenvolvimento de um sistema que permite a simulação de ambientes RFID (*Radio Frequency Identification*). O sistema foi desenvolvido com o intuito de auxiliar empresas e outros pesquisadores a simularem sistemas RFID sem que os mesmos necessitem ser fisicamente construídos. Como principais características, o sistema possui um modelo de simulação RFID que considera todos os parâmetros básicos de transmissão de rádio-frequência e também fatores como o tipo de material no qual a etiqueta RFID está fixada. O sistema também foi construído de forma a tornar independente o processamento de simulação com a interface gráfica onde o usuário projeta o ambiente a ser simulado. O sistema possui três formas distintas para aceitar as requisições de simulações feitas a ele: *webservice* síncrono, *webservice* assíncrono e *servlet*. O sistema aqui apresentado integra o conjunto de pesquisas sobre RFID do Polo de Inovação Tecnológica do Paranhana/Encosta da serra, sediado na FACCAT (Faculdades de Taquara).

Palavras-chave: RFID, *webservice*, *servlet*, simulação.

SYSTEM FOR RFID ENVIRONMENT SIMULATIONS

Abstract

This paper presents the RFID (Radio Frequency Identification) environment simulation system development. The system was developed in order to assist companies and researchers to simulate RFID environments without physically building them. As main features, the server have a RFID simulation model that considers all the basic radio frequency communication parameters and also factors such as the type of material that the RFID tag is attached. The system was also built to make the simulation process independent of the graphical simulation interface where the user builds the environment to be simulated. The system has three different ways to accept the simulation requests made to it: synchronous webservice, asynchronous webservice and servlet. The system here presented integrates the RFID research group at the Paranhana/Mountainside's Technological Innovations pole at FACCAT (Taquara's College).

Key-words: RFID, *webservice*, *servlet*, simulation.

1. Introdução

A tecnologia de identificação por rádio-frequência RFID (*Radio Frequency Identification*), é uma solução de identificação de dispositivos que utiliza sinais de rádio-frequência emitidos por um equipamento leitor e etiquetas identificadoras anexadas ao que deseja-se gerenciar no ambiente em questão. Essa tecnologia permite que a identificação de objetos ocorra sem que o aparelho leitor esteja em perfeito alinhamento com os itens a serem lidos (FINKENZELLER, 2010).

Bhatt e Glover (2006) afirmam que a utilização da tecnologia cresce com o passar do tempo, sendo utilizada em grandes corporações como Wal-Mart e Tesco e até mesmo pelo Departamento de Defesa dos Estados Unidos. Mas a tecnologia ainda apresenta desafios para aqueles que desejam utilizá-la. Jung *et al.* (2011) definem os principais desafios da tecnologia, a saber: (i) colisão de sinais emitidos pela etiqueta eletrônica quando mais de uma responde ao equipamento leitor ao mesmo tempo, (ii) a interferência de sinal RF provenientes do material onde as etiquetas estão fixadas, objetos e ruídos no ambiente e reflexão de sinal, (iii) a baixa potência retornada das etiquetas passivas ao equipamento leitor, o que faz com que seu raio de comunicação diminua.

A partir desses problemas, uma forma de simular o processo de comunicação em ambientes RFID foi proposta, visando integrar sistemas que modelam ambientes RFID com um modelo matemático de predição de comunicação da tecnologia RFID. Dessa ideia o servidor proposto foi idealizado e desenvolvido. Este artigo apresenta os resultados do desenvolvimento de tal servidor, apresentando todas as fases do projeto, bem como resultados obtidos através das simulações do mesmo.

O trabalho está organizado da seguinte maneira: a seção 2 apresenta o estado da arte, a seção 3 mostra os trabalhos relacionados, a seção 4 descreve as tecnologias utilizadas, a seção 5 apresenta a metodologia aplicada, a seção 6 traz os resultados obtidos e a seção 7 traz as conclusões do artigo.

2. Referencial teórico

2.1 RFID

Conforme Bernardo (2004), a tecnologia de identificação por rádio-frequência RFID foi desenvolvida com o intuito de automatizar a detecção e rastreamento de produtos, e criada com o objetivo de melhorar o processo de leitura por código de barras removendo, em algumas das frequências utilizadas, a necessidade da proximidade entre o aparelho leitor e a etiqueta identificadora e a intervenção humana no processo. A tecnologia foi desenvolvida em uma parceria entre o *Massachusetts Institute of Technology* (MIT) e outros centros de pesquisa na década de 1980. A partir dessa pesquisa surgiu o Código Eletrônico de Produtos – EPC, em inglês *Electronic Product Code*.

A tecnologia RFID, como descrevem Bhatt e Glover (2006), é dividida entre a camada física e a camada de aplicação. Enquanto na camada física encontram-se um leitor e etiquetas de identificação cada qual com sua devida antena para a comunicação entre eles a camada de aplicação é composta por (i) uma aplicação de interface de programação, (ii) um sistema de comunicação, (iii) um gerenciador de eventos e (iv) um subsistema para as antenas da leitora.

Azambuja (2011) explica que o canal de comunicação da leitora é compartilhado com todas as etiquetas ativas no ambiente. Os protocolos anti-colisão são os responsáveis pela divisão (multiplexação) desse canal de comunicação entre as diversas etiquetas presentes no ambiente, os protocolos serão detalhados posteriormente neste trabalho.

De acordo com Finkenzeller (2010) existem dois tipos principais de etiquetas RFID as etiquetas ativas e as passivas. As etiquetas ativas possuem sua própria fonte de energia proveniente de uma bateria ou de um painel solar e não utilizam a energia transmitida do aparelho leitor para ativar seus componentes. As etiquetas passivas não possuem fonte própria de energia, utilizando a energia transmitida do aparelho leitor para ativar seus componentes; esse tipo de etiqueta é alimentado com a energia do campo de RF emitido pela leitora. Apesar dessa diferença, os dois tipos emitem informações somente após terem sido acionadas por um equipamento leitor. De acordo com Radio Electronics (2012) as frequências comumente utilizadas pela tecnologia RFID são: (i) 125 até 134,2 kHz e 140 até 148,5 kHz que alcança até 0,5 metros; (ii) 13,553 até 13,567 MHz e 26,957 até 27,283 MHz possui um alcance de até 1 metro; (iii) 858 até 930 MHz possui um alcance de até 10 metros em média; (iv) 2400 até 2483 GHz possui um alcance de a partir de 3 metros. O que determina o uso de cada frequência são o alcance de leitura que deseja-se possuir para o sistema e leis de regulamentação regionais de alocação do espectro de sinal do rádio, cada região possui um controle individual da frequência, principalmente no espectro UHF (*Ultra High Frequency*, corresponde a 858 até 930 MHz), esse controle é necessário para que não ocorra interferência entre as tecnologias que utilizam esse meio para comunicação. Neste trabalho, o foco são as etiquetas passivas que atuam na frequência UHF, em especial, em 915 MHz.

2.2 Componentes da camada física de um sistema RFID

A camada física, conforme Azambuja (2011) é a camada composta pelos equipamentos físicos (*hardwares*) que compõem o sistema RFID, a leitora, as etiquetas e suas antenas. Nela o processo de comunicação entre leitora e etiquetas inicia-se, a leitora energiza as etiquetas no seu raio de alcance através da propagação do sinal de rádio-frequência, com a energia recebida a etiqueta a utiliza para ativar seus componentes e com isso a etiqueta pode então transmitir seus dados.

2.2.1 Leitora RFID

Finkenzeller (2010) afirma que a leitora num sistema RFID possui três papéis importantes para o funcionamento da leitura dos dados: (i) é responsável por ativar as etiquetas passivas que estão no alcance do sinal de rádio-frequência (as etiquetas absorvem energia obtida através do campo eletromagnético da leitora e utilizam esta energia para tornarem-se ativas), (ii) controla a recepção e interpretação dos dados enviados pelas etiquetas e (iii) faz a comunicação com o servidor da aplicação que está acoplado ao ambiente RFID.

A Figura 1 apresenta os dois estados de funcionamento das antenas de uma leitora transmissão de sinal RF e recepção do sinal.

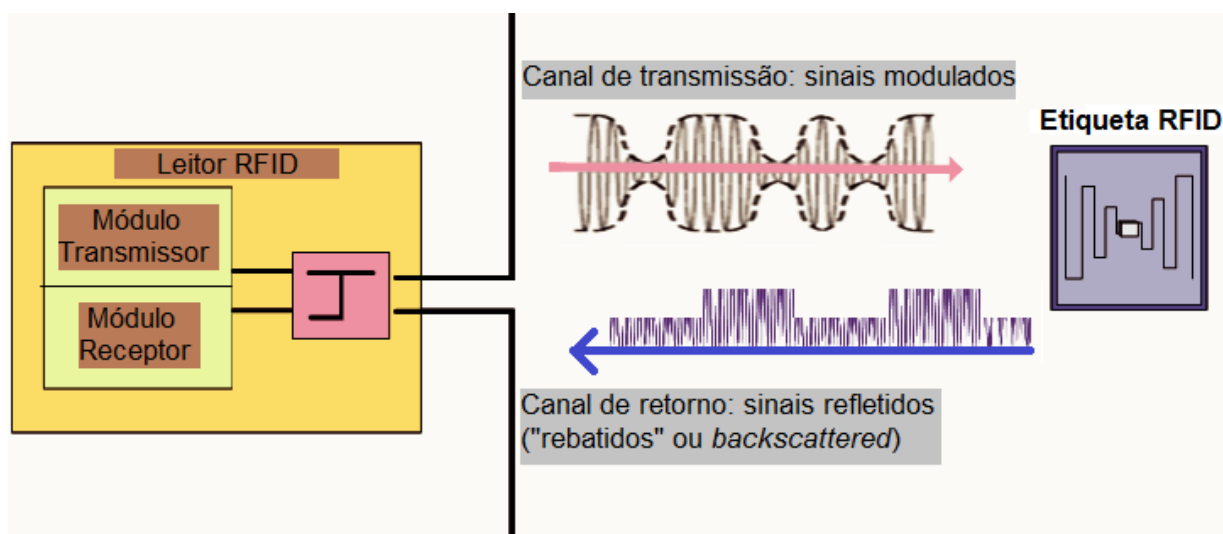


FIGURA 1: Comunicação entre etiqueta e leitora RFID

Fonte: Jung *et al.* (2011)

Como pode ser visto na Figura 1, o sinal transmitido pelo módulo transmissor atinge uma etiqueta no seu raio de ação e esta rebate o sinal de volta para o aparelho leitor. O canal de comunicação entre etiquetas e leitoras é compartilhado por todos os equipamentos que compõem o ambiente RFID (JUNG *et al.*, 2011; AZAMBUJA 2011).

2.2.2 Etiquetas RFID

As etiquetas de identificação RFID, segundo Azambuja (2011), são as responsáveis por identificar os objetos desejados no ambiente. Elas são os dispositivos que possuem os dados a serem lidos pela leitora e respondem aos sinais emitidos por ela. Uma etiqueta para conseguir transmitir seus dados para a leitora, se for do tipo passiva, precisa, após utilizar a energia absorvida pelo campo magnético, ter carregado potência o suficiente para que seus dados sejam transmitidos novamente ao alcance da leitora. A antena é a maior parte da etiqueta, enquanto seu chip,

comparado à antena, possui um tamanho insignificante. A Figura 2 apresenta um exemplo de etiqueta.

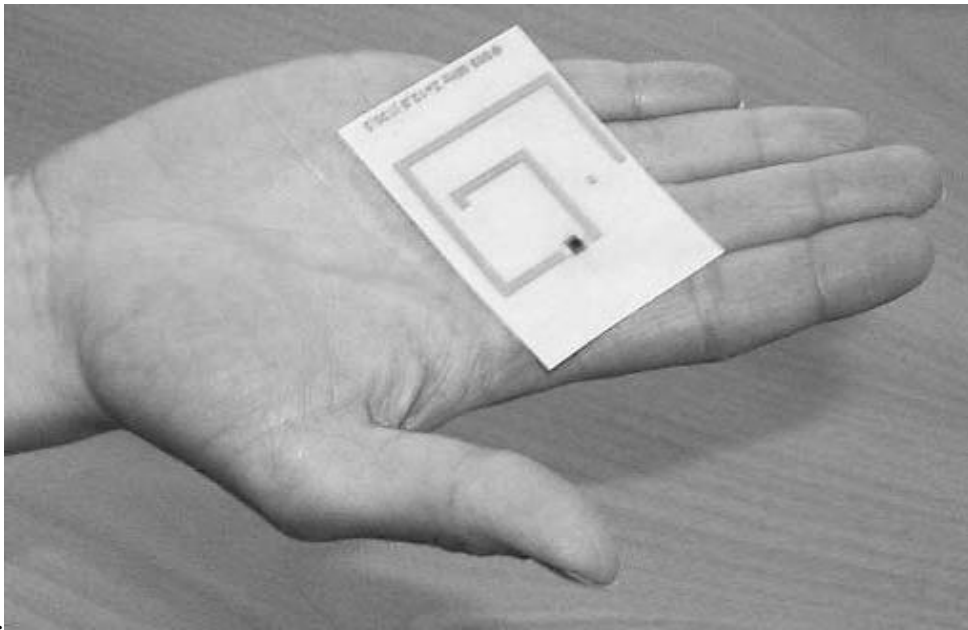


FIGURA 2: Etiqueta RFID. O ponto escuro na imagem é o chip, enquanto a trilha metálica é a antena

Fonte: Finkenzeller (2010)

2.3 Componentes da camada de aplicação de um sistema RFID

De acordo com Bhatt e Glover (2006), a camada de aplicação também é conhecida como *middleware* e é uma parte importante do funcionamento do sistema, pois é encarregada de gerenciar a comunicação correta entre os componentes da camada física.

2.3.1 Sistema de comunicação e a API

Bhatt e Glover (2006) afirmam que o sistema de comunicação é encarregado de gerenciar a comunicação da leitora com o servidor de aplicação que o sistema RFID está acoplado. É o responsável, por exemplo, por consultar os dados de um produto em uma base de dados através do código que se encontra gravado na etiqueta, estes dados são então retornados para a API (*Application Programming Interface*, em português, interface de programação de aplicativos), que apresenta então os dados obtidos pelo sistema de comunicação juntamente com os dados que a leitora coleta das etiquetas no ambiente, além de gerenciar as configurações da própria leitora.

2.3.2 Protocolo anti-colisão

Azambuja (2011) afirma que os protocolos anti-colisão são os encarregados do controle da comunicação entre as antenas da leitora com a das etiquetas no ambiente. Estes protocolos gerenciam como o leitor ou as etiquetas devem comportar-se quando os sinais enviados por mais de uma etiqueta ao mesmo tempo não interfiram no funcionamento correto do sistema e para que essa possibilidade seja minimizada.

2.4 EPC

O EPC (*Electronic Product Code*) é o código que identifica uma dada etiqueta. Seu tamanho em *bits* pode variar, mas o propósito dos dados continua o mesmo: uma parte do código é reservada para identificar a empresa e outra o objeto em questão. A estrutura do código segue a seguinte regra: a primeira parte é reservada para identificar a versão do código, a segunda o fabricante, a terceira o produto e a última parte identifica o número serial (EPC GLOBAL, 2010; JUNG *et al.*, 2011).

3. Trabalhos relacionados

De acordo com Azambuja (2011), não existem modelos de simulação de ambientes RFID em âmbito comercial, enquanto que em âmbito acadêmico apesar de existirem trabalhos sobre o assunto, nenhum possui resultados satisfatórios. Nessa mesma referência, modelos matemáticos para simulações de ambientes físicos são apresentadas, sendo esses modelos utilizados como base para o servidor de simulações proposto neste artigo.

Os modelos matemáticos para simulação de ambientes descritos por Azambuja (2011) foram utilizados para as simulações, mas tais modelos foram desenvolvidos para somente tratar valores da parte física de um sistema RFID e não levam em consideração a parte de comunicação da camada lógica e, como pode ser visto em Floerkemeier (2005), os diferentes protocolos anti-colisão possuem resultados de desempenho a serem considerados, visto que as colisões de transmissão de sinais entre as etiquetas e a leitora ocorrem com frequência. Por esse motivo, além do modelo matemático para simulações de interferências do ambiente, uma forma de simular a comunicação dos protocolos anti-colisão foi implementada no servidor. A parte responsável por simular o tempo de comunicação e o controle de colisão de sinais é baseada na estrutura do sistema RFID-Env, outro sistema criado no Polo de Inovação Tecnológica do Paranhana/Encosta da serra responsável por calcular o tempo que uma quantidade informada de etiquetas no ambiente leva em média para

serem corretamente lidas por um aparelho leitor utilizando um dos protocolos anti-colisão disponíveis, esse sistema é apresentado detalhadamente por Azambuja (2011) na seção 5.3.

De acordo com Sápiras (2011), sistemas que permitem a modelagem de ambientes RFID são escassos, na sua maioria em ambiente *desktop* e não permitem a fácil personalização dos dados do ambiente, sendo que quando possível alterar modelos de simulação e ambiente o próprio sistema deve ser alterado e não somente o modelo de simulação. Por este motivo, no mesmo artigo o sistema RF3D é proposto e apresentado pelo autor. O sistema então foi desenvolvido visando a facilitar modelagem e personalização do ambiente e ser totalmente utilizável via *web*. Ambientes complexos podem então ser facilmente modelados através desta ferramenta e esta permite que um servidor de testes utilize seus dados para simular a comunicação entre etiquetas e leitora. Tanto o projeto RF3D quanto o servidor de simulações proposto fazem parte do mesmo setor de pesquisa do Polo de Inovação Tecnológica da instituição FACCAT, mas não são mutuamente exclusivos, fazendo com que o servidor seja receptível a requisições de quaisquer outros sistemas.

Para fazer com que as simulações de ambientes fossem feitas utilizando um modelo matemático existente e para alcançar a integração entre o servidor de simulação e o sistema de modelagem de ambientes as tecnologias apresentadas na seção 4 foram utilizadas.

4. Tecnologias de desenvolvimento de software utilizadas

Java é uma linguagem de programação que tem como diferencial o funcionamento em qualquer sistema operacional sem a alteração do arquivo executável. A linguagem Java possui suporte para desenvolvimento de aplicações tanto *desktop* quanto *web* onde a estrutura da linguagem é mantida sem alterações (JAVA, 2011).

A linguagem XML (*Extensible Markup Language*) é um formato de texto flexível que visa facilitar a troca de informação através da internet sem o impedimento de linguagens ou sistemas diferentes. É uma linguagem que permite a estruturação de qualquer hierarquia de informação (XML, 2011).

O protocolo SOAP (*Simple Object Access Protocol*) é um protocolo de comunicação baseado em XML que permite que dados sejam transmitidos através do protocolo HTTP, é um protocolo que independe de linguagem de programação (SOAP, 2012).

Servlet é uma classe Java que permite desenvolver recursos baseados numa estrutura de requisição e resposta. São consideradas extensões de servidores e, apesar de aceitar qualquer forma de requisição, a forma de comunicação mais utilizada é a HTTP (*Hypertext Transfer Protocol*) (SERVLET, 2002).

De acordo com Johnson e Hoeller (2002), o Spring é um *framework* que abstrai códigos que não sejam específicos do sistema, permitindo que o desenvolvedor use seu tempo para somente o

desenvolvimento do mesmo e que utilize soluções já testadas e funcionais para as rotinas internas. Seus diferenciais ao ser comparado a outros *frameworks* são: é um *framework* leve, a curva de aprendizado da sua arquitetura é baixa, a criação de testes unitários é feito de forma rápida e a inversão de controle permite que ocorram alterações em partes do sistema e que o impacto em outros setores seja mínimo, permitindo o desenvolvimento de componentes desacoplados e de fácil manutenção.

A Figura 3 apresenta todos os módulos que compõem o *framework* Spring.

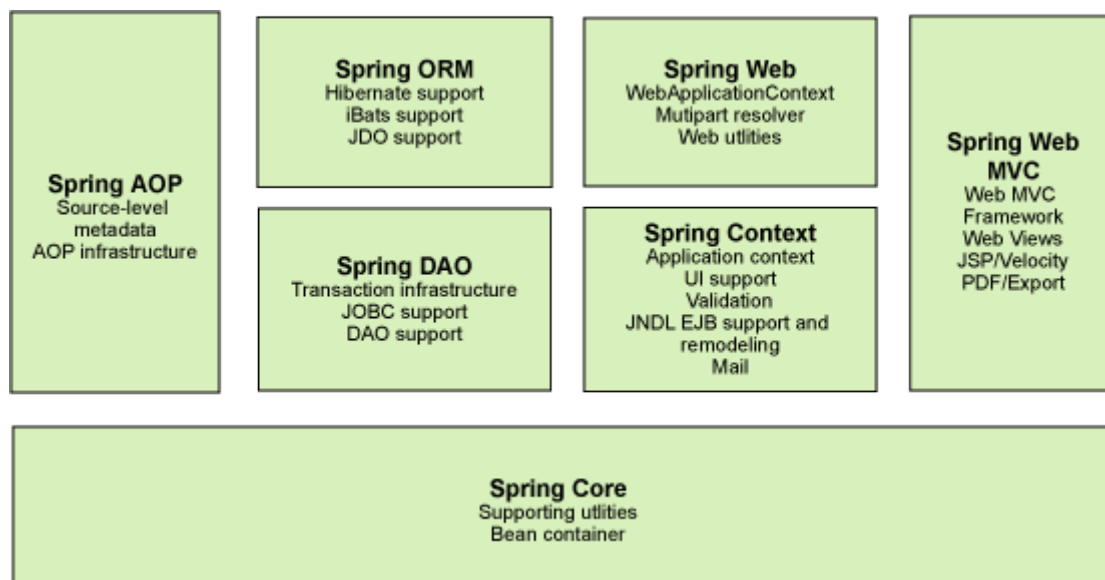


FIGURA 3: Estrutura do *framework* Spring

Fonte: SPRING FRAMEWORK (2008)

A base de todos os módulos disponíveis do *framework* Spring é o seu núcleo (na Figura 3, *Spring Core*). É o núcleo que define como os objetos são criados, configurados e gerenciados, portanto mesmo que somente um ou outro módulo seja utilizado pelo desenvolvedor, o núcleo estará de alguma forma presente juntamente com os módulos, já que estes não funcionariam sem o seu núcleo (WALLS e BREIDENBACH 2008).

O Hibernate foi desenvolvido com o intuito de facilitar a integração da linguagem Java com bancos de dados. Esse *framework* diminui a complexidade dessa integração, principalmente no desenvolvimento de consultas e atualizações no banco de dados, através da transformação das classes Java para tabelas de dados (e da transformação dos objetos Java para seus respectivos da SQL) (HIBERNATE, 2011).

Pamplona (2010) afirma que *webservice* é uma tecnologia que visa à comunicação entre sistemas. A comunicação entre os serviços é padronizada possibilitando a independência de plataforma e de linguagem de programação, tal independência é conseguida devido ao fato que todos *webservices* comunicam-se através do formato XML.

O módulo Spring Webservice foi utilizado para a construção do *webservice* do servidor e, conforme Sattari e Kunjumohamed (2012) a estrutura de comunicação interna de um *webservice* Spring é demonstrada na Figura 4.

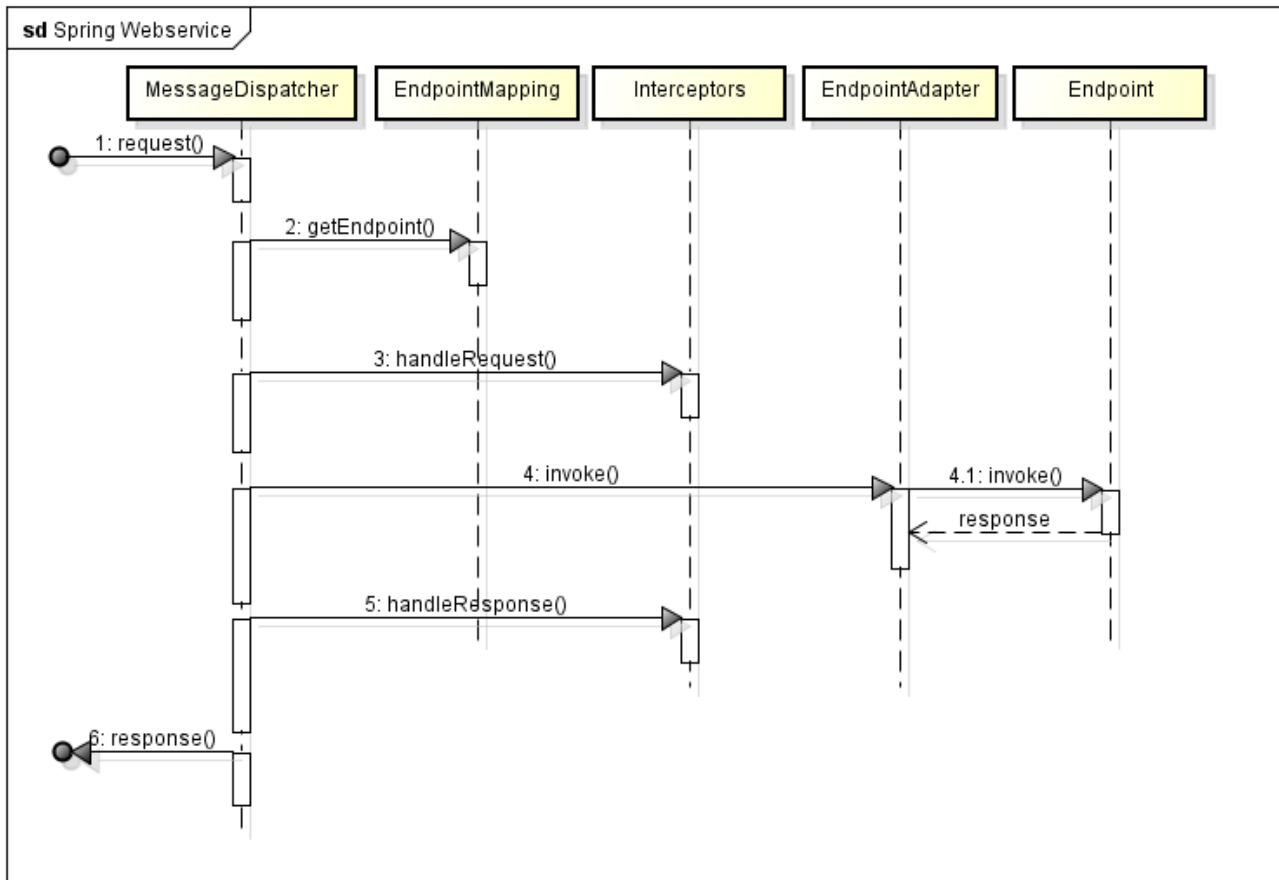


FIGURA 4: Estrutura de um *webservice* Spring

Fonte: Adaptado de Sattari e Kunjumohamed (2012)

5. Metodologia

Para o desenvolvimento do sistema o método de análise e desenvolvimento selecionado foi o modelo espiral, que segundo Boehm (1988), é uma forma de permitir que equívocos ocasionados em algum momento do processo de desenvolvimento sejam corrigidos sem impactar todo o projeto, já que nesse modelo iterações de cada processo do desenvolvimento são repetidas mais de uma vez. O projeto foi dividido em quatro partes recursivas: análise, projeto, codificação e testes.

A ferramenta utilizada para a criação de todos os diagramas da parte de análise e projeto foi selecionado o Astah Community, ferramenta que pertence a empresa Change View e possui uma versão grátis para uso geral. Todos os diagramas seguem os padrões do UML (*Unique Modeling Language*) e apresentam a estrutura do sistema.

Para testes iniciais e de simulação de acesso de outros sistemas que não fossem o RF3D ao servidor a ferramenta SOAP UI da empresa SmartBear foi utilizada, a ferramenta permite que seja

possível analisar todo o processo de comunicação entre o servidor e os clientes que irão acessar suas funcionalidades, permitindo a visualização de toda a estrutura das trocas de mensagens entre os sistemas. Após o sistema estar totalmente funcional para qualquer outro sistema que desejasse utilizar o servidor a total integração com o sistema RF3D foi desenvolvida.

5.1 Análise

O processo de análise para a modelagem do servidor teve como foco a identificação dos requisitos que iriam permitir a integração entre o servidor e o sistema RF3D. A partir das necessidades que o sistema RF3D possuía a estrutura do servidor foi desenvolvida. Apesar disso, outros requisitos foram mapeados com o objetivo de tornar o servidor acessível a outros tipos de sistemas existentes.

As necessidades identificadas envolviam a capacidade de enviar um modelo em XML de um ambiente físico com dados sobre posições de etiquetas e antenas RFID e, a partir deste modelo, simular se as etiquetas seriam lidas ou não considerando interferências do ambiente, retornando os dados em uma estrutura de XML já existente. Para a parte de simulação da leitura correta ou não das etiquetas no ambiente, será utilizado o modelo matemático descrito em Azambuja (2011), apresentado na seção 4. Também foi especificado que o servidor deve ser acessível por outros sistemas que implementem a estrutura de mensagens especificadas e, finalmente, ser acessível por qualquer navegador *web*.

Para permitir que vários sistemas com estruturas de requisições diferentes pudessem usufruir do servidor de simulação, foi então decidido que três pontos distintos de comunicação seriam criados, (i) um especificadamente modelado para gerenciar as requisições do sistema RF3D (*webservice* com requisição síncrona¹) que já existia e por isso não poderia sofrer grandes alterações na sua estrutura de comunicação, (ii) uma estrutura de *webservice* com requisição assíncrona², voltada mais para sistemas que não necessitam aguardar um resultado imediato para as simulações requisitadas e (iii) um *servlet*³ para sistemas que não possuam uma estrutura de comunicação SOAP e usuários que desejam utilizar os dados de simulações e não possuam um sistema cliente. Com isso, o diagrama de caso de uso apresentado na Figura 5 foi elaborado.

¹ Requisição síncrona: estrutura de comunicação em que uma requisição ao *webservice* faz com que este processe e retorne os dados processados referente à requisição.

² Requisição assíncrona: estrutura de comunicação em que uma requisição ao *webservice* faz com que este retorne uma referência (chave de consulta) para que o usuário em outro momento fazendo uma nova requisição utilizando esta referência possua os dados processados referente a primeira requisição.

³ *Servlet*: estrutura mais simplificada de comunicação com o servidor, não envolvendo o protocolo de comunicação SOAP.

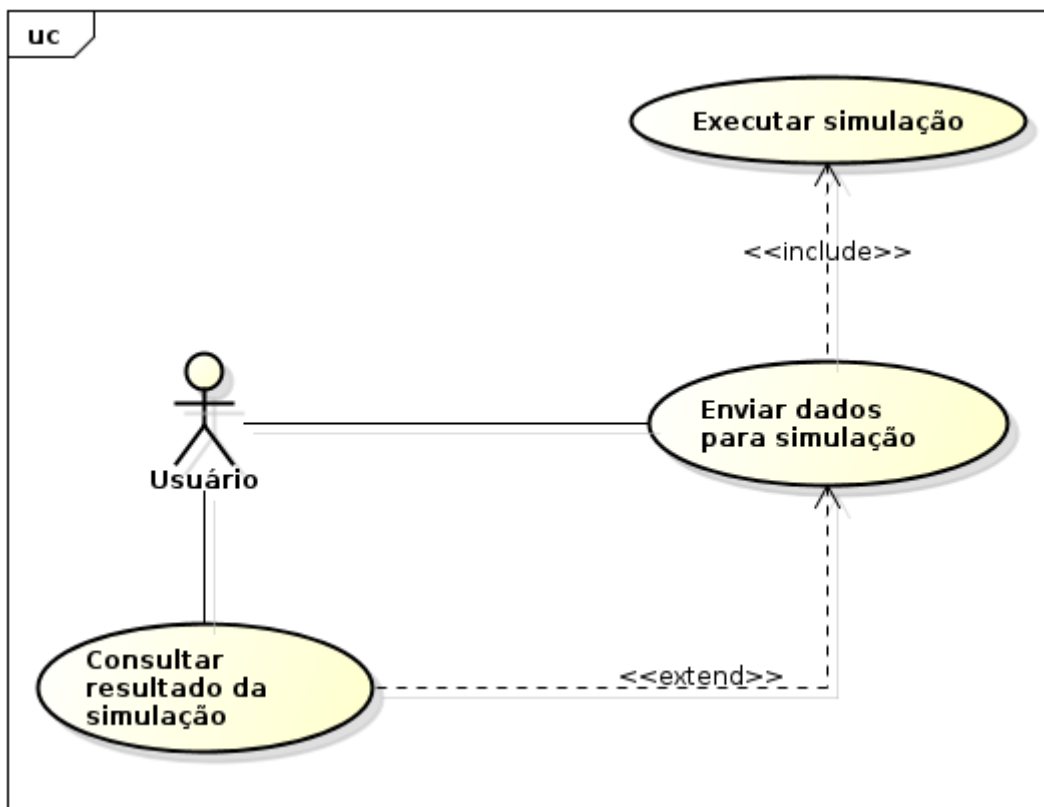


FIGURA 5: Diagrama de caso de uso

Baseando-se na estrutura descrita em Sápiras (2011) e em reuniões com o desenvolvedor do sistema RF3D, o servidor define a mesma estrutura de mensagens que irão ser trocadas entre servidor e sistema, permitindo a total integração entre os dois. Para outros sistemas que não o RF3D, uma estrutura com mais dados sobre a camada de aplicação foi desenvolvida.

5.2 Projeto

O usuário, quando utilizando outro sistema que não o RF3D, possui três formas de comunicar-se com o servidor, sendo: (i) a mesma que o RF3D utiliza, onde o envio de dados para a simulação, a execução da simulação e o retorno dos resultados são feitos na mesma requisição, utilizando o *webservice* síncrono; (ii) Outra forma de comunicação envolve num primeiro momento enviar uma requisição de simulação com os dados a serem simulados e, com a chave de consulta retornada, num momento posterior fazer uma nova requisição para obter o resultado da simulação a partir da chave de consulta que foi retornada como resposta à primeira requisição, utilizando as duas requisições do *webservice* assíncrono; (iii) a última forma de comunicação envolve o acesso via *servlet*, com isso o servidor está apto a permitir que qualquer navegador *web* tenha como enviar requisições e receber resultados de simulações, mesmo que um sistema cliente específico não seja desenvolvido. Para a segunda e terceira opções os resultados da simulação possuem dados sobre a

camada de aplicação, já o processo que atende o sistema RF3D não possui estes dados nos seus resultados, somente o tempo total para ler todas as etiquetas que foram ativadas pela antena da leitora.

A partir do diagrama de caso de uso a estrutura do servidor foi elaborada, iniciando pela modelagem do sistema. As primeiras versões dos diagramas de classes e dos diagramas de sequência foram desenvolvidas, representando uma forma conceitual do sistema. Enquanto o sistema era desenvolvido tanto os diagramas de classes quanto os de sequência eram aprimorados para manterem a integridade da informação.

5.3 Desenvolvimento

O NetBeans foi utilizado como plataforma para o desenvolvimento, por ser a opção mais viável que possui suporte a linguagem Java. O PostgreSQL como o banco de dados e o servidor de aplicação Apache Tomcat para executar o sistema no ambiente *web*. A Figura 6 apresenta a estrutura de comunicação entre o servidor e sistemas clientes.

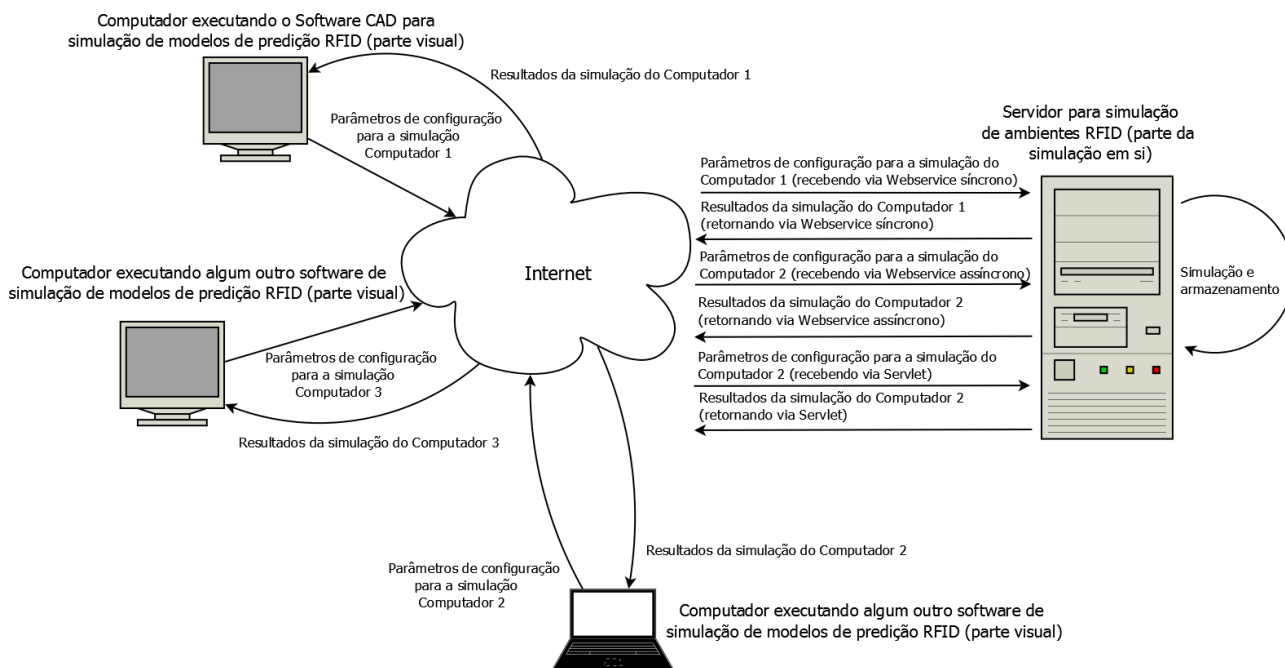


FIGURA 6: Estrutura de comunicação entre servidor e clientes

A arquitetura apresentada na Figura 6 pode ser entendida da seguinte forma: o servidor aguarda por uma requisição de algum sistema cliente, a partir dessa requisição os dados para iniciar a simulação são extraídos e validados. Caso os dados sejam corretamente validados, estes são salvos no banco de dados para base de informações para futuros trabalhos e uma chave de consulta é gerada e retornada para o usuário se o *webservice* assíncrono foi o utilizado para requisitar uma simulação. O sistema, através de rotinas de pesquisa interna, seleciona uma das requisições

aguardando simulações e a executa. Os dados da simulação são então armazenados e, se o *webservice* síncrono ou o *servlet* foram utilizados para a requisição da simulação, estes dados são retornados ao usuário, caso o *webservice* assíncrono foi o utilizado, os dados só serão retornados ao usuário quando este requisitar tais informações através da chave de consulta informada.

Para que um sistema cliente comunique-se com o servidor de simulação é necessário que a estrutura de mensagens XML de requisição e resposta sejam implementadas. A estrutura de requisição ao servidor é apresentada na Tabela 1.

TABELA 1 – *Tags* que compõem um XML de requisição de simulação

Tag	Descrição	Obrigatório
frequência	Frequência em que os aparelhos RFID comunicam-se	Sim
protocolo	Protocolo anti-colisão que é utilizado	Sim
polarização	Polaridade em que o sinal RF é enviado pela leitora	Sim
perdaCaboConector	Perda de sinal que ocorre entre os componentes de leitura da etiqueta	Sim
escala	Escala matemática do ambiente simulado	Não
componentes	<i>Tag</i> que indica a lista de componentes no ambiente	Sim
componente	<i>Tag</i> que indica os dados do componente em questão	Sim
id	Número de identificação do componente do ambiente	Sim
tipo	Tipo do componente (etiqueta, antena, ou tipo de material)	Sim
posição	<i>Tag</i> que indica os dados da posição do componente no ambiente	Sim
x	Posição do componente no eixo x	Sim
y	Posição do componente no eixo y	Sim
z	Posição do componente no eixo z	Sim
anguloHorizontal	Ângulo horizontal do componente no ambiente	Não
anguloVertical	Ângulo vertical do componente no ambiente	Não
ganho	Ganho de sinal do componente	Sim
materialFixado	Quando o componente é uma etiqueta deve-se informar o material fixado nesta	Sim
influenciaMaterial	Quando o componente é uma etiqueta caso o usuário possua dados sobre a influência que este material causa ao sinal RF este dado é informado neste campo	Não
backscatteredLossDbm	Quando o componente é uma etiqueta deve-se informar a quantidade de sinal que esta utiliza para ativar-se e enviar o sinal para a leitora	Sim
potencia	Quando o componente é uma antena deve-se informar a potência do sinal que esta transmite	Sim
potenciaMinima	Quando o componente é uma antena deve-se informar qual a potência mínima que esta antena deve receber para que uma leitura correta seja feita	Sim

Com os dados informados na requisição ao servidor, as simulações da camada física e da camada lógica são então feitas, mas para cada módulo de requisição a estrutura de envio de dados e a obtenção dos resultados simulados diferem em alguns pontos. A Figura 7 apresenta o diagrama de sequência com a primeira parte da requisição assíncrona ao servidor que constitui na solicitação de simulação enviando os dados a serem simulados recebendo uma chave de consulta para uso posterior como resposta.

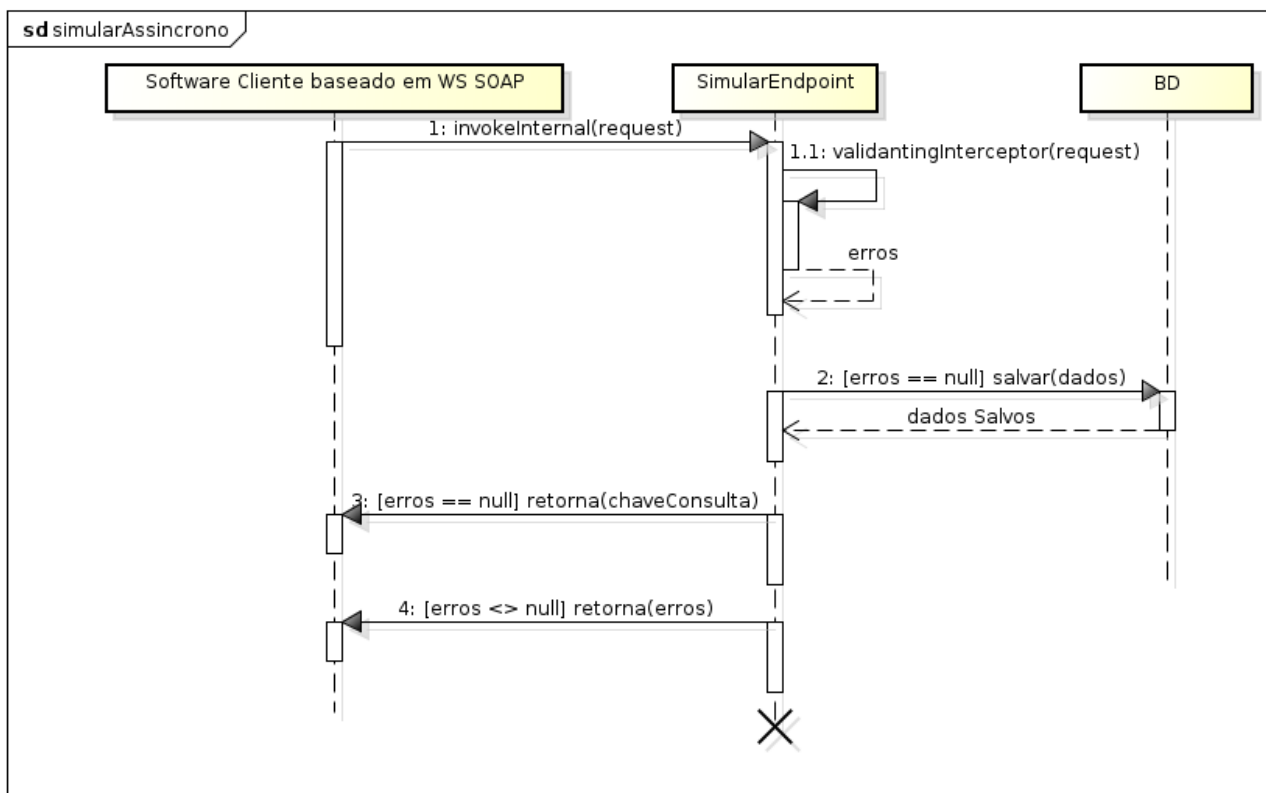


Figura 7: Requisição de simulação assíncrona

Após a requisição ser salva no banco de dados com todos os seus dados a serem simulados e uma chave de consulta ser gerada e retornada como resposta à requisição, esta requisição aguarda até que o servidor selecione os seus dados para serem simulados. A Figura 8 mostra um diagrama de sequência apresentando como o servidor executa suas rotinas de simulações a partir de requisições assíncronas feitas ao servidor.

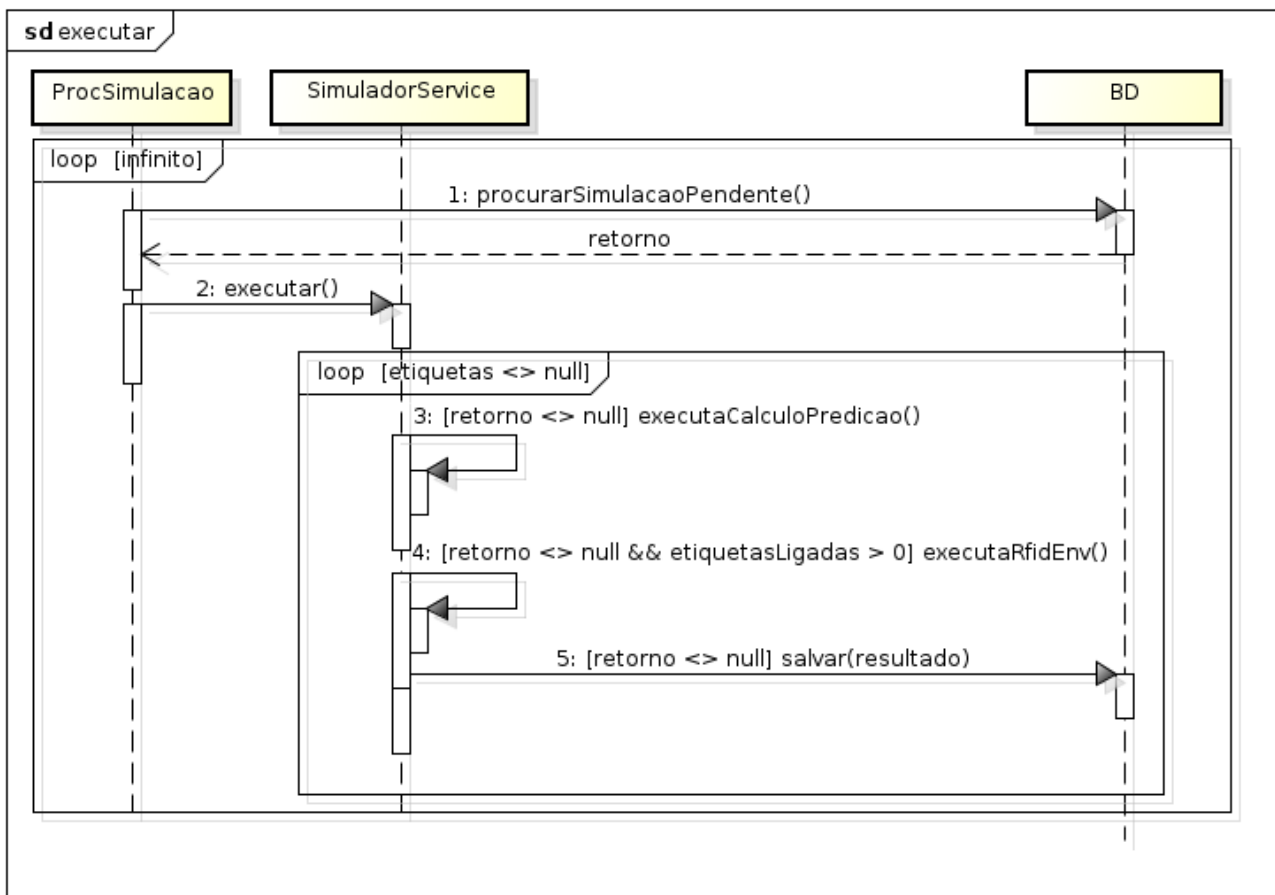


FIGURA 8: Diagrama de Sequência da estrutura das simulações no modo assíncrono

Como a Figura 8 demonstra, o servidor procura uma simulação pendente no banco de dados para executar seus cálculos, pegando os dados a serem utilizados no banco de dados da aplicação. Ao encontrar uma simulação pendente o processo cálculos de predição inicia-se, primeiramente os cálculos da camada física são executados, fazendo com que sejam descobertas quais das etiquetas informadas conseguiram ser ativadas e retornar dados para a leitora utilizando o cálculo de predição de Azambuja (2011). Com a descoberta de quais etiquetas conseguiram ser ativadas e reconhecidas pela leitora, são então feitos os testes da camada lógica (ou “testes de protocolo anti-colisão”), as etiquetas ativas são enviadas para o RFID-Env e este testa as rotinas de comunicação, gerando dados médios para a quantidade de tempo que seria necessário para que todas as etiquetas fossem lidas corretamente, como também a quantidade de campos de leitura⁴ (com colisão de sinal, vazios e o total geral).

Após a conclusão da simulação, seus resultados são salvos no banco de dado e disponibilizados para consulta através da chave de consulta retornada na primeira requisição ao servidor. A Figura 9 mostra o diagrama de sequência para uma requisição de resultados ao servidor.

⁴ Campos de leitura: *slots* de tempo que cada protocolo divide o canal de comunicação da leitora com as etiquetas, essa técnica é utilizada para que não aconteça que todas as etiquetas ativas no ambiente enviem seu sinal ao mesmo tempo para a leitora. (FLOERKEMEIER, 2005)

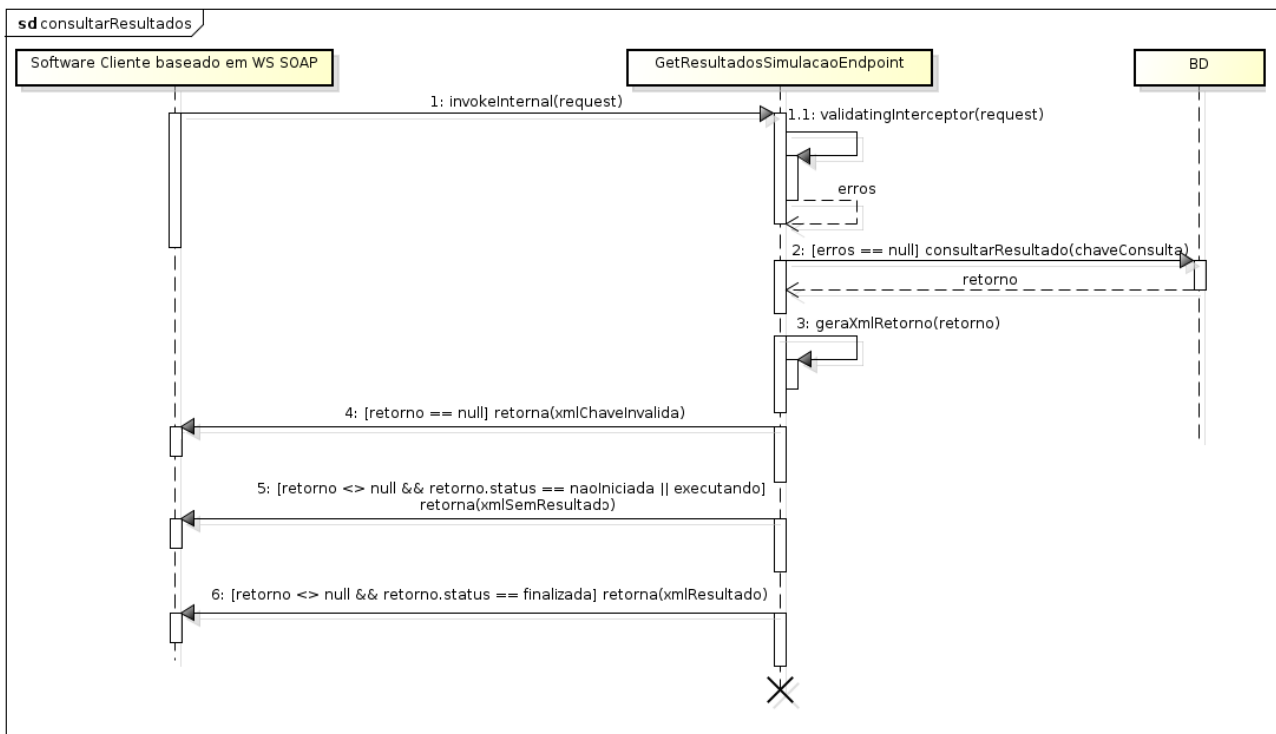


FIGURA 9: Requisição de resultados da simulação assíncrona

A estrutura para requisição e simulação do módulo síncrono e do *servlet* diferencia-se do módulo assíncrono somente na parte de requisição e retorno de resultados, e toda a estrutura de simulação é mantida sem alterações. Quando uma requisição síncrona é recebida pelo servidor esta é priorizada e todo o processo de novas simulações assíncronas é suspenso até que a simulação síncrona recebida seja completamente executada. Isso ocorre porque a requisição síncrona necessita possuir um retorno de simulações na mesma requisição de simulação ao servidor, caso isso não ocorra, a requisição não será válida, pois não terá retornado nenhum resultado dos dados enviados para a simulação do servidor. Enquanto o módulo assíncrono divide-se em três partes (requisição de simulação, execução da simulação e requisição de resultados da simulação) o módulo síncrono e o *servlet* executam estas três etapas no mesmo fluxo de comunicação com o servidor, permitindo que os resultados da simulação sejam obtidos imediatamente após o envio dos dados a serem simulados para o servidor. Toda estrutura destes dois últimos módulos é apresentada no diagrama de sequência da Figura 10.

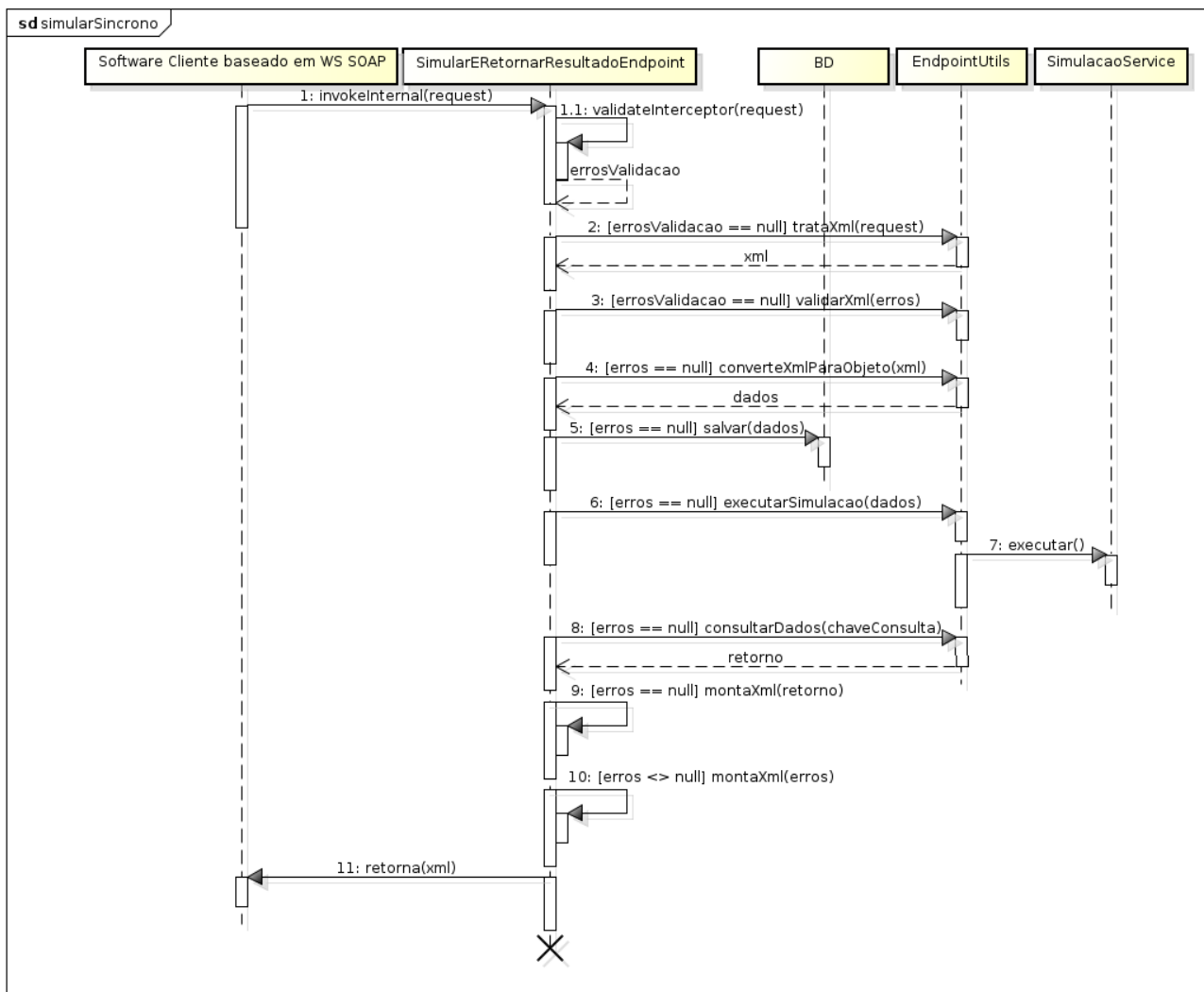


FIGURA 10: Diagrama de Sequência da estrutura das simulações no modo síncrono e *servlet*

Existem três tipos de retorno para as simulações. A Tabela 2 apresenta a estrutura básica compartilhada pelos três tipos e, quando houver algum erro ou a consulta de resultados for inválida ou precipitada, esta também será a estrutura retornada para informar ao usuário. A Tabela 3 apresenta os dados que, junto com os da Tabela 2, estruturam o retorno do *webservice* síncrono, enquanto que a Tabela 4 apresenta os dados adicionais que, junto com os das Tabela 2 e Tabela 3, são retornados pelo *webservice* assíncrono e pelo *servlet*.

TABELA 2 – Estrutura básica do XML de retorno

Tag	Descrição
Versão	Versão do servidor de simulação
Informações	Dados sobre o resultado (Se ocorreu algum erro este será informado aqui)
Autor	Nome do servidor de simulações
dataHora	Data e hora da requisição
simulacaoRealizada	Informa se a simulação ocorreu ou não

TABELA 3 – Dados do XML do *webservice* síncrono

Tag	Descrição
componentes	Tag que indica a lista de componentes no ambiente
componente	Tag que indica os dados do componente em questão
id	Número de identificação do componente do ambiente
tipo	Tipo do componente
lida	Indica se a etiqueta foi lida ou não
potenciaRecebida	Potência recebida da etiqueta pela leitora
distanciaPontos	Distância em linha reta entre a etiqueta e a antena da leitora mais próxima
distanciaEscala	Quando informado uma escala, retorna a informação acima considerando a escala

TABELA 4 – Dados do XML do *webservice* assíncrono e do *servlet*

Tag	Descrição
protocolo	Tag que indica os dados da simulação do protocolo anti-colisão selecionado
nome	Nome do protocolo
tempoLeitura	Tempo que foi necessário para ler todas as etiquetas ativas, considerando velocidade de leitura de dados da leitora
tempoTotal	Tempo que foi necessário para ler todas as etiquetas ativas, considerando o tempo que cada campo de leitura possui em cada protocolo
totalRounds	Quantidade de grupos de campos de leitura que foram utilizados
totalCampos	Quantidade total de campos de leitura utilizados na comunicação
totalCamposVazios	Quantidade de campos de leitura sem envio de dados
totalCamposColisoes	Quantidade de campos de leitura que mais de uma etiqueta enviou seus dados

Os dados retornados fazem com que o sistema cliente tenha como informações quais de suas etiquetas não conseguiram ser lidas em seu ambiente e quanto tempo em média será necessário para que as etiquetas que foram ativadas sejam lidas corretamente pela leitora. Com esses dados em mãos o usuário pode então modelar o seu ambiente da forma que mais for conveniente, sem que nenhum aparelho ou objeto real sejam movidos.

A Equação 1 é utilizada nesse trabalho para os cálculos de predição da camada física. Esse modelo foi proposto por Azambuja (2011), e leva em consideração informações tais como perda de sinal dos cabos conectores, influência do material fixado à etiqueta, potência emitida pela leitora e outros valores técnicos.

$$P_{3r} = P_{3t} + 2G_1 + 2G_2 + 40 \log_{10} \left(\frac{\lambda}{4\pi r} \right) + B_{LdB} - C_L + K_{dB} \quad (1)$$

Na Equação 1 P_{3r} é a potência final do sinal que a leitora recebe de volta da etiqueta após passar pelos cabos conectores à antena. P_{3t} é a potência inicial que a leitora envia antes de passar pelos cabos conectores à antena, G_1 é o ganho de potência de sinal da antena da leitora (multiplica-se por dois, pois é considerado o ganho de sinal de envio e o de retorno), G_2 é o ganho de potência de sinal da etiqueta (também multiplicado por dois, pois é considerado o ganho de sinal de envio e o de retorno), $40\log_{10}(\lambda/4\pi r)$ é o cálculo de perda de potência de sinal a partir do raio (r) informado, B_{Ldb} é a potência do sinal recebido que a etiqueta utiliza para se ativar e enviar seu sinal, C_L é a perda de potência de sinal que ocorre no cabeamento entre antena e leitora e K_{dB} é a influência que o material fixado à etiqueta ocasiona a potência do sinal recebido por ela.

Com essa estrutura o servidor pode então simular a camada física e a camada lógica de um ambiente RFID e auxiliar na estruturação do ambiente sem envolver testes reais para os usuários que desejam implantar a identificação por rádio-frequência.

5.4 Testes

Os testes da aplicação foram feitos com o auxílio da ferramenta SoapUI que foi utilizada para simular um sistema cliente acessando a aplicação, tanto pelo *webservice* síncrono como também o assíncrono, na última iteração de testes o próprio sistema RF3D foi utilizado para testar a comunicação entre o cliente e o *webservice* síncrono, validando assim a integração entre os dois sistemas. A Figura 11 apresenta a tela do *software* SoapUI comunicando-se com o servidor, à esquerda da imagem a mensagem de requisição é mostrada e à direita a mensagem de retorno é apresentada.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:obterResultadoResponse xmlns:ns2="urn:xmethods-delayed-confirmation">
      <rf3d>
        <frequencia>915.25</frequencia>
        <protocolo>BTREE</protocolo>
        <polarizacao>HORIZONTAL</polarizacao>
        <perdaCaboConector>1</perdaCaboConector>
        <escala>0.01</escala>
        <componentes>
          <componente>
            <id>0</id>
            <tipo>antena</tipo>
            <posicao>
              <x>-115.63812326133394</x>
              <y>1</y>
              <z>-150.9083779481639</z>
            </posicao>
            <anguloHorizontal>0</anguloHorizontal>
            <anguloVertical>0</anguloVertical>
            <ganho>2</ganho>
            <potencia>1</potencia>
            <potenciaMinima>-80</potenciaMinima>
          </componente>
          <componente>
            <id>1</id>
            <tipo>tag</tipo>
            <posicao>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
      <ns2:obterResultadoResponse xmlns:ns2="urn:xmethods-delayed-confirmation">
        <ns2:xmlResposta><![CDATA[
          <rf3d>
            <versao>1</versao>
            <informacoes>Nenhuma etiqueta conseguiu transmitir seus dados</informacoes>
            <autor>RFID Web Service Simulator</autor>
            <dataHora>17/08/2012 09:02</dataHora>
            <simulacaoRealizada>true</simulacaoRealizada>
            <componentes>
              <componente>
                <id>1</id>
                <tipo>tag</tipo>
                <lida>>false</lida>
                <potenciaRecebida>-184.70</potenciaRecebida>
                <distanciaPontos>310.98</distanciaPontos>
                <distanciaEscala>31098</distanciaEscala>
              </componente>
            </componentes>
          </rf3d>
        ]]></ns2:xmlResposta>
      </ns2:obterResultadoResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

```

FIGURA 11: Tela do SoapUI comunicando-se com o *webservice* assíncrono

Os testes de comunicação com o *servlet* foram feitos nos navegadores *web* Google Chrome, Internet Explorer e Mozilla Firefox. A Figura 12 apresenta a tela de recepção que o usuário utiliza para enviar seus dados para a simulação do servidor, o retorno dessa requisição é um arquivo XML que então pode ser salvo e utilizado da forma que mais interessar o usuário. O campo de texto é aonde o usuário colocará os dados que este deseja simular, respeitando a estrutura do arquivo XSD responsável por validar a requisição de simulação.



FIGURA 12: Tela inicial do servidor quando acessado para uma requisição por *servlet*

6. Resultados

Os resultados apresentados pelo servidor de simulação foram analisados utilizando dados de testes reais, obtidos através das pesquisas conduzidas e apresentadas por Azambuja (2011). Os dados dos testes em ambientes físicos apresentados nesta seção pertencem ao referido trabalho, e como o servidor utiliza o mesmo modelo matemático proposto neste trabalho, os dados utilizados para validar o modelo consequentemente validam os resultados do servidor.

Azambuja (2011) utilizou quatro tipos de ambientes para validar seu modelo matemático. O primeiro ambiente consiste de uma simulação em local aberto, sem interferência de objetos entre a leitora e a etiqueta, e este ambiente tem o objetivo de servir de parâmetro de referência, pois sem nenhum obstáculo a comunicação sofre poucas interferências. Nesse ambiente o modelo matemático obteve uma precisão de 95,3% de acerto em relação aos testes no ambiente real. A Figura 13 mostra a estrutura e a configuração deste ambiente.



FIGURA 13: Ambiente externo em “espaço livre”

Fonte: Azambuja (2011)

O segundo e terceiro ambientes foram criados com estruturas parecidas a fim de se verificar se ambientes semelhantes gerariam resultados equivalentes quanto a leitura ou não das etiquetas. Esses ambientes são duas salas com vários obstáculos próximos a leitora e a etiqueta, sendo que essa configuração foi intencional, para verificar como o modelo matemático responderia quando testado com variáveis desconhecidas dentro do ambiente, pois em um ambiente desse tipo certamente muita interferência é gerada nos sinais originais de RF. Nesses ambientes o modelo matemático obteve uma precisão de 90,7% na sala A e 87% na sala B de acerto em relação aos testes no ambiente real. A Figura 14 mostra a estrutura de um destes ambientes.

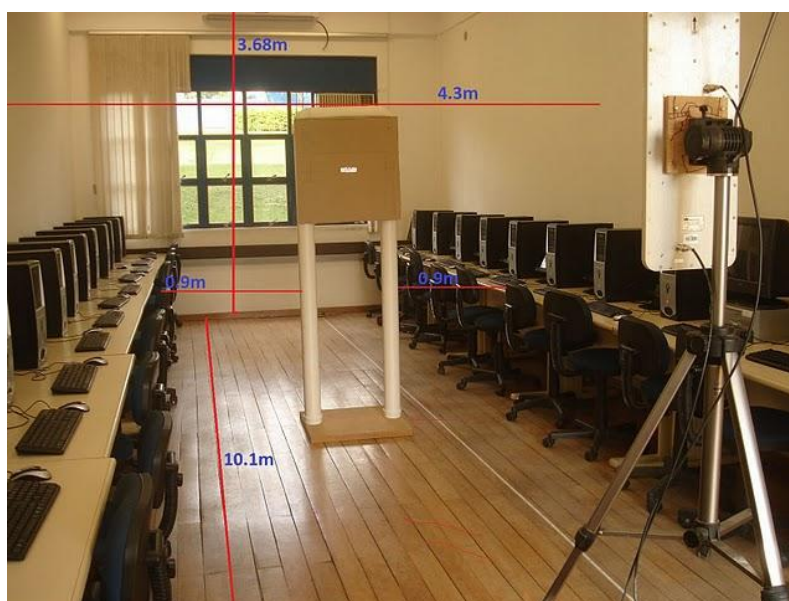


FIGURA 14: Ambiente interno sala A

Fonte: Azambuja (2011)

O último ambiente consistia de uma sala com obstáculos distantes da leitora e da etiqueta. Nesse ambiente o modelo matemático obteve uma precisão de 88,8% de acerto em relação ao ambiente real. A Figura 15 mostra a estrutura deste ambiente.

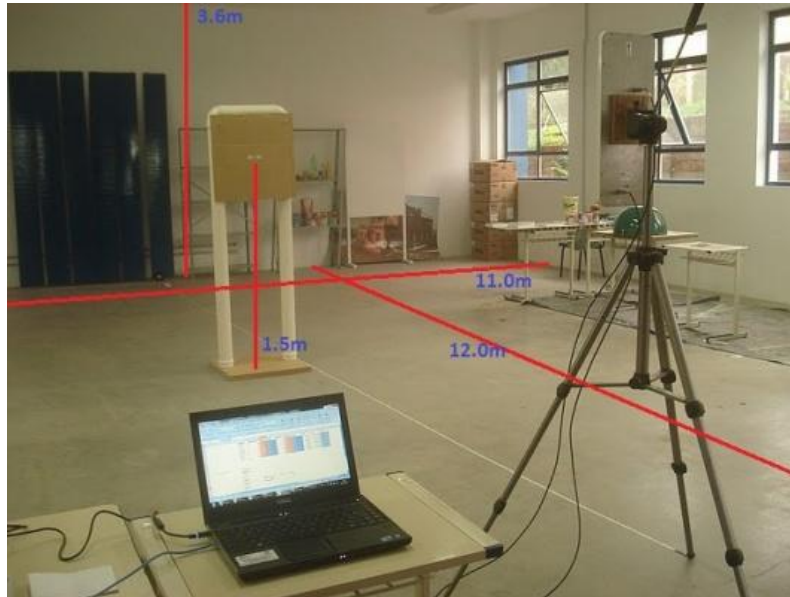


FIGURA 15: Ambiente interno sala C

Fonte: Azambuja (2011)

A partir dos dados dos ambientes apresentados em Azambuja (2011), alguns testes foram executados no servidor para verificar a consistência do mesmo. As configurações utilizadas para coletar os dados são apresentadas na Tabela 5.

TABELA 5 – Configurações para a coleta de resultados.

Frequência	915.25 UHF
Etiquetas no ambiente	1
Antenas do aparelho leitor	1
Obstáculos no ambiente	0
Perda de potência nos cabos conectores	3,1 dB
Potência enviada pelo aparelho leitor	36,1 dB
Perda de potência nas etiquetas	6,8%
Ganho de potência da antena do equipamento leitor	7 dB
Ganho de potência da antena da etiqueta	2 dB

A Tabela 6 apresenta os resultados obtidos utilizando as configurações fixas, descritas na Tabela 5, para cada teste executado, além destas configurações fixas, outras configurações variáveis

foram utilizadas tendo sido colhido resultados para cada uma delas. Os valores coletados representam a potência retornada ao equipamento leitor.

TABELA 6 – Resultados em dB (decibéis) de testes feito no servidor de simulação.

Material/Distância(m)	1,0	2,0	3,0	4,0	5,0	7,0	10,0
Isopor	-30,80	-42,84	-49,88	-54,88	-58,75	-64,60	-70,80
Madeira	-40,13	-52,17	-59,21	-64,21	-68,08	-73,93	-80,13
Nenhum	-30,73	-42,77	-49,81	-54,81	-58,68	-64,53	-70,73
Papelão	-33,43	-45,47	-52,51	-57,51	-61,38	-67,23	-73,43
Plástico	-32,73	-44,77	-51,81	-56,81	-60,68	-66,53	-72,73
Vidro	-37,34	-49,38	-56,42	-61,42	-65,29	-71,14	-77,34

A partir dos dados obtidos, a interferência dos objetos em que as etiquetas estão fixadas é uma variável importante em simulações RFID, visto que o material absorve o sinal emitido pelo equipamento leitor diminuindo a potência do sinal recebido pela etiqueta, o que faz com que a potência refletida novamente para o aparelho leitor, dependendo da distância entre os equipamentos, não seja o suficiente para uma leitura correta dos dados. Azambuja (2011) afirma que uma potência de até -80 dB refletida da etiqueta ao equipamento leitor permite que a leitura dos dados contidos nesta etiqueta ocorra sem problemas, com esse valor em mente e com os valores obtidos através das simulações apresentadas na Tabela 6, é possível perceber que etiquetas fixadas em madeira não são mais lidas a partir de 10 metros. Com isso, pode-se notar a importância em simulações de comunicação entre etiquetas e leitora para que problemas como a distância máxima de leitura de certo objeto e a potência que o equipamento leitor deve transmitir no ambiente sejam conhecidos sem que tempo e esforços humanos e financeiros sejam gastos sem necessidade.

7. Conclusão

O trabalho de pesquisa e desenvolvimento teve como objetivo desenvolver um servidor *web* de simulação de ambientes que utilizam a tecnologia RFID, além de armazenar dados de tais simulações para futuras pesquisas. O servidor permite a integração por aplicações que implementem a comunicação via *webservice* e é acessível inclusive por navegadores *web* através da tecnologia *servlet*.

Com o modelo de predição utilizado por este servidor, uma taxa de acerto nas simulações superior a 86% é alcançada em diversos tipos de ambientes. Em ambientes em espaço livre a taxa de acerto alcança 95,3%, em ambientes fechados com objetos próximos as antenas dos aparelhos a taxa alcançou 90,7% e 87% enquanto que sem obstáculos próximos a taxa de acerto foi de 88%.

Esses números indicam que o modelo apesar de alcançar uma grande taxa de acerto, ainda possui variáveis desconhecidas relacionadas a objetos que não possuem relação com o sistema RFID, como as paredes da sala por exemplo.

Com a integração do modelo matemático que simula a comunicação na parte física e o RFID-Env que simula a comunicação na parte lógica, ainda foi possível alcançar mais informações relativas a média de tempo que as etiquetas do ambiente serão lidas, expandindo em mais uma parte as simulações de ambientes RFID.

Mesmo sendo facilmente integrado com outras aplicações, o servidor foi desenvolvido e modelado com o objetivo de complementar o sistema RF3D já existente. Com isso, um sistema maior que permite em uma extremidade modelar o ambiente RFID conforme a necessidade do usuário e na outra extremidade simular este ambiente e retornar os resultados a primeira extremidade foi alcançado.

O servidor é acessível a qualquer sistema que respeite qualquer uma das três estruturas de comunicações disponíveis (*webservice* síncrono, *webservice* assíncrono e *servlet*) e exige somente uma conexão ativa de *internet* para a correta utilização do mesmo. Por já existir um sistema cliente e o servidor ter a necessidade de comunicar-se corretamente com este, algumas dificuldades foram encontradas no desenvolvimento do projeto, como por exemplo, deixar de utilizar facilitadores da tecnologia Spring (como a validação de arquivos XML automaticamente) para utilizar uma forma mais manual para o mesmo propósito, a partir dessa dificuldade a estrutura *webservice* foi dividida entre síncrona e assíncrona.

Apesar dos resultados obtidos serem satisfatórios, este servidor não está totalmente finalizado. Em trabalhos futuros existe espaço para o melhoramento do modelo de predição, fazendo com que objetos em movimento sejam simulados e a interferência de outros objetos que não estão fixados as etiquetas sejam estudadas mais profundamente. Apesar disso, o modelo de predição já é suficientemente robusto para simular com alta taxa de acerto diversos tipos de ambientes.

Com os resultados alcançados, o servidor pode ser utilizado tanto como fonte para estudos acadêmicos sobre ambientes RFID, como também pode ser útil no setor comercial explorando sua capacidade de permitir testar qual a melhor configuração para que o ambiente RFID funcione corretamente.

Referências

AZAMBUJA, Marcelo Cunha de. **Modelos e técnicas para simulação de sistemas UHF de Identificação por Radio Frequência (RFID)**. Tese submetida como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação da Pontífice Universidade Católica do Rio Grande do Sul, 2011.

JUNG, Carlos F. *et al.* **Análise de problemas e soluções propostas para uso da tecnologia RFID em sistemas produtivos.** Artigo aprovado para XI Semana de Engenharia de Produção Sul-Americana, 2011.

SÁPIRAS, Leonardo Augusto. **Software para apoio ao projeto de ambientes RFID aplicado a modelos de predição.** Trabalho de Conclusão apresentado ao Curso de Sistemas de Informação das Faculdades Integradas de Taquara, 2011.

FINKENZELLER, Klaus. **RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification** – 3 ed. Munique: John Wiley & Sons, 2010.

FLOERKEMEIER, Christian; WILLE, Matthias. **Comparison of Transmission Schemes for Framed ALOHA based RFID Protocols.** Institute for Pervasive Computing ETH Zurich, Zurique, 2005.

BHATT, Himanshu e GLOVER, Bill. **RFID Essentials.** O’Reilly, 2006.

PRADO, Neli R. da Silveira Almeida; PEREIRA, Néocles Alves e POLITANO, Paulo Rogério. **Dificuldades para a adoção de RFID nas operações de uma cadeia de suprimentos.** XXVI ENEGEP - Fortaleza, CE, 2006.

BOEHM, Barry W. **A Spiral Model of Software Development and Enhancement.** TRW Defense Systems Group, 1988.

EPC Global. **EPC Tag Standard Version 1.5.** Disponível em <http://www.gs1.org/gsm/kc/epcglobal/tds/tds_1_5-standard-20100818.pdf>. Acesso em: 22 out. 2011.

JAVA. **The History of Java Technology.** Disponível em: <<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>>. Acesso em: 22 out. de 2011.

XML. **Extensible Markup Language.** Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 22 out. de 2011.

HIBERNATE. **Relational Persistence for Java and .NET.** Disponível em: <<http://www.hibernate.org/>>. Acesso em: 22 out. 2011.

PAMPLONA, Vitor Fernando. **Web Services. Construindo, disponibilizando e acessando Web Services via J2SE e J2ME.** Disponível em: <<http://javafree.uol.com.br/artigo/871485/Web-Services-Construindo-disponibilizando-e-acessando-Web-Services-via-J2SE-e-J2ME.html>>. Acesso em: 22 out. de 2011.

BERNARDO, Cláudio Gonçalves. **A tecnologia RFID e os benefícios da etiqueta inteligente para os negócios.** São Paulo, 2004.

JOHNSON, Rod e HOELLER, Juergen. **Expert One-on-One J2EE Design and Development.** Wrox, 2002.

SATTARI, Hamidreza e KUNJUMOHAMED, Shameer. **Spring Web Services 2 Cookbook.** Reino Unido: Packt Publishing Limited, 2012.

WALLS, Craig e BREIDENBACH, Ryan. **Spring in Action.** Greenwich: Manning Publications Co., 2008.

SPRING FRAMEWORK. **Spring Framework Tutorials.** Disponível em: <<http://www.developersbook.com/spring/spring-tutorials/spring-tutorials.php>>. Acesso em: 15 ago. 2012.

SERVLET. **Java Servlet Technology.** Disponível em: <http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html>. Acesso em: 14 ago. 2012.

SOAP. **Soap Introduction.** Disponível em: <http://www.w3schools.com/soap/soap_intro.asp>. Acesso em: 14 ago. 2012.

RADIO ELECTRONICS. **RFID Frequencies and Frequency Bands.** Disponível em: <<http://www.radio-electronics.com/info/wireless/radio-frequency-identification-rfid/low-high-frequency-bands-frequencies.php>>. Acesso em: 27 set. 2012.