

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS DE
INFORMAÇÃO

EDUARDO MENDES SARMENTO

**CONTROLE DE PRESENÇA UTILIZANDO RFID:
UM ESTUDO DE CASO UTILIZANDO A LINGUAGEM RUBY**

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2011

EDUARDO MENDES SARMENTO

CONTROLE DE PRESENÇA UTILIZANDO RFID:
UM ESTUDO DE CASO UTILIZANDO A LINGUAGEM RUBY

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Desenvolvimento de Sistemas de Informação da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador Prof. Msc Fernando Schutz

MEDIANEIRA

2011



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Gerência de Ensino
Curso Superior de Tecnologia em
Desenvolvimento de Sistemas de Informação



TERMO DE APROVAÇÃO

CONTROLE DE PRESENÇA UTILIZANDO RFID:
UM ESTUDO DE CASO UTILIZANDO A LINGUAGEM RUBY

Por

Eduardo Mendes Sarmento

Este Trabalho de Diplomação (TD) foi apresentado às 11:10h do dia 10 de junho de 2011 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Desenvolvimento de Sistemas de Informação, da Universidade Tecnológica Federal do Paraná, Campus Medianeira. O candidato foi argüido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. MSc. Fernando Schutz
UTFPR – *Campus* Medianeira
(Orientador)

Prof^a. MSc Alessandra B. G. Hoffmann
UTFPR – *Campus* Medianeira
(Convidada)

Prof. MSc. Pedro Luiz de Paula Filho
UTFPR – *Campus* Medianeira
(Convidado)

Prof. MSc. Juliano Rodrigo Lamb
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

RESUMO

Existe uma dificuldade muito grande para controlar a presença dos participantes em um evento ou dos alunos em uma sala de aula, com chamadas, listas, carimbos ou pessoas fazendo o controle na entrada. Estes procedimentos muitas vezes podem provocar atrasos ou perda de tempo, tanto no hora de fazer o controle quanto para repassar as informações para o sistema. Buscando uma solução mais simples para tal problema, este projeto teve por objetivo a criação de cartões com etiquetas de rádio frequência, que seriam utilizados por alunos ou participantes que passariam na entrada das salas para registrar suas presenças e estas já seriam repassadas imediatamente para o sistema do evento. Os leitores de RFID passariam as informações para um sistema feito com Ruby, que gravaria as informações em bancos de dados apropriados, que o sistema do evento acessaria para verificar as frequências. Foi criada uma espécie de serviço para ser acessado pelos sistemas. Os resultados indicaram que o processo poderá ser utilizado para agilizar o acesso e o controle de presença de participantes em eventos.

Palavras-chave: Controle de Presença, RFID, Ruby.

ABSTRACT

There is great difficulty to control the presence of participants in an event or of students in a classroom, with calls, lists, stamps or persons making the control input. These procedures can often lead to delays or loss of time, both in time to make the control as to relay the information to the system. Seeking a simpler solution to this problem, this project aimed to create cards with radio frequency tags, which students or participants to pass the entrance halls to register their presence and they would have immediately passed on to the system's event. RFID readers would pass the information to a system made with Ruby, who would record the information in the appropriate data base, the system would access the event to check the frequencies. It created a kind of service to be accessed by the systems. The results indicated that the process can be used to speed access and control of presence of participants at events.

Key- words: control the presence, RFID, Ruby.

LISTA DE FIGURAS

FIGURA 1 - LINHA HISTÓRICA DA RFID.....	5
FIGURA 2 - ARQUITETURA RFID.....	6
FIGURA 3 - APLICAÇÕES DE ETIQUETAS RFID.....	8
FIGURA 4 - LEITORES RFID.....	9
FIGURA 5 - LEITOR RFID COM ANTENA ACOPLADA.....	10
FIGURA 6 - LEITOR RFID DA MARCA TOUCHATAG.....	14
FIGURA 7 - CASOS DE USO DO ADMINISTRADOR.....	17
FIGURA 8 - CASO DE USO DE CADASTRO DE PRESENÇA.....	17
FIGURA 9 - MODELO DE ENTIDADES E RELACIONAMENTOS.....	18
FIGURA 10 – TELA INICIAL DO SISTEMA.....	19
FIGURA 11 - TELA DE CADASTRO DE NOVOS PARTICIPANTES.....	20
FIGURA 12 - POP-UP DE LEITURA DE TAG.....	20
FIGURA 13 - POP-UP DE TELA DE LEITURA DOS CRACHÁS.....	22

LISTA DE QUADROS

QUADRO 1 - FREQUÊNCIAS E RESPECTIVAS APLICAÇÕES.....10

LISTA DE SIGLAS E ABREVIATURAS

DRY	<i>Don't Repeat Yourself</i>
EPC	<i>Eletronic Product Code</i>
IC	Circuito Integrado
IDE	<i>Integrated Development Environment</i>
IFF	<i>Identify Friend or Foe</i>
MIT	<i>Massachusetts Institute of Technology</i>
MVC	<i>Model-view-controller</i>
OOP	<i>Object-oriented programming</i>
RAM	<i>Random Access Memory</i>
RFID	<i>Radio-Frequency IDentification</i>
YAML	<i>YAML Ain't Markup Language</i>
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

1 INTRODUÇÃO	1
1.1 OBJETIVO GERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 JUSTIFICATIVA	2
1.4 DIVISÃO DO TRABALHO	2
2 REFERENCIAL TEÓRICO	4
2.1 IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA	4
2.1.1 Histórico	4
2.1.2 Arquitetura	5
2.2 LINGUAGEM DE PROGRAMAÇÃO RUBY	10
2.2.1 Histórico	11
2.2.2 Ruby on Rails	11
2.2.3 RubyGems	12
3 MATERIAL E MÉTODOS	13
3.1 DESCRIÇÃO DO AMBIENTE EXPERIMENTAL	13
3.1.1 Configuração do hardware	13
3.1.2 Sistema Operacional	14
3.2 APLICATIVOS E BIBLIOTECAS	14
3.2.1 Netbeans	14
3.2.2 NFC	15
4 RESULTADOS E DISCUSSÃO	16
4.1 ANÁLISE GERAL DO SISTEMA	16
4.1.1 Requisitos	16
4.1.2 Casos de Uso	17
4.1.3 Modelo de Entidades e Relacionamentos	18
4.2 TELAS	19
5 CONCLUSÃO	23
REFERÊNCIAS	24

1 INTRODUÇÃO

Cada vez mais as tarefas de uma empresa estão sendo substituídas por novas tecnologias com processos automatizados, fazendo com que a empresa tenha um controle cada vez melhor das informações e dos serviços. Estas mudanças tecnológicas não estão voltadas e disponíveis apenas para empresas: instituições de ensino, sociedades de classe, órgãos governamentais e até mesmo os lares estão utilizando cada vez mais a tecnologia da informação e comunicação.

Joseph Wood Krutch (apud BUCHSBAUM, 2006), afirmou que “a tecnologia torna as grandes populações possíveis, grandes populações tornam a tecnologia indispensável”. Sendo assim percebe-se que os processos tecnológicos devem ser adaptados ao dia a dia.

Em alguns eventos o número de participantes é cada vez maior, um problema que organizadores de evento e professores têm em comum é o controle de presença em seus eventos. Seja em uma palestra, mini-curso ou aula, perde-se tempo com tal controle, e processos manuais de identificação em eventos causam a formação de filas e, na sua maioria, são ineficientes. Uma solução para este problema é fazer o controle da presença dos participantes um processo automático.

Em busca de uma linguagem para desenvolvimento do sistema para controlar as presenças, encontrou-se Ruby, uma linguagem de programação interpretada multiparadigma, de tipagem dinâmica e forte, com gerenciamento de memória automático, originalmente planejada e desenvolvida no Japão em 1995, por Yukihiro "Matz" Matsumoto. Foi inspirada principalmente por Python, Perl, Smalltalk, Eiffel, Ada e Lisp. Atualmente, Ruby é a 12ª linguagem de programação mais popular do mundo (RUBY BRASIL, 2010).

Uma *tag* ou etiqueta RFID é um dispositivo de comunicação eletrônico complementar de automação e cujo objetivo é receber, amplificar e retransmitir um sinal em uma frequência diferente ou transmitir de uma fonte uma mensagem pré-determinada em resposta à outra pré-definida. Um pequeno objeto que pode ser colocado em uma pessoa, animal, equipamento, embalagem ou produto, dentre outros.

Neste contexto esta pesquisa visou o estudo e a utilização da tecnologia de etiquetas RFID, juntamente com a linguagem de programação Ruby, criando um

sistema que possa ler o crachá dos participantes e automaticamente gravar estas informações em banco de dados.

1.1 OBJETIVO GERAL

Criar uma infra estrutura para controlar a presença dos participantes em eventos, utilizando etiquetas RFID adaptadas em seus crachás e paralelamente desenvolver um sistema na linguagem de programação Ruby que fizesse a leitura desses cartões e o armazenamento dessas informações.

1.2 OBJETIVOS ESPECÍFICOS

- Elaborar um referencial teórico sobre RFID e suas aplicações;
- Elaborar maneiras de fazer a comunicação do leitor RFID com o sistema na linguagem de programação Ruby;
- Elaborar uma aplicação utilizando a linguagem Ruby e RFID para simular o controle de presença de participantes em determinado evento;
- Testar etiquetas RFID adaptadas em crachás no sistema desenvolvido.

1.3 JUSTIFICATIVA

Com o crescimento dos eventos e do numero de participantes nos mesmo, o fluxo de pessoas em palestras ou mini-cursos também é bem maior. Este grande fluxo de pessoas inevitavelmente acaba gerando filas, e a necessidade de fazer o controle dos participantes presentes algumas vezes agrava ainda mais o problema das filas. Devido a isso, um estudo de caso de tecnologias que sejam capazes de automatizar este processo, é um bom material para buscar novas saídas para evitar transtornos tanto de participantes quanto de organizadores.

1.4 DIVISÃO DO TRABALHO

O presente trabalho está dividido em seis capítulos arranjados da forma apresentada a seguir.

O primeiro capítulo inicia com a introdução, que tem uma breve descrição dos problemas aos quais se deseja criar uma solução, com uma abordagem das

tecnologias que se pretende utilizar, e seguido disso, os objetivos e a justificativa da realização do trabalho.

Já no segundo capítulo, é possível conferir todo referencial teórico, com detalhes da tecnologia RFID e da linguagem de programação Ruby.

Os materiais, e os métodos utilizados para desenvolvimento do projeto, estão no capítulo três, e os resultados obtidos são apresentados no quarto capítulo.

O capítulo cinco apresenta a conclusão, com alguns detalhes de custo/benefício e sugestão de trabalho futuro.

2 REFERENCIAL TEÓRICO

Este capítulo tem por objetivo demonstrar os resultados de pesquisas feitas sobre a tecnologia RFID e da linguagem de programação Ruby para realizar este projeto.

2.1 IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA

Identificação por rádio frequência, RFID, é um método de identificação automática através de sinais de rádio, recuperando e armazenando dados através de dispositivos denominados etiquetas RFID.

2.1.1 Histórico

As raízes do RFID surgiram em 1897 quando Guglielmo Marconi inventou o rádio e se lançou na década de 30 quando o Exército e Marinha Americana tiveram o desafio de criar um sistema que fosse capaz de diferenciar forças inimigas das aliadas, no ar ou em terra então em 1937 o Laboratório de Pesquisa Naval Norte Americano, desenvolveu o *Identification Friend-or-Foe* (IFF) o qual se tornou a base para o controle de tráfego aéreo e se tornou pioneiro do que é hoje a RFID (GUEDES, 2005).

Progressos na área de comunicação por rádio frequência continuaram através das décadas de 50 e 60. Cientistas e acadêmicos dos Estados Unidos, Europa e Japão realizaram pesquisas e apresentaram estudos explicando como a rádio frequência, poderia ser utilizada para identificar objetos remotamente.

Em 1999, com a ajuda de cientistas do Instituto de Tecnologia de Massachusetts (MIT), um consórcio de empresas formaram a Auto-ID Center, um centro de pesquisa continuada sobre a natureza e a utilização de identificação por frequência de rádio. Desenvolveram um projeto para viabilizar a utilização de etiquetas RFID em todos os produtos feitos pelas empresas, e rastreá-los. As etiquetas teriam micro-chips simples capazes de guardar pouca informação, um código serial, e mais detalhes sobre tais produtos eram armazenados em bancos de

dados externos (SWEENEY, 2005). Na Figura 1 pode-se identificar alguns marcos na história do RFID.

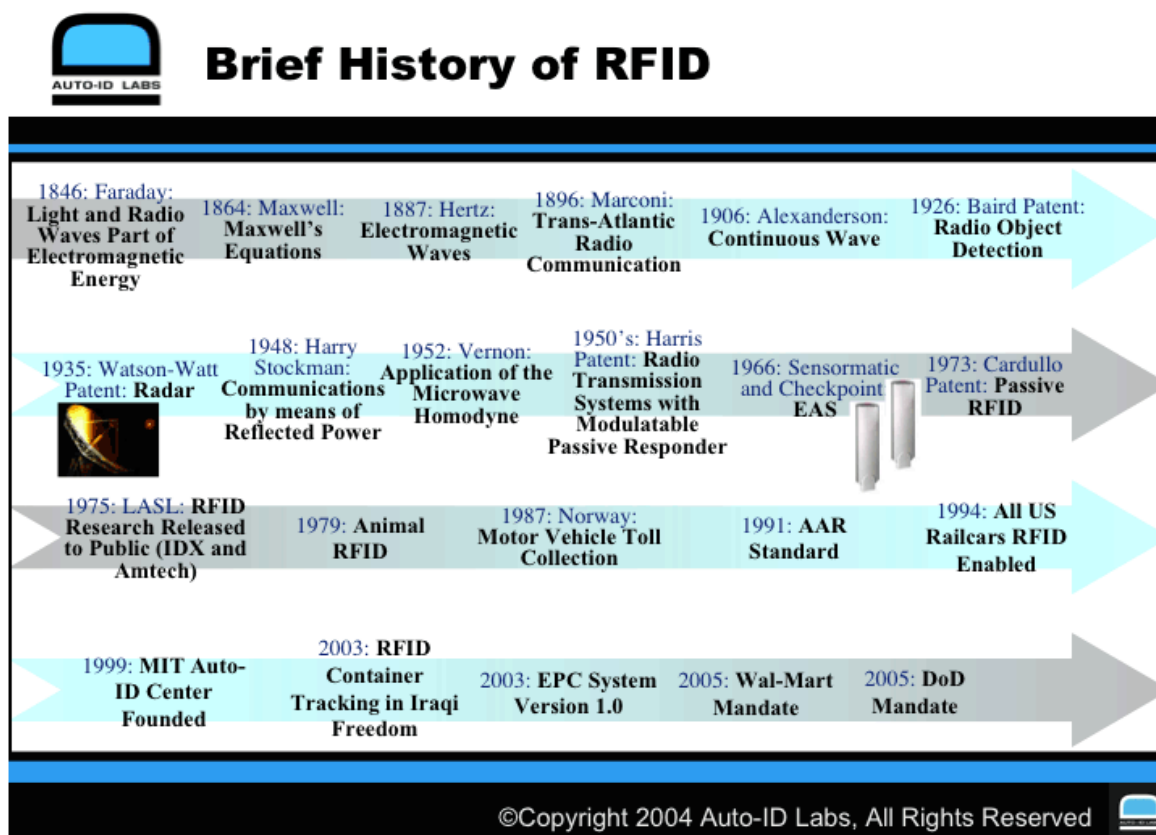


Figura 1 - Linha histórica da RFID
Fonte: Preemptive Media(2010)

2.1.2 Arquitetura

Com a expansão do RFID é necessário que se entenda como funciona esta tecnologia. Um sistema de RFID é formado basicamente de três componentes: as antenas, as etiquetas e o leitor. Na Figura 2 observa-se a arquitetura do RFID.

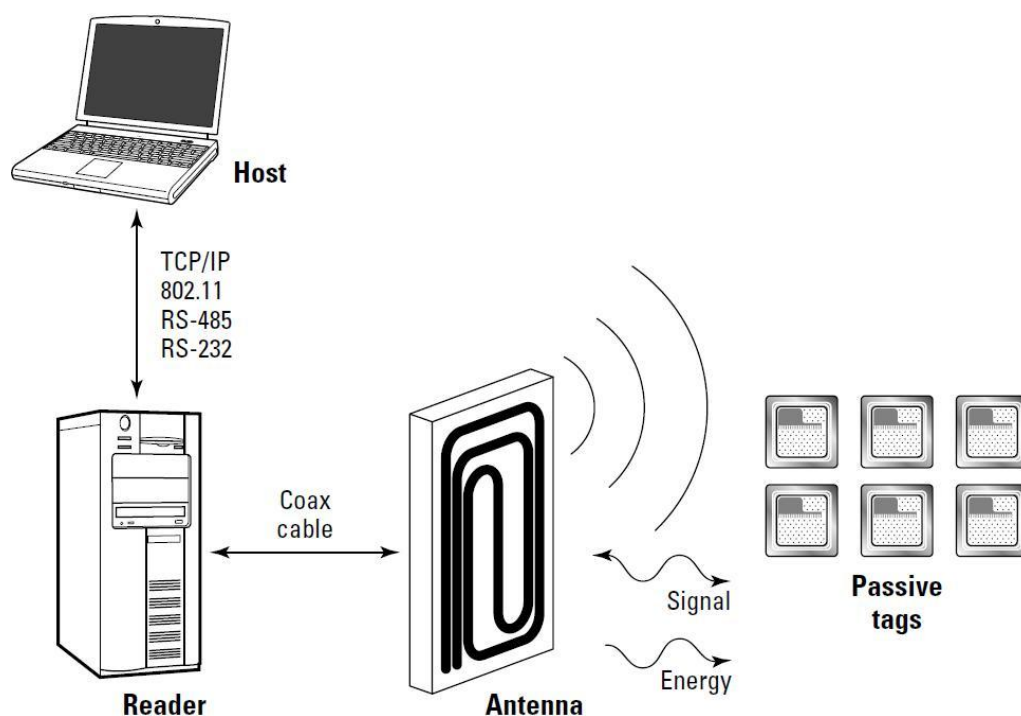


Figura 2 - Arquitetura RFID.
Fonte: SWEENEY (2005).

Resumidamente um sistema com tecnologia RFID funciona da seguinte maneira, um leitor é conectado a uma antena e juntos geram um sinal de rádio frequência e quando uma *tag* entra na área de cobertura do sinal gerado, esta é energizada e aciona seus circuitos para que o leitor possa receber seus dados, processar e mandar para um host que trabalha com suas informações (FERREIRA, 2006).

Um fator essencial para a utilização da tecnologia RFID é a frequência de operação, que com o atual estado da tecnologia se estende de 135 kHz (baixa frequência) para 5,8 GHz (super alta frequência ou microondas), e que tem um feito importante sobre a forma da antena. No Quadro 1 podemos ver uma lista com as frequências e suas principais aplicações.

Frequência	Tipo de Aplicação
Baixa Frequência	Identificação Animal
Menos de 135Khz	Automatização Industrial
	Controle de Acesso

Alta Frequência 13.56Mhz	Smart Label Controle de Acesso Anti-Pirataria Localização de diversos itens com livros, bagagem, vestuário, etc. Prateleira Inteligente Identificação e Monitoramento de Pessoas
Ultra Frequência 433Mhz e 860 a 930Mhz	Fornece a Cadeia de Suprimentos e Logística: - Controle de Inventário - Administração de Estoque - Localização de Itens
Microonda 2.45GHz e 5.8GHz	Controle de Acesso Coleta eletrônica de Pedágios Automação Industrial

Quadro 1 - Frequências e respectivas aplicações.
Fonte: Adaptado de GUEDES (2005).

2.1.2.1 Etiquetas

Uma tag RFID, que é também chamado de etiqueta RFID, consiste em um circuito integrado que contém um processador e memória combinada com uma antena. As etiquetas podem ser classificadas em termos de sua capacidade de armazenamento e se os dados armazenados podem ser reescritos.

- *Single-bit*: É tipo mais simples de etiqueta somente leitura e armazena exatamente um *bit* e não contém um *chip*. Estas *tags* são anexadas a produtos de varejo para proteção contra roubo.
- Identificador único: Estas *tags* como, próprio nome diz, possuem um identificador único, como um código serial, e são utilizados para identificar um objeto. Normalmente um banco de dados é usado para associar o número com maiores detalhes do objeto.
- *WORM tag*: Estas etiquetas podem ser codificadas uma vez com dados não modificáveis, podem ser usadas para guardar códigos seriais.

terem a bateria essas etiquetas são maiores e mais caras, mas tem um raio de comunicação maior. Esta etiqueta por necessitar da onda de rádio do leitor, seu campo também é restrito a área da leitura (GUEDES, 2005).

2.1.2.2 Leitores

O leitor também é chamado de interrogador, ele emite ondas de radiofrequência que alimentam as etiquetas que estiveram no seu campo de radio frequência. Quando a *tag* passa pela área de cobertura da antena, o campo magnético é detectado pelo interrogador, que recebe os dados codificados na etiqueta, descodifica-os e passa para um controlador realizar o processamento. (SANTANA, 2007).



Figura 4 - Leitores RFID
Fonte: KOULEV, 2007

2.1.2.3 Antenas

O papel da antena é determinar como o campo eletromagnético será gerado, realizando de maneira confiável a troca de informações entre o leitor e a etiqueta. A antena pode ser considerada como um elemento a parte nos sistemas de RFID já que não necessariamente precisa estar no mesmo invólucro do leitor, a não ser que

necessita-se maior mobilidade dos equipamentos, neste caso as antenas são usadas juntamente com os leitores (GLOVER; BHATT, 2007).



Figura 5 - Leitor RFID com antena acoplada
Fonte: HECKEL, 2007

2.2 LINGUAGEM DE PROGRAMAÇÃO RUBY

Desde seus primórdios, Ruby, foi construído como uma linguagem de programação que adere aos princípios da programação orientada a objetos (OOP).

OOP é um paradigma de programação que surgiu pela primeira vez na década de 1960, mas se lançou definitivamente em 1980 com C e C++. A idéia central por trás dela é que os sistemas devem ser compostos por entidades individuais, ou objetos, cada qual tem a capacidade de se comunicar com outros objetos ao seu redor. Além disso, cada objeto pode ter a facilidade de armazenar dados internamente (LENZ, 2007).

A linguagem de programação Ruby, para Matsumoto e Flanagan (2009), é “totalmente orientada a objetos, mas é também apropriada para procedimento e formatações de programação funcional.” Dado a isso, é dito que a linguagem é multi-paradigma.

Ruby é independente de plataforma, e possui diversas implementações, como por exemplo, em Java (JRuby) e .NET (*IronRuby* e Ruby.NET), além das implementações para os sistemas operacionais mais utilizados (Windows, Linux e Mac OS) (FRANÇA, 2008).

2.2.1 Histórico

Ela foi criada por Yukihiro Matsumoto (Matz), em 1994, inspirada nas linguagens Python e Perl, com o objetivo de criar uma linguagem que fosse de fácil compreensão e desenvolvimento.

Por volta de 2005, o interesse pela linguagem Ruby subiu em conjunto com o Ruby on Rails, um *framework* de aplicações web escrito em Ruby. Rails é freqüentemente creditada como a aplicação que tornou Ruby "famosa" e a associação é tão forte que ambos são muitas vezes confundidos por programadores que são novos no Ruby (CARVALHO, 2008).

2.2.2 Ruby on Rails

Ruby on Rails surgiu como um aplicativo chamado Basecamp, solução criada pelo dinamarquês David Heinemeier da 37signals. Ele decidiu lançar Rails como software de código aberto para fundamentalmente refazer a maneira como sites *web* são construídas. A primeira versão beta do Rails foi lançado inicialmente em julho de 2004, com a seguinte versão 1.0 em 13 de dezembro de 2005 (LENZ, 2007).

Rails é um *framework* de desenvolvimento de aplicações web escritas na linguagem Ruby. Ele é projetado para tornar as aplicações web de programação mais fácil, fazendo suposições sobre o que cada desenvolvedor precisa para começar. Com ele é possível escrever menos código enquanto se faz mais do que muitas outras linguagens e *frameworks*.

O núcleo do Rails possui o Modelo, Visão, Controlador arquitetura, normalmente chamado apenas de MVC. Seguindo este modelo é possível separar a lógica do negócio da interface, seguir a filosofia de programação DRY, que significa não repetir o mesmo código varias vezes, e deixar claro aonde vai estar cada tipo de código, facilitando a manutenção (RAILS, 2010).

2.2.3 RubyGems

RubyGems é o sistema para gerenciar os pacotes, gems, do Ruby. Um gem é uma aplicação Ruby *package* ou biblioteca, cada um tem um nome (rake, por exemplo) e uma versão (por exemplo, 0.4.16). Estes são gerenciadas no seu sistema usando o comando gem, é possível instalar, remover e consultar (entre outras coisas) qualquer gem usando este comando.

Para instalar o *framework* do Ruby on Rails no Ruby com todas suas dependências, por exemplo:

```
“gem install rails --include-dependencies“
```

O RubyGems além de facilitar a gerência dos pacotes gems, cria padrões para distribuição de programas Ruby e bibliotecas (DUNCAN).

3 MATERIAL E MÉTODOS

Este trabalho consistiu em estudar a tecnologia de identificação por rádio frequência através de simulação. Foi desenvolvido um sistema com Ruby que é capaz de buscar informações disponibilizadas pelo leitor.

No presente trabalho, do ponto de vista da sua natureza foi utilizada a pesquisa aplicada, que teve por objetivo a geração do conhecimento para aplicação de práticas dirigidas à solução de problemas específicos (SILVA e MENEZES, 2001). Do ponto de vista dos procedimentos técnicos foi utilizada pesquisa experimental, quando se determinou o objeto de estudo, foram selecionadas as variáveis que influenciavam no processo e as formas de controle e de observação dos efeitos que a variável produzia no objeto (GIL,1991 apud SILVA e MENEZES, 2001).

3.1 DESCRIÇÃO DO AMBIENTE EXPERIMENTAL

Para o desenvolvimento do sistema foi utilizado o Dia versão 0.97.1 para desenhar o diagrama de entidades do sistema, para codificação utilizamos a IDE NetBeans versão 6.9 com *plugin* para linguagem Ruby versão 1.8.7 e o banco de dados SQLite3.

3.1.1 Configuração do hardware

Apenas uma máquina foi utilizada para desenvolvimento e implementação do sistema, sendo esta um *desktop* com processador Intel Core 2 Duo 2.2 GHz e 2Gb de memória RAM.

O leitor RFID utilizado foi da marca *Touchatag* e trabalha a 13.56 MHz e tem uma distância de leitura de aproximadamente 4 cm, quando utilizado suas etiquetas padrão. Ele foi conectado no computador pela entrada USB. Na Figura 6 observa-se o leitor RFID utilizado.



Figura 6 - Leitor RFID da marca Touchatag

3.1.2 Sistema Operacional

A máquina utilizada rodou o sistema operacional GNU/Linux através da distribuição *Ubuntu* versão 11.04 e todas as atualizações devidamente instaladas.

3.2 APLICATIVOS E BIBLIOTECAS

Para o desenvolvimento do sistema foi utilizado a IDE *Netbeans* para escrita dos códigos da linguagem Ruby e a biblioteca NFC para auxiliar na comunicação com o leitor RFID

3.2.1 Netbeans

Netbeans é uma IDE produzida pela *Sun Microsystems* e hoje mantida pela *Oracle Corporation*, e liberada como software livre e gratuito. Esta possui seu foco na integração com a plataforma Java (também produzida pela Sun), entretanto também suporta programação em outras linguagens como C, C++, *Python*, Ruby entre outros. Porém foi utilizado apenas o suporte à linguagem Ruby para desenvolvimento do sistema.

A versão utilizada foi a 6.9 do *Netbeans* e a linguagem Ruby na versão 1.8.7.

3.2.2 NFC

O NFC é uma biblioteca, ou *gem* na linguagem Ruby, criada para facilitar a comunicação entre uma aplicação desenvolvida na linguagem Ruby e uma outra biblioteca chamada LibNFC, que é escrita em C e tem as habilidades de se conectar com leitores RFID.

4 RESULTADOS E DISCUSSÃO

Este trabalho utilizou um sistema para simular o controle de presença dos participantes nos eventos e os resultados obtidos indicam a viabilidade da utilização das tecnologias e mostram como o sistema deixaria este controle muito mais rápido e seguro.

Existem poucas referências que auxiliem na comunicação do leitor com o sistema na linguagem de programação Ruby, se o sistema fosse desenvolvido na linguagem de programação Java, a comunicação seria mais fácil, existem inúmeros trabalhos realizados neste seguimento utilizando destas tecnologias, com a linguagem Ruby a única referencia encontrada foi a biblioteca NFC, que auxilia a comunicação com dispositivos RFID.

O sistema foi testado com 10 *tags* sendo uma tratada como administrador do sistema, e o restante como participantes.

4.1 ANÁLISE GERAL DO SISTEMA

O sistema proposto é aparentemente bem simples cada participante tem cadastrado o código da *tag* RFID, que estará acoplada em seu crachá e quando o participante passar seu crachá pelo leitor RFID, o sistema será capaz de retornar todos os dados desse participante, se este estiver devidamente cadastrado. Alguns casos, verificando se ele pode ou não ter acesso a determinada parte do evento. O sistema guarda também as informações de quem esteve no evento, cadastrando todos que entrarem e não liberando a entrada, caso este já tenha passado pela entrada uma vez.

4.1.1 Requisitos

Inicialmente foi feito um levantamento de requisitos para analisar o que o sistema necessitaria:

- Manter participantes;
- Manter sub eventos;
- Associar permissões;
- Manter presenças.

4.1.2 Casos de Uso

O administrador do sistema é quem mantém os participantes, os sub eventos e associa as permissões dos participantes em cada sub evento, conforme é detalhado no caso de uso da Figura 7.

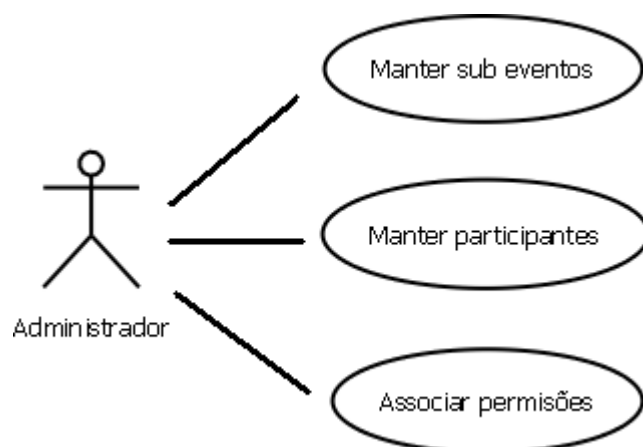


Figura 7 - Casos de uso do Administrador

Já para o cadastro das presenças quem faz é o sistema, o participante apenas passa seu crachá e o sistema verifica se este tem permissão para entrar no sub evento e se tiver libera sua entrada e cadastra sua presença. Na Figura 8 é apresentado o caso de uso deste fluxo.

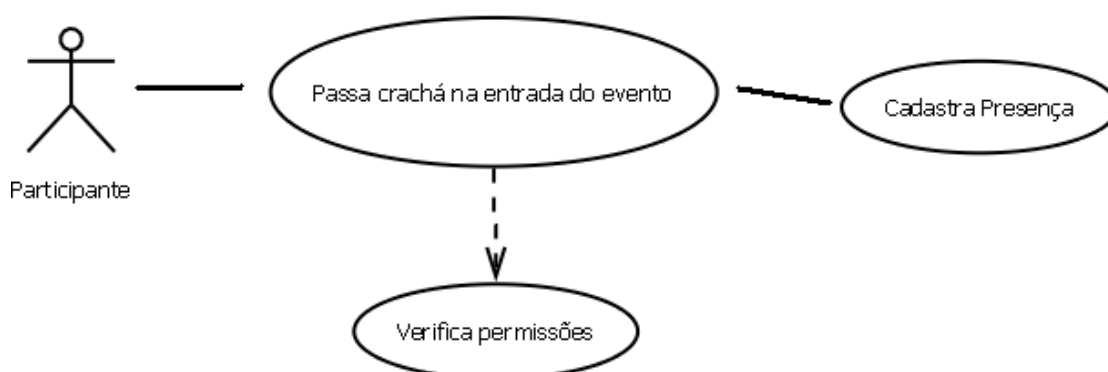


Figura 8 - Caso de uso de Cadastro de presença

4.1.3 Modelo de Entidades e Relacionamentos

Na Figura 9 é apresentado o MER, modelo de entidades e relacionamentos, do sistema.

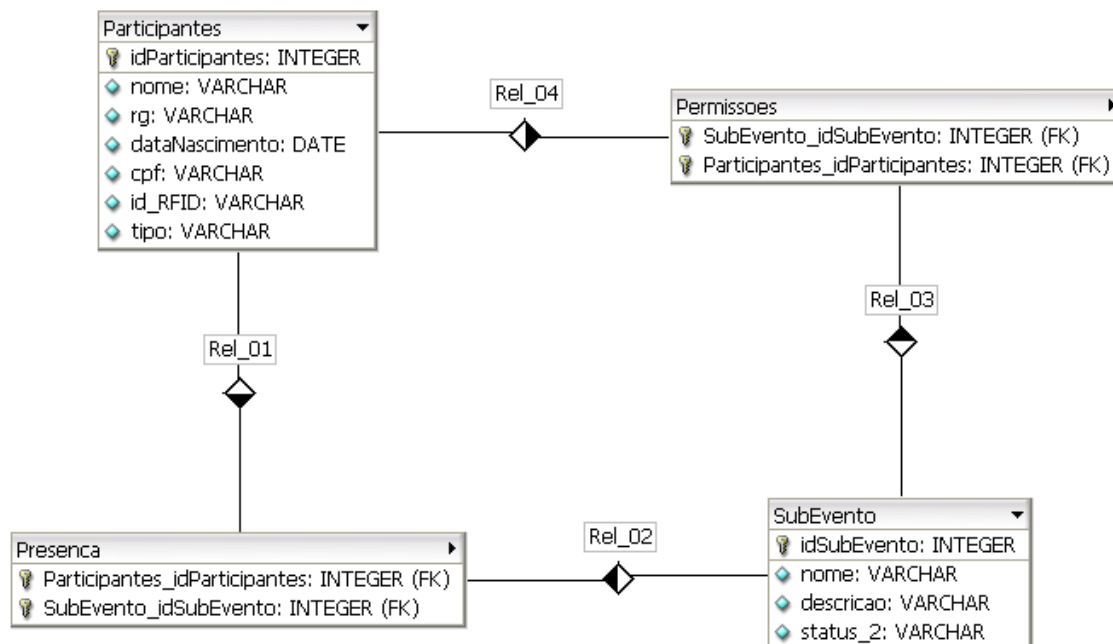


Figura 9 - Modelo de entidades e relacionamentos.

Pelo MER pode-se ter uma visão de como é definida a modelagem do sistema, onde na tabela Participantes estarão todos os participantes cadastrados e SubEvento são as palestras, mini cursos, ou qualquer outra atividade de um evento. Na tabela Permissões serão cadastrados quais sub eventos o Participante pode acessar, e na tabela Presença, ficarão registradas as presenças dos Participantes no SubEvento. No momento em que ele passar seu crachá pelo leitor RFID o sistema verifica se o participante tem acesso ao sub evento e se tiver, libera sua entrada e cadastra sua presença.

Os administradores do sistema também são cadastrados como Participantes, mas possuem um atributo dizendo que são os administradores. O sistema só inicia o registro da entrada dos participantes depois que um administrador passar o seu crachá, ou for colocada a senha do sistema. E em seguida é selecionado um sub evento para cadastrar as presenças, o sistema fica aguardando a entrada dos participantes. O encerramento do cadastro das presenças poderá ser pela passagem do crachá do administrador ou pela senha do sistema.

4.2 TELAS

As telas do sistema foram desenvolvidas utilizando um *framework* para criação de interface, RSense na versão 2.1, que utiliza arquivos da linguagem de marcação YAML, utilizada no Ruby, para desenhar as telas. Na Figura 10 observa-se a tela inicial do sistema.



Figura 10 – Tela inicial do sistema
Fonte: Autoria própria.

Na Figura 10, pode ser visualizada a tela inicial do sistema, onde observa-se uma mensagem inicial e logo abaixo duas abas. Na aba “Sub Eventos” tem-se a lista dos sub eventos cadastrados para o evento e na aba “Cadastro”, é possível

cadastrar novos participantes. Na Figura 11 pode-se observar a aba de cadastro.

A imagem mostra a interface de usuário para o cadastro de participantes em um evento. O título da página é "Bem vindos ao Evento X". Há duas abas: "Sub Eventos" e "Cadastro", com a segunda estando selecionada. O formulário de cadastro contém os seguintes campos: "Nome:" (campo de texto), "RG:" (campo de texto), "CPF:" (campo de texto), "Telefone:" (campo de texto) e "Tag RFID*:" (campo de texto). À direita do campo "Tag RFID*" há um botão "Ler". Abaixo do formulário, há uma mensagem de informação: "*Info: Clique em 'Ler' para obter o código da etiqueta RFID pelo leitor Touchatag ou digite o código da tag."

Figura 11 - Tela de cadastro de novos participantes
Fonte: Autoria própria

No momento em que um participante é cadastrado no sistema, será necessário informar qual o código da etiqueta RFID que vai estar no seu crachá. Para isso o usuário poderá passar a *tag* no leitor para recuperar seu código ou ainda poderá digitar o código manualmente, o que não é muito interessante, pois se este for escrito errado, o participante não será reconhecido na entrada dos sub eventos.

No momento em que o usuário clicar no botão "Ler" na aba de cadastro, no sistema abrirá um *pop-up* com o status da leitura e o código da etiqueta lida, como pode ser visualizado na Figura 12.

A imagem mostra a mesma interface de usuário da Figura 11, mas com um pop-up de leitura de tag RFID aberto. O pop-up tem o título "Leitura de tag RFID" e o conteúdo "Aguardando leitura da tag...". Abaixo disso, há o texto "Leitura: OK" seguido de um campo de texto contendo o código "040A9D9A12581". No rodapé do pop-up, há dois botões: "OK" e "Cancelar".

Figura 12 - Pop-up de leitura de tag.
Fonte: Autoria própria.

Após o cadastro dos participantes nos eventos, pode-se iniciar a leitura da entrada de participantes em qualquer sub eventos.

Na listagem de sub eventos, Figura 10, poderão ser verificados alguns detalhes, como data, horário e status do sub evento. Se o mesmo já foi iniciado, a leitura das entradas estará com status “Em andamento”, e ainda se o evento já foi encerrado a leitura estará como “Finalizado”. Nestes dois casos serão disponibilizadas as opções de verificar a listagem dos participantes que estão ou estiveram presente no sub evento, no caso as palestras. Se algum sub evento for cancelado, o status deste poderá ser alterado para “Cancelado” nos detalhes do mesmo, que podem ser acessados pelo clique no nome do evento que é um *link*.

Quando o sub evento ainda não começou e ainda não foi iniciada a sua leitura, o status o mostrará como “Aguardando” e será possível iniciar a leitura dos participantes pelo botão “Iniciar”, que apresentará um *pop-up* com uma lista dos usuários que já entraram e passaram seu crachá pelo leitor, como é visto na Figura 13, no instante em que o leitor lê a etiqueta, o sistema busca pelas informações do participante associado ao *tag* e adiciona seu nome na lista do pop-up. Na Figura 13 verifica-se o *Pop-up* da tela de leitura dos participantes na entrada dos sub eventos.

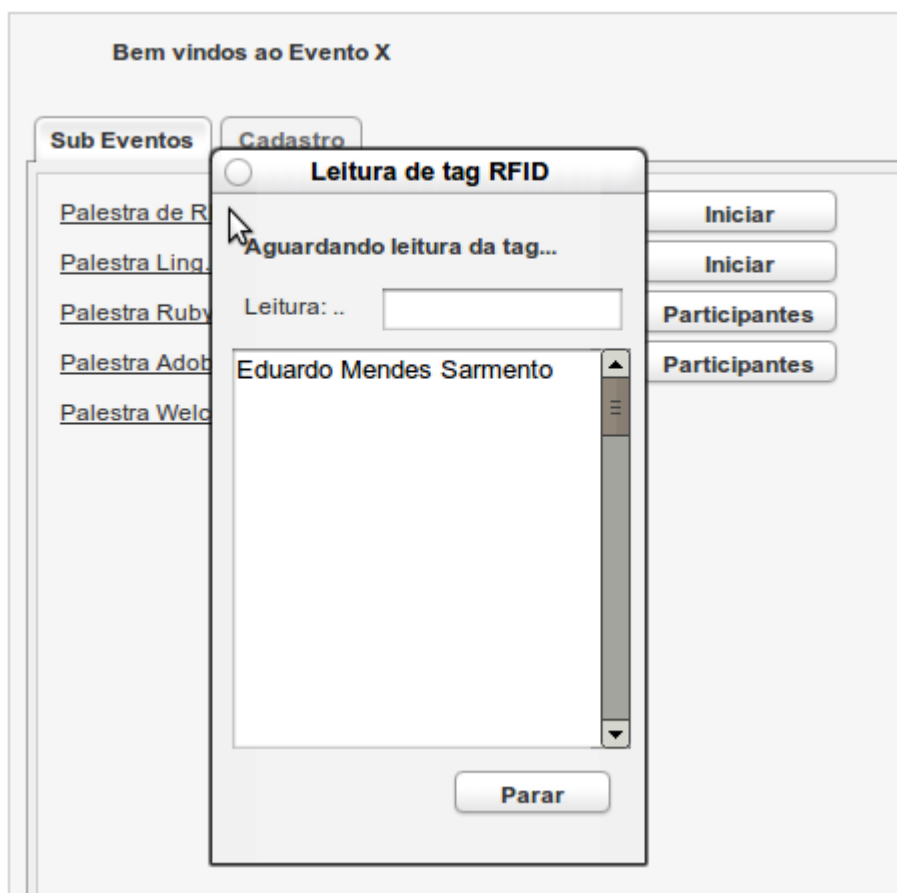


Figura 13 - Pop-up de tela de leitura dos crachás.
Fonte: Autoria própria.

Quando o sub evento acabar, basta que o usuário aperte o botão “Parar” e o sistema vai parar de fazer a leitura, gravar a lista de participantes que esteve presente, e mudará o status do sub evento para “Finalizado”.

5 CONCLUSÃO

Com esta aplicação verificou-se que a implantação da tecnologia RFID para controlar a presença de participantes em eventos torna esta atividade muito mais simples, a maior dificuldade é a comunicação com os leitores, que são hardwares externos e se não tiverem uma boa documentação pode ser muito complicado fazer essa comunicação com determinadas linguagens.

Outro fator que pode ser muito interessante é que se usados outros leitores e *tags* RFID, a leitura das etiquetas poderia ser feita apenas com o participante passando próximo do leitor, e se este estiver próximo da porta de entrada, assim que entrarem os participantes estes já serão cadastrados.

Neste trabalho, os resultados indicaram que é possível a utilização de etiquetas RFID, adaptadas em crachás, para agilizar o acesso e controlar a presença de participantes em eventos, utilizando um sistema na linguagem Ruby para fazer a leitura e o armazenamento das informações.

As chaves para o sucesso da implantação de um projeto de RFID no padrão EPC são principalmente: mudanças profundas nos processos atuais, escolha correta dos produtos que serão utilizados para controle no projeto e, ainda buscar novas formas de agregar valor a implantação fazendo com que o custo do projeto seja diluído mais facilmente.

Uma pesquisa válida para trabalhos futuros é o de desenvolver um sistema que disponibilize serviços para o sistema acadêmico e possa fazer o controle dos alunos na faculdade ou em qualquer estabelecimento de ensino.

REFERÊNCIAS

BUCHSBAUM, André. **Do bestial ao genial: frases da política**. Rio de Janeiro: Ediouro, 2006.

DUNCAN, Paul. **RubyGems User Guide**. Disponível em <<http://docs.rubygems.org/read/book/1>>. Acesso em 12 de março, 2011.

FERREIRA, Levi. **A Tecnologia de RFID no padrão EPD e soluções para implementação desta tecnologia em empilhadeiras**. São Paulo, 2006.

FRANÇA, Rafael M.. **A história do Ruby**. Disponível em <<http://ligeirinhos-ufs.blogspot.com/2008/10/histria-do-ruby.html> >. Acesso em 12 de abril, 2011.

GLOVER, Bill; BHATT, Himanshu. **Fundamentos de RFID: Teoria em Prática**. Rio de Janeiro: Alta Books, 2007.

GUEDES, Rodrigo. **Desenvolvimento de uma aplicação frente de caixa automatizada para varejo utilizando Java e RFID**. Maringá, 2006

HACKNMOD. **RFID Tutorial – Everything You Need**. Disponível em <<http://hacknmod.com/hack/rfid-tutorial-everything-you-need/>>. Acesso em 20 de abril, 2011.

HANSEN, Wolf-Ruediger. **RFID for optimization of Business Processes**. Wiley, NJ, 2008.

HECKEL, Andrei Pedro. **Identificação por radiofrequência (RFID) Estudo teórico e experimentação via Simulação**. Novo Hamburgo, 2007.

KOULEV, Jean. **RFID readers**. Disponível em <http://www.flickr.com/photos/jean_koulev/galleries/72157622642249739>. Acesso em 20 de Abril, 2011.

LENZ, Patrick. **Build your own Ruby on Rails Web Application**. SitePoint, 2007

MATSUMOTO, Y; FLANAGAN, D.. **A Linguagem de Programação Ruby**. Alta Books, 2008.

PREEMPTIVE. **A Report of the FTC RFID workshop**. Disponível em <<http://www.preemptivemedia.net/ftc/index.html>>. Acesso em 13 de março, 2011.

RAILS, Ruby. **Desenvolvimento web sem dor**. Disponível em <<http://www.rubyonrails.pro.br/>>. Acesso em 18 de março, 2011.

Ruby Brasil. **O que é Ruby?**. Disponível em <http://ruby-br.org/?page_id=15>. Acesso em 22 de abril, 2011.

SANTANA, Sandra R. M.. **RFID – Identificação por Rádio Frequência**. Disponível em:
<http://www.wirelessbrasil.org/wirelessbr/colaboradores/sandra_santana/rfid_01.html>. Acesso em: 15 de junho de 2011.

SILVA, Edna Lucia; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. 3. ed. rev. Atual. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

SWEENEY, Patrick. **RFID for Dummies**. Wiley Publishing, 2005