

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

PROTÓTIPO DE INVENTÁRIO AUTOMATIZADO COM
RFID

RAFAEL ANTONIO LOSI

BLUMENAU
2016

RAFAEL ANTONIO LOSI

**PROTÓTIPO DE INVENTÁRIO AUTOMATIZADO COM
RFID**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Dr. Mauro Marcelo Mattos

**BLUMENAU
2016**

PROTÓTIPO DE INVENTÁRIO AUTOMATIZADO COM RFID

Por

RAFAEL ANTONIO LOSI

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: _____
Prof. Mauro Marcelo Mattos, Orientador, Doutor – FURB

Membro: _____
Prof. Miguel Alexandre Wisintainer, Mestre – FURB

Membro: _____
Prof. Aurélio Faustino Hoppe, Mestre – FURB

Blumenau, 08 de dezembro de 2016

Dedico este trabalho a meus pais, esposa e filho.

AGRADECIMENTOS

Agradeço a meus pais pelo esforço que fizeram para me proporcionar a oportunidade de realizar essa graduação.

Agradeço a minha esposa e filho pela compreensão e apoio que me deram até a conclusão do curso.

Agradeço ao Prof. Dr. Mauro Marcelo Mattos pela dedicação e orientação passada possibilitando a realização deste trabalho.

A distância entre a insanidade e a genialidade
é medida pelo sucesso.

Bruce Feirstein

RESUMO

O presente trabalho descreve uma solução para o problema de gestão patrimonial utilizando *tags* RFID (Radio-Frequency IDentification) na frequência de 13,56 MHz e apresenta a integração de um *hardware* leitor de baixo custo baseado no Arduino Uno R3. O *hardware* será responsável pela identificação das *tags* RFID, permitindo vincular esta identificação a um bem patrimonial e também realizar o inventário patrimonial por meio dessa identificação com o leitor. Foi desenvolvido uma interface *desktop* para permitir a comunicação do hardware com o software patrimonial para alimentar as informações no sistema pela interface USB. Também foi desenvolvido um módulo Web para permitir o cadastro e movimentação dos bens patrimoniais. Para validar o projeto, foram especificados casos de uso e diagramas MERs (Modelo Entidade Relacionamento). Os resultados obtidos permitiram identificação de bens patrimoniais pela tecnologia de RFID, realização de transferências via interface web e delegação de responsabilidade do inventário patrimonial por departamento e local. Desta forma os objetivos iniciais foram alcançados e o desenvolvimento com a plataforma arduino para construção de um hardware se mostrou prática, rápida e de baixo custo.

Palavras-chave: Inventário. Patrimônio. Conferência. RFID. Radiofrequência.

ABSTRACT

The present work describes a solution to the problem of asset management using RFID tags in the 13.56 MHz frequency, and presents the integration of a low-cost reader hardware based on the Arduino Uno R3. The hardware will be responsible for the identification of the RFID tags, allowing to link this to identify a patrimonial asset and also perform the inventory of assets through this identification with the reader. A desktop interface was developed to allow the communication of the hardware with the patrimonial software to feed the information in the system through the USB interface. A Web module was also developed to allow the registration and movement of assets. To validate the project, use cases and MERs (Entity Relationship Model) diagrams were specified. The results obtained allowed the identification of patrimonial assets by RFID technology, realization of transfers through web interface and delegation of responsibility of the patrimonial inventory by department and local. In this way the initial objectives were achieved and the development with the arduino platform for construction of a hardware proved practical, fast and low cost.

Key-words: Inventory. Patrimony. Conference. RFID. Radio frequency.

LISTA DE FIGURAS

Figura 1 - Layout da empresa	14
Figura 2 - Sistema de identificação por rádio frequência	20
Figura 3 - Etiqueta eletrônica (<i>tag</i>).....	21
Figura 4 - Cartão e chaveiros contendo tags MIFARE	22
Figura 5 - Placa Arduino UNO.....	23
Figura 6 - Conectores Arduino UNO	24
Figura 7 - Shield Ethernet.....	25
Figura 8 - Módulo de radiofrequência RC522.....	25
Figura 9 - Módulos do sistema	29
Figura 10 - Casos de uso do sistema	30
Figura 11 - MER dos cadastros	32
Figura 12 - MER entidades transferência	33
Figura 14 - Pacotes do módulo desktop	35
Figura 15 - Identificação do Bem	35
Figura 16 - Identificação de <i>tag</i>	36
Figura 17 - Realizando a sincronização do inventário	39
Figura 18 - Circuito elétrico Arduino com RC522.....	40
Figura 19 - Conexão do Patrimonio.desktop com o Leitor RFID	41
Figura 20 - Monitor Serial: <i>tag</i> lida.....	42
Figura 21 - Monitor Serial: Lista de <i>tags</i> armazenadas.....	42
Figura 22 - Diagrama de arquitetura.....	43
Figura 23 - Arquitetura cliente - servidor.....	48
Figura 24 - Cadastro de Departamentos	54
Figura 25 - Cadastro de Bens	54
Figura 26 - Transferência de bens	55
Figura 27 - Destino da transferência.....	55
Figura 28 - Aprovação de transferência	56
Figura 29 - Cadastro de período de inventário	56
Figura 30 - Leitor RFID	57

LISTA DE QUADROS

Quadro 1 - Frequências de operação do RFID	21
Quadro 2 - Características dos trabalhos correlatos	27
Quadro 3 - Requisitos funcionais	31
Quadro 4 - Requisitos não funcionais	31
Quadro 5 - Comunicação Patrimonio.desktop com Leitor RFID	37
Quadro 6 - Solicitando a lista de identificações ao leitor	38
Quadro 7 - Identificando bens pela tag RFID.....	39
Quadro 8 - Método de conversão da identificação para hexadecimal.....	40
Quadro 9 - Container gerado a partir do EF6	44
Quadro 10 - Classe Transferencia	45
Quadro 11 - Classe Transferencia na camada de negócio.....	45
Quadro 12 - Classe Sistema.....	46
Quadro 13 - Classe Repositorio	47
Quadro 14 - Página bens.html	48
Quadro 15 - Script da página bens.html	49
Quadro 16 - API do serviço de transferência	50
Quadro 17 - Classe Transferencia no pacote Models.....	51
Quadro 18 - Programa principal do Leitor RFID	52
Quadro 19 - Método NovaTag.....	53
Quadro 20 - Método ArmazenaTag.....	53

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

CRC – Cyclic redundancy check

DB – Database

EEPROM – Electrically-Erasable Programmable Read-Only Memory

EF6 – EntityFramework 6

EPC – Electronic Product Code

GPS – Global Position System

HTTP – Hypertext Transfer Protocol

IEC – International Electrotechnical Commission

IIS – Internet Information Services

INMETRO – Instituto Nacional de Metrologia, Qualidade e Tecnologia

ISO – International Organization for Standardization

KB – Kilobyte

MER – Modelo Entidade Relacionamento

MVC – Model View Controller

NFC – Near-field Communication

REST – Representational State Transfer

RFID – Radio-Frequency IDentification

SIG – Sistema de informação gerencial

SRAM – Static Random Access Memory

USB – Universal Serial Bus

ZI – Zona de Interrogação

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	16
1.2 ESTRUTURA.....	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 GESTÃO PATRIMONIAL.....	17
2.1.1. SIG na gestão patrimonial.....	18
2.2 IDENTIFICAÇÃO POR RADIOFREQUENCIA (RFID).....	19
2.3 ARDUINO.....	22
2.4 TRABALHOS CORRELATOS	26
2.4.1. Protótipo de um sistema de identificação eletrônica de animais através de rádio frequência.....	26
2.4.2. Sistema de informação aplicado à cadeia de suprimentos utilizando tecnologia RFID. 26	
2.4.3. Sistema de controle de patrimônio via RFID.	27
2.4.4. Características dos trabalhos correlatos.	27
3 DESENVOLVIMENTO	29
3.1 MODELAGEM	29
3.3.1 Diagrama de casos de uso	29
3.2 ESPECIFICAÇÃO DOS REQUISITOS	30
3.3 MODELO ENTIDADE RELACIONAMENTO	32
3.4 MÓDULO PATRIMÔNIO WEB.....	34
3.5 MÓDULO PATRIMÔNIO DESKTOP	35
3.5.1. Identificação de Bens	35
3.5.2. Realização de inventário	38
3.6 MÓDULO LEITOR RFID	40
3.6.1. Protocolo de comunicação	41
3.7.1. Pacote Patrimonio.dal	43
3.7.2. Pacote Patrimonio.dll	45
3.7.3. Pacote Patrimonio.web.....	47
3.7.4. Módulo Leitor RFID	51
3.7.5. Operacionalidade da implementação	53
3.7.6. Configuração do ambiente	54

3.7.7. Movimentação de bens.....	55
3.7.8. Realização de inventário	56
3.7 RESULTADOS E DISCUSSÕES.....	56
4 CONCLUSÕES.....	59
5 REFERÊNCIAS.....	61
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO	64

1 INTRODUÇÃO

“O dinamismo do mundo globalizado vem exigindo grandes mudanças das organizações” (MACHADO, 2014, p. 1). Segundo Caseiro (2013, p. 8) “as empresas [...] buscam a internacionalização primordialmente nos maiores e mais dinâmicos mercados do globo como forma de reforçar os laços com os clientes”. Essa expansão deve vir acompanhada de um rigoroso controle de gestão para manter a acurácia da informação e antecipar possíveis problemas para que eles não se tornem maiores do que o negócio proposto.

As organizações, em face de suas complexidades e seu tamanho, transacionam constantemente seus recursos patrimoniais, ora adquirindo, ora vendendo, ou trocando-os, sempre com o objetivo maior de otimizar os processos e proporcionar maior satisfação, tanto para cliente externo como interno. A gestão dos recursos patrimoniais é, portanto, atividade fundamental na administração empresarial. (POZO, 2010, p.190)

Atualmente manter o controle patrimonial com informações atualizadas referentes a localização, responsabilidade e estado do bem, torna-se uma atividade difícil devido a constante movimentação dos mesmos. “A maior dificuldade relatada na gestão patrimonial é a mudança de localização dos bens nos órgãos” (LEÃO; SOUSA, 2014, p. 13). A necessidade de realização de inventários em curtos períodos de tempo para atualizar as informações. Essa é uma tarefa difícil para algumas organizações, visto que isso demanda alocação de recursos pessoais para o levantamento dos bens e controle para que não haja alterações durante o processo de inventário, o que poderia tornar o trabalho inútil ao final. Outra dificuldade relacionada ao processo de inventário é a identificação dos bens, já que comumente é realizada através de plaquetas com um código para tal propósito, tornando a conferência um trabalho demorado, principalmente em locais de difícil acesso ou onde há pouca visibilidade (LEÃO; SOUSA, 2014).

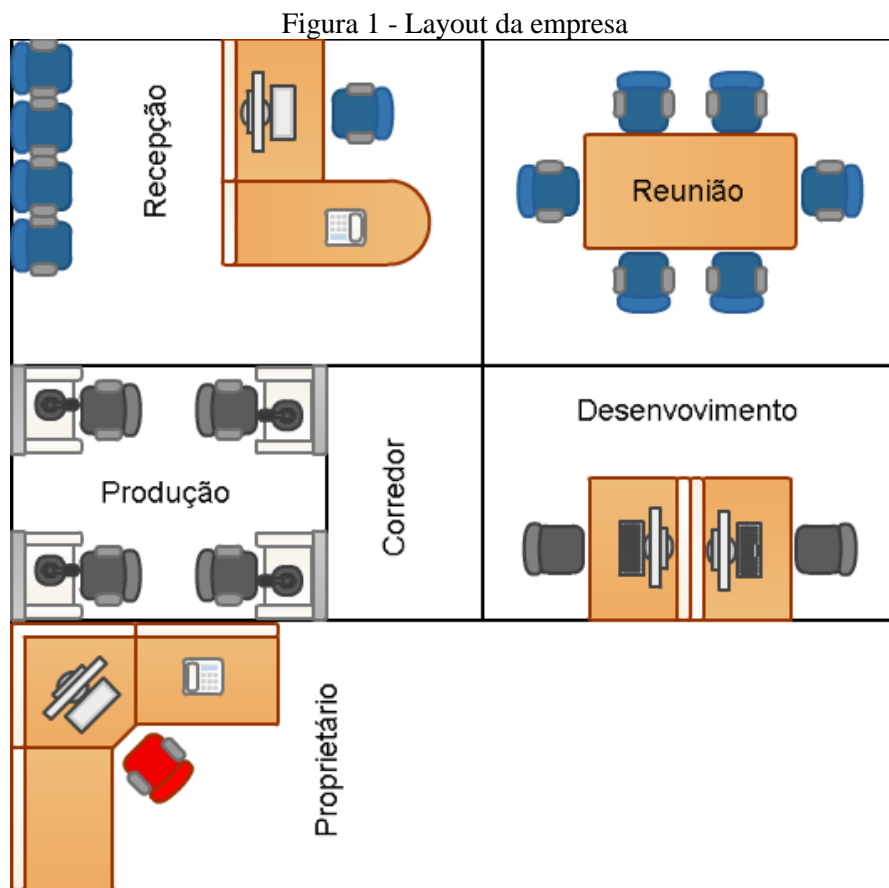
A tecnologia de identificação por radiofrequência (RFID) possibilita a rápida identificação de itens mesmo em locais de difícil acesso. Com a variedade de frequências de operação disponível, essa tecnologia pode ser aplicada a diversos ramos necessitando apenas adaptar o processo para a necessidade do cliente.

Este trabalho apresenta o desenvolvimento de uma solução aliando a tecnologia RFID a um sistema de informação para identificação de bens patrimoniais e movimentações dos mesmos em tempo real tornando o processo de inventário descomplicado e ágil. Para leitura das etiquetas RFID foi desenvolvido um leitor utilizando uma placa arduino.

Para demonstrar o problema que este trabalho trata, foi elaborado o seguinte cenário: Uma empresa de pequeno porte do segmento têxtil é dividida em alguns setores com os seguintes ativos patrimoniais:

- na recepção quatro cadeiras para os visitantes, uma mesa com cadeira, um computador e um telefone para a recepcionista;
- na sala de reuniões uma mesa com seis cadeiras;
- no setor de desenvolvimento duas mesas com cadeiras e computadores;
- na produção quatro mesas com máquinas de costura e cadeiras;
- na sala do proprietário da empresa com uma mesa, cadeira, computador e telefone.

A Figura 1 representa o Layout do ambiente descrito para melhor entendimento.



Fonte: Elaborado pelo autor.

Com o passar do tempo o desgaste e manutenção de equipamentos é inevitável. Quando uma cadeira da sala de produção se quebra a mesma vai para manutenção e outra deve ser colocada em seu lugar para que o colaborador continue seu trabalho, então é emprestada uma cadeira da sala de reunião.

Na semana seguinte uma reunião é marcada entre o dono da empresa, a estilista do desenvolvimento, a supervisora de produção e três representantes do fornecedor de um

produto, mas falta uma cadeira na sala de reuniões então é emprestada uma cadeira da sala de desenvolvimento já que uma colaboradora entrou de férias nessa semana.

Agora essa movimentação de bens começa a ficar complexa para ser revertida, então uma solução comumente aplicada nas empresas do mundo real é utilizada nesse caso: a recepcionista começa a lista de bens, localização e responsável em uma planilha eletrônica. O levantamento físico é realizado e a lista de bens começa a ser controlada.

Parece uma solução razoável, o problema é que os colaboradores não devem ser incumbidos da tarefa de atualizar status de localização ou responsável pelo bem, afinal isso não faz parte do negócio da empresa nem do processo de produção. Então quando a cadeira que estava em manutenção retorna para a empresa, chega em período de intervalo do almoço, quando a recepcionista que mantém a lista de bens não está presente e um colaborador recebe a cadeira e deixa na sala de desenvolvimento, afinal está faltando uma cadeira lá. Agora a planilha de controle já não representa mais a real situação dos bens.

Então o computador da recepção começa a ter problemas de desempenho dificultando o trabalho da recepcionista sendo necessário enviá-lo à manutenção. Agora o controle de bens já não está mais disponível e talvez nem seja mais recuperado. Nesse caso o levantamento físico deve ser iniciado novamente e ainda assim o histórico de movimentações não estará mais disponível, caso a empresa não possua um servidor para efetuar o backup de seus arquivos, evitando assim a perda de informação.

Esse é um cenário comumente vivenciado em empresas que não possuem a cultura de adoção de um sistema de informação para gestão de seus processos. A dinâmica do ambiente empresarial conduz a inevitável expansão das atividades. Expansão significa também aumento do ativo patrimonial e proporcional aumento de movimentação de bens. Se para o exemplo de um pequeno negócio o problema pode se tornar complexo, o problema para uma organização com várias unidades pode se tornar incontrolável.

A falta de controle do patrimônio gera problemas não apenas quando se trata de saber a localização de um bem ou a quantidade em estoque, mas principalmente porque o balanço patrimonial faz parte dos demonstrativos contábeis que a empresa deve apresentar ao final de cada exercício.

No atual cenário onde a informação acompanha as pessoas em vários meios, os negócios acontecem rápido, assim como as mudanças também. Neste contexto, manter o negócio e principalmente o patrimônio sob controle é uma tarefa difícil em face dessas mudanças.

Utilização de recursos pessoais para manter o patrimônio atualizado é uma tarefa onerosa por demandar muito tempo de levantamento e conferência de inventário. Um sistema de informação para gestão do patrimônio é a solução ideal para o problema, mas ainda tem a questão de tempo de realização de todo o processo do inventário para ser resolvida. É nesse sentido que a tecnologia RFID pode auxiliar.

A tecnologia de identificação por rádio frequência (RFID) possibilita rápida identificação a um custo cada vez menor tornando a relação entre o custo e benefício cada vez mais atrativa. Atualmente já é possível encontrar as etiquetas (*tags*) para identificação por menos de R\$ 2,00 reais, módulos para identificação, para desenvolver leitores, por cerca de R\$ 30,00 reais ou leitores prontos para uso por cerca de R\$ 50,00. O custo é baixo considerando o benefício que a automação do processo traz ao negócio possibilitando a implantação de organizações de qualquer porte.

1.1 OBJETIVOS

O objetivo geral do trabalho proposto é o desenvolvimento de uma solução para controle patrimonial e inventário utilizando a tecnologia de RFID.

Os objetivos específicos do trabalho proposto são:

- a) integrar um leitor de etiquetas RFID com arduino e modulo de rádio frequência;
- b) desenvolver um sistema web para gestão patrimonial;
- c) criar um cenário de testes para validar a solução.

1.2 ESTRUTURA

Os capítulos seguintes estão estruturados da seguinte forma: No capítulo 2 a fundamentação teórica apresenta os conceitos de gestão patrimonial, identificação por rádio frequência (RFID) e arduino, que é a placa micro controladora escolhida para o desenvolvimento de um leitor RFID para integrar com o sistema; O capítulo 3 trata o desenvolvimento do sistema proposto, iniciando com a configuração de ambiente, movimentações e requisitos e finaliza com a modelagem do sistema. Também será explicado como foi desenvolvido o leitor RFID e o protocolo de comunicação utilizado para integração entre leitor e sistema; No capítulo 4 são apresentadas as conclusões e futuras extensões.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os seguintes temas: gestão patrimonial, a tecnologia de identificação por radiofrequência (RFID), Arduino, além de trabalhos correlatos.

2.1 GESTÃO PATRIMONIAL

Para Fijor (2014, p. 40), “O controle patrimonial [...] compreende uma sequência de atividades que iniciam com a aquisição e terminam com a retirada ou alienação do bem do patrimônio”. Durante esse período ocorrem movimentações físicas ou até mudanças na estrutura organizacional alterando localização ou responsabilidade dos bens, necessitando controle dessa informação.

A maior dificuldade de manter a informação atualizada é quanto a localização de bens móveis, que de acordo com Nunes (2015 apud SANTOS 2012) “bens móveis são aqueles que podem ser deslocados sem alteração de sua forma e geralmente constituem a maioria dos bens das instituições”. Devido a isso Fijor (2014, p. 41 apud TORRES JÚNIOR; SILVA, 2003) afirma que “toda organização deve possuir em sua estrutura administrativa uma unidade ou departamento responsável pelo controle do patrimônio”.

Com a globalização o sucesso das empresas está cada vez mais ligado a capacidade de gestão eficiente para manter a competitividade. Informações precisam estar atualizadas e acessíveis a qualquer momento em qualquer lugar para rápida tomada de decisão. (BOWERSOCKS; CROSS, 1996 apud SOUZA; CARVALHO; LIBOREIRO, 2006).

Os recursos patrimoniais constituem os elementos primordiais para uma organização poder operar, produzir produtos e serviços que irão atender às demandas de mercados (POZO, 2010).

Para Dalfovo (2004), com a globalização das organizações surge a especialização e departamentalização da produção, com isso observa-se que não é mais possível trabalhar com um tipo de gerenciamento informal. Com os negócios ocorrendo com tamanha rapidez é difícil acompanhar de perto todos os processos. É aí que atua um sistema de informação gerencial, guiando os negócios, controlando e informando os gestores sobre o andamento do processo. (CRUZ, 1998)

A informação é fundamental para as organizações, constituindo-se, em um ferramental seguro e eficaz, pois sem ela, as tomadas de decisão não contêm a qualidade necessária, podendo ocasionar resultados ruins para as organizações, tais como o não atingimento dos objetivos traçados. (PORTO; BANDEIRA, 2006).

Para Narciso (2008) “Um sistema de bens patrimoniais é aquele que contém todas as informações de bens patrimoniais da empresa, isto é, tipo de bem, onde fica, quem é o responsável, número ou identificação do patrimônio”. Esse conjunto de dados representa as

informações mínimas para que se possa ter o controle de localização e responsabilidade dos bens. Nunes (2015) afirma que “é de suma importância para a instituição, que todo bem móvel possa ser identificado individualmente, estando atrelado a um local específico e sob a responsabilidade de um servidor”.

A manutenção das informações acerca dos bens patrimoniais não é apenas uma necessidade para controle operacional, o artigo 176 da Lei n.º 6.404, de 15 de dezembro de 1976, define que o balanço patrimonial é um dos demonstrativos financeiros que a organização deverá entregar ao final de cada exercício social, que deverão exprimir com clareza a situação do patrimônio da companhia e mutações ocorridas no exercício.

As demonstrações contábeis são uma representação monetária estruturada da posição patrimonial e financeira em determinada data e das transações realizadas por uma entidade no período findo nessa data. O objetivo das demonstrações contábeis de uso geral é fornecer informações sobre a posição patrimonial e financeira, o resultado e o fluxo financeiro de uma entidade, que são úteis para uma ampla variedade de usuários na tomada de decisões. As demonstrações contábeis também mostram os resultados do gerenciamento, pela Administração, dos recursos que lhe são confiados. (ZANLUCA, 2009 apud INBRACON (NPC 27))

2.1.1. SIG na gestão patrimonial

Para O'Brien e Marakas (2013), um sistema de informação gerencial (SIG) gera produto de informação e apoia na tomada de decisão da administração. Para Porto e Bandeira (1999) “a essência do planejamento e do controle é a tomada de decisões, que depende de informações oportunas e confiáveis”.

Gestão patrimonial não é diferente, é um grande desafio manter o controle dos ativos da empresa, principalmente em face da expansão dos negócios e a descontrolada movimentação de bens para atender os colaboradores que necessitam de matérias a pronta entrega e rápida reposição. Uma das atividades mais importantes na administração dos recursos patrimoniais é registrar e controlar todos os bens patrimoniais da empresa (POZO, 2010).

Um sistema de gestão patrimonial deve permitir a rápida localização e identificação de bens. Assim que um item necessita de reposição, o gestor deve ser capaz de identificar facilmente a localização de outro item da mesma espécie disponível para o colaborador e de fácil atualização das informações notificando os interessados para tomar ciência da movimentação. Esse processo de controle do patrimônio é feito através do inventário patrimonial, que se atualizado constantemente e corretamente, se torna um item estratégico fundamental para os executivos e gerentes da empresa. (CPCON, 2011).

[...] a competitividade de uma empresa é determinada pela qualidade de seus recursos, pelo conhecimento que é capaz de produzir e pela capacidade de aplicar a ciência, a tecnologia e o conhecimento na produção de bens e serviços cada vez mais eficientes [...] (DALFOVO, 2004).

Para minimizar os erros de operação durante o inventário patrimonial o sistema deve ser o mais automatizado possível, requerendo o mínimo de intervenção manual do operador durante a realização do inventário. De acordo com FIJOR (2014 apud NEVES, 2009) “Uma das formas de se conhecer a realidade patrimonial de uma organização é por intermédio da realização de inventários patrimoniais, os quais são realizados de maneira manual por servidores, muitas vezes sem o devido treinamento e com prazo exíguo para o cumprimento de suas atribuições, o que leva a margens elevadas de erros”.

Bhuptani e Maradpour (2005) demonstram a melhoria significativa nos processos de identificação ao comparar o advento da tecnologia de identificação por radiofrequência à invenção da roda.

Essas aplicações bem conhecidas da roda melhoram e facilitam a nossa vida, porém elas representam uma pequena parte da infinidade formas com que a roda é empregada hoje. Quase da mesma forma, porém de um modo significativamente mais acelerado, a invenção da tecnologia de identificação por radiofrequência (RFID) está seguindo um caminho paralelo como facilitadora onipresente para que as coisas sejam feitas de um jeito melhor do que estamos acostumados a fazer. (BHUPTANI; MARADPOUR, 2005).

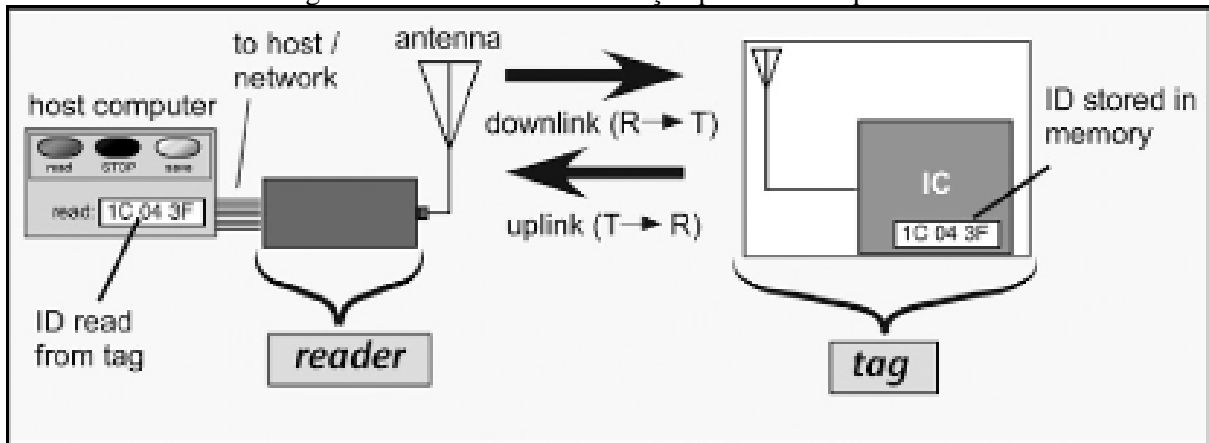
O processo de identificação consiste na etiquetagem dos bens utilizando uma etiqueta eletrônica (*tag*), equipada com um *transponder* contendo um identificador único. Quando este entra no raio de alcance do leitor, chamado de zona de interrogação (ZI), é energizado e emite um sinal com os dados contidos em sua memória.

2.2 IDENTIFICAÇÃO POR RADIOFREQUENCIA (RFID)

O RFID (Radio-Frequency IDentification) é um método de propagação de dados sem fio e tem sido utilizado desde a segunda guerra mundial como forma de identificar aviões aliados e inimigos. Seu primeiro registro de patente ocorreu no ano de 1973. (PAULI, 2016).

Um sistema de identificação por rádio frequência é composto por dois componentes principais: Um leitor e uma *tag*. A Figura 2 mostra um sistema completo de identificação por rádio frequência contendo um computador ligado a um leitor e antena se comunicando com uma *tag* contendo um identificador. (DOBKIN, 2013).

Figura 2 - Sistema de identificação por rádio frequência



Fonte: Dobkin (2013).

Em um sistema passivo de RFID, o leitor utiliza uma antena que emite ondas eletromagnéticas a um raio de alcance definido de acordo com a frequência de operação utilizada. Essas ondas energizam a *tag* que emite um sinal com o conteúdo armazenado em sua memória que é captado de volta pelo leitor. (BHUPTANI; MARADPOUR, 2005).

Um dos aspectos mais importantes ao adotar a solução de RFID é definir a frequência de operação, já que ela delimita o alcance, velocidade de comunicação dos dados e também o custo da solução. As frequências mais baixas são de menor alcance e velocidade, mas podem ultrapassar obstáculos mais facilmente, ideal para quando a *tag* necessita ser colocada dentro de metais e líquidos, o que dificulta a leitura em frequências mais altas. Por isso os dados técnicos não bastam para decidir com qual frequência trabalhar, é necessário avaliar o ambiente no qual será utilizado para evitar interferências e dificuldades de leitura o que atrapalharia o funcionamento da solução. (DIAS; BADALEI, 2012)

As frequências de operação são classificadas em: baixa (LF de 125 kHz), alta (HF de 13,56 MHz) ou ultra alta (UHF de 840~960MHz), sendo as duas primeiras as frequências de trabalho comumente encontradas em *tags* passivas e a última em *tags* ativas. (BHUPTANI; MARADPOUR, 2005, p.57). O Quadro 1 demonstra as frequências de operação, características e aplicações.

Quadro 1 - Frequências de operação do RFID

Banda de Frequência	Características	Aplicações Típicas
Baixa: 100 a 500 KHz	- Faixa de curta até média leitura - Baixo custo - Baixa velocidade de leitura	- Controle de acesso - Identificação de animais - Controle de inventário
Média: 10 a 15 MHz	- Faixa de curta até média leitura - Potencialmente de baixo custo - Média velocidade de leitura	- Controle de acesso - Smart Card
Alta: 850 a 950 MHz e 2,4 a 5,8 GHz	- Faixa de larga leitura - Alto custo - Alta velocidade de leitura - Linha de visão requerida	- Monitoração de veículos em estradas

Fonte: Narciso (2008).

As *tags* de um sistema RFID se dividem em duas categorias: Ativa e Passiva. As *tags* passivas são produzidas a um custo menor pois elas não necessitam de fonte de energia própria, já que utilizam energia eletromagnética emitida pelo leitor e por isso também tem o alcance mais limitado, algo em torno de 10 centímetros. Já as *tags* ativas custam mais pois possuem sua própria fonte de energia interna o que permite alcances maiores, em torno de 10 a 30 metros, e podem transmitir o sinal a qualquer momento sem depender do leitor. (BHUPANI; MARADPOUR, 2005, p.51). A Figura 3 mostra um exemplo de etiqueta RFID.

Figura 3 - Etiqueta eletrônica (*tag*)

Fonte: DXD Rotulos (2011).

MIFARE é uma marca de *tags* RFID de alta frequência da NXP bastante conhecida no mercado, com uma distância de operação de aproximadamente dez centímetros, ela é usada em mais de 40 aplicações diferentes ao redor do mundo. Os produtos MIFARE cumprem o padrão internacional ISO/IEC 14443, que são usados atualmente por mais de 80% dos cartões inteligentes sem fio. Suas características e base sólida e confiável de produtos permitem os provedores de serviço expandir suas ofertas, reduzindo custos e aumentando a conveniência do usuário. (NXP, 2016). A Figura 4 mostra um exemplo de cartão e chaveiros que contém uma *tag* MIFARE em seu interior.

Figura 4 - Cartão e chaveiros contendo tags MIFARE



Fonte: Elaborado pelo autor.

2.3 ARDUINO

Arduino nasceu como uma ferramenta de rápida prototipagem e fácil uso para iniciantes. Em sua versão mais simples, chamada de Arduino UNO (Figura 5), é equipada com um microcontrolador ATmega328P, memória flash de 32 KB, SRAM de 2 KB, EEPROM de 1 KB, 14 pinos de entrada/saída digital e 6 pinos de entrada analógica. A memória flash é usada para armazenar o programa, a SRAM é usada para manipulação de variáveis, a EEPROM para armazenar informações a longo prazo e os pinos analógico e digitais usados para integrar com sensores e módulos. (CREATIVE COMMONS, [2016?]).

Figura 5 - Placa Arduino UNO

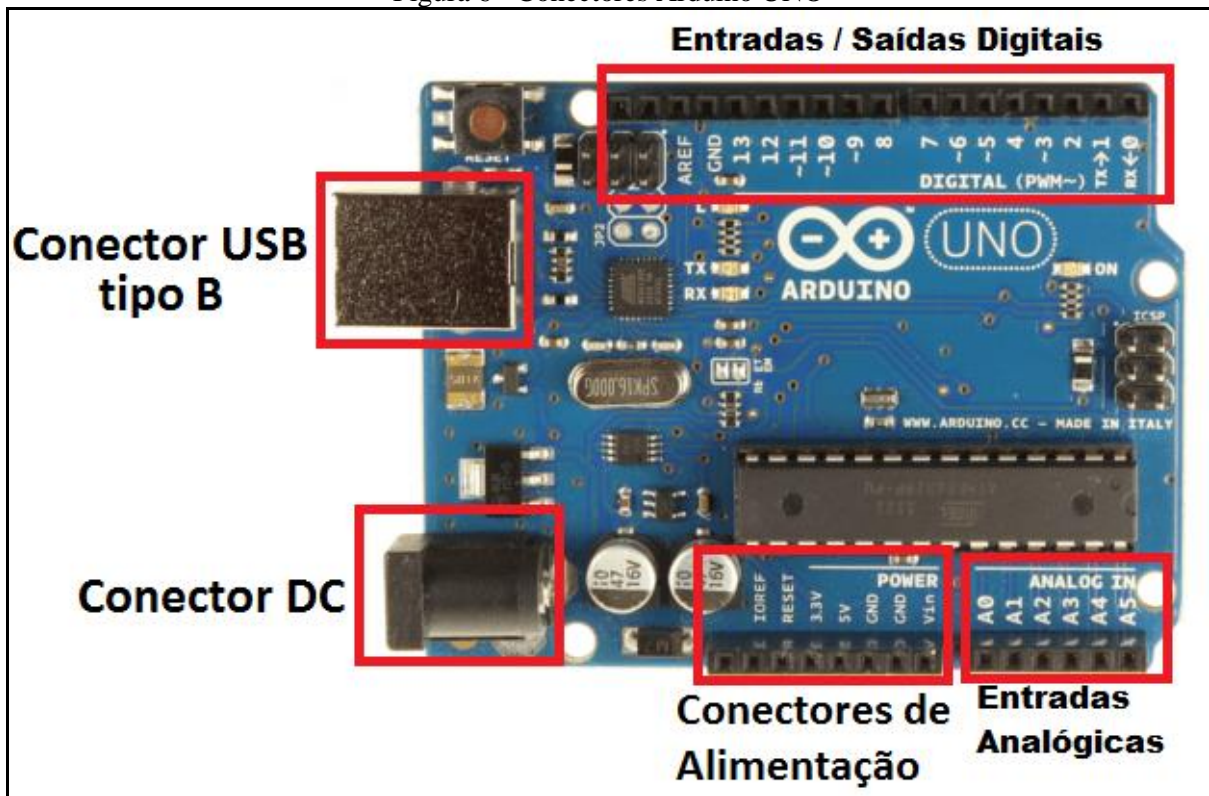


Fonte: Arduino (2016).

As placas Arduino possuem um microcontrolador, que é um pequeno computador dentro de um chip, equipado de processador, memória e pinos de entrada e saída que o ligam aos demais componentes. O microcontrolador é o responsável por interpretar as instruções, via software (geralmente, em C ou Assembly), inseridas em sua memória (PAULI, 2016). Além do microcontrolador o Arduino pode ser complementado com placas acessórias, denominadas *shields*. (MONK, 2013).

A placa Arduino UNO pode ser alimentada por fonte de energia recomendada de 7V a 12V pelo conector DC ou via conector USB tipo B, possui quatorze portas digitais de entrada ou saída, seis portas de entrada analógica e ainda conectores de alimentação de energia externa para utilização de suas shields (SOUZA, 2013). A Figura 6 mostra os as portas descritas.

Figura 6 - Conectores Arduino UNO



Fonte: Elaborado pelo autor.

As placas Arduino podem ser conectadas por fio a outras placas promovendo fácil conexão elétrica fornecendo recursos extras, como Ethernet, Wi-Fi, GPS e controle motor. Essas placas acessórias são conhecidas na comunidade Arduino como *shields* (CRAFT, 2013). *Shields* são basicamente uma extensão da placa que podem ser colocadas sobre a placa Arduino (SCHWARTZ, 2016). A Figura 7 mostra uma placa Ethernet que pode ser utilizada com o Arduino.

Figura 7 - Shield Ethernet



Fonte: Schwartz (2016).

O módulo RC522 (Figura 8) é um leitor de radiofrequência que trabalha na frequência de 13.56 MHz e suporta o padrão ISO/IEC 14443 das *tags* MIFARE padrão comumente encontrado no mercado. Esse módulo pode ser encontrado a um custo em torno de R\$ 25,00 reais, e é compatível com o Arduino, podendo ser usado para ler as *tags* RFID e enviar os dados ao programa armazenado no Arduino.

Figura 8 - Módulo de radiofrequência RC522



Fonte: ICStation (2016?).

2.4 TRABALHOS CORRELATOS

Esta seção tem por finalidade apresentar três trabalhos relacionados, seus objetivos, fundamentos e aplicações.

Dentre os trabalhos selecionados estão: Protótipo de um sistema de identificação eletrônica de animais através de rádio frequência (SANTOS, 2001), o sistema de informação aplicado à cadeia de suprimentos utilizando tecnologia RFID (ZEINDIN, 2005), e o sistema de controle de patrimônio via RFID (DRESCH; EFROM; GRUMOVSKI, 2008). A descrição e características de cada trabalho estão relacionados nas subseções abaixo.

2.4.1. Protótipo de um sistema de identificação eletrônica de animais através de rádio frequência.

O trabalho de Santos (2001) tem por objetivo aplicar a tecnologia RFID para aquisição de dados em atividades produtivas na área de produção animal, visando o auxílio no gerenciamento de informações. Este trabalho foi motivado pela observação de deficiências que ocorrem no processo de produção de animais. Para solucionar o problema foi desenvolvido um sistema de informações em Delphi 5 para manter os dados referentes a criação como cadastramento dos animais, criadores, contagem, pesagem e sanidade mantendo assim um histórico do animal desde o nascimento até o abate.

O sistema se integra com um leitor de RFID (o SCL05 da Olimex) que utiliza a interface RS-232 para leitura do *transponder* implantado no animal com a finalidade de criar um identificador único para cada animal. Esse leitor trabalha na frequência de 128 kHz e 9600 bps. Com essas características o alcance de leitura entre o leitor e o transponder é de aproximadamente 50mm.

O *transponder* utilizado no trabalho é também desenvolvido pela empresa Olimex sendo do tipo passivo (sem bateria), o qual se energiza eletromagneticamente a partir do sinal de rádio frequência emitido pelo leitor e transmite os dados contidos em seu microchip. Com o transponder implantado no animal é possível vincular o identificador de cada animal ao cadastro no sistema.

2.4.2. Sistema de informação aplicado à cadeia de suprimentos utilizando tecnologia RFID.

O trabalho de Zeindin (2005) trata de um sistema de informação que utiliza a tecnologia RFID juntamente com o padrão EPC (Electronic Product Code), integrada a um ERP, para controlar a quantidade de produtos em estoque em toda cadeia produtiva. A motivação do trabalho foi a observação de tecnologias promissoras e recentes no mercado,

objetivando aplicá-las a um sistema de informação para controle e auxílio de processos administrativos.

O trabalho foi realizado utilizando o leitor Z3070, desenvolvido pela empresa ZEBEK de Taiwan, que opera na frequência de 13.56 MHz a uma taxa de transmissão de 24 Kbps e alcance de 100 mm. Esse leitor tem a função de identificar a entrada e saída de produtos do estoque e enviar os dados ao sistema para atualização.

2.4.3. Sistema de controle de patrimônio via RFID.

O artigo de Dresch, Efrom e Grumovski (2008) apresenta o desenvolvimento de um sistema eletrônico para o controle do patrimônio. O foco do trabalho foi controlar equipamentos e mobiliário, que estão em um determinado ambiente pré-estabelecidos, através de *tags* RFID. A justificativa do trabalho foi de que o levantamento de patrimônio possui uma dinâmica difícil quando se utilizam os sistemas como numeração para leitura visual ou códigos de barras.

O trabalho desenvolveu dois protótipos com a finalidade de identificar bens de valores distintos. O protótipo 1 foi desenvolvido para identificar itens de baixo valor, utilizando para isso o leitor ACUPROX AP-09 e etiquetas World Tags, que operam na frequência de 125 kHz. O protótipo 2 foi desenvolvido com o objetivo de monitoramento em tempo real dos itens de alto valor, para isso foi utilizado o leitor AcuWave Active Reader L-RX201 associado às etiquetas AcuWave Tag Asset L-TG800, que compõem um sistema ativo de RFID e operam em alta frequência.

2.4.4. Características dos trabalhos correlatos.

O Quadro 2 mostra as características dos três trabalhos correlatos comparando basicamente ambiente de desenvolvimento, aplicação e dados de operação.

Quadro 2 - Características dos trabalhos correlatos

Trabalho	Santos (2001)	Dresch, Efrom e Grumovski (2008)	Zeindin (2005)
Características			
Ambiente de desenvolvimento	Delphi 5	Não informado	Visual Studio 2003
Linguagem	Delphi	Não informado	C#.Net
Aplicação	Rastreamento Animal	Controle de patrimônio	Cadeia de Suprimentos
Frequência de Operação	128 khz	Protótipo 1 125 khz e protótipo 2 UHF não especificada.	13.56 Mhz
Distância de Operação	50 mm	50 m	100 mm

Fonte: Elaborado pelo autor.

Apesar de as aplicações dos trabalhos correlatos serem muito parecidas, as frequências de operação são bem distintas. Isso se deve ao fato de a frequência de operação escolhida depender dos requisitos de ambiente onde será utilizado o sistema, distância de operação necessária e velocidade de leitura. Já que o custo dos componentes varia bastante em cada faixa de frequência, procura-se atender os requisitos mínimos na escolha da frequência da operação.

Outra distinção entre os trabalhos é o ambiente de desenvolvimento e linguagem. Como visto no quadro, um sistema de identificação RFID pode ser desenvolvido independente do ambiente de desenvolvimento escolhido, já que a comunicação entre leitor e software é normalmente feita através de porta serial ou protocolo HTTP, que são interfaces bem conhecidas nos ambientes de desenvolvimento atualmente.

3 DESENVOLVIMENTO

Este capítulo trata das questões consideradas para o desenvolvimento da solução proposta como requisitos e diagramas. Descreve como foi dividida a aplicação para que a comunicação entre leitor e software fosse realizada e também como foi estruturado os dados para permitir a identificação da localização de um bem.

Para possibilitar o acesso e manutenção das informações em qualquer local, as funções principais do sistema foram desenvolvidas em plataforma Web e esse módulo foi chamado de Patrimônio Web. Pela necessidade de comunicação serial para sincronizar os dados do leitor RFID com o sistema, foi desenvolvido um outro sistema em plataforma desktop, já que o navegador não disponibiliza o acesso a periféricos por questões de segurança. Este módulo foi chamado de Patrimônio Desktop. A Figura 9 mostra os módulos do sistema desenvolvido.

Figura 9 - Módulos do sistema



Fonte: Elaborado pelo autor.

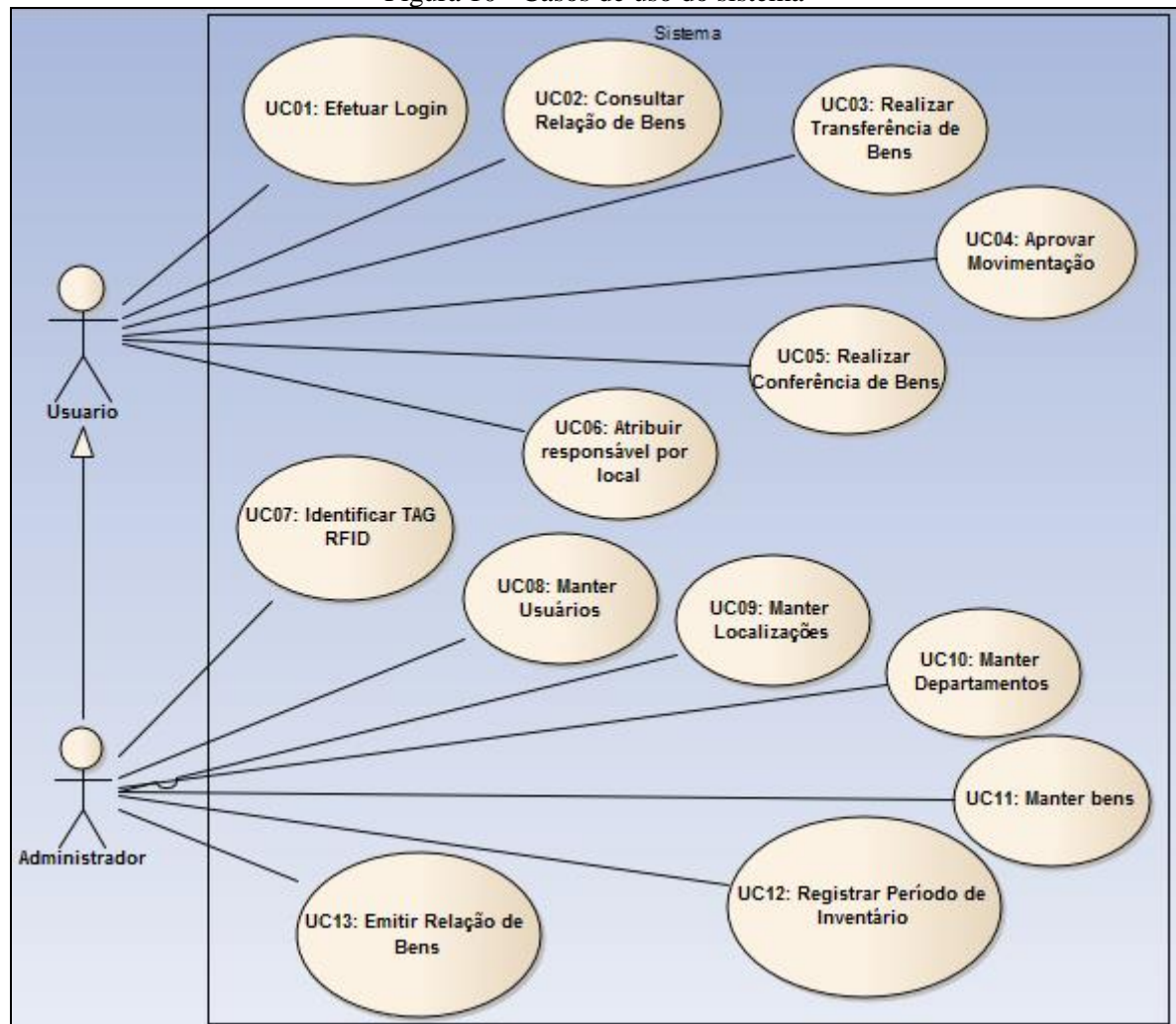
3.1 MODELAGEM

Esta seção apresenta o diagrama necessário para o entendimento do sistema proposto.

3.3.1 Diagrama de casos de uso

Esta seção apresenta o diagrama de casos de uso preliminar do sistema proposto, sendo que os detalhamentos dos casos de uso encontram-se no Apêndice A. A Figura 10 mostra os casos de uso do sistema proposto.

Figura 10 - Casos de uso do sistema



Fonte: Elaborado pelo autor.

3.2 ESPECIFICAÇÃO DOS REQUISITOS

O Quadro 3 apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Quadro 3 - Requisitos funcionais

Requisitos Funcionais	Caso de Uso
RF01: O sistema deverá permitir ao usuário se autenticar.	UC01
RF02: O sistema deverá exibir ao usuário uma lista de bens sob sua responsabilidade.	UC02
RF03: O sistema deverá permitir que o usuário filtre os bens por departamento e localização.	UC02
RF04: O sistema deverá permitir que o usuário transfira seus bens a outro local ou departamento.	UC03
RF05: O sistema deverá registrar um pedido de aprovação da transferência caso o usuário responsável pelo destino da transferência não seja o mesmo.	UC03
RF06: O sistema deve permitir o usuário consultar as transferências aguardando sua aprovação	UC04
RF07: O sistema deve permitir o usuário realizar a conferência dos bens em período de inventário	UC05
RF08: O sistema deve permitir o usuário sincronizar os dados do leitor de <i>tags</i> com o seu inventário	UC05
RF09: O sistema deve notificar o usuário quando uma <i>tag</i> identificada não estiver registrada em seu inventário solicitando confirmação para a transferência	UC05
RF10: O sistema deve permitir a leitura das <i>tags</i> dos bens por meio do leitor e armazenar a lista das <i>tags</i> lidas para posterior sincronização com o sistema.	UC07
RF11: O sistema deve permitir ao administrador manter o cadastro de usuários.	UC08
RF12 O sistema deve permitir ao administrador manter o cadastro de localizações.	UC09
RF13: O sistema deve permitir ao administrador manter o cadastro de departamentos.	UC10
RF14: O sistema deve permitir ao administrador manutenção do cadastro de bens.	UC11
RF15: O sistema deve permitir ao administrador identificar um bem através de uma <i>tag</i> RFID.	UC11
RF16: O sistema deve permitir ao administrador cadastrar um período de inventário.	UC12
RF17: O sistema deve permitir ao administrador consultar e emitir uma relação de todos os bens da organização agrupados por departamento.	UC13

Fonte: Elaborado pelo autor.

O Quadro 4 lista os requisitos não funcionais previstos para o sistema.

Quadro 4 - Requisitos não funcionais

Requisitos Não Funcionais
RNF01: O sistema deverá ser desenvolvido em com a linguagem C# e framework .Net.
RNF02: O sistema deverá utilizar o banco de dados SQL Server 2014 Express.
RNF03: O sistema deverá ser desenvolvido em ambiente web.
RNF04: O sistema deverá ser compatível com os navegadores Google Chrome (versão 49 ou superior), Mozilla Firefox (versão 45 ou superior) e Microsoft Edge (versão 25 ou superior).
RNF05: O leitor de tags RFID deve ser desenvolvido com placa Arduino UNO e modulo de radiofrequência RC522.

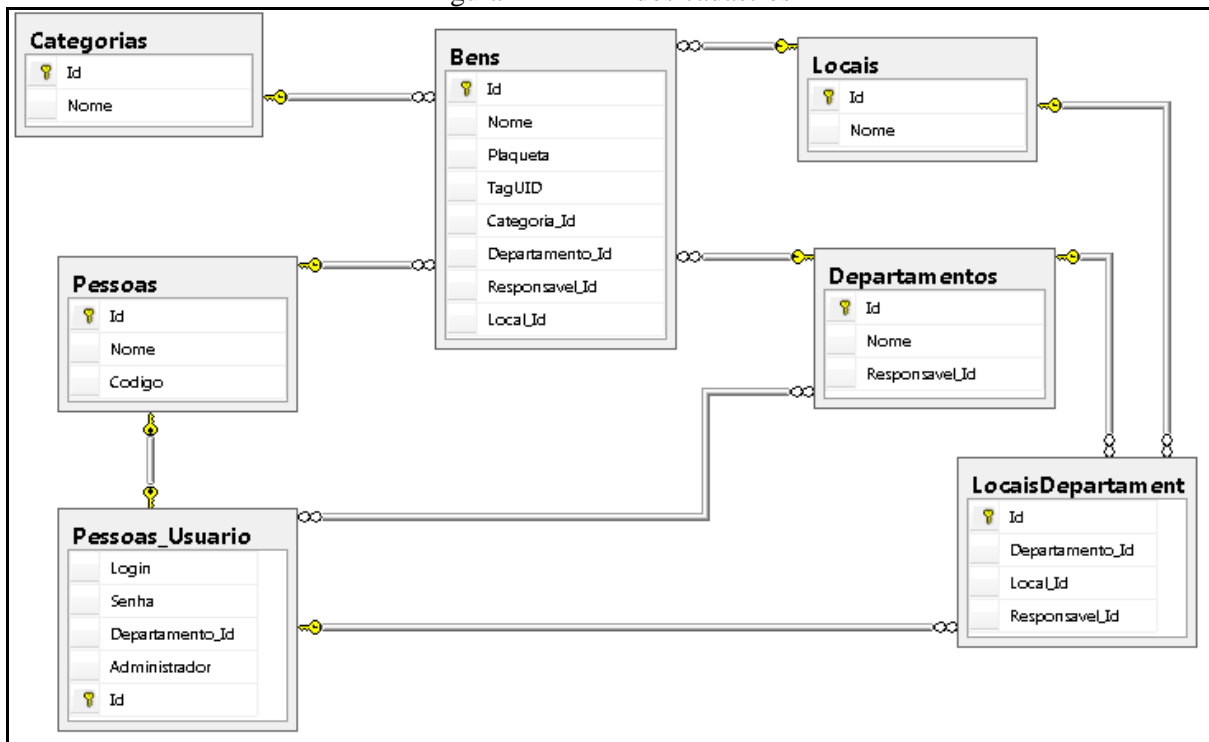
Fonte: Elaborado pelo autor.

3.3 MODELO ENTIDADE RELACIONAMENTO

Para facilitar o entendimento das entidades criadas, os diagramas foram divididos por assunto, agrupados em três assuntos principais referentes a ações realizadas no sistema: cadastros básicos, transferências e inventário.

A Figura 11 apresenta o MER (Modelo Entidade Relacionamento) dos cadastros básicos do sistema desenvolvido.

Figura 11 - MER dos cadastros



Fonte: Elaborado pelo autor.

A tabela `Bens` é a entidade principal do sistema, todas as movimentações são relacionadas a ela. É nessa entidade que será contida a identificação da `tag` RFID, além da descrição do bem e número da plaqueta de identificação. Além dos atributos mencionados, o bem está relacionado a categoria, departamento, local e responsável. A categoria do bem é um agrupador que possui apenas uma descrição, que pode ser mesa, cadeira, computador, etc.

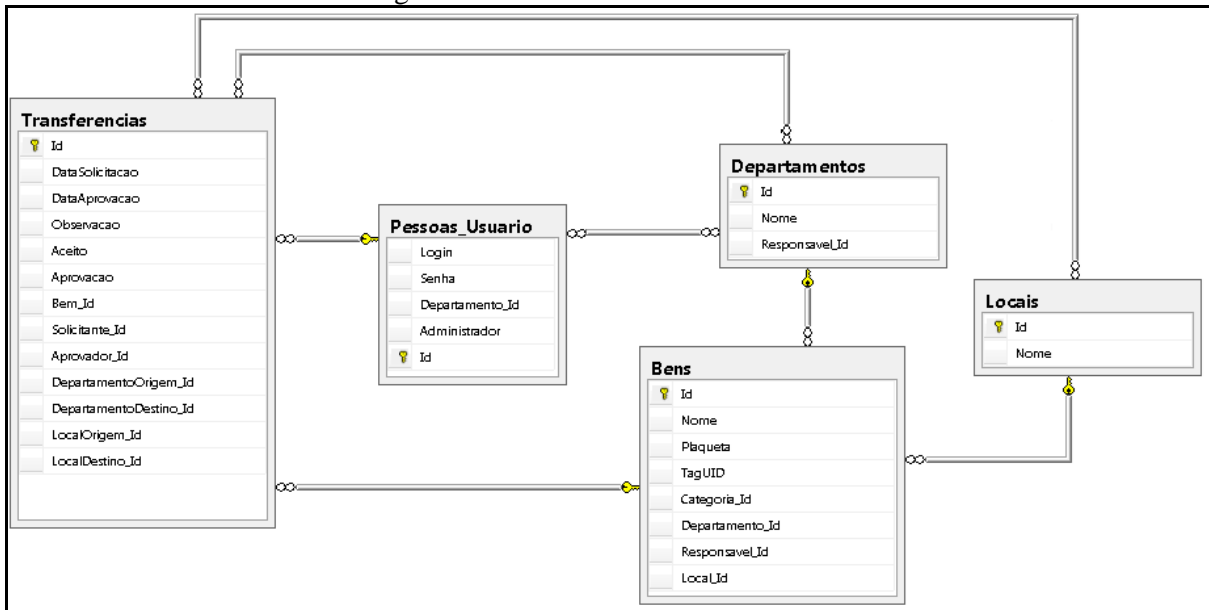
O responsável pelo bem está registrado na entidade `Pessoas`. Essa pessoa não necessariamente tem acesso ao sistema, apenas está cadastrada para registrar sob posse de quem está o bem. A entidade `Usuarios` sim registra as pessoas que possuem acesso ao sistema, e esse usuário também está relacionado a um departamento onde trabalha.

Para registrar a localização de um bem, existem as entidades `Departamentos` e `Locais`. Um departamento pode ser um setor, centro de custo ou organograma da empresa, não é uma localização física e sim conceitual, e está relacionado a um usuário que é o responsável pelo inventário do departamento. Um local representa a localização física do

bem, como um prédio, andar ou sala, ele pode possuir um responsável pelo inventário designado por departamento, para isso existe a entidade `LocaisDepartamento`, que relaciona departamento e local a um usuário. Dessa forma é possível delegar a função de inventariar para um departamento que possui vários locais.

A Figura 12 apresenta o MER das entidades envolvidas nas movimentações de transferência de bens.

Figura 12 - MER entidades transferência



Fonte: Elaborado pelo autor.

A entidade `Transferecias` registra as solicitações de transferência de um bem para outro departamento e/ou local. A entidade relaciona o usuário solicitante e bem a ser transferido, além do departamento e local de origem e destino da transferência. O atributo `Aprovacao` indica quem deve realizar a aprovação da transferência, na origem do bem ou no destino, isso porque a solicitação pode partir de qualquer dos lados e a contrapartida deve realizar o aceite da transferência. Quando a contrapartida realizar o aceite `DataAprovacao` é gravada juntamente com o indicador `Aceito`, que pode ser sim ou não. No caso de uma transferência não aceita uma observação pode ser registrada com o motivo da negação.

A entidade `Inventarios` registra um período de inventário com data de início, data de fim e usuário que registrou. Os inventários realizados pelos usuários são registrados na entidade `InventariosRealizados`, onde relaciona o usuário que registrou, a data de realização, o inventário vigente ao qual está relacionado, departamento e local do inventário. Os bens inventariados são registrados na entidade `BensInventariados`, que relaciona um bem a um inventário realizado.

3.4 MÓDULO PATRIMÔNIO WEB

O módulo web disponibiliza ao usuário administrador as funcionalidades para manutenção de cadastros básicos para o funcionamento do sistema como:

- a) cadastro de usuários: os usuários cadastrados tradicionalmente serão os gestores ou responsáveis por um departamento que utilizarão o sistema para registrar a movimentação de seus bens e fazer a conferência em período de inventário;
- b) cadastro de localizações: essas localizações serão atribuídas aos bens cadastrados para manter o controle de sua localização física;
- c) cadastro de departamentos: departamentos, centros de custo ou unidades de negócio da empresa;
- d) cadastro de bens: esses devem ser atribuídos a um departamento, localização e identificador da *tag* RFID;
- e) registro de período de inventário: essa ação notifica os responsáveis sobre o período em que devem realizar a conferência de seus bens para atualização do inventário;
- f) emitir relação de bens: nessa relação constam todos os bens patrimoniais da organização agrupados por departamento e localização.

Com essas funções o administrador será capaz de configurar o ambiente e manter as configurações para que os responsáveis deem continuidade a manutenção das informações dos bens de seus departamentos.

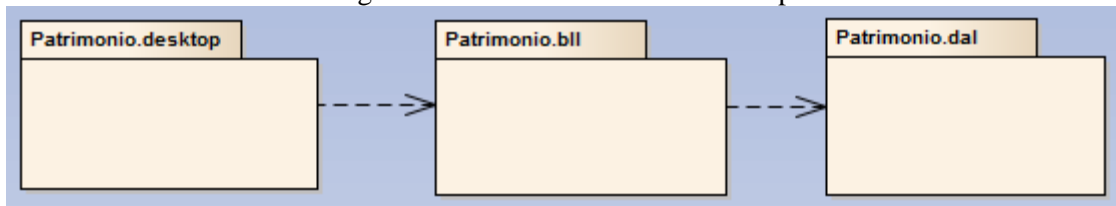
Aos usuários responsáveis por departamento ou local serão disponibilizadas as funções:

- a) consultar relação de bens sob sua responsabilidade: interface com a opção de filtro por departamento e localização;
- b) movimentação de bens para outro departamento ou localização: essa ação pode ficar condicionada a aprovação do responsável pelo destino caso não seja o mesmo usuário;
- c) aprovação de movimentação de bens: quando um responsável por outro departamento registra uma movimentação para o seu departamento;
- d) realizar a conferência dos bens em período de inventário: essa conferência é feita com um leitor de *tags* RFID e posteriormente sincronizada com o sistema.

3.5 MÓDULO PATRIMÔNIO DESKTOP

O módulo desktop é composto por três pacotes (Figura 13): `Patrimonio.desktop`, `Patrimonio.bll` e `Patrimonio.dal`. Sua função é permitir a interação entre Leitor RFID e Sistema, enviando os dados recebidos do leitor para a camada de negócio realizando a atualização dos dados. O Patrimônio Desktop tem duas funções: Identificação de bens e realização de inventário.

Figura 13 - Pacotes do módulo desktop



Fonte: Elaborado pelo autor.

3.5.1. Identificação de Bens

Para identificar um bem, o usuário administrador acessa o módulo Patrimônio Desktop, seleciona um departamento e local, escolhe um bem a ser identificado e aciona a ação `Identificar` destacada na Figura 14.

Figura 14 - Identificação do Bem

	Nome	Plaqueta	TagUID	Identificar
▶	Cadeira	CAD-1105		
	Cadeira	CAD-3038		
	Cadeira	CAD-5527		
	Mesa	MES-5558		
	Mesa	MES-9893		
	Mesa	MES-0624		

Fonte: Elaborado pelo autor.

O módulo irá solicitar a aproximação da *tag* ao leitor para obter a identificação, conforme demonstrado na Figura 15.

Figura 15 - Identificação de *tag*

Fonte: Elaborado pelo autor.

O Quadro 5 representa a comunicação entre o módulo `Patrimonio.desktop` e `Leitor RFID` para localizar a porta conectada no momento de realizar a leitura de uma *tag* para atualização de um bem patrimonial pelo método `LocalizarLeitor`.

O Quadro 5 apresenta o código responsável pelas funções descritas anteriormente, quais sejam: `LocalizarLeitor`, `serialPort1_DataReceived` e `InterpretaMensagem`.

Quadro 5 - Comunicação Patrimonio.desktop com Leitor RFID

```

private void LocalizarLeitor()
{
    encontrouPorta = false;

    //adiciona todas as COM disponíveis na lista
    foreach (string s in SerialPort.GetPortNames())
    {
        if (serialPort1.IsOpen == true) // se porta aberta
            serialPort1.Close();

        serialPort1.PortName = s;
        serialPort1.Open();

        serialPort1.Write("HelloArduino");

        System.Threading.Thread.Sleep(3000);

        if (encontrouPorta)
        {
            portaArduino = s;

            continue;
        }
    }

    lblPorta.Text = portaArduino;
}

private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    var data = serialPort1.ReadExisting();

    if (data.Contains('#'))
        incoming = !incoming;

    if (data.Where(c => c.Equals('#')).Count() > 1)
        incoming = !incoming;

    message += data;

    if (!incoming)
        InterpretaMensagem();
}

private void InterpretaMensagem()
{
    message = message.Replace('#', ' ').Trim();

    if (message.Equals("HelloComputer"))
    {
        encontrouPorta = true;
    }
    else
    if (message.Contains("TAGUID:"))
    {
        bem.TagUID = message.Split(':')[1];

        bll.Repositorio<dal.Bem>.Commit();
        serialPort1.Write("LimparLista");

        this.Invoke(new EventHandler(delegate { this.Hide(); }));
    }

    message = "";
}
}

```

Fonte: Elaborado pelo autor.

Ao entrar no sistema o método `LocalizarLeitor` é chamado, que lista as portas seriais disponíveis no computador e envia uma mensagem “HelloArduino” uma a uma aguardando três segundos antes de seguir para a próxima porta. O evento `serialPort1_DataReceived` é executado ao receber uma mensagem da porta serial ao qual o componente está conectado, então ele trata a mensagem para receber a mensagem que está entre os caracteres “#” e depois chama o método `InterpretaMensagem`. Se a mensagem recebida for “HelloComputer”, então o Arduino foi localizado e a variável `encontrouPorta` é setada para `true`. Se o componente não receber a mensagem de volta, então depois de três segundos segue para tentar localizar o Arduino na próxima porta disponível, até que as opções se esgotem.

3.5.2. Realização de inventário

Em período de inventário, após coletar as identificações das *tags* RFID dos bens patrimoniais, o usuário responsável deve acessar o módulo Patrimônio Desktop para realizar a sincronização dos dados do Leitor RFID. Ao entrar no módulo a porta a qual o Arduino está conectado é localizada assim como na Identificação de Bens. Ao encontrar a porta a mensagem “Download” é enviada solicitando a lista de identificações armazenadas no leitor (Quadro 6). Ao receber a lista de identificações o método `IdentificaTags` é chamado, este localiza os bens pela identificação para registrar um inventário realizado (Quadro 7).

Quadro 6 - Solicitando a lista de identificações ao leitor

```
private void InterpretaMensagem()
{
    message = message.Replace('#', ' ').Trim();

    if (message.Equals("HelloComputer"))
    {
        encontrouPorta = true;
        serialPort1.Write("Download");
    }
    else
    {
        var listaTags = message.Split(';');

        IdentificaTags(listaTags);
    }

    message = "";
}
```

Fonte: Elaborado pelo autor.

Quadro 7 - Identificando bens pela tag RFID

```
private void IdentificaTags(string[] listaTags)
{
    // Verifica se recebeu dados na lista
    if (listaTags[0] == "")
        return;

    listaLeitura.Clear();

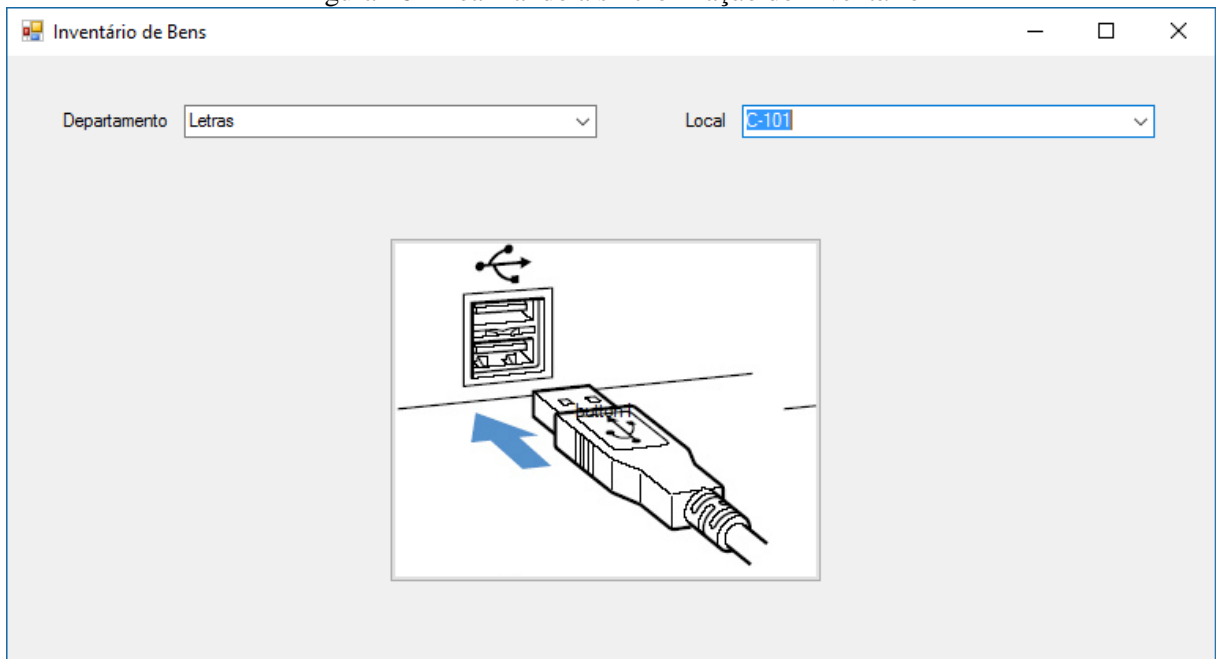
    foreach (var tag in listaTags)
    {
        if (tag.Trim() != "")
        {
            var bem = bll.Bem.Pesquisar(tag);

            listaLeitura.Add(bem);
        }
    }
}
```

Fonte: Elaborado pelo autor.

Nessa etapa ele deve selecionar o departamento e local desejado, e selecionar a opção *sincronizar* para registrar o inventário com os bens identificados conforme exibido no Figura 16. Após registrar o inventário o sistema envia a mensagem “LimparLista” ao leitor para limpar as identificações armazenadas em memória EEPROM, ficando assim disponível para realizar o inventário de outro local.

Figura 16 - Realizando a sincronização do inventário



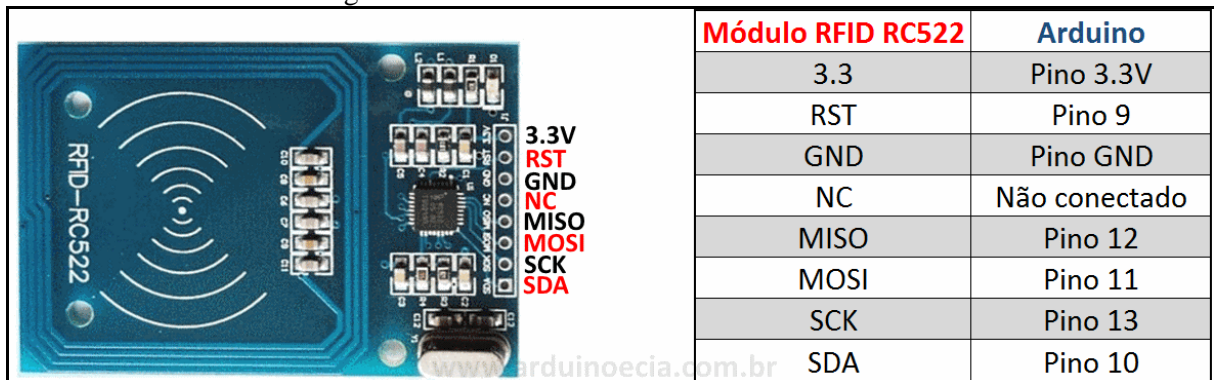
Fonte: Elaborado pelo autor.

3.6 MÓDULO LEITOR RFID

O leitor RFID foi desenvolvido utilizando a placa Arduino Uno e módulo de RFID RC522 conforme descrito na fundamentação teórica. A função desse leitor é captar os dados das *tags* na zona de interrogação (ZI) e enviar para porta serial onde terá o sistema Patrimônio Desktop monitorando para alimentar a informação recebida no banco de dados.

A Figura 17 mostra o circuito para conexão do Arduino Uno ao módulo RC522.

Figura 17 - Circuito elétrico Arduino com RC522



Fonte: Arduino e Cia (2014)

Foi desenvolvido para o arduino um programa que fica constantemente monitorando os dados do módulo RC522, esse programa utiliza a biblioteca `MFRC522` criada por Miguel Balboa para interpretar os dados recebidos do módulo RC522 que utiliza uma interface SPI. Quando uma *tag* entra na zona de interrogação do módulo o sistema captura o identificador e realiza duas tarefas: Armazena a identificação na memória EEPROM do arduino e envia a identificação para porta serial. O código é mostrado na seção de arquitetura do leitor RFID, item 3.7.4.

A identificação da *tag* Mifare utilizada é composta por quatro bytes. Antes de enviar a identificação para porta serial ela é convertida para o formato hexadecimal, padronizando e facilitando a identificação visual em um formato de 8 dígitos. O Quadro 8 mostra a conversão do código da identificação.

Quadro 8 - Método de conversão da identificação para hexadecimal.

```
void printHex(int addr) {
  TagUID tag;
  EEPROM.get(addr, tag);

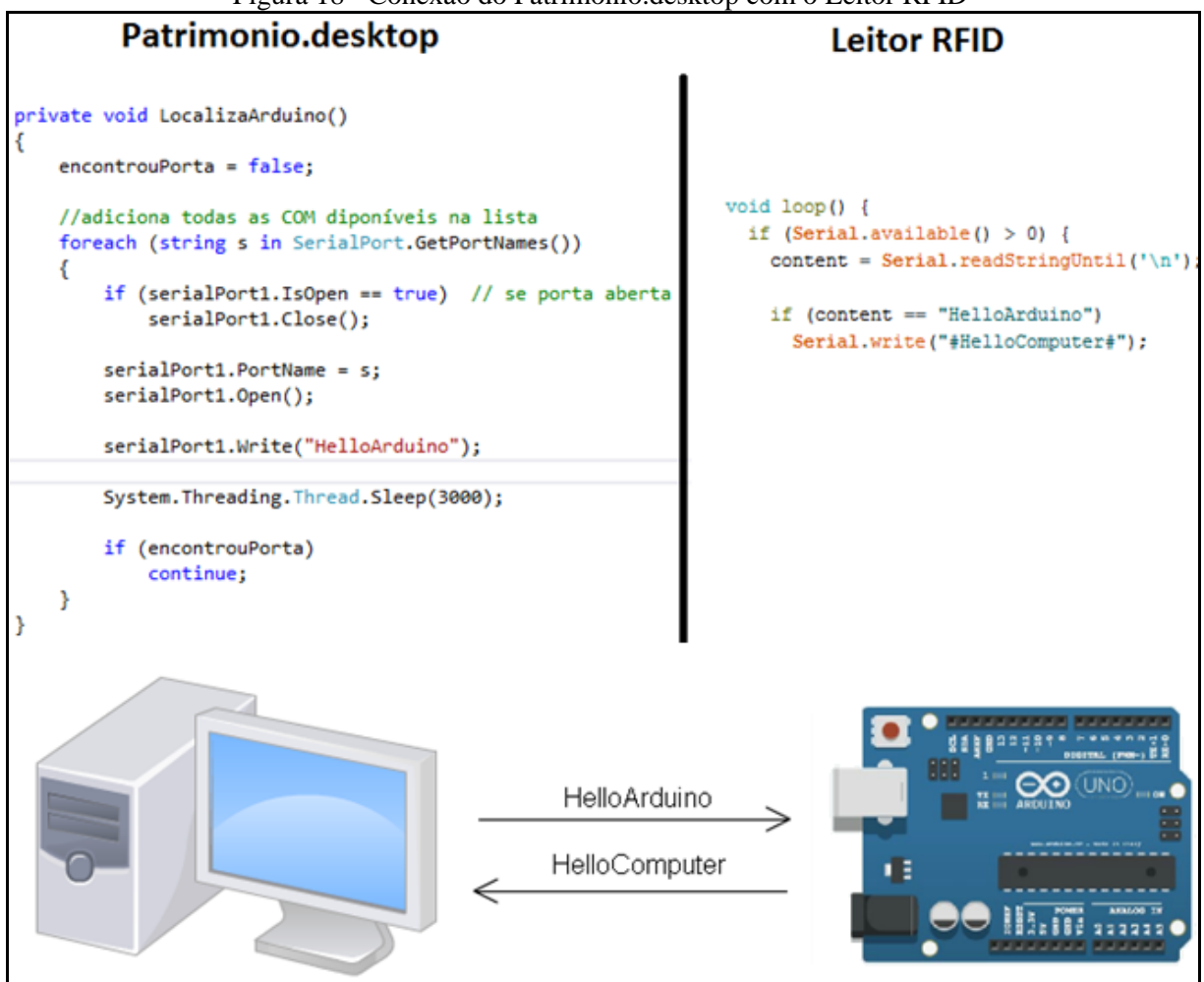
  for (byte i = 0; i < 4; i++) {
    Serial.print(tag.valor[i] < 0x10 ? "0" : "");
    Serial.print(tag.valor[i], HEX);
  }
}
```

Fonte: Elaborado pelo autor.

3.6.1. Protocolo de comunicação

Para sincronizar o sistema Patrimônio Desktop com o arduino ocorre uma troca de mensagens, caracterizando o *HandShake*. O sistema desktop lista as portas seriais disponíveis no computador e envia a mensagem “HelloArduino” para cada uma aguardando a resposta “HelloComputer” por três segundos, caso não receba o retorno então envia a mensagem para a próxima porta até se esgotarem as alternativas (Figura 18), então caso não tiver localizado o arduino conectado as portas o sistema exibe uma mensagem solicitando conectar o arduino ao computador para continuar.

Figura 18 - Conexão do Patrimonio.desktop com o Leitor RFID

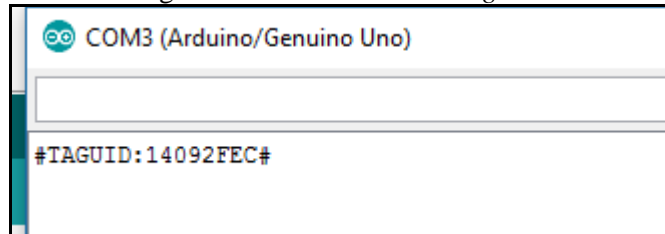


Fonte: Elaborado pelo autor.

Para a correta leitura da informação no destino, as mensagens são enviadas com o caractere “#” no início e final, identificando onde começa e termina cada mensagem. Isso é necessário pois devido a frequência de operação do arduino e computador que podem não estar em sincronia, ocasionando de a mensagem chegar ao destino fracionada em mais de um pacote.

A Figura 19 mostra o monitor serial da porta onde o arduino está conectado ao computador no momento em que ele lê uma *tag*.

Figura 19 - Monitor Serial: *tag* lida

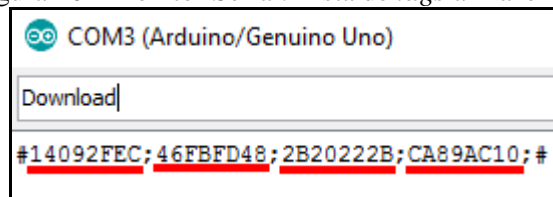


Fonte: Elaborado pelo autor.

A lista com todas as *tags* armazenadas pode ser obtida do arduino enviando o comando serial “Download” para ele. O sistema armazenado no arduino enviará a lista identificando o início e final da mensagem com o caractere “#” e separando as identificações com o caractere “;” para delimitar cada identificação.

A Figura 20 mostra a resposta do sistema ao enviar o comando serial “Download” após ler quatro *tags* diferentes.

Figura 20 - Monitor Serial: Lista de *tags* armazenadas

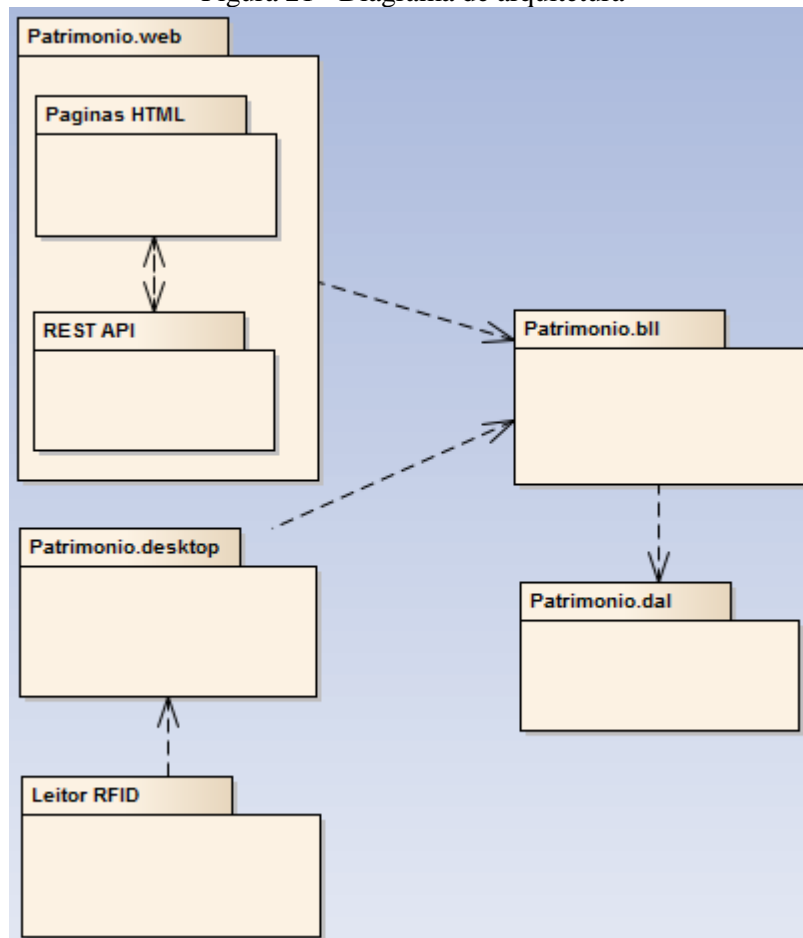


Fonte: Elaborado pelo autor.

3.7 ARQUITETURA DA SOLUÇÃO

A solução foi dividida em módulos para integração dos sistemas, aproveitamento de código e facilitar a manutenção. Na Figura 21 está o diagrama de pacotes contendo a estrutura do projeto.

Figura 21 - Diagrama de arquitetura



Fonte: Elaborado pelo autor.

3.7.1. Pacote `Patrimonio.dal`

Para a persistência de dados com o banco foi utilizado o EntityFramework 6 (EF6), que é um framework nativo da Microsoft e integrado ao Visual Studio. Esse framework já mapeia as classes de objetos do DB (Database) dentro de um container, o qual permite que se trabalhe com os objetos em memória até a execução do comando `SaveChanges`, então o framework trata da persistência de todos os dados em memória para o banco de dados, minimizando assim escrita de código e manutenção do sistema. (DRISCOLL, 2013). O Quadro 9 mostra a classe `PatrimonioContainer` com as entidades mapeadas.

Quadro 9 - Container gerado a partir do EF6

```
namespace Patrimonio.dal
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class PatrimonioContainer : DbContext
    {
        public PatrimonioContainer()
            : base("name=PatrimonioContainer")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Categoria> Categorias { get; set; }
        public virtual DbSet<Bem> Bens { get; set; }
        public virtual DbSet<Pessoa> Pessoas { get; set; }
        public virtual DbSet<Local> Locais { get; set; }
        public virtual DbSet<Departamento> Departamentos { get; set; }
        public virtual DbSet<LocalDepartamento> LocaisDepartamentos { get; set; }
        public virtual DbSet<Transferencia> Transferencias { get; set; }
        public virtual DbSet<Inventario> Inventarios { get; set; }
        public virtual DbSet<InventarioRealizado> InventariosRealizados { get; set; }
        public virtual DbSet<BensInventariados> BensInventariados { get; set; }
    }
}
```

Fonte: Elaborado pelo autor.

Esse container e o mapeamento das classes são gerados no pacote `Patrimonio.dal`. O sufixo `dal` se refere a sigla de Data Access Layer, que representa a camada de acesso aos dados do DB. No Quadro 10 é apresentado um exemplo de mapeamento de classe gerado pelo EF6.

Quadro 10 - Classe Transferencia

```

namespace Patrimonio.dal
{
    using System;
    using System.Collections.Generic;

    public partial class Transferencia
    {
        public int Id { get; set; }
        public System.DateTime DataSolicitacao { get; set; }
        public Nullable<System.DateTime> DataAprovacao { get; set; }
        public string Observacao { get; set; }
        public Nullable<bool> Aceito { get; set; }
        public string Aprovacao { get; set; }

        public virtual Bem Bem { get; set; }
        public virtual Usuario Solicitante { get; set; }
        public virtual Usuario Aprovador { get; set; }
        public virtual Departamento DepartamentoOrigem { get; set; }
        public virtual Departamento DepartamentoDestino { get; set; }
        public virtual Local LocalOrigem { get; set; }
        public virtual Local LocalDestino { get; set; }
    }
}

```

Fonte: Elaborado pelo autor.

3.7.2. Pacote Patrimonio.bll

No módulo Patrimonio.bll estão contidas as classes de negócio da solução, é nela que estão implementadas as validações e regras de negócio. O sufixo bll se refere a Business Logic Layer. O Quadro 11 mostra um exemplo de classe de negócio desse módulo.

Quadro 11 - Classe Transferencia na camada de negócio

```

namespace Patrimonio.bll
{
    public class Transferencia
    {
        public enum Aprovacao
        {
            Origem,
            Destino
        }

        public static void Solicitar(dal.Usuario solicitante, dal.Bem bem, dal.Departamento depDestivo, dal.Local locDestino, Aprovacao aprovacao)
        {
            Solicitar(solicitante, bem, depDestivo, locDestino, aprovacao == Aprovacao.Origem ? "O" : "D");
        }

        public static void Solicitar(dal.Usuario solicitante, dal.Bem bem, dal.Departamento depDestivo, dal.Local locDestino, string aprovacao)
        {
            Repositorio<dal.Transferencia>.Adicionar(new dal.Transferencia()
            {
                Solicitante = solicitante,
                Bem = bem,
                DepartamentoOrigem = bem.Departamento,
                LocalOrigem = bem.Local,
                DepartamentoDestino = depDestivo,
                LocalDestino = locDestino,
                DataSolicitacao = DateTime.Now,
                Aprovacao = aprovacao
            });
            Repositorio<dal.Transferencia>.Commit();
        }
    }
}

```

Fonte: Elaborado pelo autor.

No pacote `Patrimonio.bll` também foram criadas duas classes chaves para padronizar o acesso aos dados: a classe `Sistema` (Quadro 12) e a classe `Repositorio` (Quadro 13).

Quadro 12 - Classe `Sistema`

```
namespace Patrimonio.bll
{
    public class Sistema
    {
        private static PatrimonioContainer _context;
        public static PatrimonioContainer Context
        {
            get
            {
                if (_context == null)
                    _context = new PatrimonioContainer();

                return _context;
            }
            set { _context = value; }
        }

        private static dal.Usuario _UsuarioLogado;
        public static dal.Usuario usuarioLogado
        {
            get
            {
                return _UsuarioLogado;
            }
            set
            {
                _UsuarioLogado = value;
            }
        }
    }
}
```

Fonte: Elaborado pelo autor.

A classe `Sistema` contém uma propriedade estática do container do EF6, que é instanciada na primeira utilização e se mantém em memória para os futuros acessos, garantindo uma única instancia criada em memória, aplicando assim o conceito de *Singleton* do padrão de projeto de software Design Pattern (BISHOP, 2007). Também contém uma instancia estática do usuário logado no sistema para acesso a qualquer momento.

A classe `Repositorio` é uma classe abstrata que aplica o conceito de *Generics* e contém os métodos de acesso aos objetos do container do EF6. O acesso aos objetos é feito a partir da instância do container contida na classe `Sistema`.

Quadro 13 - Classe Repositorio

```

public abstract class Repositorio<T> where T : class
{
    public static IQueryable<T> GetTodos()
    {
        return Sistema.Context.Set<T>();
    }

    public static IQueryable<T> Get(Expression<Func<T, bool>> predicate)
    {
        return Sistema.Context.Set<T>().Where(predicate);
    }

    public static T Find(params object[] key)...

    public static T First(Expression<Func<T, bool>> predicate)...

    public static void Adicionar(T entity)...

    public static void Atualizar(T entity)...

    public static void Deletar(Func<T, bool> predicate)
    {
        Sistema.Context.Set<T>()
        .Where(predicate).ToList()
        .ForEach(del => Sistema.Context.Set<T>().Remove(del));
    }

    public static void Commit()
    {
        Sistema.Context.SaveChanges();
    }
}

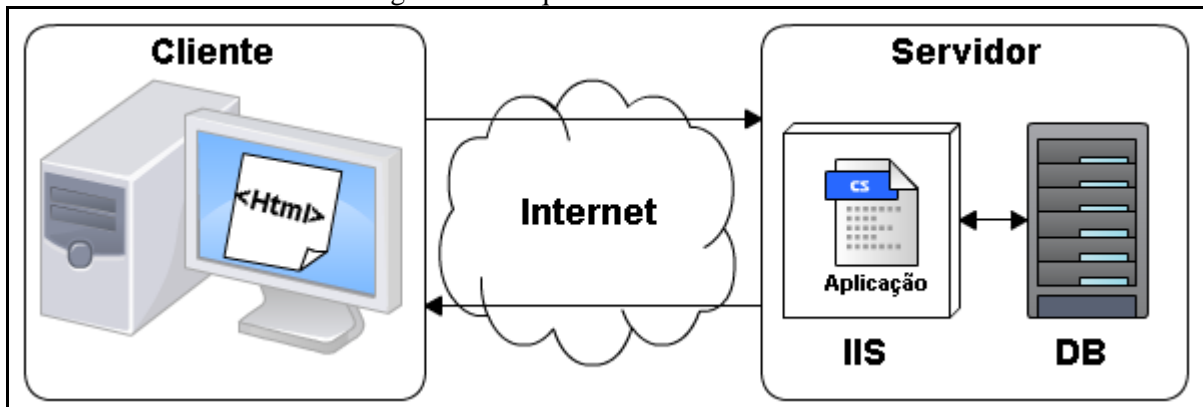
```

Fonte: Elaborado pelo autor.

3.7.3. Pacote Patrimonio.web

No módulo `Patrimonio.web` estão contidos os arquivos do sistema Web para interface cliente e servidor, nesse módulo foi aplicado o padrão MVC (*Model View Controller*) onde as *Views* são as páginas HTML enviadas para o cliente, os *Models* são classes contendo as propriedades mínimas necessárias para cada caso de uso e os *Controllers* são Web APIs que recebem e devolvem mensagens HTTP no formato JSON. A Figura 22 mostra a arquitetura cliente servidor do módulo `Patrimonio.web`.

Figura 22 - Arquitetura cliente - servidor



Fonte: Elaborado pelo autor.

A interface cliente é a que roda no navegador do usuário, ela é formada por páginas HTML e CSS que executam código Javascript. Para seu desenvolvimento foram utilizados os frameworks AngularJS e Bootstrap.

O AngularJS é um framework *client-side* que veio para apoiar na produtividade, flexibilidade e manutenibilidade do desenvolvimento de sistemas Web. Um recurso poderoso dessa ferramenta é o “*two-way binding*” onde diretivas inseridas no código HTML vinculam o componente a um objeto Javascript, então qualquer alteração no valor do componente reflete automaticamente ao objeto e o mesmo ocorre quando alterado valor do objeto via script. (BRANAS, 2014).

O Quadro 14 mostra um exemplo de página HTML desenvolvida utilizando os frameworks mencionados e o Quadro 15 mostra um script *client-side* da mesma página desenvolvido utilizando AngularJS.

Quadro 14 - Página bens.html

```
<table class="table table-striped">
  <tr>
    <th><input type="checkbox" ng-model="allChecked" /></th>
    <th><a href="" ng-click="ordenacao('Categoria')">Categoria</a></th>
    <th><a href="" ng-click="ordenacao('Nome')">Nome</a></th>
    <th><a href="" ng-click="ordenacao('Plaqueta')">Plaqueta</a></th>
    <th><a href="" ng-click="ordenacao('TagUID')">TagUID</a></th>
  </tr>
  <tr ng-repeat="bem in filteredBens | orderBy: criterioOrderBy : direcaoOrderBy">
    <td><input type="checkbox" ng-model="bem.Selecionado"/></td>
    <td>{{bem.categoria.Nome}}</td>
    <td>{{bem.Nome}}</td>
    <td>{{bem.Plaqueta}}</td>
    <td>{{bem.TagUID}}</td>
  </tr>
</table>

<div class="container text-center">
  <button class="btn btn-primary" ng-click="transferir()">Transferir</button>
</div>
```

Fonte: Elaborado pelo autor.

Quadro 15 - Script da página bens.html

```
app.  
.controller('bensCtrl', function ($scope, $rootScope, $location, bensService, departamentoService  
    , localService, pessoaService, categoriaService) {  
    $scope.departamentos = [];  
    $scope.locais = [];  
    $scope.bens = [];  
    $scope.responsaveis = [];  
    $scope.categorias = [];  
  
    $scope.depFilter = '';  
    $scope.locFilter = '';  
    $scope.filteredBens = [];  
  
    // Inicializa tela carregando Bens do usuario logado  
    bensService.getBens()  
        .then(function (response) {  
            $scope.bens = response.data;  
            $scope.filteredBens = $scope.bens;  
  
            // Carrega os departamentos  
            departamentoService.getDepartamentos()  
                .then(function (response) {  
                    $scope.departamentos = response.data;  
  
                    localService.getLocais()  
                        .then(function (response) {  
                            $scope.locais = response.data;  
  
                            // Carrega Categorias  
                            categoriaService.getAll()  
                                .then(function (response) {  
                                    $scope.categorias = response.data;
```

Fonte: Elaborado pelo autor.

No lado servidor da aplicação temos um Serviço *REST* rodando no ISS (*Internet Information Services*). Esse serviço possui as APIs que recebem as requisições HTTP, buscam e tratam a informação requisitada e devolvem em formato JSON. O Quadro 16 abaixo mostra um exemplo de API.

Quadro 16 - API do serviço de transferência

```
namespace Patrimonio.web.Controllers
{
    public class TransferenciaController : ApiController
    {
        [HttpPost]
        public JsonResult<Models.Response> Transferir([FromUri] int[] bens, int departamento, int local, string aprovacao)
        {
            try
            {
                foreach (int idBem in bens)
                {
                    bll.Transferencia.Solicitar(bll.Sistema.usuarioLogado, bll.Bem.Get(idBem), bll.Departamento.Get(departamento));
                }

                return Json(new Models.Response() { Sucesso = true });
            }
            catch (Exception ex)
            {
                return Json(new Models.Response() { Sucesso = false, Mensagem = ex.Message });
            }
        }

        [HttpGet]
        public JsonResult<List<Models.Transferencia>> PendentesAprovador()
        {
            var lista = bll.Transferencia.GetPendentesAprovador(bll.Sistema.usuarioLogado)
                .Select(s => (Models.Transferencia) s).ToList();

            return Json(lista);
        }
    }
}
```

Fonte: Elaborado pelo autor.

Essas APIs recuperam os dados da camada de negócio e transformam o retorno na classe contida no namespace `Models`, que é destinada a apresentação, como demonstrado no método `PendentesAprovador` do Quadro 16. Isso serve para restringir as informações enviadas para a máquina cliente e também reduzir os dados trafegados pela rede ao mínimo necessário. O Quadro 17 mostra a classe `Transferencia` contida nesse pacote que é retornada no método mencionado anteriormente.

Quadro 17 - Classe `Transferencia` no pacote `Models`

```

namespace Patrimonio.web.Models
{
    public class Transferencia
    {
        public int id { get; set; }
        public string solicitante { get; set; }
        public string plaqueta { get; set; }
        public string nome { get; set; }
        public string depOrigem { get; set; }
        public string depDestino { get; set; }
        public string locOrigem { get; set; }
        public string locDestino { get; set; }
        public string aprovacao { get; set; }

        public static explicit operator Transferencia(dal.Transferencia data)
        {
            return new Transferencia()
            {
                id = data.Id,
                solicitante = data.Solicitante.Nome,
                plaqueta = data.Bem.Plaqueta,
                nome = data.Bem.Nome,
                depOrigem = data.DepartamentoOrigem.Nome,
                depDestino = data.DepartamentoDestino.Nome,
                locOrigem = data.LocalOrigem.Nome,
                locDestino = data.LocalDestino.Nome,
                aprovacao = data.Aprovacao
            };
        }
    }
}

```

Fonte: Elaborado pelo autor.

Toda classe contida nesse pacote `Models` possui um método para transformar o objeto de dado retornado da camada de negócio para um objeto da camada de apresentação. Isso permite a realização do *cast* direto.

3.7.4. Módulo Leitor RFID

O módulo Leitor RFID é o programa que foi desenvolvido para o Arduino para captura da identificação das *tags* e enviar para porta serial para que o módulo `Patrimonio.desktop` possa atualizar no banco de dados. O programa fica em constante loop esperando receber mensagens vindas da porta serial ou vindas do módulo RFID como demonstrado no Quadro 18

Quadro 18 - Programa principal do Leitor RFID

```
void loop() {  
  if (Serial.available() > 0) {  
    content = Serial.readStringUntil('\n');  
  
    if (content == "HelloArduino")  
      Serial.write("#HelloComputer#");  
  
    if (content == "Download")  
      DescarregarLista();  
  
    if (content == "LimparLista")  
      LimparLista();  
  
  }  
  
  // Look for new cards  
  if (!rfid.PICC_IsNewCardPresent())  
    return;  
  
  if (!rfid.PICC_ReadCardSerial())  
    return;  
  
  if (NovaTag(rfid.uid.uidByte)) {  
    ArmazenaTag(rfid.uid.uidByte);  
  }  
}
```

Fonte: Elaborado pelo autor.

As *tags* são armazenadas na memória EEPROM do Arduino, portanto o método que verifica se a *tag* lida já foi armazenada, varre a lista de endereços da EEPROM em busca do identificador da *tag*. O Quadro 19 exhibe o código fonte do método `NovaTag` e o Quadro 20 exhibe o código fonte que armazena o identificador da *tag* lida na memória EEPROM.

Quadro 19 - Método NovaTag.

```

bool NovaTag(byte *buffer) {
    bool result = true;
    int eeAddr = 0;

    TagUID tag;

    while (eeAddr < EEPROM_MAX_ADDR) {
        // Se não encontrar valor armazenado no endereço encerra a busca
        if (EEPROM.read(eeAddr) == 0)
            break;

        // Le o valor da EEPROM na variavel
        EEPROM.get(eeAddr, tag);
        eeAddr += sizeof(TagUID);

        // Checa se a tag verificada é diferente da encontrada na EEPROM
        result = tag.valor[0] != buffer[0]
            || tag.valor[1] != buffer[1]
            || tag.valor[2] != buffer[2]
            || tag.valor[3] != buffer[3];

        if (!result)
            break;
    }

    return result;
}

```

Fonte: Elaborado pelo autor.

Quadro 20 - Método ArmazenaTag

```

void ArmazenaTag(byte *buffer) {
    // Armazena a Tag no array
    TagUID tag = { {
        buffer[0],
        buffer[1],
        buffer[2],
        buffer[3]
    } };

    EEPROM.put(eeSize, tag);

    Serial.print(F("#TAGUID:"));
    printHex(eeSize);
    Serial.print(F("#"));
    Serial.println();

    // incrementa endereço da EEPROM para proximo armazenamento
    eeSize += sizeof(TagUID);
}

```

Fonte: Elaborado pelo autor.

3.7.5. Operacionalidade da implementação

O fluxo de trabalho no sistema pode ser dividido em três etapas: configuração de ambiente, movimentação de bens e realização de inventário.

3.7.6. Configuração do ambiente

Nessa etapa realizam-se os cadastros básicos do sistema pelo administrador. Os locais são identificados e cadastrados e os departamentos configurados com os devidos responsáveis.

A Figura 23 mostra o cadastro de departamentos no sistema Patrimônio Web, disponível para o usuário administrador.

















Figura 23 - Cadastro de Departamentos

Patrimônio Web				Aprovação	Responsáveis	Inventário	Administração
Nome	Responsável	Editar	Excluir	Novo Departamento			
Administração	Clara Rodrigues Pereira						
Arquitetura e Urbanismo	Gustavo Santos Silva						
Letras	Rafael Losi						
Química	Mauro Mattos						
Matemática	Mauro Mattos						
Sistemas e Computação	Mauro Mattos						

Fonte: Elaborado pelo autor.

Os usuários responsáveis devem fazer o levantamento físico de seus departamentos e enviar ao administrador para realizar o cadastro (Figura 24). De posse da lista dos bens, o administrador efetua o cadastro vinculando cada bem a uma *tag* RFID identificando o mesmo. Após a realização dos cadastros as *tags* são enviadas junto com a descrição do bem ao responsável para que o mesmo os etiquete.

Figura 24 - Cadastro de Bens

Patrimônio Web								Aprovação	Responsáveis	Inventário	Administração
Cadastro de Bens											
Pesquisar <input type="text" value="informe seu filtro aqui"/>											
Departamento	Local	Categoria	Nome	Plaqueta	TagUID	Editar	Excluir				
Administração	A-101	Quadro	Quadro	QUA-4991							
Administração	A-101	Computador	Computador	CMP-4474							
Administração	A-101	Cadeira	Cadeira	CAD-8427							
Administração	A-101	Mesa	Mesa	MES-5783							
Administração	A-101	Computador	Computador	CMP-8483							
Administração	A-101	Cadeira	Cadeira	CAD-0560							
Administração	A-101	Mesa	Mesa	MES-0431							
Administração	A-101	Computador	Computador	CMP-2296							

Fonte: Elaborado pelo autor.

A identificação é realizada através do módulo Patrimônio Desktop, apresentado na seção 3.5.

3.7.7. Movimentação de bens

Quando um bem é transferido fisicamente para outro local ou departamento, o responsável deve registrar a movimentação no sistema. Essa ação é realizada pelo módulo Patrimônio Web, onde o usuário responsável pode acessar a lista de todos os bens sob sua responsabilidade e realizar a transferência conforme mostra a Figura 25.

Figura 25 - Transferência de bens

The screenshot shows the 'Patrimônio Web' interface. At the top, there are navigation links for 'Aprovação', 'Responsáveis', and 'Inventário'. Below these, there are dropdown menus for 'Departamentos:' (set to 'Letras') and 'Locais:' (set to 'C-101'). The main area contains a table of assets with columns for 'Categoria', 'Nome', 'Plaqueta', and 'TagUID'. The first three rows are selected, indicated by checked checkboxes in the left margin. At the bottom of the table, a blue 'Transferir' button is highlighted with a red circle.

	Categoria	Nome	Plaqueta	TagUID
<input type="checkbox"/>	Quadro	Quadro	QUA-6499	2B20222B
<input checked="" type="checkbox"/>	Computador	Computador	CMP-3168	46FBFD48
<input checked="" type="checkbox"/>	Cadeira	Cadeira	CAD-4504	14092FEC
<input checked="" type="checkbox"/>	Mesa	Mesa	MES-5013	CA89AC10
<input type="checkbox"/>	Computador	Computador	CMP-2485	
<input type="checkbox"/>	Cadeira	Cadeira	CAD-4115	
<input type="checkbox"/>	Mesa	Mesa	MES-6739	
<input type="checkbox"/>	Computador	Computador	CMP-5356	
<input type="checkbox"/>	Cadeira	Cadeira	CAD-1780	
<input type="checkbox"/>	Mesa	Mesa	MES-0470	

Fonte: Elaborado pelo autor.

Em seguida será solicitado o destino da transferência conforme indicado pela Figura 26.

Figura 26 - Destino da transferência

The screenshot shows the 'Patrimônio Web' interface for selecting the destination of a transfer. The title is 'Selecione o destino da Transferencia'. There are two dropdown menus: 'Departamento' and 'Local'. The 'Departamento' dropdown is highlighted with a red circle. Below the dropdowns is a table with columns for 'Departamento', 'Local', 'Plaqueta', and 'Nome'. At the bottom, a blue 'Transferir' button is visible.

Departamento	Local	Plaqueta	Nome
Letras	C-101	CMP-3168	Computador
Letras	C-101	CAD-4504	Cadeira
Letras	C-101	MES-5013	Mesa

Fonte: Elaborado pelo autor.

Caso o destino seja de outro responsável esta movimentação deve ser submetida a uma aprovação. Assim que o responsável destino receber o bem em seu local ou departamento o mesmo realiza a aprovação da movimentação. Só então as atualizações são realizadas no cadastro do bem.

A Figura 27 mostra a ação de aprovação a ser realizada no sistema Patrimônio Web.

Figura 27 - Aprovação de transferência

Solicitante	Plaqueta	Bem	Departamento	Local	Aceite
Rafael Losi	CMP-3168	Computador	Sistemas e Computação	A-101	 
Rafael Losi	CAD-4504	Cadeira	Sistemas e Computação	A-101	 
Rafael Losi	MES-5013	Mesa	Sistemas e Computação	A-101	 

Fonte: Elaborado pelo autor.

3.7.8. Realização de inventário

A realização do inventário deve ser realizada periodicamente como forma de auditoria dos dados para localizar inconsistências e possíveis bens perdidos ou até roubados. Para isso, o administrador deve cadastrar um período de inventário no sistema e os responsáveis serão notificados.

O usuário administrador deve cadastrar o período de inventário para possibilitar o acesso dos usuários responsáveis por departamento ou local ao sistema Patrimônio Desktop como mostra a Figura 28.

Figura 28 - Cadastro de período de inventário

Data Inicio	Data Fim	Usuario Cadastrante	Editar	Excluir
20/10/2016	30/10/2016	Administrador		
01/11/2016	30/11/2016	Administrador		

Fonte: Elaborado pelo autor.

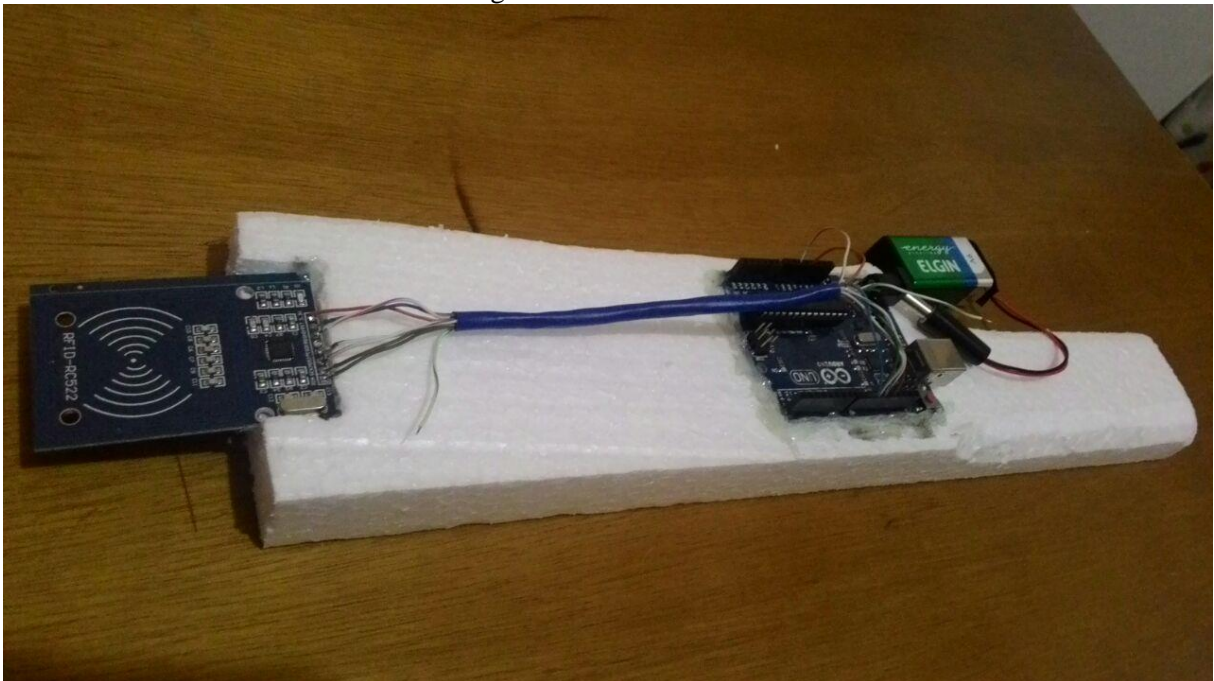
Os responsáveis utilizarão o leitor RFID para identificar todos os bens de um determinado departamento e local. Após a identificação dos bens, o responsável pelo departamento ou local realizará a sincronização dos dados com o sistema. Para isso o usuário responsável faz conexão com o Patrimônio Desktop, detalhado na seção 3.6.

3.7 RESULTADOS E DISCUSSÕES

O trabalho através dos módulos desenvolvidos alcançou os resultados esperados, assim, possibilitando a identificação dos bens patrimoniais pela tecnologia de identificação por rádio frequência e consequentemente auxiliando o processo de inventário patrimonial.

O leitor RFID integrado permitiu o levantamento do inventário de determinado local e departamento com a facilidade de não necessitar conferir a numeração de plaquetas uma a uma, necessitando apenas aproximar o leitor de cada etiqueta RFID e posteriormente apenas sincronizar os dados do leitor com o sistema. Isso elimina os erros de conferência, necessitando o mínimo de intervenção manual possível para atualizar as informações dos bens referente a localização. Como fonte de alimentação do leitor RFID foi utilizado a bateria Alcalina de 9V da Elgin, que são aprovadas pelo INMETRO e possuem um bom custo benefício. A Figura 29 mostra o leitor RFID desenvolvido com a placa Arduino e o Modulo RC522.

Figura 29 - Leitor RFID



Fonte: Elaborado pelo autor.

Em relação ao trabalho de Santos (2001), este trabalho se assemelha pelo fato de também visar o auxílio no gerenciamento de informação, gerando uma identificação única para cada animal, garantindo a rastreabilidade e histórico de seus dados. O leitor utilizado no seu trabalho, o SCL05 da Olimex, utiliza a frequência de 128 kHz, o que diminui o alcance e velocidade de leitura.

O trabalho de Zending (2005) também combina a identificação de rádio frequência a um sistema de informação com o intuito de controle de estoque, facilitando a identificação dos produtos por meio de um leitor RFID. O leitor utilizado (Z3070 da ZEBEK) opera na mesma frequência do leitor integrado nesse trabalho, 13.56 MHz, porém a utilização de um leitor comercializado não permite a customização do processo para se adaptar ao trabalho objetivado. O leitor desenvolvido aqui foi programado para atender duas funções, a

identificação e sincronização imediata da *tag*, para utilizar no processo de identificação do bem e também armazenamento das *tags* em memória para o processo de inventário. O custo é outro ponto em que os leitores se diferenciam, onde foi possível encontrar o leitor da Zebek na internet por cerca de R\$ 120,00, já o leitor desenvolvido aqui com Arduino e módulo RC522 custou cerca de R\$ 65,00 e pode ser utilizado com uma bateria externa eliminando a necessidade de utilizá-lo conectado ao computador, dando mais liberdade de operação, diferente do leitor da Zebek.

O trabalho de Dresch, Efrom e Grumovski (2008) também trata do controle patrimonial por meio de identificação por rádio frequência. Com o desenvolvimento de dois protótipos foi possível separar a necessidade de monitoramento em tempo real de alguns itens de alto valor, de outros de menor valor que podem ter seu inventário realizado periodicamente. Isso devido ao fato de monitoramento em tempo real exigir a aplicação de um sistema ativo de RFID, que eleva o preço e impossibilita a aplicação em itens de baixo valor.

4 CONCLUSÕES

O principal objetivo deste trabalho foi a identificação de bens patrimoniais e automação do inventário com tags RFID. Para isso foi desenvolvido os módulos web e desktop realizando a integração com o leitor RFID. Esse objetivo foi atendido.

Durante o processo de desenvolvimento foram encontradas dificuldades pela falta de conhecimento e experiência em eletrônica, e principalmente pela falta de instrumentos para diagnóstico, tornando alguns contratemplos uma grande dor de cabeça. Um exemplo disso foram os cabos utilizados para conectar a placa Arduino ao módulo RC522, pela praticidade e preço foi optado por utilizar cabo de rede e a falta de conhecimento da fragilidade desses cabos resultou em algumas horas de retestes e troca de cabos.

O trabalho desenvolvido atendeu a expectativa e objetivos definidos inicialmente. O módulo RC522, apesar do baixo custo, atendeu aos requisitos e possibilitou a identificação das *tags* satisfatoriamente. O ambiente de desenvolvimento e frameworks escolhidos também não deixaram a desejar, possibilitando rápido desenvolvimento e validação da solução proposta.

Um destaque nesse trabalho foi o baixo custo dos componentes para desenvolvimento do leitor RFID e facilidade de encontra-los no mercado. A placa arduino por ser um projeto de código aberto, se encontra com facilidade sendo comercializado na internet por um preço abaixo de R\$ 50,00.

Deve ser destacado que o autor inicialmente desconhecia Arduino e a tecnologia RFID. Mas a oportunidade de aplicar a tecnologia para solucionar o problema de identificação de bens no inventário patrimonial e o desejo de estudar a plataforma Arduino, levaram a escolha pelo tema dessa proposta. Tendo sido dedicado uma parcela do tempo para estudos e construção de protótipos.

4.1 LIMITAÇÕES

Uma limitação da placa utilizada (Arduino Uno) é sua memória EEPROM de 1Kb, sendo que cada *tag* possui quatro bytes, pode-se armazenar no leitor apenas duzentas e cinquenta e seis *tags* simultaneamente. Se considerarmos um ambiente de sala de aula, por exemplo, onde para cada estação podemos ter uma CPU, uma mesa, uma cadeira e um monitor, então seria possível fazer a leitura de apenas sessenta e quatro estações.

Caso a sala de aula possuir mais estações teria que ser inventariada em mais de uma única vez, ou substituir a placa utilizada no leitor pelo Arduino Mega, que possui 4Kb de memória EEPROM.

4.2 EXTENSÕES

Como melhorias para esse trabalho pode-se incluir no leitor RFID um Buzzer (componente eletrônico que quando recebe uma fonte de energia emite uma frequência sonora), que pode ser utilizada para que a cada *tag* RFID armazenada no leitor seja disparado o sinal indicando que a operação foi realizada. Dessa forma o usuário tem um reforço sonoro confirmando a leitura. Também é possível implementar uma verificação de redundância cíclica (CRC) no envio do código da *tag* do leitor para o software, garantindo que a identificação enviada não seja distorcida e recebida equivocadamente pelo software. Outra sugestão de melhoria é utilizar o módulo WiFi ESP8266 no leitor RFID, enviando a identificação das *tags* lidas diretamente para o Web Server, efetuando a realização do inventário diretamente do leitor sem a necessidade de sincronização dos dados armazenados posteriormente.

Como extensão propõe-se o desenvolvimento de uma solução para dispositivos móveis, integrada a arquitetura já desenvolvida. Dessa forma pode-se unificar os três módulos, podendo-se utilizar o sensor NFC para leitura das *tags*, o que poderia dispensar o uso do leitor RFID.

5 Referências

- ARDUINO E CIA. **Controle de acesso com módulo RFID RC522**. São Paulo, dez. 2014. Disponível em: < <http://www.arduinoecia.com.br/2014/12/control-de-acesso-modulo-rfid-rc522.html>>. Acessado em: 10 Dez. 2016.
- BHUPTANI, Manish; MORADPOUR, Shahram. **RFID: implementando o sistema de identificação por radiofrequência**. São Paulo : IMAM, 2005. 250 p, il.
- BISHOP, Judith. **C# 3.0 Design Patterns: Use the Power of C# 3.0 to Solve Real-World Problems**. Sebastopol, CA: O'reilly Media, 2007. 318 p.
- BRANAS, Rodrigo. **AngularJS Essentials**. Birmingham, Uk: Packt Publishing, 2014. 180 p.
- BRASIL. Lei n.º 6.404, de 15 de dezembro de 1976. Dispõe sobre as Sociedades por Ações. Diário Oficial [da] República Federativa do Brasil. Brasília. Disponível em < http://www.planalto.gov.br/ccivil_03/leis/L6404compilada.htm>. Acessado em: 28 Mar 2016.
- CASEIRO, Luiz Carlos Zalaf. **Novas estratégias de internacionalização de empresas brasileiras: expansão geográfica, determinantes e alternativas de política industrial**. 2013, 201 f. Dissertação (Mestrado em Sociologia). Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo. São Paulo.
- CPCON. **Inventário Patrimonial: Fundamental para sua Empresa.**, 2011. Disponível em: <<http://www.cpccon.eng.br/gestao-patrimonial/control/inventario-patrimonial/>>. Acessado em: 29 Out 2016.
- CRAFT, Brock. **Arduino projects for dummies**. 1 ed. Chichester: John Wiley & Sons , 2013. 408 p.
- CREATIVE COMMONS. **Whats is Arduino?**. 2015. Disponível em < <https://www.arduino.cc/en/Guide/Introduction> >. Acessado em: 06 Maio 2016.
- CRUZ, Tadeu. **Sistemas de informações gerenciais: Tecnologia da informação e a empresa do Século XXI**. Sao Paulo : Atlas, 1998. - 231p. :il..
- DALFOVO, Oscar. **Sistemas de informação : estudos e casos : o uso da informação pelos administradores e executivos que obtêm vantagem competitiva**. Blumenau : Acadêmica, 2004. xi, 293 p.
- DIAS, Renata Rampim de Freitas; BADALEI, Suely De Pieri. **Diferenças entre as frequências do sistema RFID passivo**. Jun. 2012. Disponível em: < <https://brasil.rfidjournal.com/artigos/vision?9591/>>. Acessado em: 06 Maio 2016.
- DIENER, Rodrigo Eglinton. **Protótipo de um sistema para monitoração de carros de competições, utilizando rádio frequência**. 2002 - xii, 59p. :il. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- DOBKIN, Daniel M. **The RF in RFID: UHF RFID in practice**. 2 ed. Waltham : Elsevier Inc. 2013 - 528p.
- DRESCH Jr, Antonio; EFROM, Danny R.; GRUMOVSKI, Dieison. Sistema de controle de patrimônio via RFID. **E-Tec: Atualidades Tecnológicas para Competitividade Industrial**, Florianópolis, v. 1, n. 1, p. 47-57, 1º. sem., 2008.
- DRISCOLL, Brian et al. **Entity Framework 6 Recipes**. New York, Ny: Apress, 2013. 548 p.

DXD ROTULOS. **Etiquetas RFID**. São Paulo, 2016. Disponível em <<http://www.dxdrotulos.com.br/etiquetas-rfid.php>> Acesso em: 26 Mar 2016.

FIJOR, Thiago Frederic Albert. **O controle patrimonial como ferramenta de governança pública**: O caso da universidade tecnológica federal do paran . 2014. 119 p.. Disserta o (Mestrado em Planejamento e Governan a P blica). Universidade Tecnol gica Federal do Paran , Curitiba.

ICStation. **RC522 RFID Module**. China, 2016. Disponível em: <<http://www.icstation.com/icstation-rc522-rfid-module-p-5322.html>> Acesso em: 07 Maio 2016.

LE O, Fabiana Ramalho Carneiro; SOUSA, Gisele Gomes de. **A automa o do invent rio patrimonial**: a experi ncia de Pernambuco no controle e localiza o dos bens m veis. In: CONGRESSO CONSAD DE GEST O P BLICA, 25 Mar. 2014, Bras lia.

MACHADO, L cia Cristina Vargas. **A influ ncia da tecnologia e da internet no marketing**. 2014. 10 f. Artigo Cient fico (Mestrado em Administra o pela Universidade Est cio de S ) – Faculdade Cenecista da Ilha do Governador, Rio de Janeiro.

MONK, Simon. **Programa o com Arduino**: come ando com sketches. Porto Alegre : Bookman, 2013. xi, 147 p., il.

NARCISO, Marcelo Gon alves. Aplica o da tecnologia de identifica o por r diofreq ncia (RFID) para controle de bens patrimoniais pela web. **Global science and technology**, v. 01, n. 07, p.50 - 59, dez/mar. 2008.

NUNES, Laura Cristina Menezes. **O emprego da teoria agente/principal para defini o de modelo de gest o de bens m veis na universidade de Bras lia**. 2015. 151 f. Disserta o (Gest o P blica) – Curso de P s gradua o em Gest o P blica, Faculdade UnB Planaltina, Bras lia – DF.

NXP Semiconductors Austria GmbH Styria. **About MIFARE**. Austria, 2013. Disponível em: <<https://www.mifare.net/en/about-mifare/>>. Acessado em: 17 Nov. 2016.

O'BRIEN, James A; MARAKAS, George M. **Administra o de sistemas de informa o**. 15. ed. Porto Alegre : AMGH, 2013. xxix, 590 p, il.

PAULI, Rog rio. **DOM TICA**: Automa o residencial usando plataforma arduino. 2016. 89 p. Trabalho de Conclus o de Curso (Bacharelado em Tecnologia em Redes de Computadores), Universidade Federal de Santa Maria, Santa Maria.

PORTO, Maria Alice Guedes; BANDEIRA, Anselmo Alves. **A import ncia dos sistemas de informa o gerenciais para as organiza o**. In: XIII SIMPEP - Bauru, SP, Brasil, 6 a 8 de Novembro de 2006

POZO, Hamilton. **Administra o de recursos materiais e patrimoniais**: uma abordagem log stica. 6. ed. S o Paulo : Atlas, 2010. 210 p., il.

SANTOS, Andr  Fraga. **Prot tipo de um sistema de identifica o eletr nica de animais atrav s de r dio freq ncia**. 2001. - xi, 77 f. :il. Trabalho de Conclus o de Curso (Bacharelado em Ci ncias da Computa o) – Centro de Ci ncias Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SCHWARTZ, Marco. **Internet of things with arduino**: Cookbook. 1 ed. Livery Place: Packt Publishing, 2016. 187 p.

SOUZA, F bio. **Arduino UNO**. 2013. Disponível em <<https://www.embarcados.com.br/arduino-uno/>>. Acessado em: 13 Nov. 2016.

SOUZA, Gleim Dias de; CARVALHO, Maria do Socorro M. V. de; LIBOREIRO, Manuel Alejandro Martínez. Gestão da cadeia de suprimentos integrada à tecnologia da informação. **Revista de Administração Pública**. Rio de Janeiro, v. 40, n. 4, ago. 2006.

ZANLUCA, Júlio César. **Demonstrações Contábeis**. [2009]. Disponível em <<http://www.portaldecontabilidade.com.br/tematicas/demonstracoescontabeis.htm>> Acesso em: 28 Mar 2016.

ZEINDIN, Denise Carla dos Anjos. **Sistema de informação aplicado à integração da cadeia de suprimentos utilizando tecnologia RFID**. 2005. 87 p. :il. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

APÊNDICE A – Descrição dos Casos de Uso

Esta seção da proposta apresenta a descrição dos casos de uso conforme previstos no(s) diagrama(s) apresentado(s) na seção 3.3.1.

UC01 Efetuar Login

Permite aos usuários fornecerem suas credenciais para sua identificação e acesso ao sistema.

Constraints

Pré-condição: O usuário deve estar cadastrado no sistema.

Pós-condição: Usuário conectado.

Cenários

Login {Principal}.

1. Usuário acessa o endereço eletrônico da aplicação;
2. Usuário efetua autenticação;
3. Sistema localiza o cadastro do usuário;
4. Usuário fornece a senha de usuário;
5. Sistema valida senha;
6. Sistema direciona para página principal da aplicação.

Usuário não localizado {Exceção}

No passo 3, ao buscar pelo cadastro de usuário, caso o mesmo não for localizado o sistema deve possibilitar o usuário solicitar o seu cadastro para acesso ao sistema.

- 1.1 Sistema exibe caixa para preenchimento do motivo do acesso;
- 1.2 Usuário preenche motivo;
- 1.3 Sistema exibe dados básicos de usuário;
- 1.4 Usuário preenche dados solicitados;
- 1.5 Usuário envia solicitação de acesso;
- 1.6 Sistema notifica o administrador do sistema com os dados para cadastro.

UC02 Consulta relação de Bens

Permite ao usuário visualizar os bens de sob sua responsabilidade.

Constraints

Pré-condição: O usuário deve estar autenticado.

UC03 Transferência de Bens

Permite ao usuário registrar a movimentação de bens para outro departamento ou local.

Constraints

Pré-condição: O usuário deve estar cadastrado no sistema.

Pré-condição: O usuário deve possuir bens sob sua responsabilidade.

Pós-condição: Solicitação de transferência gerada.

Pós-condição: Dados do bem atualizados.

Cenários

Registro de Transferência {Principal}.

1. Usuário seleciona bens a transferir;
2. Usuário seleciona departamento e local de destino;
3. Sistema valida os dados;

Solicitação de aprovação da Transferência {Secundário}.

Ao informar os dados de destino caso o responsável pelo departamento ou local não seja o mesmo usuário, o sistema deve gerar uma solicitação de aprovação.

- 3.1 Sistema gera solicitação de aprovação da transferência;
- 3.2 Sistema envia notificação ao responsável do departamento e local de destino.

UC04 Aprovação de movimentação

Permite ao usuário aprovar ou reprovando uma solicitação de transferência de bem para seu departamento ou local.

Constraints

Pré-condição: O usuário deve ser responsável por algum departamento ou local.

Pós-condição: Solicitação aprovada ou reprovada.

Pós-condição: Dados do bem em questão atualizados.

Cenários**Aprovação da transferência {Principal}.**

1. Usuário seleciona os bens a aprovar ou reprovando;
2. Usuário informa um motivo opcionalmente;
3. Usuário informa ação desejada (Aprovar / Reprovar);
4. Sistema envia notificação de aprovação ao responsável origem com o motivo informado;

UC05 Conferência de Bens

Permite ao usuário sincronizar os dados das *tags* lidas de departamentos e locais sob sua responsabilidade para atualização do sistema em período de inventário.

Constraints

Pré-condição: O usuário deve ser responsável por algum departamento ou local.

Pós-condição: Registro de inventário inserido.

Cenários**Conferência {Principal}.**

1. Usuário seleciona departamento e local;
2. Usuário conecta o leitor ao computador;
3. Usuário dispara ação para sincronizar dados obtidos do leitor;
4. Sistema localiza a porta USB onde está conectado o leitor;
5. Sistema solicita identificação das *tags* lidas ao leitor;
6. Sistema exibe em tela os bens identificados pelo leitor;
7. Usuário confirma dados exibidos;
8. Sistema valida informações com os bens no atual departamento e localização;
9. Sistema insere registro de inventário;
10. Sistema envia mensagem ao leitor para apagar os itens em memória.

Bem identificado não pertence ao departamento atual {exceção}.

No item 8 ao sistema validar as informações pode identificar bens que não estão departamento ou local atual. Nesse caso deve indicar ao usuário a opção de solicitar transferência.

- 8.1 Sistema indica os bens que não estão no departamento ou local onde esta sendo realizado o inventário no momento;
- 8.2 Sistema indica a opção de solicitar a transferências dos bens em questão;
- 8.3 Usuário aceita solicitação de transferência;
- 8.4 Sistema gera registro da solicitação de transferência.

UC06 Atribuir responsável por local

Permite o usuário atribuir um responsável secundário a um local que possui bens sob sua responsabilidade. Apenas um responsável pode ser atribuído por local.

Constraints

Pré-condição: O usuário deve ser responsável por algum departamento.

Pós-condição: Registro de responsável por departamento e local inseridos.

Cenários**Atribuição de responsável {Principal}.**

1. Usuário seleciona um departamento sob sua responsabilidade;
2. Sistema exhibe os locais que já possuem outros responsáveis atribuídos;
3. Usuário seleciona opção de inserir novo responsável por local;
4. Usuário seleciona um local;
5. Usuário seleciona outro usuário para atribuí-lo como responsável;
6. Sistema registra o responsável pelo local e departamento.

Remoção de responsável {Alternativo}.

No item 2 ao listar os responsáveis já atribuídos o usuário pode remover um desses responsáveis.

- 2.1 Usuário seleciona opção de remover atribuição;
- 2.2 Sistema remove registro de responsável por local.

UC07 Identificação da tag RFID

Permite o usuário identificar as tags RFID nos bens sob sua responsabilidade para posterior sincronização com o sistema;

Constraints

Pré-condição: O usuário deve estar de posse de um leitor RFID.

Pós-condição: Leitor contendo as identificações das tags.

Cenários**Identificação {Principal}.**

1. Usuário aproxima o leitor da tag colocada no bem;
2. O leitor recebe os dados da tag e verifica se o identificador está na lista em memória;
3. Caso não esteja na lista o leitor insere o item na lista.

UC08 Manutenção de usuários

Permite ao administrador cadastrar novos usuários do sistema.

Constraints

Pré-condição: O usuário deve ser administrador.

Pós-condição: Cadastro de novo usuário inserido.

Pós-condição: Cadastro de usuário inabilitado.

Cenários

Consulta de usuário {Principal}

1. Administrador informa o login do usuário;
2. Sistema localiza usuário cadastrado;
3. Sistema exibe os dados do usuário.

Cadastro de usuário {Alternativo}.

No item 2, caso o login do usuário não for localizado o sistema disponibiliza a inclusão do mesmo.

- 2.1 Administrador informa e-mail do novo usuário;
- 2.2 Administrador confirma o cadastro de usuário;
- 2.3 Sistema cadastra o novo usuário.

Exclusão de usuário {Alternativo}.

No item 3, ao exibir os dados do usuário o sistema disponibiliza a opção de exclusão do mesmo. Caso o usuário seja responsável por departamento ou local um substituto deverá ser indicado.

- 3.1 Administrador seleciona opção de exclusão de usuário;
- 3.2 Sistema solicita confirmação da ação;
- 3.3 Administrador confirma exclusão;
- 3.4 Sistema verifica os departamentos ou locais sob responsabilidade do usuário;
- 3.5 Sistema solicita indicação de um responsável substituto;
- 3.6 Administrador informa um usuário substituto;
- 3.7 Sistema efetua a alteração dos departamentos e locais;
- 3.8 Sistema inativa o cadastro do usuário.

UC09 Manutenção de Localizações

Permite ao administrador consultar e cadastrar novos locais.

Constraints

Pré-condição: O usuário deve ser administrador.

Pós-condição: Local cadastrado.

Cenários

Consulta de localizações {Principal}.

1. Administrador acessa cadastro de locais;
2. Sistema exibe os locais cadastrados.

Inclusão de Local {Alternativo}.

No item 2, ao visualizar os locais cadastrados o administrador pode cadastrar novos locais.

- 2.1 Administrador acessa função de cadastro de local;
- 2.2 Administrador informa descrição de novo local;
- 2.3 Administrador confirma dados do novo local;
- 2.4 Sistema efetua o registro do local.

Exclusão de Local {Alternativo}.

No item 2, ao visualizar os locais cadastrados o administrador pode excluir um local.

- 2.1 Administrador seleciona a ação de excluir um local;
- 2.2 Sistema verifica se local possui bens atribuídos a ele;
- 2.3 Sistema informa ao administrador registro de bens atribuídos a esse local;
- 2.4 Sistema solicita um local substituto;
- 2.5 Administrador informa local substituto;
- 2.6 Sistema Atualiza a informação de local dos bens atribuídos ao local a ser excluído;
- 2.7 Sistema notifica os usuários responsáveis pelos departamentos sobre a substituição;
- 2.8 Sistema remove o registro do local.

UC10 Manutenção de Departamentos

Permite ao administrador consultar e cadastrar novos departamentos.

Constraints

Pré-condição: O usuário deve ser administrador.

Pós-condição: Departamento cadastrado.

Cenários**Consulta de departamentos {Principal}.**

1. Administrador acessa cadastro de departamentos;
2. Sistema exhibe os departamentos cadastrados.

Inclusão de departamento {Alternativo}.

No item 2, ao consultar os departamentos o administrador pode cadastrar novos.

- 2.1 Administrador acessa a ação de inclusão de departamento;
- 2.2 Administrador informa a descrição do novo departamento;
- 2.3 Administrador informa o responsável pelo novo departamento;
- 2.4 Administrador confirma a inclusão do departamento;
- 2.5 Sistema registra o cadastro do novo departamento.

Exclusão de departamento {Alternativo}.

No item 2, ao consultar os departamentos cadastrados o administrador pode remover um deles.

- 2.1 Administrador acessa função de exclusão de um departamento;
- 2.2 Sistema verifica se o departamento possui bens cadastrados;
- 2.3 Sistema informa a existência de bens cadastrados para o departamento;
- 2.4 Sistema solicita ao administrador informar um departamento substituto;
- 2.5 Administrador informa um departamento substituto;
- 2.6 Sistema atualiza o departamento dos bens em questão;
- 2.7 Sistema remove o registro do departamento.

UC11 Manutenção de Bens

Permite ao administrador consultar e cadastrar novos bens patrimoniais.

Constraints

Pré-condição: O usuário deve ser administrador.

Pós-condição: Bem cadastrado.

Cenários**Consulta de Bens {Principal}.**

1. Administrador acessa cadastro de bens;
2. Sistema exhibe os bens cadastrados.

Inclusão de bem {Alternativo}.

No item 2, ao consultar os bens o administrador pode cadastrar novos.

- 2.1 Administrador acessa a ação de inclusão de bem;
- 2.2 Administrador informa a descrição do novo bem;
- 2.3 Sistema solicita conexão com o leitor de *tags* RFID;
- 2.4 Administrador conecta leitor de *tags*;
- 2.5 Sistema envia mensagem ao leitor de início de leitura;
- 2.6 Administrador aproxima o leitor da *tag*;
- 2.7 Leitor envia identificação da *tag* para o sistema;
- 2.8 Sistema valida identificador da *tag*;
- 2.9 Sistema cadastra o bem identificado.

Baixa de bem patrimonial {Alternativo}.

No item 2, ao consultar os bens cadastrados o administrador pode dar baixa de um bem informando o motivo para isso.

- 2.1 Administrador acessa função de baixa de um bem;
- 2.2 Administrador informa motivo de baixa do bem;
- 2.3 Sistema notifica responsável pelo bem referente a baixa;
- 2.4 Sistema efetua a baixa do bem patrimonial.

UC12 Registro de período de inventário

Permite ao administrador efetuar o registro de período de inventário para que os responsáveis por departamento ou local efetuem a conferência de seu inventário.

Constraints

Pré-condição: O usuário deve ser administrador.

Pós-condição: Registro do período de inventário efetuado.

Cenários**Consulta de períodos cadastrados {Principal}.**

1. Administrador acessa o cadastro de período de inventário;
2. Sistema exibe os períodos de inventários cadastrados.

Inclusão período de inventário {Alternativo}.

No item 2, ao consultar os períodos de inventários cadastrados o administrador pode cadastrar novo períodos.

- 2.1 Administrador acessa a ação de inclusão de período de inventário;
- 2.2 Administrador informa a data de início e fim do período de inventário;
- 2.3 Sistema registra o período de inventário;
- 2.4 Sistema notifica os responsáveis por departamento e local sobre o período de inventário cadastrado;

Exclusão de período de inventário {Alternativo}.

No item 2, ao consultar os bens cadastrados o administrador pode remover um período de inventário cadastrado que não iniciou ainda.

- 2.1 Administrador acessa função de exclusão do período de inventário;
- 2.2 Sistema remove o período de inventário cadastrado;
- 2.3 Sistema notifica responsáveis por departamento e local sobre a exclusão do período de inventário.

UC13 Emissão de relação de bens

Permite ao administrador emitir um relatório com todos os bens da organização agrupados por departamento.

Constraints

Pré-condição: O usuário deve ser administrador.

Pós-condição: Relatório de bens emitido.

Cenários**Emissão da relação de bens {Principal}.**

1. Administrador acessa a relação de bens;
2. Administrador emite a relação de bens;
3. Sistema gera relatório com todos os bens agrupados.