

Sistema de compras automatizado em supermercados

Vinícius Perondi¹ e Fernando Covolan Rosito²

TCC2 - Curso de Engenharia de Controle e Automação

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Câmpus Farroupilha
Farroupilha, Brasil

Discente¹, orientador²

perondivinicius@gmail.com¹, fernando.rosito@farroupilha.ifrs.edu.br²

Resumo — Este artigo é elaborado dentro da disciplina de Trabalho de Conclusão de Curso II do curso de Engenharia de Controle e Automação do IFRS - Câmpus Farroupilha e tem como principal objetivo desenvolver uma solução para otimizar o tempo de compras e das filas no ato do pagamento nos supermercados através do uso da tecnologia *RFID*. A identificação por radiofrequência é uma tecnologia usada para identificar eletronicamente a presença de uma etiqueta (*tag*), usando sinais de rádio. Desta forma, este trabalho propõe o desenvolvimento de um sistema que identifica automaticamente as mercadorias inseridas no carrinho de compras, informando ao consumidor os itens e o somatório total do valor a ser cobrado pelo supermercado. O carrinho de compras automatizado é composto por um microcontrolador, uma antena, um leitor de radiofrequência, um tablet com ícones de conexão, desconexão e atualização de compras, um módulo *bluetooth* e uma memória externa. Cada produto será identificado com uma *tag*, que será lida pelo leitor *RFID* no momento em que a mercadoria estiver dentro do carrinho. O consumidor poderá visualizar a lista de mercadorias e o valor total de suas compras em um *tablet* instalado, também, no carrinho. Ao final, o consumidor deverá se dirigir ao caixa, onde o funcionário fará a cobrança com base nos dados transmitidos, via *bluetooth*, do carrinho ao computador do caixa de pagamento. Desta forma, o procedimento de compras e, principalmente, o ato de pagamento é otimizado, reduzindo o tempo de permanência do consumidor no supermercado. O resultado deste trabalho evidencia cuidados que se deve ter ao trabalhar com *RFID* exemplificando alguns passos e afirmando o bom desempenho da tecnologia nos supermercados.

Palavras-chave — *Tag, RFID, bluetooth*.

I. INTRODUÇÃO

Os supermercados são empresas varejistas, ou seja, representam o último elo da cadeia comercial entre um produto e seu consumidor final. Estes estabelecimentos vendem proeminentemente mercadorias dispostas em um formato de autoatendimento e dispõem de caixas para pagamento na saída. Estas empresas têm-se preocupado em melhorar o nível de serviços ao consumidor para fidelizá-lo, com o intuito de aumentar o volume de vendas. Desta maneira, pode-se observar uma forte tendência de grandes redes de supermercados em fazer uso de tecnologias mais avançadas que permitam utilizar um menor número de empregados, maior número de caixas para atendimento, maior

flexibilidade de horário de funcionamento, disposição mais organizada das mercadorias e maior variedade dos produtos [1].

Em meados da década 90, a consultora *Price Waterhouse* previu que em 2005 os supermercados não teriam mais operadores de caixas e carrinhos de compras, e os depósitos seriam operados por robôs que entregariam as compras já embaladas. Na maioria dos supermercados isto não ocorreu, contudo, nos supermercados *Winn Dixie* (EUA), por exemplo, os caixas são ditos automáticos (não têm funcionários – método mais autônomo), ficando o cliente responsável por todo o processo de passar os itens no caixa e pelo seu pagamento. Em 1996, segundo a consultora *Nielsen*, os principais motivos apontados de queixas dos clientes em supermercados referiam-se a filas no caixa (56%), supermercado lotado (41%), falta de empacotador (35%) e falta de preço nos produtos (25%). Nota-se que todos estes motivos afetam o tempo de permanência do consumidor no supermercado [1].

Outra pesquisa encomendada pela AMIS (Associação Mineira de Supermercados) e divulgada durante a 23^o SUPERMINAS (congresso e feira supermercadista), em outubro de 2009, revelou que 63% dos entrevistados disseram que não gostavam de filas, 7% não gostavam dos preços abusivos e 5,4% reclamaram do mau atendimento. Os 24,6% restantes foram divididos entre outros fatores de descontentamentos dos clientes [2].

Assim, uma preocupação dos gerentes de supermercados é reduzir o tamanho das filas nos caixas. Ao aumentar o número de caixas em operação, o gerente reduz o tempo médio de espera em filas, mas, por outro lado, estará aumentando os custos operacionais do sistema [2].

Neste contexto, o presente trabalho tem como principal objetivo desenvolver uma solução para os supermercados através do uso da tecnologia *RFID* (*Radio-Frequency Identification*), visando otimizar o tempo de compras e das filas no ato do pagamento.

A estrutura do trabalho é apresentada da seguinte forma: O capítulo I apresenta a introdução, o capítulo II expõe uma revisão bibliográfica, o capítulo III apresenta a fundamentação teórica, o capítulo IV descreve o desenvolvimento da proposta estabelecida como projeto, o capítulo V apresenta os

resultados obtidos e, por fim, o capítulo VI relata a conclusão do trabalho.

II. REVISÃO BIBLIOGRÁFICA

A identificação por radiofrequência *RFID* é uma tecnologia que tem potencial para fazer grandes impactos na economia. Estudos estão revolucionando a cadeia de suprimentos, controle de estoque e logística [3]. Desta maneira, é uma ferramenta importante no tema deste artigo.

O trabalho [3] consiste em um carrinho de supermercado com hardware anexado, composto, basicamente, por um microcontrolador, um display, um leitor *RFID* (responsável por ler as etiquetas nos produtos), um transceptor *Zigbee* e uma bateria como fonte de energia. A união desses equipamentos, necessários para efetuar a identificação dos produtos, denomina-se *PID* (*Product Identification Device*).

Desta forma, cada carrinho possui um microcontrolador, que está acoplado a um leitor de etiquetas (*tags*) e a um transceptor *Zigbee*. O microcontrolador é responsável por receber, do leitor de etiquetas, informações dos produtos, e enviá-las, através da comunicação *Zigbee*, ao sistema central de faturamento, que é localizado nos caixas e composto por um computador. Este sistema central de faturamento possui um banco de dados e calcula o preço líquido para os produtos adquiridos. O cliente pode obter suas informações de faturamento no caixa de pagamento de acordo com o número de identificação do carrinho. O cliente somente faz o pagamento no caixa, sem há necessidade de passar as compras em um leitor de código de barras como normalmente é feito em caixas convencionais [3].

Em outro artigo, cada carrinho tem um *PID* contendo um microcontrolador, uma tela *LCD* 16x4, um leitor de *RFID*, um teclado, uma campainha, um sensor de entrada de produtos no carrinho e outro sensor de retirada de produtos, um módulo transceptor de radiofrequência (*RF*), e uma fonte de alimentação de 12V [4].

Cada produto colocado ou retirado do carrinho é atualizado na tela *LCD* com o valor total das compras. O carrinho contém um banco de dados salvo na memória do microcontrolador e o sistema de faturamento é centralizado e automatizado usando *RF*, ou seja, ao chegar ao caixa de pagamento, o carrinho transmite as informações das compras via *RF*. Assim, o computador, localizado no caixa de pagamento, gera a nota fiscal com os itens e o total a ser pago. No projeto elaborado, existe também a opção em que o cliente pode informar o valor total que pretende gastar e, no momento em que este valor for atingido, um bipe é disparado, alertando do limite estipulado pelo cliente [4].

No trabalho [5], realizou-se uma pesquisa onde leitores *RFID* são instalados nos corredores dos supermercados, ao invés de serem colocados nos carrinhos. O estudo teve como objetivo diminuir filas, facilitar o pagamento e reduzir custos com antenas e leitores. Este artigo teve como foco apresentar um estudo da aplicação do *RFID* a nível de corredor, diferente dos demais artigos citados, onde utilizou-se o *PID* diretamente instalado no carrinho. Foram feitas apenas simulações e verificações para identificar a intensidade dos sinais e a

possibilidade de implantação. Não possui protótipo para validação dos dados.

III. FUNDAMENTAÇÃO TEÓRICA

A identificação por radiofrequência (*RFID*) é uma tecnologia usada para registrar eletronicamente a presença de um objeto usando sinais de rádio [3]. Esta tecnologia teve sua origem em 1897 quando Guglielmo Marconi inventou o rádio, e se difundiu na década de 30, quando o Exército e a Marinha Americana tiveram o desafio de projetar um sistema que fosse capaz de identificar objetos no chão, no mar e no ar. Então, em 1937, o Laboratório de Pesquisa Naval Norte Americano desenvolveu o *Identification Friend or Foe (IFF)* que se tornou base para o controle de tráfego aéreo e se tornou precursor do que hoje é a *RFID* [6].

Nos meados dos anos 60 e 70 as empresas *Sensormatic* e *Checkpoint Systems* desenvolveram uma das primeiras aplicações baseadas em *RFID*, que era o controle de segurança de produtos por meio eletrônico chamado de *Electronic Article Surveillance (EAS)*. Este sistema utilizava uma etiqueta passiva (sem uso de energia) de 1 bit acopladas aos objetos que, ao passarem por um dispositivo de leitura que ficava localizada na saída do recinto, acionava um alarme alertando os funcionários de roubos. Em outras situações, o sistema acionava câmeras de vídeo para capturar o movimento do objeto [6].

No início dos anos 90, ocorreu a evolução das etiquetas (*tags*), onde as passivas que trabalhavam em baixa frequência (*Low Frequency - LF*) e alta frequência (*High Frequency - HF*) eram muito restritivas devido às poucas distâncias de leitura. Assim, surgiu a etiqueta de frequência extremamente alta (*Ultra High Frequency - UHF*), que permitiu o avanço na utilização comercial, pois com este tipo de etiqueta era possível ler centenas e até milhares de etiquetas ao mesmo tempo passando, em alta velocidade, pelo leitor, devido à alta frequência alcançada pelas etiquetas [6].

Portanto, a arquitetura da tecnologia *RFID* se baseia nos seguintes componentes: etiqueta, antena e leitor *RFID*.

A. Etiqueta

A etiqueta (*tag*) tem a função de armazenar dados sobre o produto, como: preço, detalhes nutritivos, data de validade, entre outros [4], [5]. Estas podem ser ativas, passivas ou semi-passivas. A etiqueta passiva é um tipo de etiqueta que não necessita de bateria. Esta utiliza o próprio sinal enviado pela leitora para reenviar o sinal de retorno contendo os dados do objeto [6]. *Tags* ativas possuem uma bateria integrada que permite o envio de sinal a uma área maior. Normalmente este tipo de etiqueta permite regravação dos dados e é comumente usada para rastreamento de cargas [6]. Já as etiquetas semi-passivas é um tipo de etiqueta que se utiliza da bateria para operação do chip da etiqueta e também se utiliza do sinal da leitora para re-enviar o sinal. Esta etiqueta, por necessitar da onda de rádio do leitor, seu campo também é restrito a área da leitura [6].

B. Antena

A Antena é encontrada tanto nas etiquetas quanto na leitora e tem a função de servir como canal de comunicação entre os dispositivos através de sinais de rádio. Na etiqueta a antena é conectada diretamente ao seu microprocessador, já no dispositivo de leitura, a antena pode estar conectada diretamente ao próprio dispositivo ou pode estar conectada a grandes distâncias da leitora [6].

C. Leitor

Para que as *tags* se comuniquem com o restante do sistema, os leitores de *RFID* coletam e leem os sinais de rádio frequência emitidos pelas etiquetas [5] e são capazes de identificar um número elevado de *tags* em um curto espaço de tempo. A aplicação destes leitores acoplados aos microcontroladores possibilita gerenciar os dados coletados e automatizar processos.

Um fator importante para a comunicação entre etiquetas e leitores é a frequência, pois a frequência *UHF* (*Ultra High Frequency*), pode ser encontrada na faixa de 433 MHz e também de (860 a 930) MHz. Este tipo de frequência apresenta como principais características: maior área de leitura do que a alta frequência (HF), longa área de transmissão para 433 MHz, grande aceitação no mercado mundial devido a utilização da cadeia de suprimentos e varejos [6].

IV. PROPOSTA

O presente trabalho tem como principal objetivo desenvolver uma solução para supermercados através do uso da tecnologia *RFID*, visando otimizar o tempo de permanência do consumidor no supermercado e as filas no ato do pagamento. A Fig. 1 demonstra os equipamentos utilizados para realizar este trabalho, ou seja: uma antena conectada ao um leitor *RFID* que detecta as *tags* fixadas nos produtos; um Arduino Mega [7] que processa e gerencia as informações coletadas; um cartão de memória *SD* [8] que potencializa o armazenamento dos produtos a serem acumulados durante a compra; um módulo *bluetooth* [9] que conecta a um *tablet* (fixado no carrinho) e ao computador do caixa, com o intuito de enviar as informações gerenciadas pelo Arduino.

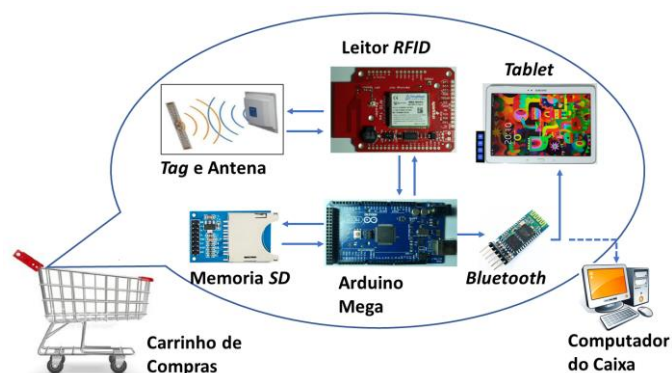


Fig. 1. Equipamentos utilizados no projeto.

Utilizando estas tecnologias o cliente passa a ter a comodidade de entrar no supermercado, selecionar seus

produtos, colocar dentro do carrinho e a qualquer momento visualizar a lista e o valor total das compras pretendidas nos pontos de apoio. Após o término da seleção dos produtos é necessário passar em um caixa, que possui um computador, que de forma rápida coletará as informações dos produtos do carrinho com o valor total das compras através de uma transmissão de dados via *bluetooth*, pareando o computador com o carrinho (Arduino Mega). Assim, se efetua a cobrança de maneira ágil, sem a necessidade de manusear cada item e passar em um leitor de código de barras, como atualmente são feitos nos caixas na maior parte das redes de supermercados do Brasil.

A. Comunicação.

A comunicação entre os dispositivos é feita por etapas, iniciando pelo dispositivo leitor de *RFID*. Este dispositivo é utilizado para captar o sinal das *tags* e envia as informações coletadas via serial *UART* (*Universal Asynchronous Receiver Transmitter*), *RX / TX* (*Receiver/ Transmitter*) para o microcontrolador do dispositivo Arduino Mega.

Após realizar as capturas, o Arduino envia os dados para serem armazenados em um cartão *SD*. Esta comunicação é realizada via *SPI* (*Serial Peripheral Interface*) [8].

Na etapa final das compras quando o cliente efetua a atualização dos itens contidos no carrinho, é necessário enviar as informações para um módulo *bluetooth*. As informações até o módulo são enviadas via *UART*, *RX/TX*, para somente então a informação ser enviada via *bluetooth* para os dispositivos *tablet* ou computador.

B. Ações do Cliente.

Ao chegar no supermercado e selecionar um carrinho de compras com leitor *RFID*, que foi projetado com duas bandejas de forma que cada compartimento tenha um tipo específico de embalagem, o cliente deve escolher os produtos que deseja comprar. Ele deve acomodar os itens no interior do carrinho de forma que os produtos líquidos, como, por exemplo, refrigerantes, produtos de limpeza e os produtos enlatados (leite condensado, achocolatados, etc.) sejam inseridos na parte de baixo do carrinho, organizando os itens com a *tag* posicionada para cima, para que seja identificado mais rapidamente pelo leitor. O restante dos produtos pode ser alocado na parte superior do carrinho, sem ter a preocupação de ter a *tag* exposta.

Assim, após selecionar seus itens o cliente posiciona o carrinho nos locais predeterminados no supermercado (para evitar interferência de outros carrinhos e produtos das prateleiras), pressiona o ícone “Atualizar Compras” no visor do *tablet* e em poucos segundos a lista de compras aparecerá no display do *tablet*, discriminando o item e o valor unitário, assim como o valor total dos produtos. Desta forma, o sistema permite ao cliente verificar sua lista de compras antes de chegar até o caixa.

A etapa final do processo de compras é feita no caixa de cobrança. O cliente deve aproximar o carrinho ao guichê do caixa, e um operador irá parear o carrinho com o computador, utilizando o serial monitor disponibilizado pela plataforma de

desenvolvimento Arduino atrelado a um *firmware* desenvolvido especificamente para este tipo de conexão, transmitindo assim a lista de compras, salva no cartão de memória do carrinho. Para parear o carrinho com o computador, é necessário que um operador pressione o ícone “Desconectar” no *tablet*. Neste momento a comunicação entre o *tablet* e o carrinho é interrompida. Após é realizada a conexão do carrinho com o computador do caixa, e para isso, é necessário seguir os procedimentos descritos no decorrer do item *Firmware* deste trabalho.

Assim, com essa última conexão, é possível gerar a nota fiscal através do computador do caixa. Com este processo de compras não será necessário manusear todos os itens do carrinho para efetuar a leitura do preço. Com a implementação deste projeto deve ser possível evitar as longas filas nos caixas tornando o ato de fazer compras otimizado. A Fig. 2 ilustra os passos de todo o processo de compras. Neste trabalho não será gerado a nota fiscal, somente será gerado uma lista de compras que será salva no arquivo *Lista_Compras.txt*.

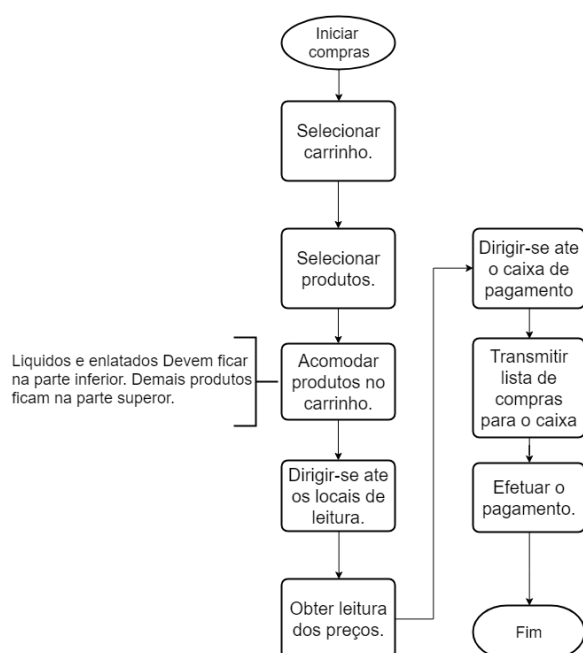


Fig. 2. Processo de compras.

C. Instalação da Antena e Estrutura

Para definir o local da instalação da antena realizou-se diversos testes de posicionamento da mesma, de forma a atingir toda a área do carrinho. A Fig. 3 (a) mostra os primeiros testes de posicionamento e leitura das *tags*, já a Fig. 3 (b) ilustra o posicionamento final com um revestimento interno em plástico. Verifica-se que a estrutura do suporte foi alterada, de um material de madeira para um de alumínio, assim como a angulação, deixando a antena um pouco mais abaixo para detectar melhor as *tags* na parte inferior do carrinho. Para tal posição, foi necessário realizar vários testes para encontrar a melhor instalação da antena, tendo em vista a necessidade de leitura dos produtos nos dois compartimentos do carrinho. O revestimento interno foi adicionado para evitar interferência na leitura das *tags* em função do material utilizado para

construção do carrinho. Assim, foi utilizado um material plástico para evitar que os produtos fiquem em contato com as partes metálicas da estrutura.



(a) Posicionamento inicial da antena;

(b) Posicionamento final da antena.

Fig. 3. (a) Posicionamento inicial da antena;
(b) Posicionamento final da antena.

Para um bom funcionamento do sistema de leitura no supermercado, deve-se evitar realizar leituras dos produtos contidos no carrinho próximo as prateleiras e gondolas, pois o sistema está configurado para efetuar leituras no alcance de até 4,9 m. Esta configuração foi necessária para atingir a parte inferior do carrinho. Desta maneira, o supermercado deve ter pontos de leitura e atualização da lista de compras.

D. Leitura dos Dados da Tag

Com a utilização do módulo leitor de *tags SRTR (Simultaneous RFID Tag Reader)* [10], existem duas maneiras de identificar as *tags* e, respectivamente, os produtos. Uma maneira é efetuando a leitura dos dados que possuem 64 bytes. Esses dados permitem que o programador descreva de forma simples os itens lidos, como mostra a Fig. 4, que ilustra a informação de uma *tag* que foi lida e programada para aparecer a descrição conforme destaque. Para isso basta utilizar um *firmware* dedicado para modificação e edição dos dados das *tags*. Cada byte representa um caractere, ou seja, utilizando o modo dados, o programador tem como possibilidade de escrita apenas 64 caracteres. Como neste modo a informação fica armazenada na *tag* (preço, descrição, validade), não necessita de um banco de dados, somente necessita a aplicação dos filtros para evitar repetições de leitura. Porém, este método possui maior quantidade de caracteres a ser lido e conseqüentemente acaba demorando mais para atualizar a lista de compras.

A *tag*, uma vez gravada previamente com os dados, ela não pode ser regravada. Caso algum item esteja com seus dados errados, não poderá corrigir a informação na *tag*. A alternativa é descartar e utilizar outra. Esta situação pode ser um problema sempre que houver a necessidade de mudança de preço dos itens, por exemplo, quando um item entra em promoção, o supermercado teria que substituir todas as *tags* dos produtos da promoção para mudar para o novo preço.

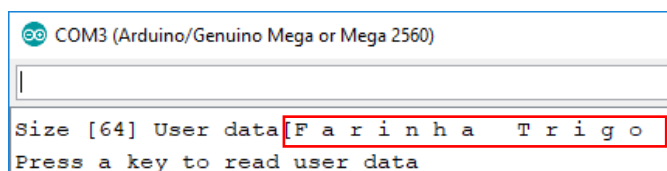


Fig. 4. Dados de 64 bytes lidos individualmente de uma *tag*.

A outra maneira de identificar as *tags* é realizando a leitura do *EPC* (*Electronic Product Code*), que funciona como um número de identificação das *tags*. Esse número é descrito em 12 bytes e cada *tag* possui uma identificação diferente vindo de fábrica, ou seja, não necessita que o programador grave previamente os dados na *tag*, basta somente realizar uma leitura para identificar o *EPC* e utilizar este número, atrelando-o a um produto. Para isso, esse produto deve estar previamente cadastrado em um banco de dados. A Fig. 5 demonstra uma leitura de *EPC*, em destaque, os 12 bytes, porém, sem identificação do produto, preço, validade. Pois, todas as informações estarão salvas no banco de dados.

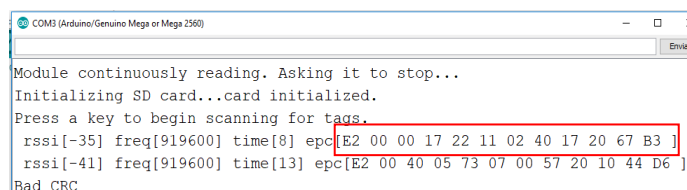


Fig. 5. Leitura de dados do *EPC* da *tag*.

Inicialmente, testes foram realizados para identificar os dados referentes às informações dos produtos (Produto, Preço, Validade), utilizando o primeiro método (de 64 bytes). Estes testes obtiveram êxito somente quando testados individualmente. Ao testar leituras no interior do carrinho com diversos produtos, não foi possível captar estas informações, pois somente foi possível detectar o item que estava mais próximo da antena, impossibilitando a leitura dos demais itens. Tendo em vista que para esta aplicação é necessária a leitura de todos os itens do carrinho, essa opção de leitura foi descartada. Para contornar essa situação outro teste foi realizado buscando identificar somente o *EPC* e com isso foi possível perceber que, executando dessa forma, o leitor captava todas as *tags* existentes no carrinho. Em função disto foi utilizado este método para realizar a leitura dos produtos.

A vantagem da utilização da leitura do *EPC* em relação à leitura dos dados da *tag*, se refere ao tamanho da informação captada pelo leitor. A leitura do *EPC* é mais rápida, pois possui 12 bytes de informação. Porém, a desvantagem é que neste formato é necessária a criação de um banco de dados, que requer uma estrutura de sistema um pouco maior. Utilizando a leitura dos dados diretamente das *tags*, todas as informações já estariam armazenadas na *tag* (nos 64 bytes), minimizando o tempo de filtragem necessário para contabilizar os itens no display do carrinho.

Para associar as *tags* aos seus respectivos produtos, utilizando a leitura do *EPC*, cada *tag* deve passar por um cadastramento, onde ao lado do seu número identificador vai existir as informações pertinentes aos produtos. Neste trabalho foram utilizadas no cadastro, as informações de descrição do

produto, preço, validade. Esse cadastro foi salvo em um arquivo txt e funciona como banco de dados. A Fig. 6 mostra o modelo do banco de dados. Este arquivo fica armazenado no cartão de memória *SD*, ao qual o Arduino tem acesso.

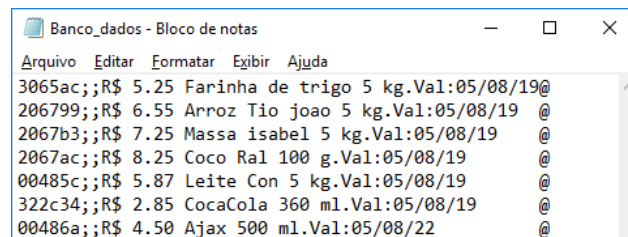


Fig. 6. Modelo do banco de dados.

E. Etiquetas

Para obter as leituras das etiquetas passivas (*tags*) de forma eficiente, foi necessário revestir o carrinho com material plástico. O carrinho utilizado possui dois compartimentos (superior e inferior), conforme é possível visualizar na Fig. 3 (b). Para realizar a leitura de todos os produtos é necessário alocar determinados produtos em partes diferentes do carrinho. A parte inferior foi destinada para os enlatados e líquidos, que são produtos que impedem a reflexão do sinal das *tags* para antena, impedindo a leitura de um produto que estiver atrás destes. A parte superior do carrinho ficou destinada para os demais produtos, cujos mesmos não interferem na leitura da *tag*. Ou seja, mesmo que os produtos estejam empilhados, o leitor identificará cada produto.

A decisão de dividir em grupos ocorreu com base nos testes realizados. Foi possível perceber que os líquidos e os enlatados, quando inseridos na parte superior do carrinho, impossibilitavam a leitura dos demais itens que ficavam com as *tags* não visíveis à antena. Em função disso, esses itens foram colocados na parte inferior, para um bom funcionamento do sistema. Ademais, todos os produtos da parte inferior não podem ter suas *tags* fixadas diretamente aos produtos em função da interferência, nem ter suas *tags* “escondidas”. Com isso foi adotado a utilização de cintas plásticas como suporte de fixação. Assim as *tags* são fixas nas cintas, conforme demonstrado o destaque na Fig 7, permitindo manter uma distância aceitável para que o leitor identifique o produto.



Fig. 7. *Tag* fixada na cinta para evitar interferências.

Nos demais produtos as *tags* são fixadas na própria embalagem e ficam dispostos na parte superior do carrinho, conforme mostra o destaque na Fig. 8.



Fig. 8. Tag fixada na própria embalagem do produto.

F. Aplicativo

O aplicativo foi desenvolvido com o objetivo de auxiliar o cliente a visualizar todos os itens inseridos no carrinho, dispondo na tela, o valor de cada mercadoria, descrição do item, bem como o valor total das compras. Inicialmente um aplicativo pronto e gratuito faria a demonstração das mercadorias contidas no carrinho e um teclado externo conectado ao Arduino faria a interface entre o cliente e o carrinho. Porém observou-se que seria possível criar um aplicativo através da plataforma *AppInventor*, obtendo-se assim um melhor aproveitamento do *tablet*. Com o desenvolvimento de um aplicativo dedicado para este trabalho, foi possível incluir no mesmo as funcionalidades do teclado, aprimoramento do projeto. Dessa forma, foi possível eliminar o hardware projetado para as teclas, tornando o trabalho otimizado.

O aplicativo possui dois ícones inicialmente (Conectar, Atualizar Compras) e estão dispostos na tela inicial como mostra a Fig. 9.

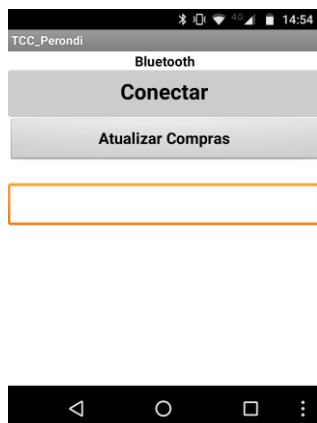


Fig. 9. Layout inicial de apresentação do aplicativo.

Ao clicar no ícone “Conectar” pela primeira vez, uma tela mostra as opções de dispositivos que estão disponíveis na rede, como mostra a Fig. 10.

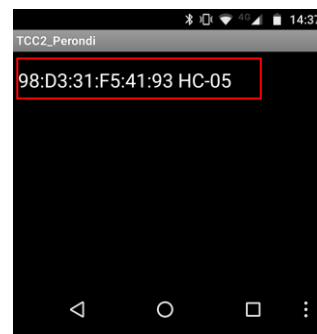


Fig. 10. Dispositivos disponíveis na rede.

O destaque feito na Fig. 10 corresponde a opção de dispositivos disponíveis para conexão. Após selecionado e estabelecido a conexão, o dispositivo fica memorizado, não necessitando mais adicionar a senha do dispositivo. É interessante ressaltar que esse procedimento não será feito pelo cliente, apenas pelo programador quando energizar o sistema. O cliente ao pegar o carrinho, verá na tela do *tablet* as opções “Atualizar Compras” e “Desconectar”, sendo que ele só utilizará o botão “Atualizar Compras” para ver a lista de compras, conforme Fig. 11.

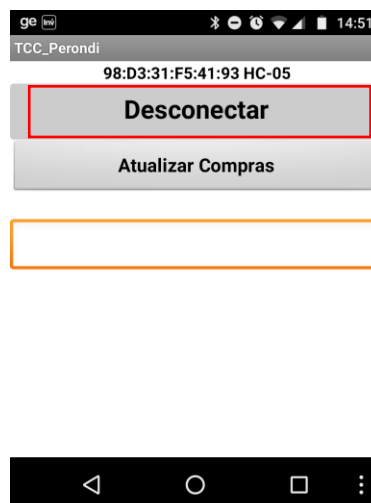


Fig. 11. Layout de apresentação do aplicativo para uso do cliente.

Ao pressionar no ícone “Atualizar Compras” um comando é enviado via *bluetooth* para o Arduino, que deverá apagar os arquivos salvos do cartão de memória referente a lista armazenada anteriormente. Após apagar, deverá gerar um novo arquivo com as compras atualizadas. Caso ocorra algum problema com o arquivo gerado ou qualquer outro erro, será emitido um aviso na tela solicitando a presença do gerente ou responsável pelo estabelecimento e não serão contabilizadas as compras, resultando um valor total de R\$ 0.00, como mostram os destaques da Fig. 12.

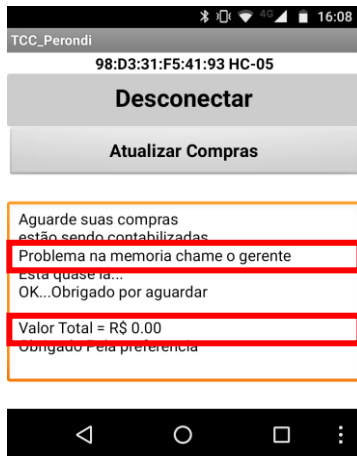


Fig. 12. Mensagens de erro.

Por fim, ao chegar até o caixa é necessário desconectar o carrinho (Arduino) do *tablet*, para que seja possível a conexão do carrinho com o computador para cobrança. Para isso, basta clicar no ícone “Desconectar” na Fig. 11.

No que se refere às facilidades do aplicativo do *tablet*, a plataforma *AppInventor* dispõe o aplicativo criado para *download* de duas maneiras. Uma das opções é o programador solicitar que um código *QR* (*Quick Response*) seja disponibilizado no site para a realização do *download* do aplicativo. Porém, após gerado o código *QR*, o usuário tem duas horas para fazer o *download*. Após as duas horas o código expira e o *QR* fica indisponível, necessitando que o programador peça uma nova solicitação. Outra opção é publicar o aplicativo, assim todos que acessam a *play store*, podem efetuar o *download*. Neste trabalho foi utilizado o *QR code* para *download*. Esta captura pode ser feita pela tela de um computador. A Fig. 13. Mostra detalhes do *QR code*.

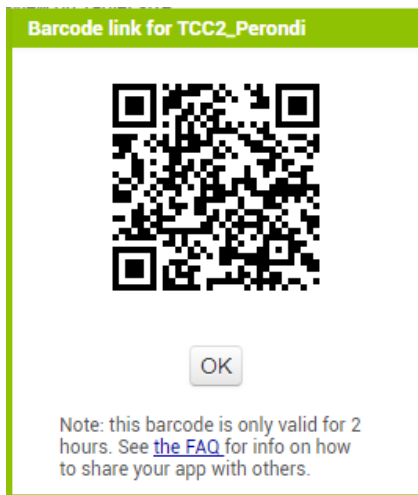


Fig. 13. Formato *QR* disponível para *download*.

G. Firmware

Para unir todas as funcionalidades propostas neste trabalho, foram desenvolvidos três *firmwares*. Um dos *firmwares* foi desenvolvido na plataforma *AppInventor* e é destinado

exclusivamente para dispositivos que contenham o sistema operacional Android, preferencialmente a versão 8.0. Estes sistemas operacionais são encontrados em *smartphones* e *tablets*. Neste projeto foi utilizado um *tablet*, em que a interface do *firmware* com o usuário está demonstrada na Fig. 11. A linguagem de programação é baseada em montagem de blocos como mostra a Fig. 14.

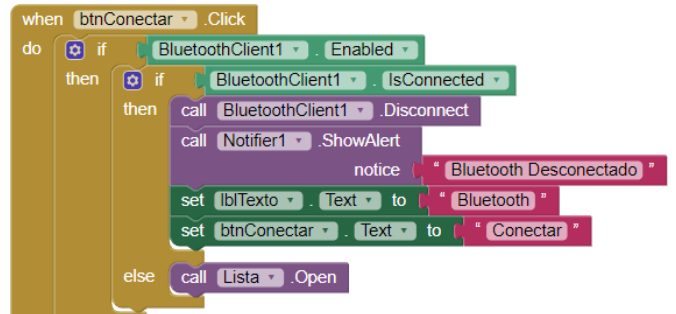


Fig. 14. Bloco de programação realizado no *AppInventor*.

Os demais *firmwares* desenvolvidos utilizam a plataforma de desenvolvimento específica para o dispositivo Arduino. Para realizar os programas dois e três deste trabalho, foi utilizada a linguagem de programação C / C++. O *firmware* dois foi desenvolvido para gerenciar todos os dados de filtragem, coleta de dados, armazenamento e envio dados externo via *bluetooth*. Com o intuito de mostrar as diferentes linguagens utilizadas no projeto, a Fig. 15 mostra um exemplo da linguagem de programação utilizada no Arduino, exigindo o código para filtrar todos os arquivos e que aparece em vários pontos do programa.

```

163 int orienta_leitura_dado2 = 0;
164 // for (int i = 0; i < (linhaDADOS + 1); i++) {
165 int conta05 = 0;
166 int N2 = 0;
167 while (N2 < linhafiltro) {
168     N2 ++;
169     orienta_leitura_dado2++;
170
171     if (conta05 != linhafiltro) {
172         conta05++;
173     }
174 }

```

Fig. 15. Fragmento da programação realizado na *IDE* do Arduino.

A *IDE* (*Integrated Development Environment*) do Arduino dispõe de um recurso chamado monitor serial. Esse recurso permite a visualização de dados enviados pela porta serial. Neste projeto foi desenvolvido o terceiro *firmware* para simular a função de caixa do supermercado, utilizando o serial monitor. Este *firmware* simula uma porta serial e permite que esta porta seja atrelada ao módulo *bluetooth* do computador. Assim, todos os dados recebidos pelo computador via *bluetooth* são mostrados na tela do monitor serial.

Como já mencionado o gerenciamento dos dados é feito com o Arduino. Porém para que todo processo de leitura

automatizada funcione, é necessário que o programador insira um arquivo de banco de dados no cartão de memória. Este arquivo deve conter em ordem, o endereço do *EPC*, valor do produto, breves descrições do produto e a data de validade. Este preenchimento deve obedecer ao número máximo de 51 caracteres disponível por linha de produto. Entretanto, deve-se manter inalterado o local destinado para o endereço *EPC* e a posição dos delimitadores. Também, o nome do arquivo deve ser Banco_dados.txt. A Fig. 16 mostra o cadastramento completo de um item, onde tem em destaque os 3 últimos bytes do *EPC* e os delimitadores.

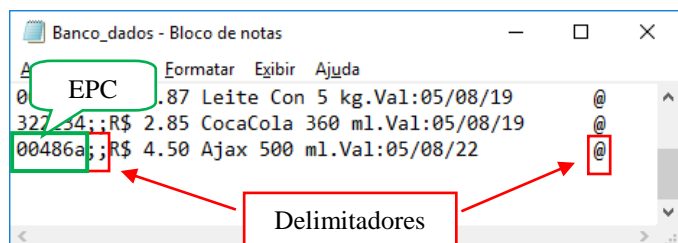


Fig. 16. Arquivo Banco_dados.txt.

O *firmware* foi organizado em funções e, para o bom funcionamento, deve seguir as instruções do diagrama da Fig. 2, principalmente no que se refere ao acomodamento correto dos produtos no carrinho, respeitando os limites do carrinho para evitar erro de leitura das *tags*.

Deste modo, ao acomodar todos os produtos selecionados dentro do carrinho, o cliente, ao posicionar o carrinho em um dos pontos de leitura dispostos no mercado, deve pressionar o ícone “Atualizar Compras” na tela do *tablet*. Com esta seleção, o aplicativo, apaga a lista anterior e envia um comando para que o Arduino inicie a leitura de *tags*. Com isso todas as listas de compras anteriores são deletadas recriando assim uma nova lista. Sempre que uma nova lista é criada todas as variáveis são resetadas para evitar problemas de armazenamento indevido.

Para efetuar a leitura das *tags*, é utilizada uma biblioteca disponibilizada pelo fabricante do leitor, chamada <SparkFun_UHF_RFIDReader.h>. Esta biblioteca tem a função de capturar as informações de 12 bytes referentes ao *EPC* emitidos pelas *tags*. Neste trabalho foi desenvolvido uma função para truncar o *EPC* em 3 bytes, conforme destaque na Fig. 16. Com isso é possível minimizar o tempo de filtragem do *firmware*. A escolha da utilização de 3 bytes de *EPC* teve como intuito a redução do tempo de leitura, porém implica na redução de itens totais cadastrados no supermercado.

Esta primeira triagem feita por uma das funções do *firmware* é salva em um arquivo chamado DADOSRFID.txt no cartão de memória *SD* no qual o Arduino tem acesso. O limite do cartão de memória é de 4 GB.

Para demonstração deste projeto foi utilizado 30 leituras de *tags*. Estas leituras podem identificar várias vezes a presença de uma *tag*, salvando os mesmos produtos várias vezes. A Fig. 17, destaca uma *tag* que foi lida 4 vezes e salva no arquivo DADOSRFID.txt. Nessa mesma Fig. 17, percebe-se o formato reduzido de 3 bytes do *EPC*.

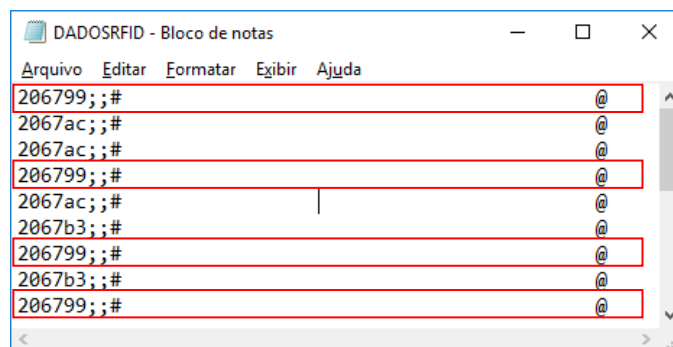


Fig. 17. Arquivo DADOSRFID.txt gerado sem filtros.

Após esta primeira triagem, é possível perceber dois arquivos salvos no *SD card*: o arquivo de banco de dados chamado Banco_dados.txt e o arquivo DADOSRFID.txt.

Para evitar que existam repetições dos itens, uma varredura foi feita no arquivo DADOSRFID.txt. Foi elaborado um *loop* de comparações entre os endereços de *EPCs* salvos no arquivo DADOSRFID.txt. O resultado das comparações é salvo em um novo arquivo chamado FILTRO.txt. Neste arquivo é possível perceber que somente uma amostra de cada *tag* foi salva, evitando que os produtos sejam somados duas vezes. A Fig. 18 mostra o arquivo FILTRO.txt e os *EPCs* filtrados (sem repetir).

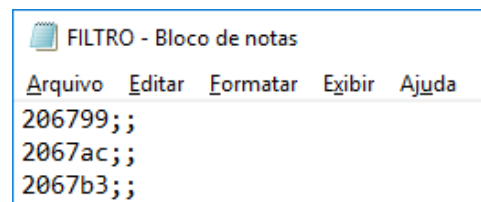


Fig. 18. Arquivo FILTRO.txt e os *EPCs* filtrados.

A última etapa do programa do segundo *firmware* compara todos os endereços salvos no arquivo FILTRO.txt com o arquivo Banco_dados.txt. Com isso é gerado o quarto arquivo chamado Lista_Compras.txt. Este último arquivo deverá conter todos os itens pretendidos pelo cliente e o valor total da compra. A Fig. 19 mostra um exemplo de uma lista final de compras com a descrição dos itens e o valor total da compra.

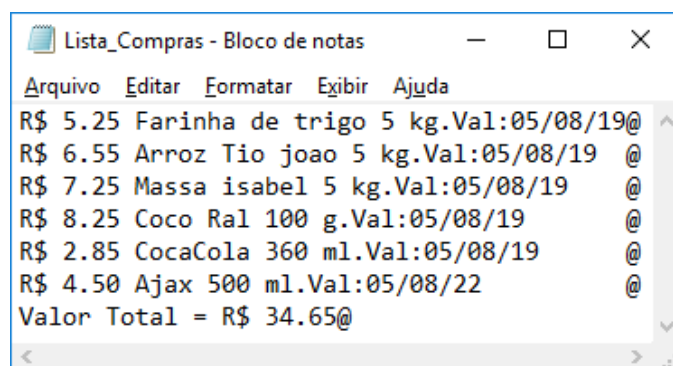


Fig. 19. Modelo da lista final.

O arquivo Lista_Compras.txt, fica armazenado na memória *SD card* e, no momento de efetuar a formalização da compra no caixa, este é transmitido via *bluetooth* para o computador.

Para efetuar a transmissão, inicialmente, é necessário um procedimento de desconexão entre o Arduino e o *tablet*. Para desconectar basta pressionar o ícone “Desconectar” no *tablet*. A Fig. 20, apresenta a lista final de compras que aparece para o cliente e o ícone “Desconectar” como opção a ser selecionada.

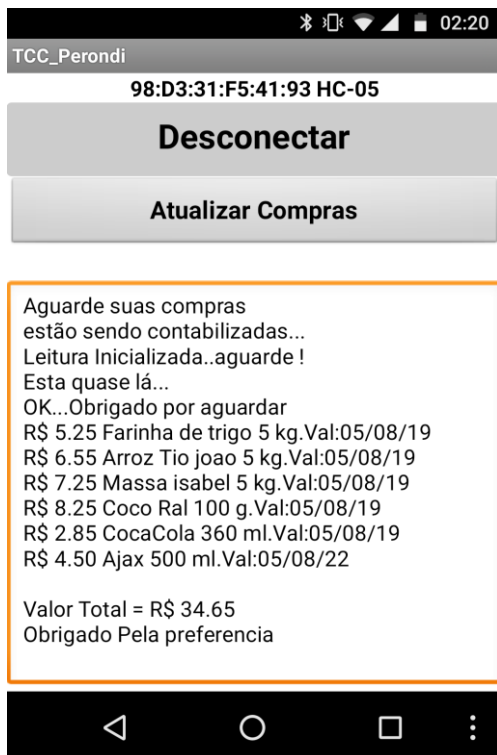


Fig. 20. Lista de compras final apresentada para o cliente.

Para efetuar o procedimento de conexão do carrinho com o computador do caixa é utilizado o terceiro *firmware*. Este *firmware* utiliza a rede *bluetooth* do computador e foi desenvolvido especificamente para capturar o arquivo *Lista_Compras.txt* do carrinho.

Para acessar o dispositivo *bluetooth* do carrinho pela primeira vez, para transmissão dos dados, foi necessário ir nas configurações do computador (na aba *bluetooth*) e outros dispositivos, adicionar um novo dispositivo, selecionar o dispositivo HC-05, inserir a senha 1234 e conectar. Ao efetuar este procedimento de conexão uma porta serial virtual é criada. Para identificar a porta deve-se acessar o painel de controle do computador, exibir impressoras e dispositivos, clicar com o botão direito do mouse em HC-05, propriedades e *hardware*. Com isso será possível visualizar a porta serial que está sendo utilizada para a comunicação *bluetooth* do computador com o modulo *bluetooth* do carrinho de acordo com a Fig. 21. As configurações para a conexão normalmente ficam memorizadas pelo computador. Logo, após, realizada a comunicação pela primeira vez, somente será necessário refazer este procedimento caso ocorra algum problema para estabelecer a conexão.

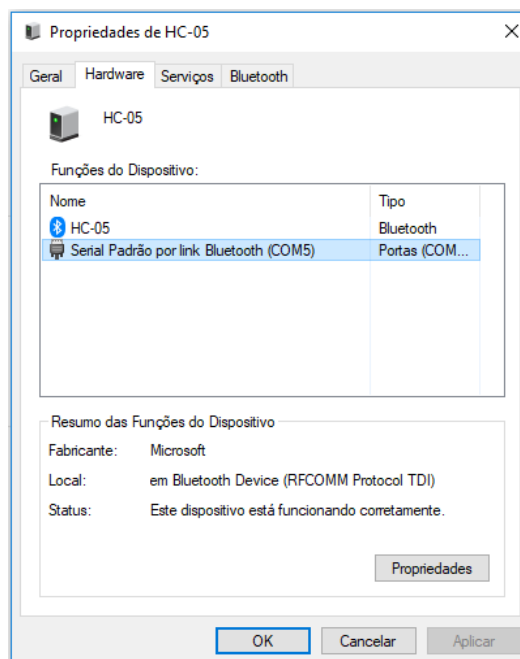


Fig. 21. Comunicação *bluetooth* do computador com o modulo *bluetooth* do carrinho.

Dessa maneira, o operador do caixa deve selecionar esta mesma porta serial na *IDE* do Arduino conforme destaque na Fig. 22. Assim no momento que o operador abrir o serial monitor, ocorre automaticamente a conexão entre o carrinho e o computador do caixa. Quando o serial monitor estiver aberto na tela do computador, basta o funcionário do caixa digitar a letra “t” e pressionar “Enter”. Desta maneira, toda a lista armazenada na memória do carrinho é transmitida para o computador, permitindo a geração da nota fiscal e a finalização das compras. A Fig. 23 mostra a tela do monitor serial, com a transferência da lista de compras que aparece na tela do computador do caixa, após o operador pressionar a letra “t”.

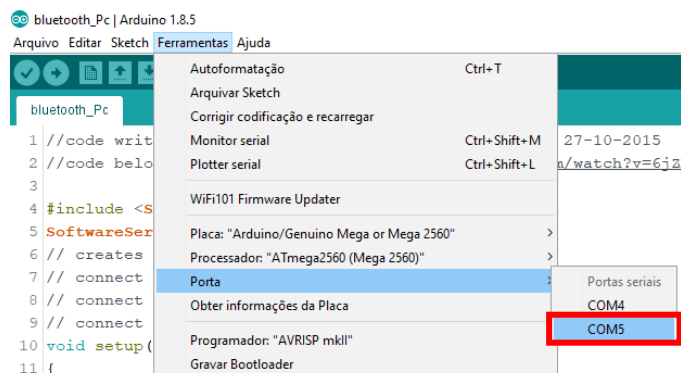


Fig. 22. Tela de seleção da porta serial na *IDE* do Arduino.

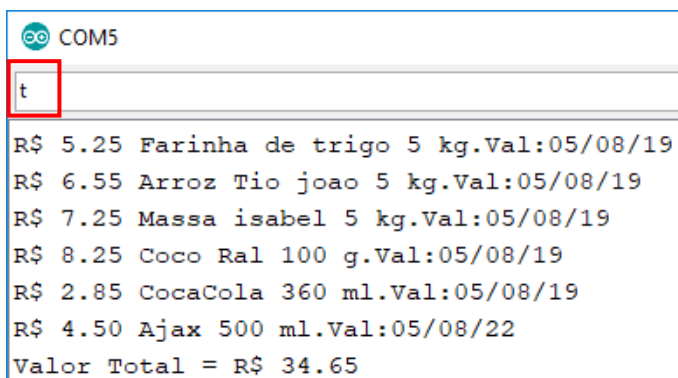


Fig. 23. Tela do monitor serial com a transferência da lista de compras.

Todo o processo de leitura das *tags* e transmissão via *bluetooth* está demonstrado no diagrama da Fig. 24.

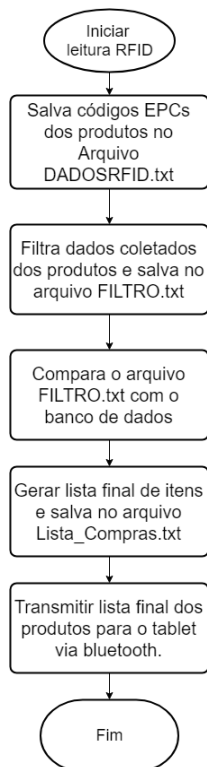


Fig. 24. Processo de leitura das *tags* e transmissão via *bluetooth*.

H. Simulações.

Para fins de comparação foi observado e cronometrado, dentro de um supermercado, o tempo médio de cada item lido por um operador de caixa tradicional (com leitura de código de barras). A ação considerada foi desde o cliente retirar o produto do carrinho e colocar em cima da esteira até o produto passar pelo leitor de código de barras pelo operador. Este processo teve em média 4 segundos para cada item. O sistema *RFID* proposto neste artigo necessita, em média, de 7 segundos para cada item lido, sendo este valor proporcional ao número de produtos. Dessa maneira, caso o supermercado

possua vários pontos de leitura, o cliente, realiza a atualização de compras nesses pontos e se dirige ao caixa apenas para efetuar o pagamento. Neste contexto, a fila nos caixas será reduzida em comparação ao modo tradicional. Caso o supermercado não possua muitos pontos de leitura e o cliente necessite realizar a atualização das compras no caixa, provavelmente ocorrerá a formação de filas devido à demora na leitura dos itens.

V. RESULTADOS

Após a realização de testes de cada etapa do trabalho, algumas melhorias foram sendo implementadas, como: seleção do tipo de *tag* a ser usada; teste de leitura das *tags*; posicionamento e suporte da antena; revestimento do carrinho; local de acomodação dos produtos para a leitura correta; e implementação do *tablet* como interface com o cliente.

No início deste trabalho havia algumas dúvidas referentes ao bom funcionamento das *tags* adquiridas, visto que foram utilizados dois modelos com dimensões diferentes. No entanto comprovou-se através de testes que ambas funcionaram de forma satisfatória nas mesmas condições. Conforme indicado pelo fabricante as *tags* inibem o sinal de reflexão quando em contato com metais e líquidos. Em função disso foi desenvolvido um modo de fixação alternativa das *tags* para alguns produtos, onde as mesmas ficam dispostas nos itens de forma suspensa, ou seja, não entram em contato com fontes de interferências (metais, líquidos). A Fig. 7 exemplifica um dos modelos adotados, onde é possível observar que uma cinta plástica afasta a *tag* da lata de refrigerante, evitando o contato com o metal e garantido a reflexão do sinal para a antena.

Um dos desafios deste trabalho foi o posicionamento da antena, pois tendo em vista que o ambiente da aplicação se tratava de um carrinho metálico, diversos testes de posicionamento tiveram que ser realizados a fim de localizar a posição aonde seria possível a identificação dos produtos na parte superior e inferior do carrinho. A Fig. 3 (a) e a Fig. 3 (b) demonstram respectivamente o teste da posição válida e a posição final da antena. Outro fator envolvendo a antena é o alcance. A ideia inicial era efetuar a leitura das *tags* em qualquer lugar de um supermercado, porém à medida que os testes foram sendo realizados, observou-se que esta opção não seria viável para este trabalho em função da estrutura metálica utilizada no compartimento superior do carrinho impedir a propagação do sinal *RF*, necessitando um estudo mais aprofundado sobre o tema. Para contornar essa situação a proposta baseou-se em criar pontos específicos dentro do supermercado para a realização da leitura e atualização da lista de compras, evitando a proximidade com prateleiras e gôndolas.

Diversos testes foram feitos com relação à leitura de dados diretamente coletada das *tags* (modelo de leitura de 64 bits), porém, os resultados não foram satisfatórios, pois a expectativa inicial era a possibilidade de ler 150 *tags* por segundo no formato de dados. No entanto, não foi possível atingir estes parâmetros de leitura. Uma leitura satisfatória só foi atingida quando realizado a leitura do *EPC*. A solução acabou sendo utilizar o *EPC* e criar um banco de dados para comparar os produtos cadastrados.

No que se refere ao tempo necessário para o processo de leitura, foi observado que a leitura dos itens através do *EPC* proporciona maior rapidez na aquisição dos dados. Porém, para que os dados coletados de fato representem os itens contidos no carrinho, é necessária a aplicação de filtros. Com isso o processo acaba ficando mais lento. Em função disso para demonstrar o funcionamento do trabalho, optou-se por utilizar poucos itens contidos no carrinho, evitando um período longo de processamento.

Em relação ao projeto inicial algumas melhorias puderam ser realizadas. Uma delas foi a substituição do teclado matricial por ícones inseridos no aplicativo. Este teclado inicialmente seria físico, através de um hardware exclusivo para as teclas, porém com o andamento do trabalho observou-se a oportunidade de inserir as funcionalidades deste teclado em um aplicativo. Com isso o cliente poderá selecionar todas as funcionalidades somente utilizando o *tablet*.

Embora os resultados tenham mostrado um bom funcionamento do sistema, alguns fatores poderiam ser melhorados, como, por exemplo: para minimizar problemas referentes à leitura das *tags*, poderia se utilizar um carrinho com revestimento em polímero. Este simples fator possibilitaria um diferente posicionamento da antena de forma a tornar o designer do carrinho mais otimizado, minimizando custos com o projeto que está estimado em R\$ 2.500,00. O tempo de filtragem e atualização dos produtos é um item que demanda novas pesquisas e pode ser melhorado na implementação de novos trabalhos. Um dos fatores para reduzir o tempo de leitura é a utilização de um microcontrolador com maior capacidade de processamento.

Para validação deste trabalho optou-se por utilizar até 10 itens no carrinho. A inclusão de mais itens no carrinho pode acarretar em um tempo de atualização maior. Com tudo, os resultados do trabalho foram satisfatórios e foi possível aplicar o conceito de *RFID* na identificação dos produtos contidos em um carrinho de compras, podendo simular situações existentes nos supermercados.

VI. CONCLUSÃO

A tecnologia *RFID* vem se tornando cada vez mais aliada no ramo de varejo. Esta tecnologia que utiliza a radiofrequência para capturar dados foi criada há muitas décadas, porém seu uso na cadeia produtiva, suprimentos e de sistemas de identificação é bem recente.

Esta tecnologia ascendente foi utilizada neste trabalho com a intenção de tornar o ato de fazer compras mais agradável. O principal objetivo do trabalho foi otimizar o tempo de compras e das filas no ato do pagamento nos supermercados através do uso da tecnologia *RFID*.

Os resultados deste estudo foram muito satisfatórios, pois foi possível atingir o objetivo de identificar produtos através da rádio frequência. Com os testes realizados referentes à seleção do tipo de *tag* a ser usada, ao teste de leitura das *tags*, posicionamento e suporte da antena, ao revestimento do carrinho, ao local de acomodação dos produtos para a leitura correta e à implementação do *tablet* como interface com o cliente, foi necessário mudar alguns itens do projeto inicial

para facilitar o desenvolvimento do sistema. Assim, com a verificação e validação dos testes, o trabalho atendeu as expectativas propostas e se mostrou uma alternativa útil para ajudar os supermercados a facilitar o processo de compras dos clientes.

O sistema utilizado deve ser aperfeiçoado para que se torne viável implementar nos supermercados. Se o sistema for modificado para efetuar a leitura em qualquer ponto do supermercado, e não especificamente nos pontos de leitura, minimizará o tempo que hoje é perdido nas filas dos caixas, de modo que o cliente somente fará o pagamento no caixa, pois as leituras dos itens deverão ser feitas antes de chegar até o caixa.

Os custos com o protótipo deste projeto estão estimados em R\$ 2.500,00. Tendo em vista que o custo de um carrinho de supermercado tradicional é R\$ 300,00, a curto prazo a implantação deste projeto em um supermercado não é viável, porém, levando em consideração algumas melhorias no carrinho e no aplicativo a proposta do projeto pode se tornar atraente para grandes redes de supermercados.

Referências

- [1] J. Rinaldi, R. Moranbito, V. Tachibana, "A importância da rapidez de atendimento em supermercados: um estudo de caso.", São Carlos, Brasil, 2009.
- [2] D.P. Luiz Fernando; "Teoria de filas- Analisando o fluxo de atendimento e o número de atendentes em um supermercado", UFRGS, Porto Alegre, Brasil, 2010.
- [3] P. Chandrasekar, T. Sangeetha, "Smart shopping cart with automatic billing system through RFID and zigbee", *Information Communication and Embedded Systems (ICICES) 2014 International Conference on*, pp. 1-4, 2014.
- [4] K. Prasiddhi, Dhanashri H. Gawali, "Innovative Shopping Cart For Smart Cities", *Recent Trends in Electronics Information & Communication Technology (RTEICT) 2017 2nd IEEE International Conference on*, pp.1067-1071. 2017.
- [5] A. Karmouche, Y. Salih-Alj, J. Abrache, "Distributed aisle-level scanning approach for RFID shopping systems", *2014 International Conference on Logistics and Operations Management (GOL'14)*, pp. 1-7, Jun. 2014.
- [6] S.G. Rodrigo, "Desenvolvimento de uma aplicação frente de caixa automatizada para o varejo utilizando Java e *RFID*", Maringá- PR, Brasil, 2005.
- [7] Datasheet Arduino Mega. Disponível em: <http://ww1.microchip.com/downloads/en/icedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf>. Acesso em: 11 nov. 2018.
- [8] Datasheet módulo *SDcard*. Disponível em: <http://www.convict.lu/pdf/ProdManual_SDCardv1.9.pdf>. Acesso em: 11 nov. 2018.
- [9] Datasheet módulo *bluetooth*. Disponível em: <<https://www.gme.cz/data/attachments/dsh.772-148.1.pdf>>. Acesso em: 11 nov. 2018.
- [10] Datasheet leitor *RFID SRTR*. Disponível em: <<https://cdn.sparkfun.com/datasheets/Sensors/ID/NANO-final-12Mar2015.pdf>>. Acesso em: 11 nov. 2018.