



UNIVERSIDADE ESTADUAL DE GOIÁS - UEG
CÂMPUS ITABERAÍ
CURSO DE SISTEMAS DE INFORMAÇÃO

**PROJETO DE UM SISTEMA COMERCIAL DISTRIBUÍDO BASEADO EM
DISPOSITIVOS EMBARCADOS**

Aguinaldo Gabriel dos Reis e Silva

Luciano Pinto e Silva
Orientador

Novembro de 2017
Itaberaí - GO

Aguinaldo Gabriel dos Reis e Silva

PROJETO DE UM SISTEMA COMERCIAL DISTRIBUÍDO BASEADO EM
DISPOSITIVOS EMBARCADOS

Trabalho de conclusão de curso
apresentado ao curso de Bacharelado em
Sistemas de Informação da Universidade
Estadual de Goiás – Câmpus Itaberaí sob
a orientação do professor Luciano Pinto e
Silva.

Novembro de 2017

Itaberaí – GO

“Science is a differential equation. Religion is a boundary condition.”

Alan Turing

Homenagens póstumas à Nelci Evaristo
Barbosa, avó materna de Aginaldo
Gabriel dos Reis e Silva.

RESUMO

O uso de dispositivos embarcados é algo crescente nos últimos anos devido à sua viabilidade econômica e criação de novos projetos. Este trabalho tem por finalidade demonstrar o desenvolvimento de um sistema comercial para controle de estoque de uma organização, utilizando o Raspberry Pi como servidor podendo ser utilizado mais um dispositivo compartilhando informações. O desenvolvimento deste projeto estimulou diversas áreas de TI passando por arquitetura, engenharia, redes, sistemas distribuídos e programação, agregando valor tanto técnico quanto prático.

Palavras-chave: dispositivos embarcados, sistema comercial, Raspberry Pi, sistemas distribuídos.

ABSTRACT

The use of embedded devices is something growing in recent years due to its economic viability and creating new projects. This work aims to demonstrate the development of a commercial system to control inventory of an organization, using the Raspberry Pi as a Web server. The development of this project has instigated diverse IT areas through computer architecture, requirements engineering, computer networks, distributed systems and programming, adding both technical and practical value.

Keywords: embedded devices, commercial system, Raspberry Pi, distributed systems.

LISTA DE ABREVIATURAS, SIGLAS OU TERMOS OPERACIONAIS

PHP	Hypertext Preprocessor
TI	Tecnologia da Informação
WEB (WWW)	World Wide Web (Internet)

LISTA DE FIGURAS

Figura 1: Modelo Espiral	20
Figura 2: Ilustração do Raspberry Pi 2 Model B+	21
Figura 3: Interface do Ubuntu MATE em um Raspberry Pi	22
Figura 4: Linguagens de consulta MySQL e MongoDB	24
Figura 5: Coleção Parceiro	32
Figura 6: Coleção Entrada	32
Figura 7: Coleção Saída	32
Figura 8: Tela de login no sistema	33
Figura 9: Tela de registros de entradas	33
Figura 10: Tela de registros de produtos	34
Figura 11: Tela de consulta de parceiros	34

LISTA DE QUADROS

Quadro 1: Recursos suportados entre o MySQL e o MongoDB	24
Quadro 2: Cronograma de desenvolvimento do trabalho	35

LISTA DE TABELAS

Tabela 1: Lista de requisitos funcionais	27
Tabela 2: Lista de requisitos não funcionais	28
Tabela 3: Objeto Parceiro	28
Tabela 4: Objeto E-mail	30
Tabela 5: Objeto Endereço	30
Tabela 6: Objeto Telefone	30
Tabela 7: Objeto Entrada	30
Tabela 8: Objeto Saída	31
Tabela 9: Objeto Produto	31

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Contextualização	12
1.2 Delimitação do Assunto	13
1.3 Justificativa	13
1.4 Objetivos	13
1.4.1 Objetivo Geral	14
1.4.2 Objetivos Específicos	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 Sistemas de controle de estoque de empresas de pequeno porte	15
2.2 Sistema de Informação de apoio à operação	16
2.3 Engenharia de Software	18
2.4 Engenharia de Requisitos	18
2.5 Análise Orientada à Objetos	18
2.6 Metodologia de Desenvolvimento	19
3 TECNOLOGIAS UTILIZADAS	21
3.1 Raspberry Pi	21
3.2 Ubuntu MATE	22
3.3 Nginx	23
3.4 MongoDB	23
3.5 PHP	25
3.6 Golang	25
4 LEVANTAMENTO DO SISTEMA	27
4.1 Levantamento de Requisitos	27
4.1.1 Requisitos Funcionais	27
4.1.2 Requisitos Não Funcionais	28
4.2 Objetos identificados	28
4.3 Diagrama de Coleções do MongoDB	31
5 PROTOTIPAÇÃO	33
6 CRONOGRAMA	35
7 CONCLUSÕES	36
8 REFERÊNCIAS BIBLIOGRÁFICAS	37

APÊNDICES	39
APÊNCICE A	40

1 INTRODUÇÃO

Neste capítulo aborda-se uma perspectiva inicial do trabalho passando pela sua contextualização, delimitação do assunto, justificativa e chegando aos seus objetivos.

1.1 Contextualização

No momento atual é bastante perceptível as frequentes alterações no cenário econômico brasileiro e é fato que, mesmo em diferentes nichos de mercado, existe a necessidade frequente de evolução das atividades gerenciais de uma organização. Descomplicar tais atividades e ainda agregar produtividade e confiabilidade proporciona nítidas vantagens operacionais principalmente à micro e pequenas empresas onde, no estado de Goiás, elas somam mais de 200 mil ou 99,1% das empresas presentes no estado (SEBRAE/GO, 2013).

Ainda segundo o SEBRAE (Serviço Brasileiro de Apoio às Micro e Pequenas Empresas), as falhas no gerenciamento do negócio é uma das principais razões para o alto índice de falência nos primeiros anos de vida da empresa.

À par desta informação ou não, muitos empreendimentos são iniciados sem considerar a possibilidade de adoção de sistemas comerciais sendo estes muitas vezes considerados como um investimento desnecessário ou ainda como um custo ao negócio, devido boa parte das opções de o mercado contar com preço elevado, seja ele software ou hardware.

Outro motivo que leva o empreendedor não utilizar qualquer meio eletrônico ou no máximo fazer uso de planilhas eletrônicas para armazenar as informações, é ineficácia dos sistemas, tanto por não suprirem as necessidades do negócio, tanto por conterem uma curva de aprendizado bastante íngreme para boa parte de seus utilizadores.

Neste sentido, foi identificado uma oportunidade de negócio visando criar um modelo eficaz e de baixo custo para diversos tipos de empreendimentos tendo como mola propulsora a experiência obtida através dos anos com armazéns de alto fluxo para trazendo uma visão simplificada dos processos envolvidos.

1.2 Delimitação do Assunto

Este trabalho se restringe em apresentar uma alternativa para gerenciamento de um negócio do ponto de vista comercial, com gestão de vendas, compras, estoque e parceiros.

A aplicação será hospedada em dispositivo embarcado, o Raspberry Pi, com a possibilidade de se utilizar vários dispositivos em um mesmo ambiente, fazendo uso do conceito de sistemas distribuídos, e/ou ainda utilizar a Internet como meio e conectar filiais de uma mesma empresa à um único e conciso sistema.

1.3 Justificativa

Antes de tudo, há um grande entusiasmo por parte do autor em amplificar o conhecimento e habilidades para sistemas Web, tanto por ser uma área em evidência nos últimos tempos, tanto por ser propícia à mais diferentes soluções tecnológicas.

O autor também conta com conhecimento em diferentes setores de uma empresa e as aplicações comerciais utilizadas. Através disto foi proposto demonstrar uma solução muito interessante para desenvolvimento de sistemas robustos de alta escalabilidade utilizando hardware de tamanho reduzido e sem a utilização de tecnologia proprietária, tornando-o financeiramente bastante viável. Assim é pretendido a popularização da área de TI, em suas várias formas, no apoio à gestão comercial.

Outra justificativa é a conclusão do curso de Bacharelado em Sistemas de Informação pela Universidade Estadual de Goiás - Unidade Itaberaí.

1.4 Objetivos

A seguir o objetivo geral e específicos relatados neste trabalho de conclusão de curso.

1.4.1 Objetivo Geral

Analisar e desenvolver um modelo de um sistema comercial distribuído baseado em dispositivos embarcados.

1.4.2 Objetivos Específicos

Afim de atingir o objetivo geral, os seguintes objetivos específicos foram estabelecidos:

- Entender e desenvolver aplicações para dispositivos embarcados;
- Desenvolver um sistema comercial com interface eficiente e intuitiva;
- Otimizar a organização física dos produtos;
- Quantificar entrada e saída de produtos;
- Oferecer uma visão geral de toda a empresa;
- Possibilitar melhorias no atendimento ao cliente.
- Produzir relatórios precisos e condizente com a necessidade da empresa;

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata dos conceitos utilizados para a concepção do trabalho.

2.1 Sistemas de controle de estoque de empresas de pequeno porte

Dentro dos estoques, é necessário um preciso controle, de forma que esteja sempre organizado e planejado, sendo fundamental a implantação de um sistema que controle a metodologia e os procedimentos e rotinas a serem efetuadas. Braido e Martens (2013) citam Rezende (2003)¹ para elucidar o gerenciamento das informações: “qualquer sistema que manipule e gere informação, independente de usar recursos tecnológicos, pode ser considerado um sistema de informação, e o seu objetivo deve ser de auxiliar no processo de tomada de decisão nas organizações”.

De acordo com Slack (2002), deve ser levado em conta os custos do estoque, através dos custos de colocação do pedido, desconto de preços, da falta de estoque, de capital de giro, de armazenagem, de obsolescência e de ineficiência de produção. Dessa forma, ensina que:

O estoque é usualmente gerenciado por meio de sistemas de informação computadorizados sofisticados, que têm algumas funções: de maneira mais importante, a atualização dos registros do estoque, a geração de pedidos, a geração de relatórios de *status* de estoque e a previsão de demanda. (SLACK, 2002, p. 408)

Nesse sentido, para Costa (2012):

Nas organizações voltadas às novas técnicas de gerenciamento e controle do estoque, a utilização de recursos de automação é muito importante, pois otimiza os processos de forma a integrar as operações da empresa racionalizando tarefas, eliminando margens de erros gerando maior confiabilidade e, sobretudo, reduzindo consideravelmente o tempo e os custos operacionais e administrativos. A automação também integra o controle do estoque às diversas áreas da empresa, tais como: contabilidade, financeira, vendas e compras. (COSTA, 2012, p. 25)

Assim, os processos de automação além de melhorar a relação entre clientes e fornecedores, tornam o processo de compra e venda de materiais mais rápida e eficiente, o que envolve contas muito complexas, sendo bastante útil o uso de computadores. Os sistemas de informação de estoque incluem a atualização dos registros de estoque quando ocorre qualquer tipo de transação ou movimento de um material, a função da geração de pedidos e de registros de estoque gerando relatórios para o controle do armazenamento e a previsão de demandas (SLACK, 2012, p. 406-407).

Com o controle de estoque informatizado, compõe-se um banco de dados que reúne vários cadastros interligados de grupos e classes, materiais, fornecedores, operações, clientes e movimentos. O autor aponta que esses cadastros substituem muito bem as “antigas fichas de controle de estoque”, e acabou por extingui-los, devido as vantagens da integração de outros sistemas, o uso de vários setores simultaneamente e a diversidade de módulos de acordo com a operação pretendida (COSTA, 2012).

Para Manter as prioridades de estoque, Slack (2012) aponta o sistema “ABC” que divide os itens de acordo com sua taxa de movimentação e valor.

2.2 Sistema de Informação de apoio à operação

De acordo com o Professor Damasco (2016), os sistemas de informação de apoio à operação têm o papel de processar transações de forma eficiente, controlar os processos industriais e apoiar os processos de colaboração e comunicação. São divididos em Sistema de Processamento de Transações (SPT), Sistemas de Controle de Processos (SCP) e Sistemas Colaborativos.

São divididos em Sistemas de Processamento de Transações aplicam os conceitos e tecnologia em transações comuns no negócio através da coleta, manipulação e armazenamento de dados e produção de documentos. Hoje em dia, essa tecnologia já é realizada em tempo real. Processam os dados com uma variedade de documentos, mantém a precisão e a integridade dos dados, são céleres, práticos e aumentam a eficiência do trabalho por diminuir o desgaste para os funcionários. (DOS SANTOS, 2008).

Os Sistemas de Controle de Processos coordenam o processo continuamente durante sua elaboração. De acordo com Slack:

O controle estatístico de processo preocupa-se com checar um produto ou serviço durante sua criação. Se há razões para acreditar que há um problema com o processo, ele pode ser interrompido (onde é possível e adequado) e os problemas podem ser identificados e retificados. (SLACK, 2012, p. 564)

O SCP utiliza as amostras de longos períodos para formular gráficos de controle avaliando o desempenho dos processos, averiguando as variações de qualidade, de aceitabilidade de suas variações e suas causas. A esse processo, pode ser adicionado o estabelecimento de limites de controle, usualmente na média de desvio padrão de 0,3%, além de gráficos de controle para atributos e para variáveis. Hoje em dia, esse controle é visto como estratégico na produção para ganhar vantagem competitiva no mercado (SLACK. 2012).

Já os Sistemas Colaborativos, de acordo com Oliveira (2006), são utilizados para facilitar os trabalhos em grupo, promovendo o suporte e a colaboração entre os usuários. São classificados de acordo com o lugar e o tempo das interações. Para Mora e Felipe (2009), são responsáveis por gerenciar as informações relevantes dentro da equipe através da comunicação, coordenação e cooperação.

Dessa forma, elucidam os autores que:

Sua principal meta é permitir o trabalho em conjunto de maneira mais fácil e eficaz, ajudando a colaboração entre os indivíduos envolvidos em um processo, possibilitando que as pessoas envolvidas no projeto tenham uma visão geral do trabalho, permitindo um entendimento compartilhado sobre o andamento das tarefas ou de todo o trabalho. Eles devem gerenciar as informações relevantes disponibilizando à equipe do projeto as informações no momento apropriado, reduzindo os impactos causados pela fragmentação dos trabalhos. (MORA E FELIPE, 2008, p. 2)

Assim, contribuem para melhorar os processos operacionais através do compartilhamento de informações de tarefas cumpridas de forma coletiva, atendendo as demandas mundiais em que a divisão do trabalho é multidisciplinar e sua execução não ocorre necessariamente em tempo simultâneo.

2.3 Engenharia de Software

A Engenharia de Software é "a criação e utilização de sólidos princípios da Engenharia a fim de obter software de maneira econômica, que seja confiável para trabalhar eficientemente em máquinas reais" (PRESSMAN, 2006). Ela se utiliza de da matemática, da ciência da computação (SEEK, 2004) e da engenharia para solucionar problemas no domínio da aplicação (SOMMERVILLE, 2006).

Com foco na qualidade, a Engenharia de Software é uma "tecnologia em camadas" PRESSMAN (1995) onde, acima da camada da qualidade, se encontra os processos, os métodos e as ferramentas, nessa ordem.

Para apoiar o processo de desenvolvimento existem diversas metodologias que foram criadas com o passar do tempo e, apesar de que nenhuma pode ser considerada o modelo ideal, todos abrangem a análise de requisitos, projeto, testes e manutenção (SOMMERVILLE, 2003), fundamentais para atingir a qualidade de software.

Atualmente as metodologias ágeis são as mais adotadas pelas empresas por sua estrutura mais prática e também para atender as demandas do mercado.

2.4 Engenharia de Requisitos

A Engenharia de Requisitos "é o ramo da Engenharia de Software que está preocupado com os objetivos do mundo real para as funções e restrições aplicáveis a sistemas de software" (ZAVE, 1997).

É nesta etapa que acontece a comunicação com o cliente, onde se identifica as tarefas no qual o *software* deverá realizar dentro do negócio. Dessa forma é conduzido o emprego de técnicas sistemáticas afim de gerar requisitos completos e consistentes uma vez que erros poderão impactar no prazos e custos da aplicação.

2.5 Análise Orientada à Objetos

A Análise Orientada à Objetos (AOO) é um paradigma que surgiu na década de 1970 visando ser uma alternativa à Análise Estruturada. Baseia-se em conceitos

que nos fora ensinado desde crianças como objetos e atributos, agrupando objetos de características semelhantes em classes. Esta tipo de análise nos permite entender melhor a aplicação e o ambiente no qual ela está inserida podendo se aproximar o máximo possível do modelo do mundo real.

Para Correia (2006), "as pessoas que usam os objetos não precisam se preocupar em saber como eles são constituídos internamente acelerando o tempo de desenvolvimento.

Sommerville (2007) acredita que "análise orientada a objetos concentra-se no desenvolvimento de um modelo orientado a objetos do domínio da aplicação. Os objetos nesse modelo refletem as entidades e as operações associadas ao problema a ser resolvido".

2.6 Metodologia de Desenvolvimento

Para a metodologia de desenvolvido a ser seguida no trabalho foi designado o modelo espiral como sendo o ideal devido à sua flexibilidade e revisão dos requisitos.

O ciclo de vida em espiral permite que os requisitos sejam definidos progressivamente, e apresenta alta flexibilidade e visibilidade. Esse modelo permite que, em pontos bem-definidos, os usuários possam avaliar partes do produto e fornecer realimentação quanto as decisões tomadas (Wilson de Pádua Paula Filho, 2011, pg.94).

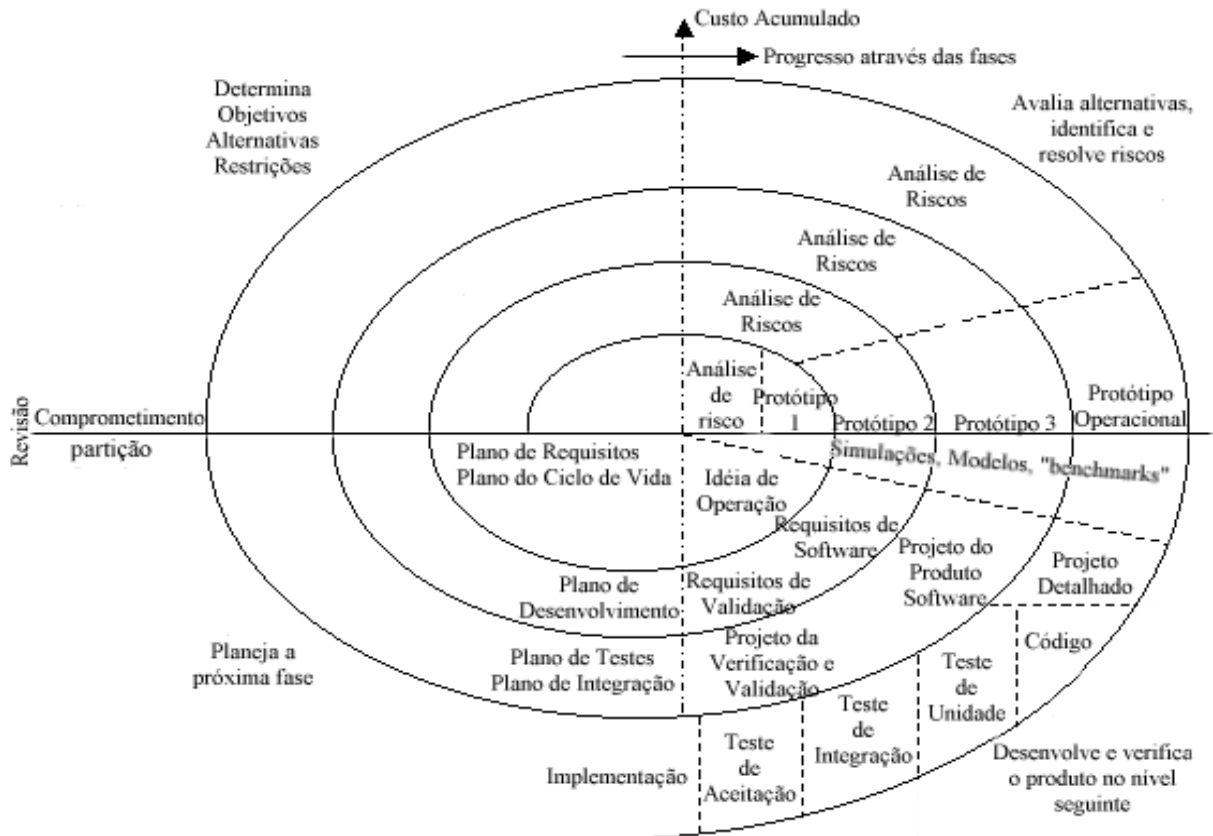


Figura 1: Modelo Espiral. Fonte: Sommerville (2007)

Este modelo assume que usuários, analistas de desenvolvedores adquirem conhecimento do projeto com o decorrer do tempo, por isso é considerado um dos modelos mais realísticos (SOMMERVILLE, 2008).

3 TECNOLOGIAS UTILIZADAS

O capítulo discorre sobre as diversas ferramentas que compõem o sistema.

3.1 Raspberry Pi

A aplicação utilizará o Raspberry Pi como servidor, recebendo, armazenando e enviando dados na rede no qual fará parte. Ele é um computador de baixo custo lançado na Inglaterra em 2012 pela Raspberry Pi Foundation, uma organização sem fins lucrativos. É considerado um dispositivo embarcado assim como aparelhos de TV, DVD e Blu-ray, tocadores de MP3, sistemas de alarme, celulares e etc.

Foi primeiramente concebido para estimular o aprendizado à programação, porém se tornou bastante popular para as mais diversas finalidades desde centrais multimídias, estações meteorológicas e aplicações industriais.

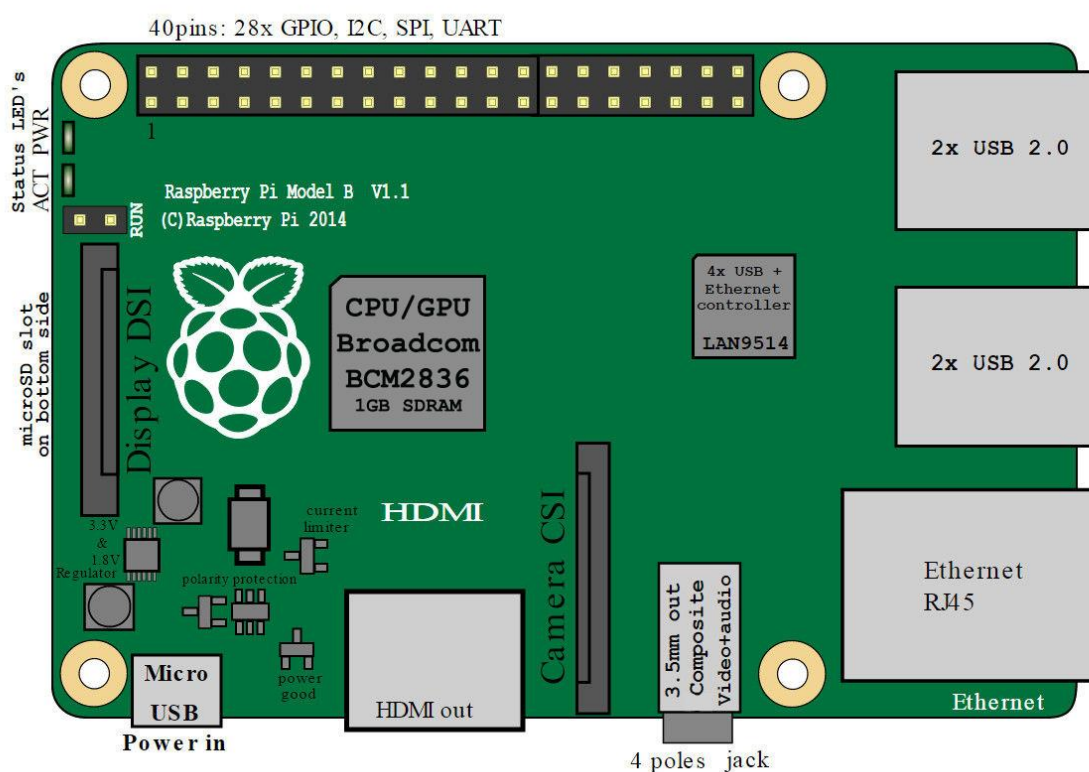


Figura 2: Ilustração do Raspberry Pi 2 Model B+. Fonte: SUSE.

Características do Raspberry Pi 2 Model B+ utilizado neste projeto:

- CPU ARM Cortex-A7 quad-core de 900MHz
- GPU VideoCore IV 3D (HDMI 1080p)
- 1 GB RAM
- 4 portas USB
- 1 porta Ethernet
- 1 slot para cartão Micro SD
- Com Wi-Fi e Bluetooth
- 40 pinos GPIO
- Consumo 5W

3.2 Ubuntu MATE

Para assumir o papel do sistema operacional no Raspberry Pi foi escolhido o GNU/Linux Ubuntu MATE 16.04 LTS. Baseado no Ubuntu, possui como interface gráfica o próprio MATE que leva no nome e também é compilado para a processadores com a arquitetura ARM.

A principal característica deste SO é o baixo consumo de recursos, o que faz ser ideal para um hardware mais modesto, e a vasta documentação encontrada em livros e na própria Internet.

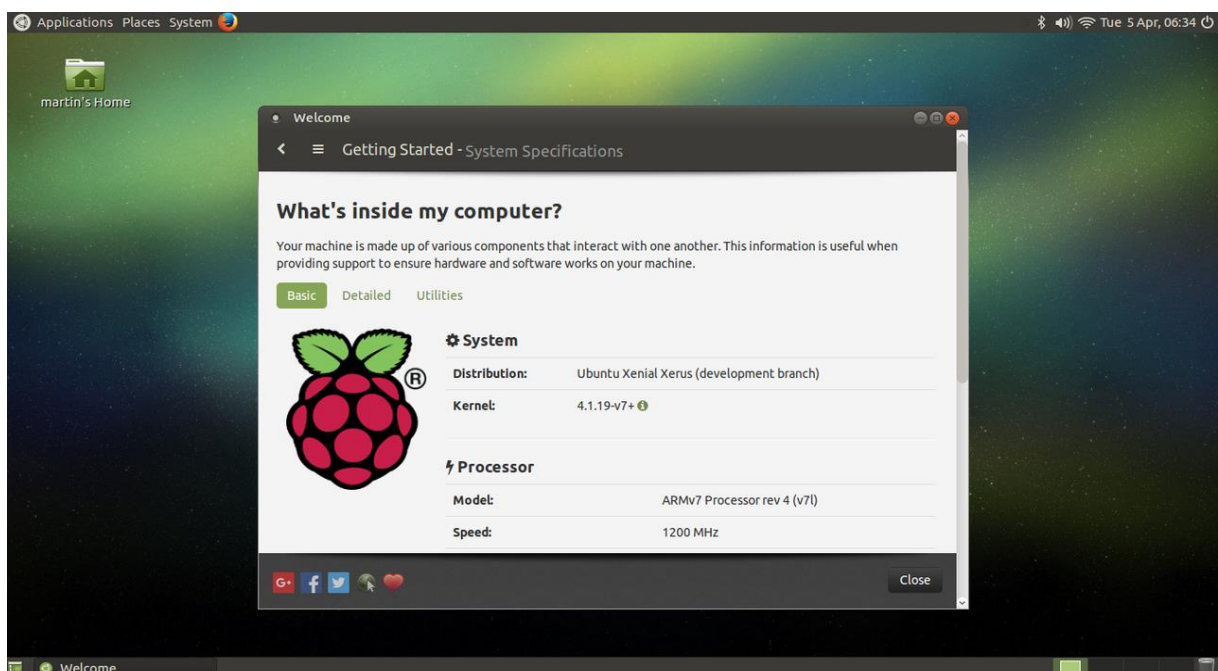


Figura 3: Interface do Ubuntu MATE em um Raspberry Pi. Fonte: Ubuntu-MATE.org

3.3 Nginx

O Nginx (Engine X) teve o seu desenvolvimento iniciado em 2002 pelo russo Igor Sysoev sendo disponibilizado ao público dois anos mais tarde, em 2004. Tem por objetivo ser um servidor Proxy HTTP reverso de baixo uso de memória, de grande escalabilidade e de alta performance.

Considerado o segundo servidor Web mais utilizado no mundo, o Nginx possui inúmeras possibilidades de configuração para a melhor performance, tornando-o o mais conveniente em sites de maior tráfego de informações.

Das principais características do Nginx, destacam-se:

- Proxy reverso;
- Baseado em eventos;
- Suporte à HTTP 2.0;
- Suporte à SSL auto assinado;
- Fácil integração com a linguagens como o PHP;
- Pode ser usado em conjunto com o servidor Apache;

3.4 MongoDB

O MongoDB (do inglês humongous, "monstruoso"), foi inicialmente desenvolvido por Dwight Merriman e Eliot Horowitz, que juntos formaram a empresa 10gen, em 2007, o MongoDB se tornou um software livre em 2009. É uma aplicação de alta performance, sem esquemas de colunas e orientado a documentos.

Eventualmente os bancos de dados orientados a documentos são bem diferentes dos habituais bancos de dados relacionais. O armazenamento das informações acontece dentro de documentos em um formato chamado BSON (uma versão binária de JSON), dispondo os dados em pares de chave/valor.

O MongoDB armazena, recupera e gerencia dados semiestruturados onde cada registro é chamado de documento, ou seja, uma unidade de dados armazenável. Estes documentos são endereçados no banco de dados via uma chave única que representa o documento. Os valores dos campos podem incluir outros documentos, vetores e vetores de documentos.

MySQL

```
INSERT INTO users (user_id, age, status)
VALUES ('bcd001', 45, 'A')
```

```
SELECT * FROM users
```

```
UPDATE users SET status = 'C'
WHERE age > 25
```

MongoDB

```
db.users.insert({
  user_id: 'bcd001',
  age: 45,
  status: 'A'
})
```

```
db.users.find()
```

```
db.users.update(
  { age: { $gt: 25 } },
  { $set: { status: 'C' } },
  { multi: true }
)
```

Figura 4: Linguagens de consulta MySQL e MongoDB. Fonte: MongoDB, Inc.

É considerado um dos projetos NoSQL mais elogiáveis devido suas consultas avançadas de alta performance e de fácil conversão para um banco de dados relacional podendo admitir mais de um banco de dados (neste caso chamado de coleções) armazenados um ao lado do outro com permissões individuais. É possível também consultar documentos de coleções distintas.

Das maiores vantagens de se utilizar o MongoDB é com certeza a proposta ágil no desenvolvimento de aplicações no mesmo tempo que possibilita a escalabilidade horizontal automática.

	MySQL	MongoDB
Rich Data Model	No	Yes
Dynamic Schema	No	Yes
Typed Data	Yes	Yes
Data Locality	No	Yes
Field Updates	Yes	Yes
Easy for Programmers	No	Yes
Complex Transactions	Yes	No
Auditing	Yes	Yes
Auto-Sharding	No	Yes

Quadro 1: Recursos suportados entre o MySQL e o MongoDB. Fonte: MongoDB, Inc.

3.5 PHP

Para o desenvolvimento de parte do sistema proposto foi escolhida a linguagem de programação PHP (Hypertext Preprocessor).

Criada no ano de 1994 por Rasmus Lerdorf, um engenheiro de software canadense-dinamarquês, o PHP encontra-se na sua sétima versão e é atualmente mantido por uma organização chamada The PHP Group.

É voltada para a criação de scripts do lado servidor com uso eficaz de recursos, sendo possível tanto ser incorporado junto com o HTML ou ainda como um binário independente, se mostrando uma forte alternativa à, por exemplo, o ASP (Active Server Pages) da Microsoft e o JSP (Java Server Pages) da Sun entre outras linguagens proprietárias e/ou patenteadas. Esta é, inclusive, uma das razões de ser das linguagens Web mais empregadas podendo-se citar também a sua filosofia de software livre, a vasta comunidade existente, a grande riqueza na documentação e a curva de aprendizado não tão íngreme.

Das principais características do PHP, destacam-se:

- Case sensitive;
- Execução rápida;
- Multiplataforma;
- Tipagem dinâmica;
- Linguagem procedural;
- Linguagem interpretada;
- Suporte à variados protocolos (IMAP, SNMP, NNTP, POP3, etc);
- Suporte à variados bancos de dados (mySQL, PostgreSQL, mongoDB, Redis, etc).

3.6 Golang

A linguagem de programação Go (ou Golang para facilitar pesquisas) foi criada em 2007 por Robert Griesemer, Rob Pike e Ken Thompson como um projeto interno do Google, contudo, somente em 2009 foi publicada e teve seu código aberto à comunidade. Tem como principais características ser uma linguagem compilada, com sintaxe simples, tipagem estática e ainda contar com um coletor de lixo.

Go foi projetado para resolver os problemas enfrentados no desenvolvimento de software do Google, e, apesar de não ser tão inovadora quanto parece, é uma excelente ferramenta para engenharia de grandes projetos de software. (PIKE, 2012)

Sendo assim a linguagem Go se mostra uma ótima opção para a criação de sistemas distribuídos em rede, característica principal do sistema proposto no presente trabalho.

4 LEVANTAMENTO DO SISTEMA

Neste capítulo são apresentados os requisitos funcionais e não funcionais, objetos e classes, e projeto de banco de dados.

4.1 Levantamento de Requisitos

A seguir serão demonstrados os requisitos funcionais e não funcionais oferecidos pelos sistemas web e móveis.

Na sequência serão demonstrados os requisitos que o sistema deverá ser capaz de atender.

4.1.1 Requisitos Funcionais

Os requisitos funcionais listam todas as funcionalidades que deverão existir em um sistema. Ressalta-se, no entanto, que um requisito determina ao que o software seja capaz de realizar, não importando a forma com que ele o faz.

A seguir a tabela de requisitos funcionais para o sistema proposto:

ID	Interesse	Descrição
RF01	Manter empresa	Cadastrar, editar, atualizar, excluir e pesquisar empresas ou filiais no sistema
RF02	Manter usuário	Cadastrar, editar, atualizar, excluir e pesquisar usuários no sistema
RF03	Manter parceiros	Cadastrar, editar, atualizar, excluir e pesquisar parceiros (fornecedores, clientes, etc) no sistema
RF04	Efetuar login	Efetuar login no sistema usando usuário e senha
RF05	Manter entradas de estoque	Cadastrar, editar, atualizar, excluir e pesquisar entradas de estoque no sistema
RF06	Manter saídas de estoque	Cadastrar, editar, atualizar, excluir e pesquisar saídas de estoque no sistema
RF07	Emitir relatórios	Emitir relatórios dos dados da empresa no sistema

Tabela 1: Lista de requisitos funcionais: Fonte: Autor.

4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais são, de forma simples, as qualidades desejáveis de um sistema, portanto não sendo visualizadas pelo usuário final na maioria das vezes.

A seguir a tabela de requisitos não funcionais:

ID	Interesse	Descrição
RN01	Manter conexão criptografada	Manter a conexão com o sistema criptografada por certificados HTTPS
RN02	Criar interfaces inteligentes	Criar interfaces de usuário responsivas e intuitivas
RN03	Manter desempenho	Manter o desempenho do sistema mesmo em pico de tráfego ou perda de nós

Tabela 2: Lista de requisitos não funcionais: Fonte: Autor.

4.2 Objetos Identificados

Esta seção trata dos objetos identificados através das listas de requisitos. Esta é uma modelagem utilizada para projetos que fazem uso do MongoDB como banco de dados.

Parceiro		
Campo	Tipo	Descrição
_id	Inteiro	Identificador exclusivo para o parceiro.
ativo	booleano	Determina se cadastro do parceiro está ativo ou não.
tipo_pessoa	texto	Tipo de pessoa física ou jurídica para o parceiro.
data_cadastro	data	Data e hora de cadastro do parceiro.
banco	texto	Banco em que o parceiro possui conta.
agencia	inteiro	Número da agência do parceiro.
conta	inteiro	Número da conta do parceiro.
ultima_atualizacao	data	Data e hora da última atualização do cadastro do parceiro.
Nome	texto	Opcional. Nome do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa física.

sobrenome	texto	Opcional. Sobrenome do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa física.
nascimento	data	Opcional. Data de nascimento do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa física.
sexo	texto	Opcional. Sexo do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa física.
cpf	texto	Opcional. CPF do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> sendo preenchido caso seja pessoa física e por algoritmo matemático.
rg	texto	Opcional. RG do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa física.
razao_social	texto	Opcional. Razão social do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa jurídica.
nome_fantasia	texto	Opcional. Nome fantasia do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa jurídica.
cnpj	inteiro	Opcional. CNPJ do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> sendo preenchido caso seja pessoa jurídica e por algoritmo matemático.
inscricao_municipal	inteiro	Opcional. Inscrição municipal do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> sendo preenchido caso seja pessoa jurídica e caso cliente tenha uma inscrição municipal.
inscricao_estadual	inteiro	Opcional. Inscrição estadual do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> sendo preenchido caso seja pessoa jurídica e caso cliente tenha uma inscrição estadual.
fundação	data	Opcional. Data de fundação do parceiro. É validado à nível de software pelo campo <i>tipo_pessoa</i> , sendo preenchido caso seja pessoa jurídica.
emails	matriz	Campo é uma matriz de e-mails.
enderecos	matriz	Campo é uma matriz de endereços.
telefones	matriz	Campo é uma matriz de telefones.

Tabela 3: Objeto Parceiro: Fonte: Autor.

E-mail		
Campo	Tipo	Descrição
tipo_email	texto	Tipo de e-mail (pessoal, comercial, etc)
email	texto	E-mail do parceiro.

Tabela 4: Objeto E-mail: Fonte: Autor.

Endereço		
Campo	Tipo	Descrição
tipo_endereco	texto	Tipo de endereço (pessoal, entrega, recebimento, etc).
rua	texto	E-mail do parceiro.
numero	inteiro	Número do endereço do parceiro.
cep	texto	CEP do parceiro.
bairro	texto	Bairro do parceiro.
cidade	texto	Cidade do parceiro
uf	texto	UF do parceiro.
pais	texto	País do parceiro.

Tabela 5: Objeto Endereço: Fonte: Autor.

Telefone		
Campo	Tipo	Descrição
tipo_telefone	texto	Tipo de telefone (residencial, comercial, celular, etc)
telefone	texto	Telefone do parceiro.

Tabela 6: Objeto Telefone: Fonte: Autor.

Entrada		
Campo	Tipo	Descrição
_id	inteiro	Identificador exclusivo para a entrada.
Empresa	inteiro	Identificador exclusivo para uma empresa cadastrada. Veja <i>Empresa</i> .
Tipo_operacao	inteiro	Tipo de entrada (compra, devolução, remessa futura, etc.).
usuario	inteiro	Usuário que cadastrou a entrada.
Data_movimentacao	data	Data e hora da movimentação.
Observação	texto	Opcional. Observação sobre a entrada de produtos.
Produtos	matriz	Campo é uma matriz de <i>Produto</i> .
Pagamento	inteiro	Forma de pagamento da entrada.

Tabela 7: Objeto Entrada: Fonte: Autor.

Saída		
Campo	Tipo	Descrição
_id	inteiro	Identificador exclusivo para a saída.
Empresa	inteiro	Identificador exclusivo para uma empresa cadastrada. Veja Empresas.
Tipo_operacao	inteiro	Tipo de saída (venda, avaria, simples remessa, etc.).
usuario	inteiro	Usuário que cadastrou a saída.
Data_movimentacao	data	Data e hora da movimentação.
Observação	texto	Opcional. Observação sobre a saída de produtos.
Produtos	matriz	Campo é uma matriz de <i>Produto</i> .
Pagamento	inteiro	Forma de pagamento da saída.

Tabela 8: Objeto Saída: Fonte: Autor.

Produto		
Campo	Tipo	Descrição
_id	inteiro	Identificador exclusivo para o produto.
Descricao	texto	Descrição sobre o produto.
Grupo	texto	Grupo do produto.
Unidade	texto	Unidade do produto.
Marca	texto	Marca do produto.
Quantidade	decimal	Quantidade do produto.
Valor	decimal	Opcional. Valor unitário do produto.
Usuario	inteiro	Usuário que cadastrou o produto.

Tabela 9: Objeto Produto: Fonte: Autor.

4.3 Diagrama de Coleções do MongoDB

Afim de modelar as coleções que abrigarão os dados, foi gerado os seguintes esquemas:

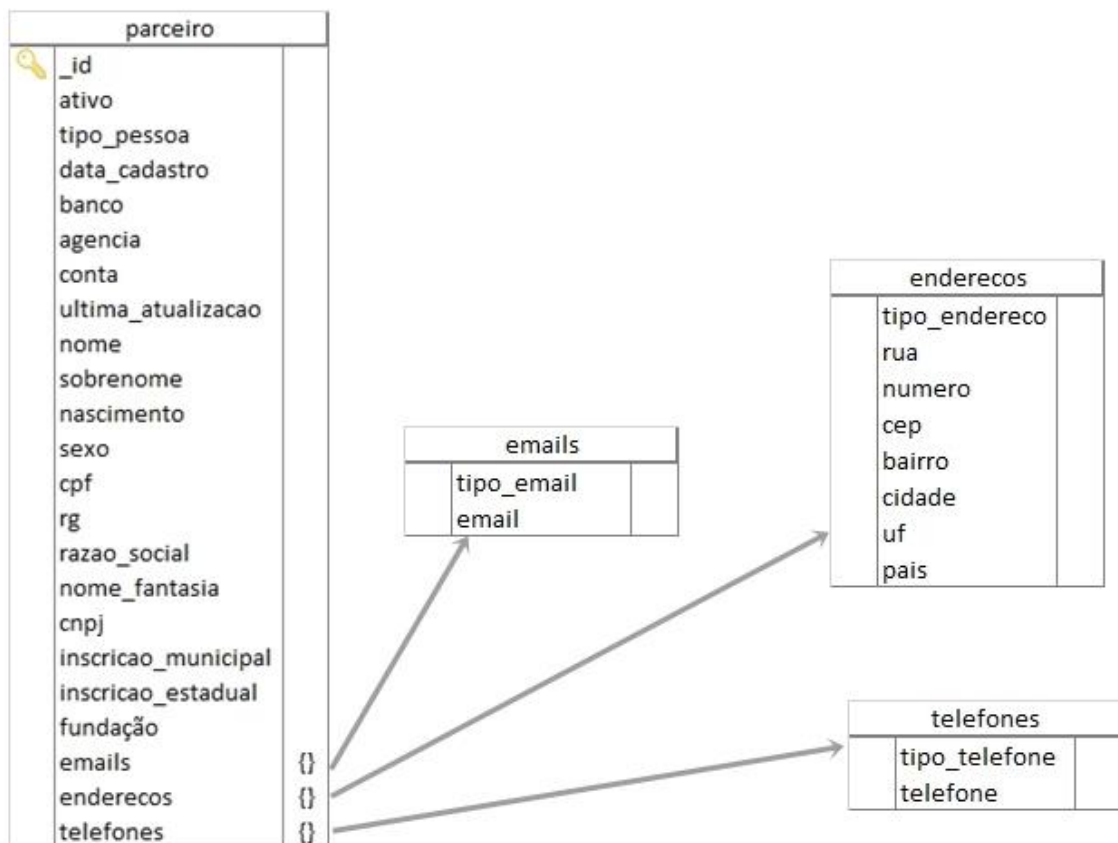


Figura 5: Coleção Parceiro. Fonte: Autor.

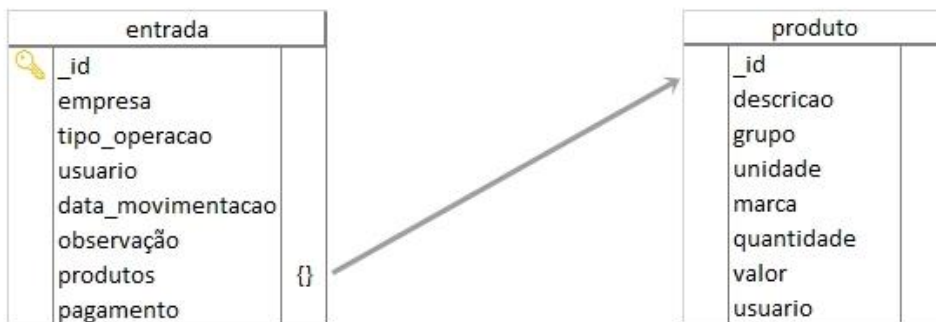


Figura 11: Coleção Entrada. Fonte: Autor.

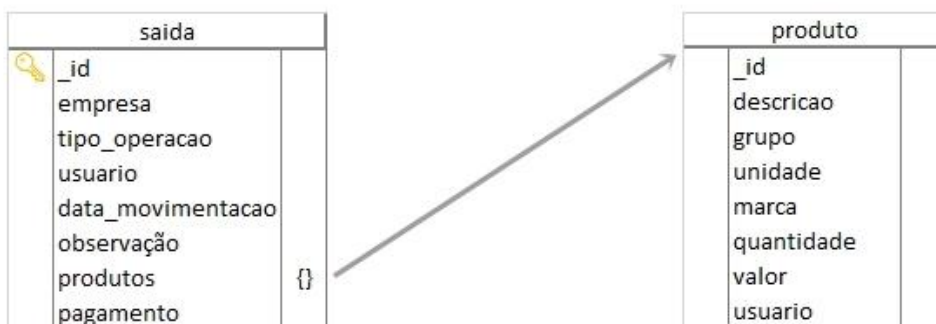


Figura 12: Coleção Saída. Fonte: Autor.

5 PROTOTIPAÇÃO

A seguir será exibido a prototipação inicial do sistema proposto:

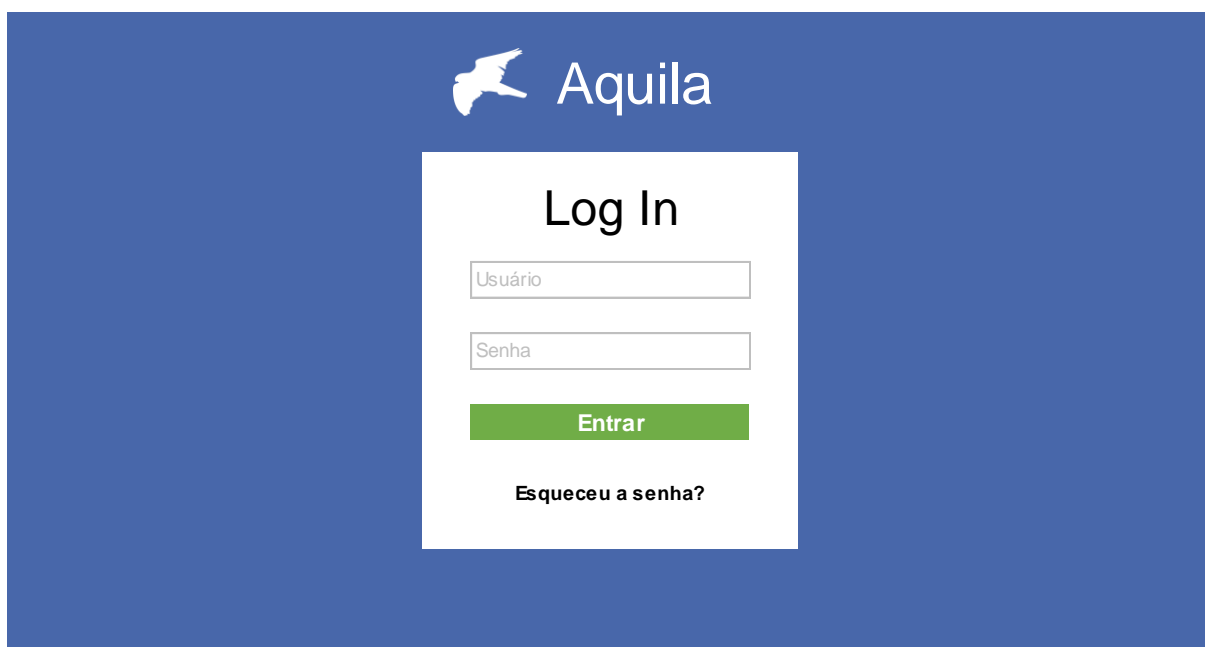


Figura 8: Tela de login no sistema. Fonte: Autor.

ENTRADAS					
ID	EMPRESA	TIPO DE OPERAÇÃO	USUÁRIO	DATA	OBSERVAÇÃO
#1	Empresa 1	Compra à vista	Administrador	20/10/2017	-
#2	Empresa 1	Compra à vista	Administrador	20/10/2017	-
#3	Empresa 1	Devolução parcial	Administrador	20/10/2017	Avaria
#4	Empresa 1	Compra à vista	Administrador	20/10/2017	-
#5	Empresa 1	Compra à prazo	Administrador	23/10/2017	-
#6	Empresa 1	Fabricação	Administrador	23/10/2017	-
#7	Empresa 1	Devolução parcial	Administrador	23/10/2017	Devolução normal
#8	Empresa 1	Compra à prazo	Administrador	23/10/2017	-
#9	Empresa 1	Fabricação	Administrador	23/10/2017	Produção normal
#10	Empresa 1	Fabricação	Administrador	23/10/2017	Produção normal

Figura 9: Tela de registros de entradas. Fonte: Autor.

Empresa 1 Administrador

PRODUTOS

ID	DESCRIÇÃO	GRUPO	MARCA	QUANTIDADE
#1	Produto 1	Grupo 7	Marca 2	2
#2	Produto 2	Grupo 5	Marca 2	11
#3	Produto 3	Grupo 5	Marca 3	12
#4	Produto 4	Grupo 8	Marca 1	8
#5	Produto 5	Grupo 3	Marca 3	15
#6	Produto 6	Grupo 4	Marca 3	5
#7	Produto 7	Grupo 6	Marca 1	9
#8	Produto 8	Grupo 5	Marca 3	8
#9	Produto 9	Grupo 6	Marca 1	16
#10	Produto 10	Grupo 6	Marca 1	18

Figura 10: Tela de registros de produtos. Fonte: Autor.

Empresa 1 Administrador

PARCEIROS

Pessoa Física Pessoa Jurídica

Figura 11: Tela de consulta de parceiros. Fonte: Autor.

6 CRONOGRAMA

Abaixo é descrito o cronograma de desenvolvimento definido para o trabalho como um todo:

ATIVIDADES / MÊS	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ	JAN	FEV
Levantamento Bibliográfico	█	█	█	█								
Levantamento de Requisitos			█	█	█							
Qualificação TC				█								
Análise de Sistemas					█	█	█	█				
Apresentação TC									█			
Desenvolvimento								█	█	█	█	
Testes										█	█	█

Quadro 2: Cronograma de desenvolvimento do trabalho. Fonte: Autor.

7 CONCLUSÕES

Neste trabalho foi apresentado um sistema comercial distribuído baseado em dispositivos embarcados. O sistema se mostrou uma alternativa viável do ponto de vista técnico e econômico, provendo ferramentas eficientes que apoiem a gestão do negócio e redução do investimento em infraestrutura e licenças de *software*. Seguindo este raciocínio poderemos tornar mais nítido o uso da tecnologia no ambiente organizacional, facilitando a tomada de decisões, melhorando o fluxo de trabalho, etc.

Além disto foi aprimorado o entendimento de como um sistema de informação é criado, desde a concepção da ideia, as transformações que essa ideia passa até a análise e desenvolvimento da mesma, proporcionando uma rica bagagem ao autor.

Contudo é importante ressaltar que os estudos não param ao final deste trabalho. O aprimoramento deste projeto e a abertura à novas ideias são novos pontos de partida para se tomar, colocando em prática tudo aquilo que foi aprendido no decorrer do curso.

8 REFERÊNCIAS BIBLIOGRÁFICAS

BALLOU, Ronald H. Logística Empresarial. São Paulo: Atlas, 1993.

BOAGLIO, Fernando; MongoDB: Construa novas aplicações com novas tecnologias. 1.ed. São Paulo. Casa do Código, 2015.

CEZARINO, Luciana O; CAMPOMAR, M. C. Micro e pequenas empresas: características estruturais e gerenciais. São Paulo, 2006. Disponível em: <<http://www.unifafibe.com.br/revistasonline/arquivos/hispecielemaonline/sumario/10/19042010081633.pdf>>. Último acesso em out. 2017.

CORREIA, Carlos; TAFNER, Malcon. Análise orientada a objetos. 2. ed. Florianópolis: Visual Books, 2006.

COSTA, Fabio. Introdução à administração de Materiais em Sistemas Informatizados. IE Editora, São Paulo, 2002. Disponível em: <<https://pt.scribd.com/doc/6582816/Introducao-a-Administracao-de-Materiais-Em-Sistemas-Informatizados-Nadia>>. Último acesso em set. 2017.

DAMASCO, Miguel. Sistema de Apoio às Operações, 2016. Disponível em: <<http://www.profdamasco.site.br.com/SigEaD/1ApoioOperacoes.htm>>. Último acesso em out. 2017.

DOS SANTOS, Sandro da S. Sistemas de Processamento de Transações, 2008. Disponível em: <<https://sssantos.files.wordpress.com/2008/04/sig-aula-spt.pdf>>. Último acesso em out. 2017.

FILIPINI, Caio; Programando em Go: Crie aplicações com a linguagem do Google. 1.ed. São Paulo. Casa do Código, 2014.

GOIAS EMPRESAS. Disponível em: <<http://goiasempresas.com.br/>>. Último acesso em mai. 2017.

MOTA, Denysson A. R; FELIPE, André A. C. Gestão do conhecimento em empresas através de sistemas colaborativos (GROUPWARE), 2008. Disponível em: <<https://denysson.files.wordpress.com/2009/11/artigo-sistemas-colaborativos-kmbrasil-2009.pdf>>. Último acesso em out. 2017.

NGINX. Nginx News. 2017. Disponível em: <<http://nginx.com/>>. Último acesso em ago. 2017.

OLIVEIRA, Carla. Conceito de Sistemas Colaborativos, 2006. Disponível em: <<http://imasters.com.br/artigo/4655/gerencia-de-ti/sistemas-colaborativos-conceito-caracteristicasdes-e-funcionalidades?trace=1519021197&source=single>>. Último acesso em out. 2017.

SLACK, Nigel; CHAMBERS, Stuart; JOHNSTON, Robert. Administração da Produção. São Paulo: Editora Atlas, 2002

SYSOEV, Igor. nginx documentation. Disponível em: <<https://nginx.org/en/docs/>>. Último acesso em jun. 2017.

APÊNDICES

APÊNDICE A – Exemplo JSON gerado para um *Parceiro*:

```
[
  {
    "_id": 1,
    "ativo": 1,
    "tipo_pessoa": "Pessoa Física",
    "data_cadastro": "2017-10-10 12:53",
    "banco": "Caixa Econômica",
    "agencia": 4581,
    "conta": 145294,
    "ultima_atualizacao": "2017-11-01 08:04",
    "nome": "Aguinaldo Gabriel",
    "sobrenome": "dos Reis e Silva",
    "nascimento": "1994-30-08",
    "sexo": "Masculino",
    "cpf": "414.147.516-56",
    "rg": "452457",
    "razao_social": "",
    "nome_fantasia": "",
    "cnpj": "",
    "inscricao_municipal": "",
    "inscricao_estadual": "",
    "fundação": "",
    "emails" :
    [
      {
        "tipo_email": "Pessoal",
        "email": "gabrielreis@gmail.com"
      },
      {
        "tipo_email": "Comercial",
        "email": "contato@aguinaldogabriel.com.br"
      }
    ]
  }
]
```

```
    }
  ],
  "enderecos" :
  [
    {
      "tipo_endereco": "Residencial",
      "rua": "Rua 1 Quadra 3 Lote 5",
      "numero": "5",
      "cep": "76630-000",
      "bairro": "Jardim Diamante",
      "cidade": "Itaberaí",
      "uf": "Goiás",
      "pais": "Brasil"
    }
  ],
  "telefones" :
  [
    {
      "tipo_telefone": "Pessoal",
      "telefone": "+55 (62) 9 9988-7766"
    },
    {
      "tipo_telefone": "Comercial",
      "telefone": "+55 (62) 3375-9999"
    }
  ]
}
]
```