

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ENGENHARIA INDUSTRIAL ELÉTRICA: ÊNFASE
EM ELETRÔNICA/TELECOMUNICAÇÕES

FELIPE AUGUSTO DESTEFANI
FILLIPE LUCCHIN PAUKNER

**Servidor VoIP com Sistema de Segurança Doméstica Integrada de
Baixo Custo**

TRABALHO DE CONCLUSÃO DE CURSO

CURITBA

2013

FELIPE AUGUSTO DESTEFANI
FILLIPE LUCCHIN PAUKNER

**Servidor VoIP com Sistema de Segurança Doméstica Integrada de
Baixo Custo**

Trabalho de Conclusão de Curso de graduação, apresentado na disciplina de Projeto Final II, do Curso de Engenharia Elétrica com ênfase em Eletrônica / Telecomunicações do Departamento Acadêmico de Eletrônica – DAELN – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Engenheiro.

Orientador: Prof. Dr. Kleber Kendy Horikawa Nabas

CURITBA

2013

FELIPE AUGUSTO DESTEFANI
FILLIPE LUCCHIN PAUKNER

**SERVIDOR VOIP COM SISTEMA DE SEGURANÇA DOMÉSTICA
INTEGRADA DE BAIXO CUSTO**

Este trabalho de conclusão de curso foi apresentado no dia 17 de setembro de 2013, como requisito parcial para obtenção do título de Engenheiro em Engenharia Industrial Elétrica: Ênfase em Eletrônica/Telecomunicações, outorgado pela Universidade Tecnológica Federal do Paraná. Os alunos foram arguídos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Hilton José Silva Azevedo
Coordenador de Curso
Departamento Acadêmico de Eletrônica

Prof. Dr. Dario Eduardo Amaral Dergint
Coordenador de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Rubens Alexandre de Faria

Prof. Dr. Kleber Kendy Horikawa Nabas

Prof. Msc. Lincoln Herbert Teixeira

Dedicamos este trabalho de conclusão de curso aos nossos pais, irmãos, familiares, e amigos que de muitas formas nos incentivaram e ajudaram de maneira a tornar possível a conclusão deste trabalho.

AGRADECIMENTOS

Gostaríamos de agradecer a nosso Orientador, Professor Kleber Nabas por suas instruções e seu auxílio.

Também somos gratos aos docentes do DAELN que nos proporcionaram uma extensa gama de conhecimentos durante todo o desenvolvimento do curso sem os quais não seria possível desenvolver este projeto.

Agradecemos também aos nossos colegas de curso que, assim como nós, batalharam muito nesses cinco anos para chegarem ao término desta graduação e cuja parceria e os momentos de descontração foram essenciais no decorrer deste processo.

Também queremos agradecer aos nossos familiares que foram fundamentais em todo nosso processo estudantil nos educando e nos ensinando a nunca desistir e nem desanimar frente às dificuldades. Obrigado queridos pais por todo seu apoio incondicional em todos esses momentos.

Por último, agradecemos a Deus pelas bênçãos diárias que nos concede, proporcionando vida, saúde e força de vontade para enfrentar todas as dificuldades da vida.

Nossos mais sinceros agradecimentos a todos,

Felipe Augusto Destefani

Fillipe Lucchin Paukner

RESUMO

Este projeto visa a implementação de um sistema VoIP domiciliar de baixo custo. A metodologia consiste em especificar o projeto, tomando conhecimento da importância para os potenciais clientes e usuários finais do produto, modelar e definir soluções que atendam as funcionalidades previstas, implementar e integrar os diversos módulos que compõem o sistema para então validar o protótipo e finalizar o projeto, realizando levantamentos acerca de tudo que envolveu seu desenvolvimento. O produto final, o Fly-IP é composto de várias células com funcionalidades diversas, porém que são controladas por dois servidores que comunicam-se entre si. Um dos servidores trabalha com a comunicação VoIP, abrange a telefonia e a maioria das possíveis operações da telefonia convencional atual, mas com uma vantagem, ligações sem custo. O segundo servidor se comunicará com o primeiro de forma que haja uma interface com várias possíveis entidades de controle com uma finalidade específica, por exemplo, um usuário do sistema como um todo pode usar seu telefone para abrir o portão da sua casa via internet.

Palavras Chave: Mini-PC, VoIP, Asterisk, Sip Phone.

ABSTRACT

This project aims to develop a low cost home VoIP system. The methodology consists in specifying the project, starting by learning this projects importance to the potential clients and product final users, then modeling and defining solutions with the specified functionalities, implementing and integrating the various modules which compose the whole system to then validate the prototype and finish the project, gathering information from all parts along the project development. The final product, Fly-IP has various cells with various possible functionalities, but there are two main cells composed by two servers which communicate between themselves. The first server works with VoIP communication, it will cover the telephony part of the system with the majority of the possible operations from the actual telephony. The second server will communicate with the first making an interface with various possible control entities, for example, a system user may use his own cellphone to open his house entrance.

Key Words: Mini-PC, VoIP, Asterisk, Sip Phone.

LISTA DE FIGURAS

Figura 1 - Diagrama em blocos do projeto	17
Figura 2 - Beagle Board	21
Figura 3 – Panda Board	22
Figura 4 – Raspberry Pi.....	23
Figura 5 - Raspberry Pi - Características	23
Figura 6 – Comparação entre modelos A e B	24
Figura 7 – Modelos SIP Phone por Hardware e por Software	27
Figura 8 – Win32 Disk Imager	31
Figura 9 - Página Inicial do FreePBX	34
Figura 10 – Adicionando Extensão no FreePBX	35
Figura 11 - Adicionando Extensão SIP ao FreePBX	35
Figura 12 – Raspberry Pi GPIO.....	38
Figura 13 - Sensor PIR.....	40
Figura 14 - Esquemático do módulo de automação para segurança domiciliar.	43
Figura 15 – Efetuando Ligação pelo SIP Phone.....	44
Figura 16 – Módulo de automação para segurança domiciliar – vista superior.....	45
Figura 17 - Módulo de automação para segurança domiciliar – vista lateral.....	45
Figura 18 - Alocação de espaço do layout da empresa.....	55

LISTA DE TABELAS

Tabela 1 - Update do Raspberry e instalação de programas úteis.....	31
Tabela 2 - Download e extração do Asterisk.....	32
Tabela 3 - Configuração e instalação do Asterisk.....	32
Tabela 4 - Instalação do repositório PHP Pear.....	32
Tabela 5 - Baixando e extraíndo o FreePBX.....	32
Tabela 6 - Inicialização do Banco de Dados.....	33
Tabela 7 - Criando Usuário do MySQL.....	33
Tabela 8 - Protegendo MySQL com Palavra Chave.....	33
Tabela 9 - Renicializando o Asterisk.....	33
Tabela 10 - Configurando Apache.....	33
Tabela 11 - Reiniciando o Apache.....	34
Tabela 12 - Instalação do FreePBX.....	34
Tabela 13 - Permissões de acesso ao FreePBX por usuários Apache.....	34
Tabela 14 - Exemplo de configuração de dialplan.....	36
Tabela 15 – Exemplo de configuração de chamada automática.....	36
Tabela 16 - Exemplo de configuração de trunk.....	37
Tabela 17 - Instalação do Python.....	37
Tabela 18 - Instalação da Bibliotecas de acesso ao GPIO.....	38
Tabela 19 - Script python de controle do Sensor PIR.....	40
Tabela 20 - Script para realização de chamada de aviso de falha de segurança.....	41
Tabela 21 - Tratamento das ligações para ativar/desativar atuadores.....	42
Tabela 22 - Chamada do shell script que integra dialplan com Python.....	42
Tabela 23 - Scripts Python de tratamento da GPIO, pino 24.....	42
Tabela 24 - Cronograma de desenvolvimento.....	46
Tabela 25 - Relação de carga horária para cada etapa do projeto.....	47
Tabela 26 - Capital Social.....	51
Tabela 27 - Estudo dos Concorrentes.....	52
Tabela 28 - Estudo dos Fornecedores.....	52
Tabela 29 - Necessidade de Pessoal.....	57
Tabela 30 - Máquinas e Equipamentos.....	57
Tabela 31 – Veículos.....	57
Tabela 32 - Móveis e utensílios.....	58
Tabela 33 - Estimativa do Estoque Inicial.....	58
Tabela 34 - Caixa Mínimo.....	58
Tabela 35 - Cálculo do Prazo Médio de Compras.....	59
Tabela 36 - Cálculo de Necessidade Média de Estoque.....	59
Tabela 37 - Cálculo da Necessidade Líquida de Capital.....	59
Tabela 38 - Caixa Mínimo Calculado.....	59
Tabela 39 - Capital de Giro Resumido.....	60
Tabela 40 - Investimentos Pré Operacionais.....	60
Tabela 41 - Investimento Total Resumido.....	60

Tabela 42 - Estimativa do Faturamento Mensal da Empresa	60
Tabela 43 - Custos de Matéria-Prima, Materiais Diretos e Terceirizações.....	61
Tabela 44 - Estimativa dos Custos de Comercialização	61
Tabela 45 - Custo dos Materiais Diretos E/Ou Mercadorias Vendidas.....	61
Tabela 46 - Estimativa dos Custos com Mão-De-Obra	62
Tabela 47 - Estimativa do Custo com Depreciação.....	62
Tabela 48 - Estimativa de Custos Fixos Operacionais Mensais.....	63
Tabela 49 - Demonstrativo de Resultados	63
Tabela 50 - Ponto de Equilíbrio	64
Tabela 51 - Lucratividade	64
Tabela 52 - Rentabilidade	65
Tabela 53 - Prazo de Retorno de Investimento	65
Tabela 54 - Construção de Cenários.....	65
Tabela 55 - Análise da Matriz F.O.F.A.	66

SUMÁRIO

1. INTRODUÇÃO	14
1.1 Justificativa.....	15
1.2 Objetivos	16
1.3 Diagrama em blocos	16
1.4 Resultados Esperados.....	17
1.5 Metodologia	18
2. FUNDAMENTAÇÃO TEÓRICA	20
2.1 Mini PCs	20
2.1.1 Beagle Board.....	21
2.1.2 Panda Board	22
2.1.3 Raspberry Pi.....	23
2.2 O sistema VoIP	24
2.3. Asterisk	25
2.4 Sip Phone	26
3. METODOLOGIA	28
3.1 Escolhendo um Mini-PC	28
3.2 Distribuição Linux – Debian	29
3.3 Escolha do Python como linguagem de desenvolvimento	29
3.4 Desenvolvimento do Sistema de Comunicação VoIP	30
3.4.1 Instalação do sistema.....	30
3.4.2 Instalação do Debian no Cartão SD	30
3.4.3 Instalação do Asterisk	31
3.4.4 Criando SipPhones no Asterisk.....	35
3.4.5 Configurações do Asterisk.....	36
3.5 Desenvolvimento do Sistema de Automação Domiciliar	37
3.5.1 Instalação do Python	37
3.5.2 Instalação da biblioteca de acesso ao GPIO do Raspberry.	38
3.5.3 Interface com sensores e atuadores	39
3.5.4 Sensor de Presença PIR	39
3.5.5 Atuadores para alerta de falha de segurança.....	41
3.5.6 Indicador de Funcionamento do Sensor.....	43
3.6 Desenvolvimento do hardware	43
4. TESTES E RESULTADOS	44
5. GESTÃO	46
5.1 Cronogramas de Planejamento	46

6. PLANO DE NEGÓCIO	48
6.1. Sumário Executivo	48
6.1.1. Resumo dos Principais Pontos Do Plano De Negócio	48
6.1.2. Dados dos Empreendedores, Experiência Profissional e Atribuições	49
6.1.3. Dados do Empreendimento	50
6.1.4. Missão da Empresa	50
6.1.5. Setores de Atividade	50
6.1.6. Forma Jurídica.....	50
6.1.7. Enquadramento Tributário.....	50
6.1.8. Capital Social	51
6.1.9. Fonte de Recursos	51
6.2. Análise de Mercado.....	51
6.2.1. Estudo dos Clientes.....	51
6.2.2. Estudo dos Concorrentes	52
6.2.3. Estudo dos Fornecedores	52
6.3. Plano de Marketing.....	53
6.3.1. Descrição dos Principais Produtos E Serviços	53
6.3.2. Preço	53
6.3.3. Estratégias Promocionais	53
6.3.4. Estrutura de Comercialização	53
6.3.5. Localização do Negócio	54
6.4. Plano Operacional	55
6.4.1. <i>Layout</i> ou Arranjo Físico.....	55
6.4.2. Capacidade Produtiva/Comercial/Serviços	56
6.4.3. Processos Operacionais.....	56
6.4.4 Necessidade de Pessoal	57
6.5. Plano Financeiro.....	57
6.5.1. Estimativa dos Investimentos Fixos.....	57
6.5.2. Capital de Giro.....	58
6.5.3. Capital de Giro Resumido	60
6.5.4. Investimentos Pré-Operacionais.....	60
6.5.5. Investimento Total Resumido	60
6.5.6. ESTIMATIVA DO FATURAMENTO MENSAL DA EMPRESA.....	60
6.5.7. Estimativa dos Custos de Matéria-Prima, Materiais Diretos e Terceirizações.....	61
6.5.8. Estimativa dos Custos de Comercialização.....	61
6.5.9. Apuração do Custo dos Materiais Diretos e/ou Mercadorias Vendidas	61
6.5.10. Estimativa dos Custos com Mão-De-Obra	62
6.5.11. Estimativa do Custo com Depreciação	62
6.5.12. Estimativa de Custos Fixos Operacionais Mensais	63
6.5.13. Demonstrativo de Resultados	63
6.5.13. Indicadores de Viabilidade	64
6.6. Construção de Cenários	65
6.7. Avaliação Estratégica	66

6.7.1. Análise da Matriz F.O.F.A.	66
6.8. Avaliação do Plano De Negócio	66
7. CONCLUSÕES	67
REFERÊNCIAS BIBLIOGRÁFICAS	68

1. INTRODUÇÃO

Os sistemas de telecomunicações atuais são baseados em centrais telefônicas automáticas. O paradigma atual é se utilizar uma conexão via um par metálico para enviar/receber dados de telefonia e já há alguns anos que não se nota nenhuma grande diferença nos serviços prestados ao usuário final. As empresas de telecomunicações dominam áreas muito grandes no Brasil, isto faz com que seja possível um baixo preço para todos os clientes de uma mesma operadora, pois entre seus sistemas não é necessário nenhum tipo de configuração extra.

Os preços de ligações variam conforme a distância que a mensagem deve percorrer via dispositivos, caso a ligação permaneça dentro da própria operadora, o custo é mínimo tanto para o usuário quanto para a operadora, porém se a ligação é internacional, a ligação deve sair da operadora e chegar em outra operadora em outro país, para então chegar ao destinatário, isto envolve gastos maiores que são repassados ao usuário final.

Este paradigma traz uma complicação quando se trata de integrar este sistema com algum outro. Neste trabalho, será desenvolvida uma interação entre um sistema de telecomunicações com um sistema de automação residencial.

A automação eletrônica residencial tem crescido aos poucos, algumas pessoas têm um aquecedor a gás microcontrolado, outras possuem um sistema de acender e apagar luzes batendo palmas, porém não é uma tendência, visto que muitas vezes são automações isoladas que não geram muita vantagem para os sistemas convencionais.

O maior avanço das telecomunicações hoje está na facilidade de se comprar um aparelho celular e fazer ligações para qualquer pessoa no mundo, por um preço acessível. O avanço na área eletrônica também permite que estes aparelhos celulares também sejam superdesenvolvidos e possuam muitas funcionalidades que alguns anos atrás não existiam, podemos operar hoje um telefone como se este fosse um computador. A intenção aqui é aproveitar o celular como interface de automação para a residência do seu dono, seja para controlar o fluxo de gás do aquecedor de água da sua casa, ou para abrir o portão de sua garagem.

Um grande avanço das telecomunicações que ainda não é tão popularizado como produto é o servidor PABX. Existem vários servidores destes dispostos na

rede como *freewares*, ou seja, são isentos de taxas para se utilizar em produtos comerciais, também são *open-source*, o que quer dizer que seus códigos são abertos a terceiros para serem corrigidos ou para que sejam geradas novas funcionalidades. Porém, dado a complexidade de se utilizar um servidor destes ou configurar, não é possível tornar esta uma opção viável a todas as pessoas, caso contrário, o serviço de telecomunicações passaria a ser acessível gratuitamente a todos que possuam um computador, um celular, acesso a internet e um modem com wireless – o que a grande maioria das pessoas já possui.

A complexidade é uma grande barreira à popularização de um produto, pode-se considerar que a automação residencial é bastante complexa ainda hoje, pois para uma casa totalmente automatizada existem várias possíveis interfaces diferentes utilizadas para automatizar cada parte possível, o número de possibilidades é muito grande. Portanto, ao se escolher o celular como uma interface com toda automatização da residência e caso seja criada uma interface *user-friendly* para a configuração de um servidor PABX, existirá um produto muito mais fácil de ser popularizado, considerando que seu gasto com comunicação será muito mais baixo e o usuário seria livrado do paradoxo da escolha ao utilizar o celular.

1.1 Justificativa

Ao longo dos anos, o paradigma de uma central telefônica se fortaleceu no mundo todo, porém a comunicação, na maioria dos casos se dá entre apenas dois usuários. Mesmo havendo a possibilidade de existirem várias ligações simultâneas, a ideia de existir um intermediador entre todos estes usuários está estagnada há muito tempo, pois podemos considerar que todos estes usuários possuem uma conexão ADSL com uma taxa de transferência suficientemente grande para transmitir e receber voz em suas residências.

Os objetivos do sistema são: garantir que o usuário usufrua gratuitamente da possibilidade de comunicação através da internet diretamente com outro usuário; e trazer uma alternativa de sistema de segurança doméstica com acesso facilitado através da telefonia. Desta maneira, haverá a possibilidade de o usuário inserir o IP de outro usuário na *database* do seu sistema e ligar diretamente para esta pessoa sem o intermédio de um servidor, assim como possibilitará a configuração de níveis de privilégio para acesso ao sistema de segurança.

1.2 Objetivos

- Fornecer uma opção barata compatível com a tecnologia atual (telecomunicações), considerando todas possíveis operações de telefonia.
- Criar uma rede de automação domiciliar.
- Melhorar a segurança domiciliar.

1.3 Diagrama em blocos

A figura 1 ilustra o diagrama em blocos do sistema Fly-IP. Através dele é possível compreender melhor o sistema e seus módulos integrantes.

A interface com o usuário será na grande maioria do tempo o seu próprio celular, através de um aplicativo de *SIP-phone*. O celular deverá estar conectado à internet, seja pela rede de casa, ou uma rede de fora, seja por uma conexão 2G, 3G, 4G, Wifi, etc. e deverá se registrar no seu servidor VoIP para que possa se comunicar com o resto do sistema.

O kit de desenvolvimento será utilizado como protótipo do servidor do sistema, pois pode-se instalar tanto um servidor VoIP (o PABX) quanto trabalhar com uma interface com a GPIO (*general purpose in/out pins*). Assim, há um núcleo responsável pelo PABX e sua interface de configuração e outro núcleo responsável pelos sensores/atuadores ligados.

Uma placa com drivers tanto para sensores quanto atuadores fará uma conexão com a GPIO. O sinal que os sensores enviam e os sinais que os atuadores recebem serão tratados diretamente pelas saídas do processador, portanto não é necessário se utilizar um protocolo de comunicação, apenas deixando o pino em tensão alta ou baixa para representar a informação.

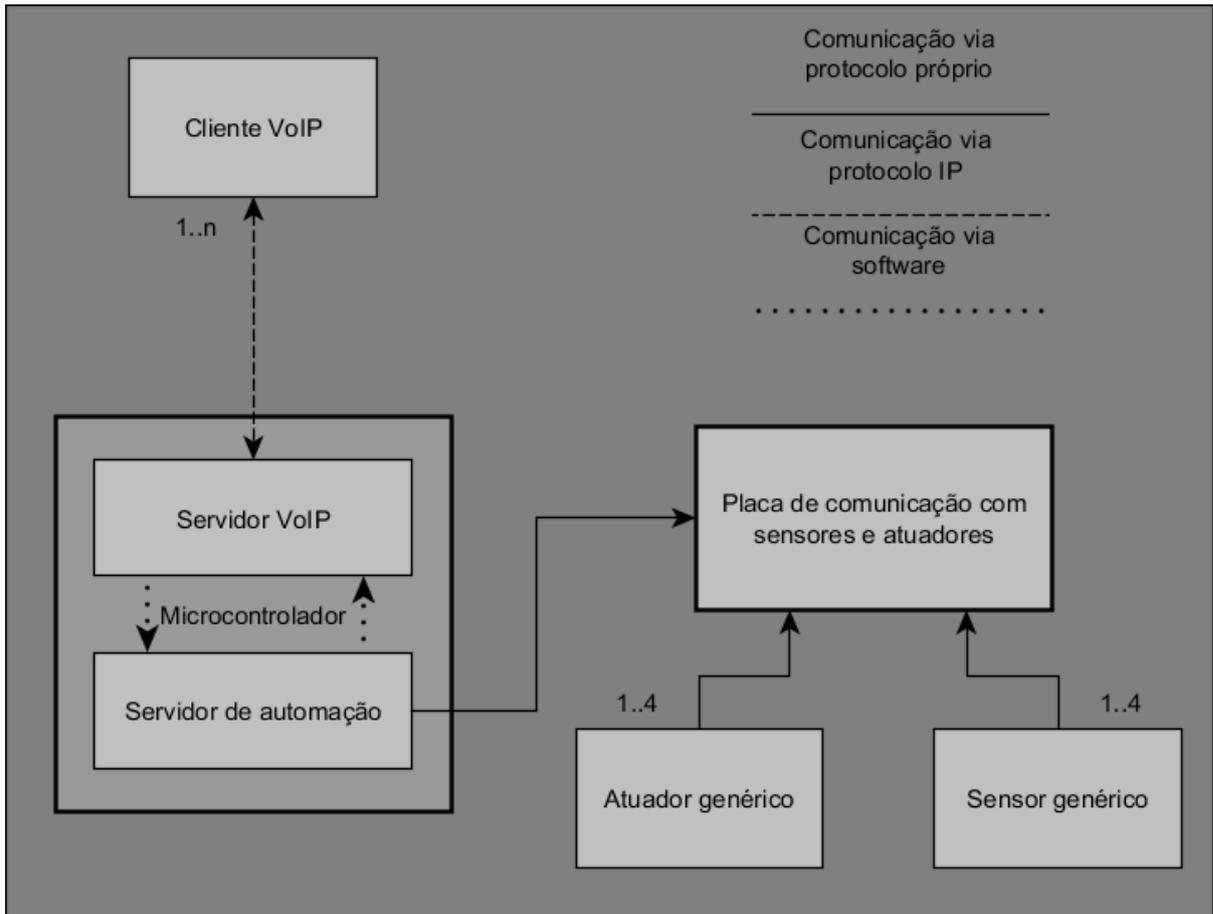


Figura 1 - Diagrama em blocos do projeto

Fonte: autoria própria

1.4 Resultados Esperados

Além de atingir os objetivos propostos, outros resultados também são esperados, tais como:

a) Tecnológicos:

Desenvolver uma solução completa de comunicação Voip que garanta a comunicação gratuita entre servidores.

b) Econômicos

Desenvolver uma solução de baixo custo que forneça ligações gratuitas entre servidores e ainda permita a automatização residencial através de módulos externos.

c) Sociais

Ligações gratuitas é uma forma de aproximar as pessoas, encurtar as distâncias e derrubar barreiras econômicas que possam impedir familiares distantes de se comunicarem.

b) Científicos

Não se aplica.

1.5 Metodologia

O desenvolvimento do projeto foi dividido em etapas, para que dessa forma seja possível ter um melhor controle da evolução do cronograma.

Etapa 1 - Especificar projeto

Foi realizado um estudo sobre a importância que o projeto poderia ter para os clientes potenciais e os usuários finais do produto. Assim, foi possível descobrir quais funcionalidades deveriam ser contempladas pelo produto para que ambas as demandas fossem atendidas. Dessa forma, o produto se tornaria ainda mais inovador e interessante em termos comerciais. Nessa etapa também foi realizada uma análise de viabilidade do produto no mercado.

Etapa 2 - Modelar soluções

Entendido o problema e qual o escopo do projeto, partiu-se então para a modelagem de tecnologias que poderiam atender as funcionalidades previstas. Assim, foi necessário estudar quais as opções de hardware para aquisição e processamento dos dados levando em conta o melhor custo-benefício. Foram elencados quais os sensores, microcontroladores e protocolos de comunicação são os mais adequados, assim como demais componentes do sistema a partir da análise de *datasheets*, catálogos ou mesmo testes. Essa etapa é importante, pois permite que existam planos alternativos para a solução do problema central.

Etapa 3 - Definir solução

Uma vez que foram pesquisadas diferentes tecnologias de interesse existentes no mercado, alcançou-se a etapa de definição da melhor forma dentre as encontradas para solucionar o problema. Essa definição foi resultado da análise da

viabilidade do produto e da complexidade da solução, ela foi indispensável para que se tivesse certeza das opções escolhidas para o desenvolvimento do projeto.

Etapa 4 - Implementar os módulos

Primeiramente os módulos do projeto foram tratados de forma independente. Assim, foi possível desenvolver as seções paralelamente para dar flexibilidade na execução do projeto e atenuar possíveis atrasos decorrentes de empecilhos no desenvolvimento de um dos módulos.

Etapa 5 - Integrar os módulos

Como os módulos do projeto foram tratados individualmente em um primeiro momento, após sua finalização foi necessário integrar todas as frações para constituir o sistema final. Essa integração requereu a devida atenção quanto à ergonomia e à estética do produto final a fim de que este tivesse um bom acabamento e fosse atrativo comercialmente.

Etapa 6 - Validar o protótipo

Com o produto finalizado foi realizada uma série de testes para descobrir o seu real comportamento nas diversas condições de uso. Para isso, o teste consistiu no uso do sistema em condições de stress por várias horas. Foram verificadas as partes problemáticas e feitas as devidas correções.

Etapa 7 - Finalizar

Foram feitos levantamentos acerca dos objetivos alcançados, dos modos como os eventuais obstáculos foram superados e dos pontos de melhoria a serem realizados. A partir disso, o projeto foi devidamente documentado na forma de relatório final e apresentado para a banca examinadora.

2. FUNDAMENTAÇÃO TEÓRICA

Um sistema PABX IP consiste de um ou mais telefones SIP, um servidor PABX IP e um gateway VoIP opcional para conectar-se as linhas da rede pública existentes. O servidor PABX IP funciona de forma similar a um servidor proxy: os clientes SIP, sendo tanto soft-fones quanto aparelhos IP's, registram-se com o servidor PABX IP, e quando eles desejam fazer uma chamada eles solicitam ao PABX IP estabelecer a conexão. (Wintermeyer, Bosch. 2009) [1]

O PABX IP tem arquivos com registro dos telefones/usuários e seus endereços SIP correspondentes, assim é capaz de conectar uma chamada interna ou rotear uma chamada externa tanto por um gateway VoIP como por uma operadora de serviço VoIP.

Para o desenvolvimento deste servidor PABX se pensou inicialmente em montar um computador dedicado inclusive com uma placa modem ADSL interno. Entretanto, como a oferta de modems ADSL internos é pequena, decidiu-se por utilizar apenas um computador dedicado.

Como o projeto prevê o desenvolvimento de um produto final, achou-se que a utilização de um computador completo tornaria o produto final muito caro, além de apresentar um tamanho desproporcional ao produto desejado. Dessa maneira, passou-se a pesquisar possíveis "Mini-PC's", como substitutos ao PC dedicado.

2.1 Mini PCs

Mini PC's são sistemas que englobam todos os periféricos que um computador igual ao que utilizamos em casa possui – mesmo que com uma potencia muito menor, em geral. Existem várias versões no mercado por um preço muito baixo, como a Raspberry Pi, a Beagle Board, a Panda Board, etc.

A escolha de um sistema mais propício para o desenvolvimento se dá baseando-se nos periféricos que serão utilizados no projeto, pois kits diferentes podem possuir funcionalidades diferentes, assim como potenciais diferentes em cada funcionalidade.

2.1.1 Beagle Board

A Beagle Board é um mini-Pc de Hardware Open-Source produzido pela Texas Instruments em associação com a Digi-key e a Newark Element14. Ela foi desenvolvido para ser utilizada com softwares livres de desenvolvimento e para demonstrar as capacidades do SoC (System-on-a-Chip) OMAP3530.

A placa foi desenvolvida como uma placa educacional que poderia ser utilizada no ensino das capacidades de Hardwares e Softwares livres.

A Beagle Board mede cerca de 7.5cm x 7.5cm e apresenta todas as funcionalidade de um computador de uso pessoal. Ela é construída com um SoC OMAP 3530 que inclui um processador ARM Cortex-A8 um DSP para aceleração de vídeo e decodificação de áudio e uma unidade de processamento gráfico (GPU) PowerVR SGX530 com suporte para OpenGL ES 2.0. Também possui um PoP Chip (*package on package*) que inclui 256MB de memória Flash e 256MB de memória RAM. Apresenta ainda saída de vídeo S-Video e HDMI, saída e entrada de áudio Stereo, suporte a cartão SD e porta Ethernet, USB e USB On-the-Go, além de conexão serial RS232 e conexão JTAG. (beagleboard.org) [2]

A Beagle Board é capaz de rodar Sistemas Operacionais como o Ubuntu, o Angstrom e o Android.

A figura 2 apresenta uma imagem da placa com descrição de seus periféricos.

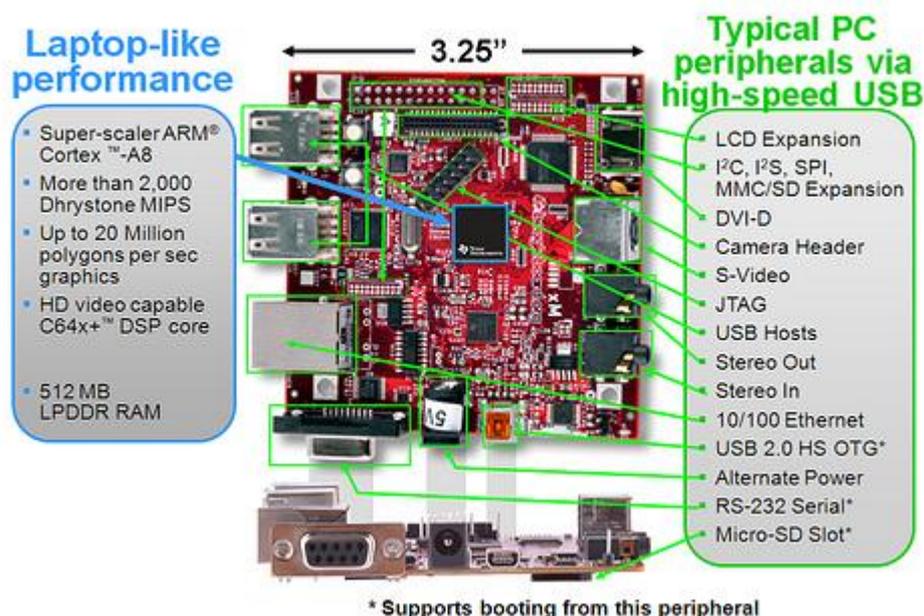


Figura 2 - Beagle Board

Fonte: beagleboard.org [2]

2.1.2 Panda Board

A Panda Board é um mini PC utilizado como plataforma de desenvolvimento. Ela utiliza o chip SoC OMAP4330 da Texas Instruments. A placa foi disponibilizada para o público com um preço subsidiado de US\$174,00 em 27 de outubro de 2010. A comunidade pandaboard.org é uma comunidade aberta responsável pelo suporte aos desenvolvedores que fazem uso da placa em seus projetos.

A Panda Board apresenta um processador ARM Cortex-A9 Dual-Core de 1GHz, uma unidade de processamento gráfico (GPU) PowerVR SGX540 a 304MHz, um acelerador de Hardware Multimídia com um DSP e ainda 1GB SDRAM DDR2, sendo capaz de gerar imagens pelas saídas DVI e HDMI a 1080p. Ela ainda possui conectividade Ethernet 10/100, wireless Ethernet e Bluetooth, além de possuir duas portas USB e uma porta USB on-the-go, suportando USB 2.0.

O dispositivo é capaz de rodar variadas distribuições Linux como o Debian e o Ubuntu, além de ter suporte para outros sistemas operacionais como o Android e o Mozilla Firefox OS. A fundação Linaro suporta a Panda Board com atualizações de imagens de distribuições Ubuntu e Android mensais.

Na figura 3 é possível conferir o guia de conexões da placa Panda Board.

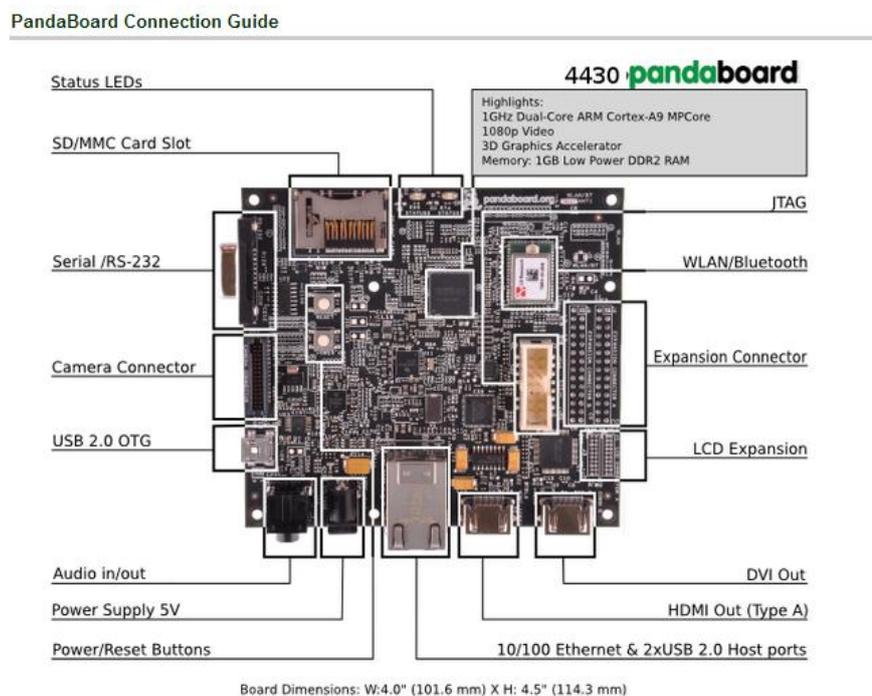


Figura 3 – Panda Board

Fonte: pandaboard.org [3]

2.1.3 Raspberry Pi

O Raspberry Pi é um mini-PC do tamanho de um cartão de crédito desenvolvido no Reino Unido na Universidade de Cambridge com a intenção de promover o ensino básico de computação e programação nas escolas.

Abaixo é possível conhecer o mini-PC Raspberry Pi.

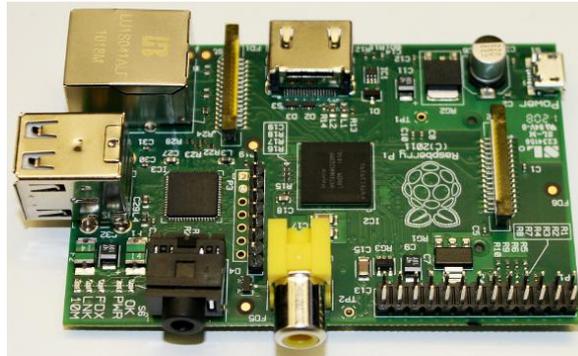


Figura 4 – Raspberry Pi

Fonte: raspberrypi.org [4]

O Raspberry Pi é equipado com um SoC Broadcom BCM2835, composto por um Processador Arm 11 rodando a 700MHz, com capacidade de Overclock por software de até 1GHz, uma GPU VideoCore IV que proporciona uma tecnologia Open GL ES 2.0, hardware acelerado OpenVG e admite imagens de alta resolução 1080p H.264. O SoC ainda possui 256MB de Memória RAM na primeira versão, e 512MB de RAM na versão atual (para o modelo B da placa). A placa utiliza cartões SD para Boot do sistema operacional e para armazenamento de dados.

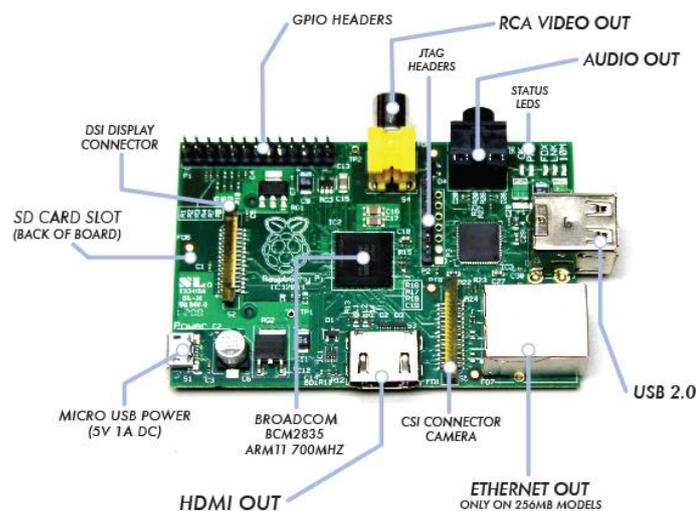
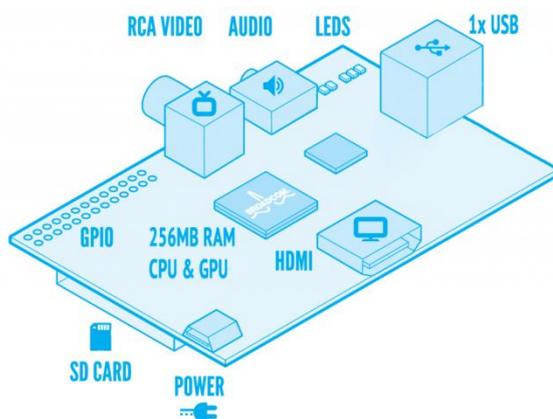


Figura 5 - Raspberry Pi - Características

Fonte: pcmag.com [5]

O que chama mais atenção neste mini-PC é seu preço: US\$25,00 para o modelo A da placa e US\$35,00 para o modelo B. Apesar do modelo A ser dez dólares mais barato, a diferença entre os dois modelos não é muito grande: o Modelo A tem 256 MB de RAM, uma porta USB e não tem porta de Ethernet; o Modelo B atualmente tem 512 MB de RAM, duas portas USB e uma porta de Ethernet. Além disso, ambos os modelos contam com porta HDMI, slot para cartão de memória SD, saída de áudio de 3,5 mm, e foram projetados para rodar Linux. O modelo B está disponível no mercado desde 29 de fevereiro de 2012 enquanto que o modelo A só foi disponibilizado em 4 de fevereiro de 2013.

RASPBERRY PI MODEL A - \$25



RASPBERRY PI MODEL B - \$35

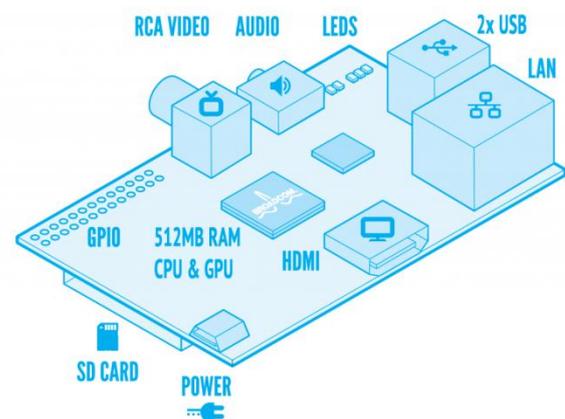


Figura 6 – Comparação entre modelos A e B

Fonte: hackthings.com [6]

O Raspberry Pi apresenta inúmeras versões de Sistemas Operacionais com versões compiladas especificamente para o dispositivo que podem ser encontradas diretamente na comunidade de suporte raspberrypi.org. Esta comunidade é atualizada diariamente com ideias de projetos e notícias e possui guias para novos usuários e seção de download de versões de Sistemas Operacionais variados.

2.2 O sistema VoIP

Os parâmetros para escolha de um servidor VoIP foram a quantidade de funcionalidades, facilidade de integração e preço. O escopo deste projeto é domiciliar, portanto o servidor não estará exposto a uma carga de processamento muito pesada, logo não precisamos considerar como um parâmetro a velocidade de

processamento de dados. Em geral servidores de telecomunicações são rápidos o suficiente para este escopo, pois são utilizados como *call-centers* por empresas maiores.

O servidor escolhido é o Asterisk, pois se encaixa em todos os parâmetros de escolha, possui várias funcionalidades, possui arquivos de configuração que tornam a integração simples seja com o sistema operacional ou entre métodos de comunicação diferentes (ex.: comunicação entre protocolos VoIP e protocolos PSTN) e por ser um software livre, é gratuito e seu código é aberto. (asterisk.org) [7]

2.3. Asterisk

O software Asterisk inclui muitos recursos disponíveis em sistemas de PABX: correio de voz, chamada em conferência, resposta de voz interativa (menus do telefone), e distribuição automática de chamadas.

É possível criar novas funcionalidades por scripts do plano de discagem diretamente nos arquivos de configuração do Asterisk, adicionando módulos carregáveis personalizados escritos em C, ou através da implementação de programas de Asterisk do Gateway Interface (AGI), utilizando qualquer linguagem de programação capaz de se comunicar através do sistema de stream padrão (stdin e stdout) ou por rede sockets TCP. (BRYANT, MADSEN, MEGGELEN, p.9. 2013) [8]

Asterisk suporta uma ampla gama de protocolos VoIP, incluindo o SIP, o Media Gateway Control Protocol (MGC) e H.323. O Asterisk pode interoperar com a maioria dos telefones SIP, atuando tanto como secretário e como um gateway entre telefones IP e PSTN. As taxas do protocolo Inter-Asterisk (IAX2), RFC 5456, nativa do Asterisk, fornecem *trunking* eficiente de chamadas entre PABXs Asterisk, além da lógica de configuração distribuída.

Ao apoiar um mix de serviços de telefonia tradicional e VoIP, o Asterisk permite construir novos sistemas de telefonia, ou gradualmente migrar os sistemas existentes às novas tecnologias. Alguns sites estão usando servidores Asterisk para substituir PABX proprietários, outros para fornecer recursos adicionais (como correio de voz ou menus de resposta de voz, ou lojas virtuais de chamadas), ou para reduzir os custos de transporte de chamadas de longa distância através da Internet (ligação bypass).

Além dos protocolos VoIP, o Asterisk suporta muitos protocolos de comutação de circuito tradicionais, tais como ISDN e SS7 . Isso requer hardware de interface adequado para apoiar tais protocolos, comercializados por vendedores de terceiros. Cada protocolo requer a instalação de módulos de software.

2.4 Sip Phone

Telefones SIP são semelhantes a telefones VoIP ou softphones. São telefones que permitem fazer chamadas telefônicas através da tecnologia VoIP (*voice over internet protocol*).

O Protocolo SIP (*Session Initiation Protocol*) é um protocolo de comunicação para início de sessão amplamente utilizado no controle de sessões de comunicação multimídia, como chamadas de voz e vídeo VoIP. O protocolo define as mensagens que são enviadas entre os pares que regem a criação, extinção e outros elementos essenciais de uma chamada. O protocolo SIP pode ser usado para criar, modificar e terminar sessões entre pares (unicast) ou sessões entre múltiplos pares (multicast) constituídos por um ou vários fluxos de mídia. (BRYANT, MADSEN, MEGGELEN, p.740. 2013) [8]. Outras aplicações SIP incluem videoconferência, streaming de distribuição multimídia, mensagens instantâneas, informações de presença, transferência de arquivos, fax sobre IP e jogos online.

Existem dois tipos de telefones SIP. O primeiro tipo é o hardware do telefone SIP, que se assemelha ao telefone comum, mas pode fazer e receber chamadas usando a internet ao invés do sistema tradicional PSTN.

Telefones SIP também podem ser desenvolvidos em software. Estes permitem que qualquer computador possa ser utilizado como um telefone mediante o uso de um microfone e uma saída de áudio. Também é necessário a conexão de banda larga e a conexão do SIP Phone com um provedor VoIP ou um servidor SIP.

Existe uma grande variedade de SIP Phones no mercado, sejam estes por software ou por hardware. Dentre os SIP Phones por hardware pode-se citar exemplos como os das marcas: Cisco, Linksys, Aastra e LinkPro. Com relação aos SIP Phones por software, existe grande variedade disponível, inclusive ótimos exemplos em software livre. Pode-se citar como exemplos de licenças proprietárias softwares como o 3Com, 3CXPhone, AskoziaPBX, Dialogic, XLite; e como exemplo de licenças gratuitas e de código aberto softwares como Blink, Jitsi, Linphone e

Twinkle. Muitas destes softwares estão disponíveis tanto para computadores de uso pessoal quanto para tablets e celulares.



Figura 7 – Modelos SIP Phone por Hardware (Astra 6739i) e por Software (3CX Phone)

Fonte: 888voip.com [9] e 3cx.com [10]

3. METODOLOGIA

Para iniciar o desenvolvimento da solução de comunicação VoIP era necessária escolher qual plataforma utilizar, assim como quais configurações de Hardware escolher para o desenvolvimento do servidor PABX.

3.1 Escolhendo um Mini-PC

Para iniciar o planejamento da solução do problema proposta era essencial escolher a plataforma sobre a qual se desenvolveria o projeto, motivo pelo qual foram estudadas possíveis soluções, dentre as quais citadas acima. Tendo delimitado o campo de escolhas, fez-se necessário analisar prós e contras de cada placa para selecionar qual placa seria utilizada.

As três soluções apresentam tamanhos reduzidos, de maneira que esta não seria uma boa métrica avaliativa de escolha, entretanto, as 3 placas apresentam poder de processamento dispare, e custos que estão relacionados ao mesmo. Como o grande objetivo era desenvolver um projeto que viesse a se tornar um futuro produto, passou-se a analisar a real necessidade de quanto poder de processamento seria necessário. Sabia-se se com as três placas era possível desenvolver sobre plataformas Linux embarcadas, o que já seria suficiente para desenvolver o projeto. Sendo assim, não faria sentido escolher a opção da Panda Board pelo fato da mesma apresentar um processador ARM Cortex A9 dual core e nem a Beagle Board com seu ARM Cortex A8, uma vez que o processador ARM 11 do Raspberry Pi seria mais do que suficiente e apresentaria um custo benefício muito alto, pois o mesmo apresenta custos subsidiados que tornam atraente a placa ao público.

Selecionou-se, dessa maneira, o Raspberry Pi como plataforma de desenvolvimento. A escolha se mostrou muito feliz uma vez que a placa havia sido recém-lançada e já apresentava uma comunidade desenvolvedora em constante crescimento. Esta comunidade desenvolvedora se mostrou de grande utilidade durante as pesquisas para desenvolvimento do projeto.

3.2 Distribuição Linux – Debian

Tendo escolhido a plataforma de desenvolvimento, passou-se a identificar qual sistema operacional suportado pela placa seria mais interessante para o desenvolvimento do projeto. Os SOs disponíveis e compatíveis eram: Debian Gnu/Linux; Fedora Linux; Arch Linux Arm; e o RISC OS. Como os desenvolvedores do projeto já haviam trabalhado com distribuições Debian GNU/Linux para computadores pessoais, achou-se prudente escolher este sistema operacional para o desenvolvimento da solução.

O Sistema Operacional Debian GNU/Linux é o nome de uma distribuição não comercial gratuita e de código aberto de GNU/Linux. Trata-se de um conjunto de programas básicos e utilitários responsável pelo funcionamento de um computador. (debian.org) [11].

No núcleo deste Sistema Operacional está o Kernel, é o componente central do sistema operativo da maioria dos computadores; ele serve de ponte entre aplicativos e o processamento real de dados feito a nível de hardware. As responsabilidades do núcleo incluem gerenciar os recursos do sistema (a comunicação entre componentes de hardware e software).

O Debian utiliza atualmente o Kernel Linux, núcleo monolítico criado inicialmente por Linus Torvalds com a ajuda de milhares de programadores espalhados por todo o mundo.

3.3 Escolha do Python como linguagem de desenvolvimento

O ambiente de desenvolvimento do sistema de automação a ser integrado com o sistema VoIP pode ser o mais simples para o contexto deste projeto, portanto será utilizada a linguagem Python. A escolha do Python ocorreu devido à sua popularidade na comunidade dos usuários do Raspberry Pi, uma vez que existem bibliotecas disponibilizadas para acesso direto à GPIO por scripts em Python.

A linguagem Python é bastante simples, pode-se trabalhar diretamente com a GPIO com scripts de apenas algumas linhas. É uma vantagem quanto a trabalhar diretamente com scripts shell do Linux, pois possuem uma linguagem mais complicada e um tratamento mais complexo/completo da GPIO. Conforme será descrito posteriormente, o servidor VoIP escolhido possui uma integração simples com scripts shell, portanto a solução escolhida para o sistema foi montar uma base de integração em script shell que comunica os scripts Python com o servidor VoIP.

Por ser uma linguagem de alto nível, existem algumas desvantagens, como por exemplo, criar um sistema de comunicação através da GPIO, pois na biblioteca disponibilizada não existem definições para pinos Rx/Tx de UART ou SDA/SCL para comunicação I2C. Para o escopo do problema, isto não chega a ser um problema, visto que a GPIO será utilizada apenas para ligar/desligar saídas ou ler entradas digitais.

3.4 Desenvolvimento do Sistema de Comunicação VoIP

3.4.1 Instalação do sistema

O Raspberry Pi não possui memória flash, de maneira que tanto o sistema operacional quanto os arquivos devem ser guardados no cartão SD. Além disso, o mesmo não irá iniciar sem que esteja inserido um cartão SD contendo um bootloader e um sistema operacional.

3.4.2 Instalação do Debian no Cartão SD

A comunidade raspberrypi.org oferece múltiplas imagens de sistemas operacionais compiladas para o processador do raspberry pi e otimizadas para seu hardware, de maneira que não era vantagem fazer uma compilação específica do sistema operacional escolhido. Dessa maneira, tendo optado pela utilização do Debian GNU/Linux como Sistema Operacional, bastou baixá-lo do site em questão e seguir os passos para a sua instalação no cartão SD.

A comunidade recomenda a utilização de um cartão SD de no mínimo 2GB. Como ainda seria necessário realizar várias instalações e não se tinha previsão do tamanho final que o sistema operacional e as instalações ocupariam no cartão SD, optou-se por utilizar um cartão de 8GB, o que foi uma ótima escolha, uma vez que esses tipos de cartões tem se tornado cada vez mais baratos.

A versão do Debian GNU/Linux utilizada é a 7, conhecida como “wheezy”. Por ser uma versão compilada e otimizada para Raspberry Pi, essa versão do Debian é conhecida como “Raspbian”.

Existem inúmeras maneiras de fazer a instalação do SO no cartão SD. A maneira escolhida pelo grupo foi a utilização do programa Win32DiskImager, que pode ser conferido Figura 8.

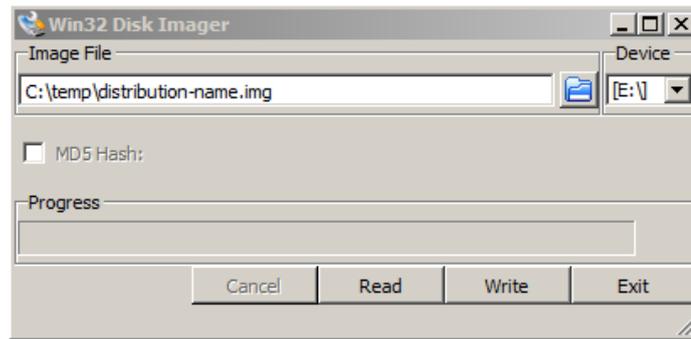


Figura 8 – Win32 Disk Imager

Fonte: elinux.org [12]

Após baixar a imagem e extraí-la, deve-se descompactá-la, gerando um arquivo com o nome do Sistema operacional baixado: “*Raspbian.img*”. Após isso, insere-se o cartão SD no computador e confere-se que letra de driver está indexando o dispositivo. Abre-se o programa Win32DiskManager como administrador e seleciona-se a imagem do sistema operacional baixado e o driver correto do cartão SD inserido. Basta, portanto, clicar com o botão esquerdo do mouse no botão “write” do programa, para preparar o SD para ser utilizado pelo Raspberry Pi.

Tendo executado corretamente a gravação do cartão SD, o Raspberry Pi está pronto para funcionar como um mini-PC rodando Linux.

3.4.3 Instalação do Asterisk

Antes de fazer a instalação do Asterisk, fez-se uma atualização do Raspberry com bibliotecas e programas úteis a se trabalhar com o Asterisk, conforme a tabela1:

Tabela 1 - Update do Raspberry e instalação de programas úteis.

```
sudo apt-get update

sudo apt-get -y install make gcc g++ libxml2 libxml2-dev ssh libncurses5
libncursesw5 libncurses5-dev libncursesw5-dev linux-libc-dev sqlite libnewt-dev
libusb-dev zlib1g-dev libmysqlclient15-dev libsqlite0 libsqlite0-dev bison openssl
libssl-dev libeditline0 libeditline-dev libedit-dev mc sox libedit2 libedit-dev
curl libcurl4-gnutls-dev apache2 libapache2-mod-php5 php-pear openssh-server build-
essential openssh-client zlib1g zlib1g-dev libtiff4 libtiff4-dev libnet-telnet-perl
mime-construct libipc-signal-perl libmime-types-perl libproc-waitstat-perl mpg123
libiksemel-dev php5 php5-cli mysql-server php5-mysql php-db libapache2-mod-php5
php5-gd php5-curl mysql-client vim
```

Fonte: (BRYANT, MADSEN, MEGGELEN, p.37. 2013) [8]

Após a atualização, faz-se o download e a extração do pacote do Asterisk, conforme instruções abaixo:

Tabela 2 - Download e extração do Asterisk

```
cd /usr/src
sudo wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8-
current.tar.gz
sudo tar xvfz asterisk-1.8-current.tar.gz
cd asterisk-1.8*
```

Fonte: (BRYANT, MADSEN, MEGGELEN, p.38. 2013) [8]

Utilizando um editor de texto, faz-se alterações no arquivo “makeopts.in” para configurar o Asterisk para ser instalado em plataforma ARM. Para isso, basta alterar a opção “proc=” do arquivo para “proc=arm”, e depois salvá-lo. Tendo realizado a alteração, deve-se compilar o Asterisk e depois instalá-lo e reiniciar o Raspberry conforme as instruções seguintes:

Tabela 3 - Configuração e instalação do Asterisk

```
sudo ./configure
sudo make
sudo make install
sudo make samples
sudo make config
sudo init 6
```

Fonte: (BRYANT, MADSEN, MEGGELEN, p.39. 2013) [8]

Para testar a instalação, basta utilizar o comando “sudo asterisk-r”.

Além do Asterisk, também foi instalado o FreePBX, uma interface gráfica gratuita de código aberto que permite fazer configurações no servidor PBX através da janela de um browser de internet. (BRYANT, MADSEN, MEGGELEN, p.675. 2013) [8]. O FreePBX utiliza repositório PHP Pear, de maneira que o mesmo deve ser instalado:

Tabela 4 - Instalação do repositório PHP Pear

```
sudo apt-get install php-pear
sudo pear install db
```

Fonte: Fonte: freepbx.org [13]

Para baixar e extrair o FreePBX usa-se os comandos:

Tabela 5 - Baixando e extraindo o FreePBX

```
cd /usr/src/
sudo wget http://mirror.freepbx.org/freepbx-2.9.0.tar.gz
tar zxvf freepbx*
```

Fonte: freepbx.org [13]

Feito isso, pode-se iniciar o banco de dados e importar as tabelas que o FreePBX fará uso juntamente com o Asterisk:

Tabela 6 - Inicialização do Banco de Dados

```
sudo service mysql start
cd /usr/src/freepbx*

sudo mysqladmin create asterisk
sudo mysqladmin create asteriskcdrdb
sudo mysql asterisk < SQL/newinstall.sql
sudo mysql asteriskcdrdb < SQL/cdr_mysql_table.sql
sudo mysql
```

Fonte: Fonte: freepbx.org [13]

No *prompt* do MySQL adiciona-se o novo usuário, conferindo-se os corretos privilégios ao mesmo:

Tabela 7 - Criando Usuário do MySQL

```
GRANT ALL PRIVILEGES ON asteriskcdrdb.* TO asteriskuser@localhost IDENTIFIED BY
'MyPassWord';
GRANT ALL PRIVILEGES ON asterisk.* TO asteriskuser@localhost IDENTIFIED BY
'MyPassWord';
flush privileges;
exit
```

Fonte: Fonte: freepbx.org [13]

Adiciona-se uma palavra chave de segurança ao MySQL para torná-lo seguro:

Tabela 8 - Protegendo MySQL com Palavra Chave

```
sudo mysqladmin -u root password 'MyRootPassWord'
```

Fonte: freepbx.org [13]

Reinicializa-se o Asterisk com as corretas permissões:

Tabela 9 - Reinicializando o Asterisk

```
cd /usr/src/freepbx*
sudo service asterisk stop
sudo ./start_asterisk start
```

Fonte: (BRYANT, MADSEN, MEGGELEN, p.38. 2013) [8]

Configura-se o Apache para trabalhar com o Asterisk e o FreePBX. Basta abrir o arquivo `/etc/apache2/apache.conf` com um editor de texto e editar as opções 'user' e 'group' conforme exemplo abaixo:

Tabela 10 - Configurando Apache

```
User asterisk
Group asterisk
```

Fonte: freepbx.org [13]

Reinicia-se o Apache com o comando:

Tabela 11 - Reiniciando o Apache

```
sudo service apache2 restart
```

Fonte: freepbx.org [13]

Após o Apache ser reiniciado, pode-se instalar o FreePBX utilizando o nome de usuário e senha previamente criado com o banco de dados.

Tabela 12 - Instalação do FreePBX

```
sudo ./install_amp --username=asteriskuser --password=MyPassWord
```

Fonte: <http://moishtech.blogspot.co.uk/2012/06/setup-and-run-asterisk-and-freepbx-on.html>

Tendo completado esse processo, confere-se a correta permissão para que o FreePBX possa ser acessada por usuários Apache.

Tabela 13 - Configurando permissões de acesso ao FreePBX por usuários Apache.

```
sudo chmod 777 /var/www/* -R
```

Fonte: freepbx.org [13]

Finalmente, para garantir o correto funcionamento da instalação reinicializa-se os processos do Apache, do MySQL e do Asterisk. A partir deste ponto o Asterisk e o FreePBX está configurado e pronto para uso no Raspberry.

Acessando a página de configuração por um browser em algum computador da rede a seguinte tela do FreePBX deverá aparecer, indicando que todos os processos correspondentes estão instalados e em funcionamento:

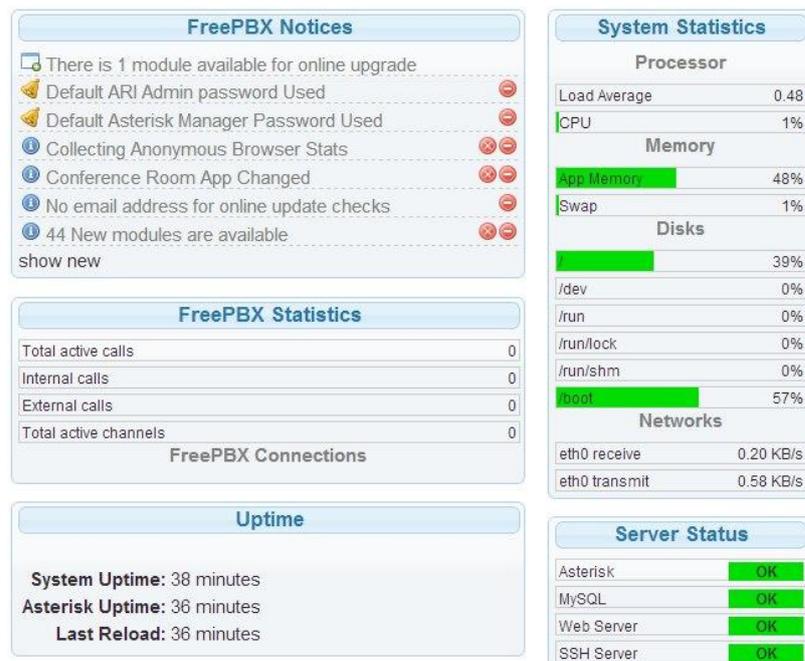


Figura 9 - Página Inicial do FreePBX

Fonte: Autoria Própria

3.4.4 Criando SipPhones no Asterisk

A ideia de utilizar o Free PBX no Servidor PBX, surgiu com o intuito de utilizar um “front-end” para o usuário poder criar novos SIP Phones à medida que fosse se tornando necessário, diretamente pelo Browser de seu computador. O processo é de fácil aprendizagem e segue os passos a seguir.

Figura 10 – Adicionando Extensão no FreePBX

Fonte: Autoria Própria

Figura 11 - Adicionando Extensão SIP ao FreePBX

Fonte: Autoria Própria

Como pode ser visto pelas Figuras 10 e 11, adicionar uma extensão SIP ao Asterisk através do FreePBX é um processo fácil. Basta acessar o ip do Raspberry em algum browser, *logar* no servidor e, conforme a Figura 10, selecionar o botão “application”, depois “add extension”, e marcar a opção “Generic SIP Device”. Feito

isso, a configuração no novo número SIP pode ser feita conforme a Figura 11, sendo essencial apenas inserir um novo número na opção “User Extension” e adicionar um nome em “Display Name”.

3.4.5 Configurações do Asterisk

A configuração do Asterisk é distribuída em vários arquivos com extensões diferentes, existem configurações referentes a vários tipos de funcionalidades diferentes. Para o escopo deste projeto, serão utilizados três tipos de configurações: “*outgoing calls*”, *dialplan* e “*call trunks*”. (voip-info.org) [14]

A estrutura *dialplan* serve para configurar scripts internos do Asterisk dado um contexto de ligação (contexto de ligação serve para separar grupos dentro de um mesmo servidor Asterisk, ou seja, para dar privilégios diferentes a grupos diferentes). Neste projeto, considera-se apenas o contexto de ligações internas, portanto, todos telefones SIP que estiverem cadastrados no servidor terão os mesmos privilégios. As configurações se encontram no arquivo chamado “*extensions.conf*”, e é lá que são montadas as estruturas que chamam Shell scripts referentes aos atuadores.

Tabela 14 - exemplo de configuração de dialplan: quando um usuário chamar o número 123 no contexto [Interno], a extensão 123 responderá.

```
[Interno]
exten => 123,1,Answer
exten => 123,2,Playback(hello-world)
exten => 123,3,wait(1)
exten => 123,4,Hangup
```

Fonte: Autoria própria

As “*outgoing calls*” são chamadas originadas no próprio Asterisk direcionadas a algum telefone. Esta estrutura é usada a partir da criação de arquivos de extensão “*.call” que são colocados na pasta chamada “/var/spool/asterisk/outgoing”, logo após o script de chamada ser copiado a esta pasta, ele é executado. Esta estrutura serve para tratar informações vindas dos sensores.

Tabela 15 – exemplo de configuração de chamada automática: quando este script for colocado na pasta “/outgoing”, o sistema irá gerar uma chamada para a extensão 1000 e tocará o som “hello world” quando a ligação for atendida.

```
Channel: SIP/1000
Application: Playback
Data: hello-world
```

Fonte: Autoria própria

Os “*call trunks*” servem para configurar a comunicação entre dois ou mais servidores Asterisk. Estas configurações são feitas no arquivo “/etc/asterisk/sip.conf” e servem para configurar qualquer tipo de comunicação com servidores ou telefones externos.

Tabela 16 - neste exemplo de configuração de trunk, está sendo habilitado que os telefones internos do servidor A liguem para o servidor B, que deverá ter uma configuração para receber estas ligações conforme vontade do usuário.

```
[AsteriskServerB_Trunk]
type=peer
fromdomain = [AsteriskServerA IP]
host = [AsteriskServerB IP]
outboundproxy = [AsteriskServerB IP]
context = Interno
```

Fonte: Autoria própria

3.5 Desenvolvimento do Sistema de Automação Domiciliar

3.5.1 Instalação do Python

O Python é uma linguagem interpretada de alto nível, orientada a objeto, muito atrativa para aplicações rápidas assim como para desenvolvimento de scripts e para servir de ligação entre variados programas diferentes. Python é uma linguagem simples e de fácil aprendizado que torna fácil o desenvolvimento e a manutenção de softwares.

Para iniciar a programar em python no Raspberry Pi é necessário fazer a instalação de alguns pacotes, que devem ser instalado no terminal do debian em modo super usuário (usando o comando sudo). Abaixo segue listagem dos pacotes a serem baixados assim como forma de baixá-los e instalá-los:

Tabela 17 - Instalação do Python.

```
sudo apt-get install python
sudo apt-get install python-dev
sudo apt-get install libjpeg-dev
sudo apt-get install libfreetype6-dev
sudo apt-get install python-setuptools
sudo apt-get install python-pip
```

Fonte: Autoria Própria

Após essas instalações o Raspberry já está preparado para interpretar códigos python através do comando `sudo python programa.py`.

3.5.2 Instalação da biblioteca de acesso ao GPIO do Raspberry.

General Purpose Input/Output (GPIO) são basicamente portas programáveis de entrada e saída de dados. São utilizadas para prover uma interface entre os periféricos e os microcontroladores/microprocessadores.

O Raspberry Pi permite que periféricos e placas de expansão se conectem ao processador através de seus pinos de GPIO. Os níveis de tensão dos pinos são de 3.3V e não de 5V, e como a placa não possui proteção, deve-se utilizar o GPIO com cuidado.

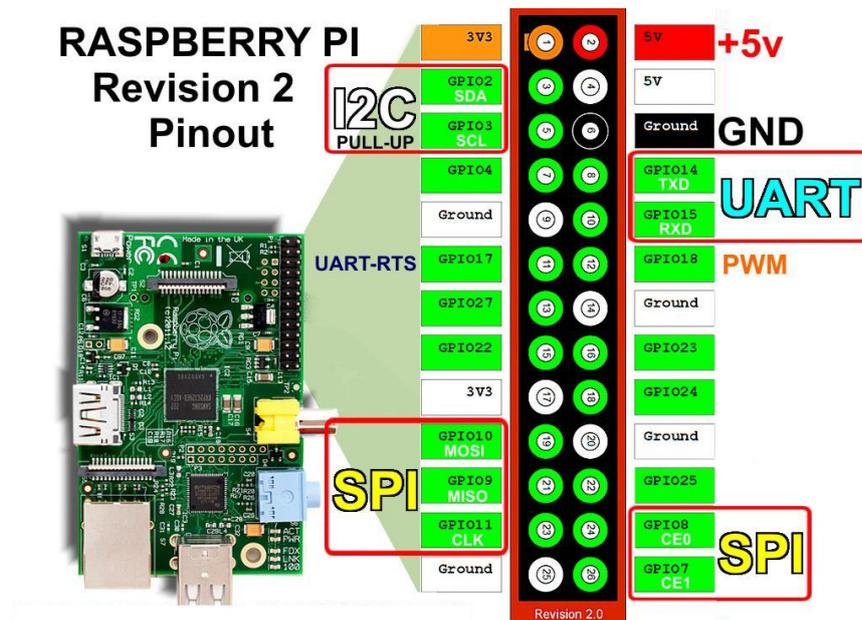


Figura 12 – Raspberry Pi GPIO

Fonte: raspberrypi.org [4]

Existem bibliotecas que permitem a utilização desses pinos no desenvolvimento de programas que rodem sobre o Linux. Como o desenvolvimento do projeto está baseado na linguagem python, utilizou-se do PIP, uma ferramenta para instalar pacotes python para fazer a instalação do Rpi.GPIO, uma biblioteca que permite o acesso aos pinos. Essa e outras bibliotecas podem ser instaladas no terminal do Debian através dos seguintes comandos:

Tabela 18 - Instalação da Bibliotecas de acesso ao GPIO.

```
sudo pip install RPi.GPIO
sudo pip install pySerial
sudo pip install nose
sudo pip install cmd2
```

Fonte: Autoria Própria

3.5.3 Interface com sensores e atuadores

Tendo realizado os testes de comunicações entre os sip-phones, iniciou-se o desenvolvimento da segunda parte do projeto, relacionada com automação domiciliar utilizando o próprio servidor PBX. Dessa maneira, pesquisou-se possíveis sensores e atuadores que seriam úteis para demonstrar tais capacidades do servidor. A ideia escolhida foi relacionada a segurança doméstica, mas variados módulos podem ser acoplados ao GPIO do Raspberry para situações variadas de automação.

3.5.4 Sensor de Presença PIR

O Sensor PIR (*Passive Infrared*) é um sensor Piroelétrico (detector de calor) que consegue sentir variações de calor através da análise de variações no espectro infravermelho.

Todos os objetos emitem energia em forma de radiação. Normalmente esta radiação é invisível ao olho humano, uma vez que ela ocorre no espectro infravermelho, entretanto, pode ser detectado por componentes eletrônicos construídos para tal propósito. Esses componentes funcionam através da detecção da energia recebida de outros objetos. É importante explicar que os sensores PIR não fazem medições de calor, mas como detectam a radiação infravermelha, pode-se fazer um paralelo com a temperatura do objeto.

O sensor PIR, como o nome já diz, é um sensor passivo, o que significa que ele não possui uma fonte de infravermelho própria.

Como todos os objetos emitem calor na forma de radiação infravermelha, o que o sensor PIR faz é comparar o valor lido com um valor pré-definido para temperaturas normais. Quando este detecta uma radiação infravermelha acima do valor pré-definido, o sensor emite o aviso de presença, que na maioria dos casos trata-se apenas de um pino de saída indicando nível alto ou baixo.

Sensores PIR são fabricados de um material cristalino que libera cargas elétricas quando exposto a radiação infravermelha.

Sensores PIR com o objetivo de detectar presença, costumeiramente são produzidos com filtros especiais chamados de lentes de Fresnel que focalizam a radiação infravermelha sobre o sensor PIR.



Figura 13 - Sensor PIR

Fonte: Tato Equipamentos Eletrônicos. [15]

O Sensor escolhido possui três pinos, sendo dois de alimentação (Vcc e Gnd) e um de saída. Quando o sensor PIR indica presença, a saída gera sinal em nível lógico alto, senão continua em nível lógico baixo.

Para o correto funcionamento do circuito, desenvolveu-se o programa em python a seguir, que quando detecta alguma presença, liga a saída do Buzzer e abre o script responsável por efetuar uma chamada de aviso para a extensão SIP pré-configurada.

Tabela 19 - Script python de controle do Sensor PIR

```
# Import required Python libraries
import RPi.GPIO as GPIO
import time
import subprocess

# Use BCM GPIO references
# instead of physical pin numbers
GPIO.setmode(GPIO.BCM)

# Define GPIO to use on Pi
GPIO_PIR = 7

print "PIR Module Test (CTRL-C to exit)"

# Set pin as input
GPIO.setup(GPIO_PIR,GPIO.IN)      # Echo

Current_State = 0
Previous_State = 0

try:

    print "Waiting for PIR to settle ..."

    # Loop until PIR output is 0
    while GPIO.input(GPIO_PIR)==1:
        Current_State = 0
```

```

print " Ready"

# Loop until users quits with CTRL-C
while True :

    # Read PIR state
    Current_State = GPIO.input(GPIO_PIR)

    if Current_State==1 and Previous_State==0:
        # PIR is triggered
        print " Motion detected!"
        #set buzzer
        GPIO.output(11,True)
        # call to sip phone
        p = subprocess.Popen('./script.sh', shell=True)
        # Record previous state
        Previous_State=1
    elif Current_State==0 and Previous_State==1:
        # PIR has returned to ready state
        print " Ready"
        Previous_State=0
        time.sleep(0.01)

except KeyboardInterrupt:
    print " Quit"
    GPIO.cleanup()

```

Fonte: Autoria Própria

A chamada é realizada através do Shell script a seguir:

Tabela 20 - Shell Script para realização de chamada de aviso de falha de segurança

```

#-----
cat << EOF > /tmp/hello.call
Channel: SIP/1000
Application: Playback
Data: hello-world
AlwaysDelete: yes
EOF
chown asterisk:asterisk /tmp/hello.call
mv /tmp/hello.call /var/spool/asterisk/outgoing/
#-----

```

Fonte: Autoria Própria

3.5.5 Atuadores para alerta de falha de segurança

Existem dois atuadores diferentes para controle de segurança, um primeiro que serve para ativar/desativar a leitura do sensor PIR, e um segundo que serve para ativar/desativar alerta sonora local de falha de segurança. Ambos os atuadores possuem um controle simples, pois em *hardware* são constituídos apenas por um transistor cada um, que opera em região de corte ou região ativa conforme a polarização.

As implementações foram feitas no *dialplan* do Asterisk e em scripts Python para lidar diretamente com a GPIO. As implementações estão descritas nas tabelas 21, 22 e 23.

Tabela 21 - Tratamento das ligações para ativar/desativar atuadores (extensão 660 para ativar e 661 para desativar)

```
;extensions.conf >> dialplan

[from-internal]
exten => 660,1,Answer
exten => 660,n,Playback(activated)
exten => 660,n,System(/usr/local/bin/evento.sh start)
exten => 660,n,Hangup()

exten => 661,1,Answer
exten => 661,n,Playback(de-activated)
exten => 661,n,System(/usr/local/bin/evento.sh stop)
exten => 661,n,Hangup()
```

Fonte: autoria própria.

Tabela 22 - Chamada do shell script que integra dialplan com Python.

```
#!/bin/sh

case $1 in
start)
sudo python /tmp/gpio4.py
;;
stop)
sudo python /tmp/gpio5.py
;;
esac
```

Fonte: autoria própria.

Tabela 23 - Scripts Python de tratamento da GPIO, pino 24.

```
%gpio4.py
import RPi.GPIO as GPIO
GPIO.setup(24, GPIO.OUT)
GPIO.output(24, False)

%gpio5.py
import RPi.GPIO as GPIO
GPIO.setup(24, GPIO.OUT)
GPIO.output(24, True)
```

Fonte: autoria própria.

As tabelas acima descrevem todo o tratamento feito via software do atuador que habilita o sensor de movimento. O usuário deve executar uma ligação para a extensão 660 para habilitar ou 661 para desabilitar.

O atuador que habilita o buzzer, para avisar localmente quando há falha na segurança, possui um tratamento análogo. Para habilitá-lo deve-se executar uma ligação para a extensão 760 para habilitar ou para 761 para desabilitar, e está alocado na GPIO 23.

3.5.6 Indicador de Funcionamento do Sensor

Para indicar que o sensor está em funcionamento, inseriu-se um led em paralelo com a entrada do “gate” do transistor responsável por chavear o sensor. Dessa maneira, ao ativar o sensor através da chamada VoIP, um led também indicará que o sensor de presença está ativo.

3.6 Desenvolvimento do hardware

Tendo-se planejado a solução e decidido que sensores e atuadores utilizar, avaliou-se a melhor disposição dos pinos de GPIO do Raspberry a se selecionar para a utilização do módulo de automação para segurança domiciliar.

O Desenho do esquemático desenvolvido pode ser analisado na figura abaixo:

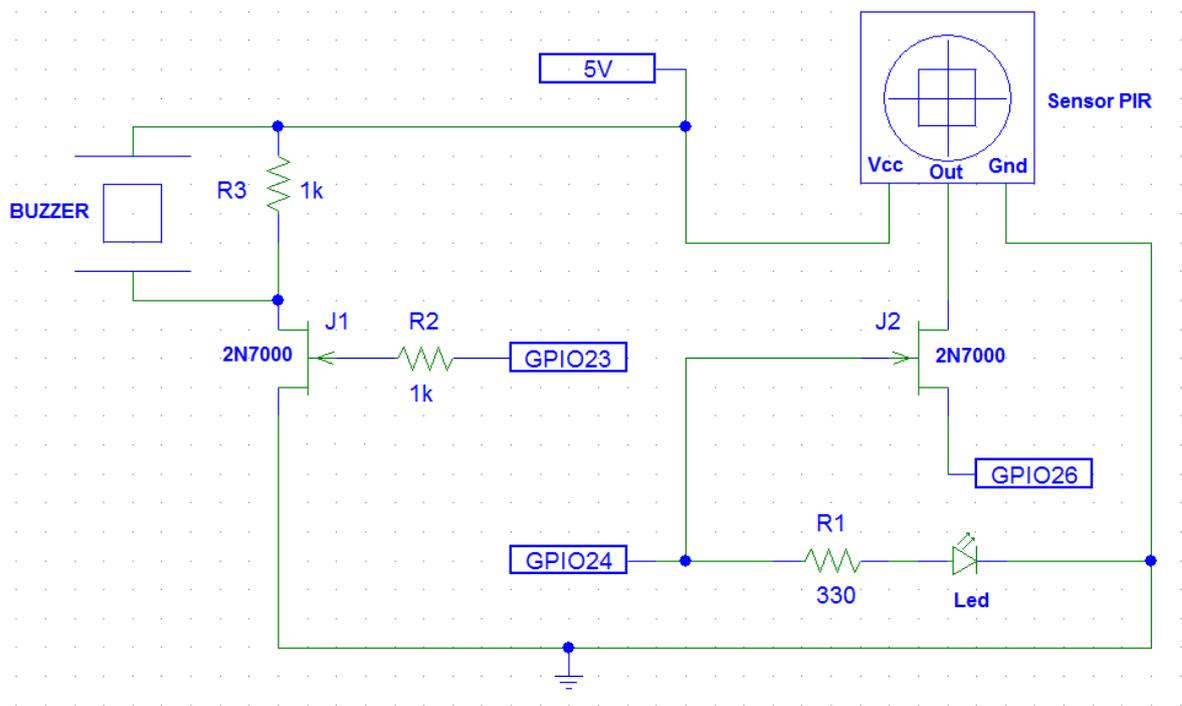


Figura 14 - Esquemático do módulo de automação para segurança domiciliar.

Fonte: Autoria Própria.

4. TESTES E RESULTADOS

Pode-se avaliar a portabilidade das distribuições Linux para plataformas ARM e analisou-se que as mesmas possuem um desempenho muito bom mesmo em sistemas muito pequenos. Por serem versáteis e populares, tem alcançado inúmeras implementações na comunidade.

Como sistemas de telecomunicações a nível residencial possuem uma carga de ligações muito inferior a de centrais telefônicas comerciais, como os ditos *call centers* e provedores de telefonia, a capacidade de processamento do Raspberry Pi se mostrou eficiente em testes de utilização. Além disso, se cada residência fizesse a utilização de um sistema como o Fly-IP conectado a internet, promovendo assim a comunicação por meio do VoIP, a utilização da telefonia convencional poderia ser drasticamente reduzida.

Com relação à facilidade de utilização do produto pelo usuário final, o projeto desenvolvido se mostrou eficaz, uma vez que gerenciar SIP Phones é uma tarefa facilitada pela Front End do programa FreePBX e ligações podem ser facilmente completadas utilizando-se SoftPhones como por exemplo o programa 3CXPhone, representado na figura 15.



Figura 15 – Efetuando Ligação pelo SIP Phone

Fonte: Autoria Própria

Muitos sensores/atuadores apresentam fácil integração com um sistema de automação, e ainda mais fácil a implementação de suas rotinas de tratamento, porém, existem muitos eletrodomésticos que trazem uma complexa compatibilização e dado a variedade de marcas, diferentes tipos de implementações para o mesmo resultado.

Nas figuras 16 e 17 é possível analisar o módulo de automação para segurança domiciliar já montado.

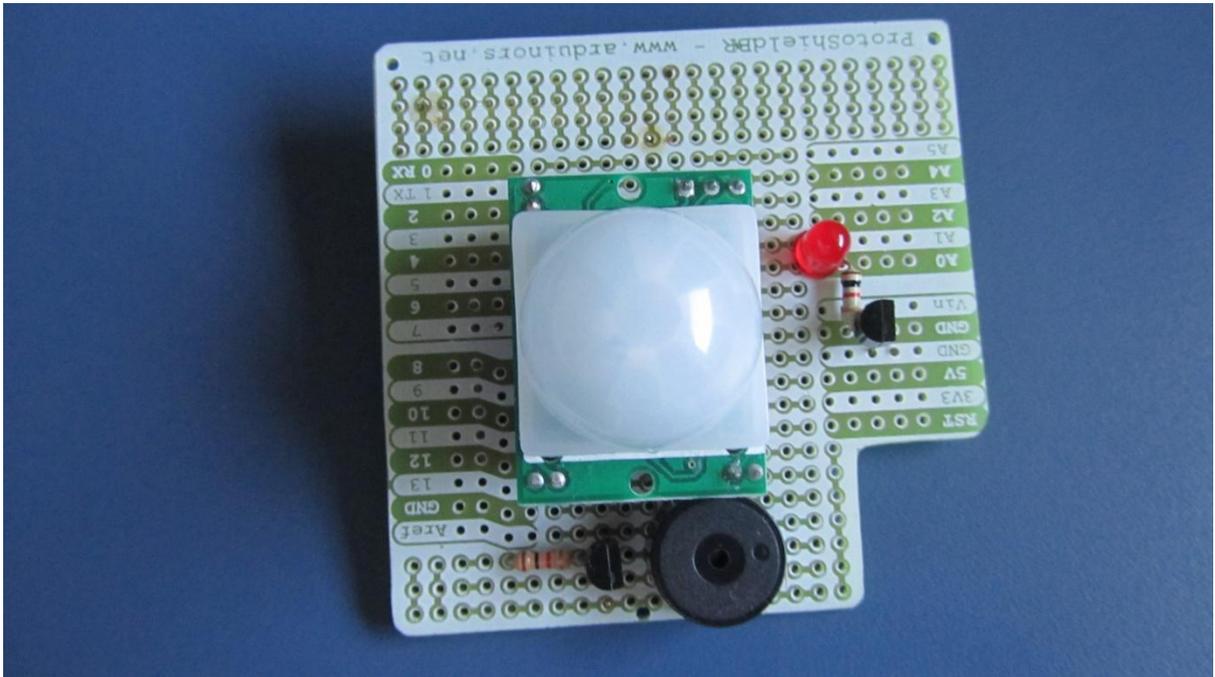


Figura 16 – Módulo de automação para segurança domiciliar – vista superior

Fonte: Autoria Própria

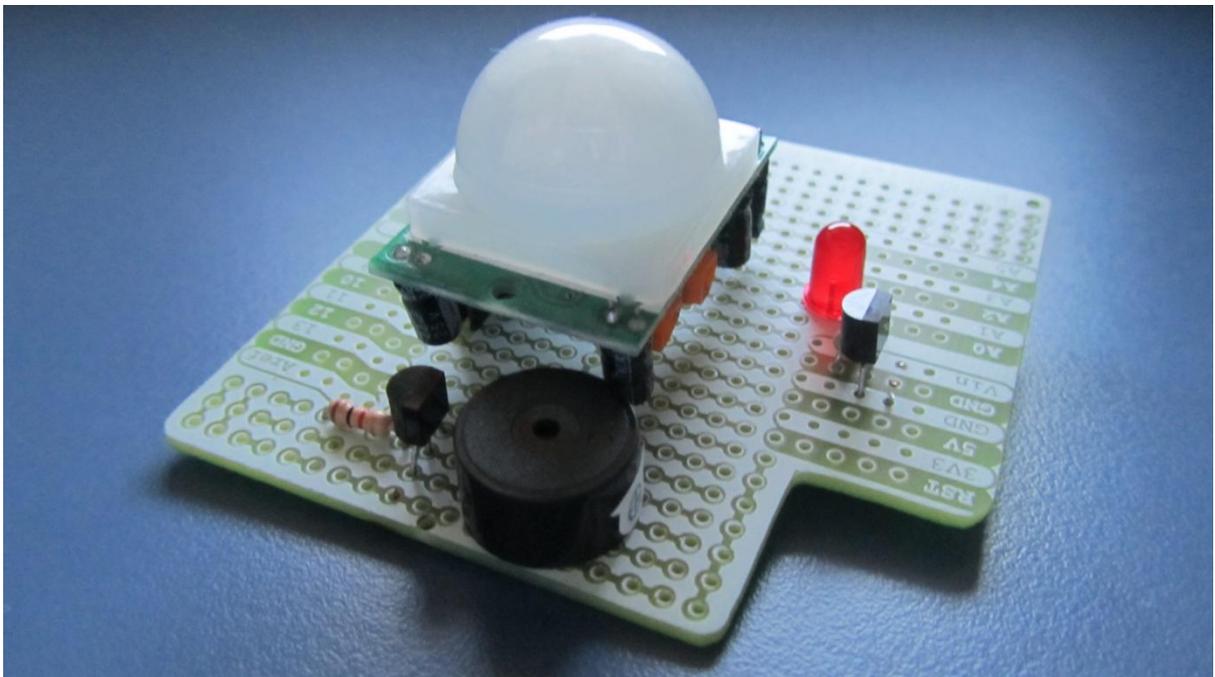


Figura 17 - Módulo de automação para segurança domiciliar – vista lateral

Fonte: Autoria Própria

5. GESTÃO

A divisão do projeto em módulos pode ser encarada como fator vantajoso, pois flexibiliza a execução do mesmo em várias frentes de trabalho. Deve-se também estar atento que é fundamental haver uma organização de metas e objetivos rígidos no decorrer para que a situação acabe ficando fora de controle.

A fim de se obter sucesso na realização do projeto, era necessário fazer um detalhado planejamento das etapas envolvidas. Assim, foram obtidos parâmetros para comparação da evolução dos custos e do tempo disponível.

A partir dessas informações, o planejamento deve conter também a utilização de recursos e o monitoramento dos custos. Além disso, os riscos que podem afetar o desenvolvimento do projeto devem estar presentes, pois dessa maneira, caso ocorram, o tempo despendido para a tomada de decisão será otimizado. A seguir, estão descritos diversos fatores que constituem a gestão do projeto.

5.1 Cronogramas de Planejamento

Elaborou-se o cronograma de forma a dividir as diversas etapas do projeto para estabelecer limites de tempo para a realização de cada tarefa. A seguir, na Tabela 24, é apresentado o cronograma resumido de planejamento.

Tabela 24 - Cronograma de desenvolvimento.

Atividades	AGO				SET				OUT				NOV				DEZ				JAN				FEV				MAR				ABR				MAI				JUN				JUL			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Etapa 1 - Especificar	█	█	█	█	█	█											█	█	█	█	█	█	█	█																								
Etapa 2 - Modelar			█	█	█	█	█	█	█	█							█	█	█	█	█	█	█	█																								
Etapa 3 - Definir solução					█	█	█	█	█	█							█	█	█	█	█	█	█	█																								
Etapa 4 - Implementar									█	█	█	█	█	█			█	█	█	█	█	█	█	█																								
Etapa 5 - Integrar													█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█																
Etapa 6 - Testar protótipo																	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█																
Etapa 7 - Finalizar																	█	█	█	█	█	█	█	█					█	█	█	█	█	█	█	█												

Fonte: Autoria Própria

No entanto, por mais que se busque contemplar todos os fatores, toda previsão comete erros (SLACK, 2009) [16] e no caso desse projeto não foi diferente, pois ocorreram atrasos na conclusão das etapas 3 a 6.

O atraso da etapa 6 se deve a demora da entrega dos componentes, pois os fornecedores com preços mais atraentes se encontram nos Estados Unidos. Então a

entrega desses sensores fica condicionada à boa vontade da alfândega em classificar o produto como canal verde.

Como a etapa 3 levou mais tempo para conclusão, a etapa 4 terminou mais tarde do que previsto. Para contornar esse empecilho foi iniciado a etapa 5 antes do previsto. No entanto, essa é uma etapa bastante meticulosa, e por isso, sua conclusão também levou uma semana a mais do que previsto.

Por esses motivos, a etapa de teste de protótipo ficou um pouco comprometida, já que se realizaram apenas os testes principais para validar o protótipo.

Na tabela 25 está apresentada a relação de carga horária necessária para cada etapa do projeto.

Tabela 25 - Relação de carga horária para cada etapa do projeto

Tarefa	Carga Horária
Etapa 1 - Especificar projeto	
Compreender a demanda	50
Etapa 2 - Modelar soluções	
Pesquisa dos sistemas existentes de PABX que utilizam VoIP.	90
Pesquisa de possíveis sensores utilizados para segurança doméstica.	90
Etapa 3 - Definir solução	
Especificar e adquirir sensor.	100
Escolha do(s) sistemas já existentes a ser(em) utilizados na implementação do projeto.	50
Etapa 4 - Implementar módulos	
Desenvolver hardware.	90
Etapa 5 - Integrar módulos	
Integrar sensores e atuadores com o sistema VoIP	60
Etapa 6 - Testar o protótipo	
Validar operações descritas	50
Etapa 7 – Finalizar	
Realizar levantamento final	80
Documentar o projeto	70
Preparar e Apresentar	30
Total	710

Fonte: Autoria Própria

6. PLANO DE NEGÓCIO

6.1. Sumário Executivo

6.1.1. Resumo dos Principais Pontos Do Plano De Negócio

O presente plano de negócio, baseado no Guia do Sebrae: “Como elaborar um plano de negócio” [17], apresenta uma solução de telefonia VOIP aliada à segurança doméstica cuja abordagem principal é trabalhar com sensores e atuadores que podem ser ativados ou desativados a distância mediante a comunicação telefônica VOIP.

O seu público-alvo é formado por famílias de classe média que procuram uma forma de economia na contra de telefone, ou que buscam mais segurança em seus lares.

A missão da empresa é: Prover uma solução eficaz de aproximar a família, garantindo contato gratuito e de qualidade entre todos seus membros, além de fornecer mais segurança e tranquilidade dentro dos lares, tudo isso com garantia e custo reduzido.

O Investimento Inicial necessário é de R\$86000.00, dispondo os proprietários de R\$20000.00 próprios, sendo que o capital restante precisará ser adquirido com terceiros e através de financiamento.

A empresa tem como pontos fortes a experiência profissional dos sócios-proprietários, aliada a um planejamento estratégico minuciosamente estruturado e um rigoroso controle dos processos de produção.

Os indicadores financeiros são:

Lucratividade: 18%

Rentabilidade: 6% ao mês.

Prazo de retorno do investimento: 17 meses

Ponto de Equilíbrio (PE): R\$ 19582.22

6.1.2. Dados dos Empreendedores, Experiência Profissional e Atribuições

Sócio 1:

Nome: Felipe Augusto Destefani

Endereço: Rua Jeronimo Durski, 437

Cidade: Curitiba

Estado: Paraná

Telefone (41) 9694-3941

Perfil: Conhecimentos avançados:

- Interface homem-máquina e entre máquinas;
- Comunicações;
- Software.

Conhecimentos intermediários:

- Hardware;
- Inteligência artificial.

Atribuições do sócio:

Primeiro desenvolvedor – responsável pelo desenvolvimento de hardware, software e interface com o usuário do sistema. Responsável pelo atendimento aos clientes para estudo da aplicação e possível replanejamento de funcionalidades.

Sócio 2:

Nome: Fillipe Lucchin Paukner

Endereço: Rua Joana Souza Gusso, nº 578

Cidade: Curitiba

Estado: Paraná

Telefone 1: (41) 33545612 Telefone 2: (41) 96327668

Perfil: Conhecimentos Avançados:

- Desenvolvimento de hardware
- Desenvolvimento de Firmware
- Redes

Conhecimentos intermediários:

- Software.
- Interface Homem Máquina.

Atribuições do Sócio:

Segundo desenvolvedor – responsável pelo desenvolvimento de hardware, software e interface com os técnicos de fabricação. Responsável pelo desenvolvimento de periféricos e integração.

6.1.3. Dados do Empreendimento

Nome da Empresa: FlyIP

CNPJ: sem CNPJ no momento,

CPF do sócio 1: 070.354.069-64

CPF do sócio 2: 071.837.659-54

6.1.4. Missão da Empresa

Prover uma solução eficaz de aproximar a família, garantindo contato gratuito e de qualidade entre todos seus membros, além de fornecer mais segurança e tranquilidade dentro dos lares, tudo isso com garantia, qualidade e custo reduzido.

6.1.5. Setores de Atividade

- Indústria de tecnologia
- Comércio
- Serviços

6.1.6. Forma Jurídica

- Sociedade Limitada

6.1.7. Enquadramento Tributário

A) ÂMBITO FEDERAL

- Regime Simples

B) ÂMBITO ESTADUAL

- ICMS – Regime Simplificado

6.1.8. Capital Social

Tabela 26 - Capital Social

	Nome do Sócio	%
SÓCIO 1	Felipe Augusto Destefani	50%
SÓCIO 2	Fillipe Lucchin Paukner	50%
Total		100%

Fonte: Autoria Própria

6.1.9. Fonte de Recursos

Os recursos iniciais para abertura da empresa e início de produção serão provenientes de fontes pessoais dos proprietários e de possíveis financiadores externos respectivamente.

6.2. Análise de Mercado

6.2.1. Estudo dos Clientes

- Público-alvo (perfil dos clientes)

Famílias. Cada família que possui seu lar conseguirá automatizá-lo e controlá-lo através de seu telefone com este produto. Em um âmbito geral, será dada uma alternativa à telefonia convencional; considerando que a rede ADSL já é disponibilizada em praticamente todas as residências, o produto visa usufruir desta para fornecer telefonia VOIP configurável pela família, assim tornada gratuita pelo contexto. Para as famílias que puderem investir um pouco mais, o sistema poderá contar com módulos de sensores e atuadores configuráveis conforme a necessidade/vontade do cliente.

- Comportamento dos clientes (interesses e o que os levam a comprar)

O comportamento do cliente pode ser descrito em duas áreas diferentes, sendo a primeira relativa ao sistema de comunicação. A telefonia VOIP reduz a praticamente zero os custos de ligações entre os usuários (considerando que os usuários já possuem acesso à internet e considerando os terminais da ligação VOIP). A segunda área comportamental está em função do sistema de segurança e da sua integração com o sistema de telefonia proposto; isto cria uma interface mais facilitada entre o usuário e o seu sistema de segurança. Um exemplo prático é que o usuário final ao instalar um sistema de alarme contra invasão não necessita contratar uma empresa de segurança (passa a ser opcional), e baseado em

configurações pré-estabelecidas o servidor pode ligar diretamente para o usuário e/ou à polícia.

6.2.2. Estudo dos Concorrentes

Tabela 27 - Estudo dos Concorrentes

	Qualidade(Telecom/Segurança residencial)		Preço	Condições de pagamento	Localização	Atendimento	Serviços aos clientes	Garantias oferecidas
Nossa empresa	Aprimoramento da tecnologia VoIP para substituir gradativamente a tecnologia de telefonia convencional.	Integração de todos os recursos da telefonia VoIP para a segurança. Porém o foco inicial não é a segurança, tornando este um serviço de baixa qualidade no início.	Deve estar dentro de uma faixa que qualquer família possa pagar. Cerca de uns 500 reais (kit com um servidor, aplicativo para terminal VoIP e um terminal VoIP físico.	Pagamento à vista ou cartão de crédito, podendo variar conforme venda pessoal.	Curitiba, a definir.	Técnicos treinados para atender conforme as necessidades do cliente.	Busca abranger todas as funcionalidades da telefonia e segurança domiciliar.	Produtos de alta qualidade e acessíveis a qualquer público. Vários serviços disponíveis.
Skype	O serviço gratuito possui bastante qualidade e simplicidade para facilitar o acesso dos usuários.	Não possui .	Gratuito (integração de alguns serviços é pago).	Cartão de crédito ou através de empresa terceirizada.	Virtual.	Atendimento via próprio skype ou telefone.	Expansão dos serviços de telecom/segurança a qualquer momento.	Produto acessível a qualquer público.
G5 segurança	Possui serviços integrados com a telefonia convencional, o que torna muito mais caro. (Qualidade padrão)	Alta qualidade.	Alto.	Conforme negociado com empresa.	Curitiba.	Técnicos e teleatendimento.	Explora serviços da telefonia convencional e serviços de segurança.	Produtos e serviço de alta qualidade.

Fonte: Autoria Própria

6.2.3. Estudo dos Fornecedores

Tabela 28 - Estudo dos Fornecedores

Ordem	Descrição dos itens a serem adquiridos (matérias-primas, insumos, mercadorias e serviços)	Nome do fornecedor	Preço	Condições de pagamento	Prazo de entrega	Localização (estado e/ou município)
1	Componentes eletrônicos (por servidor)	Beta	10 reais	À vista	Se encomendado, no mesmo dia.	Curitiba
2	Invólucro do servidor físico	Alibaba.com	1usd	Cartão de crédito	15 a 30 dias	China
3	Telefones VoIP (Aparelho Telefone Voip Stracta Ipt 160ii)	Fonte: MercadoLivre	25 reais(unidade) + 30 reais de frete	Cartão de crédito	1 semana	Minas Gerais
4	Circuito impresso	Circuibras	30 reais	À vista/cartão de crédito	1 semana	Curitiba

Fonte: Autoria Própria

6.3. Plano de *Marketing*

6.3.1. Descrição dos Principais Produtos E Serviços

Principais produtos (a serem fabricados, revendidos ou serviços prestados).

- Servidor VOIP
- Servidor Modular
- Módulos de segurança/automação

6.3.2. Preço

- PREÇO DE SAÍDA: R\$500.00
- SUPORTE: R\$300.00
- INSTALAÇÃO: R\$150.00
- FUNCIONALIDADES EXTRAS: Preço variável

6.3.3. Estratégias Promocionais

A princípio, as primeiras unidades devem ser fornecidas a pessoas conhecidas com o intuito de começar a propagar o sistema pelas famílias. Um recurso que deve ser utilizado como marketing do produto é a possibilidade de comunicação entre dois servidores via internet, assim, aliado a um sistema de configuração acessível ao usuário, a imagem a ser vendida é de um produto simples e prático que dá acesso à comunicação entre todos seus usuários. Assim que o produto comece a ser popularizado, pode-se estipular um preço de acordo com a realidade (considerando custos de produção e seu nível de popularização).

6.3.4. Estrutura de Comercialização

A estrutura inicial de comercialização está baseada no planejamento de marketing, de maneira que as primeiras unidades serão disponibilizadas praticamente sem custos. Ao fim da fase inicial de marketing, deve existir uma lista de possíveis novos usuários aos quais serão realizadas demonstrações do sistema como um todo.

Após a finalização da fase de demonstração espera-se estar com uma base de produtos instalados no mercado, de maneira que será possível decidir uma estratégia de comercialização mais eficaz em relação ao porte do negócio.

6.3.5. Localização do Negócio

TECPAR

Endereço: Rua Professor Algacyr Munhoz Mader, 3775, CIC, 81350-010, Curitiba, Paraná

Endereço http: <http://intec.tecpar.br/s/home/>

Fone: (55)(41) 3316-3188

Fax: (55)(41) 3346-2056

Considerações sobre o ponto (localização), que justifiquem sua escolha: É a localização de Curitiba em que se há mais facilidade para a abertura de novos negócios.

6.4. Plano Operacional

6.4.1. *Layout* ou Arranjo Físico

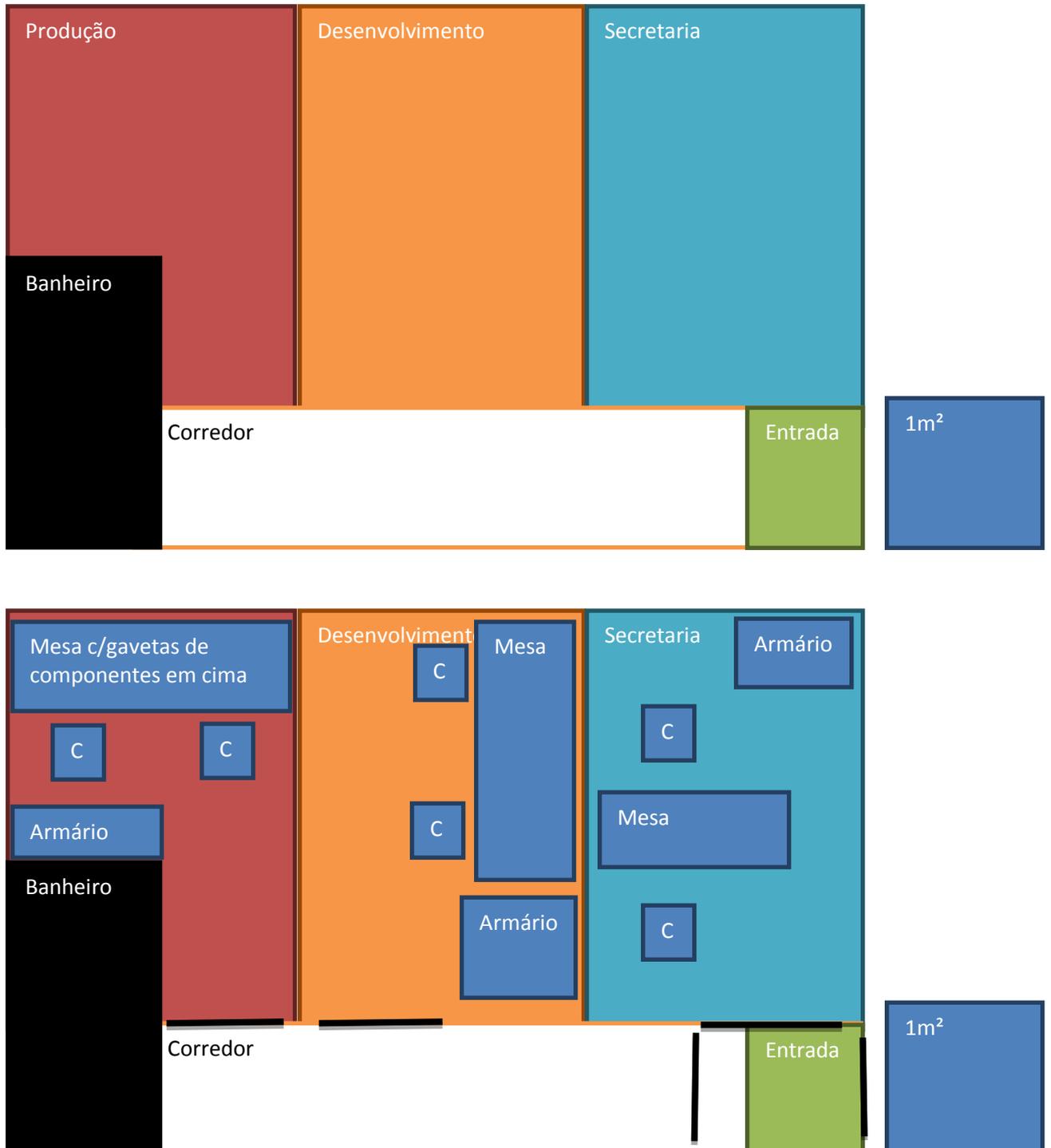


Figura 18 - Alocação de espaço do layout da empresa.

Fonte: Autoria Própria

6.4.2. Capacidade Produtiva/Comercial/Serviços

Inicialmente serão produzidos servidores conforme estabelecido no plano de marketing. Enquanto se dá a implementação do plano de marketing, devem ser produzidas mais unidades para posteriores vendas. A posterior produção deve ficar proporcional à pesquisa realizada no plano de marketing.

6.4.3. Processos Operacionais

A) Setor de Produção.

No setor de produção, deve haver quatro pessoas, duas responsáveis pelo desenvolvimento e duas pessoas responsáveis pela fabricação. No desenvolvimento uma pessoa é responsável pela comunicação com o atendimento aos clientes em conjunto com a secretaria e a outra pessoa é responsável pela comunicação com os técnicos da fabricação; ambos são responsáveis pelo desenvolvimento do projeto. No setor de fabricação, deve haver um técnico em software e um técnico de instalação do sistema; e ambos são responsáveis pela fabricação do sistema. Toda esta divisão de tarefas fica conforme alocação de tempo pré-estabelecida ou necessária.

B) Setor de vendas e de atendimento ao cliente.

Atendimento/vendas são feitos por três pessoas, um secretario, uma do desenvolvimento e um técnico de instalação. A prioridade de atendimento via telefonia é para o secretário e conforme haja clientes a serem atendidas localmente (ou mais ligações) estas ligações poderão ser repassadas ao desenvolvedor ou técnico responsável por atendimento.

C) Suporte técnico. Atendimento / Instalação.

A princípio para realizar o plano de marketing e enquanto a demanda for pequena o atendimento será feito como descrito no tópico anterior e a instalação pode ser feita pelo técnico de instalação e pelos próprios desenvolvedores.

6.4.4 Necessidade de Pessoal

Tabela 29 - Necessidade de Pessoal

Cargo	Função	Qualidades necessárias
Técnico de fabricação	Técnico em software.	Conhecimento de software.
	Técnico de instalação.	Conhecimento de redes.
Desenvolvedor	Atendimento aos clientes; Desenvolvimento de interface com o usuário; atendimento aos clientes em suas casas (fase inicial).	Conhecimento de gestão de pessoas; Conhecimento de marketing; Extroversão; Conhecimentos de eletrônica em geral.
	Comunicação com técnicos de fabricação; Desenvolvimento de funcionalidades do sistema.	Conhecimento de gestão de pessoas; Conhecimento de hardware; Conhecimento de redes; Conhecimentos de eletrônica em geral.
Secretário	Atendimento aos clientes via telefone ou pessoalmente.	Extroversão; Saber vender.

Fonte: Autoria Própria

6.5. Plano Financeiro

6.5.1. Estimativa dos Investimentos Fixos

- A – Máquinas e equipamentos

Tabela 30 - Máquinas e Equipamentos

	Descrição	Quantidade	Valor Unitário (R\$)	Total (R\$)
1	Computador	5	1300	6500
2	Estação de solda	3	200	600
3	Multímetro	3	200	600
4	Osciloscópio	1	2000	2000
5	Telefone	2	100	200
Total			R\$	9900

Fonte: Autoria Própria

- B – Veículos

Tabela 31 – Veículos

	Descrição	Quantidade	Valor Unitário (R\$)	Total (R\$)
1	Carro	1	35,000	35000
Total			R\$	35000

Fonte: Autoria Própria

- C – Móveis e utensílios

Tabela 32 - Móveis e utensílios

Descrição	Quantidade	Valor Unitário (R\$)	Total (R\$)
1 Cadeira	6	400	2400
2 Mesa	3	200	600
3 Armário	3	200	600
4 Luminária	3	25	75
5 Privada	1	100	100
Total	R\$		3775

Fonte: Autoria Própria

6.5.2. Capital de Giro

- A – Estimativa do estoque inicial

Tabela 33 - Estimativa do Estoque Inicial

Descrição	Quantidade	Valor Unitário (R\$)	Total (R\$)
1 Kit sistema (marketing)	15	0	0
2 Kit sistema	15	500	7500
Total	R\$		7500

Fonte: Autoria Própria

- B – Caixa Mínimo

1º Passo: Contas a receber – Cálculo do prazo médio de vendas

Tabela 34 - Caixa Mínimo

Prazo médio de vendas	%	Número de dias	Média ponderada em dias
A vista	60	1	0.6
A prazo (1)	30	30	9
A prazo (2)	8	45	3.6
A prazo (3)	2	60	1.2
Prazo médio total			14.4

Fonte: Autoria Própria

2º Passo: Fornecedores – Cálculo do prazo médio de compras

Tabela 35 - Cálculo do Prazo Médio de Compras

Prazo médio de vendas	%	Número de dias	Média ponderada em dias
A vista	40	1	0.4
A prazo (1)	40	30	12
A prazo (2)	15	45	6.75
A prazo (3)	5	60	3
		Prazo médio total	22.15

Fonte: Autoria Própria

3º Passo: Estoques – Cálculo da necessidade média de estoques

Tabela 36 - Cálculo de Necessidade Média de Estoque

Necessidade média de estoques	Número de dias
30 unidades	2 meses

Fonte: Autoria Própria

4º Passo: Cálculo da necessidade líquida de capital de giro em dias

Tabela 37 - Cálculo da Necessidade Líquida de Capital

Recursos da empresa fora do seu caixa	Número de dias
1. Contas a Receber - prazo médio de vendas	16
2. Estoques - necessidade média de estoques	60
Subtotal 1 (item 1 + 2)	76
Recursos de terceiros no caixa da empresa	
3. Fornecedores - prazo médio de compras	21
Subtotal 2	21
Necessidade líquida de capital de giro em dias (Subtotal 1 - Subtotal 2)	55

Fonte: Autoria Própria

- B – Caixa Mínimo Calculado

Tabela 38 - Caixa Mínimo Calculado

1. Custo fixo mensal (Estimativa dos custos fixos operacionais mensais) em reais	2000
2. Custo variável mensal (Demonstrativo de resultados) em reais	4000
3. Custo total da empresa (item 1 + 2)	6000
4. Custo total diário (item 3 / 30 dias)	200
5. Necessidade líquida de capital de giro em dias (vide resultado do quadro anterior)	55
Total de caixa mínimo (em reais)	11000

Fonte: Autoria Própria

6.5.3. Capital de Giro Resumido

Tabela 39 - Capital de Giro Resumido

Descrição	R\$
A - Estoque inicial	7500
B - Caixa mínimo	11000
Total do capital de giro	18500

Fonte: Autoria Própria

6.5.4. Investimentos Pré-Operacionais

Tabela 40 - Investimentos Pré Operacionais

Investimentos pré-operacionais	R\$
Despesas de legalização	2000
Obras civis e/ou reformas	1000
Divulgação	1000
Cursos e treinamentos	0
Total	4000

Fonte: Autoria Própria

6.5.5. Investimento Total Resumido

Tabela 41 - Investimento Total Resumido

Descrição dos investimentos	Valor	(%)
1. Investimentos fixos	60000	69.76744
2. Capital de giro	11000	12.7907
3. Investimentos pré-operacionais	15000	17.44186
Total	86000	100
Fontes de recursos	Valor	(%)
1. Recursos próprios	20000	23.25581
2. Recursos de terceiros	10000	11.62791
3. Outros	56000	65.11628
Total	86000	100

Fonte: Autoria Própria

6.5.6. ESTIMATIVA DO FATURAMENTO MENSAL DA EMPRESA

Tabela 42 - Estimativa do Faturamento Mensal da Empresa

Produto/serviço (primeiro mês)	Quantidade	Preço em venda unitário (em R\$)	Faturamento Total (em R\$)
1 Kit sistema	25	500	12500
2 Suporte	25	300	7500
3 Funcionalidades requeridas	5	800	4000
4 Instalação	25	150	3750
Total			27750

Fonte: Autoria Própria

6.5.7. Estimativa dos Custos de Matéria-Prima, Materiais Diretos e Terceirizações.

Tabela 43 - Estimativa dos Custos de Matéria-Prima, Materiais Diretos e Terceirizações

Produto/serviço	Quantidade	Preço em venda unitário (em R\$)	Faturamento Total (em R\$)
1 Componentes eletrônicos (por servidor)	30	10	300
2 Involucro do servidor físico	30	2	60
3 Telefones VoIP	30	25	750
4 Circuito impresso	30	30	900
Total			2010

Fonte: Autoria Própria

6.5.8. Estimativa dos Custos de Comercialização

Tabela 44 - Estimativa dos Custos de Comercialização

Descrição	% x	Faturamento estimado	Custo Total
1 – Impostos			
Impostos Federais			
Simplex	5.97%	R\$ 27,750.00	R\$ 1,656.68
Impostos Estaduais			
ICMS	1.86%	R\$ 27,750.00	R\$ 516.15
Subtotal 1			R\$ 2,172.83
2 - Gastos com Vendas			
Comissões	8%	R\$ 27,750.00	R\$ 2,220.00
Propaganda	8%	R\$ 27,750.00	R\$ 2,220.00
Taxa de Cartão de Crédito	3%	R\$ 27,750.00	R\$ 832.50
Subtotal 2			R\$ 5,272.50
Total (Subtotal1+Subtotal2)			R\$ 7,445.33

Fonte: Autoria Própria

6.5.9. Apuração do Custo dos Materiais Diretos e/ou Mercadorias Vendidas

Tabela 45 - Apuração do Custo Dos Materiais Diretos E/Ou Mercadorias Vendidas

Produto/serviço	Estimativa de Vendas	Custo unitário de materiais	CMD/CMV
Kit Sistema	30	R\$ 67.00	R\$ 2,010.00
Instalação	30	R\$ 30.00	R\$ 900.00
Total			R\$ 2,910.00

Fonte: Autoria Própria

6.5.10. Estimativa dos Custos com Mão-De-Obra

Tabela 46 - Estimativa dos Custos com Mão-De-Obra

Função	Quant.	Salário Mensal	Subtotal	% de encargos	Encargos sociais	Total
Técnico em software.	1	R\$ 1,000.00	R\$ 1,000.00	33.77%	R\$ 337.70	R\$ 1337.70
Técnico de instalação.	1	R\$ 1,000.00	R\$ 1,000.00	33.77%	R\$ 337.70	R\$ 1337.70
Secretário	1	R\$ 1,000.00	R\$ 1,000.00	33.77%	R\$ 337.70	R\$ 1,337.70
Total			R\$ 3,000.00		R\$ 1013.10	R\$ 4013.10

Fonte: Autoria Própria

6.5.11. Estimativa do Custo com Depreciação

Tabela 47 - Estimativa do Custo com Depreciação

Ativos fixos	Valor Bem	Vida útil (anos)	Depreciação anual (R\$)	Depreciação mensal (R\$)
Computador	R\$ 2,000.00	3	R\$ 666.67	R\$ 55.56
Estação de solda	R\$ 600.00	10	R\$ 60.00	R\$ 5.00
Múltímetro	R\$ 600.00	10	R\$ 60.00	R\$ 5.00
Osciloscópio	R\$ 25.00	10	R\$ 2.50	R\$ 0.21
Telefone	R\$ 200.00	10	R\$ 20.00	R\$ 1.67
Cadeira	R\$ 2,400.00	10	R\$ 240.00	R\$ 20.00
Mesa	R\$ 600.00	10	R\$ 60.00	R\$ 5.00
Armário	R\$ 600.00	10	R\$ 60.00	R\$ 5.00
Luminária	R\$ 75.00	10	R\$ 7.50	R\$ 0.63
Privada	R\$ 100.00	25	R\$ 4.00	R\$ 0.33
Carro	\$35,000.00	5	R\$7,000.00	R\$ 583.33
TOTAL	R\$42,200.00		R\$ 8,180.67	R\$ 681.72

Fonte: Autoria Própria

6.5.12. Estimativa de Custos Fixos Operacionais Mensais

Tabela 48 - Estimativa de Custos Fixos Operacionais Mensais

Descrição	Custo Total Mensal
IPTU	R\$ 50.00
Água	R\$100.00
Energia Elétrica	R\$ 150.00
Telefone	R\$200.00
Honorários do contador	R\$480.00
Pró-labore	R\$ 6,000.00
Manutenção dos equipamentos	R\$ 100.00
Salários + Encargos	R\$ 4,013.10
Material de limpeza	R\$100.00
Material de escritório	R\$ 150.00
Combustível	R\$ 250.00
Depreciação	R\$ 681.72
TOTAL	R\$ 12,274.82

Fonte: Autoria Própria

6.5.13. Demonstrativo de Resultados

Tabela 49 - Demonstrativo de Resultados

Descrição	R\$	%
1 - Receita Total em Vendas	R\$ 27,750.00	100%
2 - Custos variáveis Totais		
Custos com materiais diretos e ou CMV	R\$ 2,910.00	10%
Impostos sobre vendas	R\$ 2,172.83	8%
Gastos com vendas	R\$ 5,272.50	19%
Subtotal de 2	R\$ 10,355.33	37%
3 - Margem de Contribuição (1-2)	R\$ 17,394.68	63%
4 -Custos Fixos Totais	R\$ 12,274.82	44%
Resultado Operacional (3-4)	R\$ 5,119.85	18%

Fonte: Autoria Própria

6.5.13. Indicadores de Viabilidade

A) PONTO DE EQUILÍBRIO

$$\text{A) PE} = \frac{\text{Custo Fixo Total}}{\text{Índice da Margem de Contribuição (*)}}$$

(*) Observação

$$\text{Índice da Margem de Contribuição} = \frac{\text{Margem de Contribuição (Receita Total - Custo Variável Total)}}{\text{Receita Total}}$$

Tabela 50 - Ponto de Equilíbrio

Custo fixo total	R\$ 12,274.82
Índice de margem de contribuição	0.626835135
Margem de contribuição	R\$ 17,394.68
Receita Total	R\$ 27,750.00
Custo variável total	R\$ 10,355.33
Ponto de Equilíbrio	R\$ 19,582.22

Fonte: Autoria Própria

B). LUCRATIVIDADE

$$\text{Lucratividade} = \frac{\text{Lucro Líquido}}{\text{Receita Total}} \times 100$$

Tabela 51 - Lucratividade

Lucro líquido	R\$ 5,119.85
Receita Total	R\$ 27,750.00
Lucratividade	18%

Fonte: Autoria Própria

C) RENTABILIDADE

$$\text{Rentabilidade} = \frac{\text{Lucro Líquido}}{\text{Investimento Total}} \times 100$$

Tabela 52 - Rentabilidade

Lucro Líquido	R\$ 5,119.85
Investimento total	R\$ 86,000.00
Rentabilidade (mês)	6%

Fonte: Autoria Própria

D) PRAZO DE RETORNO DO INVESTIMENTO

$$\text{Prazo de Retorno do Investimento} = \frac{\text{Investimento Total}}{\text{Lucro Líquido}}$$

Tabela 53 - Prazo de Retorno de Investimento

Investimento Total	R\$ 86,000.00
Lucro Líquido	R\$ 5,119.85
Prazo de Retorno do Investimento (Meses)	R\$ 16.797358

Fonte: Autoria Própria

6.6. Construção de Cenários

Tabela 54 - Construção de Cenários

Descrição	Cenário Provável		Cenário Pessimista		Cenário Otimista	
	R\$	%	R\$	%	R\$	%
1 - Receita Total em Vendas	R\$ 27,750.00	100%	R\$ 23,000.00	100%	R\$ 32,500.00	100%
2 - Custos variáveis Totais						
Custos com materiais diretos e ou CMV	R\$ 2,910.00	10%	R\$ 2,910.00	13%	R\$ 2,910.00	9%
Impostos sobre vendas	R\$ 2,172.83	8%	R\$ 1,800.90	8%	R\$ 2,544.75	8%
Gastos com vendas	R\$ 5,272.50	19%	R\$ 4,370.00	19%	R\$ 6,175.00	19%
Subtotal de 2	R\$ 10,355.33	37%	R\$ 9,080.90	39%	R\$ 11,629.75	36%
3 - Margem de Contribuição (1-2)	R\$ 17,394.68	63%	R\$ 13,919.10	61%	R\$ 20,870.25	64%
4 -Custos Fixos Totais	R\$ 12,274.82	44%	R\$ 12,274.82	53%	R\$ 12,274.82	38%
Resultado Operacional (3-4)	R\$ 5,119.85	18%	R\$ 1,644.28	7%	R\$ 8,595.43	26%

Fonte: Autoria Própria

Ações corretivas e preventivas:

Para evitarmos o cenário pessimista pode-se recorrer a outras formas de venda de equipamentos ou serviços de maneira a aumentar o faturamento para próximo do cenário provável.

6.7. Avaliação Estratégica

6.7.1. Análise da Matriz F.O.F.A.

Tabela 55 - Análise da Matriz F.O.F.A.

	Fatores Internos	Fatores Externos
Pontos Fortes	<i>Forças</i>	<i>Oportunidades</i>
	Produto diferenciado e modular. Atendimento personalizado ao cliente. Preço de venda competitivo. Equipe treinada e motivada. Experiência profissional dos sócios-proprietários	Existência de linhas de financiamento voltado a tecnologias inovadoras. Poucos concorrentes na região. Aumento crescente da demanda conforme marketing de rede.
Pontos Fracos	<i>Fraquezas</i>	<i>Ameaças</i>
	Falta de experiência empreendedora.	A maioria dos fornecedores é internacional.

Fonte: Autoria Própria

6.8. Avaliação do Plano De Negócio

Mediante o planejamento desenvolvido com este plano de negócio espera-se estar preparado para iniciar um empreendimento com grande probabilidade de sucesso e que comece a ganhar mercado e a se expandir logo nos seis primeiros meses de vida do novo negócio.

7. CONCLUSÕES

Para transformar este protótipo em produto, é necessário separar em dois blocos: uma central de ligações e uma central de atuadores para que usuários possam escolher qual ou quais serviços são importantes em seu sistema final.

O sistema de aquisição/envio de dados de sensores/atuadores também deve ser melhorado visto que para a automação de uma residência inteira, um sistema de comunicação sem fio se mostraria mais vantajoso.

Baseado no plano de custos do Plano de Negócios que pode ser analisado na sessão 6 deste relatório, este projeto se mostrou viável, porém, pode-se otimizar o lucro do mesmo com a separação do sistema em módulos que sejam vendidos separadamente, mantendo a facilidade de integração.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WINTERMEYER, Stefan; BOSCH, Stephen. **Practical Asterisk 1.4 and 1.6: From Beginner to Expert**. Addison-Wesley Professional; 840p. 1edition. 2009.
- [2] Comunidade da Beagle Board. Disponível em <<http://www.beagleboard.org>> Último Acesso em: 11/09/2013.
- [3] Comunidade da Panda Board. Disponível em <<http://www.pandaboard.org>> Último Acesso em: 11/09/2013.
- [4] Comunidade do Raspberry Pi. Disponível em <<http://www.raspberrypi.org>> Último Acesso em: 11/09/2013.
- [5] Revista Digital PC Mag. Disponível em <<http://www.pcmag.com.br/us/article2/0,2817,2407058,00.asp>> Último Acesso em: 11/09/2013.
- [6] **Raspberry Pi Headlines Last Week: summary**. Disponível em <<http://www.hackthings.com/raspberry-pi-headlines-last-week-summary/>> Último Acesso em: 11/09/2013.
- [7] Comunidade Asterisk. Disponível em <<http://www.asterisk.org/>> Último Acesso em: 11/09/2013.
- [8] BRYANT, R; MADSEN, L; MEGGELEN, J. V.; **Asterisk™: The Definitive Guide, Fourth Edition**. O’ Reilly Media, Inc. 845p. 2013.
- [9] **888VoIP**. Disponível em <<http://www.888voip.com>> Último Acesso em: 11/09/2013.
- [10] **3CX**. Disponível em <<http://www.3cx.com>> Último Acesso em: 11/09/2013.
- [11] Comunidade Debian. Disponível em <<http://www.debian.org>> Último Acesso em: 11/09/2013.
- [12] Comunidade elinux. Disponível em <http://elinux.org/RPi_Easy_SD_Card_Setup> Último Acesso em: 11/09/2013.
- [13] Comunidade FreePBX. Disponível em <<http://www.freepbx.org>> Último Acesso em: 11/09/2013.
- [14] Comunidade VoIP-Info. Disponível em <<http://www.voip-info.org>> Último Acesso em: 11/09/2013.
- [15] Tato Equipamentos Eletrônicos. Disponível em <http://www.tato.ind.br/detalhe_produto.php?codigo_chave=177> Último Acesso em: 11/09/2013.
- [16] SLACK, N *et al.* **Administração da Produção**. 2 ed. São Paulo: Atlas, 2002

[17] ROSA, Cláudio Afrânio. **Como elaborar um plano de negócio**. SEBRAE, 120 p. Brasília, 2007.