UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕEs  
PRÓ-REITORIA DE ENSINO, PESQUISA E PÓS-GRADUAÇÃO  
CÂMPUS DE SANTO ÂNGELO

DEPARTAMENTO DAS ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO  
CURSO DE ENGENHARIA ELÉTRICA

Leonardo felipe da silva dos santos

Arquitetura de hardware e software open source para automação residencial

SANTO ÂNGELO – RS  
2018

Leonardo felipe da silva dos santos

Arquitetura de hardware e software open source para automação residencial

Trabalho de conclusão de curso apresentado como  
requisito parcial à obtenção do grau de  
Engenheiro Eletricista, Departamento das  
Engenharias e Ciência da Computação da  
Universidade Regional Integrada do Alto  
Uruguai e das Missões – Campus de Santo  
Ângelo.

Orientador: Me. Claudio Felipe Tomm.

SANTO ÂNGELO – RS  
2018

Leonardo felipe da silva dos santos

Arquitetura de hardware e software open source para automação residencial

Trabalho de conclusão de curso apresentado como  
requisito parcial à obtenção do grau de  
Engenheiro Eletricista, Departamento das  
Engenharias e Ciência da Computação da  
Universidade Regional Integrada do Alto  
Uruguai e das Missões – Campus de Santo  
Ângelo.

Orientador: Me. Claudio Felipe Tomm.

Santo Ângelo, \_\_\_\_ de \_\_\_\_\_\_\_ de \_\_\_\_\_\_\_.

Banca Examinadora

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof.

(instituição)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof.

(instituição)

Dedico este trabalho

Agradecimentos

Desejamos expressar nossos agradecimentos mais sinceros ao Departamento de Produção Animal I da Universidade de León pelos meios que colocaram a nossa disposição para a realização deste trabalho e à Universidade Regional Integrada do Alto Uruguai e das Missões por nos ter concedido permissão, no período de realização dessa tese, liberando-me das obrigações durante minha ausência.

Assim mesmo, desejamos manifestar nosso reconhecimento mais sincero a todas as pessoas que colocaram à nossa disposição seus conhecimentos, auxílio, sugestões e espírito crítico.

O único lugar onde o sucesso vem antes do trabalho é no dicionário.

(Albert Einstein)

RESUMO

O cenário atual na área de domótica aponta para a necessidade da utilização de um sistema operacional de código aberto que incentive o investimento na integração entre diferentes soluções e permita a criação de produtos que possam ser facilmente instalados, utilizados e atualizados pelo proprio usuário. Há empresas no mundo em geral no estados unidos e europa oferecem uma solução inovadora de automação por controle de dispositivos conectados à nuvem através de um pacote de ferramentas de software e hardware disponibilizados por eles porém não os disponibilizam por open source, normalmente é ela quem presta os serviços de atualização ou modificação dos produtos. No presente trabalho, o objetivo final é um produto qual o utilitário possa criar, modificar ou customizar conforme sua vontade, tanto quanto hardware ou software, desde a progamação a ligação da lampada. O trabalho visa utilizar a ESP-32S, controlada por um dispositivo provido de windows e/ou um dispositivo Android. Na automação residencial, diferentemente da automação industrial, respostas rápidas, precisas e altamente imunes a falhas podem dar lugar ao conceito de produtos fáceis de operar e com um bom acabamento. Além disso, as interfaces dos aplicativos desenvolvidos para esse trabalho serão compartilhadas, ou seja sendo iguais tanto em dispositivos móveis quantos em dispositivos fixos como os computadores.Utilizando elementos intuitivos para usuários de dispositivos móveis e também dos computadores. Através do aplicativo, é possível gerenciar todos os domos ligados ao sistema e colocados online. O projeto apresenta, também, uma alta capacidade de adaptação para diferentes aplicações, como o controle dos mais diversos tipos de equipamentos, assim como o controle de carga por meio de medidas de tensão e corrente, quais poderá ser visualizado em todos os dispositivos.

**Palavras-chave:** Domótica. Eficiência energética. Baixo Custo. Protótipo. Aplicação Residencial. Modelagem de Sistemas de Domos. Modelagem de Software e Hardware. Open Source.

ABSTRACT

(TRADUZIR RESUMO) O cenário atual na área de domótica aponta para a necessidade da utilização de um sistema operacional de código aberto que incentive o investimento na integração entre diferentes soluções e permita a criação de produtos que possam ser facilmente instalados, utilizados e atualizados pelo proprio usuário. Há empresas no mundo em geral no estados unidos e europa oferecem uma solução inovadora de automação por controle de dispositivos conectados à nuvem através de um pacote de ferramentas de software e hardware disponibilizados por eles porém não os disponibilizam por open source, normalmente é ela quem presta os serviços de atualização ou modificação dos produtos. No presente trabalho, o objetivo final é um produto qual o utilitário possa criar, modificar ou customizar conforme sua vontade, tanto quanto hardware ou software, desde a progamação a ligação da lampada. O trabalho visa utilizar a ESP-32S, controlada por um dispositivo provido de windows e/ou um dispositivo Android. Na automação residencial, diferentemente da automação industrial, respostas rápidas, precisas e altamente imunes a falhas podem dar lugar ao conceito de produtos fáceis de operar e com um bom acabamento. Além disso, as interfaces dos aplicativos desenvolvidos para esse trabalho serão compartilhadas, ou seja sendo iguais tanto em dispositivos móveis quantos em dispositivos fixos como os computadores.Utilizando elementos intuitivos para usuários de dispositivos móveis e também dos computadores. Através do aplicativo, é possível gerenciar todos os domos ligados ao sistema e colocados online. O projeto apresenta, também, uma alta capacidade de adaptação para diferentes aplicações, como o controle dos mais diversos tipos de equipamentos, assim como o controle de carga por meio de medidas de tensão e corrente, quais poderá ser visualizado em todos os dispositivos.

**Palavras-chave:** Domótica. Eficiência energética. Baixo Custo. Protótipo. Aplicação Residencial. Modelagem de Sistemas de Domos. Modelagem de Software e Hardware. Open Source.

**LISTA DE SIGLAS**

sql, html, css e php, IDE,

**LISTA DE ILUSTRAÇÕES**

[Figura 1 Pinagem de NodeMCU-32S 12](#_Toc521165219)

[Figura 2 MCP23016 Pinagem 13](#_Toc521165220)

[Figura 3 Ligação MCP23016 14](#_Toc521165221)

[Figura 4 Sensor de Corrente 15](#_Toc521165222)

[Figura 6 Ligação Sensor De Corrente 16](#_Toc521165223)

[Figura 6 Circuito de Ativação relé com baixa potência. 17](#_Toc521165224)

[Figura 7 PNP 18](#_Toc521165225)

[Figura 8 NPN 19](#_Toc521165226)

[Figura 9 Fonte 5V 19](#_Toc521165227)

[Figura 10 Diagrama básico de funcionamento de protocolo 20](#_Toc521165228)

[Figura 11 Funcionamento de Porta 21](#_Toc521165229)

[Figura 12 Funcionamento Socket 22](#_Toc521165230)

[Figura 13 Diagrama de funcionamento completo do Socket 23](#_Toc521165231)

[Figura 14 Esquemático do Chip WROOM32. 24](#_Toc521165232)

[Figura 15 Diagrama de Blocos ESP32S 26](#_Toc521165233)

**LISTA DE TABELAS**

Nenhuma entrada de índice de ilustrações foi encontrada.

Sumário

[1. Introdução 8](#_Toc521165234)

[1.1. Caracterização e Justificativa 8](#_Toc521165235)

[1.1.1. Caracterização 8](#_Toc521165236)

[1.1.2. Contextualização 8](#_Toc521165237)

[1.1.3. Justificativa 8](#_Toc521165238)

[1.2. Objetivos 8](#_Toc521165239)

[1.2.1. Objetivo Principal 8](#_Toc521165240)

[1.2.2. Objetivos específicos 9](#_Toc521165241)

[2. Referencial Teorico 10](#_Toc521165242)

[2.1 Java 10](#_Toc521165243)

[2.1.1 O que é java 10](#_Toc521165244)

[2.1.2 Funcionamento do Visual Studio 10](#_Toc521165245)

[2.1.3 Funcionamento da IDE do Arduino 10](#_Toc521165246)

[2.2 Wireless 11](#_Toc521165247)

[2.2.1 Como configurar a wireless da ESP-32S 11](#_Toc521165248)

[2.3 Portas da Esp-32s 11](#_Toc521165249)

[2.3.1 Portas Digitais(I/O) 11](#_Toc521165250)

[2.3.2 ADC e DAC 12](#_Toc521165251)

[2.4 protocolo de comunicação i2c 13](#_Toc521165252)

[2.4.1 Utilização do I²C do MCP23016 13](#_Toc521165253)

[2.5 Circuitos 14](#_Toc521165254)

[2.5.1 Sensor de Corrente 14](#_Toc521165255)

[2.5.2 Esquema sensor de corrente 15](#_Toc521165256)

[2.5.3 Relés 16](#_Toc521165257)

[2.5.4 Optoacoplador 17](#_Toc521165258)

[2.5.5 Transistores 17](#_Toc521165259)

[2.5.6 Fonte para ligação módulos relés e outros sensores 19](#_Toc521165260)

[3. Metodologia 20](#_Toc521165261)

[3.1 Socket; 20](#_Toc521165262)

[3.1.1 Protocolo 20](#_Toc521165263)

[3.1.2 Porta 21](#_Toc521165264)

[3.1.3 Funcionamento Socket 21](#_Toc521165265)

[3.1.4 Servidor 22](#_Toc521165266)

[3.1.5 Cliente 22](#_Toc521165267)

[3.2 Hardware 23](#_Toc521165268)

[3.2.1 ESP-32 23](#_Toc521165269)

[3.3 Software 26](#_Toc521165270)

[3.4 Aplicativo Móvel 27](#_Toc521165271)

[4. ANÁLISE DOS RESULTADOS 28](#_Toc521165272)

[5. Pesquisa Avaliativa para implantação do sistema 29](#_Toc521165273)

[6. CONSIDERAÇÕES FINAIS 30](#_Toc521165274)

[7. Bibliografia 31](#_Toc521165275)

# Introdução

RECONSTRUIR

## Caracterização e Justificativa

### Caracterização

O projeto é construído para que a domótica, hoje num país como o Brasil é realmente bem cara, fazendo com que uma central básica de controle de iluminação, alarme e ar condicionado passe facilmente dos R$5.000,00, sendo que o projeto com uma plataforma de prototipagem rápida, com isso se tende a baratear o custo e também possa servir de base para novos projetos de construção desde o aplicativo móvel e software para distribuição open Source.

### Contextualização

A contribuição do projeto deve ser barateamento dos sistemas, hoje em dia o sistema em geral demanda de um alto custo, são necessários sistemas proprietários e protocolos, como há muitas estruturas já prontas, podemos então utilizar destes sistemas já prontos.

Hoje os modelos de arquitetura se restringem a ser de operação de sistemas e controle remotos, monitores de sistemas, atuadores e sensores específicos de automação. Isso torna esses equipamentos de domótica, bastante variáveis.

### Justificativa

Os sistemas disponíveis utilizando um servidor SQL, html5, css3 e php, qual pode ser acessado em uma rede local gerando assim um servidor local, qual pode rodar os scripts para a automação da residência, com a utilização dessas tecnologias gratuitas, o sistema pode ser distribuído Open Source.

O sistema pode facilitar, a criação de módulos flexíveis, módulos em que temos apenas poucas ligações como uma pequena parte da casa, esse sistema pode fazer toda a automação facilmente e rapidamente conforme o programado.

## Objetivos

### Objetivo Principal

Criar um protótipo de arquitetura, em que possa ser utilizado equipamentos e materiais disponíveis no mercado, para automação residencial.

### Objetivos específicos

* Criar uma arquitetura para automação residencial
* Medir o consumo de energia.
* Proteger o sistema elétrico residencial.
* Supervisionar e controlar por meio de aplicativo de smartphone.
* Supervisionar e controlar por meio de Software para computador.
* Aplicar o sistema em residências.
* Desenvolver um aplicativo qual rode em um sistema operacional Windows superior ao Windows XP.
* Desenvolver um aplicativo móvel qual atenda a versão minima Android 4.0.2, da Google.
* Desenvolver um hardware qual possa ser adaptado conforme a necessidade da estrutura residencial.

# Referencial Teorico

## Java

### O que é java

Java é uma linguagem de programação desenvolvida por James Gosling, juntamente com outros colaboradores, no início da década de 1990, na empresa Sun Microsystems.

A linguagem de programação Java é orientada a objetos (comportamento dos objetos determinados por classes) e compilada em *bytecode* (as instruções são executadas através de uma Máquina Virtual Java - JVM e podem ser processadas em sistemas com suporte a C++). A sintaxe da linguagem Java é similar às linguagens C e C++.

A linguagem Java é a mais importante da Plataforma Java, que agora pertence à empresa Oracle. Outra linguagem desenvolvida para a plataforma Java é a Groovy.

Grande parte das instruções Java são disponibilizadas gratuitamente para download. Desde 2007, o código fonte Java foi liberado sob licença da GNL (*General Public License*).

### Funcionamento do Visual Studio

O IDE (ambiente de desenvolvimento interativo) do Visual Studio é um painel de inicialização criativa que pode ser usado para exibir e editar quase todo tipo de código e, em seguida, para depurar, compilar e publicar aplicativos para Android, iOS, Windows, Web e nuvem. Há versões disponíveis para o Mac e Windows. Este tópico apresenta os recursos do IDE do Visual Studio.

### Funcionamento da IDE do Arduino

O software para programação do Arduino é uma IDE que permite a criação do códigos para as placas suportadas. A linguagem de programação é criada a partir da linguagem [Wiring](http://wiring.org.co/). A IDE apresenta um alto grau de abstração, possibilitando o uso de um microcontrolador sem que o usuário conheça o mesmo, nem como deve ser usado os registradores internos de trabalho. A IDE possui uma linguagem própria baseada na linguagem C e C++.

O Ciclo de programação do Arduino pode ser dividido da seguinte maneira:

1. Conexão da placa a uma porta USB do computador;
2. Desenvolvimento de um sketch com comandos para a placa;
3. Upload do código para a placa, utilizando a comunicação USB.
4. Aguardar a reinicialização, após ocorrerá à execução do código criado.

## Wireless

### Como configurar a wireless da ESP-32S

A configuração do wireless é simples e rápida, sendo ainda mais fácil utilizar pois ela pode ser por OTA (over the air), sendo ela fácil de ser configurada, só colocar o SSID e a senha da rede qual damos o SSID e já estamos com a ESP-32S “Online”, o passo a passo será dada nos resultados.

## Portas da Esp-32s

### Portas Digitais(I/O)

Como consta na imagem abaixo, o ESP32 tem 34 pinos GPIOs que podem ser atribuídos a várias funções, como:

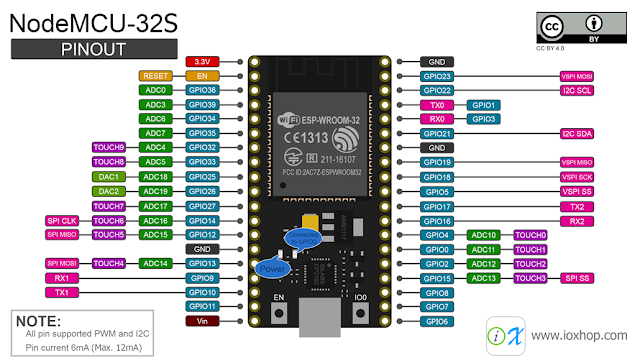
Digital-only, somente configurados com saída ou entradas digitais (O/I);

Analog-enabled (podem ser configurados como digital);

Capacitive-touch-enabled (podem ser configurados como digital);

Vale também lembrar que todos os pinos suportam PWM e I²C;

Figura 1 Pinagem de NodeMCU-32S



Fonte: IOXHOP

### ADC e DAC

#### Analog-to-Digital Converter (ADC)

O Esp32 integra ADCs de 12 bits e suporta medições em 18 canais (analog-enabled pins). O ULP-coprocessador no ESP32 também é projetado para medir as tensões enquanto opera em modo sleep, que permite o baixo consumo de energia. A CPU pode ser despertada por uma configuração de limite e/ou através de outros gatilhos.

* + - 1. Digital-to-Analog Converter (DAC)

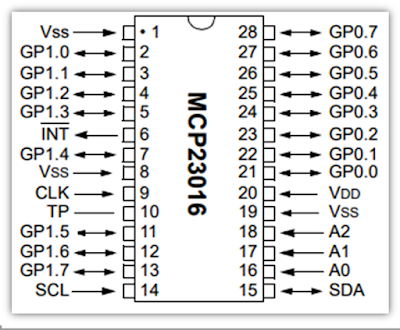
Dois canais DAC de 8 bits podem ser usados para converter dois sinais digitais em duas saídas de tensão analógica. Estes DAC duplos suportam a fonte de alimentação como referência de tensão de entrada e pode conduzir outros circuitos. Os canais duplos suportam conversões independentes.

## protocolo de comunicação i2c

* + 1. Utilização do I²C do MCP23016

Aqui temos o esquemático do expansor, que tem dois grupos de oito bits, ou seja, um total de 16 portas. Além de um pino de interrupção, ele tem o pino CLK, que é para ligar o capacitor e o resistor, que internamente estão ligados em uma porta lógica. Isso é para formar o clock, utilizando a ideia de um cristal oscilador, que precisa de 1MHz de clock. O pino TP serve para medir o clock. Os pinos A0, A1 e A2 são endereçamentos binários.

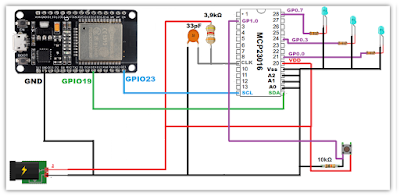
Figura 2 MCP23016 Pinagem



Fonte: datasheet mcp23016 [microchip technology](http://ww1.microchip.com/downloads/en/DeviceDoc/20090C.pdf)

Para montagem desse MCP23016 na NODEMCU-32(Dev. Kit) vamos utilizar uma montagem cedida por um especialista no assunto Fernando K, qual é diretor da Appsis, uma empresa de tecnologia digital. Sendo a montagem do Circuito teste para MCP23016 de ativação de 3 leds para teste e adaptação de código fonte.

Figura 3 Ligação MCP23016



Fonte: Fernando K.

## Circuitos

### Sensor de Corrente

Este sensor abaixo é um sensor não invasivo qual suporta até 100A e transforma para 50mA de saída, foi o modelo qual será utilizado no projeto pois é de fácil aplicação e de montagem em circuito também possuindo 2 fios fáceis de se ligar.

Figura 4 Sensor de Corrente

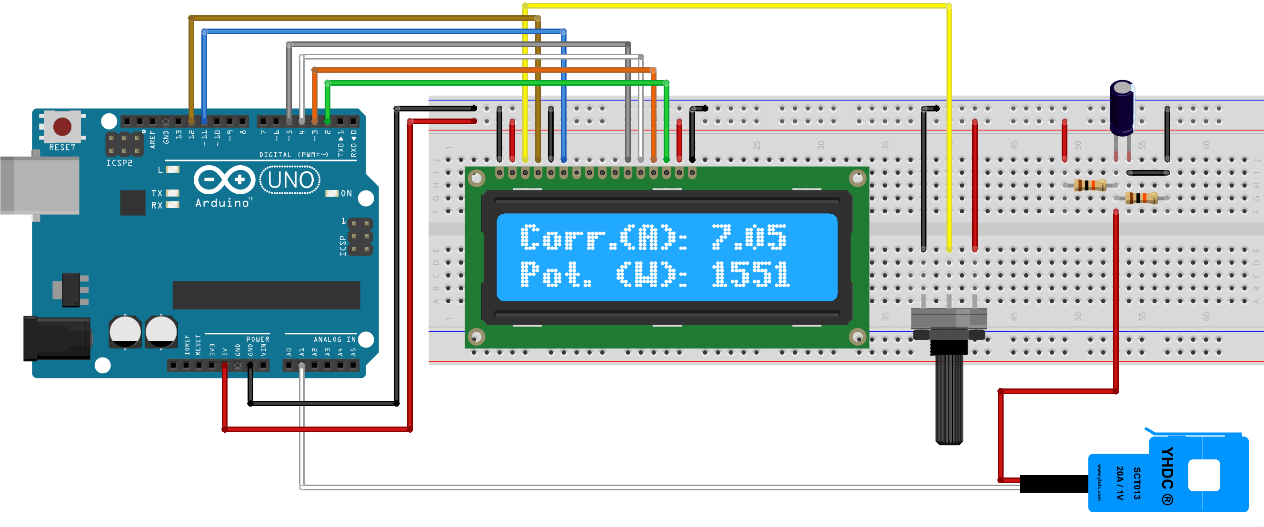


Fonte: datasheet sct013

### Esquema sensor de corrente

O desenho do sensor sendo colocado foi o mesmo dessa imagem a baixo, sendo utilizado para demonstração em tela no aplicativo. Sendo o fio branco Ligado no ESP32 em uma porta Analógica para medição.

Figura 6 Ligação Sensor De Corrente



Fonte: filipeflop

### Relés

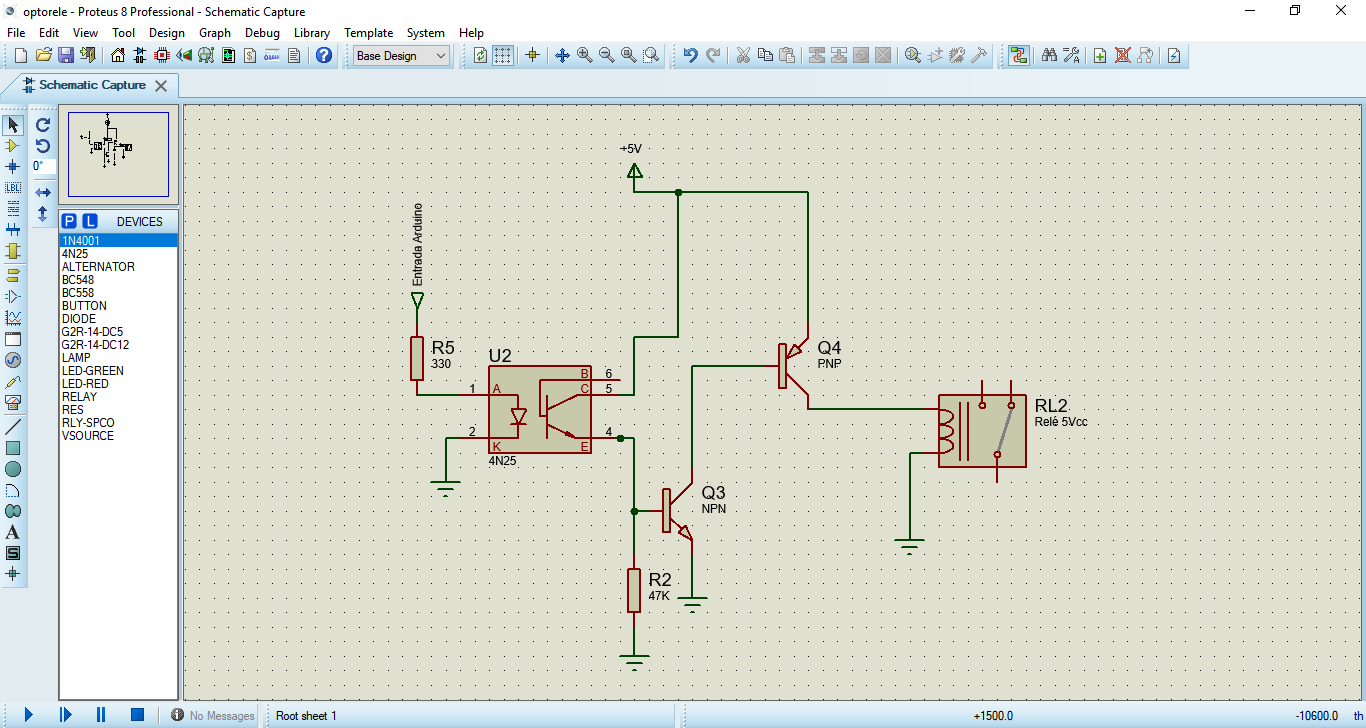
Relé é um dispositivo eletromecânico formado de um magneto (bobina) móvel, qual se desloca unindo dois contatos metálicos, relés utilizados em circuitos eletrônicos possuem 5 pinos, sendo 2 para a bobina, 1 contato comum, 1 contato Normalmente Fechado (NC) e 1 Contato Normalmente Aberto (NO). A utilização do relé faz com que a carga e o controle do circuito estejam isolados, podendo funcionar em diferentes tensões.

#### Circuito de Ativação

O circuito de ativação normalmente é visto no esquema, qual utiliza:

* 1-Optoacoplador;
* 1-Transistor PNP;
* 1-Transistor NPN;

Figura 6 Circuito de Ativação relé com baixa potência.



Fonte: Datasheet SRD-05VDC-SL-C

### Optoacoplador

O funcionamento do optoacoplador é baseado no efeito fotoelétrico, onde um feixe de luz infravermelha, produzido pelo diodo LED, polariza a base do fototransistor, forçando a condução entre base e emissor. Dessa forma, eleva-se a tensão BE (base-emissor) a 0,7 V, colocando este semicondutor em saturação. Retirado do Site: ”<https://www.embarcados.com.br/circuito-de-interface-para-microcontroladores/>”;

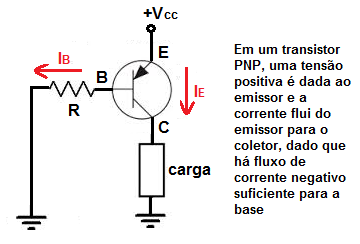
### Transistores

O Transistor é um componente semicondutor utilizado como amplificador ou interruptor de sinais ou energia elétrica. É composto de um material semicondutor com pelo menos três terminais para conexão com um circuito externo. Os tipos de transistores mais conhecidos e utilizados são os PNP, NPN e MOSFET.

#### PNP

Um transistor PNP recebe tensão positiva no terminal do emissor. A tensão positiva para o emissor permite que a corrente flua do emissor para o coletor, dado que há corrente negativa para a base (corrente que flui para fora da base para terra).

Figura 7 PNP

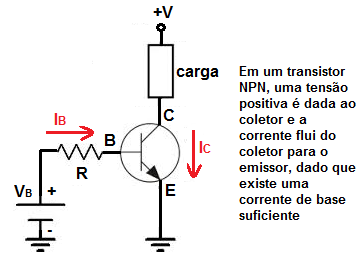


Fonte: learning about electronics

#### NPN

Um transistor NPN recebe tensão positiva no terminal do coletor. Essa tensão positiva para o coletor permite que a corrente flua do coletor para o emissor, dado que existe uma corrente de base suficiente para ativar o transistor.

Figura 8 NPN



Fonte: learning about electronics

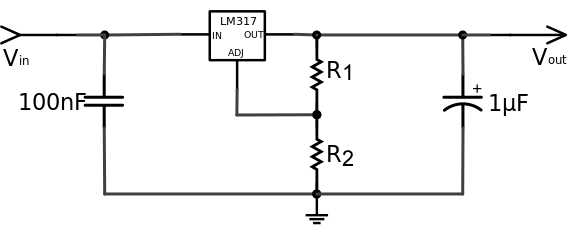
### Fonte para ligação módulos relés e outros sensores

Determinar fonte pela formula:

(1)

Definições extras feitas pelo datasheet do regulador de tensão as ser escolhido. Utilizado o Esquema abaixo:

Figura 9 Fonte 5V



Fonte: Eletrocode.

# Metodologia

## Socket;

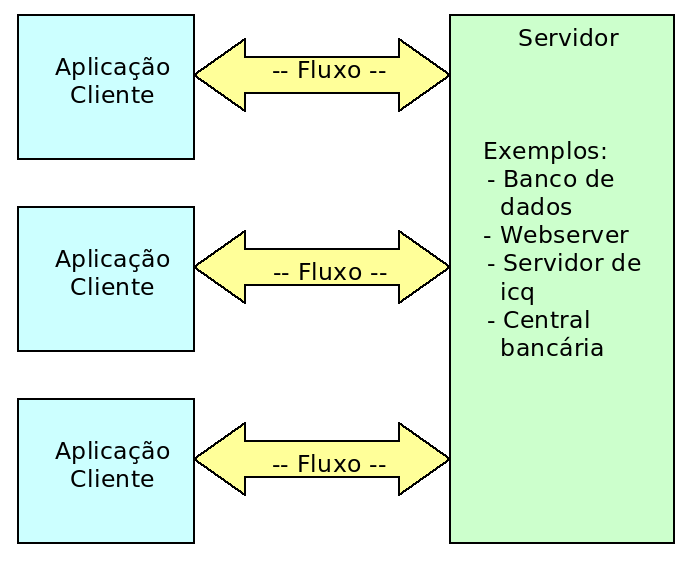
Socket é um mecanismo de comunicação, usado normalmente para implementar um modelo cliente/servidor, que permite a troca de mensagens entre os processos de uma máquina/aplicação servidor e de uma máquina/aplicação cliente. (Behrouz A. Forouzan, 2013).

Socket no método Java e também IDE Arduino é retirado por meio de API existentes em Java disponíveis na própria IDE NetBeans e possível instalação de biblioteca na IDE Arduino.

### Protocolo

O protocolo utilizado é o TCP (transmission Control Protocol), com ele é possível ter um fluxo entre dois dispositivos como no sistema qual vamos utilizar.

Figura 10 Diagrama básico de funcionamento de protocolo



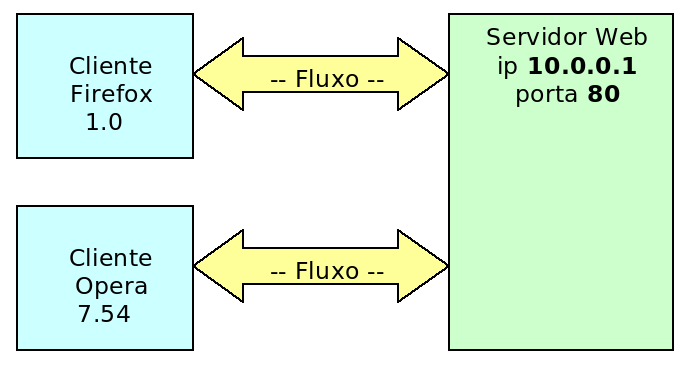
Adaptado de: Forouzan & Mosharraf

Utilizando esse protocolo podemos ter mais de um cliente ao mesmo tempo no mesmo servidor. As classes em Java já são disponibilizadas para os utilizadores por meio do pacote “java.net” qual pode ser implementado em qualquer Applet qual rode Java.

### Porta

Normalmente clientes possuem apenas um endereço físico, a porta é a solução para identificar diversas aplicações em uma máquina ou servidor. Assim como o IP serve para definir uma máquina na rede, a porta é um número de 2 bytes que varia de 0 a 65535, porém se todas as portas não estiverem disponíveis não é possível se conectar.

Figura 11 Funcionamento de Porta



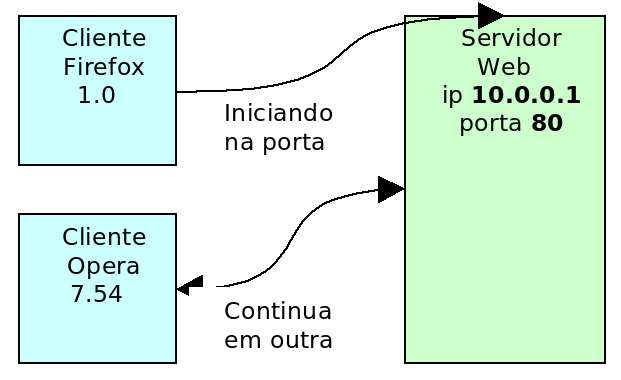
Adaptado de: Forouzan & Mosharraf

### Funcionamento Socket

Se um cliente se conecta a uma porta de um servidor enquanto ele não se desconectar dessa porta será impossível que outro cliente se conecte, porém o Socket vem com a função que quando o servidor aceita a conexão ele redireciona tal cliente para outra porta liberando a porta inicial sendo assim permite várias conexões de outros clientes.

No Java isso é feito pelos threads para que o processo de aceitar uma conexão seja rodado o mais rápido possível.

Figura 12 Funcionamento Socket



Adaptado de: Forouzan & Mosharraf

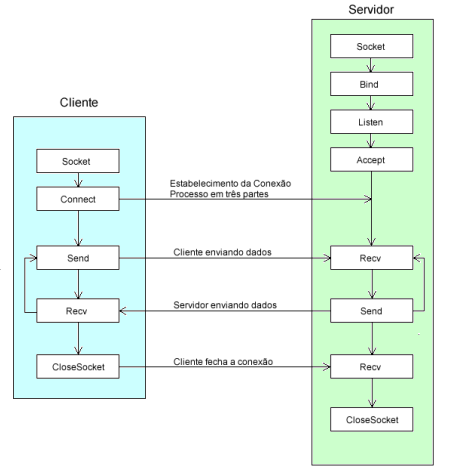
### Servidor

É a parte qual deve aceitar os pedidos de clientes, e servi-los com respostas conforme a linguagem utilizada, no caso o Java, teremos que abrir a conexão com a porta primeiramente, logo após aceitar a conexão pelo ip e pela porta ligada, a partir daí pode-se ler as informações que o cliente envia até que a conexão seja encerrada ou pelo cliente ou pelo servidor.

### Cliente

A tarefa do programa cliente é enviar mensagem para o servidor, primeiro é utilizado o comando qual utilize o servidor IP e porta qual foi definida no servidor, após isso o cliente pode começar a enviar comandos para o servidor para que ele os interprete, a conexão dura até que uma das conexões seja fechada.

Figura 13 Diagrama de funcionamento completo do Socket



Adaptado de: Forouzan & Mosharraf

## Hardware

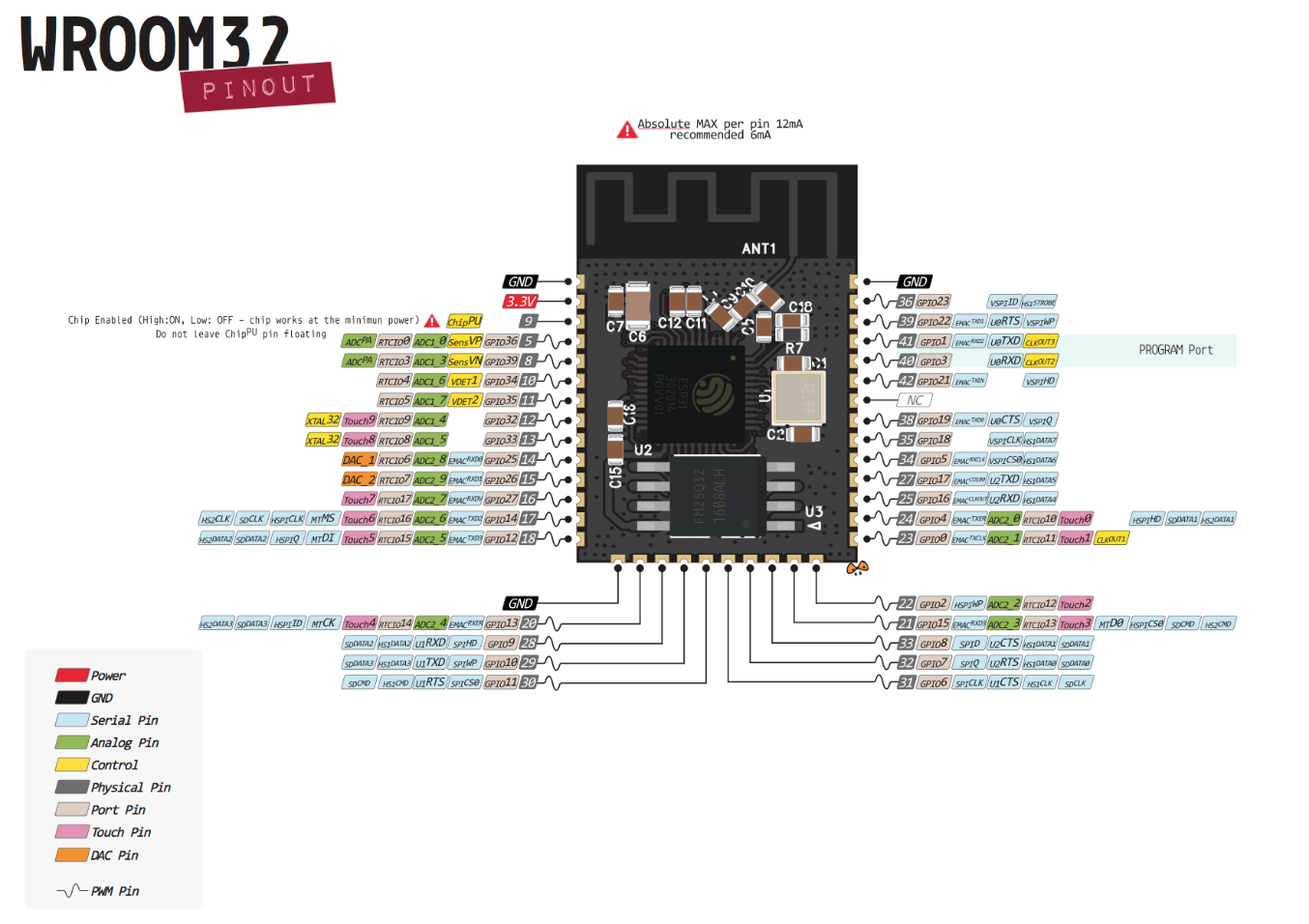
### ESP-32

O ESP32 possui 48 GPIOS, sendo que os pinos analógicos podem ser configurados como digitais.

Características do microcontrolador:

* Bluetooth 4.2 BLE
* Dual core 32bits LX6 Xtensa
* 448KB ROM
* 520KB SRAM
* 16KB SRAM no RTC

Figura 14 Esquemático do Chip WROOM32.



Fonte: Espressif (Fabricante).

O ESP32 também possui periféricos nativos com compartilhamento de pinos em alguns casos. Lista de periféricos disponibilizados na WROOM32, segundo a fabricante:

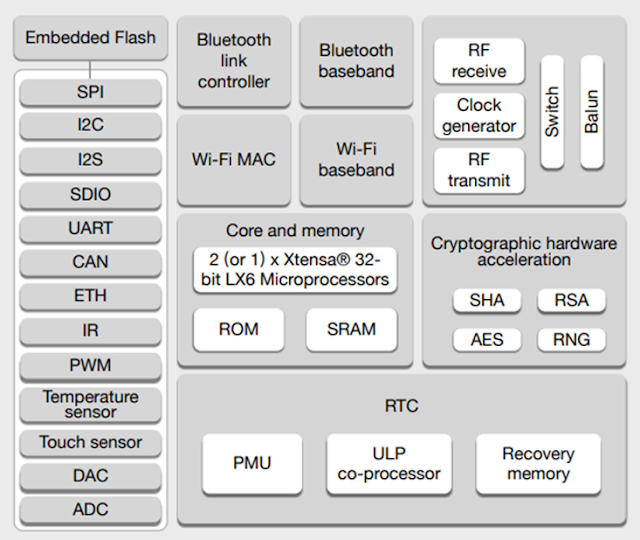
* 3x SPI
* 2x I2C
* 2x I2S
* SDIO/SD/MMC
* 3x UART
* CAN
* Ethernet 10/100Mbps
* IR (Infrared)
* 16x PWM
* Sensor de Temperatura interno
* Sensor touch de 10 canais
* ADC de 12bits e suporte para até 18 canais (multiplexado)
* DAC de 8bits e 2 canais

O ESP32 trabalha com tensões entre 2.2V e 3.6V, ele ainda conta com um modo de Ultra Low Power para quando ligado a uma bateria, mesmo com a ULP ligada ela continua com seu poder máximo. Além disso tem outras compatibilidades:

* Possui suporte a redes WiFi padrão 802.11 b/g/n
* O WiFi opera na faixa de 2.4 a 2.5GHz
* WiFi possui opções de segurança WPA / WPA2 / WPA2-Enterprise / WPS
* WiFi possui opções de criptografia AES / RSA / ECC / SHA
* Opera nos modos Station / SoftAP / SoftAP + Station/ P2P
* Possui Antena integrada
* Possui Bluetooth v4.2 BR / EDR e BLE (Bluetooth Low Energy)
* Opera em temperaturas na faixa de -40º a 85ºC
* Programável via USB, host ou WiFi (OTA / Over The Air)
* Compatível com a IDE do Arduino
* Compatível com módulos e sensores utilizados no Arduino

Segue abaixo o diagrama de blocos da ESP32.

Figura 15 Diagrama de Blocos ESP32S



Autor: Espressif (Fabricante).

## Software

Desenvolvimento de um aplicativo qual rode em um sistema operacional Windows superior ao Windows XP ou superior, qual terá pontos básicos:

* Desenvolver um sistema de gerenciamento dos usuários autorizados a utilizar o sistema de domótica;
* Desenvolver um sistema qual possa gerenciar todos os DOMOs presentes na residência;
* Desenvolver um sistema de gerenciamento programado dos DOMOs presentes na residência (agendamento de operação);
* Implementar o sistema bidirecional entre servidor, usuário e DOMO através da biblioteca de comunicação Sockets;
* Desenvolver um padrão de rotinas e mensagens para a comunicação bidirecional entre servidor, usuário e DOMO;
* Desenvolvimento de uma interface amigável e intuitiva, qual facilitaria a atuação e visualização, sem restrição de informações e funcionalidades disponíveis.
* Desenvolver um sistema de banco de dados utilizando MySQL, para armazenamento e comunicação de todo o sistema.

## Aplicativo Móvel

Desenvolver um aplicativo móvel qual atenda a versão Android 4.0.2, da Google, sendo assim traçados pontos básicos de desenvolvimento:

* Implementação de um sistema de comunicação bidirecional entre usuário, servidor e DOMO, através do protocolo de Sockets;
* Desenvolver um padrão de rotinas e mensagens para a comunicação bidirecional entre servidor, usuário e DOMO, seguindo as regras aplicadas no servidor de gerenciamento;
* Desenvolvimento de uma interface amigável e intuitiva, qual facilitaria a atuação e visualização, com restrição de informações e funcionalidades disponíveis ao do software de gerenciamento.

# ANÁLISE DOS RESULTADOS

# Pesquisa Avaliativa para implantação do sistema

# CONSIDERAÇÕES FINAIS

# Bibliografia

MONK, Simon. (12/2013). *Projetos com Arduino e Android, 1st edição*.

DEITEL, Paul, DEITEL, Harvey, WALD, Alexander. (01/01/2016). *Android 6 para Programadores, 3rd edição*.

GOODRICH, T., M., TAMASSIA, Roberto. (01/2013). *Estruturas de Dados e Algoritmos em Java, 5th edição*.

MACHADO, Prestes, R., FRANCO, Islabão, M., BERTAGNOLLI, Castro, S. D. (01/01/2016). *Desenvolvimento de Software III, 1st edição*.

FOROUZAN, A., B., MOSHARRAF, Firouz. (01/2013). *Redes de Computadores, 1st edição*.

EDELWEISS, Nina, GALANTE, Renata. (04/2011). *Estruturas de Dados - V18 - UFRGS*.

ALVES, Pereira, W. (06/2014). *Banco de Dados, 1st edição*

CARDOSO, M., V. (03/2008). *Sistemas de Banco de Dados, 1ª Edição, 1st edição*.

DATASHEET SCT013. Datasheet sct013. Disponível em: <http://www.mcielectronics.cl/website\_mci/static/documents/datasheet\_sct013.pdf>. Acesso em: 25 mai. 2018.

ELETROCODE. Reguladores de tensão. Disponível em: <http://www.eletrocode.com.br/reguladores-de-tensao/>. Acesso em: 25 mai. 2018.

FILIPEFLOP. Como fazer um medidor de energia elétrica com arduino. Disponível em: <https://www.filipeflop.com/blog/medidor-de-energia-eletrica-com-arduino/>. Acesso em: 25 mai. 2018.

IOXHOP. Ioxhop esp32s. Disponível em: <www.ioxhop.com>. Acesso em: 25 mai. 2018.

LEARNING ABOUT ELECTRONICS. Difference between an npn and a pnp transistor. Disponível em: <http://www.learningaboutelectronics.com/articles/difference-between-a-npn-and-a-pnp-transistor>. Acesso em: 25 mai. 2018.

ESPRESSIF. Espressif esp32. Disponível em: <https://www.espressif.com/en/products/hardware/esp32/overview>. Acesso em: 25 mai. 2018.