



**CENTRO UNIVERSITÁRIO DE BRASÍLIA -UniCEUB**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**Reinaldo Oliveira Neto**

**AUTOMAÇÃO DE ILUMINAÇÃO RESIDENCIAL UTILIZANDO  
MICROCONTROLADOR *ARDUINO* E *TABLET* IPAD VIA *WI-FI***

**Orientador: Prof. M.C. Maria Marony Sousa Farias**

Brasília

Junho, 2011

**Reinaldo Oliveira Neto**

**AUTOMAÇÃO DE ILUMINAÇÃO RESIDENCIAL UTILIZANDO  
MICROCONTROLADOR *ARDUINO* E *TABLET IPAD* VIA *WI-FI***

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof. M.C. Maria Marony Sousa Farias

Brasília

Junho, 2011

**Reinaldo Oliveira Neto**

**AUTOMAÇÃO DE ILUMINAÇÃO RESIDENCIAL UTILIZANDO  
MICROCONTROLADOR ARDUINO E TABLET IPAD VIA WI-FI**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof. Maria Marony Sousa Farias

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -FATECS.

---

Prof. Abiezer Amarilia Fernandez

Coordenador do Curso

**Banca Examinadora:**

---

Prof. Maria Marony Sousa Farias, mestre em Engenharia Elétrica  
Orientadora

---

Prof. José Julimá Bezerra Junior.  
Instituto Militar de Engenharia – Rio de Janeiro - RJ

---

Prof. Luis Cláudio Lopes de Araújo, Mestre.  
UniCEUB – Centro Universitário de Brasília

## **AGRADECIMENTOS**

A Deus, por não ter permitido que eu fraquejasse durante esta etapa.

Agradeço a minha orientadora, Prof. Maria Marony, por ter acreditado no meu trabalho, pela paciência e apoio ao longo destes anos.

Agradeço ao amigo e Prof. Luis Magno, por todo o apoio no desenvolvimento da monografia.

Aos meus amigos de faculdade em especial, Thiago Alencar, Leonardo Lima e Camilla Oliveira, por toda a atenção e conselhos durante a jornada na Universidade.

Aos meus pais por todo incentivo e criação.

## SUMÁRIO

AGRADECIMENTOS .....	4
SUMÁRIO.....	5
LISTA DE FIGURAS.....	6
LISTA DE TABELAS.....	7
RESUMO .....	8
ABSTRACT .....	9
CAPÍTULO 1 – INTRODUÇÃO .....	10
1.1 Apresentação do Problema .....	10
1.2 Objetivos do Trabalho .....	10
1.3 Justificativa e Importância do Trabalho .....	11
1.4 Escopo do Trabalho .....	12
1.5 Resultados Esperados .....	12
1.6 Estrutura do Trabalho .....	12
CAPÍTULO 2 – AUTOMAÇÃO NO CONTEXTO HISTORICO E SOCIAL .....	14
2.1. Taxonomia.....	14
2.2. A evolução histórica e social .....	15
2.3. Benefícios .....	17
2.4. Aplicações nos dias atuais .....	18
2.5. Integrando.....	20
2.6. Acessibilidade.....	22
CAPÍTULO 3 – BASES METODOLÓGICAS E REFERENCIAL TEÓRICO .....	24
3.1. Componentes de comunicação .....	24
3.2. Componentes Físicos.....	27
3.3. Componentes de <i>Software</i> .....	31
CAPÍTULO 4 – MODELO PROPOSTO .....	35
4.1. Apresentação Geral do Modelo Proposto .....	35
4.2. Camadas e seu funcionamento .....	37
4.3. Descrição do funcionamento .....	40
CAPÍTULO 5 - APLICAÇÃO PRÁTICA DO MODELO PROPOSTO .....	62
5.1. Apresentação da área de Aplicação do modelo .....	62
5.2. Descrição da Aplicação do Modelo.....	62
5.3. Resultados da Aplicação do Modelo .....	64
5.4. Custos do modelo proposto .....	65
5.5. Avaliação Global do Modelo.....	66
CAPÍTULO 6 - CONCLUSÃO.....	68
6.1. Conclusões.....	68
6.2. Sugestões para Trabalhos Futuros .....	69
REFERÊNCIAS .....	70
APÊNDICE A.....	72
APÊNDICE B.....	75

## LISTA DE FIGURAS

Figura 3.1 – Esquemático ligando relé e transistor ao <i>Arduino</i> .....	28
Figura 3.2 – Placa <i>Arduino UNO</i> .....	29
Figura 3.3 – Crescimento do iPad no Mercado .....	30
Figura 3.4 – Ambiente de personalização <i>TouchOSC</i> .....	32
Figura 3.5 – Interface <i>TouchOSC</i> .....	34
Figura 4.1 – Esquemático .....	35
Figura 4.2 – Fluxo das Camadas .....	37
Figura 4.3 – Planta baixa .....	40
Figura 4.4 – Foto maquete sendo construída.....	41
Figura 4.5 – Ligações Físicas .....	42
Figura 4.6 – Ligações Físicas .....	42
Figura 4.7 – Ambiente de desenvolvimento <i>Arduino</i> .....	44
Figura 4.8 – Configuração da interface serial.....	44
Figura 4.9 – Configuração do modelo de placa escolhido.....	45
Figura 4.10 – Definindo os pinos no código.....	46
Figura 4.11 – Configuração da porta de entrada.....	47
Figura 4.12 – Leitura da porta com taxa de transmissão “9600” .....	48
Figura 4.13 – Comandos para ligar ou desligar lâmpadas.....	49
Figura 4.14 – Escrita do comando na no pino respectivo.....	50
Figura 4.15 – Envio de comandos via Serial Monitor .....	51
Figura 4.16 – Interface do aplicativo <i>TouchOSC editor</i> .....	52
Figura 4.17 – Criando Botões.....	52
Figura 4.18 – Personalizando a interface.....	53
Figura 4.19 – Importando a Biblioteca <i>oscP5</i> .....	54
Figura 4.20 – Código contendo as bibliotecas e objetos .....	55
Figura 4.21 – Definindo parâmetros de comunicação .....	56
Figura 4.22 – Função <i>oscEvent</i> .....	57
Figura 4.23 – Escrita do comando na porta serial .....	58
Figura 4.24 – Configurando endereço IP e porta de saída.....	59
Figura 4.25 – Sincronizando a nova interface com o iPad .....	60
Figura 4.26 – Selecionando a interface gráfica para exibição .....	61
Figura 5.1 – Detalhe do <i>hardware</i> microcontrolador e lâmpada.....	63
Figura 5.2 – Teste de funcionamento do sistema .....	64

## LISTA DE TABELAS

Quadro 1 – Cômodos x Pinos .....	43
Quadro 2 – Tabela com referencias de comando.....	45
Quadro 3 – Custo total dos equipamentos utilizando no projeto.....	66

## RESUMO

Este trabalho apresenta a especificação e implementação de um protótipo para controle de automação residencial à distância utilizando um iPad e um microcontrolador *Arduino*. Esse conjunto de dispositivos tem como objetivo automatizar a iluminação de uma residência, provendo facilidade e dinamismo para usuários deficientes ou não. De forma inovadora o *tablet* iPad poderá controlar toda a iluminação de uma residência através de simples toques em tela e comunicação *Wi-Fi*. Para isso, são necessários, além do *tablet* iPad com sua interface gráfica personalizada, um *hardware* microcontrolador *Arduino*, o *Software* livre *Processing*, para leitura dos dados enviados pela rede *Wi-Fi* ao computador e reenvio ao microcontrolador, maquete residencial para a representação de uma casa. O iPad enviará pacotes UDP contendo protocolo OSC via *Wi-Fi* ao computador com *Software Processing* instalado, este *Software* encaminhará ao microcontrolador a seqüência de ações necessárias, o qual realizará a função desejada, acender ou apagar as luzes de um determinado cômodo de uma residência, representada aqui como uma maquete.

**Palavras Chave:** iPad, *Arduino*, Automação Residencial, OSC *protocol*

## ABSTRACT

This paper presents the specification and implementation of a prototype to control remotely using an iPad and an *Arduino* microcontroller. This set of devices have the goal to automate the lighting of a residence, providing ease and dynamism for users with disabilities or not. So the innovative iPad tablet can control all the lighting in a home through simple touch-screen and Wi-Fi. For this communication will be needed beyond the iPad tablet with its custom GUI, an *Arduino* microcontroller hardware, free Software Processing, for reading the data sent over Wi-Fi network to the computer and forwarded to the microcontroller. The iPad will send UDP packets containing OSC protocol over Wi-Fi access to computer processing Software installed, this Software will send the microcontroller to the sequence of actions required, which will perform the desired function, turn on or turn off the lights for a particular room in a residence, represented here as a model.

**Keywords:** iPad, *Arduino*, Home automation, OSC protocol.

## CAPÍTULO 1 – INTRODUÇÃO

### 1.1 Apresentação do Problema

Atualmente a maneira mais habitual de se acionar uma lâmpada nas residências é através do uso de interruptores, no entanto este procedimento exige locomoção e a necessidade de estar no mesmo cômodo, com a crescente popularização e incentivo de produção dos *tablets* no mercado nacional, porque não utilizarmos esses novos *gadgets* tecnológicos de maneira inovadora dentro de nossas residências?

### 1.2 Objetivos do Trabalho

O objetivo geral do trabalho é desenvolver um protótipo que visa automatizar a iluminação de uma residência via Wi-Fi, ligado a um microcontrolador *Arduino* usando como plataforma de comunicação um *tablet* móvel iPad. Possibilitando acender ou apagar a distância as luzes de uma residência. Esse procedimento se dará através de toque de determinados botões na tela do dispositivo móvel, possibilitando escolher a luz de qual cômodo será ligada. Para o correto funcionamento da solução algumas tarefas deverão ser executadas.

Como objetivos específicos destacam-se o desenvolvimento de uma arquitetura para implementação física do projeto. Ou seja, implementar no *Arduino*, de forma correta, as lâmpadas que representam as luzes da casa, ligando nas saídas analógicas do microcontrolador com a finalidade de prover a energia necessária quando o comando for enviado pelo dispositivo móvel.

A programação do hardware microcontrolador *Arduino*. Ou seja, desenvolvimento do código de programação na linguagem própria do microcontrolador, para que este receba as informações do iPad ao computador e interprete de forma correta qual comando foi enviado.

A adequação e programação da interface gráfica para o iPad do aplicativo que enviará protocolos OSC, ou seja, desenvolvimento da interface gráfica com os botões necessários para representar cada cômodo de uma residência.

A programação da camada de comunicação entre o computador e o *Arduino*. Ou seja, desenvolvimento do código de programação na linguagem *Processing*, essa camada receberá as informações via *Wi-Fi* enviada pelo dispositivo móvel ao computador, interpretando e enviando o comando correto ao hardware *Arduino*.

A realização e adequação da maquete residencial. Ou seja, desenvolvimento de uma maquete em escala 1/33 de uma residência, para a representação dos ambientes e adequação das ligações elétricas na maquete.

### **1.3 Justificativa e Importância do Trabalho**

Com a popularização da internet, computadores, e nos últimos anos, os dispositivos móveis e redes sem fio as pessoas tem alterado o modo como se comunicam, trabalham e se entretêm. Agora, com o incentivo para a produção dos *tablets* em solo nacional, proporcionará um aumento da quantidade destes dispositivos na vida dos brasileiros, instigando a criatividade das pessoas a se familiarizarem com esses novos equipamentos eletrônicos, novas interfaces e com o conceito de interligação de dispositivos em rede. Também pode-se esperar mudanças nas rotinas domésticas, potencializando a possibilidade das residências deixarem de ser inertes e se tornarem dinâmicas e interativas com seus moradores.

É possível ver claramente as vantagens que o avanço da ciência trouxe e as comodidade que a tecnologia atual proporciona, possibilitando ainda um aumento de qualidade de vida para uma parcela populacional que possui desvantagens em relação a atual forma de organização social. Por meio deste pressuposto que o projeto em questão pretende aplicar este conceitos tecnológicos a favor de pessoas com dificuldade de acessibilidade promovendo uma maior inclusão social.

## 1.4 Escopo do Trabalho

O projeto consiste na implementação automação de iluminação residencial. O protótipo será demonstrado usando uma maquete com a iluminação correspondente de cada cômodo.

Não faz parte do escopo do projeto desenvolver o controle de automação de iluminação em uma residência real, nem o desenvolvimento de um aplicativo em linguagem específica para o sistema operacional da *Apple*, o iOS.

O projeto consiste em desenvolver mais uma funcionalidade para o *tablet* iPad, permitindo a interligação entre o microcontrolador *Arduino*, *Software Processing* e o aplicativo *TouchOSC*.

## 1.5 Resultados Esperados

Com a finalização do projeto espera-se ter uma maquete de iluminação residencial totalmente automatizada, podendo portanto acionar a distância determinadas ações como acender ou apagar uma luz através do *tablet* iPad usando comunicação sem fio.

## 1.6 Estrutura do Trabalho

Além deste capítulo introdutório esta monografia está estruturada em mais cinco capítulos e organizada da seguinte maneira:

Capítulo 2 – Automação residencial no contexto histórico e social – Esse capítulo apresenta uma descrição profunda e detalhada da como o conceito da automação residencial evoluiu durante os tempo, se tornou um símbolo de futuro livre de tarefas domésticas, facilidades e comodismo para seus usuários, além de um levantamento de como a domótica vem sendo trabalhada nos dias atuais tornando-se conceito de futurismo aliando o campo de aplicação tecnológica e se integrando com o espaço arquitetônico e de telecomunicações. Ainda mostrando as vantagens de existirem casas inteligentes, podendo ser considerada um promotor de qualidade de vida sendo que a construção destes sistemas tecnológicos

contribuiriam para a acessibilidade de pessoas com necessidade especiais, corroborando ainda para a ascensão de sua autonomia e independência.

Capítulo 3 – Bases metodológicas para resolução do problema – Neste capítulo é apresentado o referencial teórico e tecnológico que compõe a teoria por trás do projeto. Além de tratar da aplicação dos conteúdos vistos nas disciplinas pertinentes e em outras publicações, visando a implementação de soluções para o problema levantado no capítulo anterior. Para isso, serão apresentados sucintamente as ferramentas que auxiliarão no desenvolvimento do projeto.

Capítulo 4 – Modelo Proposto – Este capítulo trata do desenvolvimento e a visão do projeto, especificando as questões de hardware e Software em uma explicação detalhada e a forma de funcionamento, constitui num modelo para o desenvolvimento de um protótipo de automação de iluminação residencial, baseando-se para isto no arcabouço teórico desenvolvido nos Capítulos 2 e 3. Apresentará ainda um “passo a passo” de como implementar e demonstrar de maneira prática o protótipo de automação.

Capítulo 5 – Aplicação do Modelo Proposto – Este capítulo trata especificamente das metodologias de aplicação e testes para o modelo proposto, conta ainda com um modelo de resolução do problema, baseando-se para isto no arcabouço teórico desenvolvido nos Capítulos 2 e 3.

Capítulo 6 – Conclusão – Esse capítulo trata especificamente do final do projeto com suas conclusões e apresentando propostas para projetos futuros.

## CAPÍTULO 2 – AUTOMAÇÃO NO CONTEXTO HISTÓRICO E SOCIAL

Um dos principais marcos de evolução na história das residências foi sem dúvida a introdução da eletricidade. Não somente pelo fato de transformar o modo como elas eram iluminadas, mas também por possibilitar o desenvolvimento de novas tecnologias que transformaram o modo que vivemos. A iluminação elétrica, controlada independentemente em cada cômodo, estendeu o período noturno modificando os hábitos das pessoas e o ciclo de vida das famílias.

A localização das tomadas tinham, e de certa forma ainda têm, uma função social na casa. Determinando regiões funcionais dentro da residência e subdividindo os cômodos.

Uma outra transformação na evolução das residências automatizadas aconteceu graças ao telefone. No início eram aparelhos fixos, presos a fios e transmitiam apenas voz mas com o passar dos anos a sua mobilidade e rápida evolução tecnológica veio a tona possibilitando além da comunicação por voz a capacidade de tráfego de dados. Estes aparatos acabaram por agregar cada vez mais funcionalidade ao ponto de serem comparados a computadores. Um exemplo desta evolução é o iPhone, o *smartphone* da *Apple* (APPLE, 2008), além de integrar alta tecnologia e uma interface fácil de usar, possui custo relativamente acessível. Com a possibilidade de agregar com capacidade de toque e maior tamanho surgiram os *tablets* que por esta razão possibilitam maior acessibilidade a deficientes.

Com toda esta evolução tecnológica surgiu a idéia do autor sobre pesquisa de casas automatizadas e integração destes serviços de uma maneira mais eficiente, inteligente e flexível.

### 2.1. Taxonomia

“Automação residencial é uma coleção de equipamentos, sistemas e subsistemas, que mantêm habilidade para interagir entre si, permitindo o estabelecimento de funções independentes.” (MURATORI, 2004) “É a atuação de dispositivos nas funções de elétrica, hidráulica e ar condicionado, permitindo o uso customizado de aparelhos elétricos e

garantindo economia de energia elétrica e água.” (BOLZANI, 2004) “Inclui o uso de equipamentos especializados que podem controlar lâmpadas, eletrodomésticos, aquecedores, ar condicionado, e perceber em que local da casa as pessoas estão.” (MEYER, 2004)

Automação de residências são soluções que utilizam equipamentos microcontrolados de maneira que se integrem e interajam entre si, trocando informações e tomando decisões orientadas pelo usuário ou a ele para proporcionar conforto, acessibilidade e segurança.

A Automação Residencial se fundamenta na integração dos sistemas constituintes de uma residência, como por exemplo seus eletrodomésticos, equipamentos de áudio, vídeo e informática, persianas, portões e portas automáticas, dentre outros, que são os sistemas autônomos, presentes hoje em dia nas residências. Dessa forma, o termo Automação Residencial, também denominada Automação Doméstica ou Domótica, representa a utilização de processos automatizados em casas, apartamentos e escritórios (BOLZANI, 2004).

## **2.2. A evolução histórica e social**

A história de casas automatizadas é longa e sempre esteve presente em livros, filmes, imprensa e internet e tem revelado que tão importante é a descoberta de novas tecnologias quanto as aplicações que impulsionam sua utilização. No fim da década de 1870, Thomas Edison inventou a lâmpada comercial e impulsionou todo o mercado de equipamentos elétricos. A implantação de infraestrutura para iluminação abriu as portas para a criação de novos serviços e muitas outras aplicações.

No início a idéia de automação era focada principalmente no meio industrial, pois facilitaria controle ou linhas de produção, com o passar dos tempos a automação começou a aparecer em meios comerciais, edifícios condominiais. Mas agora o momento é a para a automação residencial, um mercado que passa se tornar realidade no Brasil com soluções diferenciadas e inteligentes, voltadas principalmente para o conforto e acessibilidade do usuário.

Juntando a automação residencial com a necessidade de poder controlar e monitorar à distância, chega-se aos dispositivos móveis.

“Os dispositivos móveis oferecem uma conectividade que outros dispositivos não possuem. A tendência é que o desenvolvimento de aplicações para os dispositivos móveis aumente significativamente em poucos anos.” (DEPINÉ, 2002).

No ano de 1996, Besen apresentou uma solução para controle de um ambiente residencial através de um microcomputador, controlando variáveis como: iluminação, temperatura e supervisão de janelas e portas. O sistema possui dois módulos: o módulo “Controlador” que foi desenvolvido utilizando o microcontrolador 8031, responsável pelo controle das variáveis; e pelo módulo “Supervisor” que opera em um microcomputador e tem como objetivo interagir com o usuário e com o “Controlador”.

Depiné em 2002 e Schaefer em 2004, apresentaram soluções que trabalhavam com dispositivos móveis. O primeiro apresentou uma solução para cálculo de planilha de rally de regularidade em um celular utilizando a plataforma *Java 2 Micro Edition (J2ME)*, onde o usuário entra com os dados do percurso e a solução retorna o tempo ideal para percorrê-lo. Já o segundo mostrou uma solução para coletar e transmitir informações através de dispositivos móveis e alimentar uma base de dados remota para controle de gastos em viagem, aplicado a uma empresa transportadora. Estes gastos são informados pelos motoristas dos caminhões que estão em viagem e enviados via e-mail. A base alimentada remotamente serve para geração de gráficos, consultas e relatórios das despesas, podendo ser gerados por período e/ou número do celular pela qual as despesas foram informadas.

Krüger em 2002 apresentou uma solução de baixo custo para monitoramento de segurança predial utilizando recursos da internet. Desenvolveu um módulo para o monitoramento de portas, janelas e de presença. Esses módulos comunicavam-se com um microcomputador através da porta paralela. As principais funções do sistema instalado no microcomputador são enviar mensagens eletrônicas ao usuário e gerar um arquivo de log com as atividades detectadas pelo monitoramento.

Galvin em 2004 apresentou a especificação e o desenvolvimento de um protótipo de *Software* para integração e troca de dados com um aplicativo cliente/servidor de uma empresa através de dispositivos móveis. A aplicação visa auxiliar o trabalho de relacionamento entre empresa e clientes diretamente no campo, disponibilizando informações importantes para o usuário sobre seus clientes e também fazendo a parte de força de vendas no front.

O desenvolvimento desse trabalho prevê a implementação de um protótipo de controle de automação de iluminação residencial a distância, onde será utilizado um dispositivo móvel que suporte envio de pacotes com mensagens OSC. Para o ambiente residencial, prevê-se o desenvolvimento de uma maquete ligada a circuitos de iluminação e *hardware* microcontrolador *Arduino*. No desenvolvimento da programação do *hardware* será usada linguagem de programação *Arduino*, para o desenvolvimento da camada de interface lógica computacional entre o dispositivo móvel e o *hardware* microcontrolador, será usada linguagem *Processing* e para a camada de interface gráfica será utilizado editor de interface para *TouchOSC*.

### **2.3. Benefícios**

Assim como outras novidades postas no mercado, automação residencial inicialmente é recebida pelos usuários como um símbolo de status e modernidade. Com o passar do tempo a aceitação da nova tecnologia passa a ser sinal de conforto, integração, acessibilidade e acesso remoto. E por fim, ela se tornará uma necessidade vital e um fator de economia.

Entre os setores que se beneficiariam com o a automação residencial, o imobiliário teria vantagens pelo fato de classificar imóveis de acordo com o nível de tecnologia empregado. Os proprietários e os construtores poderiam ser capazes de avaliar o retorno do capital investido em tecnologia para diferenciar seus imóveis dos que não possuem nenhum benefício de automatização e assim atrair compradores. Do outro lado da cadeia, os fabricantes de produtos podem melhorar o foco de seus produtos e o marketing associado a eles. Da mesma forma que a localização do imóvel, a quantidade de vagas de estacionamento e os serviços de segurança são fatores importantes que posicionam o imóvel em uma escala de valores, a presença ou não de tecnologia pode se tornar também um aspecto relevante no processo de decisão da compra.

## 2.4. Aplicações nos dias atuais

Muito se fala sobre novidades tecnológicas, tendências futuristas e outras soluções que poderão equipar nossos lares no futuro. Este é o problema, estamos sempre falando em futuro, no entanto com as atuais facilidades e até mesmo incentivo de produção de novas tecnologias em solo nacional nos deixa perceber que já é a hora de vivermos o presente da automação residencial.

Nas condições atuais o ambiente de uma “casa inteligente” pode apresentar diversos outros subsistemas, que podem de certa maneira possuir integração e controle unificado. Isso vem sendo proporcionado devido à rápida evolução de produtos e equipamentos eletrônicos e da informática, integrados podem controlar cortinas e persianas, áudio, vídeo, som ambiente, TV, segurança, iluminação, climatização, telefonia, dentre outras utilidades.

A seguir, são descritos alguns dos principais sistemas que estão dentre os mais empregados atualmente no âmbito da automação residencial.

### 2.4.1. Controle de iluminação

Os sistemas de controle de iluminação podem prover variadas funções para controlar a luminosidade de uma residência, um dos fatores que mais chamam a atenção do usuário é a possibilidade de economia proporcionada por soluções de detecção de presença ou controle de intensidade de luz, conhecida como dimerização. Podem trabalhar em conjunto com sistema de entretenimento proporcionando a intensidade de iluminação correta para cada situação, assistir filme, ler livros, dentre outros.

Ainda podem trabalhar juntamente com projetos arquitetônicos provendo cenários diferenciados para os ambientes da casa, promovendo temas como romântico ou festivo. Pode também ser programado para automaticamente ligar ou desligar, simulando a presença de pessoas mesmo quando a casa esta inabitada.

Sistemas de iluminação mais inteligentes podem incluir ainda timers ou sensores de luz solar, permitindo que as lâmpadas sejam acesas quando a luz solar for insuficiente ou conforme um horário programado.

#### 2.4.2. Sistemas de segurança

No ambiente de automação a instalação de equipamentos de monitoração e vigilância eletrônica podem ser considerados como sistema de segurança. Tem como principal objetivo aumentar a segurança e monitoramento remoto por parte dos seus moradores.

Os sistemas de segurança e alarme podem estar integrados a uma central de controle com circuito de TV fechado, sendo monitorada a distância, assim como sensores de presença, alarmes, acionamento de portões permitindo o controle de acesso por meio de internet.

#### 2.4.3. Sistemas de Entretenimento

Os sistemas de entretenimento podem, por exemplo, administrar as funções de equipamentos de áudio e vídeo através de um controle centralizando, eliminando o uso de diversos controles remotos espalhados nas residências. Outro exemplo de sistema de automação focado para o entretenimento, pode ser o fato de o controle de áudio e vídeo podendo ser distribuídos permitindo criar zonas de ambiente de som em diferente cômodos da casa, reproduzindo somente nos ambientes desejados.

#### 2.4.4. Sistemas de climatização

Os sistemas de climatização devem prover a temperatura ideal e agradável de acordo com as temperaturas ambiente, acionando ou desligando automaticamente aquecedores e condicionadores de ar, em determinados horários, através da coleta de fatores externos ou através da internet.

#### 2.4.5. Outros sistemas

Pode se automatizar acionamento de bombas de limpeza, irrigação de jardim, monitoramento de medições de água ou gás, dentre outras aplicações.

### 2.5. Integrando

#### 2.5.1. Integração com Arquitetura e Engenharia

No âmbito de integração com arquitetura e engenharia, vários fatores devem ser observados. As exigências de adequação dos imóveis modernos estão cada vez mais críticas, além de constituir as exigências de controle de iluminação, sistemas de segurança, entretenimento entre outros, estes devem ser idealizados e projetados prevendo suas integrações. Desta forma, é nítido a dependência desses sistemas com relação a sua instalação elétrica, fontes de alimentação e posicionamento de dispositivos, sendo então sugerido a implementação com acompanhamento de projetistas e arquitetos para garantir uma solução otimizada e integrada. Além destes, um projeto bem desenvolvido garantiria um melhor aproveitamento dos recursos empregados, prevendo a escalabilidade destes sistemas, permitindo futuras expansões, adições de equipamentos, atualização de *hardware* e *Software* e uso de novos dispositivos de controle. O projetista, então, deve conhecer bem as diversas tecnologias existentes, de modo a elaborar o ambiente adequado para cada caso.

Segundo OLIVEIRA,

Nos Estados Unidos, acredita-se que o investimento em sistemas de automação residencial corresponda em média a 10% do custo total da obra, com um retorno em médio prazo na forma de racionalização dos serviços de manutenção e economia nos gastos com energia, água e gás, na ordem de 30%.

De acordo com MURATORI, as fases de um projeto integrado podem ser citadas como segue:

- a. Desenho da infra-estrutura: distribuição e dimensionamento de conduítes, caixas de passagem, quadros de distribuição e demais conexões entre os sistemas;
- b. Compatibilização: eliminação de sobreposições ou redundâncias relativas aos aspectos dos diversos sistemas envolvidos no projeto integrado;
- c. Detalhamento da fiação e cabeamento: especificação das características técnicas do cabeamento especial necessário ao projeto;
- d. Planilhas de cargas automatizadas: listagem dos equipamentos e itens que serão incluídos na “automação” do imóvel, para um correto dimensionamento do sistema de automação e seus acessórios;
- e. Especificação de equipamentos: os equipamentos locados no projeto, bem como as características técnicas e especificações gerais desses equipamentos, devem ser listados no memorial descritivo que acompanha o projeto;
- f. Detalhamento das funcionalidades: descrição das funções executadas pelos equipamentos escolhidos, funcionando como “manual do usuário” para consultas e manutenção futura do sistema;
- g. Programação: muitas vezes não é atribuição do projetista, podendo ser delegada a um profissional mais familiarizado com determinados equipamentos.

#### 2.5.2. Integração com novas Tecnologias

Entre as principais preocupações dos projetista ou instaladores de sistemas de automação deve ser citada a integração entre estes sistemas e novas tecnologias. Apesar dos novos produtos possuírem tecnologias complexas, dispõem de interfaces amigáveis e que podem ser operadas com facilidade pelo usuário final.

Quando um grande numero de produtos trabalham sem comunicação e integração, acabam virando uma dificuldade para o usuário final, por exemplo, o fato de existirem dois sistemas de controle de iluminação que não conversam entre si, atuando de forma separada, as tecnologias acabariam por entrar em conflito. No entanto quando se prevê essa automação no início de um projeto de uma residência, acarretaria em economia para o usuário, diminuído a quantidade de cabos e ainda aumentando as possibilidades dos sistemas.

### 2.5.3. Integração com Telecomunicação

Entre os fatores que mais impressionam o usuário, nem sempre a engenharia por trás da automação está em primeiro lugar, e sim a forma moderna que ele estará controlando e acessando tudo.

Hoje um dos fatores que podem se tornar um grande atrativo para a automação residencial é permitir ao usuário o controle dos processos através de *tablets* e celulares, especialmente os produtos da *Apple*, iPad, iPhone ou iPod. Paralelamente a isso, está o fato de se poder controlar o sistema residencial de fora da casa usando variadas redes de telecomunicação.

O iPad é uma das interfaces com o usuário mais comentadas nos últimos tempos. Apresentando uma tela de 9,7" e um ótimo desempenho.

Disponibiliza recursos como personalização de telas, uso acelerômetro para controle de cargas, *multi-touch*, acesso a Internet entre outros.

## 2.6. Acessibilidade

Os recursos como personalização de telas e a facilidade das telas *touch-screen* têm sido remodeladas para facilitar o acesso às novas funcionalidades da casa por todos os moradores, sejam eles crianças, adultos, idosos ou portadores de alguma deficiência. Os sistemas de controle residenciais também têm evoluído com o compromisso de não só beneficiar os moradores com o máximo em conforto e segurança, mas também utilizar os recursos eficazmente.

Segundo Farias & Buchalla, a funcionalidade e a incapacidade dos indivíduos são determinadas pelo contexto ambiental onde as pessoas vivem”, ou seja, se o ambiente em que estes indivíduos vivem estiver adaptado às suas necessidades e capacidades, a sua deficiência pode ser “mascarada”.

Ainda contando com essa afirmação pode se garantir que se um ambiente tiver características de facilitação, o aumento da qualidade de vida irá aumentar.

Dessa maneira se encaixa no âmbito deste projeto o conceito de acessibilidade, que pode ser definido como processo de facilitação de acesso das pessoas a ambientes ou atividade, ampliando as facilidades e capacidades por parte de pessoas portadoras de algum tipo de deficiência.

Com a acessibilidade proposta pelo protótipo podemos trazer em caráter macro a inclusão social promovida a este tipo de pessoas.

Segundo dados do censo de 2000, aproximadamente 14,5% da população brasileira é portadora de algum tipo de deficiência mental, auditiva, visual ou motora, dentre estes tem-se aproximadamente 6% com deficiências físicas. A necessidade de acessibilidade, portanto, se mostra verdadeiramente necessária na sociedade brasileira.

Sob forma isolada, pode-se de fato encontrar diversos sistemas de automação residencial que realizam diversas tarefas como acendimento automático de luzes, abertura automática de portas, climatização do ambiente, entre outros exemplos. O caráter diferencial do projeto aqui proposto é a personalização e a facilidade de passar o controle da situação para a mão do deficiente. Por este motivo foi proposta a utilização do iPad, por ser uma plataforma totalmente intuitiva, com tela em tamanho ideal para a proposição de botões em tamanho acessível para os deficientes.

## CAPÍTULO 3 – BASES METODOLÓGICAS E REFERENCIAL TEÓRICO

O projeto compreende três divisões básicas: comunicação, componentes físicos e *Softwares*. A comunicação se dá através *wifi* e USB, utilizando como protocolos de comunicação UDP e OSC. A parte dos componentes físicos pode ser novamente dividida em partes menores, sejam elas: lâmpadas, relês, diodo, transistor, resistor, LED, microcontrolador, *notebook*, e iPad. Já a parte de *Software* será subdividida em três ambientes de desenvolvimento, sendo eles: *Software Arduino*, *Software TouchOSC interface editor* e *Software Processing*, um *Software* para o desenvolvimento da planta baixa da maquete residencial: o AutoCAD, um aplicativo para iOS: o *TouchOSC*.

“Basicamente, computação física é uma forma de comunicação entre o mundo físico real e o mundo virtual do computador. A transformação de energia é princípio básico necessário para tornar isso possível” (SEBASTIAN, 2008).

O dispositivo iPad juntamente com sua interface gráfica, transforma o fato de pressionar um botão virtual na tela do dispositivo em energia elétrica, que pode ser utilizada e processada por um sistema computacional (circuitos, microcontrolador e a parte de *Software* - programa - a eles associado). Esse sistema, por sua vez, emitirá uma resposta, na forma de energia elétrica, que por meio de atuadores será transformada novamente em formas de energia que interagem com o ser humano, a luz da lâmpada.

### 3.1. Componentes de comunicação

Ricquebourg em 2006 e outros autores sugerem que a camada de comunicação deve proporcionar duas necessidades: tornar possível a comunicação entre os diversos dispositivos e equipamentos, e conectar a casa ao mundo externo. Cook em 2003 entre outros autores adicionam mais uma necessidade: a conexão entre a casa e os usuários.

### 3.1.1. Wifi

A tecnologia *wifi* foi utilizada por possibilitar implementar comunicação e conexão sem fio entre computadores e outros dispositivos compatíveis, como celulares, impressoras e *tablets*, que estejam próximos geograficamente. Neste projeto ela é o meio de comunicação que possibilita a conexão e envio de mensagens entre o *tablet* iPad e computador, por não utilizar cabos permitirá o usuário a controlar as necessidade de qualquer ponto dentro dos limites de alcance da transmissão de radiofrequência.

“Wi-Fi é um conjunto de especificações para redes locais sem fio (WLAN - Wireless Local Area Network) baseada no padrão IEEE 802.11. O nome Wi-Fi é tido como uma abreviatura do termo inglês "Wireless Fidelity", embora a Wi-Fi Alliance, entidade responsável principalmente pelo licenciamento de produtos baseados na tecnologia, nunca tenha afirmado tal conclusão.” (<http://www.infowester.com/wifi.php>)

### 3.1.2. USB

No protótipo proposto a necessidade do uso do USB se faz pelo fato do computador precisar enviar mensagens ao *hardware* microcontrolador. Portanto é necessário que haja conexão que além de prover energia ao computador, também seja utilizada para a transmissão de informação.

Outra razão que leva a escolha da conexão USB é devido a sua simplicidade na configuração e manuseio. Por se tratar de uma conexão confiável devido a protocolos próprios, possuir compatibilidade com grande parte dos sistemas operacionais, por ser de baixo custo e visto que os computadores, atualmente, possuem várias portas USB, mostra que essa forma de conexão é a mais indicada para o projeto. Outro fator que evidencia o porquê de utilizar essa conexão é a questão da placa *Arduino UNO* possuir uma entrada e saída de dados via USB.

“USB (*Universal Serial Bus*) tem a particular função de permitir a conexão de muitos periféricos simultaneamente (pode-se conectar até 127 dispositivos em um barramento USB) ao barramento e este, por uma única tomada, se conecta à placa-mãe” (MONTEIRO, 2004).

### 3.1.3. UDP

O UDP (*user data protocol*) foi o protocolo utilizado para a transmissão dos pacotes entre o iPad e o computador, isso se dá pelo fato do aplicativo usado no iPad enviar os pacotes desta maneira.

Segundo a RFC 768, *User Datagram Protocol* (UDP) usa o protocolo IP (*internet protocol*) como subjacente, é utilizado como modo de transmissão de pacotes de comunicação por um computador ligado a uma rede de computadores. Este protocolo prevê um procedimento de envio de mensagens por programas para outros programas. O protocolo é orientado à transação, a entrega e uma dupla proteção não são garantidas.

### 3.1.4. OSC

O uso do protocolo OSC neste projeto se dá pelo fato do aplicativo utilizado para a troca de comunicação entre iPad e computador enviar as mensagens usando este formato. Os motivos pela escolha deste aplicativo serão discutidos no andamento do projeto.

Open Sound Control (OSC) é um protocolo para comunicação entre computadores, sintetizadores de som, e outros dispositivos multimídia. ([HTTP://opensoundcontrol.org](http://opensoundcontrol.org))

No projeto em questão serve para enviar mensagens entre a aplicação *TouchOSC* e o *Software Processing*. Uma aplicação é identificada pelo endereço IP e por uma “porta” (172.20.85.123:3000) As mensagens podem ser enviadas como inteiros, *floats* e *string*. É sempre iniciada por “/” por exemplo a mensagem contendo o nome botão pressionado e o seu valor deverá ser /nomedobotaovalor si mensagem com dois valores uma *string* e um inteiro.

Para o funcionamento no *Processing* é necessário o uso de uma biblioteca específica *oscP5*.

## 3.2. Componentes Físicos

### 3.2.1. Lâmpadas

A lâmpada incandescente ou lâmpada elétrica é um dispositivo elétrico que transforma energia elétrica em energia luminosa e energia térmica.

Thomas Alva Edison em 1880 construiu a primeira lâmpada incandescente utilizando uma haste de carvão muito fina que aquecendo até próximo ao ponto de fusão passa a emitir luz. No projeto em questão são usadas lâmpadas de 6,3V de menor tamanho, utilizadas na simulação de iluminação dos cômodos.

### 3.2.2. Relês

Relé é uma chave eletro-mecânica ou não, podendo estar aberta ou fechada dependendo do momento que um sinal é aplicado, a principal aplicação no projeto em questão é o acionamento das lâmpadas. No caso do Relé eletro-mecânico, a comutação é realizada alimentando-se a bobina do mesmo. Pode ser também usado para acionar lâmpadas, sirenes e outros. Dependendo da carga do relé (corrente máxima que seus contatos podem conduzir), pode-se até acionar motores.

### 3.2.3. Diodo

O diodo ligado reversamente deve estar presente sempre que se usa um relé (ou qualquer outro dispositivo que contenha uma bobina). Isso é necessário pois quando o relé muda de estado, gera uma descarga elétrica na parte do circuito que o alimentou, podendo danificar o transistor e impedir também que essa descarga passe para o circuito.

### 3.2.4. Transistor

O consumo em média de um relé é de aproximadamente 40mA. Por isso, não se pode ligá-lo diretamente na saída do *Arduino*, pois o relé consome mais corrente do que a saída do *Arduino* pode fornecer. Para resolver esse problema, usamos um transistor NPN, que será acionado pelo pino do *Arduino*, e este sim, ligará o relé, conforme esquemático da figura 3.1.

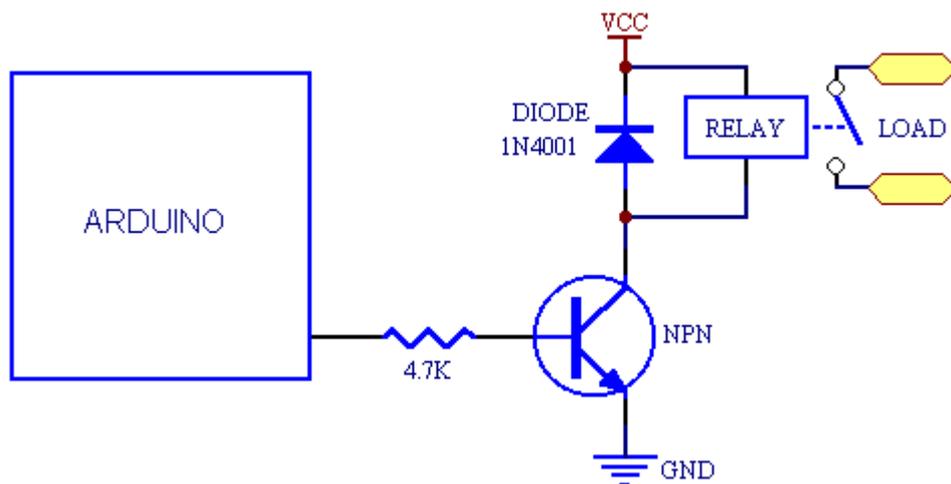


Figura 3.1 – Esquemático ligando relé e transistor ao *Arduino*

Fonte: <http://www.Arduinors.net/blog/batendo-reles-pelo-Arduino/> acessado em 01 de junho de 11

### 3.2.5. Microcontrolador

A utilidade do microcontrolador neste protótipo, consiste em interpretar as funções recebidas via *Processing* e acionar o atuador correto permitindo ligar ou desligar as lâmpadas da maquete.

“Os microcontroladores são dispositivos semicondutores e possuem uma vasta utilização no mercado, seja na área de automação residencial (microondas), automação industrial ou embarcada (robôs da linha de montagem automotiva ou computadores de bordo) e até automação predial (elevadores e alarmes). Por ser limitado, geralmente não é empregado

caso seja necessário utilizar uma quantidade de memória acima de sua capacidade” (GIMENEZ, 2004).

### 3.2.5.1. Microcontrolador Atmega328P

“O ATmega328P é um microcontrolador 8 bits de baixa potencia que foi desenvolvido pela empresa ATMEL. Esse microcontrolador de arquitetura RISC que trabalha a uma frequência de 16MHz. Este microcontrolador possui as memórias Flash, EEPROM e RAM. O ATmega328P possui 32K Bytes de memória Flash, 1k Byte de memória EEPROM e 2k bytes de RAM. Ele também e composto por catorze portas digitais e seis portas analógicas” (ATMEL, 2011).

### 3.2.5.2. *Arduino*

*Arduino*, segundo o site oficial, é uma plataforma eletrônica de prototipagem, de código aberto, baseada em *hardware* e *software* flexíveis e fáceis de usar. Seu propósito é atender a artistas, designers, *hobbistas* ou qualquer um que esteja interessado em criar objetos ou ambientes interativos. O módulo, pois, que será utilizado para controle e também como atuador mais complexo será a plataforma *Arduino* exibido na figura 3.2.

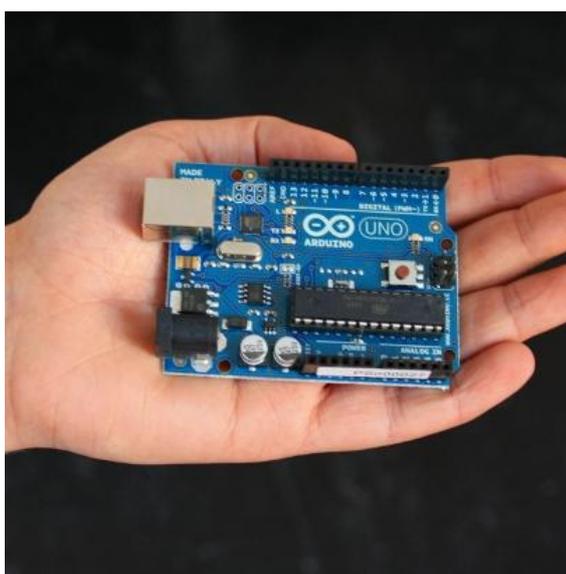


Figura 3.2 – Placa *Arduino UNO*

Fonte: <http://www.Arduino.cc/> acessado em 20 de março de 2011

A razão por ter sido escolhida a placa *Arduino UNO*, foi devido ao seu tamanho reduzido e ampla quantidade de informações disponíveis nos fóruns desenvolvedores. A plataforma de programação da placa chamada, *Arduino Environment*, oferece uma IDE para o desenvolvimento de códigos. Para que ocorra a comunicação com o computador, a placa oferece uma porta USB, na qual, é utilizada tanto para provimento de energia quanto para transmissão de informação. Esse processo ocorre de forma paralela.

### 3.2.6. iPad

Estudos feitos pela empresa Morgan Stanley mostram que o iPad pode se tornar o dispositivo móvel mais popular da história. Hoje cada vez mais as pessoas buscam mobilidade, e os *tablets* passaram a ser uma alternativa mais eficiente para as pessoas que costumavam usar *netbooks*. Por esses e outros motivos a escolha do *gadget* da Apple foi apontada para o desenvolvimento do projeto do autor

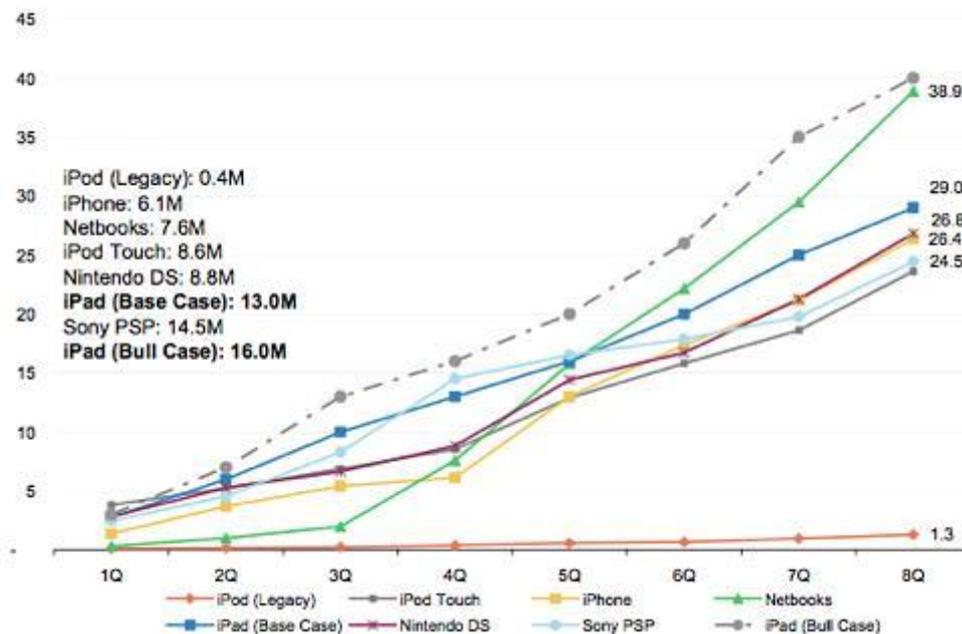


Figura 3.3 – Crescimento do iPad no Mercado

Fonte: Morgan Stanley

### 3.3. Componentes de *Software*

#### 3.3.1. *Software Arduino Development Enviroment*

Para o desenvolvimento, *upload* e debug do código de programação no *hardware* do microcontrolador *Arduino* foi necessário o uso do *Software* livre de desenvolvimento da plataforma. Nesse ambiente é necessário usar a linguagem de programação *Wiring* que é derivada da linguagem C e C++. Algumas bibliotecas específicas já fazem parte do aplicativo e por se tratar de *Software* livre, existe a possibilidade de adicionar ou criar outras, dependendo do que será criado ou modificado. O *Arduino* IDE possui a capacidade de reconhecimento de todas as estruturas da linguagem C e também de alguns recursos da linguagem C++, com o código criado e compilado é possível realizar o “*upload*” para a placa *Arduino*, ou seja, caso o código obedeça às regras da linguagem e nenhum erro seja identificado, é gerado o código binário equivalente ao que foi criado e então é gravado no microcontrolador, para posterior utilização.

#### 3.3.2. *Software Processing*

O *Processing* (*PROCESSING*, 2008; REAS & FRY, 2007) é uma linguagem de programação também licenciada como open-source e voltada para manipulação gráfica. Também foi desenvolvida no contextos das artes e do design e viabiliza que pessoas que tenham pouca familiaridade com linguagens de programação, e mestrias em outras áreas do conhecimento, possam obter resultados rapidamente. Também, para fomentar a interação entre *Processing* e *Arduino* e ampliar as possibilidades de uso da tecnologia há tutoriais simples e de fácil leitura (*ARDUINO*, 2008).

As plataformas *Arduino* e *Processing* permitem, em conjunto, ou de modo isolado, o desenvolvimento de tecnologias apropriadas para interfacear, criar e engenhar experimentos didáticos em laboratórios. Originárias no contexto das artes, seu emprego é de certo modo facilitado pelas preocupações que seus desenvolvedores iniciais tiveram em viabilizar a construção de protótipos em um ambiente gráfico interessante.

### 3.3.3. Software TouchOSC interface editor

O Software *TouchOSC editor* foi utilizado para a personalização da interface gráfica. O *download* pode ser efetuado através do site do desenvolvedor. A figura 3.4 representa o ambiente de desenvolvimento do programa.

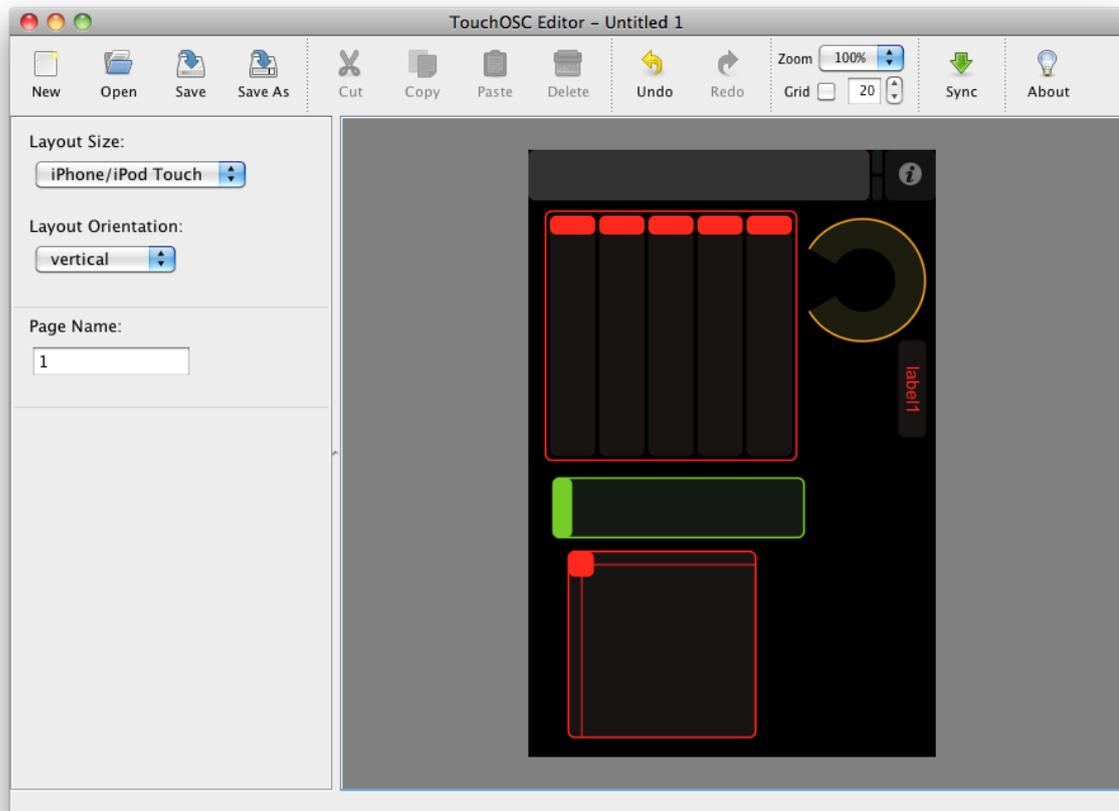


Figura 3.4 – Ambiente de personalização *TouchOSC*

Fonte: Autor

### 3.3.4. AutoCAD

AutoCAD é um *Software* do tipo CAD — *computer aided design* ou desenho auxiliado por computador criado e comercializado pela Autodesk, Inc. desde 1982. É utilizado principalmente para a elaboração de peças de desenho técnico em duas dimensões (2D) e para criação de modelos tridimensionais (3D). No desenvolvimento do projeto foi utilizado para o

desenho da planta baixa da maquete residencial. É amplamente utilizado em arquitetura, design de interiores, engenharia mecânica, engenharia geográfica, engenharia elétrica e em vários outros ramos da indústria. O AutoCAD é atualmente disponibilizado em versões para o sistema operacional *Microsoft Windows* e *Mac OS*, embora já tenham sido comercializadas versões para *UNIX*.

### 3.3.5. iOS

iOS é o sistema operacional para *devices* móveis da *Apple*, foi desenvolvido originalmente para o iPhone, no entanto hoje também é usado em iPod Touch, iPad e *Apple TV*. No projeto em questão é o sistema operacional que rodará no iPad. A *Apple* não permite o sistema operacional rodar em *hardware* de terceiros. A interface do usuário do iOS é baseado no conceito de manipulação direta, utilizando gestos em multi-toque. A interação com o sistema operacional inclui gestos como apenas tocar na tela, deslizar o dedo, e o movimento de "pinça" utilizado para se ampliar ou reduzir a imagem. Conta com acelerômetros internos que são usados por alguns aplicativos para responder à agitação do aparelho (um resultado comum é o comando de desfazer) ou rodá-la em três dimensões (um resultado comum é a mudança do modo retrato para modo paisagem).

A interface do usuário do iOS é baseado no conceito de manipulação direta, utilizando gestos *multi-touch*. Elementos de controle de Interface consistem de barras, chaves e botões. A resposta à entrada do usuário é imediato e oferece uma interface de fluidos. A interação com o sistema operacional inclui gestos, tais como tocar e apertar, que tem definições específicas dentro do contexto do sistema operacional iOS e sua interface multitoque.

### 3.3.6. Aplicativo *TouchOSC*

O aplicativo permite que controle remotamente e receba informações de outros *Softwares* ou *hardwares* que implementam protocolos OSC ou MIDI, como *Apple Logic Pro/Express*, *Renoise*, *Pure Data*, *Max/MSP/Jitter*, *Max for Live*, *OSculator*, *VDMX*, *Resolume Avenue 3*, *Modul8*, *Plogue Bidule*, *Reaktor*, *Quartz Composer*, *Vixid VJX16-4*,

Supercollider, FAW Circle, vvvv, Derivative TouchDesigner, Isadora e muitos outros. Sua interface gráfica pode ser visualizada na figura 3.5



Figura 3.5 – Interface *TouchOSC*

Fonte: [HTTP://hexler.net/Software/TouchOSC](http://hexler.net/Software/TouchOSC)

O aplicativo vem com alguns *layouts* por padrão no entanto para o projeto em questão foi utilizado o *Software free TouchOSC Editor* para a personalização da interface gráfica.

## CAPÍTULO 4 – MODELO PROPOSTO

### 4.1. Apresentação Geral do Modelo Proposto

A proposta da arquitetura do protótipo mostrada na figura 4.1 é direcionada para o cenário de automação de iluminação, porém podendo alcançar vários outros objetivos. A camada de comunicação e interação com o usuário será utilizando um *tablet* iPad o qual definirá todo o controle de iluminação, este é responsável por enviar as informações fornecidas por seu usuário via *Wi-Fi* a um computador que servirá de camada de comunicação entre o *tablet* e o *hardware* microcontrolador.



Figura 4.1 – Esquemático

Fonte: Autor

O protótipo proposto é formado por diversos aplicativos e diferenciados *hardwares* e componentes, entre eles:

- a) Microcontrolador *Arduino UNO*
- b) *Notebook*
- c) iPad
- d) Relês
- e) Lâmpadas
- f) Conexões
- g) Cabo USB
- h) Maquete residencial
- i) *Software Arduino*
- j) *Software Processing*
- k) *Software TouchOSC interface editor*
- l) Aplicativo *TouchOSC*

No *notebook* será executado o aplicativo *Processing* que deverá estar em modo *run* durante todo funcionamento do protótipo. Este será o meio de comunicação entre o iPad e o *hardware* microcontrolador. O *notebook* ainda será utilizado para a compilação e embarcação do código dentro do microcontrolador usando o *Software Arduino Development Enviroment* e a personalização da interface gráfica que será embarcada no iPad. O aplicativo *TouchOSC* é o responsável por alterar o estado dos objetos que forem conectados ao microcontrolador, esta alteração será feita via *Wi-Fi* enviando pacotes UDP do iPad para novamente ao *notebook*.

## 4.2. Camadas e seu funcionamento

A figura 4.2 representa o fluxo das camadas e funcionamento do protótipo proposto.

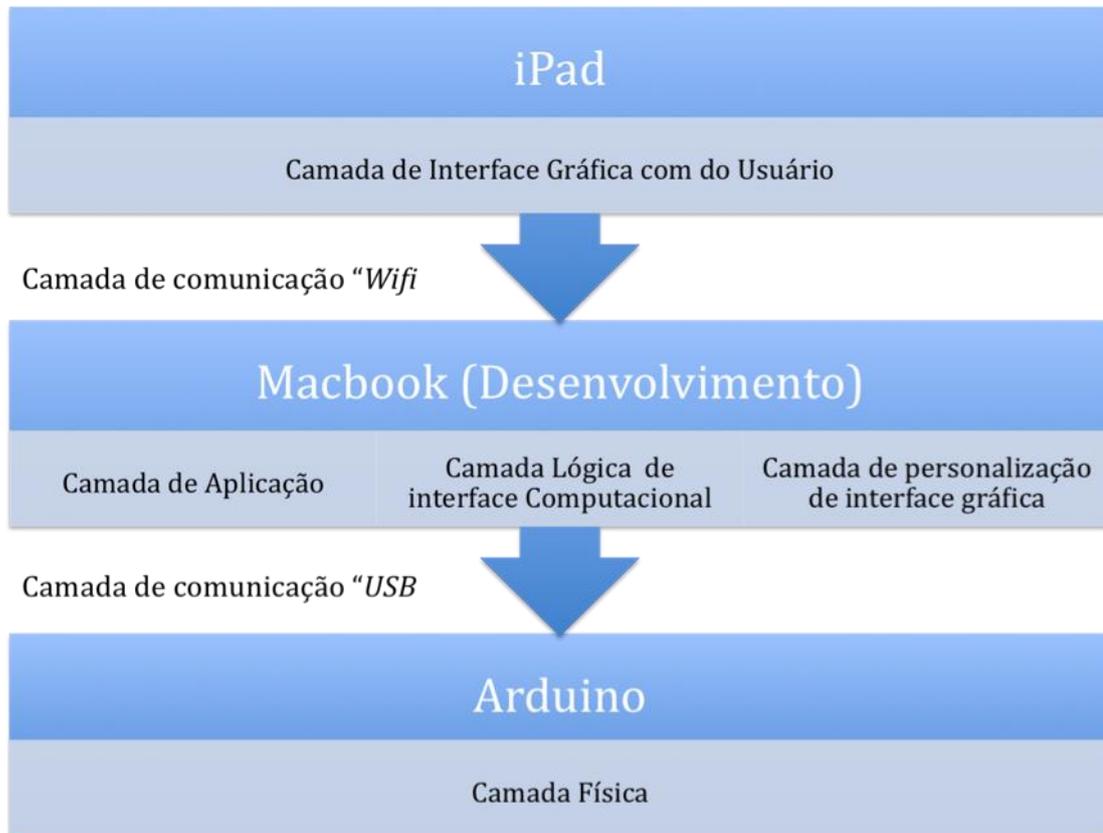


Figura 4.2 – Fluxo das Camadas

Fonte: Autor

### 4.2.1. Camada Física

A camada física agrega os seguintes dispositivos físicos: notebook, cabo USB, *hardware* microcontrolador, relés, transistores, diodos e lâmpadas. Nessa camada ocorre a conversão de fenômenos físico-químicos em sinais elétricos.

A conexão entre o *hardware* microcontrolador e o computador é feita via USB.

O microcontrolador *Arduino* está ligado fisicamente aos relés e por sua vez as lâmpadas que deverão ser acionadas.

#### 4.2.2. Camada de Comunicação

Essa camada é responsável em prover um mecanismo que, adequadamente, formate, roteie e transporte informações entre o usuário e as lâmpadas. O meio de propagação escolhido para a comunicação entre o iPad, que servirá de interface gráfica para o usuário, e o computador, será *Wi-Fi*, utilizando pacotes UDP contendo mensagens OSC.

A comunicação entre o computador e o *hardware Arduino* se dará através de conexão USB, no entanto a interpretação dos pacotes recebidos via *Wi-Fi*, ficará por conta do *Software Processing* (camada lógica de interface computacional).

O *hardware* microcontrolador enviará sinais elétricos para a ativação dos relés e conseqüentemente as lâmpadas.

#### 4.2.3. Camada lógica de Interface Computacional (Programação *Processing*)

A interpretação dos dados enviados pelo aplicativo *TouchOSC* (camada de interface gráfica do usuário) se dá através do *Software Processing*, este aplicativo será responsável por intermediar as mensagens enviadas pelo iPad e redirecionar ao *Arduino*. O código desenvolvido para o *Processing* irá interpretar os pacotes OSC enviados através da rede *Wi-Fi* por protocolo UDP, traduzir para comandos específicos que podem ser entendidos pelo microcontrolador *Arduino* e envia-los via USB para o mesmo para que possam ser executados de forma correta.(Camada de aplicação).

Toda a programação desta camada deve estar em total sintonia com as camada de interface gráfica do usuário.

#### 4.2.4. Camada de Aplicação (Programação *Arduino*)

O objetivo dessa camada é mostrar a ferramenta que foi usada para a programação do *hardware* microcontrolador *Arduino*. A programação do código do microcontrolador deve estar em total sintonia com as informações recebidas pela camada lógica de interface computacional. Esta camada deve prover o funcionamento da automação através do ciclo de

desenvolvimento, incluindo as fases de programação, simulação, instalação, teste, depuração e implementação. Toda a informação proveniente das outras camadas faz com que a programação tenha que se tornar interativa e inteligente o suficiente para que o algoritmo entenda as informações repassadas pelas outras camadas e interfira no comportamento da casa.

#### 4.2.5. Camada de personalização de Interface com o Usuário

O objetivo da camada de interface com usuário é possibilitar a personalização e adequação da interface gráfica que será provida ao usuário. Esse processo é efetuado através do *Software TouchOSC editor*, nesta etapa deverá ser criada e programada a interface gráfica, com a quantidade necessária de botões e com a nomeação correta de cada um para a integração com as outras camadas. Cada cômodo da casa automatizada é representado por um botão que poderá ser controlado através da interface gráfica do usuário.

#### 4.2.6. Camada de Interface Gráfica do Usuário

A interface gráfica do usuário roda em utilizando um *Software* já existente o *TouchOSC*, desenvolvido em iOS. Sua conexão com o computador se estabelece por meio de *Wi-Fi* e, dessa forma, permite ao usuário controlar sua residência remotamente. Esse desacoplamento é importante não só pela comodidade do gerenciamento remoto, mas também por motivos de acessibilidade permitindo adequação com o perfil do usuário que irá utilizá-la.

A interface disponibiliza um menu para o envio de comandos ao sistema de controle. Basta tocar no botão desejado e a interface enviará a mensagem OSC específica via UDP para o endereço IP definido como destino e irá alterar o estado do atuador.

### 4.3. Descrição do funcionamento

#### 4.3.1. Construção da maquete

Para a realização do protótipo foi necessária a construção de um maquete residencial a fim de representar uma casa e seus respectivos cômodos que terão suas iluminações automatizadas. O primeiro passo para essa representação foi o desenvolvimento de uma planta baixa utilizando *Software* AutoCAD, a planta é composta por dois quartos, dois banheiros, sala, cozinha e uma garagem. A Figura 4.3 ilustra quais são os cômodos automatizados e como serão fisicamente distribuídos na maquete.

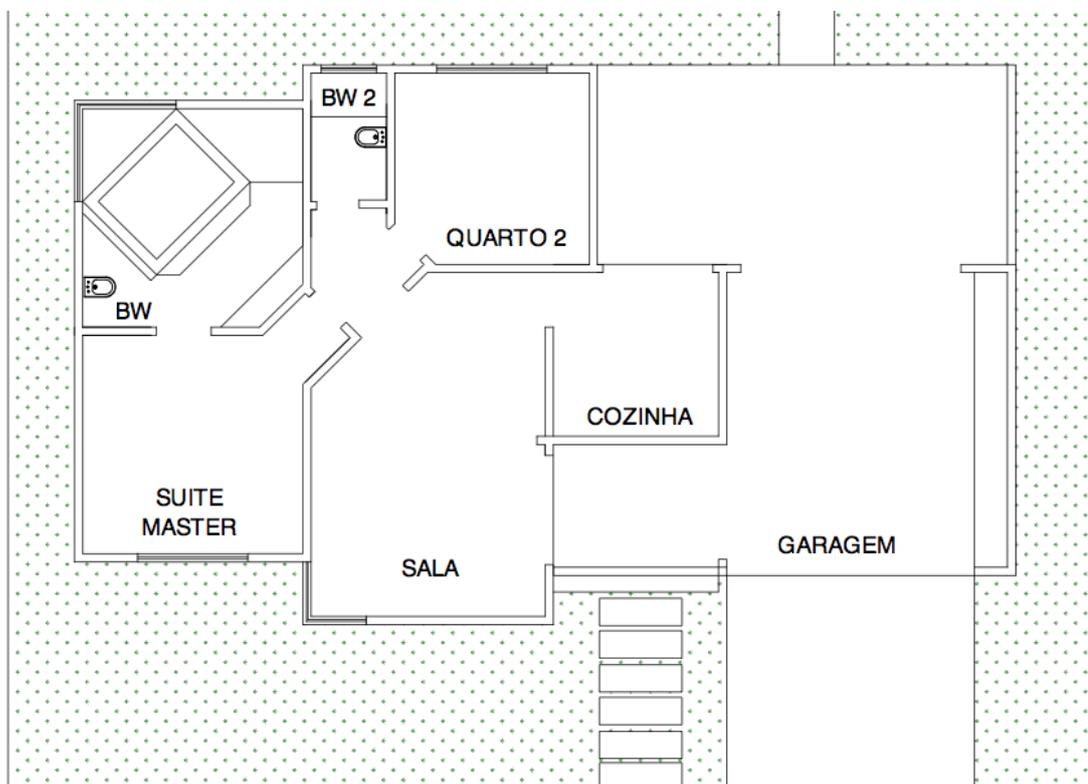


Figura 4.3 – Planta baixa

Fonte: Autor

Para a construção da maquete foi utilizado papel pluma. A maquete possui fabricação própria do autor e está em escala 1/33. Em cada cômodo da residência são implementadas funcionalidades independentes para controle de iluminação que acrescentam “inteligência” no cenário automatizado.

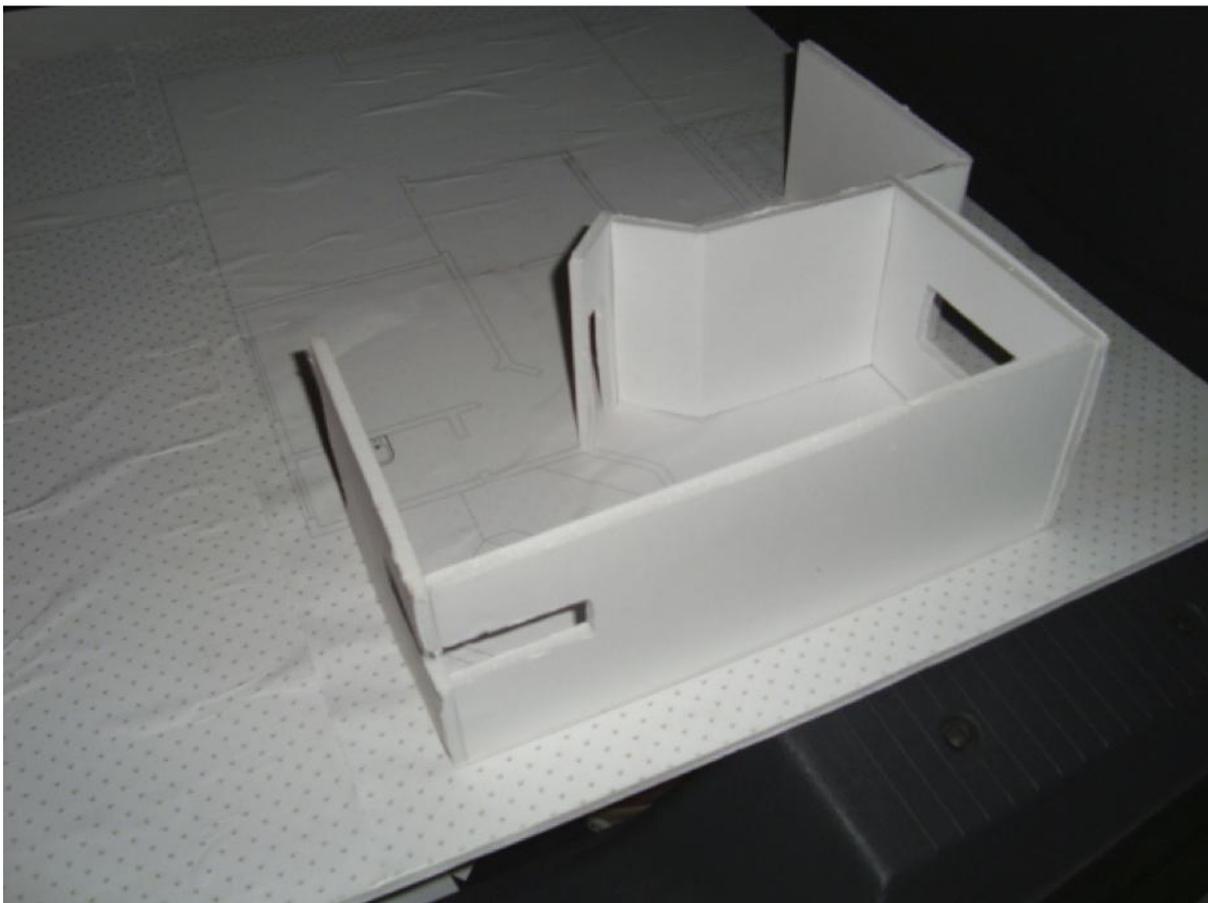


Figura 4.4 – Foto maquete sendo construída

Fonte: Autor

#### 4.3.2. Ligações físicas

Após a conclusão da maquete iniciaram os trabalhos de prototipagem e ligação física das lâmpadas nos respectivos cômodos.

Cada lâmpada deve ser ligado a um circuito como o das figuras 4.5 e 4.6 e a um determinado pino no *Arduino*.

O consumo médio de um relé é aproximadamente 40mA. Por esse motivo não se pode ligá-lo diretamente na saída do *Arduino*, pois o relé consome mais corrente do que a saída do *Arduino* pode fornecer. Para contornar essa deficiência foi usado um transistor NPN, que será acionado pelo pino do *Arduino*, e este sim, aciona o relé.

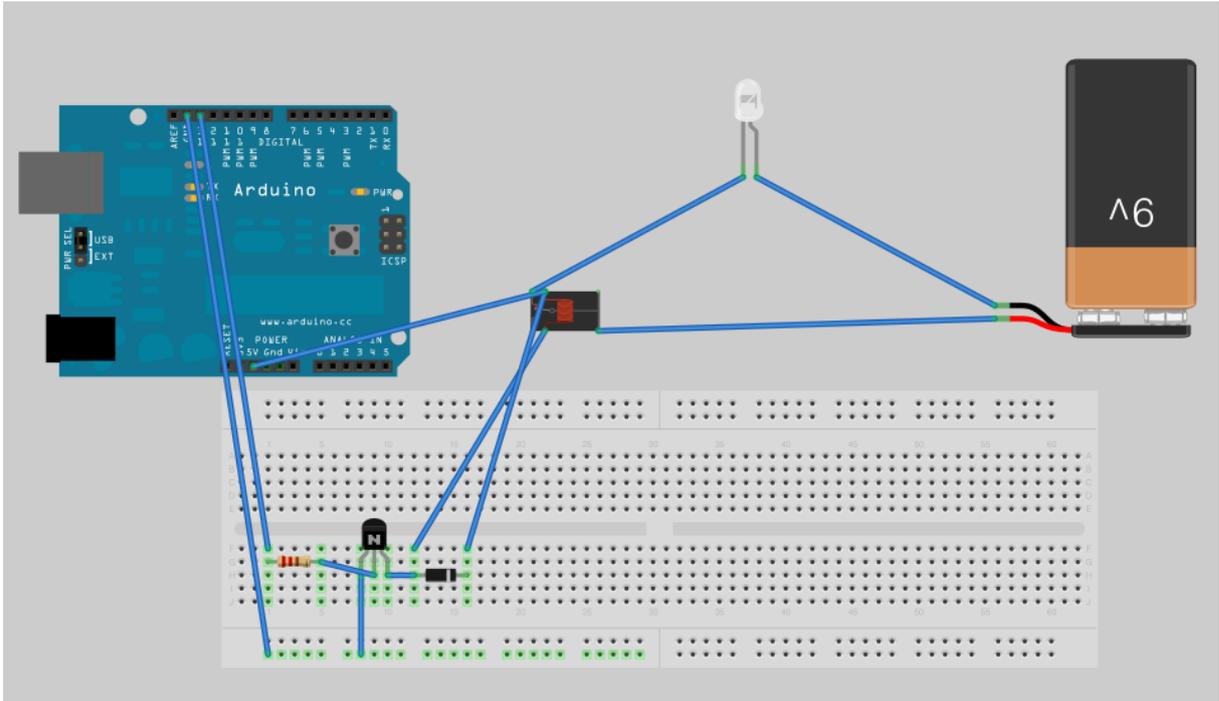


Figura 4.5 – Ligações Físicas

Fonte: Autor

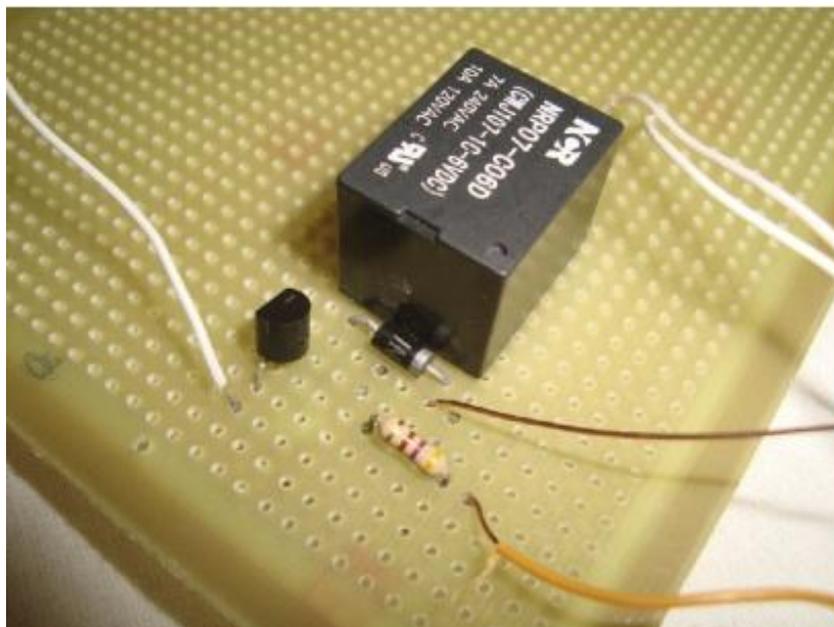


Figura 4.6 – Ligações Físicas

Fonte: Autor

Cada cômodo tem seu controle independente através de um determinado pino no *Arduino* conforme quadro 1:

Quadro 1 – Cômodos x Pinos

CÔMODO	PINO
Sala	12
Cozinha	11
Suite Master	10
Banheiro Suite	9
Quarto 2	8

Fonte: Autor

Sempre que se usa um relé (ou qualquer outro dispositivo que contenha uma bobina), é necessário colocar um diodo invertido em paralelo com a bobina. Isso é necessário pois quando o relé é desligado, gera uma descarga elétrica na parte do circuito que o alimentou. O diodo nesse caso, serve para impedir que essa descarga passe para o circuito.

#### 4.3.3. Ligando o *Arduino*

No desenvolvimento do protótipo foi utilizado como um dos ambientes de desenvolvimento o aplicativo *Arduino* conforme figura 4.7, necessário para a escrita do código e *upload* dentro do *hardware*. No caso do autor foi utilizada a versão para Mac OS X, no entanto versões para outros sistemas operacionais podem ser encontrados também através do site [HTTP://Arduino.cc/en/Main/Software](http://Arduino.cc/en/Main/Software).

A linguagem utilizada para o desenvolvimento do protótipo foi a linguagem própria do *Arduino*, que tem uma certa semelhança com o C e o C++.



Figura 4.7 – Ambiente de desenvolvimento *Arduino*

Fonte: Autor

Deve-se então iniciar o ambiente do *Arduino*, em seguida podemos conectar o microcontrolador a porta USB do seu computador, por padrão o Mac OS reconhece o *Arduino* como uma nova interface de rede, caso ocorra tal etapa poderá ser cancelada, devendo ser configurada usando o próprio ambiente de desenvolvimento do *Arduino*, conforme Figura 4.8.

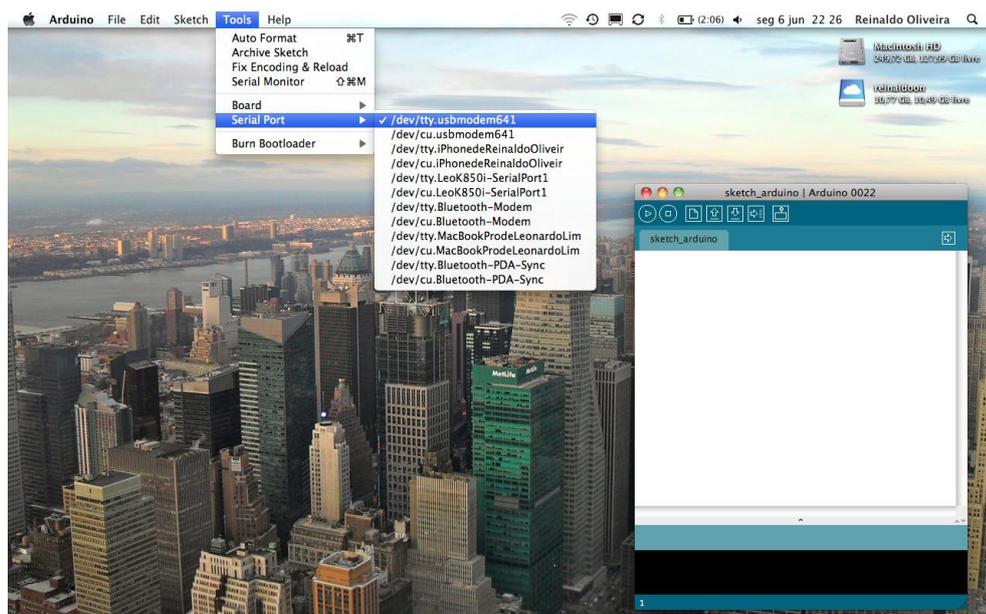


Figura 4.8 – Configuração da interface serial

Fonte: Autor

Também é necessário certificar-se que o ambiente de desenvolvimento está preparado para o *hardware* escolhido para iniciar o projeto. No caso do autor o modelo escolhido foi o *Arduino UNO*. Esta verificação pode ser resolvida conforme Figura 4.9.

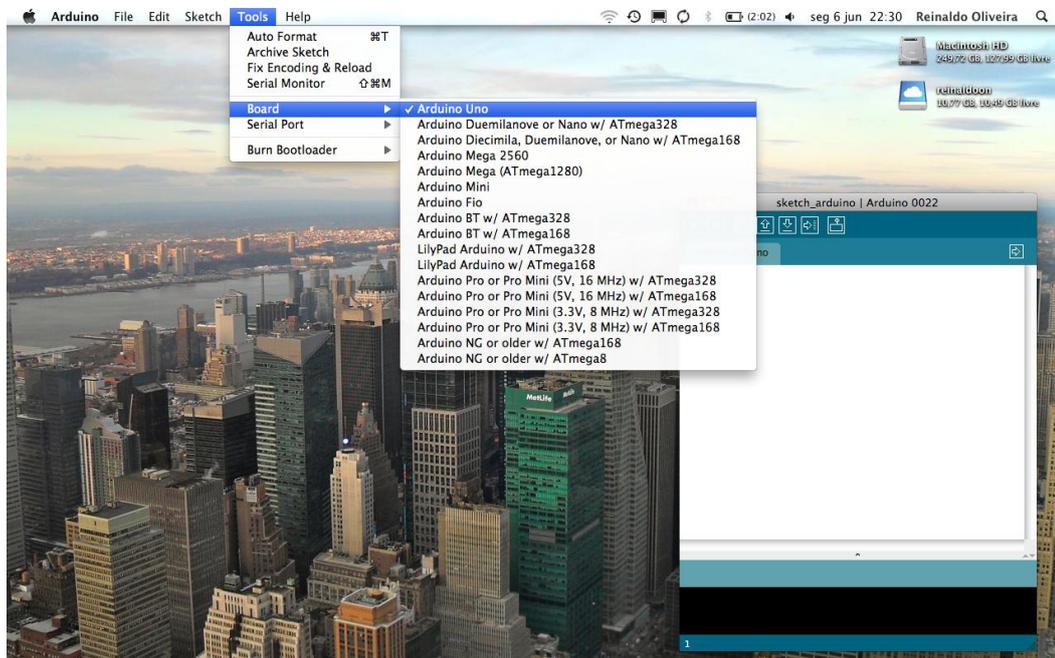


Figura 4.9 – Configuração do modelo de placa escolhido

Fonte: Autor

O desenvolvimento do código é iniciado após estes procedimentos. Conforme o quadro 2 aproveitando as informações fornecidas no quadro 1, página 43, definiremos qual cômodo cada pino representa e qual seu respectivo comando para ativar ou desativar os atuadores.

Quadro 2 – Tabela com referencias de comando

CÔMODO	PINO	COMANDO LIGA	COMANDO DESLIGA
Sala	12	S	s
Cozinha	11	C	c
Suíte Master	10	M	m
Banheiro Suite	9	B	b
Quarto 2	8	Q	q

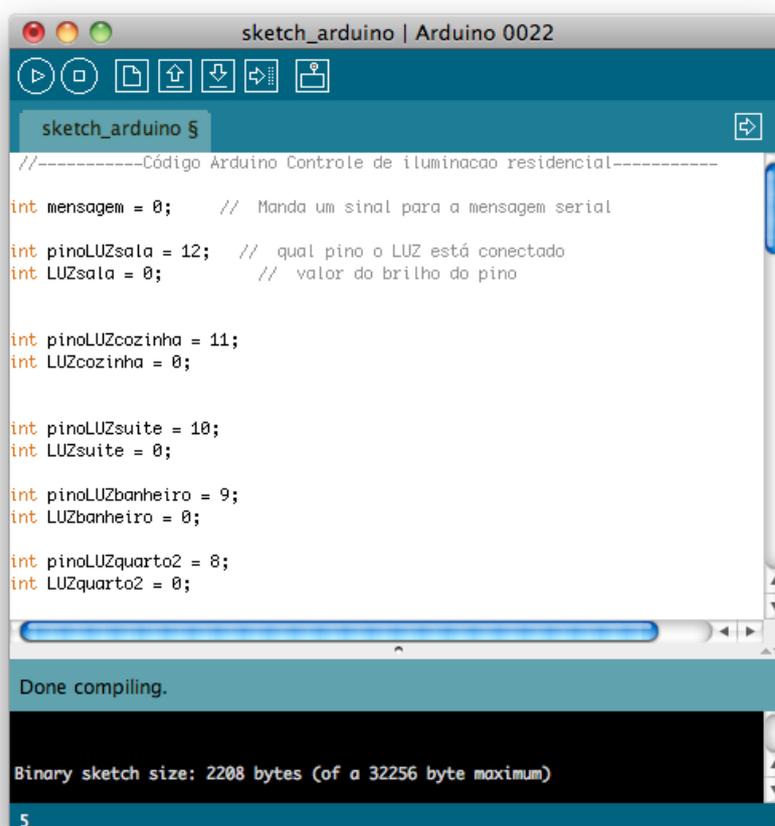
Fonte: Autor

O desenvolvimento do código deve se basear nestes comandos para que seja viável a integração com os outros códigos necessários, para a implementação do protótipo.

#### 4.3.3.1. O código *Arduino*

A primeira etapa de desenvolvimento de programação foi utilizando o ambiente de desenvolvimento do *Arduino*, O código está integralmente apresentado no Apêndice A deste documento.

A figura 4.10 ilustra o trecho do código do *Arduino* que onde se define qual pino representará cada cômodo, além disso é zerado o valor do brilho de cada lâmpada o qual poderá variar de “0” à “255” sendo que “0” representa desligado e “255” representa ligado, ou ainda LOW para desligado e HIGH para ligado.



```
sketch_arduino | Arduino 0022
sketch_arduino 5
//-----Código Arduino Controle de iluminacao residencial-----
int mensagem = 0; // Manda um sinal para a mensagem serial
int pinoLUZsala = 12; // qual pino o LUZ está conectado
int LUZsala = 0; // valor do brilho do pino

int pinoLUZcozinha = 11;
int LUZcozinha = 0;

int pinoLUZsuite = 10;
int LUZsuite = 0;

int pinoLUZbanheiro = 9;
int LUZbanheiro = 0;

int pinoLUZquarto2 = 8;
int LUZquarto2 = 0;

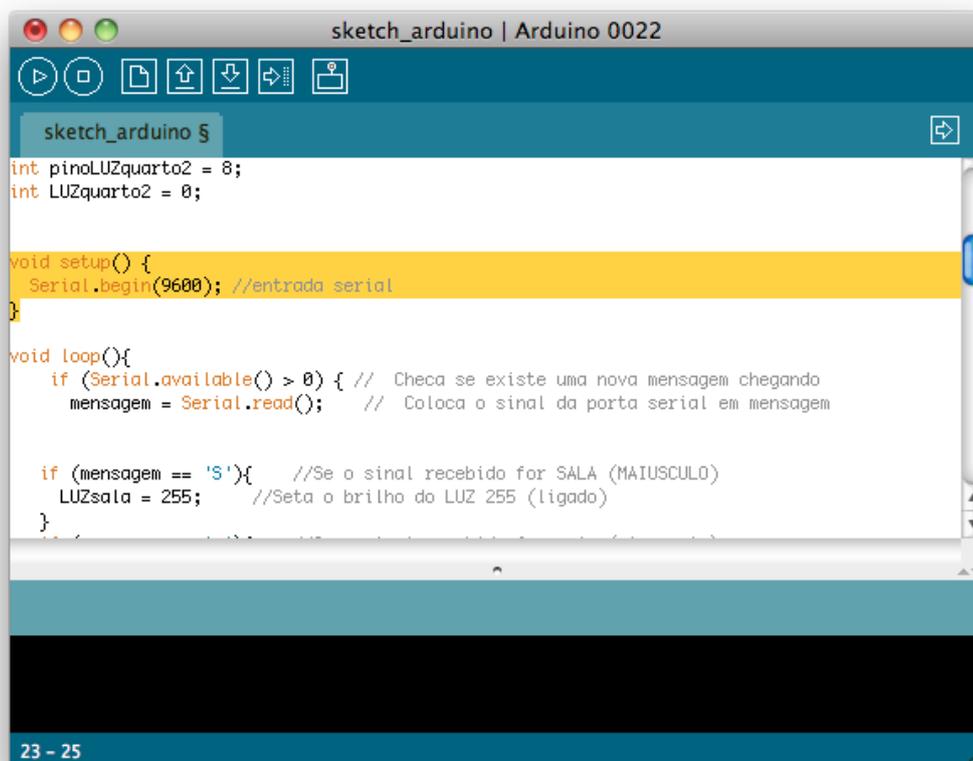
Done compiling.

Binary sketch size: 2208 bytes (of a 32256 byte maximum)
5
```

Figura 4.10 – Definindo os pinos no código

Fonte: Autor

Na seqüência do código conforme figura 4.11 se configura qual será a taxa de transmissão de para o recebimento dos dados na porta serial. Na situação do projeto foi escolhida o valor escolhido foi “9600”.



```
sketch_arduino | Arduino 0022
sketch_arduino 5
int pinoLUZquarto2 = 8;
int LUZquarto2 = 0;

void setup() {
  Serial.begin(9600); //entrada serial
}

void loop(){
  if (Serial.available() > 0) { // Checa se existe uma nova mensagem chegando
    mensagem = Serial.read(); // Coloca o sinal da porta serial em mensagem

    if (mensagem == 'S'){ //Se o sinal recebido for SALA (MAIUSCULO)
      LUZsala = 255; //Seta o brilho do LUZ 255 (ligado)
    }
  }
}
```

23 - 25

Figura 4.11 – Configuração da porta de entrada

Fonte: Autor

A figura 4.12 ilustra o trecho do código que lê as informações recebidas pela porta serial com taxa de transmissão “9600”. Constantemente o código executa a operação de checagem se existe novas mensagens chegando na porta e escreve o dado obtido em uma constante chamada “mensagem”.



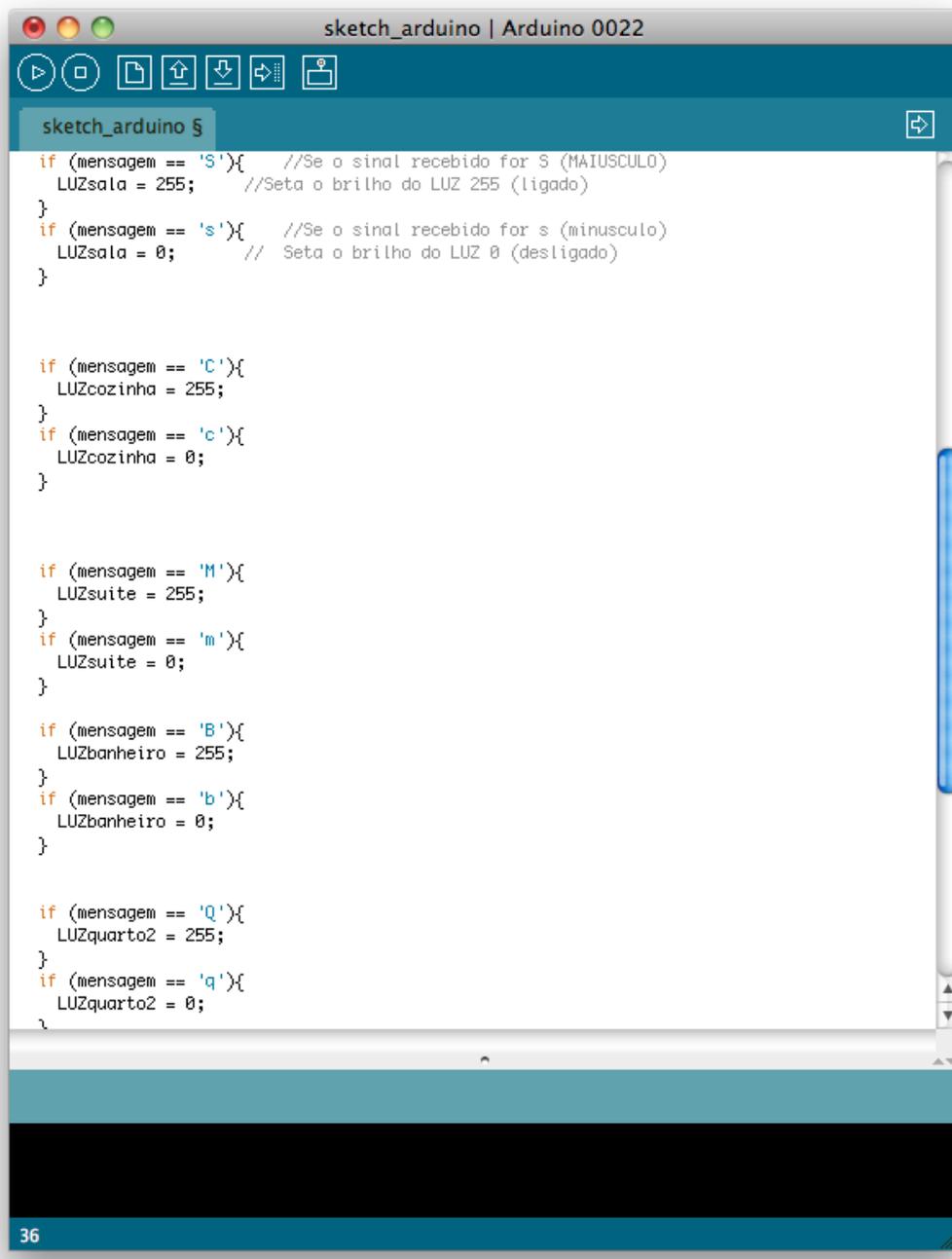
```
sketch_arduino | Arduino 0022  
sketch_arduino 5  
void setup() {  
  Serial.begin(9600); //entrada serial  
}  
void loop(){  
  if (Serial.available() > 0) { // Checa se existe uma nova mensagem chegando  
    mensagem = Serial.read(); // Coloca o sinal da porta serial em mensagem  
  }  
  if (mensagem == 'S'){ //Se o sinal recebido for SALA (MAIUSCULO)  
    LUZsala = 255; //Seta o brilho do LUZ 255 (ligado)  
  }  
  if (mensagem == 's'){ //Se o sinal recebido for sala (minuscuro)  
    LUZsala = 0; // Seta o brilho do LUZ 0 (desligado)  
  }  
}
```

27 - 29

Figura 4.12 – Leitura da porta com taxa de transmissão “9600”

Fonte: Autor

Aproveitando as informações fornecidas no quadro 1 na página 43, o código com os comandos referentes a ligar ou desligar cada lâmpada de um respectivo cômodo, é embarcado conforme figura 4.13.



```
sketch_arduino | Arduino 0022

sketch_arduino §

if (mensagem == 'S'){ //Se o sinal recebido for S (MAIUSCULO)
  LUZsala = 255; //Seta o brilho do LUZ 255 (ligado)
}
if (mensagem == 's'){ //Se o sinal recebido for s (minuscuro)
  LUZsala = 0; // Seta o brilho do LUZ 0 (desligado)
}

if (mensagem == 'C'){
  LUZcozinha = 255;
}
if (mensagem == 'c'){
  LUZcozinha = 0;
}

if (mensagem == 'M'){
  LUZsuite = 255;
}
if (mensagem == 'm'){
  LUZsuite = 0;
}

if (mensagem == 'B'){
  LUZbanheiro = 255;
}
if (mensagem == 'b'){
  LUZbanheiro = 0;
}

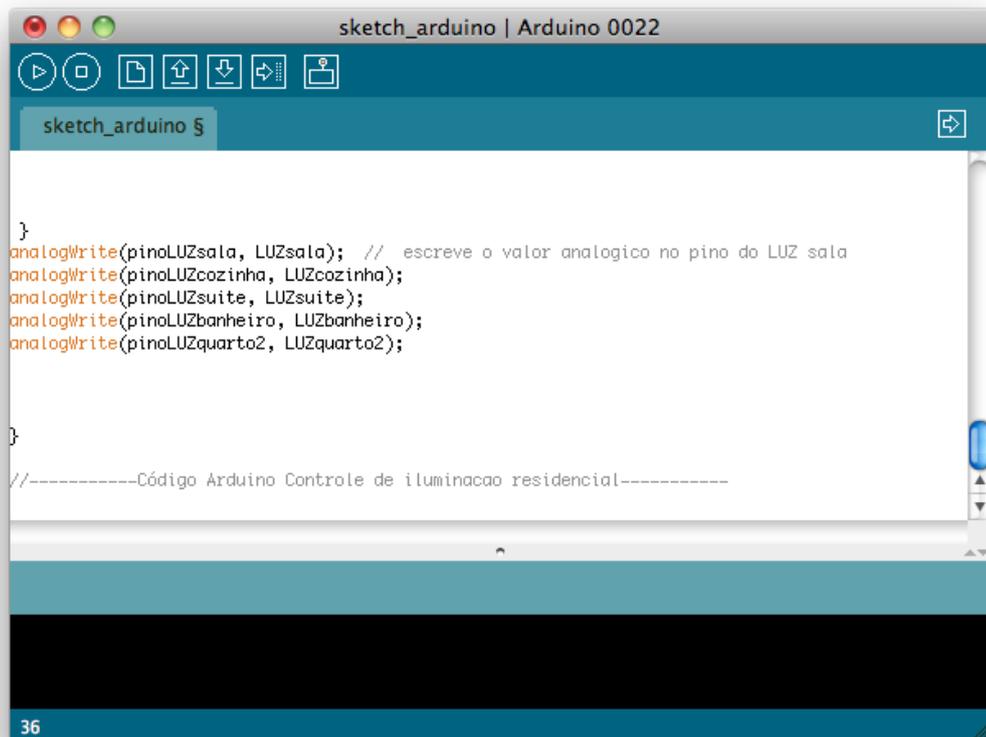
if (mensagem == 'Q'){
  LUZquarto2 = 255;
}
if (mensagem == 'q'){
  LUZquarto2 = 0;
}
}

36
```

Figura 4.13 – Comandos para ligar ou desligar lâmpadas

Fonte: Autor

Por fim o código deverá escrever o comando requisitado “0” ou “LOW” ou “255” ou “HIGH” no pino correto proporcionando acionar o atuador correto, acendendo ou apagando as lâmpadas respectivas, conforme figura 4.14.



```
sketch_arduino 5  
}  
analogWrite(pinoLUZsala, LUZsala); // escreve o valor analogico no pino do LUZ sala  
analogWrite(pinoLUZcozinha, LUZcozinha);  
analogWrite(pinoLUZsuite, LUZsuite);  
analogWrite(pinoLUZbanheiro, LUZbanheiro);  
analogWrite(pinoLUZquarto2, LUZquarto2);  
}  
  
//-----Código Arduino Controle de iluminacao residencial-----  
  
36
```

Figura 4.14 – Escrita do comando na no pino respectivo

Fonte: Autor

A verificação do código pode ser executada pressionando o primeiro botão à direita disponível dentro do ambiente de desenvolvimento, chamado *verify*. Caso o código esteja correto deve ser executado o *upload* para o *hardware*, isso é possível pressionando o penúltimo botão à esquerda.

A partir deste momento já pode ser testada a funcionalidade da automação através do envio de comandos usando a função serial monitor proveniente no ambiente de desenvolvimento do *Arduino* conforme ilustração 4.15.

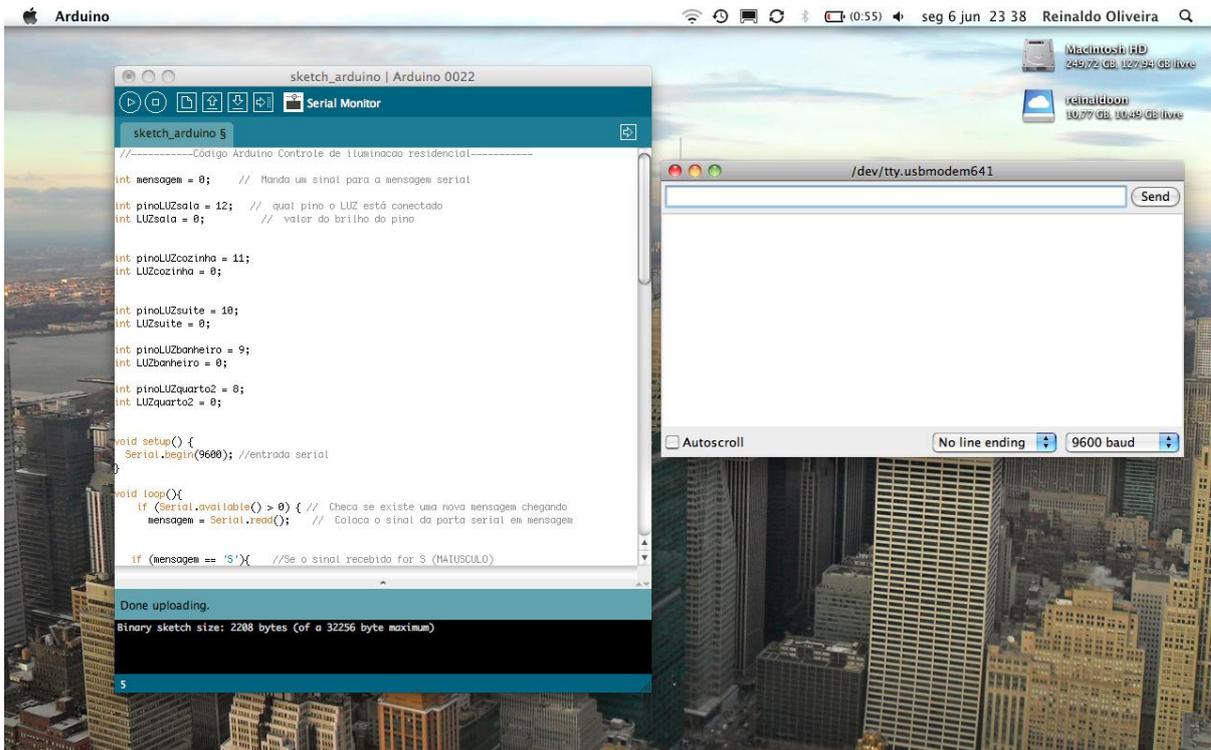


Figura 4.15 – Envio de comandos via Serial Monitor

Fonte: Autor

#### 4.3.4. Interface gráfica

A interface gráfica foi criada totalmente em conjunto com as camadas de aplicação e de interface lógica computacional, no que diz respeito à dependência de linguagem de programação e compatibilidade. Ou seja, os parâmetro usados aqui são pré-requisitos para a funcionalidade do sistema todo. No entanto, existe extrema facilidade de adicionar novas funcionalidades.. Estas etapas mostrarão como a interface irá paracer no iPad.

O aplicativo utilizado para o desenvolvimento da interface gráfica foi o *TouchOSC editor*. Possui interface amigável e de fácil configuração. Como o desenvolvimento do protótipo é utilizando o *tablet* iPad foi usado a interface conforme a Figura 4.16.

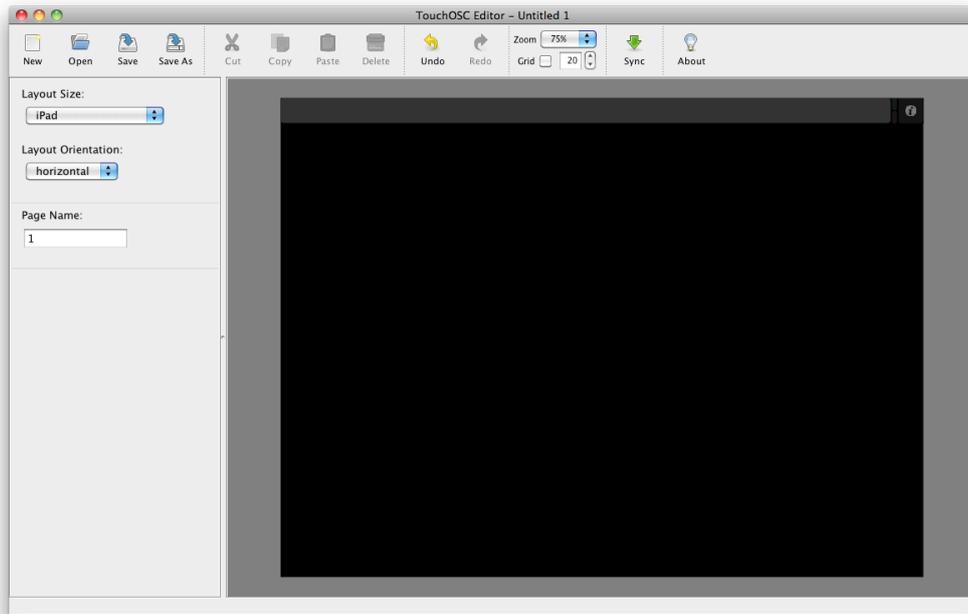


Figura 4.16 – Interface do aplicativo *TouchOSC editor*

Fonte: Autor

O próximo passo é a criação dos botões. Para o projeto em questão foi usado o modelo de botões *toggle*, por apenas ser necessário o envio de sinal “1” para ligar e “0” para desligar. A criação dos botões pode ser feita clicando com o botão direito no local desejado e escolhendo *Toggle Button* conforme figura 4.17.

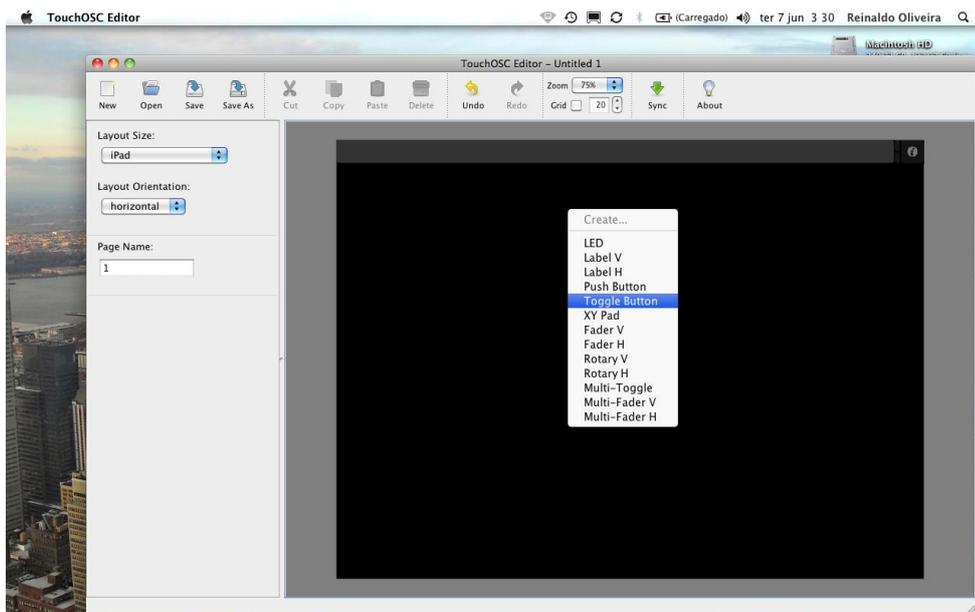


Figura 4.17 – Criando Botões

Fonte: Autor

É necessário observar na figura 4.18, na barra lateral esquerda que deve se manter o nome do botão “toggleN”, onde “N” é o número do botão, isto é necessário pois a próxima camada fará um filtro dos pacotes recebidos com este comando. Mantendo também a marcação em modo “auto”, isto fará com que o pacote UDP com protocolo OSC enviado através do iPad vá com as informação `/1/toggleN`.

Podemos adicionar também texto para a facilitação do entendimento do usuário final, clicando com o botão direito e selecionando “label V”.

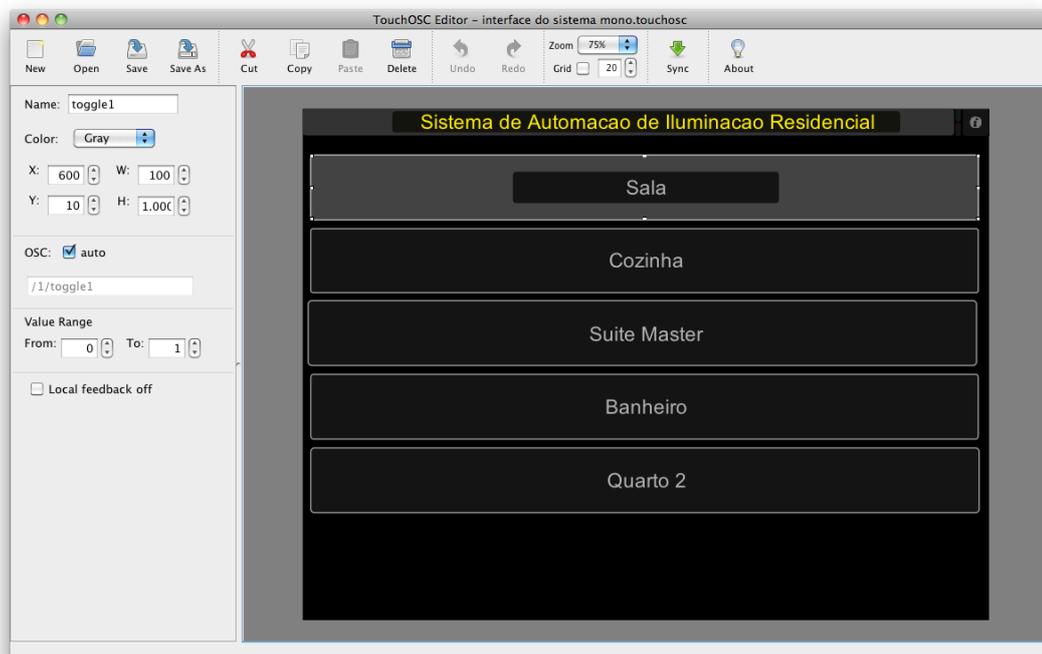


Figura 4.18 – Personalizando a interface

Fonte: Autor

#### 4.3.5. Processing

Esta etapa por representar a camada de interface lógica computacional, proporcionará comunicação correta entre o *Software TouchOSC* instalado no iPad e o *hardware* microcontrolador *Arduino*. Assim como na criação da interface gráfica esta etapa também foi criada totalmente em conjuntos com as outras camadas devendo operar de maneira que integre as 3 camadas fazendo que “conversem” usando nos códigos as mesmas constantes, parâmetros

e comandos respectivamente. Ou seja, os parâmetros usados nas outras camadas devem interagir através desta camada., sendo esta primordial para todo o funcionamento do sistema.

Para a correta leitura dos pacotes contendo protocolo OSC é necessária importação da biblioteca “oscP5”, esta biblioteca pode ser baixada através do site <http://www.sojamo.de/libraries/oscP5/>, crie uma nova pasta chamada oscp5 dentro da pasta de bibliotecas do *Processing* e coloque o conteúdo da biblioteca dentro, feito isso o ambiente estará preparado para receber os pacotes OSC.

Deve-se então iniciar o ambiente de desenvolvimento do *Processing*.

#### 4.3.5.1. Escrevendo o código *Processing*

O primeiro passo após ter iniciado o ambiente *Processing* é certificar-se que podemos incluir a biblioteca oscP5, através dos passos mostrado na figura 4.19.

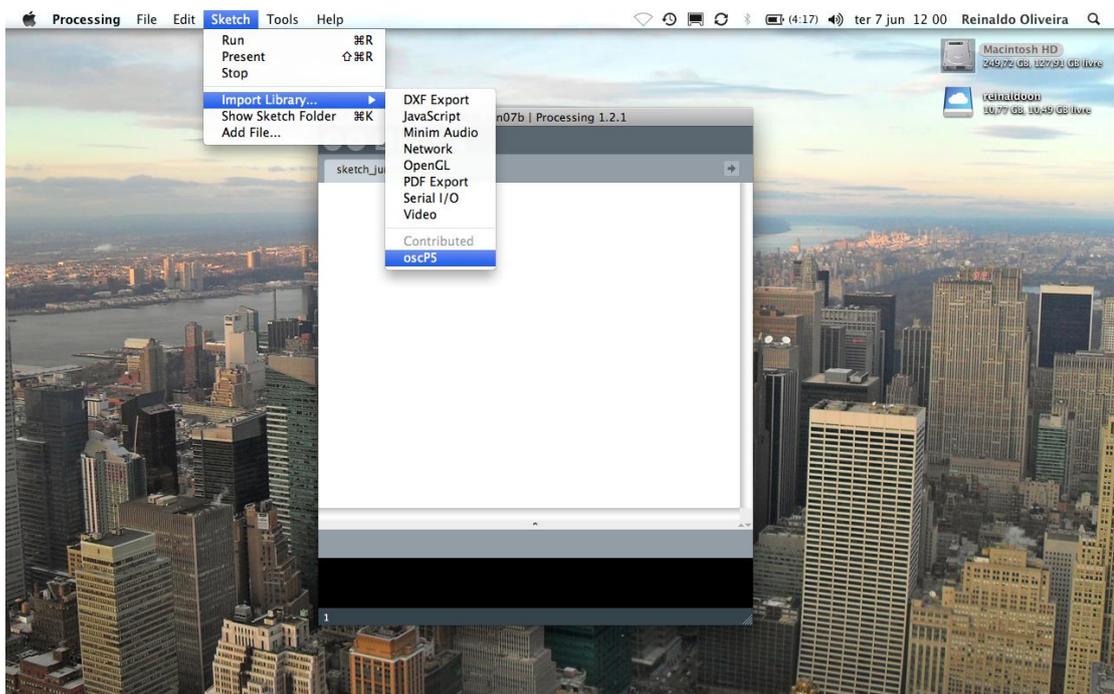
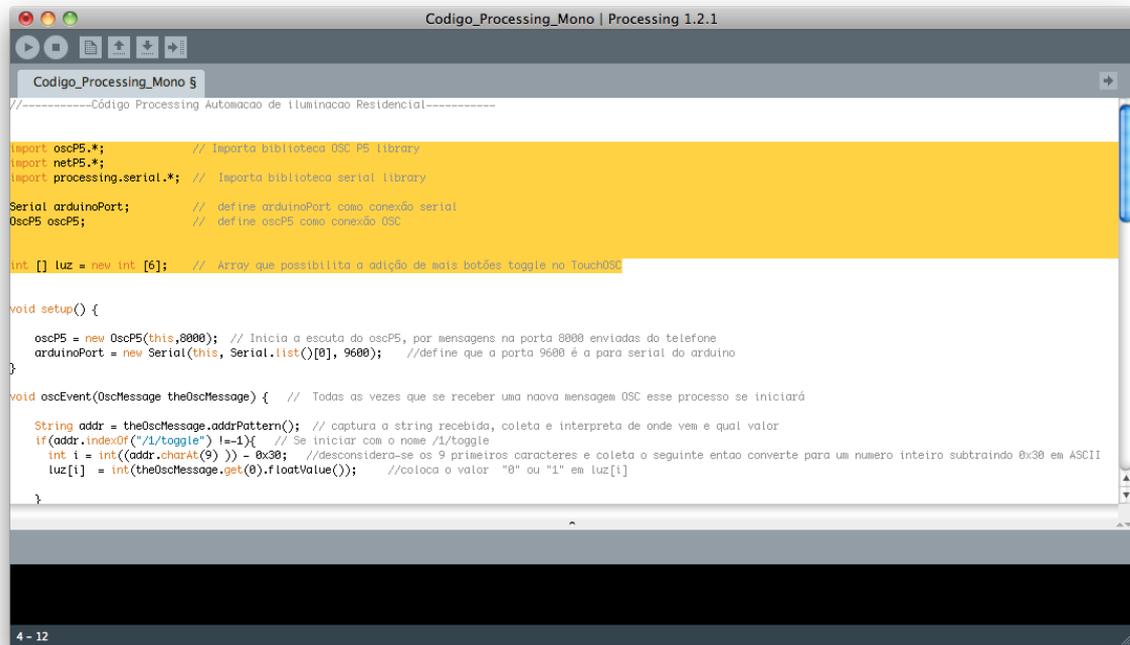


Figura 4.19 – Importando a Biblioteca oscP5

Fonte: Autor

Após a importação das bibliotecas oscP5 e Serial, necessária para enviar uma mensagem serial ao microcontrolador, devemos criar respectivos objetos e um *array* para a

inicialização e adição de mais botões. O código pode ser analisado na íntegra no APENDICE B deste documento.



```

Codigo_Processing_Mono | Processing 1.2.1
Codigo_Processing_Mono $
//-----Código Processing Automacao de iluminacao Residencial-----

import oscP5.*; // Importa biblioteca OSC P5 library
import netP5.*; // Importa biblioteca netP5 library
import processing.serial.*; // Importa biblioteca serial library

Serial arduinoPort; // define arduinoPort como conexão serial
OscP5 oscP5; // define oscP5 como conexão OSC

int [] luz = new int [6]; // Array que possibilita a adição de mais botões toggle no TouchOSC

void setup() {
  oscP5 = new OscP5(this,8000); // Inicia a escuta do oscP5, por mensagens na porta 8000 enviadas do telefone
  arduinoPort = new Serial(this, Serial.list()[0], 9600); //define que a porta 9600 é a para serial do arduino
}

void oscEvent(OscMessage theOscMessage) { // Todas as vezes que se receber uma nova mensagem OSC esse processo se iniciará
  String addr = theOscMessage.addrPattern(); // captura a string recebida, coleta e interpreta de onde vem e qual valor
  if(addr.indexOf("/1/toggle") != -1){ // Se iniciar com o nome /1/toggle
    int i = int((addr.charAt(9)) - 0x30); //desconsidera-se os 9 primeiros caracteres e coleta o seguinte então converte para um numero inteiro subtraindo 0x30 em ASCII
    luz[i] = int(theOscMessage.get(0).floatValue()); //coloca o valor "0" ou "1" em luz[i]
  }
}
  
```

Figura 4.20 – Código contendo as bibliotecas e objetos

Fonte: Autor

As linhas do código mostradas na figura 4.21 são usadas para inicializar os objetos `oscP5`, permitindo iniciar a escuta de novas mensagens recebidas através da porta “8000” a qual deverá ser também configurada na interface do iPad. E a porta serial, permitindo tanto leitura quanto escrita nas portas escolhidas. O trecho “`Serial.list()[0]`” indica qual porta gostaríamos de escrever, no projeto em questão é necessário que seja “0”. Em seguida temos o valor do *baud*, que representa o numero de eventos por segundo que são transferidos, neste protótipo foi utilizado o valor “9600”.

```

Codigo_Processing_Mono $
Serial arduinoPort; // define arduinoPort como conexão serial
OscP5 oscP5; // define oscP5 como conexão OSC

int [] luz = new int [6]; // Array que possibilita a adição de mais botões toggle no TouchOSC

void setup() {
  oscP5 = new OscP5(this,8000); // Inicia e escuta do oscP5, por mensagens na porta 8000 enviadas do telefone
  arduinoPort = new Serial(this, Serial.list()[0], 9600); //define que a porta 9600 é a para serial do arduino
}

void oscEvent(OscMessage theOscMessage) { // Todas as vezes que se receber uma naova mensagem OSC esse processo se iniciará
  String addr = theOscMessage.addrPattern(); // captura a string recebida, coleta e interpreta de onde vem e qual valor
  if(addr.indexOf("/1/toggle") !=-1){ // Se iniciar com o nome /1/toggle
    int i = int((addr.charAt(9) )) - 0x30; //desconsidera-se os 9 primeiros caracteres e coleta o seguinte entao converte para um numero inteiro subtraindo 0x30 em ASCII
    luz[i] = int(theOscMessage.get(0).floatValue()); //coloca o valor "0" ou "1" em luz[i]
  }
}

void draw() {
  if(luz[1] == 0){ // se valor recebido for 0
    arduinoPort.write("s"); // Envia o caracter s para o arduino
  }
}

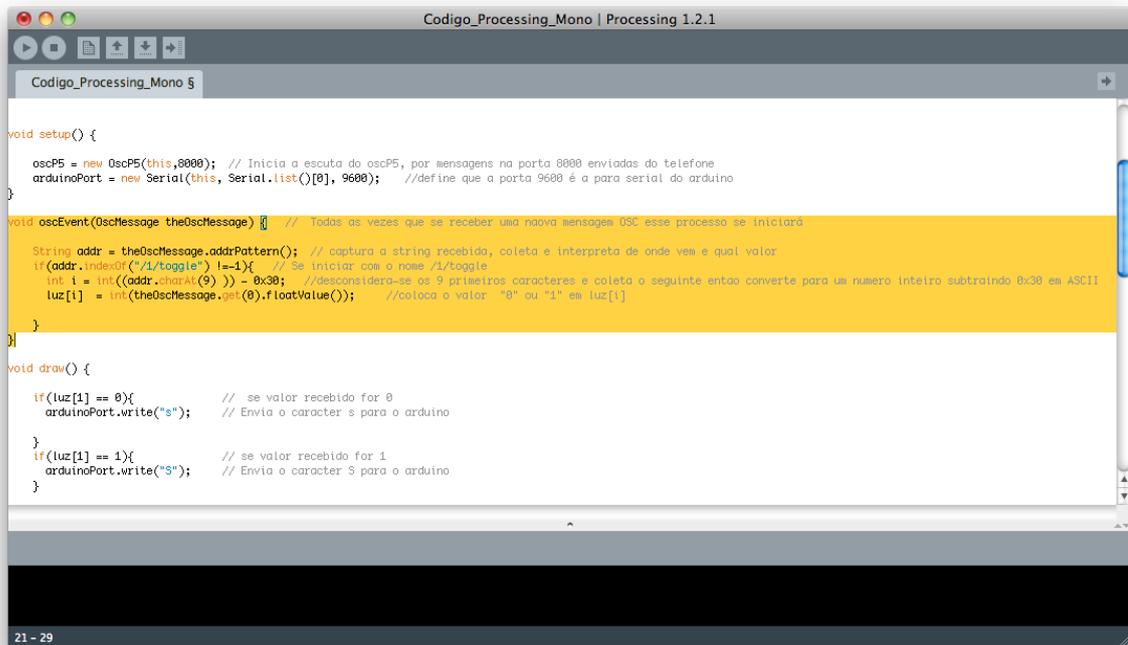
```

Figura 4.21 – Definindo parâmetros de comunicação

Fonte: Autor

Agora devemos informar no código o que o programa deve executar quando uma mensagem OSC é recebida, isto é feito através da função *oscEvent*.

A partir do momento que uma nova mensagem é interceptada pela porta “8000” as seguintes linhas do código permitem que ela possa ser salva como uma *string* interpretando quais foram os caracteres recebido e qual o valor "0" ou "1" coletado. Se a string for iniciada com os caracteres “/1/toggle”, conforme estabelecidos pela interface gráfica anteriormente, continua a execução. Para a interpretação de qual o número do botão foi pressionado na interface gráfica deve se desconsiderar os 9 primeiros caracteres da string e subtrair o último de “0x30” conforme tabela ASCII, feito isto se tem um valor inteiro que deve ser salvo em um parâmetro “i” que será utilizado na função seguinte. Para a interpretação se o comando foi de desligar ou ligar, "0" ou "1", o código insere o parâmetro no objeto luz[i] onde “i” representa o botão acionado.



```

Codigo_Processing_Mono | Processing 1.2.1

Codigo_Processing_Mono 5

void setup() {
  oscP5 = new OscP5(this,8000); // Inicia a escuta do oscP5, por mensagens na porta 8000 enviadas do telefone
  arduinoPort = new Serial(this, Serial.list()[0], 9600); //define que a porta 9600 é a para serial do arduino
}

void oscEvent(OscMessage theOscMessage) {} // Todas as vezes que se receber uma nova mensagem OSC esse processo se iniciará

String addr = theOscMessage.addrPattern(); // captura a string recebida, coleta e interpreta de onde vem e qual valor
if(addr.indexOf("/L/toggle") !=-1){ // se iniciar com o nome /L/toggle
  int i = int((addr.charAt(9)) - 0x30); //desconsidera-se os 9 primeiros caracteres e coleta o seguinte então converte para um numero inteiro subtraindo 0x30 em ASCII
  luz[i] = int(theOscMessage.get(0).floatValue()); //coloca o valor "0" ou "1" em luz[i]
}
}

void draw() {
  if(luz[1] == 0){ // se valor recebido for 0
    arduinoPort.write("s"); // Envia o caracter s para o arduino
  }
  if(luz[1] == 1){ // se valor recebido for 1
    arduinoPort.write("S"); // Envia o caracter S para o arduino
  }
}

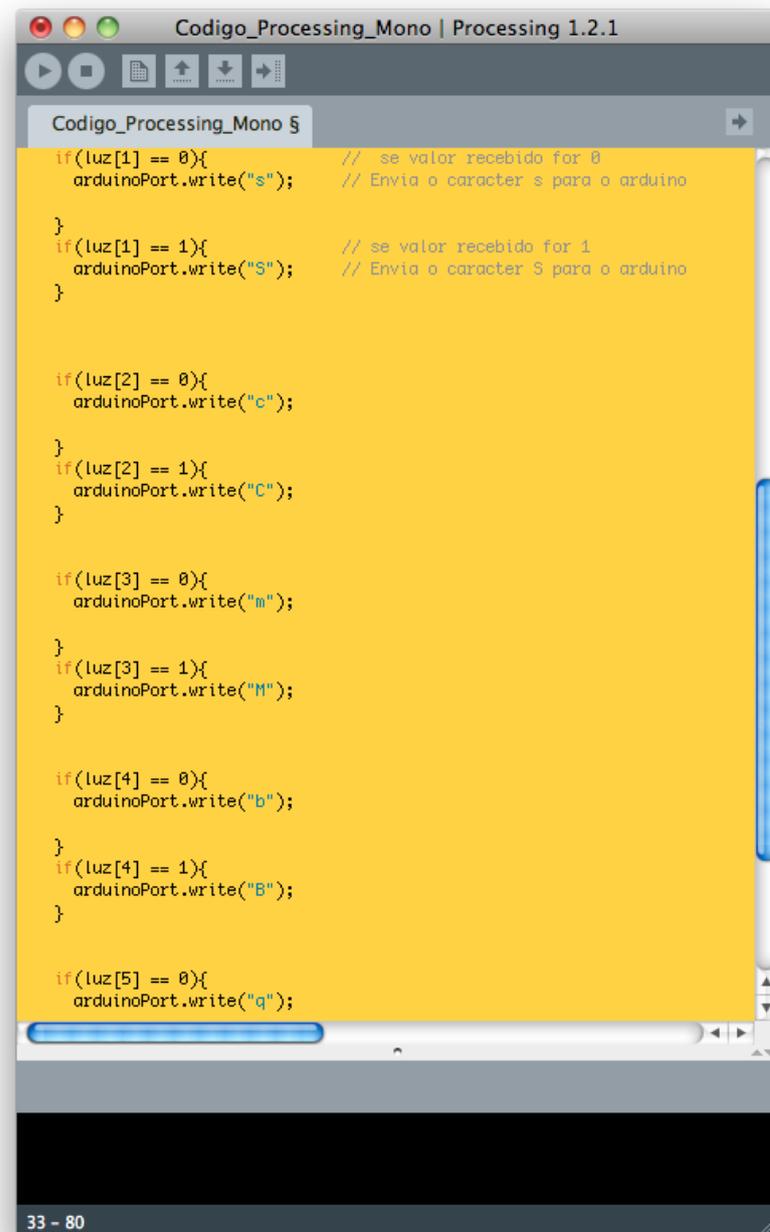
```

21 - 29

Figura 4.22 – Função *oscEvent*

Fonte: Autor

A seqüência do código permite a escrita e envio do comando correto para ligar ou desligar a lâmpada através da porta serial, aproveitando os comandos já citados no quadro 2 concluindo a integração entre as 3 camadas de desenvolvimento.

The image shows a screenshot of the Processing IDE window titled "Codigo\_Processing\_Mono | Processing 1.2.1". The code is written in a yellow-themed editor and consists of several conditional blocks. Each block checks a value in an array named 'luz' and writes a corresponding character to the serial port using 'arduinoPort.write()'. The characters are 's', 'S', 'c', 'C', 'm', 'M', 'b', 'B', and 'q'. The code is as follows:

```
Codigo_Processing_Mono §
if(luz[1] == 0){ // se valor recebido for 0
  arduinoPort.write("s"); // Envia o caracter s para o arduino
}
if(luz[1] == 1){ // se valor recebido for 1
  arduinoPort.write("S"); // Envia o caracter S para o arduino
}

if(luz[2] == 0){
  arduinoPort.write("c");
}
if(luz[2] == 1){
  arduinoPort.write("C");
}

if(luz[3] == 0){
  arduinoPort.write("m");
}
if(luz[3] == 1){
  arduinoPort.write("M");
}

if(luz[4] == 0){
  arduinoPort.write("b");
}
if(luz[4] == 1){
  arduinoPort.write("B");
}

if(luz[5] == 0){
  arduinoPort.write("q");
}
```

Figura 4.23 – Escrita do comando na porta serial

Fonte: Autor

#### 4.3.6. Configurando e sincronizando a interface gráfica no iPad

O primeiro passo deve ser fazer o download do aplicativo *TouchOSC* através da App Store da *Apple*, o custo do aplicativo na data atual é \$4.99 (quatro dólares e noventa e nove centavos).

O iPad e o computador rodando o *Processing* devem estar conectados na mesma rede *Wi-Fi*. No Mac OS X você pode localizar isso clicando na maçã no canto superior esquerdo e em seguida Sobre Este Mac>Mais informações>Rede. Caso não possua uma rede *Wi-Fi* disponível pode ser criada uma rede *ad-hoc* entre os dispositivos.

Uma vez instalado o aplicativo pressione o botão “i” na parte superior direita do aplicativo. Depois selecione *connections* a tela do iPad será como a da figura 4.24, no campo *host* deve ser informado o endereço IP do computador que estará rodando o *Processing*. A porta de saída deve ser a mesma informada no código *Processing*, neste caso “8000”.

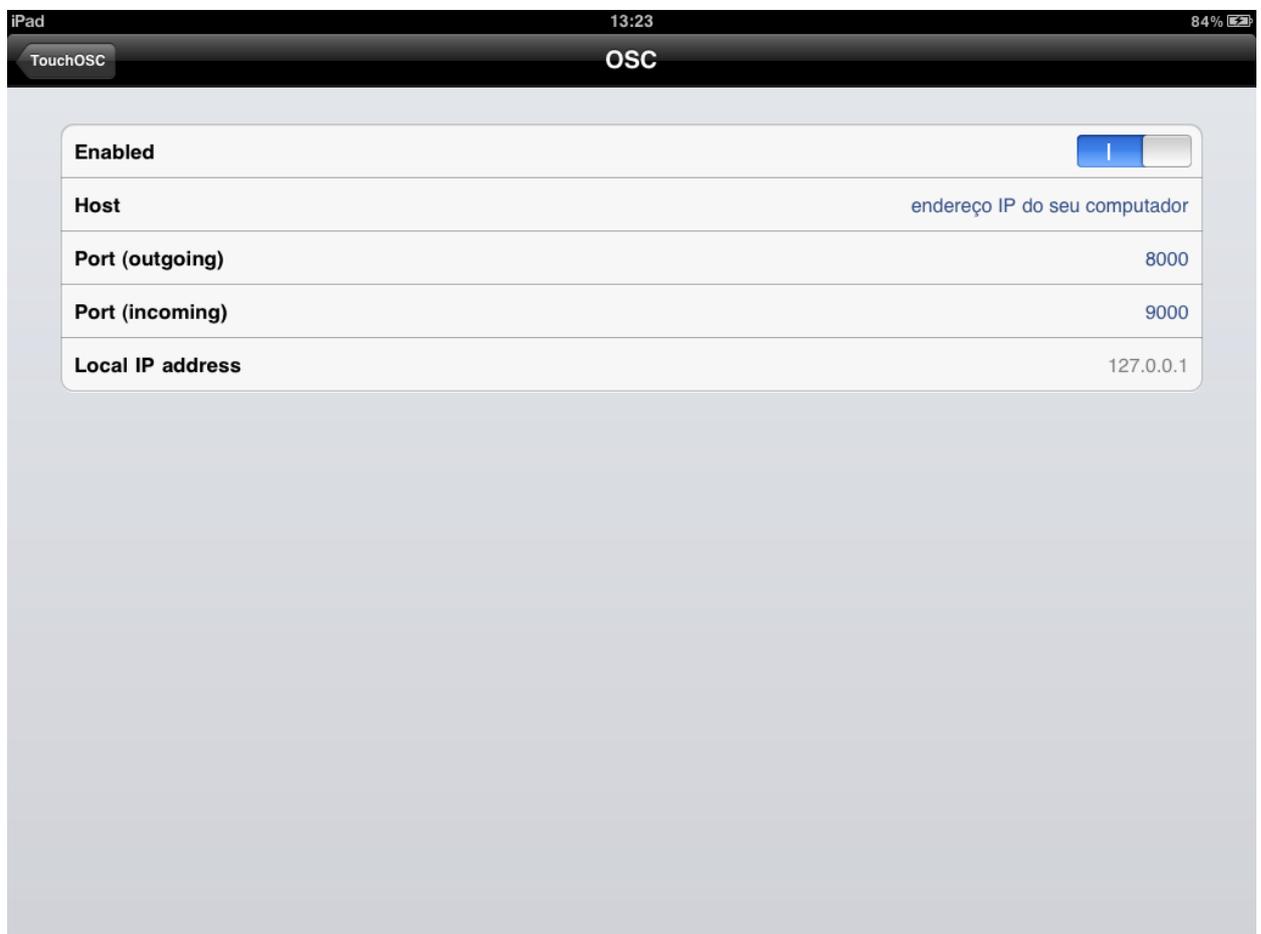


Figura 4.24 – Configurando endereço IP e porta de saída

Fonte: Autor

Em seguida devemos sincronizar a interface criada no *Software TouchOSC editor*, para isso abra o aplicativo e a interface desenvolvida anteriormente na etapa 4.3.4 Interface

Gráfica, em seguida clique no botão *sync* na parte superior do aplicativo, a tela no computador será apresentada como a Figura 4.25.

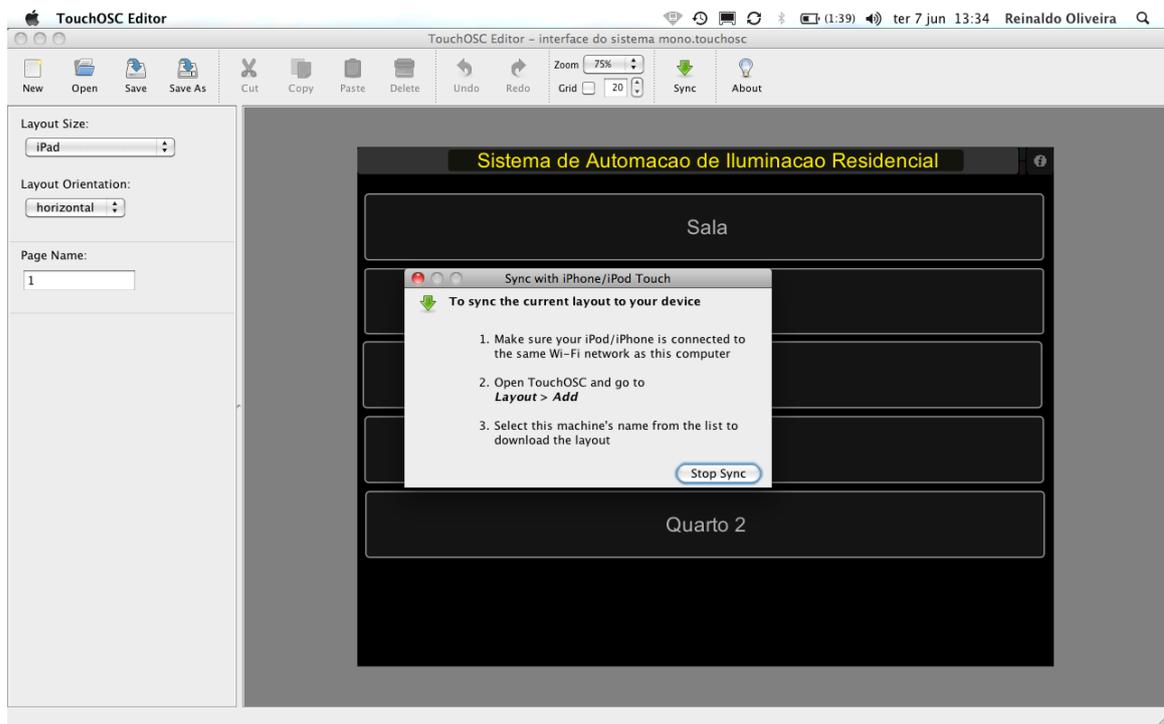


Figura 4.25 – Sincronizando a nova interface com o iPad

Fonte: Autor

Em seguida, no iPad clique sobre *Layout* e *Add*, deverá aparecer o nome da sua máquina. No momento em que clicar será feita a sincronização e download da interface criada. Depois selecione a interface sincronizada, conforme figura 4.26.



Figura 4.26 – Selecionando a interface gráfica para exibição

Fonte: Autor

Certifique-se que dentro de *Options*, *Stay Connected* esteja marcado como ligado, *accelerometer* e *send z messages* desligados.

## CAPÍTULO 5 - APLICAÇÃO PRÁTICA DO MODELO PROPOSTO

### 5.1. Apresentação da área de Aplicação do modelo

Inicialmente, por se tratar de um protótipo acadêmico, apenas testes de implementação em micro escala foram feitos, no entanto através de um estudo e avaliação podemos constatar sua viabilidade comercial e de implementação em ambientes heterogêneos, não apenas para a automação de iluminação e sim de diversos dispositivos e em diferentes tipos de ambientes desde industriais à prediais.

No desenvolvimento da plataforma considerou-se a decomposição do sistemas em camadas, cada um com suas funções, responsabilidades e protocolos de entrada e saídas de dados, no entanto a integração destas camadas foi crucial para o funcionamento do sistema, devendo portanto todas elas utilizarem os mesmos parâmetros nos códigos de programação. Na finalização todas as camadas dependem de informações das outras camadas para o correto funcionamento.

### 5.2. Descrição da Aplicação do Modelo

Alguns experimentos foram realizados para testar o funcionamento do protótipo. Em primeiro lugar, o *hardware* foi testado sem conexão *wifi*, com os comandos sendo enviados diretamente pelo notebook. Então, testou-se todo o sistema, incluindo o iPad.

#### 5.2.1. Testes do *Hardware*

Os testes do *hardware* objetivaram verificar:

- a) A correta comunicação entre o notebook e o microcontrolador.
- b) O correto funcionamento dos comandos x respectivas lâmpadas.

Para a verificação do correto funcionamento, pode ser efetuado o envio de comandos através da funcionalidade Serial Monitor, disponível no *Software* de compilação do *Arduino*, ao enviar o comando “S” maiúsculo deverá ser acionado a luz do cômodo sala, ao enviar “s” minúsculo deve ocorrer o desligamento.

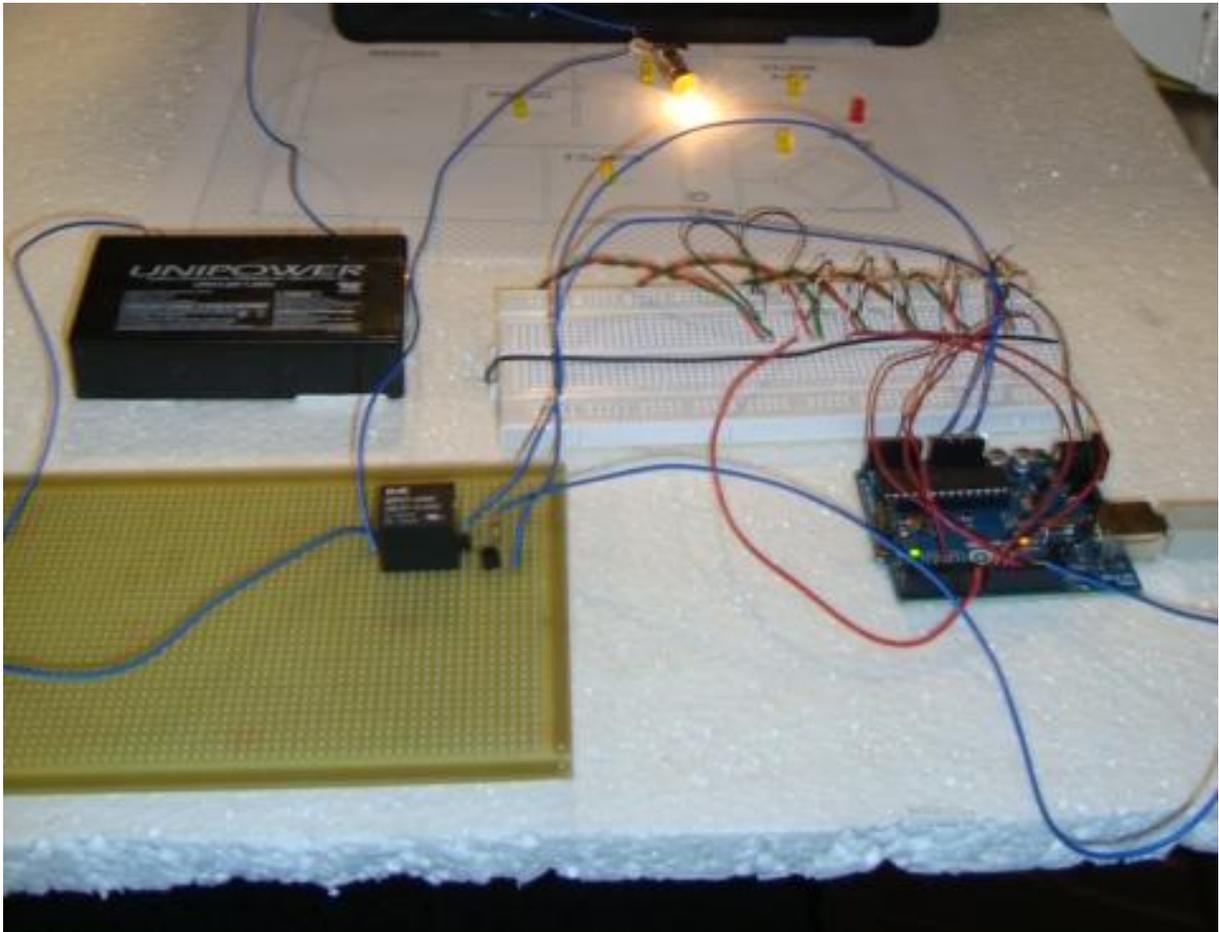


Figura 5.1 – Detalhe do *hardware* microcontrolador e lâmpada.

Fonte: Autor

### 5.2.2. Teste do Sistema

Uma vez verificada a correta comunicação entre o notebook e o microcontrolador, efetuou-se o teste da plataforma completa. O iPad possui o aplicativo *TouchOSC* previamente instalado, por meio deste aplicativo que serão enviados os comandos que acionaram as lâmpadas. Para que seja possível enviar os dados do *tablet* para a o *Arduino* é necessário que o *tablet* e o computador que receberá as informações estejam na mesma rede *wifi*, o computador

deverá possuir um IP válido e fixo, o *Software Processing* deverá estar em modo *run* e com o código preparado, portas e taxas de transmissão sendo as mesmas configuradas no aplicativo *TouchOSC* e no *hardware* microcontrolador, para receber e enviar as informações de maneira correta.

Após a correta conexão os testes de fato podem ser efetuados, ao pressionar o botão, denominado “Sala” na tela do *tablet*, a luz correspondente deverá acender na maquete.

Isso ocorre pelo fato, que o momento em que o botão pressionado é pressionado o aplicativo dispara o envio de um pacote com a string “/1/toggle1” onde “1” representa o número do botão e o valor para ligar ou desligar a luz, a string é capturada e o *Processing* faz a interpretação, desconsiderando os nove primeiros caracteres da string e transformando o último caractere em um número. O valor recebido é inserido em um outro parâmetro, o qual é submetido a uma nova leitura e envio do comando “S” maiúsculo ou “s” minúsculo, para o *Arduino* através de uma conexão serial iniciada. Com isso o *Arduino* executará a ligação da lâmpada do cômodo sala.

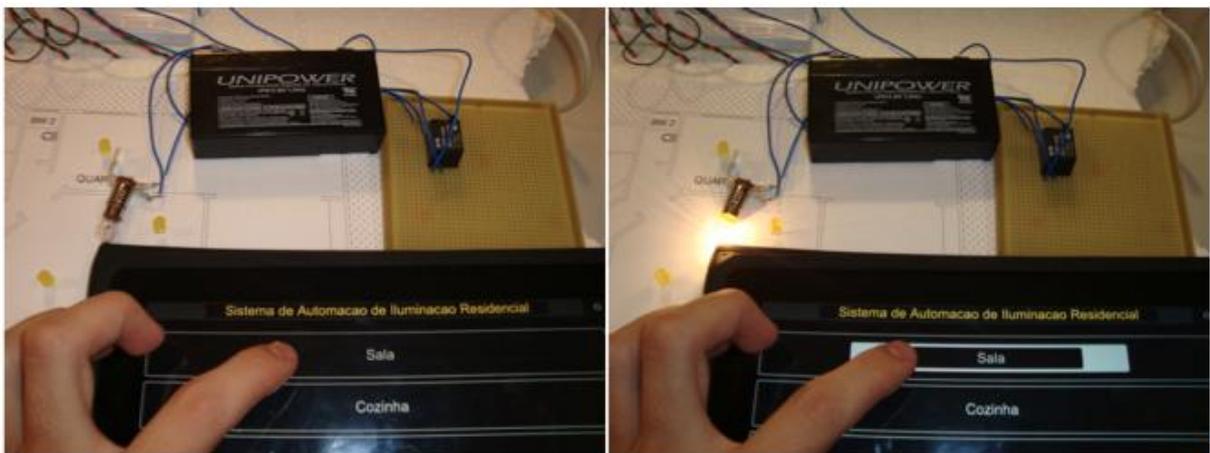


Figura 5.2 – Teste de funcionamento do sistema

Fonte: Autor

### 5.3. Resultados da Aplicação do Modelo

Como resultado da finalização do protótipo infere-se a possibilidade acionar atuadores para ligar ou desligar as luzes de uma maquete residencial.

O resultado é obtido através de um conjunto de *Softwares* e *hardwares*. Com o uso do dispositivo móvel iPad e permitido o envio de uma mensagem ao computador conectado ao *hardware* microcontrolador, interpretando e acionando o relé correto e correspondente de cada cômodo da maquete.

Se espera do iPad em conjunto com o *Software TouchOSC* o correto envio da mensagem ao computador para a análise do pacote e reenvio ao *hardware Arduino*.

#### **5.4. Custos do modelo proposto**

O orçamento do projeto foi onerado em função de ser acadêmico, portanto diversos testes e falhas foram apresentados durante a implementação, sendo necessários diversos insumos que acabaram por ser dispensados durante o processo de construção. Para a construção final do projeto não é necessária a compra dos diversos itens eletrônicos relacionados no início do desenvolvimento, O kit *Arduino* iniciante é dispensado, sendo necessário somente o *Arduino UNO*. Como sugestão para redução de custos podemos citar o fato da plataforma *Arduino* ser *Open-Hardware*, desta forma sua montagem é disponibilizada no próprio site do fabricante. Da mesma maneira com o objetivo de redução de custos, alguns itens que forma comprados em mercado nacional, se comprados por importação o custo pode ser barateado. Os custos estão representados no quadro 3.

Alguns itens foram orçados em dólar, para a disposição dos custo foi usada a taxa de cambio atual (USD X BRL).

Quadro 3 – Custo total dos equipamentos utilizando no projeto

Descrição	Origem	Quantidade	Valor Unitário	Valor Total
Kit <i>Arduino</i> iniciante		1	R\$ 220,00	R\$ 220,00
Itens eletrônicos	Contato	1	R\$ 60,00	R\$ 60,00
iPad	<i>Apple</i>	1	R\$ 2.299,00	R\$ 2.299,00
<i>Notebook</i>	<i>Apple</i>	1	R\$ 3.599,00	R\$ 3.599,00
Maquete	Própria	1	R\$ 100,00	R\$ 100,00
<i>Software TouchOSC</i>	App Store	1	R\$ 10,00	R\$ 10,00
			<b>Valor Global</b>	<b>R\$ 6.288,00</b>

Fonte: Autor

### 5.5. Avaliação Global do Modelo

O uso do iPad, notebook e *Arduino* interligados por uma rede sem fio apresenta importantes vantagens em relação aos atuais métodos de acionamento de lâmpadas. A facilidade e comodidade talvez sejam os mais nítidos no primeiro momento, no entanto a acessibilidade promovida pela solução se destaca no âmbito de importância para a sociedade, entre outros fatores a economia de energia promovida pela solução também pode ser levada em consideração. Nos testes a configuração da comunicação foi estabelecida em minutos. Neste trabalho, a interferência eletromagnética de outras fontes operando na mesma frequência como, por exemplo, telefones sem fio, roteadores Wi-Fi ou mesmo fornos de microondas não foi objeto de estudo, a obstrução devido ao fechamento de portas ou à movimentação de pessoas causou instabilidade no sistema e o número de pacotes perdidos aumentava substancialmente devido a esses eventos. Durante os testes em redes *wifi* com baixa intensidade devido a distância do roteador ou interferências externas, o protótipo se mostrou instável, ao acionar o botão no *tablet* nem sempre o pacote foi capturado pelo *Processing*, promovendo uma ilusão ao que realmente estava ocorrendo, na tela do *tablet* era mostrado que o botão estava acionado e portanto a luz também deveria estar, no entanto nem

sempre isso foi verificado.

A utilização de um notebook pessoal como meio de transporte dos pacotes é um ponto delicado para tornar o sistema comercial, esse fator pode ser contornado através do uso de outros componentes acoplados ao microcontrolador, existem shields disponíveis no mercado que possuem a funcionalidade *wifi* ou até mesmo com a possibilidade de inserir chips telefônicos GSM Neste caso, o notebook seria dispensado e o envio dos comandos seria de maneira direta entre o iPad e o microcontrolador. O uso destes complementos foi descartado no projeto em questão devido ao seu custo e a dificuldade de encontrá-los no mercado nacional.

## CAPÍTULO 6 - CONCLUSÃO

### 6.1. Conclusões

As oportunidades tecnológicas oferecidas neste século, descritas e discutidas no decorrer da pesquisa, impulsionaram o desenvolvimento do mercado de automação residencial. O panorama abordado no Capítulo 2 evidencia a necessidade e benefícios de sistemas de residências inteligentes, principalmente por pessoas deficientes, o contexto nos possibilita perceber que além de acessibilidade podemos considerar a automação como um conjunto de ferramentas que promove maior bem estar, conforto e economia para as pessoas, possibilitando amplificar as interações entre os moradores, sejam eles idosos, crianças ou deficientes, e as suas residências.

Desta forma, a criação de um sistema de automação de iluminação residencial e a viabilidade tecnológica da utilização do uso de um *tablet* para o controle no ambiente residencial, proposto nesta monografia, foi validada por meio de uma arquitetura adequada e da integração entre os códigos desenvolvidos.

A plataforma, que integrou iPad, *Arduino*, *Processing* cumpriu o estabelecido no escopo que foi proposto inicialmente.

A principal dificuldade encontrada foi a integração entre os códigos, pois todos deviam possuir os mesmos parâmetros e para receber informações dos demais. Apesar das linguagens serem baseadas em C e C++ a troca de informações se davam através de bibliotecas específicas, já que inicialmente, o autor teve que superar a falta de conhecimento das mesmas.

Apesar de atingir os objetivos propostos, o protótipo tem o limite de somente poder controlar dez pinos, isso se dá pelo fato da linguagem estar preparada para a leitura específica de um caractere da string recebida, este fator poderia ser contornado com uma alteração no modo de como é feita a leitura, construindo uma nova função e leitura para cada novo botão implementado.

## 6.2. Sugestões para Trabalhos Futuros

Como tentativa de tornar o protótipo mais eficiente e móvel, eliminando o uso do notebook e plataforma *Processing* como ponte de comunicação entre o iPad e o microcontrolador, sugere-se a inclusão de um *hardware* com capacidade *wifi* acoplado ao *Arduino*. Este *hardware* já existe no mercado, no entanto não foi implementado devido aos custos do projeto estarem relativamente altos. Seria necessária a adequação do código de programação, inclusão de novas bibliotecas permitindo os novos *hardwares* comunicarem entre si.

Outra necessidade que poderia ser desenvolvida é o aplicativo específico para o projeto, desenvolvendo uma interface gráfica podendo conter a planta baixa da residência que se quer automatizar, com isso a tela *touchscreen* seria melhor aproveitada e a garantia de recebimento dos pacotes enviados pela nova plataforma, com um fator de confirmação de recebimento do pacote por parte do *hardware* microcontrolador.

Além da automação e dimerização da iluminação poderia ser proposta a automação de outros sistemas da residência, por exemplo, o monitoramento de câmeras através da rede 3G, eliminando a necessidade do *tablet* estar na mesma rede *wifi*.

## REFERÊNCIAS

APPLE. **Apple Iphone Mobile Phone, Ipod and Internet Device**. 2008. Disponível em <http://www.apple.com/iphone/>. Acessado em: 15 de março de 2011.

ARDUINO. **Arduino**. Disponível em: <<http://www.Arduino.cc>> Acesso em: 03 de março de 2011.

BESSEN, Nelson. **Sistema doméstico para automação e controle de um cômodo residencial**. 1996. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BOLZANI, C. **Residências Inteligentes**: Editora Livraria da Física, 2004.

BOLZANI, C.; NETTO, M. **The engineering of micro agents in smart environments**. *International Journal of Knowledge-based and Intelligent Engineering Systems*, v. 13, n. 1, p. 31–38, 2009.

DEPINÉ, Fábio M. **Protótipo de Software para dispositivos móveis utilizando Java ME para cálculo de regularidade em rally**. 2002. 55 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

E. Sebastian. Going keyless: **How it will help people with mobility issues**. Disponível em <http://www.disaboom.com/Living/accessiblehomearticles/going-keyless-how-it-will-help-people-with-mobility-issues.aspx>, 2008.

GALVIN, Deleon. **Protótipo de sistema de CRM para dispositivos móveis utilizando a tecnologia .NET**. 2004. 89 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KRÜGER, Erasmo. **Protótipo de sistema de segurança predial através de monitoramento utilizando recursos da internet**. 2002. 61 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MEYER, Gordon. **Smarth Home Hacks: Tips & Tools for Automating Your House. Sebastopol: O'Reilly Mídia, 2004.**

MURATORI, José Roberto; FORTI, José Cândido; OMAI, Paulo. Associação Brasileira de Automação Residencial : Home Cabling Training Manual. 2004.

MURATORI, José Roberto, AURESIDE, **Integração de Sistemas Residenciais**, Congresso BICSI 2001, São Paulo, 2001

MURATORI, J. R.; **Instalações Elétricas com novas abordagens**. 2008. Artigo disponível em [http://www.institutodofuturo.com.br/setor\\_eletrico\\_set08.htm](http://www.institutodofuturo.com.br/setor_eletrico_set08.htm). Acesso em: 05 de março de 2011.

N. Farias; C. M. Buchalla. **A classificação internacional de funcionalidade, incapacidade e saúde da organização mundial da saúde: conceitos, usos e perspectivas**. Revista Brasileira de Epidemiologia, 8(2):187–193, 2005.

PROCESSING. **Processing**. Disponível em: <<http://www.processing.org>> Acesso em: 03 de março de 2011.

REAS, C.; FRY, B., **Processing: A Programming Handbook for Visual Designers and Artists**. Cambridge: MIT Press, 2007.

SCHAEFER, Carina. **Protótipo de aplicativo para transmissão de dados a partir de dispositivos móveis aplicado a uma empresa de transportes**. 2004. 52 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

**APÊNDICE A**

```
//-----Código Arduino Controle de iluminacao residencial-----
```

```
int mensagem = 0; // Manda um sinal para a mensagem serial
```

```
int pinoLEDSala = 12; // qual pino o led está conectado
```

```
int LEDsala = 0; // valor do brilho do pino
```

```
int pinoLEDcozinha = 11;
```

```
int LEDcozinha = 0;
```

```
int pinoLEDsuite = 10;
```

```
int LEDsuite = 0;
```

```
int pinoLEDbanheiro = 9;
```

```
int LEDbanheiro = 0;
```

```
int pinoLEDquarto2 = 8;
```

```
int LEDquarto2 = 0;
```

```
void setup() {
```

```
  Serial.begin(9600); //entrada serial
```

```
}
```

```
void loop(){  
  
  if (Serial.available() > 0) { // Checa se existe uma nova mensagem chegando  
    mensagem = Serial.read(); // Coloca o sinal da porta serial em mensagem  
  
    if (mensagem == 'S'){ //Se o sinal recebido for SALA (MAIUSCULO)  
      LEDsala = 255; //Seta o brilho do LED 255 (ligado)  
    }  
  
    if (mensagem == 's'){ //Se o sinal recebido for sala (minusculo)  
      LEDsala = 0; // Seta o brilho do LED 0 (desligado)  
    }  
  
    if (mensagem == 'C'){  
      LEDcozinha = 255;  
    }  
  
    if (mensagem == 'c'){  
      LEDcozinha = 0;  
    }  
  
    if (mensagem == 'M'){  
      LEDsuite = 255;  
    }  
  
    if (mensagem == 'm'){  
      LEDsuite = 0;  
    }  
  
    if (mensagem == 'B'){
```

```
    LEDbanheiro = 255;
}
if (mensagem == 'b'){
    LEDbanheiro = 0;
}

if (mensagem == 'Q'){
    LEDquarto2 = 255;
}
if (mensagem == 'q'){
    LEDquarto2 = 0;
}
}

analogWrite(pinoLEDSala, LEDsala); // escreve o valor analogico no pino do LED vermelho
analogWrite(pinoLEDcozinha, LEDcozinha);
analogWrite(pinoLEDsuite, LEDsuite);
analogWrite(pinoLEDbanheiro, LEDbanheiro);
analogWrite(pinoLEDquarto2, LEDquarto2);
analogWrite(pinoLEDlove, LEDlove);
}

//-----Código Arduino Controle de iluminacao residencial-----
```

## APÊNDICE B

//-----Código *Processing* Automacao de iluminacao Residencial-----

```
import oscP5.*;          // Importa biblioteca OSC P5 library
```

```
import netP5.*;
```

```
import Processing.serial.*; // Importa biblioteca serial library
```

```
Serial ArduinoPort;     // define ArduinoPort como conexão serial
```

```
OscP5 oscP5;           // define oscP5 como conexão OSC
```

```
int [] luz = new int [6]; // Array que possibilita a adição de mais botões toggle no
TouchOSC
```

```
void setup() {
```

```
    oscP5 = new OscP5(this,8000); // Inicia a escuta do oscP5, por mensagens na porta 8000
    enviadas do telefone
```

```
    ArduinoPort = new Serial(this, Serial.list()[0], 9600); //define que a porta 9600 é a para
    serial do Arduino
```

```
}
```

```
void oscEvent(OscMessage theOscMessage) { // Todas as vezes que se receber uma naova
    mensagem OSC esse processo se iniciará
```

String addr = theOscMessage.addrPattern(); // captura a string recebida, coleta e interpreta de onde vem e qual valor

```
if(addr.indexOf("/1/toggle") !=-1){ // Se iniciar com o nome /1/toggle
```

```
    int i = int((addr.charAt(9)) - 0x30); //desconsidera-se os 9 primeiros caracteres e coleta o seguinte entao converte para um numero inteiro subtraindo 0x30 em ASCII
```

```
    luz[i] = int(theOscMessage.get(0).floatValue()); //coloca o valor "0" ou "1" em luz[i]
```

```
}
```

```
}
```

```
void draw() {
```

```
    if(luz[1] == 0){ // se valor recebido for 0
```

```
        ArduinoPort.write("s"); // Envia o caracter s para o Arduino
```

```
}
```

```
    if(luz[1] == 1){ // se valor recebido for 1
```

```
        ArduinoPort.write("S"); // Envia o caracter S para o Arduino
```

```
}
```

```
    if(luz[2] == 0){
```

```
        ArduinoPort.write("c");
```

```
}
```

```
    if(luz[2] == 1){
```

```
        ArduinoPort.write("C");
```

```
}
```

```
if(luz[3] == 0){  
    ArduinoPort.write("m");  
  
}  
  
if(luz[3] == 1){  
    ArduinoPort.write("M");  
  
}  
  
if(luz[4] == 0){  
    ArduinoPort.write("b");  
  
}  
  
if(luz[4] == 1){  
    ArduinoPort.write("B");  
  
}  
  
if(luz[5] == 0){  
    ArduinoPort.write("q");  
  
}  
  
if(luz[5] == 1){  
    ArduinoPort.write("Q");  
  
}  
}  
  
//-----Fim do Código Processing Automacao de iluminacao residencial-----
```