

Home Zeus: um protótipo de gerenciamento de casa inteligente

Zeus Schmitz, Prof. Ricardo de la Rocha Ladeira (orientador)

Instituto Federal Catarinense – Campus Blumenau (IFC)
Blumenau, SC – Brasil

zeus.schmitz2@gmail.com, ricardo.ladeira@ifc.edu.br

Resumo. *A evolução da tecnologia possibilita cada vez mais conforto aos seres humanos. Existe uma gama enorme de dispositivos eletroeletrônicos instalados em uma residência para torná-la tecnológica. Entretanto, esse progresso esbarra na falta de interconectividade entre esses dispositivos. Todo esse agrupamento tecnológico precisa ser controlado de maneira que facilite e torne mais confortável a vida das pessoas. A partir do momento em que os dispositivos eletrônicos instalados nos ambientes das casas estão integrados, eles são capazes de agregar novas funcionalidades e formas de interação para determinado ambiente. A presente pesquisa apresenta um estudo teórico e a aplicação prática do controle remoto de uma placa NodeMcu ESP8266 wi-fi, que simula a automação de uma residência inteligente. O dispositivo é gerido por meio de um aplicativo Android criado, por este aluno, que controla o estado do led (ligado ou desligado), um sensor de temperatura e um sensor de barreira de qualquer lugar, desde que esteja conectado a rede de internet. O objetivo é desenvolver um estudo que possibilite a implementação do aplicativo ZETECH (Zeus Tecnologia — Controle e Monitoramento de Residência Remoto) e a prototipação da placa NodeMCU ESP8266 para controlar e monitorar as informações que passam pela placa de qualquer lugar com acesso à internet. A pesquisa se mostra importante pois a interconectividade de dispositivos eletroeletrônicos gera muitas oportunidades tanto no âmbito acadêmico quanto no industrial. Alcançou-se o resultado, já que o aplicativo ZETECH foi capaz de controlar e monitorar as informações que passam pelo módulo NodeMCU ESP8266 como: estado de uma iluminação (ligado ou desligado), monitorar a temperatura de um local desejado e recebendo notificações de alertas no seu dispositivo móvel em tempo real de qualquer lugar com acesso à internet. Por fim, concluiu-se que o microcontrolador eletrônico NodeMCU ESP8266 é um mecanismo tecnológico viável, pois foi capaz de manipular diversas tarefas determinadas no aplicativo ZETECH, bem como, verificou-se também que a placa NodeMCU ESP8266 proporciona maior integração dos dispositivos eletrônicos.*

1. Introdução

O ser humano não está conectado apenas a um celular ou computador, mas também a muitos equipamentos como TVs, computadores pessoais, automóveis, *smartphones*, consoles de jogos, *webcams* e a lista aumenta a cada dia.

Busca-se cada vez mais conforto, sofisticação, comodidade e segurança auxiliada pelos computadores e sistemas inteligentes. Novidades tecnológicas que cada vez mais equipam nossas casas, trazendo mais facilidade e possibilidade de realizar tarefas cotidianas.

A partir do momento que foram reunidos os dispositivos tecnológicos sob uma plataforma digital, onde são integrados e gerenciados a distância, se tem a concepção de casa inteligente.

Hoje qualquer residência pode ter sistema de segurança e de iluminação, persianas elétricas, circuito interno de televisão e som, controle de água e gás, todos controlados por dispositivos móveis de qualquer lugar do mundo. Apesar de já ser uma realidade em nossa sociedade, as casas inteligentes são escassas e pouco difundidas, é raro conhecer quem possua uma, ainda que existam produtos e soluções no mercado.

De uma maneira geral, os dispositivos eletroeletrônicos são alinhados de maneira autônoma ou independentes, sem que eles tenham relação um com o outro, o que pode gerar confusão, dificultar a manutenção e o controle da casa.

Os dispositivos que fazem parte das casas inteligentes formam uma rede local onde as comunicações são ativadas por diferentes protocolos. Existem muitas empresas diferentes umas das outras que criam numerosos dispositivos, com diferentes padrões e tecnologias, portanto, existe um grande problema na sua conectividade.

Diante dessa necessidade de um sistema que integre essas tecnologias com a possibilidade de coletar, armazenar e processar operações, com o fim de otimizar o controle das pessoas nos seus afazeres do dia a dia, servindo em tudo o que for necessário, desenvolveu-se um estudo com o objetivo de possibilitar a implementação do aplicativo ZETECH e a prototipação da placa NodeMCU ESP8266 para controlar e monitorar as informações que passam pela placa de qualquer lugar com acesso à internet, simulando a automação de uma residência inteligente.

A presente pesquisa apresenta um estudo teórico e a aplicação prática do controle remoto de uma placa NodeMCU ESP8266 wi-fi, que simula a automação de uma residência inteligente. O dispositivo é gerido por meio de um aplicativo *Android* criado por, este aluno, que controla o estado do led (ligado ou desligado), monitora o sensor de temperatura e um sensor de barreira de qualquer lugar, desde que esteja conectado a rede de internet.

O sistema apresentado nesse trabalho, apesar de conectar apenas três dispositivos diferentes ao módulo (gerenciados pelo aplicativo ZETECH), pode ser a resposta às demandas de uma residência tecnológica que ofereça aos seus habitantes serviços proativos que podem proporcionar conforto, segurança, economia de energia, sustentabilidade e atendimento domiciliar. Além disso, o trabalho pode gerar muitas oportunidades tanto no âmbito acadêmico quanto no industrial.

1.1. Tema/Problema

O tema proposto para o estudo é o desenvolvimento de uma forma de automatizar o controle residencial a fim de facilitar e simplificar as necessidades da vida cotidiana, o que além de criar mais conforto gera comodidade no ambiente doméstico. Pode-se destacar a comodidade pelo fato de, por exemplo, não ser necessário se levantar para monitorar a temperatura, saber se alguém adentrou na casa (sensor de barreira) ou controlar o estado do led (ligado ou desligado). De um único local remoto, pode-se monitorar os diversos dispositivos de automação instalados no sistema que simula uma residência: apagar ou acender luzes, controlar temperatura ambiente, saber se alguém adentrou ou saiu de determinado ponto, etc.

Apesar de a placa NodeMCU ESP8266 wi-fi possuir 11 pinos digitais e um analógico, neste trabalho serão utilizados apenas 2 sensores e um controlador do estado de led. Desta forma, verifica-se que poderiam ter sido utilizados vários outros tipos de sensores devido à ampla capacidade do microcontrolador, entretanto, optou-se por este número por se mostrar mais adequado para às necessidades desta pesquisa.

O problema do estudo consiste em verificar se o microcontrolador NodeMCU ESP8266 wi-fi possibilita maior integração dos dispositivos eletrônicos diante dos avanços tecnológicos.

1.2. Objetivos

O objetivo geral é desenvolver um aplicativo para plataforma *Android* com autenticação e cadastro de usuários para controlar e monitorar as informações que passam pelo módulo NodeMCU ESP8266 de qualquer lugar com acesso a internet.

Os objetivos específicos:

- Desenvolver o código para controlar as portas de entrada e saída no módulo NodeMCU ESP8266;
- Demonstrar sistemas anteriores e seus problemas de interconectividade;
- Criar o aplicativo que controlará e monitorará os objetos conectados a placa NodeMCU ESP8266;
- Usar o *Firebase*, plataforma de desenvolvimento mobile com foco em ser um *back-end*, para autenticação e cadastro de usuários;

1.3. Escopo

Neste estudo será apresentado o sistema microcontrolador NodeMCU ESP8266, que é capaz de gerenciar a automação residencial com facilidade de acesso e custo reduzido.

Toda comunicação de controle será feita com *smartphones* via wi-fi ou 3G, por meio de um aplicativo operacional através da plataforma *Android*, o qual será a forma de comunicação do microcontrolador ESP8266. A parte eletrônica foi projetada para poder ser colocada em uma placa *Protoboard* para fins de apresentação, de forma a ficar pronta para instalação em seus devidos ambientes e que permita atualizações e a resolução de problemas de software via USB.

Segundo Cloud (2020), a autenticação e serviços do *Firebase* estão de acordo com a Lei Geral de Proteção de Dados (LGPD) é a legislação brasileira de privacidade de dados

que rege o processamento de informações pessoais por empresas e organizações estabelecidas no Brasil. A LGPD entrou em vigor no dia 16 de agosto de 2020. A criptografia dos dados em trânsito usa o protocolo HTTPS.

Para essa pesquisa, desenvolveu-se um aplicativo na IDE Android Studio com autenticação de usuário que pode monitorar e controlar, por meio de um Smartphone, sensores, lâmpadas e temperatura. Não haverá uma recuperação de senha caso o usuário venha a esquecê-la, será necessário formalizar, via e-mail, a solicitação do cadastro de uma nova senha. O executável para instalar o aplicativo será disponibilizado para o cliente, assim, o cadastro poderá ser efetuado por qualquer usuário que tenha o aplicativo instalado em seu smartphone, bastará informar e-mail e senha e o cadastro será efetuado.

1.4. Viabilidade do projeto

Em uma observação preliminar, foi possível coletar dados para financiamento de recursos e materiais para construção do circuito, levantamento bibliográfico e do tempo disponível para realização do trabalho.

Apesar de o trabalho contar com poucos integrantes, o aluno e seu orientador, a pesquisa não necessita de uma demanda grande de pessoas para produzi-la, pois seu custo se tornou acessível ao ser utilizado softwares livres e instrumentos (placa NodeMCU ESP8266, sensores e relés de acionamento da lâmpada).

1.5. Estrutura do trabalho

A fim de atingir os objetivos propostos para a realização deste projeto, sua estrutura está composta por fundamentação teórica, comprovação prática e a análise de resultados. Todo fundamento teórico colocado nas partes iniciais serve como alicerces para comprovação da parte prática.

O trabalho conta com 7 capítulos. O primeiro capítulo aborda a introdução, o tema, o problema, os objetivos, o escopo, a viabilidade do projeto e a organização da pesquisa, ou seja, como será organizado o trabalho para responder ao problema da pesquisa, o conceito de Casa Inteligente e Controle Remoto. O capítulo dois versa sobre o fundamento teórico que servirá de alicerce para a estrutura de todo trabalho. No capítulo três são relatados trabalhos existentes, apresentando três diferentes projetos utilizando o ESP8266. Para o capítulo quatro serão apresentados os requisitos necessários no desenvolvimento do sistema e exposto o projeto de interface. Já no capítulo cinco é descrito a implementação do trabalho. Por fim, os dois últimos capítulos são a análise prática dos resultados obtidos do experimento e conclusão. A fim de atingir os objetivos propostos para a realização deste projeto, sua estrutura está composta por fundamentação teórica, comprovação prática e a análise de resultados. Todo fundamento teórico colocado nas partes iniciais serve como alicerces para comprovação da parte prática.

1.6. Casa Inteligente

Antes de conceituar uma casa inteligente, faz-se necessário trazer uma noção do que seria um agente inteligente, Russel et al. (2013) nos esclarece esse conceito dizendo que é tudo aquilo que pode perceber o ambiente em que se encontra, através de sensores e agir sobre este ambiente por meio de atuadores, assim como uma pessoa dispõe dos seus “sensores” como: olhos, ouvidos, tato, língua, nariz, e atuadores como mãos e pernas, um agente robô

poderia ter sensores, como câmeras, sensores infravermelhos e atuadores, como diversos tipos de motores.

Em 1935 uma casa inteligente era aquela onde cada cômodo possuía ponto de eletricidade, em 1955 eram as que possuíam televisão e telefone, em 1996 casa inteligente era aquela que possui um computador por cômodo (Weiser, 1996), já hoje, casa inteligente faz referência a uma casa automatizada, com sistemas integrados de forma cooperativa, envolvendo outros conceitos além dos tecnológicos, para gerar um ambiente seguro, limpo e saudável, sem deixar de lado a eficiência energética (Angeloni, 2016).

Outro termo interessante ligado ao assunto casas inteligentes é Domótica. O termo domótica é uma fusão da palavra “*Domus*” (casa) com “Telemática”, eletrônica mais informática, (Pereira, 2015). A domótica também é referenciada por “*Smart house*”, ou casas inteligentes. Quando a domótica surgiu, a ideia principal era controlar iluminação, condições climáticas e segurança, hoje em dia, a ideia é praticamente a mesma só que existem muito mais dispositivos que podem ser integrados a domótica (Palma, 2008).

Ao depararmos com a tecnologia da informação e sensores que podem ser implantados em nossas casas, tornando-as inteligentes, vê-se a possibilidade destes dispositivos auxiliarem pessoas com deficiência e idosos, em suas reduzidas funções físicas, em questões de segurança, na prevenção de doenças e na mobilidade. Além disso, podem ajudar também as pessoas a enfrentar o isolamento social, pois são capazes de fornecer assistência sem limitar ou perturbar o cotidiano do morador, dando-lhe maior conforto, prazer e bem-estar (Chan et al., 2008).

O que será que o futuro trará a humanidade nas casas inteligentes? Avanços tecnológicos recentes nas áreas de mobilidade global, tecnologias sem fio, miniaturização redes diferentes, informática em saúde em casa, televisões avançadas, redes de sensores, Sistemas MIMO (do inglês: *Multiple-input and multiple-output*, MIMO) e experimentos de recepção conduzidos por computação quântica abrem um novo horizonte para as tecnologias do futuro.

No futuro, os computadores e máquinas habitarão as casas conosco nos ajudando em coisas triviais como: roupas etiquetas (para rastrear a lavagem), xícaras de café (para alertar a equipe de limpeza para copos mofados), luz e interruptores (para economizar energia se ninguém estiver na sala) e lápis (para digitalizar tudo o que é desenhado). Como diz Weiser (1996), não será apenas uma interação entre computadores, máquinas e humanos, mas sim uma habitação, onde divide-se o mesmo espaço de convívio com natureza de companheiros. Isso significa que eles têm seu lugar, e nós o nosso, para que todos coexistem de maneira confortável.

1.7. Controle Remoto

O controle por acesso remoto do aplicativo ZETECH representa a informatização, interação, facilidade e conforto, além de transferir dados e informações sem a utilização de cabos e encurtar distâncias.

Quando se fala em controlar esse aplicativo de maneira remota, trata-se na verdade de se utilizar um aparelho para realizar uma operação remota em um dispositivo eletrônico. Por meio de um software hospedado na nuvem, o acesso é garantido com segurança, a partir de qualquer lugar e horário, somente por pessoal devidamente autori-

zado, isto é, o cadastro de logins e senhas é necessário para acessar a conta.

Como o acesso pode ser feito pela internet, basta ter um dispositivo móvel (como por exemplo um *smartphone*) em mãos e um ponto de conexão com a internet para poder verificar, incluir, alterar e confirmar dados de comando ao aplicativo.

Existem muitos benefícios de se aplicar uma tecnologia de acesso remoto a uma casa inteligente, alguns são mais evidentes, dentre eles se destacam o conforto, a segurança, a gestão energética e as comunicações. Na Figura 1, apresenta-se um mosaico que centraliza o controle remoto da casa inteligente com seus quatro eixos de sustentação.

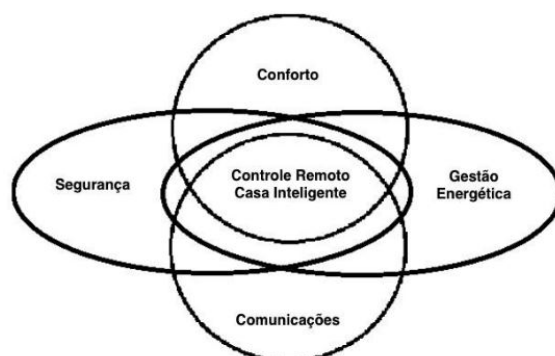


Figura 1. Mosaico dos benefícios do controle remoto. Fonte: Autor, 2020.

- **Segurança:** Podendo abranger desde um alarme de intrusão simples, passando por alarmes de incêndio, de fuga de gás, de inundação até grandes sistemas monitorados por computadores e circuito interno de TV.
- **Comunicações:** são consideradas as comunicações internas com o exterior e vice-versa, com o objetivo de fazer uma integração entre dispositivos internos e externos de forma mais eficiente e rápida, podendo ter acionamento remoto ou apenas avisos de crise por comunicação com dispositivos.
- **Gestão energética:** é possível controlar e melhorar o racionamento energético, aumentando a eficiência energética da casa desligando remotamente sistemas e aparelhos quando não estiverem em uso. Além dos produtos de automação residencial padrão que fornecem controle ativo, alguns produtos monitoram ativamente os sistemas e fornecem ao proprietário conhecimento, visão e orientação para obter maior controle e eficiência energética.
- **Conforto:** Automação de funções domésticas de rotina, controle de iluminação, regulação automática de temperatura, descentralização de equipamentos e centralização de controle, áudio e vídeo e controle a distância.

Neste trabalho, o aplicativo ZETECH controla apenas o estado de um led (ligado ou desligado), um sensor de temperatura e um sensor de barreira, entretanto, as possibilidades são muito abrangentes, já que o software pode receber melhoramentos e atualizações, conforme a necessidade.

2. Materiais e Métodos

Esse ponto descreve e explica a origem e utilização dos materiais empregados no experimento, também serão demonstrados os procedimentos metodológicos, conceitos, ideias e informações relevantes relacionados ao tema.

2.1. Procedimentos Metodológicos

O desenvolvimento do trabalho iniciou-se através de uma questão de pesquisa (como controlar e monitorar informações de aparelhos eletrônicos de forma remota e como por isso em prática?). Para implementar uma solução para esse problema houve a necessidade de se observar outras pesquisas, as quais abordavam o assunto e chegou-se à conclusão de que era preciso um hardware (módulo NodeMCUESP8266) e um software (ZETECH).

O início começou pela representação do todo a ser construído e, gradualmente, focou-se nos detalhes com uma abordagem sistemática, ou seja, ao mesmo tempo que se observa a individualidade das partes, considera-se também o todo.

Para se atingir os objetivos deste trabalho, a ação de engenharia de software se fez necessária para a organização e otimização das tarefas, definindo o procedimento necessário para a sua realização.

Segundo, Pressman (2011) a observância de um modelo de processo para atividade de criação de um software, é trazer ordem ao caos existente na área de desenvolvimento de software. Nesse sentido, é importante seguir um roteiro, já que a implementação de um software geralmente é feita por equipes de engenheiros de software. Com os modelos de processo de software, é possível seguir uma estrutura organizada, coordenada e coerente, para que o projeto não implique em uma desordem.

O modelo escolhido para o desenvolvimento e implementação do sistema foi o incremental, pois de acordo com Cordeiro (2003), cada versão desse processo é capaz de ser acrescida de funcionalidades que podem ser modificadas para melhor satisfazer as necessidades dos clientes/usuários. As novas versões podem ser planejadas de modo que os riscos técnicos possam ser administrados (ex. disponibilidade de determinado hardware). Esse modelo também é mais apropriado para sistemas pequenos, como o ZETECH mostrado na Figura 2 abaixo.

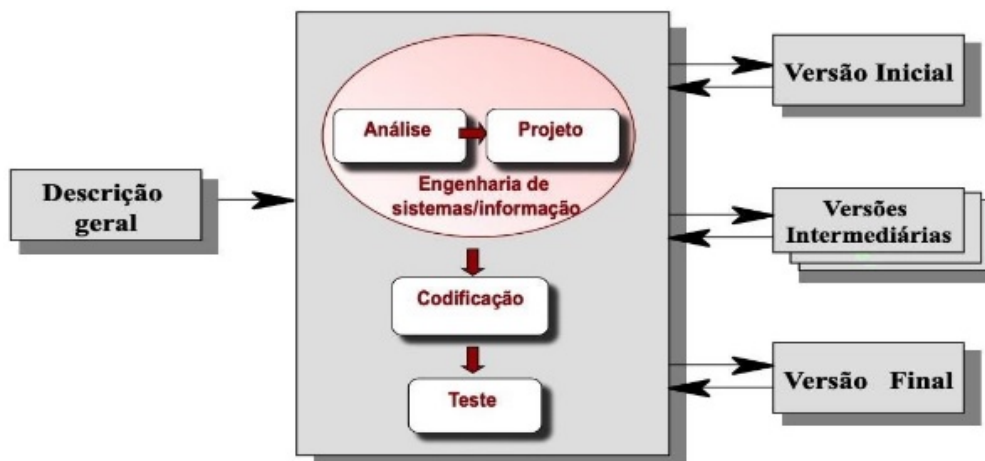


Figura 2. Modelo Incremental. Fonte: Nakagawa, 2017.

2.2. Padrão de projeto MVC

Segundo Medeiros (2013), o padrão arquitetural *Model—View—Controller* (MVC) é uma forma de quebrar uma aplicação, ou até mesmo, um pedaço da interface de uma aplicação, em três partes: o modelo, a visão e o controlador.

O aplicativo possui a arquitetura em camadas, está dividido em *back-end* e *front-end*, assim foi utilizado o padrão MVC (*Model View Controller*), conforme a Figura 3 que ilustra seu funcionamento.

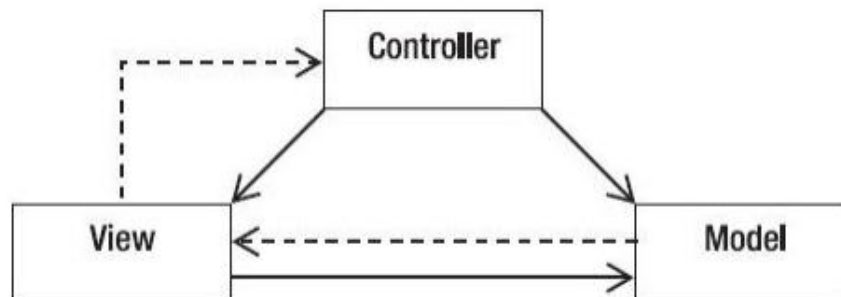


Figura 3. Interações utilizadas no projeto MVC. Fonte: Medeiros, 2013.

2.3. NodeMCU ESP8266

De acordo com Murta (2018), o módulo NodeMCU ESP8266 é uma placa que foi criada para facilitar o desenvolvimento de aplicações para o módulo ESP8266. Existe um módulo ESP-12 soldado na placa contendo todos os circuitos necessários para fazê-lo funcionar como a interface Serial-USB, regulador de tensão, leds indicadores, botões de controle (*Reset* e *Flash*) e barramentos de pinos para permitir o uso em *Protoboards*. O NodeMCU ESP8266 funciona como se fosse um Arduino, onde conecta-se o cabo USB e com a IDE Arduino carregando os programas na placa.

Murta (2018), ainda afirma que todos os pinos GPIOs (*General Purpose Input/Output* ou Portas Programáveis de Entrada e Saída), podem ser entradas ou saídas dependendo da configuração dos mesmos, não podendo ultrapassar a corrente de 12 mA em cada porta, pois poderá danificar o chip, conforme Figura 4.

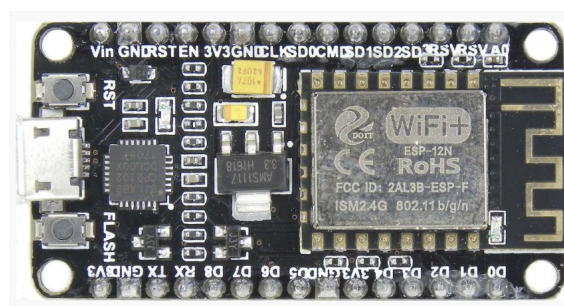


Figura 4. NodeMCU ESP8266. Fonte: Murta, 2018.

2.4. DHT11

Gay (2018) explica que o DHT11 é um sensor de temperatura e umidade que permite medir temperaturas de 0 a 50 graus Celsius e umidade na faixa de 0% a 90%, além de apresentar um controlador de 8 bits para a conversão dos dados captados em um sinal serial que pode, por sua vez, ser enviado para um microcontrolador através de seu pino de dados. Conforme a Figura 5.

O DHT11 possui um sensor de temperatura e umidade complexo com uma saída de sinal digital calibrado. Usando a aquisição exclusiva de sinal digital, técnica e tecnologia de detecção de temperatura e umidade, garante alta confiabilidade e excelente estabilidade a longo prazo (Electronics, 2018).

Percebe-se que o DHT11 tem excelentes qualidades o que o torna um componente bastante utilizado sendo então achado com facilidade em lojas de componentes eletrônicos tendo um melhor custo benefício.

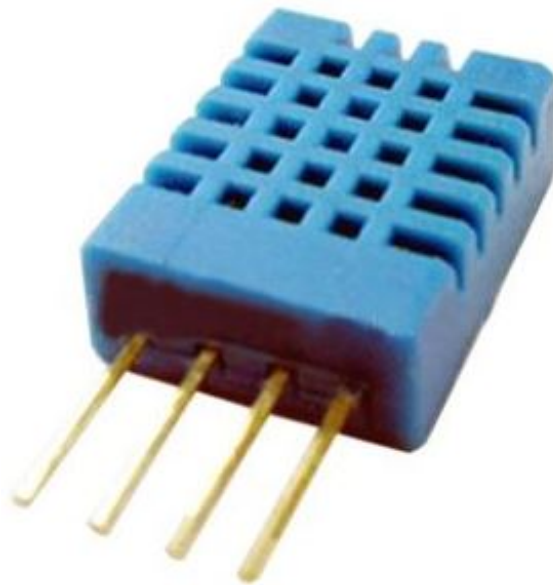


Figura 5. DHT11. Fonte: Electronics, 2018.

2.5. Sensor Obstáculo

O sensor de obstáculo detecta a luz refletida proveniente do seu próprio LED infravermelho (TechnoLabs, 2017). Esse LED está integrado na placa e é usado para indicar visualmente a presença de um objeto. Sua faixa de detecção é ajustável com um resistor variável embutido. Ele possui também um conector com três pinos, podendo ser conectado a um *protoboard* ou placa Arduino. De acordo com RoboCore (2020), o alcance desse sensor é excelente, sendo capaz de “sentir” obstáculos em uma distância de 2 a 30 cm.

Sensor de obstáculo pode ser usado como proteção residencial em pontos de entrada da casa como, por exemplo, porta ou janela. Ele monitora cada alteração de status no sensor, ou seja, caso alguém abra uma porta ou janela, o sensor irá acionar. Esse sensor é de curto alcance o que não o torna conveniente utilizar como sensor de presença para monitorar a área de um ambiente, na Figura 6 mostra o módulo completo.



Figura 6. Sensor de Obstáculos. Fonte: TechnoLabs, 2017.

2.6. Firebase

O Firebase é uma plataforma que contém várias ferramentas e uma excelente infraestrutura para ajudar desenvolvedores *web* e *mobile* a criar aplicações de alta qualidade e performance (Khawas and Shah, 2018). Essa plataforma ainda é capaz de armazenar os dados no formato *JavaScript Object Notation* (JSON), não usa consulta para inserir, atualizar, excluir ou adicionar dados a ele. É o *back-end* de um sistema usado como banco de dados para armazenar dados.

Khawas and Shah (2018) explica que o *Firebase Auth* é um serviço que pode autenticar usuários usando apenas código do lado do cliente. Ele também inclui um sistema de gerenciamento de usuários, no qual os desenvolvedores podem habilitar a autenticação do usuário com login por e-mail e senha armazenados no *Firebase*.

O *Realtime Database* é um banco de dados hospedado na nuvem oferecido pelo *Firebase*, cujos dados são armazenados como JSON e sincronizados em tempo real com todos os clientes conectados. Quando se cria aplicativos em plataformas cruzadas como SDK (Software Development Kit) para iOS, Android e JavaScript, todos os clientes compartilham uma instância do Realtime Database e recebem automaticamente atualizações com os dados mais recentes, ou seja, essa ferramenta gera uma maior integração com aplicativos *Android*, *IOS* e *JavaScript*. A integração e programação do app com *Android Studio* foi facilitada com a utilização da plataforma *Firebase*, pois levou-se em consideração o seu funcionamento em vários dispositivos, infraestrutura, atualizações e transferência de dados (Khawas and Shah, 2018).

Para essa pesquisa, aplicou-se o sistema *Firebase Auth* para autenticação, bem como o *Real-Time Database* (do inglês Banco de dados em tempo real) para armazenar toda alteração de status ocorrida nas entradas e saídas digitais ou analógicas da casa. Na Figura 7, apresentam-se as quatro categorias fornecidas pela plataforma *Firebase*.



Figura 7. Firebase — Categorias. Fonte: Orlandi, 2018.

2.7. Ide Arduíno

Segundo Banzi and Shiloh (2014), o Arduino é composto por duas partes principais, a placa, que é o hardware, e a IDE Arduino, que é o Software onde se escreverá o que se quer que a placa faça. Usa-se o IDE para criar o *sketch* (pequeno programa computacional) que carrega a placa do Arduino, o *sketch* diz para a placa o que deve ser feito.

O ESP8266 é um módulo onde a programação pode ser feita em duas linguagens LUA e C/C++. Utilizou-se a IDE Arduino programando em C/C++, sendo necessário baixar bibliotecas para o funcionamento correto do ESP8266.

2.8. Android Studio

Segundo Tsang and Chung (2019), o *Android Studio* é um sistema operacional para a maioria dos dispositivos de *smartphones*. Além disso, é uma plataforma para o desenvolvedor elaborar aplicativos, onde pode criar e depurar sem um dispositivo *Android*, pois existe um simulador/emulador de *smartphones* para a depuração.

Ja Cordeiro (2018), afirma que os aplicativos *Android* podem ser escritos com as linguagens Kotlin, Java e C++ usando o *Android SDK*.

Dentre várias ferramentas pesquisadas para criar o aplicativo, chegou-se a conclusão de usar o *Android Studio*, por ser um programa mundialmente utilizado para o desenvolvimento de aplicativos e de inúmeras soluções possuindo compatibilidade integrada no *Firebase*.

Ao se instalar o *Android Studio* efetuou-se *login* com a conta no Google para ter disponibilidade das ferramentas fornecida. O programador consegue tirar dúvidas no uso das ferramentas do *Firebase*, na própria IDE, pois há uma aba que o ajuda informando passo a passo de como usar.

2.9. Linguagens de programação

A linguagem de programação é uma estrutura padronizada para criar comandos para um computador ou um conjunto de regras que definem um programa de computador (Cor-

deiro, 2018). Elas formam o código fonte de um software ou aplicativo, que depois será traduzido para código da máquina, que é executado pelo microprocessador.

No desenvolvimento dos códigos fontes foram utilizados dois tipos de linguagens de alto nível, Java e C++. Segundo Junior et al. (2012), essas duas linguagens de programação são mais populares para programação orientada a objetos. Elas possuem um mecanismo classe/objeto que permite herança simples e herança múltipla e sobrecarga de operadores e funções.

De acordo com Junior et al. (2012) a linguagem Java baseou-se no C++, entretanto, foi especificamente projetada para ser menor e mais simples não tendo ponteiros, mas sim tipo de referência, a qual não é possível escrever subprogramas independentes. A linguagem Java possui somente herança simples, interfaces, coleta de lixo implícita e é utilizada para programação web.

2.10. GitHub

Neste trabalho, o Github foi utilizado como um serviço web que hospeda o projeto. O GitHub é um serviço de hospedagem na Web para repositórios Git (<https://github.com/about>). Ele implementa uma rede social na qual os desenvolvedores podem transmitir suas atividades para outros interessados e se inscreverem nela (Thung et al., 2013).

De acordo com Weitzel et al. (2017) usando o GitHub os programadores podem interagir e colaborar em torno de repositórios de código aberto, o que permite que eles façam download, cooperem, compartilhem além de outras funcionalidades.

Quando o projeto está hospedado no github, é possível que todos os envolvidos ou interessados no projeto tenham um acesso mais fácil e trabalhem de forma centralizada e organizada.

O controle e integração dos fluxos de trabalho, como o código fonte da placa ESP8266 e o aplicativo, foram realizados com a tecnologia Git que facilita essa associação e armazenado em repositório online. Ao terminar uma versão ou atualização, as implementações são enviadas para o repositório, onde o desenvolvedor tem acesso aos históricos de alterações.

O GitHub proporcionou diversas ferramentas para um melhor controle do projeto, como por exemplo quais usuários alteraram, o que foi alterando, quando foi alterado e possibilita aos usuários relatarem problemas.

3. Trabalhos Correlatos Existentes

Como já mencionado acima, esse trabalho busca desenvolver um estudo que possibilite a implementação do aplicativo ZETECH e a prototipação da placa NodeMCU ESP8266 para facilitar a interconectividade de objetos residenciais e simplificar tarefas domésticas.

Para o desenvolvimento dessa pesquisa, foi necessário realizar um levantamento com a finalidade de identificar os trabalhos acadêmicos relacionados ao ESP8266, Internet das Coisas e automação residencial. Abaixo estão listados três trabalhos correlatos com uma breve descrição de suas principais características:

- Uso do microcontrolador ESP8266 para automação residencial: apresenta-se, neste trabalho como implementar um sistema de lâmpadas controladas através do

celular, utilizando-se para isso o microcontrolador ESP8266. O objetivo foi verificar a possibilidade de realização de projetos de automação residencial com o uso desse controlador para diversos fins. Para tal, foi desenvolvido um circuito, um software implementado no ESP8266 e uma página web para interface celular—ESP8266, (Oliveira, 2017).

- Sistema de automação residencial de baixo custo utilizando o ESP8266: este trabalho teve objetivo de desenvolver um sistema de automação residencial (domótica) sem fio, utilizando o microcontrolador embarcado de baixo custo ESP8266, a linguagem de marcação HTML. Utilizando-se da wi-fi, característica do próprio microcontrolador, elaborou-se um software para interface com o usuário IHM. Realizaram-se estudos de circuitos eletrônicos para alimentação do sistema e acionamento da carga para desenvolver um protótipo, (da Silva, 2017).
- Automação residencial para monitoramento de temperatura, umidade e controle de iluminação utilizando ESP8266: este trabalho teve como objetivo desenvolver uma automação residencial que permita controlar remotamente diversos dispositivos encontrados na maioria das residências, permitindo assim o acionamento de lâmpadas, tomadas elétricas, fitas de LED's e ainda monitorar a umidade e temperatura do ambiente, (Santos, 2016).

Verificou-se que as pesquisas acima utilizam somente o microcontrolador ESP8266 sem integração com o NodeMCU, o que acaba por criar um circuito a parte com conversor de serial e regulador de tensão para poder compilar o código fonte via USB. Esta pesquisa se utiliza de um NodeMCU ESP8266 que conta com conversor serial, regulador de tensão próprio e o microcontrolador ESP8266, tudo em uma só placa e com pinos para uso no *proto board* facilitando a montagem do circuito para simular uma casa inteligente.

4. Desenvolvimento do Home Zeus

Este capítulo apresentará os requisitos necessários para o desenvolvimento do sistema, divididos em requisitos funcionais e requisitos não funcionais, também serão apresentados os diagramas como caso de uso, diagrama de atividades e diagrama de classes. Na sequência, será exposto o projeto de interface com as ilustrações das telas do aplicativo ZETECH criadas para o protótipo montado em um *proto board*.

4.1. Requisitos funcionais

De acordo com Sommerville (2011), características funcionais, que representam os comportamentos que um programa ou sistema deve apresentar diante de certas ações de seus usuários.

RF001 - O sistema deve permitir acesso somente a usuários autenticados e ativos.

RF002 - O sistema deve permitir cadastrar um novo usuário.

RF003 - O sistema deve permitir usuários logados terem o controle de todas as funcionalidades do sistema.

RF004 - O sistema só deve permitir acesso com autenticação de usuário.

4.2. Requisitos não funcionais

De acordo com Shah and Patel (2016), características não funcionais, que quantificam determinados aspectos do comportamento.

RNF001 - O sistema deve rodar em *Android*.

RNF002 - O sistema deve usar padrão firebase para comunicar com a placa ESP8266.

RNF003 - O sistema deve ser implementado utilizando as linguagens Java e C.

RNF004 - O sistema deve atualizar a temperatura a cada cinco minutos.

4.3. Diagramas UML

De acordo com Guedes (2018), a UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada), é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. A utilização da UML será importante pois, auxiliará na visualização dos desenhos e figuras, bem como a comunicação entre os objetos e em certos casos a identificação dos processos.

4.4. Caso de uso

Os casos de uso do sistema estão representados na Figura 8 e estão de acordo com os requisitos funcionais.

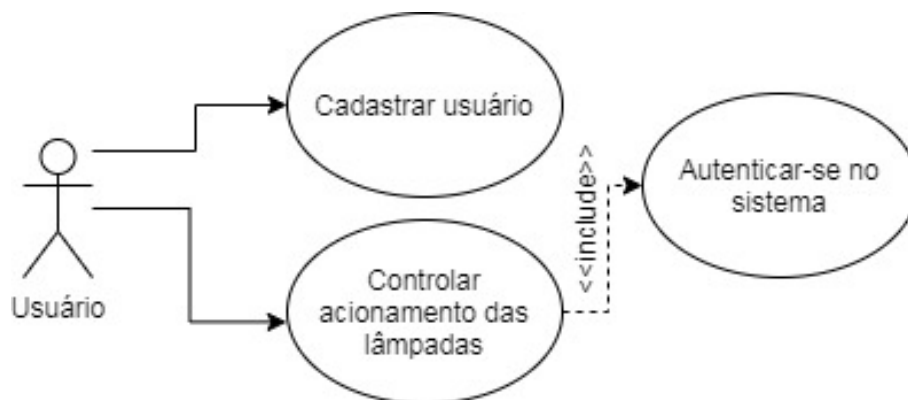


Figura 8. Diagrama caso de uso. Fonte: Elaborado pelo autor, 2020

Nome do caso de uso: Verifica autenticação

Autor: *Firebase*.

Objetivos: Verifica se usuário informado nos campos estão cadastrados no banco de dados.

Requisito: RF001.

Fluxo normal:

1. O sistema deve estar na tela de *login*.
2. O usuário preenche os campos e clica em entrar.

Fluxo alternativo:

1. O sistema reconhece um erro ao entrar e exibe o erro.
2. O usuário verifica o erro e corrige e tenta novamente.
3. O sistema entra e vai para a tela de controle.

Nome do caso de uso: Cadastrar usuário

Autor: Qualquer usuário.

Objetivos: O usuário deve acessar a tela *login* e deve selecionar a opção registrar, preencher os campos e clicar em cadastrar.

Requisito: RF002.

Fluxo normal:

1. O sistema deve estar na tela de *login*.
2. O usuário deve selecionar a opção registrar.
3. O sistema exibe uma lista com campos a serem preenchidos.
4. O usuário preenche os campos e clica em cadastrar.
5. O sistema salva o cadastro e volta para tela de *login*.

Fluxo alternativo:

1. O sistema reconhece um erro ao finalizar cadastro e exibe o erro.
2. O usuário corrige o erro, e continua o processo de cadastro.
3. O sistema salva o cadastro e volta para tela de *login*.

Nome do caso de uso: Controlar acionamentos das lâmpadas

Autor: Todos usuários.

Pré-condição: Os usuários devem ter feito seu *login* e o sistema deve ter pelo menos uma lâmpada registrada.

Objetivos: O sistema deve ligar ou desligar a lâmpada.

Requisito: RF003.

Fluxo normal:

1. O usuário deve estar autenticado.
2. O usuário deve selecionar o botão para alterar o status.
3. A o botão irá alterar o ícone de ligado para desligado, ou vice-versa.

Fluxo alternativo:

1. O usuário deve selecionar o botão.
2. Se não há alteração no status do botão e a lâmpada não ligar, um erro de conexão irá aparecer.

Nome do caso de uso: Autenticação de usuários.

Autor: Todos usuários.

Pré-condição: Os usuários devem estar cadastrados.

Objetivos: O sistema permitirá acesso a tela de controle das lâmpadas.

Requisito: RF004.

Fluxo normal:

1. O sistema deve estar na tela de *login*.
2. O usuário preenche os campos e clica em entrar.
3. O sistema entra e vai para a tela de controle.

Fluxo alternativo:

1. O sistema reconhece um erro ao entrar e exibe o erro.
2. O usuário verifica o erro e corrige e tenta novamente.
3. O sistema entra e vai para a tela de controle.

4.5. Diagrama de atividades

O diagrama de atividades, no contexto da UML, permite representar as operações e consultas de um sistema de acordo com uma sequência de eventos em um determinado caso de uso, representando qual o fluxo de dados e informações entre atores, aplicação e o controlador (Wazlawick, 2014). Portanto, ele ilustra graficamente como será o funcionamento do software, como será a execução de alguma de suas partes, como será a atuação do sistema na realidade de negócio na qual ele está inserido

Abaixo, Figura 9 será apresentado um quadro que representa as principais atividades do sistema, alinhadas de acordo com os requisitos funcionais (cadastro de usuário, controle do acionamento das lâmpadas e autenticação de usuários).

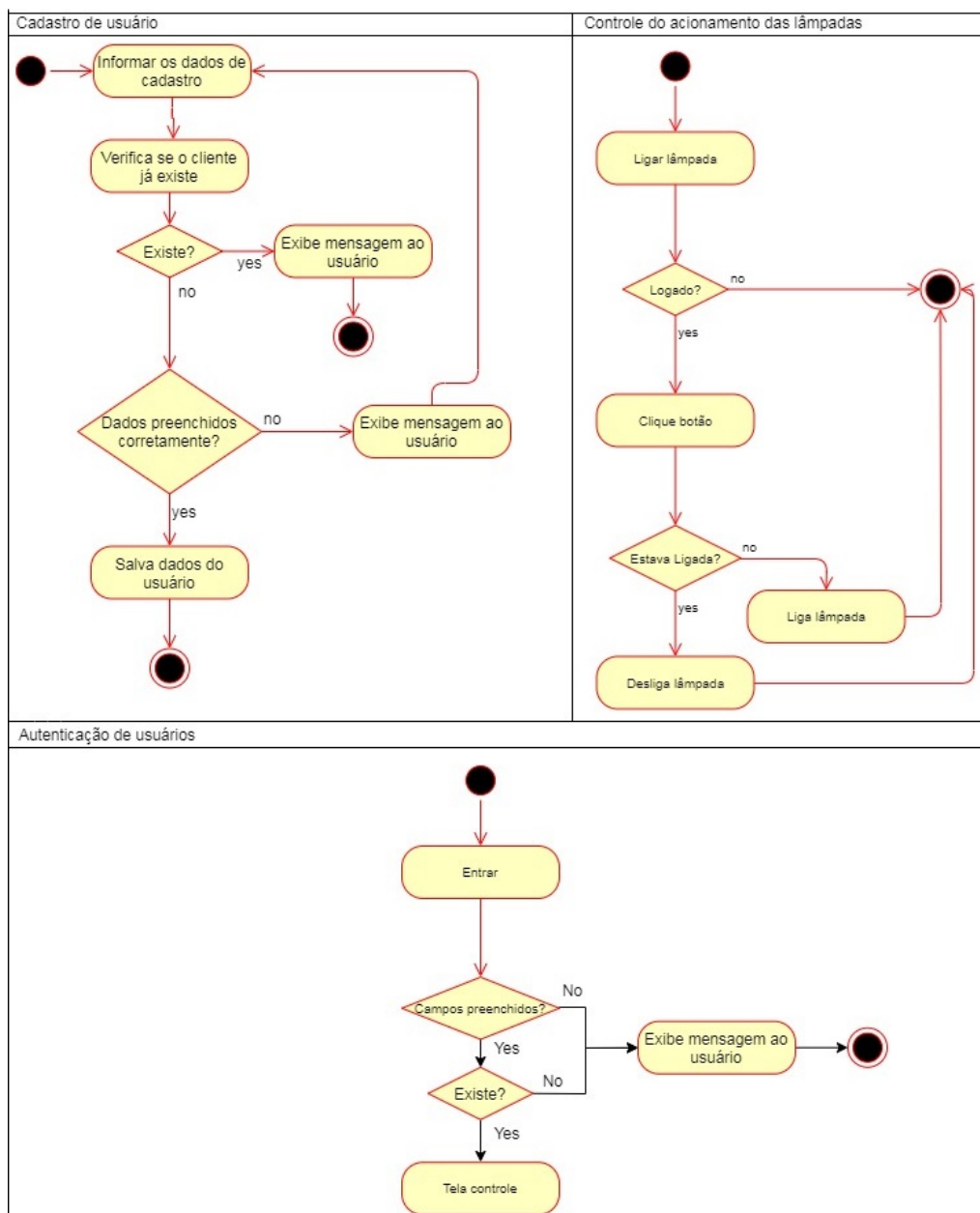


Figura 9. Diagrama de atividades. Fonte: Elaborado pelo autor, 2019

- **Cadastro de usuário:** Para cadastrar um novo usuário é verificado se o mesmo já está cadastrado, se não, após preenchido corretamente os dados na tela, o cadastro será efetuado.
- **Controle do acionamento das lâmpadas:** O sistema só acessa a tela de controle das lâmpadas se o usuário estiver autenticado.
- **Autenticação de usuários:** Para efetuar a autenticação do usuário, ele deve estar cadastrado e preencher os campos corretamente.

4.6. Diagrama de Classes

O diagrama de classes define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada uma tem, além de estabelecer como as elas se relacionam e trocam informações entre si, (Guedes, 2018).

Seu objetivo é representar a estrutura dos objetos que o sistema irá manipular. Para facilitar, a IDE *Android Studio* possui um *plugin* que gera automaticamente o diagrama.

A Figura 10 mostra todas as classes deste trabalho e como as elas são de simples construção. O controle do banco de dados é efetuado pelo *Firebase*, não houve relação entre as classes.



Figura 10. Diagrama de Classes. Fonte: Elaborado pelo autor, 2020

4.7. Projeto da Interface

A interface gráfica de uma aplicação é a camada interativa por meio da qual um usuário executa ações que serão refletidas num sistema computacional (Medeiros, 2019).

As telas foram criadas com *layout* simples e intuitivo, contendo pouca variação de cores e botões com ícones de cada função, ou seja, se o botão for para controlar a luz, ele possui um ícone de lâmpada acesa ou apagada variando de acordo com o status enviado.

A seguir, na Figura 11, apresenta-se a tela em que o usuário deve informar seu e-mail e senha para acesso ao ecrã de controle. Caso o usuário não seja cadastrado, deve-se clicar em “Não é registrado? Registrar” para efetuar o cadastro.

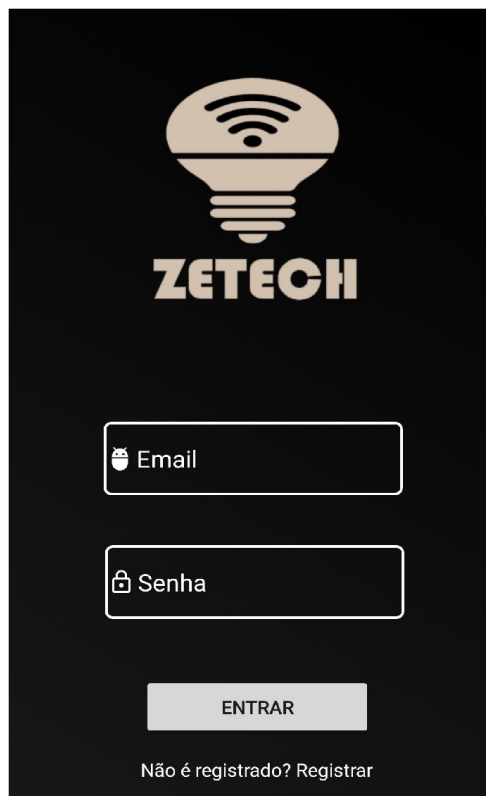


Figura 11. Tela login. Fonte: Elaborado pelo autor, 2020

A tela de cadastro é apresentada na Figura 12, usuário deve informar e-mail e senha para cadastrar.

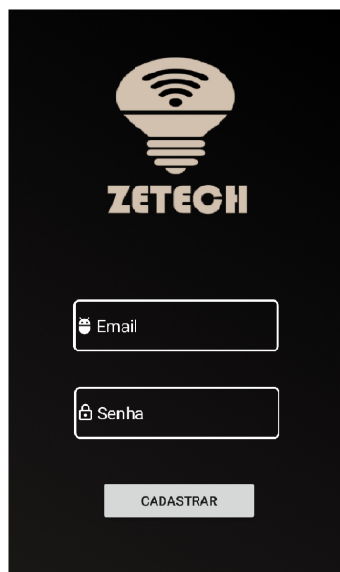


Figura 12. Tela cadastro. Fonte: Elaborado pelo autor, 2019

Após efetuado o login, o usuário terá acesso a tela de controle, apresentado na Figura 13, cuja ecrã demonstra que o usuário terá o controle da lâmpada e monitoramento da temperatura.

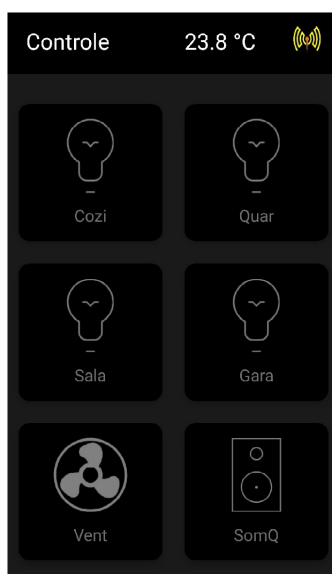


Figura 13. Tela controle. Fonte: Elaborado pelo autor, 2020

Caso o usuário tenha efetuado a autenticação errada, duas mensagens irão aparecer, como mostra a Figura 14.

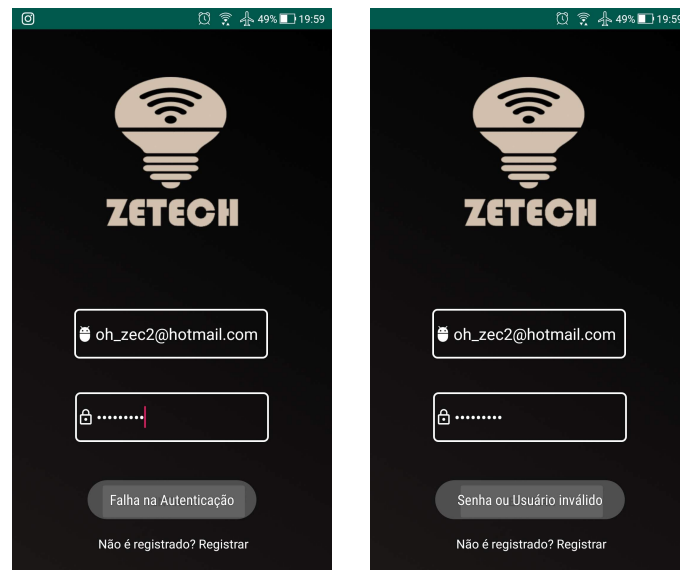


Figura 14. Tela login. Fonte: Elaborado pelo autor, 2020

A Figura 15 mostra a mensagem de boas vindas após o usuário voltar a usar o aplicativo.

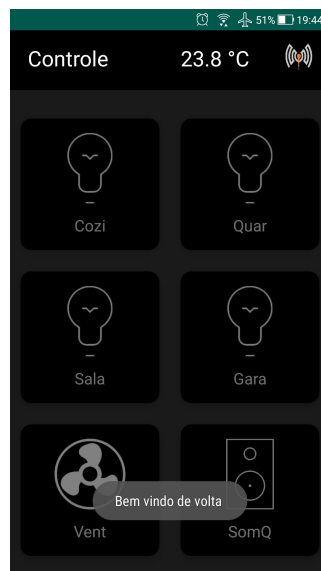


Figura 15. Tela controle. Fonte: Elaborado pelo autor, 2019

A Figura 16 demonstra que o aplicativo se encontra logado, mas sem conexão com a placa NodeMCU ESP8266, devido há alguma perda de conexão a internet, tanto para o usuário ou para a placa. Além disso, é possível notar abaixo que o ícone de antena no canto superior direito está na cor cinza, isso também informa que o aplicativo está desconectado.



Figura 16. Tela controle. Fonte: Elaborado pelo autor, 2020

A conexão entre aplicativo e placa está em perfeito funcionamento quando o ícone de antena está na cor amarela, conforme Figura 17.



Figura 17. Tela controle. Fonte: Elaborado pelo autor, 2020

5. Implementação

A fim de facilitar a implementação do software, o código-fonte foi desenvolvido primeiramente no módulo NodeMCU ESP8266 e o hardware testado diretamente no Firebase para posteriormente desenvolver o código no aplicativo. Em sequência, elaboraram-se os requisitos funcionais e não funcionais, os diagramas UML, os diagramas de atividades e de classes e por fim o projeto de interface. O código-fonte do sistema desenvolvido pode ser visto em <https://github.com/ZeusSchmitz/AndroidStudio> do aplicativo e para o código-fonte da placa ESP em <https://github.com/ZeusSchmitz/Arduino/> no arquivo `led_dht11 div`.

Pretende-se dizer que, a medida que o projeto vai sendo desenvolvido, a arte progride com sucessivos refinamentos, melhorados a cada iteração, havendo assim, sempre que necessário, o refazer de um passo para que se consiga chegar ao modelo desejado. Esta metodologia permite que se vá refinando o modelo que se está a desenvolver, de modo a que o produto final do projeto seja conforme os requisitos estabelecidos (Pressman, 2011).

No caso do protótipo de gerenciamento de casa, Pressman (2011) afirma que a primeira versão é um produto essencial, a parte mais importante, isto porque foram atendidos os requisitos básicos para que o sistema possa funcionar e atender minimamente as necessidades do cliente. Esse produto essencial utilizado pelo cliente está sujeito a modificações para melhor se adequar as necessidades do cliente e a entrega de recursos e funcionalidades adicionais.

A evolução acontece quando novas características são adicionadas a medida que são sugeridas pelo usuário. Como por exemplo, o aplicativo ZETEC pode futuramente sofrer incrementos que o habilita a gerenciar sistemas de alarme, acionar por comando de voz eletrodomésticos, ler de sensores digitais e analógicos.

6. Resultados

O resultado principal foi alcançado, obtendo-se êxito no desenvolvimento do aplicativo ZETECH. Esse aplicativo, é capaz de controlar e monitorar as informações que passam pelo módulo NodeMCU ESP8266 de qualquer lugar com acesso a internet, sendo possível também acompanhar e controlar no *smartphone* ou *tablet* toda a automação da casa, assim como estado de uma iluminação (ligado ou desligado), o monitoramento da temperatura de um local desejado e receber notificações de alertas em tempo real. Dessa forma, esse sistema se apresenta como um aplicativo inteligente que fornece conforto, segurança e informação.

Durante a execução da pesquisa, atingiram-se os objetivos específicos, como o desenvolvimento do código que a controla. Demonstrou-se também a evolução tecnológica dos sistemas e placas que gerenciam equipamentos eletrônicos, o quanto esse avanço tornaram os sistemas anteriores obsoletos. Conseguiu-se utilizar o Firebase, plataforma de desenvolvimento mobile com foco em ser um *back-end*, para autenticação e cadastro de usuários.

Com o aplicativo devidamente conectado e trocando informações com a placa, demonstrou-se, Figura 18 abaixo, que é possível ver o circuito completo com o sensor de obstáculo, de temperatura e o led que será ligado quando o usuário clicar no botão “Cozi”.

A foto da direita é possível verificar que o botão “Cozi” foi acionado, ligando o led.

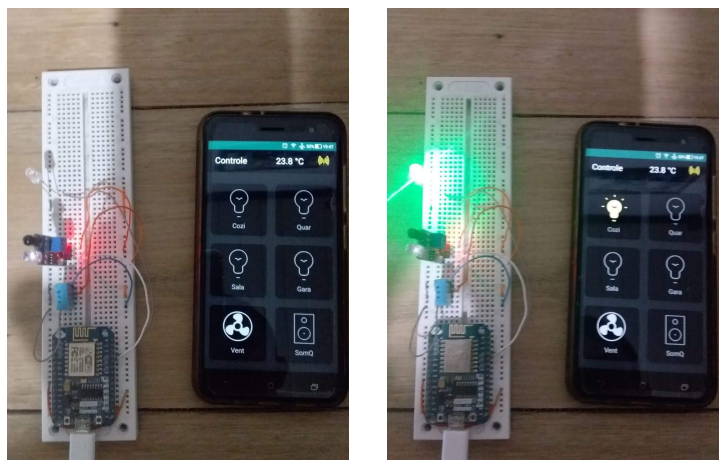


Figura 18. Funcionamento led. Fonte: Elaborado pelo autor, 2020

Como o projeto possui o sensor de obstáculo, na Figura 19 é demonstrado que não há nenhuma notificação, mas ao passar a mão pelo sensor, ao lado mostra que há uma mensagem de notificação informando que ele foi acionado, seguido de toque e vibração no celular.

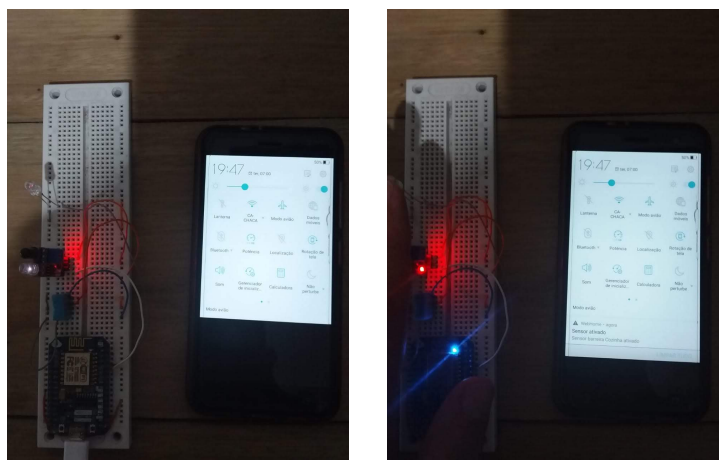


Figura 19. Funcionamento sensor presença. Fonte: Elaborado pelo autor, 2020

Assim, com os objetivos alcançados, atingiram-se os resultados esperados no início da pesquisa.

7. Conclusão e Trabalhos Futuros

Esse trabalho cumpriu o objetivo de conhecer melhor a placa NodeMCU ESP8266 wi-fi, já que, aliada ao aplicativo ZETECH, foi possível simular um ambiente de automação residencial.

Testou-se tanto o sensor de barreira, quanto o de temperatura e o controle do estado do led (ligado ou desligado). Verificou-se que o sistema ZETECH monitorou com eficácia os diversos dispositivos de automação instalados na placa NodeMCU.

Dificuldades foram encontradas, durante a construção do sistema, porém foram superadas com pesquisas em livros, artigos e vídeo aulas.

A partir do conhecimento do funcionamento dos dispositivos utilizados e dos processos automatizados, foi possível realizar as etapas de programação do software que controla o estado do led (ligado ou desligado), monitora o sensor de temperatura e um sensor de barreira de qualquer lugar, desde que estivesse conectado à rede de internet.

Verificou-se que o sistema (que no futuro projeto pode ser ligado a vários eletro-eletrônicos) é capaz de fornecer aos usuários, simultaneamente, conforto, comodidade e controle da residência.

No primeiro ponto da pesquisa, abordou-se a parte elementar do trabalho (o tema, o problema, os objetivos, o escopo, a viabilidade e a organização da pesquisa).

Já a segunda parte, tratou-se do fundamento teórico da pesquisa, o conteúdo que serve de alicerce para a estrutura de todo trabalho. A parte conceitual, funções e características de elementos que fazem parte da placa NodeMCU ESP8266, Ide Arduino, dos sensores, lâmpadas, dos softwares envolvidos no desenvolvimento do aplicativo, das linguagens de programação e de trabalhos correlatos.

O terceiro passo foi o desenvolvimento do aplicativo Home Zeus (ZETECH). Ponto em que se apresentou, de maneira didática, o passo a passo da elaboração do aplicativo e seu desenvolvimento.

Por fim, o último ponto trata dos resultados, conclusões e trabalhos futuros. O microcontrolador eletrônico NodeMCU ESP8266 mostrou-se um mecanismo tecnológico que capaz de manipular diversas tarefas determinadas no aplicativo ZETECH.

O desenvolvimento do presente estudo possibilitou verificar que a placa NodeMCU ESP8266 wi-fi proporciona maior integração dos dispositivos eletrônicos que a plataforma eletrônica Arduino. A placa Arduino não possui um protocolo de comunicação sem fio próprio (wi-fi), ao contrário da ESP8266 que vem com ela integrada. Outro fator é o custo, enquanto a placa Arduino custa mais de RS 100,00 (cem reais), a ESP8266 custa 40 por cento a menos, por volta de RS 65,00 (sessenta e cinco reais).

Os objetivos propostos foram concluídos, de maneira que a idéia principal, controlar de maneira remota equipamentos eletrônicos, foi alcançada. De uma forma simples, bastando apenas alguns instantes após autenticação e cadastro de usuário, o aplicativo ZETECH possibilitou o controle de sensores, lâmpadas e a temperatura.

Como sugestões para evolução da pesquisa, propõem-se num futuro o desenvolvimento de outros projetos acadêmicos para explorar o microcontrolador ESP8266. Demais projetos podem abordar outras tecnologias com o ESP8266, como controle de câmeras, sensores de presença, controles analógicos de motores e principalmente a parte de segurança, cujos relatórios de conteúdos poderão ser gerados, acessados e confirmados pelo usuário.

Referências

- Guilherme Campos Angeloni. **AUTOMAÇÃO RESIDENCIAL CASE: 2000m2 INTELIGENTES2.** DAS CTC USFC, 2016. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/167410/PFC20122GuilhermeCamposAngeloni.pdf?sequence=1>. Acesso em: 16 ago. 2020.
- Massimo Banzi and Michael Shiloh. *Getting started with Arduino: the open source electronics prototyping platform.* Maker Media, Inc., 2014.
- Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. **A review of smart homes—Present state and future challenges.** *Computer methods and programs in biomedicine*, 91(1):55–81, 2008. Disponível em: <http://encurtador.com.br/rtUV6>. Acesso em: 16 ago. 2020.
- Google Cloud. **Lei Geral de Proteção de Dados (LGPD).** Disponível em: <https://cloud.google.com/security/compliance/lgpd?hl=pt-br>, 2020. Acesso em: 31 ago. 2020.
- Fillipe Cordeiro. **Introdução à Linguagem de Programação Android.** Disponível em: <https://www.androidpro.com.br/blog/desenvolvimento-android/linguagem-de-programacao-android/>, 2018. Acesso em: 24 jan. 2020.
- Robson Taveira Gonçalves da Silva. **Sistema de automação residencial de baixo custo utilizando o esp8266.** Disponível em: <http://prpi.ifce.edu.br/nl/lib/file/doc1243-Trabalho/PEVPIRF.pdf>, 2017. Acesso em: 24 jan. 2019.
- Mouser Electronics. **DHT11 Humidity e Temperature Sensor.** Disponível em: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>, 2018. Acesso em: 24 jan. 2019.
- Warren Gay. *Advanced Raspberry Pi: Raspbian Linux and GPIO Integration.* Apress, 2018.
- Gilleanes TA Guedes. *ML 2-Uma abordagem prática.* Novatec Editora, 2018.
- Moacyr Azevedo Couto Junior, Guilherme HF Virtuoso, and Paulo João Martins. **Propriedades desejáveis a uma linguagem de programação: Uma análise comparativa entre as linguagens C, C++ e Java.** *Anais SULCOMP*, 1, 2012. Disponível em: <http://periodicos.unesc.net/index.php/sulcomp/article/viewFile/796/747>. Acesso em: 16 ago. 2020.
- Chunnu Khawas and Pritam Shah. **Application of firebase in android app development-a study.** *International Journal of Computer Applications*, 179(46):49–53, 2018. Disponível em: https://www.researchgate.net/profile/Chunnu_Khawas/publication/325791990_Application_of_Firebase_in_Android_App_Development-A_Study/links/5bab55ed45851574f7e6801e/Application-of-Firebase-in-Android-App-Development-A-Study.pdf. Acesso em: 16 ago. 2020.
- Higor Medeiros. **Introdução ao Padrão MVC.** Disponível em: <https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>, 2013. Acesso em: 24 jan. 2019.
- Silvino Gustavo Albuquerque de Medeiros. **Atomik: uma biblioteca C++ para construção declarativa de interfaces gráficas animadas em sistemas embarcados de baixo custo.** B.S. thesis, Universidade Federal do Rio Grande do Norte, 2019.
- Gustavo Murta. **NodeMCU – ESP12: Guia completo – Introdução (Parte 1).** Disponível em: <https://blog.eletrogate.com/nodemcu-esp12-introducao-1/>, 2018. Acesso em: 24 jan. 2019.

- Elisa Yumi Nakagawa. **Modelos de Processo de Software**. Disponível em: https://edisciplinas.usp.br/pluginfile.php/3134606/mod_resource/content/1/Aula02_ModelosProcessos_2017.pdf, 2017. Acesso em: 11 ago. 2020.
- Ricardo Rodrigues Oliveira. **Uso do microcontrolador esp8266 para automação residencial**. Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli10019583.pdf>, 2017. Acesso em: 24 jan. 2019.
- Claudio Orlandi. **Firestore: serviços, vantagens, quando utilizar e integrações**. Disponível em: <https://blog.rocketseat.com.br/firebase>, 2018. Acesso em: 24 jan. 2019.
- Diana Sobreiro da Costa Palma. **FEUP KNX: domótica KNX/EIB de baixo custo**. FEUP, 2008. Disponível em: <https://repositorio-aberto.up.pt/bitstream/10216/59047/2/Texto%20integral.pdf>. Acesso em: 16 ago. 2020.
- Janquiel Pereira. **Integração de Dispositivos Heterogêneos Via Tecnologia DLNA**. USFC, 2015. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/130044/TCC-janquiel.pdf?sequence=1>. Acesso em: 16 ago. 2020.
- Roger S Pressman. **Engenharia de Software: Uma abordagem profissional. Setima Edição**, 2011.
- RoboCore. **Sensor de Obstáculo**. Disponível em: <https://www.robocore.net/loja/sensores/sensor-de-obstaculo>, 2020. Acesso em: 24 jan. 2019.
- Stuart Russel et al. **Artificial intelligence: a modern approach**. Prentice-Hall, 2013.
- Edilson Leal Santos. **Automação residencial para monitoramento de temperatura, umidade e controle de iluminação utilizando esp8266**. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/235/10337/1/20766608.pdf>, 2016. Acesso em: 24 jan. 2019.
- Tejas Shah and SV Patel. **“Uma nova abordagem para a especificação de requisitos funcionais e não funcionais usando RDS (Esquema de Descrição de Requisitos)”**. *Procedia Computer Science*, 79:852–860, 2016. Disponível em: <https://bityli.com/xps7g>. Acesso em: 16 ago. 2020.
- Ian Sommerville. **Engenharia de Software**. ed, 2011.
- Silicon TechnoLabs. **IR Proximity Sensor**. Disponível em: <https://www.silicontechnolabs.in/upload/IR%20Proximity%20Sensor%20datasheet.pdf>, 2017. Acesso em: 24 jan. 2019.
- Ferdian Thung, Tegawende F Bissyande, David Lo, and Lingxiao Jiang. **Network structure of social coding in github**. In *2013 17th European conference on software maintenance and reengineering*, pages 323–326. IEEE, 2013.
- Tony Tsang and Lam Sai Chung. **Remote Monitoring LV Switchboard Power Status With 5G New Radio Network Application**. *arXiv preprint arXiv:1909.08510*, 2019. Disponível em: <https://arxiv.org/ftp/arxiv/papers/1909/1909.08510.pdf>. Acesso em: 16 ago. 2020.
- Raul Sidnei Wazlawick. **Análise e Design Orientados a Objetos para Sistemas de Informação**, 2014.
- Mark Weiser. **The Open House, March 1996, Written for “Review”, the web magazine of the Interactive Telecommunications Program of New York University**. *Appeared in March*, 1996. Disponível em:

<https://makingfurnitureinteractive.files.wordpress.com/2007/09/wholehouse.pdf>.

Acesso em: 16 ago. 2020.

Leila Weitzel, João Henrique Lopes Spies, and Maycon Medeiros de F Santos. **ANÁLISE DO GITHUB COMO REDE SOCIAL E REDE DE COLABORAÇÃO.** *UFF*, 2017. Disponível em: <https://bityli.com/yVsRw>. Acesso em: 16 ago. 2020.