



CRICTE 2017

XXVIII Congresso Regional de Iniciação Científica e Tecnológica em Engenharia



SISTEMA IOT PARA AUTOMAÇÃO DE AMBIENTES USANDO ARDUÍNO E ANDROID

Éder Paulo Pereira

Egresso do curso de Ciência da Computação da UNIJUÍ
unijic@gmail.com

Pablo José Pavan

Acadêmico do curso de Ciência da Computação da UNIJUÍ
adeliapavan@gmail.com

Edson Luiz Padoin

Professor do Curso de Ciência da Computação na UNIJUÍ
padoin.edson@gmail.com

Resumo. *Vivemos em uma sociedade em que as pessoas estão cada vez mais conectadas através da internet, seja em redes sociais, seja em aplicativos que nos proporcione a comunicação com nossos interlocutores, através dos nossos equipamentos. Este trabalho tem como objetivo automatizar um ambiente remoto para interação à distância, usando para isto tecnologias de IoT, internet e smartphone. Através da interligação de diferentes dispositivos, arquitetura e software adequado iremos oferecer uma solução de IoT de acordo com o problema. Para isto foram utilizados diversas tecnologia, como System on Chip (Soc), Android, Arduino, Rest, Json, Cloud e outras, cada qual sendo como uma engrenagem para que todo um sistema opere de forma adequada.*

Palavras-chave: *Arduino. Android. IoT.*

1. INTRODUÇÃO

Estamos na era em que o IDC (IDC [1]) chama de terceira plataforma. De forma sucinta, na primeira plataforma tivemos o que eram os Mainframes, na segunda, os computadores e as redes LAN, arquiteturas cliente-servidor. Tudo isso contribuiu para o surgimento da terceira plataforma, onde temos uma explosão e novas tecnologias como robótica, sistemas cognitivos,

impressão 3D, e também a tão falada Internet das Coisas (IoT). Tudo isso em uma base sólida e consolidada que é a computação na nuvem, nos dando apoio nas suas mais diversas áreas, como Software as a Service, Plataform as a Service, Big Data e afins. Temos aqui, um enorme *background* de milhões de aplicações, bilhões de usuários e trilhões de *coisas* conectadas na internet.

Neste contexto, o instituto McKinsey realizou uma pesquisa com 2000 americanos sobre algumas dificuldades com novos *gadgets* para conectividade, e os números ficam muito interessantes: 51% dos entrevistados acidentalmente já esqueceu a luz ligada em suas casas ao sair; 41% deixou o televisor ou outro aparelho ligado, consumindo energia; 36% gostaria de saber o que está acontecendo em seus lares à partir de onde estão; 35% já esqueceu o ar-condicionado ligado acidentalmente; por fim, 31% não lembra se trancou todas as portas e janelas ao sair de casa. Tal relatório demonstra que estes usuários frequentemente exibem comportamentos que indicam uma demanda latente para soluções conectadas (Kabir et al. [2]).

Por conta disso este trabalho apresenta uma proposta de integração entre dispositivos diferentes em sua arquitetura, e, como solução, temos um aplicativo para smartphone onde de forma remota,



conseguimos obter informações de nosso ambiente remoto, bem como, interagir com algum dispositivo remoto.

2. ESTADO DA ARTE

Nesta seção, iremos abordar alguns conceitos básicos que foram utilizados no desenvolvimento deste trabalho.

Relacionado à MPSoc, abreviação de Multi Processor System On Chip, trata-se, basicamente, de um sistema embarcado, com processador, memória RAM e armazenamento, tudo isso dentro de um único chip, como o nome sugere. Tal tecnologia nos proporciona seu emprego nas mais diversas áreas, como um sistema de controle de uma usina nuclear, um marcapasso, um brinquedo, um smartphone, dentre outros. Outra tecnologia utilizada em nosso trabalho foi o ARM, que é um tipo de MPSoc, acrônimo de Advanced Risc Machine, temos no ARM uma arquitetura superescalar, ou seja, podemos ter a execução de vários processos, desde que tais estejam no mesmo estágio do pipeline.

Sobre o Arduino, podemos dizer que trata-se de uma placa de prototipação eletrônica, a qual nos traz uma série de portas de entrada e saída digitais, bem como portas analógicas. O dispositivo que irei usar neste trabalho é o modelo Uno, o qual possui um chip micro controlador Atmel AVR normalmente de 8 bits, 8 Kbytes de memória Ram e 32Kbytes de ROM.

Para a infraestrutura, utilizamos um serviço de Cloud computing. Podemos dizer que as organizações necessitam cada vez mais estar conectadas. Neste sentido, temos a internet para suprir algumas das necessidades, no caso a de conectividade. Com o passar dos anos, e com a demanda crescente de serviços de infraestrutura dinâmicos, escaláveis, elásticos, surge um modelo de computação onde temos servidores e serviços disponíveis não mais em nossas empresas, mas sim disponíveis

através da internet, acessíveis aonde quer que estamos. Esta é, de forma simplória, a computação em nuvem. Neste mesmo servidor de nuvem, utilizamos um SGBD, que é a abreviação de Sistema Gerenciador de Banco de Dados. Conforme Raghu et al [2], um sistema de gerenciamento de banco de dados, ou DBMS, é um software desenvolvido para ajudar na manutenção e utilização de grandes volumes de dados, bem como servir estes mesmos dados para os sistemas que crescem rapidamente. Nos dias de hoje, praticamente todas as organizações dispõem de um sistema de gestão empresarial, o qual armazena todas as informações pertinentes à empresa em um banco de dados.

Sobre o Android, podemos dizer que trata-se de um sistema operacional desenvolvido pelo Google, para dispositivos móveis, como smartphones, tablets, e agora está em algumas smartTv's.

Sobre o Rest, o autor Alexandre [3] nos diz que todo serviço Rest é baseado nos chamados recursos, que são entidades bem definidas em sistemas, que possuem identificadores e endereços (URL's) próprios.

3. METODOLOGIAS PARA O DESENVOLVIMENTO DO PROTÓTIPO

Para este trabalho, temos 2 objetivos: medir a temperatura e umidade de um ambiente, e enviar estes dados para um servidor; o segundo objetivo, é podermos acionar um relé ligado à uma lâmpada, de onde quer que estejamos. Na concepção do projeto, logo de começo entendemos que precisaríamos de um local onde seria centralizado todas as informações, e as ações a serem tomadas seriam refletidas neste local também; tal local também é conhecido como servidor, onde alocamos na Amazon. Este servidor que possui um processador Intel Xeon E5-2676v3 contendo um core de



2,4Ghz; 1 Gb de memória Ram e um disco de 10 Gb, com sistema Red Hat Linux de 64 bits. Neste mesmo servidor, instalamos o SGBD, sendo que escolhemos o Mysql, tradicional banco de dados que se difundiu muito por ser leve e rápido, muito utilizado em aplicações web. Para fins de gerenciamento, foi armazenado além da temperatura e umidade, a data e a hora de leitura, para fins de histórico.

Na instalação do servidor web, que vai hospedar a nossa aplicação, escolhemos o servidor Apache, o qual tem em sua instalação e configuração muita simplicidade de operar, por isso se torna um tanto simples no ambiente Linux, e também se revela um ótimo servidor para a aplicação que desejamos. Para este trabalho, foi utilizado um framework baseado na linguagem PHP, o qual nos auxilia na criação dos nossos recursos que serão expostos para serem consumidos. O framework em questão é o SlimFramework, baseado na linguagem PHP. Uma das características mais interessantes dele, é que ele é muito rápido e famoso para criação de API's para serem consumidas por outros serviços ou aplicações em geral, pois já vem pronto com muitas funções prontas para este fim.

Em relação ao Arduíno, utilizamos algumas, dentre milhares, de bibliotecas externas, feitas por terceiros, que nos facilitaram a integração com o nosso servidor Rest. Para o Arduíno, temos dois métodos que são fundamentais ao nosso programa, mas não só para nós, como para qualquer programa para arduíno. O primeiro, chama-se *void setup*, o qual é invocado uma única vez, ao ligarmos o Arduíno, depois disso não é mais necessário. O segundo método, é chamado de *void loop*, onde ocorre de fato a funcionalidade do dispositivo a ser programado. Neste método que desenvolvemos nosso código principal. Definimos que a cada 10 minutos, seria feita uma leitura da temperatura e da umidade do ambiente, e então estes valores seriam

enviados para nosso Rest que está na Amazon; outra funcionalidade, é que o arduino irá consultar o nosso servidor da amazon, a cada 2 segundos, verificando qual deve ser o estado da lâmpada que está conectada através de um relé.

A decisão de fazer um aplicativo mobile para Android se deu pelo fato de o Google abrir uma gama de ferramentas gratuitas, dentre elas o Android Studio, sem a qual seria inviável o desenvolvimento para a plataforma. Todo projeto de Android possui a classe chamada MainActivity, a qual é invocada ao iniciar o aplicativo.

Gostaríamos que, à partir de nosso aplicativo, fosse possível sabermos qual a temperatura e a umidade de algum ambiente remoto, bem como ligar ou desligar uma lâmpada, usando deste mesmo aplicativo. Não iremos persistir nenhum dado no telefone. O principal motivo é que, como queremos saber as informações de temperatura e umidade de nosso ambiente remoto, então precisamos destes dados atualizados da última leitura realizada. A única forma de histórico de temperatura e umidade, será o gráfico que terá na tela do aplicativo.

Para usar o gráfico, utilizamos a biblioteca Hello Charts para o Android, a qual já tem diversos modelos de gráficos, bem como instruções de como incluir ela no projeto do Android Studio. No método onCreate() definimos que ao abrir o nosso aplicativo, iremos atualizar o gráfico, obter a última leitura de umidade e temperatura, bem como atualizar o status do botão que vai controlar a lâmpada em nosso ambiente remoto.

4. PROTÓTIPO

A Figura 1 apresenta o aplicativo desenvolvido.

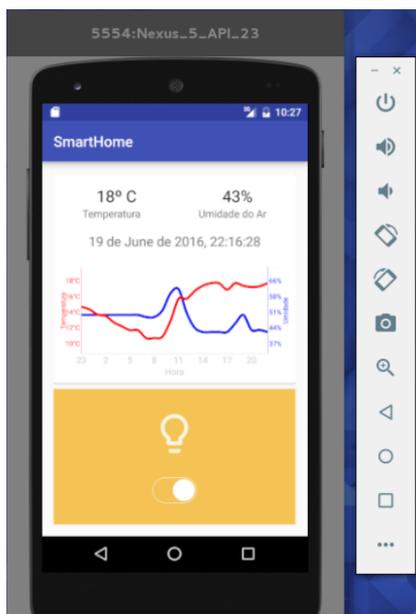


Figura 1: Versão final do aplicativo

No gráfico é plotado a evolução da temperatura e umidade ao longo das últimas 24 horas. Abaixo do mesmo, temos um botão do tipo switch, o qual é responsável por ligar e desligar a lâmpada; o mesmo é atualizado cada vez que abrimos o aplicativo, pois é feita uma chamada no rest para este fim.

Agradecimentos

Agradeço ao meu colega Pablo Pavan, pelo apoio e incentivo ao colaborar comigo para este trabalho. Ao meu professor Padoin, pelas horas desprendidas na orientação, especialmente da escrita, sem ele não seria possível a realização deste artigo.

4. REFERÊNCIAS

[1] IDC, “Explore 3rd Platform Transformation” <<http://www.idc.com/prodserv/3rd-platform/>> 2015. Online; acessado em 18 Jun. 2016.

[2] K. Ahuja and M. Patel “Delivering On The Promise Of Connected Homes” <http://www.mckinsey.com/spContent/connected_homes/index.html> 2015. Online; acessado em 19 Maio.2016.

[3] R. Ramakrishnam and J. Gehrke, Database Management Systems. McGraw Hill, 2000.

[4] A. Saudate, Construa API's inteligentes de maneira simples. Editora Casa do Código, 2014.

CONSIDERAÇÕES FINAIS

Podemos observar que vários dispositivos e arquiteturas de software estão sendo desenvolvidos para o IoT. Neste trabalho, foi automatizado um ambiente remoto, através da intercomunicação entre os dispositivos heterogêneos. Como estamos tratando de quase um sistema de tempo real, temos apenas o método void loop, sendo que precisamos administrar o que colocamos dentro dele para que não fique lento, e nem fique realizando processamentos desnecessários. Um passo importante para este trabalho, seria a adoção do protocolo HTTPS ao invés de HTTP, elevando assim a segurança da comunicação entre o servidor e o aplicativo. Para o arduino ainda não temos esta possibilidade, em função do overhead que o protocolo gera no processamento das requisições.