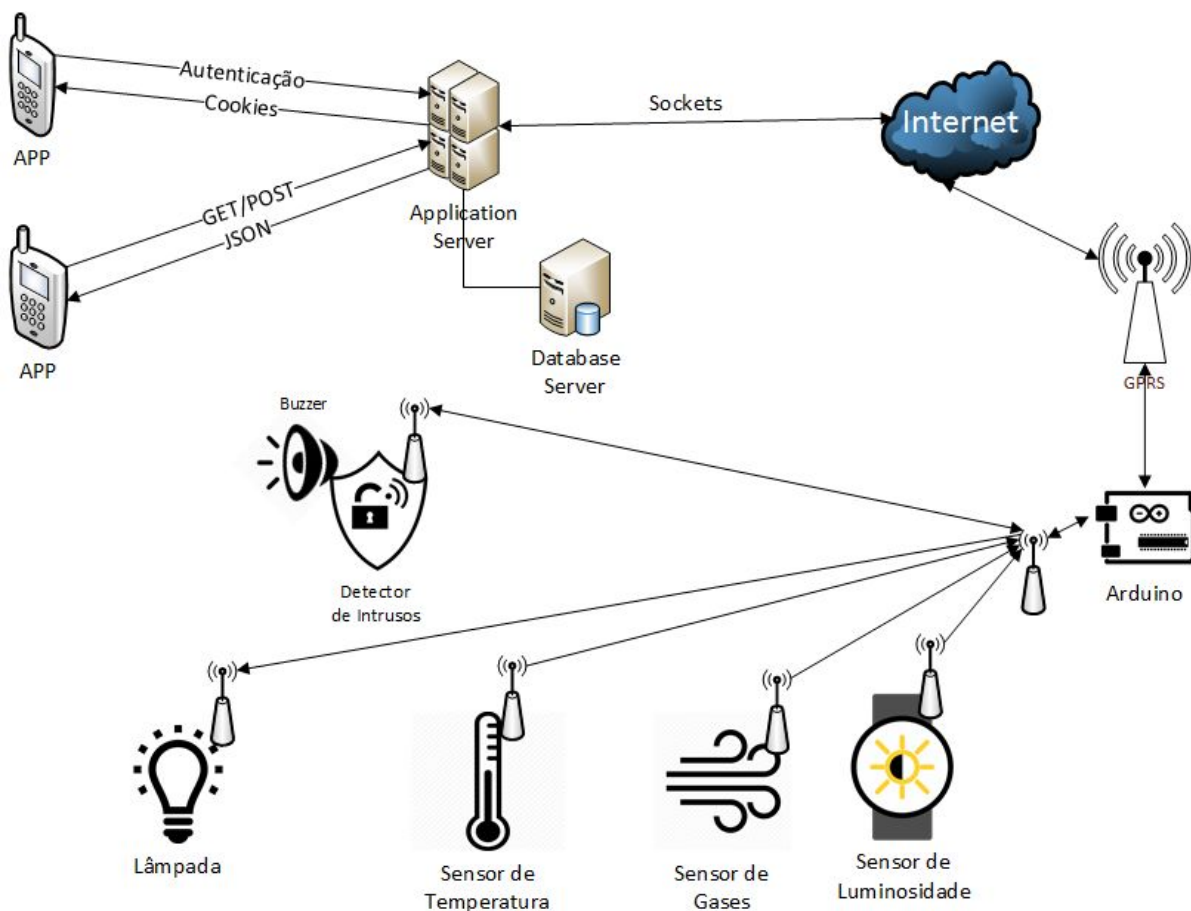


Introdução

A presente aplicação destina-se a uma casa inteligente, capaz de efetuar leitura de sensores e controlar atuadores, bem como, detectar se os seus utilizadores estão presentes ou não na residência e tomar atitudes baseando-se na localização, luminosidade ambiente, horário local, presença de gases inflamáveis/fumo e no estado de portas e janelas (aberta/fechada). Para isto, o sistema conta com uma aplicação Android, capaz de ser utilizada por múltiplos utilizadores ao mesmo tempo, uma infraestrutura de servidores Web e de base de dados, além de três dispositivos de *hardware*, sendo um destinado aos sensores, outro aos atuadores e, por fim, um outro dedicado a comunicação entre os dispositivos finais (sensores e atuadores) e o servidor Web.

Figura 1: Visão geral do sistema



Fonte: Os autores.

Visão Global

A aplicação foi projetada para funcionar com uma ou mais pessoas que vivem na mesma residência, e, desta forma, os utilizadores do sistema podem controlar e monitorizar certos dispositivos. Como por exemplo, ligar e desligar uma lâmpada, ativar e desativar o alarme da casa, que detecta se existe algum intruso, e por fim, monitorizar sensores de luminosidade, temperatura e qualidade do ar. O sistema desenvolvido possui a capacidade de detectar automaticamente a localização do utilizador, baseando-se nas coordenadas provenientes do GPS, recebidas pelo *smartphone*. Caso o utilizador se aproximar do perímetro de 30 metros da sua casa, a aplicação automaticamente desativa quais quer alarmes existentes (para que o usuário possa entrar na sua residência) e liga automaticamente as lâmpadas.

Quando o utilizador se afasta do perímetro da sua residência, a aplicação ativa os alarmes automaticamente e consequentemente, desliga os aparelhos para economizar energia. Caso após este momento alguém invadir a residência, os sensores irão informar o utilizador que a casa foi invadida, o alarme tocará e o utilizador poderá ter a opção de chamar a polícia. Da mesma forma, se o sensor de gases nocivos indicar um alto percentual de gases como álcool, LGP, butano, propano, metano, hidrogénio (gases inflamáveis), ou mesmo CO₂ ou fumo, o utilizador também é notificado e poderá efetuar uma ligação aos bombeiros.

Denominada “SCMU 2”, a aplicação necessita de uma base de dados remota e um servidor dedicado executando um conjunto de servlets, responsáveis por efetuar o login e autenticar o utilizador via cookies. Como também, atualizar os dados da aplicação, obter os dados da aplicação e enviar e receber dados ao hardware da central com Arduino, via *sockets*.

O *hardware* consiste em 3 placas, sendo duas com o microcontrolador ATMEGA328P e a outra com um Arduino Mega 2560. Sendo que, uma destas placas é destinada à leitura dos sensores (luminosidade, detecção de intrusos, temperatura e gases inflamáveis/fumo), outra para controlo dos atuadores (lâmpada e alarme com buzzer) e outra placa para a “central de automação”, responsável por trocar dados com o servidor Web. As placas são dotadas de módulos de radiofrequência para enviarem/receberem dados da central. É importante realçar que somente a placa nomeada de “central”, possui acesso à Internet, por conter um *hardware* que permite o acesso à camada TCP/IP via comunicação GPRS através da rede GSM.

Desta forma, os dados dos sensores são enviados da placa que contém os sensores até a central, sendo esta última responsável por enviar os dados até o servidor Web que fará o armazenamento das informações, para então ser lido pela aplicação móvel. De forma análoga, mas no sentido inverso é realizada o controlo dos atuadores. A aplicação envia o pedido ao servidor Web, que retransmite para a placa central e esta última envia o pacote à placa que contém o atuador.

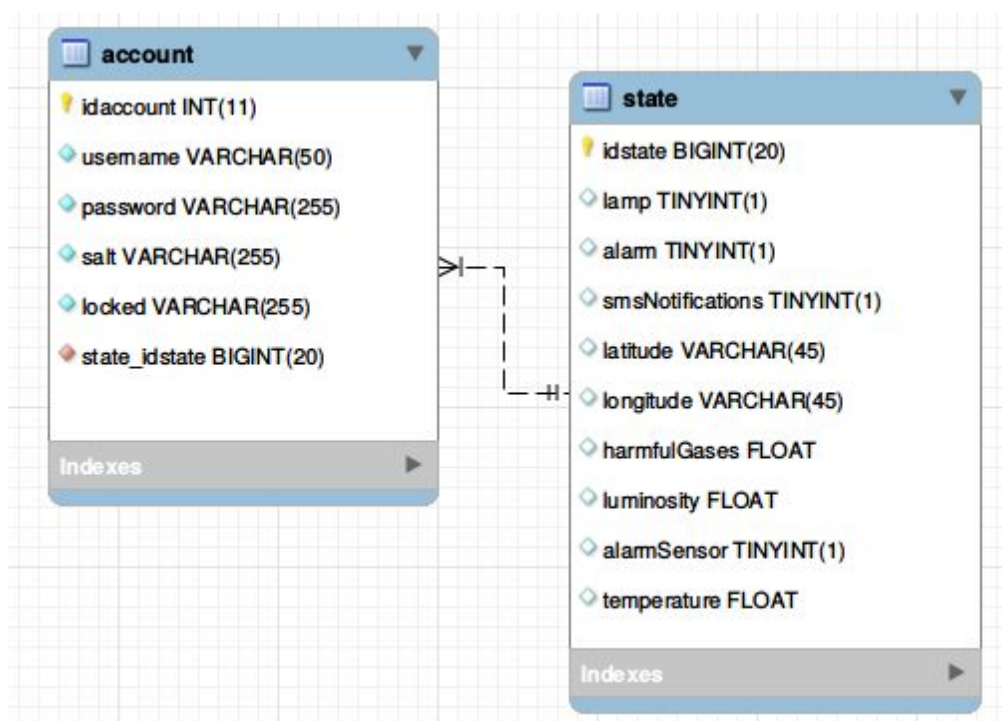
Aplicação Mobile

Base de dados

A base de dados da aplicação consiste em apenas duas tabelas, uma chamada “*account*”, que registra contas de utilizadores, e outra de nome “*state*” que registra o estado atual do sistema. Sendo que, todas as interações de leitura/escrita do sistema, passam pela base de dados. Portanto, ao efetuar uma leitura de um sensor, a aplicação lê diretamente da base de dados, e não diretamente dos sensores em si. A tarefa de atualizar a base de dados é de responsabilidade do *hardware*, que transmite os dados dos sensores periodicamente.

A aplicação permite que duas ou mais pessoas, com nome de utilizador diferente, possam interagir com os mesmos dispositivos, bastando apenas que manipulem o mesmo registro “*state*” na base de dados (Uma *account* só pode ser ligada a um registro *state*, entretanto um registro *state* pode ser ligado a várias *accounts*). Para este projeto, foi utilizado uma base de dados MySQL e o Modelo Entidade Relacionamento da base de dados pode ser visto logo abaixo:

Figura 2: Modelo Entidade-Relacionamento da base de dados.



Fonte: Os autores.

Servidor

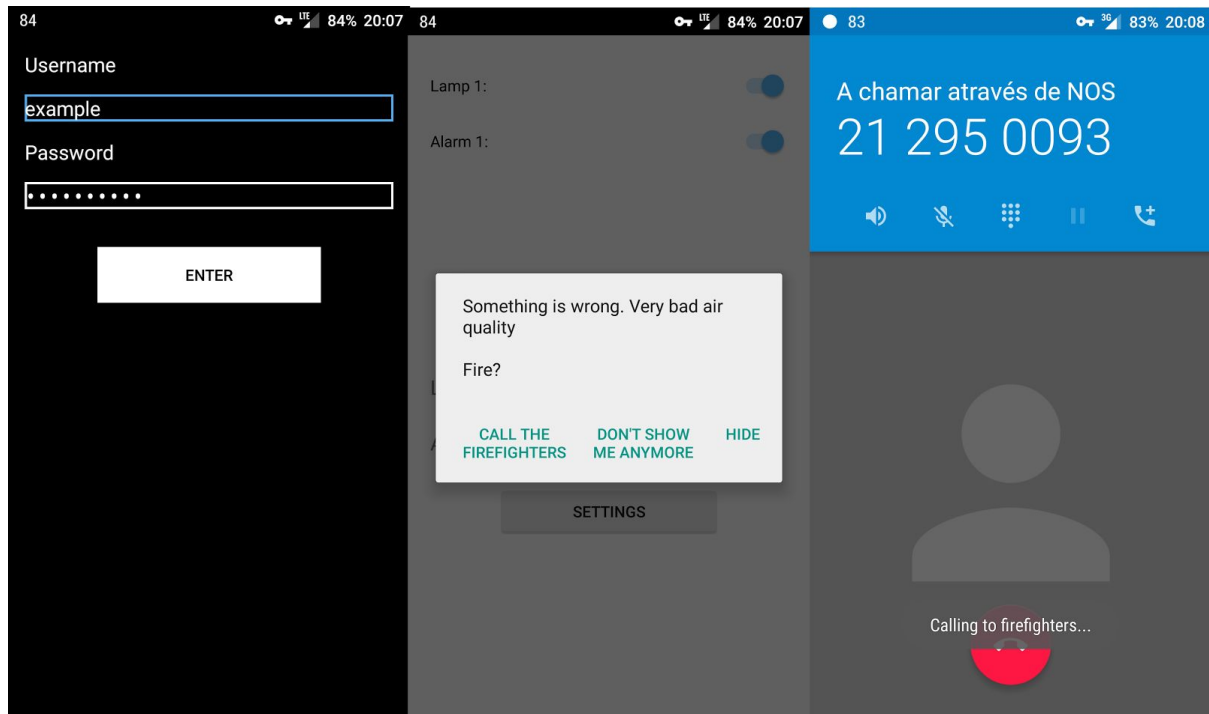
O servidor consiste em uma aplicação JavaEE que é constituída por 3 servlets, a servlet login, getState e setState. A servlet login recebe um nome de utilizador e senha, estes são responsáveis por fazer a autenticação, logo, se os dados inseridos estiverem corretos e constarem na base de dados, a servlet login retorna dois *cookies*, um com o *hash* da *password* e outro com o nome de usuário. As servlets getState e setState, são responsáveis por modificarem os registros sobre o estado da casa. A getState retorna o registro da tabela *state* em forma de JSON e a setState solicita um JSON e rescreve o registro em questão. Ambas as servlets exigem os cookies que foram criados na etapa de login, caso contrário, retornam um erro **401 Unauthorized**.

Aplicação Mobile

O sistema mobile consiste em 3 interfaces, a interface de login, painel e settings. Para aceder os recursos do aplicativo, o utilizador necessita inserir uma credencial válida, então o sistema efetua o login junto a servlet presente no servidor Web e recebe um cookie que é armazenado temporariamente. Posteriormente, caso a credencial seja válida, o utilizador é redirecionado para o painel do aplicativo, onde poderá controlar o estado do alarme e da lâmpada, como também consultar o estado do sensor de gás e de luminosidade.

Nesta etapa, o usuário pode receber duas mensagens, uma com a informação que a qualidade do ar está perigosa, caso o valor do sensor retorne entre 15% e 25%, e outra mensagem informando que algo provavelmente está errado (se o valor do sensor retornar algo acima dos 25%), o que pode indicar a presença de perigo. Em ambas as circunstâncias o utilizador tem a opção de ignorar a mensagem ou de ligar para os bombeiros. Os dados são atualizados a cada 10 segundos para economizar o número de requisições junto ao servidor e caso algum utilizador da mesma casa altere o estado da lâmpada ou do alarme, a aplicação atualiza o estado dos botões em outras instâncias do aplicativo.

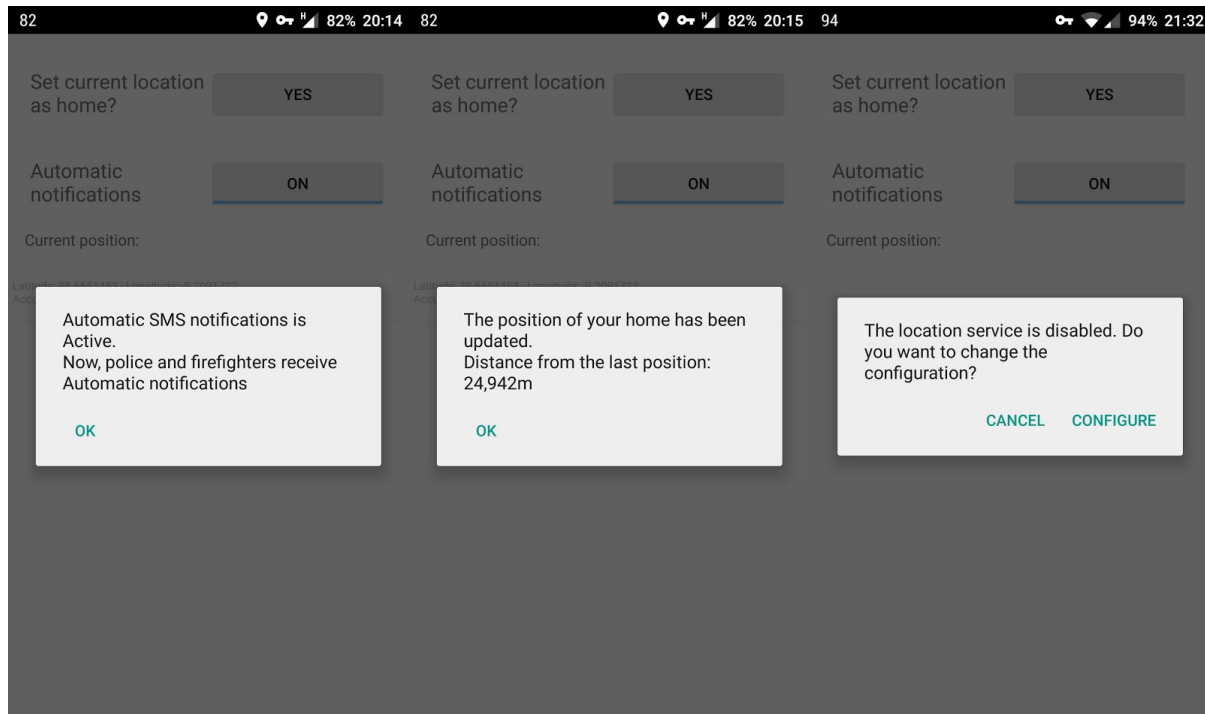
Figura 3: Interface de login, painel e ligação para os bombeiros.



Fonte: Os autores.

A partir da interface “painel” é possível aceder a interface de configurações (*settings*). Nesta interface o utilizador pode definir a localização de sua residência. Escolhendo essa opção o sistema capta a latitude e longitude atual e salva na base de dados para uso posterior. É também possível definir as notificações automáticas. Esta opção, quando marcada, envia um SMS de socorro para a polícia ou bombeiros quando a casa é invadida ou quando o sensor de gases inflamáveis/fumaça atinge 25%, um detalhe desta interface é visível na Figura 4.

Figura 4: Mensagens de informação da interface *settings*.



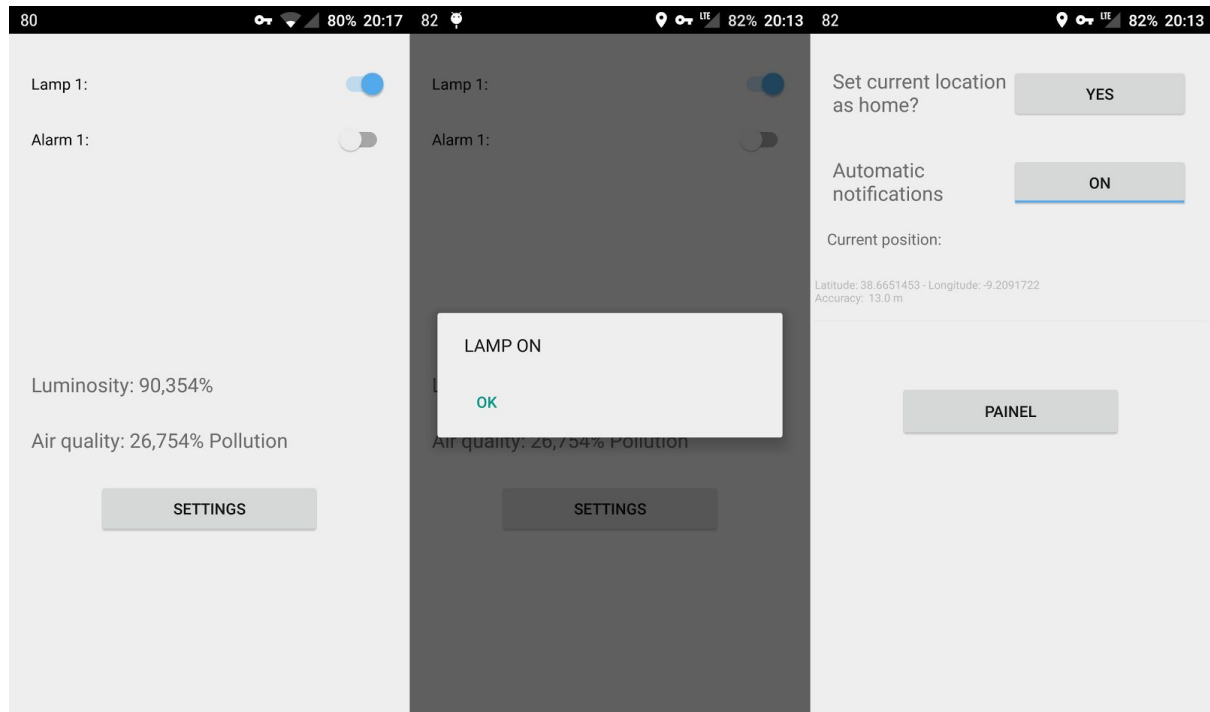
Fonte: Os autores.

Implementação

Entre os grandes desafios desta primeira etapa, pode-se destacar a detecção da localização, o envio e recebimento de dados do servidor Web, bem como, a autenticação via cookies e a sua gestão dentro da aplicação.

Para permitir que todas as Activities manipulassem os cookies, foi necessário serializar o objeto na LoginActivity e deserializá-lo nas outras Activities, para permitir que todas pudessem fazer requisições as servlets getState e setState que necessitam de cookies e estão localizadas no servidor Web. Para isto, foram criadas duas classes auxiliares CookiesImpl e CookieStoreImpl.

Figura 5: Painel (a esquerda), mensagem de informação (ao centro), interface settings exibindo localização atual (a direita)



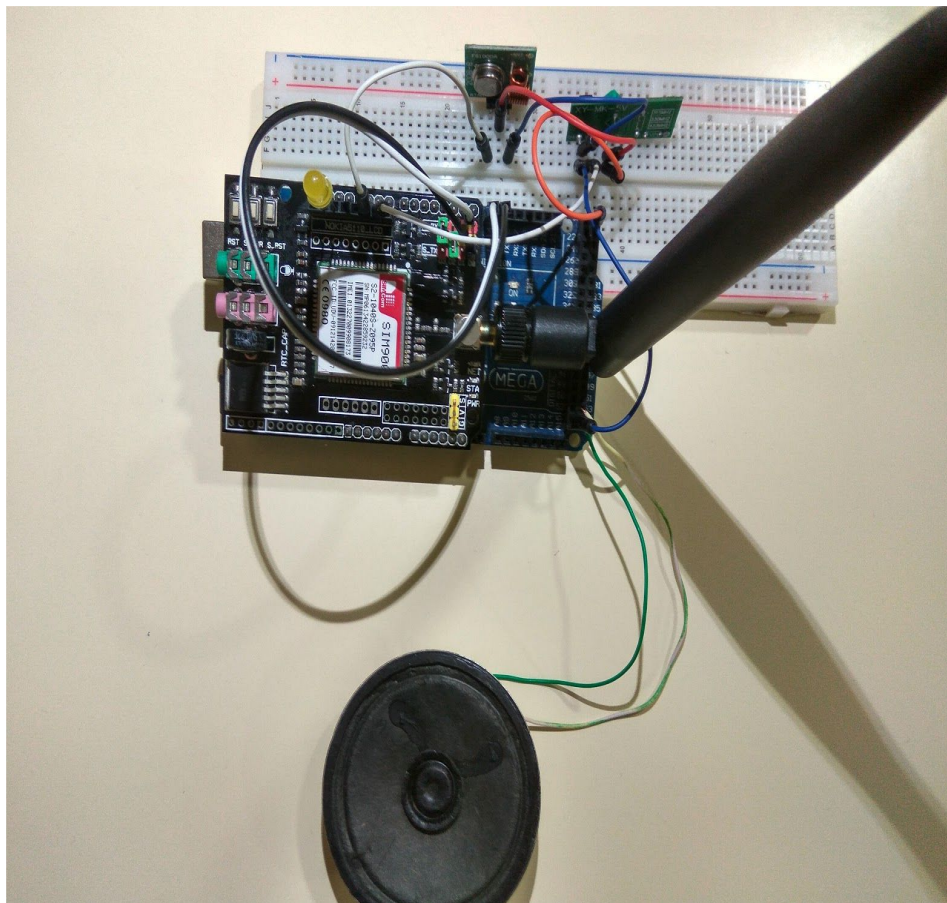
Fonte: Os autores.

Sensores e Atuadores

Placa central

A placa “central” é a principal placa do sistema, pois é a responsável por efetuar a comunicação entre os dispositivos da casa e o servidor Web. É dotada de um módulo GSM para efetuar comunicação GPRS utilizando sockets de rede e de dois módulos de radiofrequência, um receptor e um emissor para comunicação entre os outros dispositivos da residência. A central é ainda dotada de um buzzer que é utilizado para emitir um aviso sonoro quando o sistema detecta uma intrusão.

Figura 6: Placa “central”, responsável por enviar e receber dados para o servidor Web e efetuar a comunicação com as outras placas.

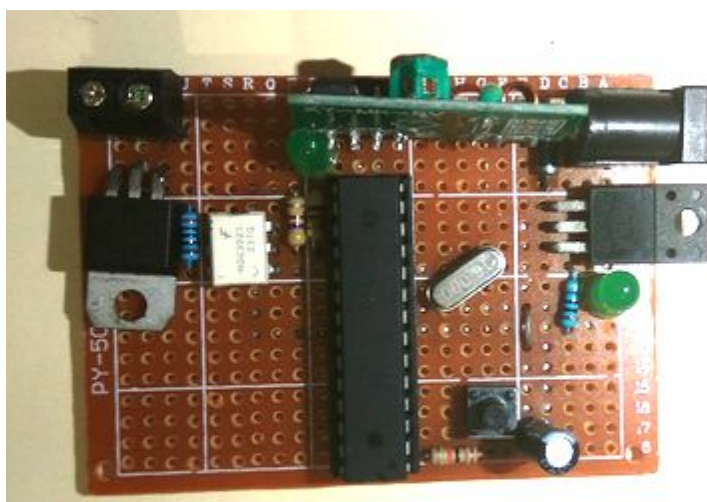


Fonte: Os autores.

Placa de atuador

A placa de atuador é capaz de ligar qualquer dispositivo ligado em série nos seus dois terminais, respeitando limites de tensão e corrente de 1000V e 16A respectivamente. Para este projeto, optou-se por ligar uma lâmpada, que representa a lâmpada de uma casa. Entretanto, o mesmo circuito e solução poderia ser utilizado para abrir um portão, acionar uma cafeteira elétrica ou ligar/desligar qualquer outro eletrodoméstico presente em uma residência.

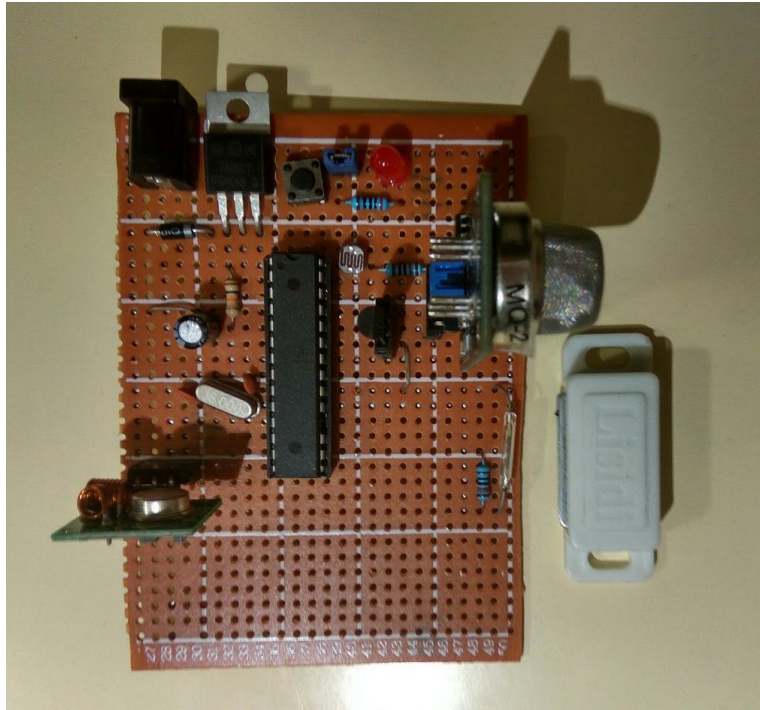
Figura 7: Placa atuadora - Responsável por ligar e desligar um dispositivo conectado aos dois terminais.



Fonte: Os autores.

Esta placa é composta basicamente de um microcontrolador, um circuito de alimentação, circuito de acionamento de potência e um módulo de radiofrequência receptor. O *firmware* é basicamente um loop que fica em listening observando os dados recebidos via radiofrequência da placa central e, caso o comando enviado corresponda a “ligado”, o firmware altera o estado da lâmpada, em caso contrário, a desliga. Pelo facto da placa central utilizar conexão móvel GSM ao invés de uma rede Wi-fi comum, observou-se um delay de aproximadamente 8 segundos em cada requisição, oriundo da autenticação com a operadora móvel.

Figura 8: Placa de sensores - Responsável por ler o estado dos sensores de temperatura, gases nocivos, intrusão e luminosidade e enviar a placa central (Figura 6)



Fonte: Os autores.

A placa de sensores é similar à de atuadores, entretanto se difere por ter um módulo de radiofrequência emissor, invés de um receptor e ser composta por 4 sensores no lugar de um atuador. O firmware é basicamente um loop que efetua a leitura dos sensores quatro sensores e envia os dados em forma de um JSON para a placa central. A central, por sua vez, envia estes dados até o servidor Web, que então guarda na base de dados para que a aplicação possa utilizar esses mesmo mais tarde.

Experiências

Sockets

A aplicação foi testada de forma modular com intuito de verificar cada parte do projeto, prevenir erros e comportamentos inesperados. Devido a placa central efetuar uma comunicação via sockets com o servidor Web (a placa GSM não suporta requisições HTTP) foi necessário implementar um sistema de autenticação e de requisições baseado em comandos de strings. Portanto, procurou-se testar primeiro o funcionamento da placa central, que se comunica com o módulo GSM por meio de comandos AT e mais tarde a comunicação com o servidor, que se dá via sockets de rede.

Para efetuar a comunicação com o servidor Web, foram criados os seguintes comandos:

`$CONNECT;{nome_do_utilizador};{senha};`

- Efetua a conexão com o servidor, caso o utilizador e senha esteja correto, devolve um “200 OK”, caso contrário, 401.

`$GET;`

- Requer os dados salvos na base de dados e devolve um JSON com todos os dados da tabela “state”.

`$PUT;{JSON};`

- Atualiza os dados na base de dados, caso o update seja efetuado com sucesso, devolve “201 Created”, caso contrário devolve “500 Internal Server Error” ou encerra a conexão.

`$POST;`

- Devolve um “100 Continue”, em seguida é necessário enviar um JSON com dados a atualizar. Caso o processo ocorra bem, devolve um “201 Created”, caso contrário, devolve “500 Internal Server Error”

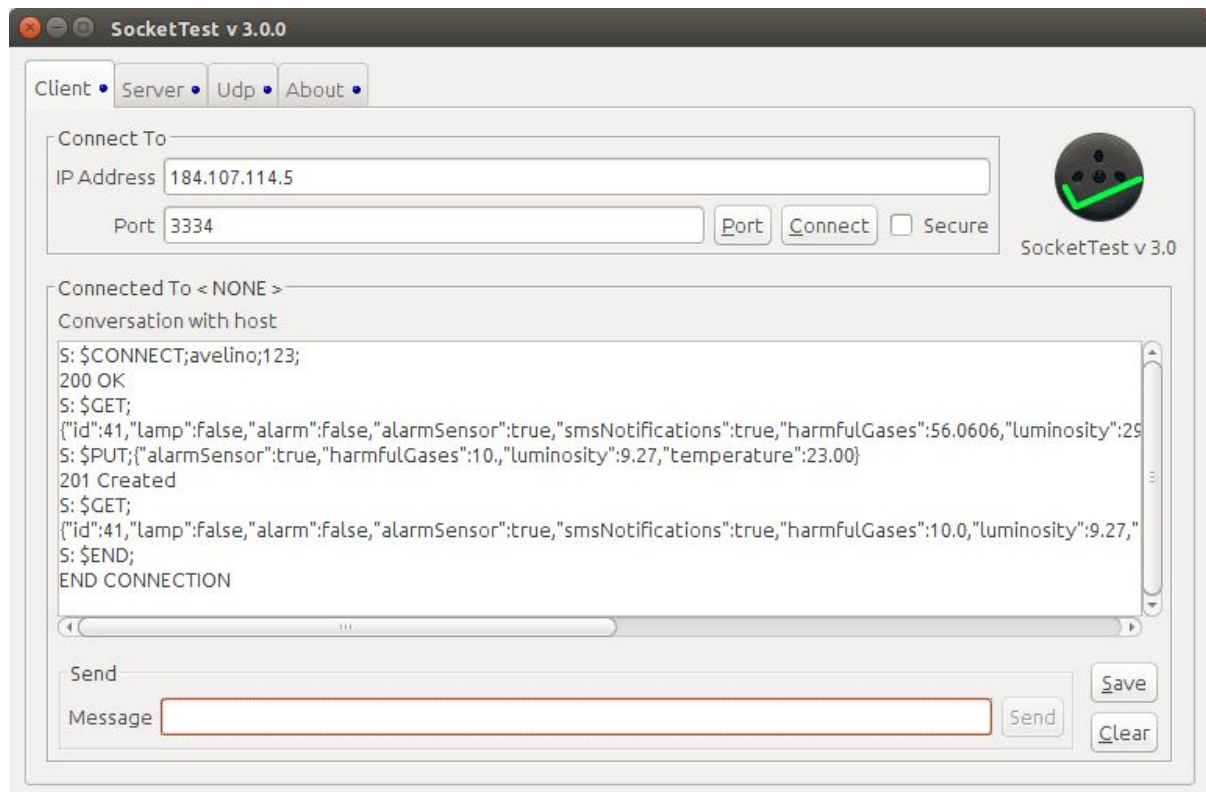
`$END;`

- Encerra a conexão e retorna “END CONNECTION”

`$HI;`

- Recebe um “HEY!” como resposta. Usado para testar conexão.

Figura 9: Ferramenta java para testar sockets. Usado para testar qualquer servidor ou cliente que use protocol TCP ou UDP para comunicar

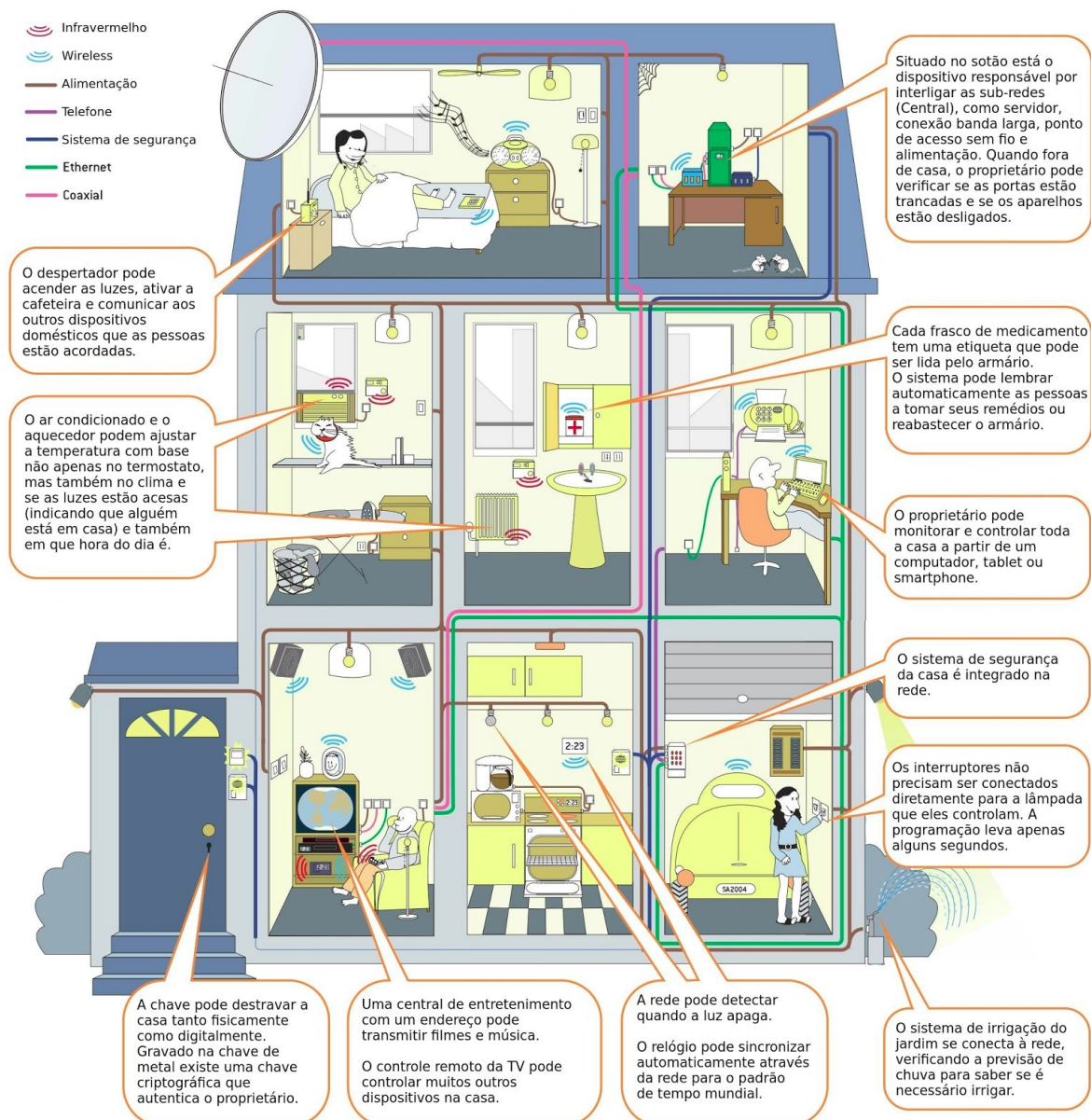


Conclusão

O artigo "The Internet of Things" publicado em 2004 na revista Scientific American, foi um dos primeiros e mais relevantes artigos a abordarem o tema *Internet of Things* (IoT). Neste, Gershenfeld, Krikorian e Cohen (2004) explicam e abordam aspectos relevantes, que variam desde a arquitetura a ser seguida até aos problemas a serem enfrentados futuramente. Neste trabalho, procurou-se desenvolver um sistema ubíquo e móvel que atendesse pequenas necessidades de uma residência. Porém, este trabalho pode ser ampliado no futuro para que se encaixe no paradigma de IoT.

De acordo com Gershenfeld, Krikorian e Cohen (2004), não é necessária uma infra-estrutura extensa, com conexões na casa dos gigabytes ou hardware sofisticado, mas os autores defendem justamente o contrário, que os dispositivos sejam pequenos, compactos, de baixo consumo, processamento e com baixo tráfego de dados. Pois ao contrário do streaming de vídeos, que demanda um grande processamento e banda, por exemplo, uma lâmpada precisa de poucos dados, a serem transmitidos, para ser controlada e monitorizada. A exemplo da aplicação desenvolvida para este trabalho, que foi construída em cima de um hardware extremamente barato, cujo componente principal, o microcontrolador ATMEGA328P, apresenta um preço de aproximadamente 1.50 euros em sites de eletrônicos e a aplicação móvel consome cerca de 140 bytes a cada requisição efetuada junto ao servidor. Futuramente, a aplicação pode vir a ser melhorada utilizando-se do hardware e a infraestrutura já existente para efetuar cruzamentos de dados entre sensores e atuadores no intuito de torná-la mais inteligente.

Figura 10: Exemplo da casa do futuro abordado no artigo da revista Scientific American



Fonte: Adaptado de Gershenfeld, Krikorian e Cohen (2004)