

Desenvolvimento de Aplicação Web para Gerenciamento e Eficientização de Sistema de Iluminação

D. W. M. Monteiro*¹, J. M. Tabora*, U. C. Paixão Jr.*, E. P. de S. Lobato*, A. R. M. de Sousa*,
C. C. M. de M. Carvalho*, M. E. de L. Tostes*

*Centro de Excelência em Eficiência Energética da Amazônia, Universidade Federal do Pará, Belém, Pará.
¹deniswmonteiro@gmail.com

Abstract: The automation and supervision of systems has been growing, providing increased comfort and reliability, as well as reduced process costs. Linked to web applications with a friendly interface, automated systems provide the efficiency of electrical systems in buildings with extreme practicality. After the development of a low-cost automation system, this article presents the development of a web application capable of managing an automated system of electric lighting from different devices with internet access. As a case study, the work presents the impact of automation associated with the web application on the consumption and cost of electricity in a building of a public education institution.

Resumo: A automação e supervisão de sistemas tem sido crescente, propiciando o aumento da comodidade e confiabilidade, bem como o reduzido gasto de processos. Vinculado a aplicativos web com interface amigável, os sistemas automatizados proporcionam a eficientização de sistemas elétricos de edificações com extrema praticidade. Após o desenvolvimento de um sistema de automação de baixo custo, este artigo apresenta o desenvolvimento de uma aplicação web capaz de gerenciar um sistema automatizado de iluminação elétrica a partir de diferentes dispositivos com acesso à internet. Como estudo de caso, o trabalho apresenta o impacto da automação associado ao aplicativo web no consumo e custo de energia elétrica de um edifício de uma instituição de ensino público.

Keywords: Web Application; Responsive Web Design; Multiplatform; Building Automation; Energy Efficiency.

Palavras-chaves: Aplicação Web; Web Design Responsivo; Multiplataforma; Automação Predial; Eficiência Energética.

1. INTRODUÇÃO

De acordo com a pesquisa anual de administração e uso de tecnologia da informação nas empresas realizada pela Fundação Getúlio Vargas (FGV), no ano de 2019 havia 230 milhões de *smartphones* e 180 milhões de computadores, *notebooks* e *tablets* em uso no Brasil. A pesquisa apontou ainda que o *smartphone* é o dispositivo mais utilizado para fazer transações bancárias, compras *online* e para acessar as mídias sociais (FGVcia, 2019). Segundo a pesquisa TIC Domicílios 2018 feita pelo Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (CETIC) o telefone celular foi utilizado por 97% dos usuários para acessar a rede, enquanto pelo *desktop* o acesso foi realizado por 43% dos usuários (CETIC.br, 2019).

A crescente utilização de aparelhos móveis impacta na redução do uso de *softwares* criados exclusivamente para *desktops*, sendo necessário que os desenvolvedores de aplicações criem sistemas que funcionem em diferentes dispositivos, contribuindo para uma evolução no desenvolvimento de aplicativos web.

A disseminação de aplicativos web também influenciou outros setores, como a construção civil e sistema elétrico, por exemplo. O crescimento tecnológico tem grande impacto no consumo energético, visto que o aumento de aparelhos elétricos demanda maior necessidade de energia elétrica para atividades residenciais, comerciais e industriais. Sendo assim, a automação de empreendimentos vem tornando os mesmos mais autônomos e eficientes, reduzindo a necessidade de pessoas no local para tomada de decisão (Monteiro, 2017). Desta forma, sistemas automatizados controlados a partir de dispositivos móveis oferecem maior qualidade de vida para usuários, introduzindo assim o conceito de edifício inteligente.

Os sistemas automatizados também contribuem com a gestão eficiente da energia elétrica, pois permitem que dispositivos eletroeletrônicos possam ser gerenciados remotamente, evitando gastos desnecessários, tanto energeticamente quanto financeiramente.

A automação predial ainda é considerada como uma aplicação de alto custo, porém o avanço recente em computação em nuvem, análise de dados e dispositivos eletrônicos de baixo custo tem contribuído para a utilização de dispositivos provenientes da internet das coisas. Deste modo, este trabalho

visa apresentar o desenvolvimento de uma aplicação web responsável por controlar um sistema automatizado de iluminação a partir de dispositivos móveis conectados à internet, bem como evidenciar o impacto energético e econômico da sua implementação em uma edificação educacional.

1.1 Problemática

O desenvolvimento deste trabalho parte da premissa que a sociedade está cada vez mais conectada à dispositivos eletrônicos, resultando em alterações no comportamento das pessoas e em sua interação com o meio ao qual vivem. As interações entre as pessoas e o ambiente são feitas cada vez mais através de sistemas automatizados, onde tarefas rotineiras podem ser executadas remotamente a partir de dispositivos eletrônicos, trazendo ao usuário mais comodidade, segurança e maior controle sobre o que o cerca. Porém, ainda é muito comum em dispositivos eletrônicos a utilização de sistemas computacionais que fazem uso de aplicativos exclusivos, sem oferecer ao usuário flexibilidade para utilizar o aplicativo em diferentes plataformas.

Visando adaptar os aplicativos a diferentes plataformas, houve o aumento no desenvolvimento e utilização de aplicativos web, tendo em vista a sua adaptação às necessidades dos seus usuários, já que estes podem utilizar os aplicativos em qualquer plataforma que faça uso de navegadores web e que tenham acesso à internet. A facilidade de adaptação das aplicações web é ampliada quando há o uso de uma interface intuitiva que auxilie o usuário no gerenciamento do sistema automatizado.

A implementação da automação de tarefas cotidianas aliada a uma aplicação com interface de fácil utilização oferece ao usuário um mecanismo que atende as suas necessidades, trazendo benefícios financeiros e, como no caso abordado neste trabalho, energéticos, através do gerenciamento eficiente da iluminação elétrica.

2. DESCRIÇÃO DO TRABALHO PROPOSTO

O projeto apresentado neste trabalho foi dividido em três partes. A primeira parte consiste na criação do *design* da aplicação, implementando as características relacionadas à parte visual da aplicação; na segunda parte se apresenta o desenvolvimento propriamente dito da aplicação, fazendo uso de tecnologias de desenvolvimento de *software* e a terceira parte consiste na conexão com o sistema automatizado.

2.1 Tecnologias Aplicadas

Foram utilizadas ferramentas que possibilitam a criação das mais diversas aplicações, independentemente do propósito das mesmas. Para desenvolvimento das páginas da aplicação, ou seja, o *front-end*, utilizou-se a linguagem de marcação HTML (do inglês, *Hypertext Markup Language*) e a linguagem de estilos CSS (do inglês, *Cascading Style Sheets*) em conjunto com o *framework* UIKit, devido este ser leve e modular para o desenvolvimento de interfaces web com maior rapidez

(UIKit, n.d.). Além destes, utilizou-se a linguagem de programação JavaScript em conjunto com o *framework* jQuery para simplificar a adição de interatividade e animações à aplicação web.

Visando a adaptação a diferentes tamanhos de tela, foi empregado o conceito web *design* responsivo à aplicação. Tendo em vista que uma página web pode ser visualizada por infinitos dispositivos (*desktops, laptops, tablets, smartphones, smart TVs* etc.) com diferentes resoluções, a responsividade tem se tornado cada vez mais relevante para que os usuários possam ter a melhor experiência ao acessar um *site* (Zemel, 2015).

Para lidar com a comunicação pelo lado do servidor, ou seja, o *back-end*, foi utilizada a linguagem de programação PHP (*PHP: Hypertext Preprocessor*) por meio do *framework* Laravel. O Laravel utiliza o padrão de arquitetura de *software* MVC (*Model-View-Controller*) para o desenvolvimento de aplicações web, minimizando a escrita manual de códigos (Gabardo, 2017), possibilitando a apresentação e interação de dados do sistema. A comunicação entre os três componentes lógicos MVC, conhecidos como camadas, permite que os dados sejam alterados de forma independente da sua representação (Sommerville, 2012).

2.2 Análise do Projeto de Automatização

Com base no trabalho elaborado por Souza Lobato (2019) para automação de sistemas de iluminação, foi desenvolvido o aplicativo para gestão do sistema. Entretanto, foram necessários ajustes no *software* que faz o controle do sistema desenvolvido pela autora para que este funcionasse corretamente em conjunto com o aplicativo desenvolvido neste trabalho. A comunicação entre a aplicação web e o serviço em nuvem CloudMQTT é realizado por meio de publicação/subscrição de tópicos e subtópicos com mensagens. Portanto, foram adicionados ao *software* o seguinte código:

```
char mqtt_topico_sub_c1[30] =  
"ceamazon/biblioteca/c1";  
char mqtt_topico_sub_c2[30] =  
"ceamazon/biblioteca/c2";
```

O código exemplifica a implantação do módulo Esp32 na biblioteca do prédio a ser automatizado, o qual possui dois circuitos de iluminação, representados por “c1” e “c2”, sendo a publicação/subscrição da mensagem representada por “tópico/subtópico/circuito”. As próximas seções descrevem os principais componentes de *hardware* do sistema.

2.2.1 Componentes do Protótipo

Visando automatizar a operação das luminárias de um dado local e oferecer conforto luminoso aliado a eficiência energética, os materiais descritos na Tabela 1 foram utilizados para construir o protótipo. Na Fig. 1 é apresentado o esquemático construído no *software* EAGLE.

Tabela 1. Especificações dos dispositivos do protótipo.

Item	Dispositivo	Descrição
1	Módulo de desenvolvimento Esp32	Esp32 DevKit v1 WROOM-32
2	Módulo relé de estado sólido com 2 canais	Entrada: 240 VCA/Saída: 5 VCC de 2 A e 8A
3	Conversor de tensão AC/DC 5 W	Hi-Link PM01/Entrada: 110V/Saída: 5 V
4	Chave gangorra de três posições	3 terminais (ON-OFF-ON)
5	Placa de circuito impresso perfurada	10x10 mm
6	Caixa de passagem	150x150x150 mm
7	Cabo elétrico	Bitola: 1,5 mm
8	Resistor cerâmico	100 Ω
9	Fusível de vidro (AC)	100 V/1 A
10	Porta fusível para PCI	5x20 mm
11	LED	Cor: verde/5 mm
12	Barra sinal	10 A/12 vias
13	Barra conectora simples	18 vias/Conector: fêmea

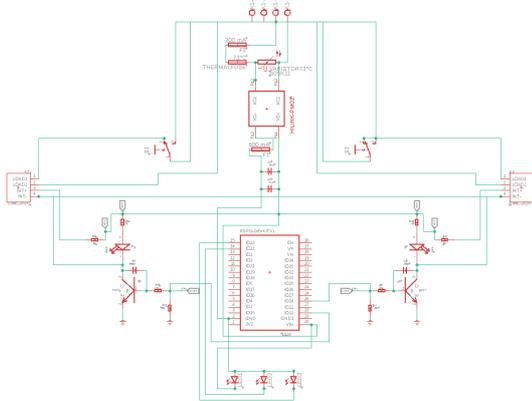


Fig. 1 Esquemático do protótipo construído.

2.2.2 Ambiente de Desenvolvimento Integrado (IDE) do Arduino

O Ambiente de Desenvolvimento Integrado, do inglês *Integrated Development Environment (IDE)*, do Arduino foi utilizado para desenvolver o código responsável pelo controle do sistema automatizado. A seguir é mostrado um trecho adaptado do código desenvolvido por Souza Lobato (2019) para o acionamento do circuito de acordo com a mensagem enviada para o módulo Esp32.

```

if(strMensagem == "on1") {
    digitalWrite(pi no1, HI GH);
    gravarStatusPi no(HI GH);
} else if(strMensagem == "off1") {
    digitalWrite(pi no1, LOW);
    gravarStatusPi no(LOW);
} if(strMensagem == "on2") {
    digitalWrite(pi no2, HI GH);
    gravarStatusPi no2(HI GH);
} else if(strMensagem == "off2") {
    digitalWrite(pi no2, LOW);
    gravarStatusPi no2(LOW);
}

```

2.3 Desenvolvimento do Sistema

2.3.1 Criação do Design da Aplicação

Para o desenvolvimento do *design* do aplicativo utilizou-se o *software* Figma. O *software* permite definir cores, tipografia, espaçamentos, entre outros, estabelecendo qual o aspecto visual de cada elemento antes da codificação destes. Desta forma, foram desenvolvidos elementos que possam constituir uma interface de fácil manuseio e que se adaptem bem ao dispositivo utilizado pelo usuário.

A Fig. 2 mostra um dos *designs* criado no Figma e corresponde à primeira página a ser vista ao acessar o aplicativo, a página inicial, com a visualização dos andares do edifício. O *layout* foi criado a partir dos planos arquitetônicos do edifício.

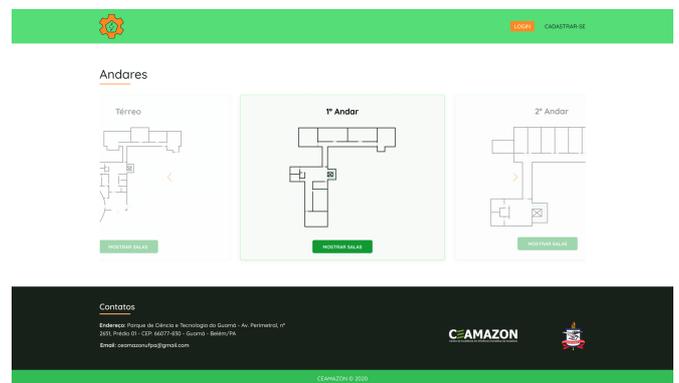


Fig. 2 Página inicial da aplicação.

Além do *design* da página inicial, foram criados os *designs* das outras páginas da aplicação, como a página de login; a página de solicitação de cadastro; a página de edição de perfil de usuário; e a página de gerenciamento de usuários, onde é possível adicionar um novo usuário, bem como editar informações de usuários já cadastrados. Outrossim foi criado o *design* da página de gerenciamento de solicitações, onde são mostradas as solicitações de cadastro feitas por usuários visitantes, além de permitir que as solicitações possam ser editadas antes que o usuário seja incluído no sistema. As Figs. 3 e 4 mostram os *designs* das páginas de acesso ao sistema e da página de solicitação de cadastro.

Também foram criadas três páginas para o gerenciamento dos circuitos de luminárias, as páginas Térreo, 1º Andar e 2º Andar, de modo que cada página possui os locais dos dispositivos descritos em um mapa, além de uma legenda no rodapé indicando ao usuário cada local e seu respectivo nome. Ao clicar no marcador do local, o usuário visualizará os dispositivos presentes neste local, podendo acionar luminárias.

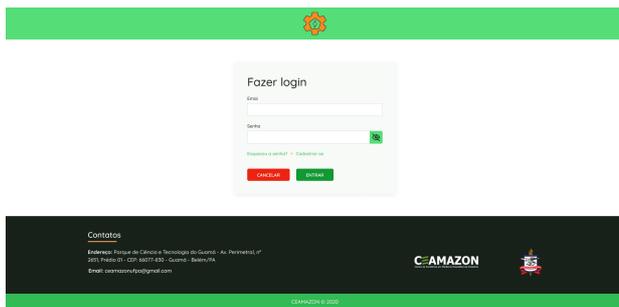


Fig. 3 Página de acesso ao sistema.

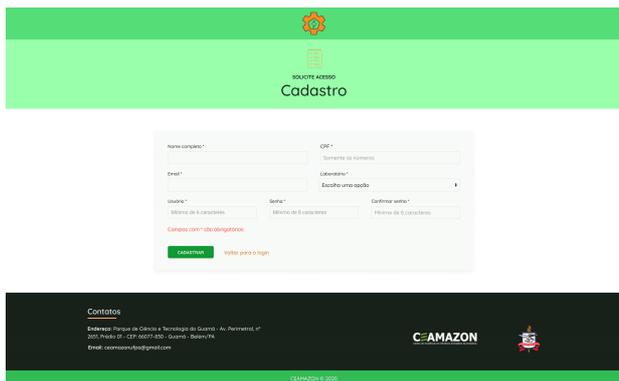


Fig. 4 Página de solicitação de cadastro.

2.3.2 Codificação da Aplicação

Após a elaboração dos *designs* de cada página da aplicação, criou-se a codificação da aplicação web. Esta foi desenvolvida utilizando o *software Visual Studio Code*, devido este oferecer suporte às tecnologias de desenvolvimento de *software* empregadas neste projeto por meio de extensões que facilitam a edição e depuração dos códigos da aplicação. Antes de fazer uso do Laravel foi necessária a instalação do Composer, uma ferramenta utilizada para o gerenciamento de dependências e instalação de bibliotecas PHP diretamente nos diretórios de desenvolvimento do aplicativo (Composer, 2020).

Ao criar um projeto com o Laravel há a divisão de todas as funcionalidades da aplicação em diretórios, permitindo melhor gerenciamento por parte do desenvolvedor. Porém, antes de criar os arquivos com os códigos da aplicação, foram definidas as rotas de direcionamento da aplicação. Rotas são ferramentas providas pelo Laravel para ajudar na criação de URLs de redirecionamento de uma parte para outra da aplicação (Laravel, 2020). A partir das rotas também é possível definir as permissões de acesso às páginas da aplicação. A Fig. 5 mostra as categorias de usuários da aplicação e quais páginas cada categoria pode acessar.

A divisão de usuários por categoria visa dar segurança ao sistema implantado, de modo que o usuário visitante poderá visualizar a página inicial e, para acessar o sistema, é necessário que o mesmo solicite o cadastro; o usuário CADASTRADO terá acesso ao sistema a partir das credenciais fornecidas e pode acessar a página inicial, a página de perfil de usuário e as páginas de acionamento das luminárias; já o usuário ADMINISTRADOR possui total

acesso ao sistema, tendo acesso à página inicial, ao acionamento das luminárias, à edição do perfil de usuário, além do gerenciamento dos usuários cadastrados e o gerenciamento de solicitações de cadastro.

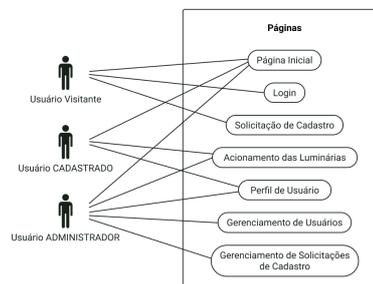


Fig. 5 Páginas que podem ser acessadas de acordo com a categoria de usuário.

Utilizando o padrão MVC (*Model-View-Controller*), criou-se a codificação das páginas do *site*, onde cada camada é responsável por parte da funcionalidade da aplicação. A camada *Model* é responsável por processar os dados referentes aos usuários, à solicitação de cadastro e aos locais gerenciáveis de cada andar do prédio. Nestes arquivos estão especificados o banco de dados utilizado, o nome do documento (*collection*) a ser salvo no banco de dados e os nomes dos campos que fazem parte deste documento, que serão preenchidos com os dados fornecidos pela aplicação. A seguir é apresentado um exemplo do código criado para a *collection* “salas_terreo”.

```
<?php
Namespace projetoautomacao;
Use Jenssegers\Mongodb\Eloquent\Model as Eloquent;

Class SalasTerreo extends Eloquent {
    Protected $connection = 'mongodb';
    Protected $collection = 'salas_terreo';
    Protected $fillable = [
        'id_sala', 'sala_nome', 'sala_localizacao',
        'qtd_circulampada', 'qtd_arcondicionado',
        'coord_circulampadas', 'coord_arcondicionado'
    ];
    Protected $guarded = ['id'];
}
```

A camada *View* está dividida em diretórios que contém os arquivos pertinentes a cada parte da aplicação que pode ser visualizada. Os diretórios são: “admin”, onde ficam os arquivos das páginas referentes ao gerenciamento de usuários e ao gerenciamento de solicitações de cadastro; “andares”, onde ficam os arquivos das páginas referentes aos andares do prédio e aos locais onde será instalado o sistema automatizado; “layouts”, onde estão os arquivos que servem como modelo para as páginas da aplicação; e “auth”, onde ficam armazenados os arquivos da página de acesso ao sistema e de registro de usuário. Além dos diretórios mencionados, há também arquivos que estão na pasta raiz da camada, como a página inicial, a página de solicitação de cadastro e a página de perfil de usuário.

A camada *Controller* foi dividida em diretórios onde cada um representa qual parte da aplicação é responsável pela requisição. Estes diretórios são: “Admin”, onde estão os

arquivos que lidam com as requisições referentes aos usuários do sistema e às solicitações de cadastro; “Andares”, contendo os arquivos que controlam as requisições de cada andar do prédio e dos locais gerenciáveis; e “Auth”, com os arquivos que lidam com as requisições referentes ao acesso autorizado ao sistema e ao registro de usuário. Outrossim, há arquivos que estão na pasta raiz desta camada, como os arquivos que controlam a página inicial, a solicitação de cadastro e a edição de perfil de usuário.

As informações referentes a aplicação foram salvas no banco de dados MongoDB, contendo os dados dos usuários; os locais do prédio; os circuitos de iluminação; e as solicitações de cadastro. Para que a aplicação funcione corretamente quando acessada pela internet foi utilizado o MongoDB Atlas, um serviço em nuvem que oferece toda a infraestrutura para o gerenciamento de um banco de dados, sendo necessária apenas a administração dos dados por parte do desenvolvedor (MongoDB, n.d.). Este serviço oferece um plano gratuito, o *MO Sandbox*, oferecendo um banco de dados com armazenamento de 512 MB e 500 conexões simultâneas, suprimindo as necessidades da aplicação neste momento.

A conexão entre a aplicação desenvolvida em PHP a partir do *framework* Laravel e banco de dados MongoDB é implementada a partir das variáveis de conexão do banco de dados. Sendo assim, a conexão contém o banco de dados a ser utilizado, o *link* de conexão contendo nome de usuário, a senha de acesso, o nome do *cluster*, o nome do banco de dados, e novamente o nome do banco de dados.

A aplicação web foi colocada *online* para ser acessada através da internet e, para isto, foi utilizada a plataforma Heroku para funcionar com o *framework* Laravel. O Heroku é uma plataforma em nuvem com o objetivo de prover para o desenvolvedor um ambiente para a execução de aplicativos (Heroku, n.d.). Esta ferramenta tem a capacidade de fazer o gerenciamento de todos os pontos de iluminação do prédio projetado.

2.3.3 Conexão entre o aplicativo e o sistema automatizado

Para fazer o acionamento dos dispositivos via Wi-Fi, o módulo Esp32 faz uso do serviço em nuvem CloudMQTT. Este serviço possibilita a utilização de servidores na nuvem para o envio de mensagens utilizando o protocolo *Message Queuing Telemetry Transport* (MQTT) através do método de publicação/subscrição de mensagens (CloudMQTT, n.d.), cuja sequência de envio e recebimento de mensagens é mostrada na Fig. 6.



Fig. 6 Diagrama de conexão utilizado pelo CloudMQTT.

Este serviço foi escolhido devido a utilização do protocolo MQTT, por possuir QoS (*Quality of Service* – Qualidade de Serviço), por utilizar *Websocket* para envio e recebimento de mensagens e por disponibilizar um plano gratuito que garante o funcionamento do sistema. As mensagens de comunicação

são oriundas de tópicos criados pelo desenvolvedor da aplicação no serviço em nuvem.

A aplicação web subscreve um tópico do CloudMQTT, monitorando todas as mensagens trocadas entre sensor e módulo Esp32. Quando há mudança no estado do objeto de estudo, o sensor publica a informação coletada no serviço em nuvem e este o publica na aplicação web, onde o dado poderá ser visto pelo usuário. Utilizou-se o plano *Cute Cat*, devido este ser gratuito e, neste momento, suprir as necessidades do projeto. Os dados necessários para a conexão são as informações do servidor, o nome de usuário, a senha, a porta, a porta utilizando o protocolo SSL (*Secure Sockets Layer* – Camada de Soquete Seguro) e a porta que utiliza *Websocket* com o protocolo TLS (*Transport Layer Security* – Segurança da Camada de Transporte).

Para fazer a conexão da aplicação web com o sistema automatizado foi utilizado o Eclipse Paho JavaScript Client, uma biblioteca escrita em JavaScript que implementa comunicação via *WebSocket* para envio e recebimento de mensagens utilizando o protocolo MQTT (Eclipse, n.d.). Além desta biblioteca foi desenvolvido um código também em JavaScript que contém todos os dados da conta criada no CloudMQTT em conjunto com funções para o gerenciamento da conexão e para o acionamento das luminárias.

Também foram criadas funções para garantir que os estados dos botões de acionamento dos circuitos de iluminação e das luminárias da imagem que corresponde ao local gerenciado pudessem ser mantidos mesmo quando a página seja recarregada pelo usuário, mostrando primeiramente que a aplicação está se conectando ao sistema e posteriormente que a conexão foi estabelecida. Além disso, a modificação do estado é propagada caso mais de um usuário esteja usando a aplicação e um deles faça o acionamento dos circuitos, isto é, caso um “Usuário A” ligue o circuito “Lâmpadas 1”, um “Usuário B” que também estiver visitando a mesma página da aplicação no mesmo momento em outro dispositivo verá este circuito sendo ligado.

3. ESTUDO DE CASO E RESULTADOS

Como estudo de caso da implantação do sistema de automação e o sistema web, será apresentado uma descrição do prédio do Centro de Excelência em Eficiência Energética da Amazônia (CEAMAZON), instalado na Universidade Federal do Pará (UFPA), demonstrando o funcionamento da aplicação em conjunto com o sistema automatizado de iluminação, além do custo do sistema, por meio de uma análise financeira da sua implantação.

3.1 Descrição do Prédio

O prédio possui 3 pavimentos, conforme podem ser visualizados nas Figs. 7(a), (b) e (c), divididos entre salas de aula, laboratórios, auditório, biblioteca e escritórios, em sua maioria, e foi subdividido em 23 locais onde serão instalados o sistema automatizado. O Térreo possui 8 locais, com 17 circuitos de luminárias; no 1º andar há 7 locais, com 16

circuitos de luminárias; e no 2º andar há 8 locais, com 16 circuitos de luminárias.

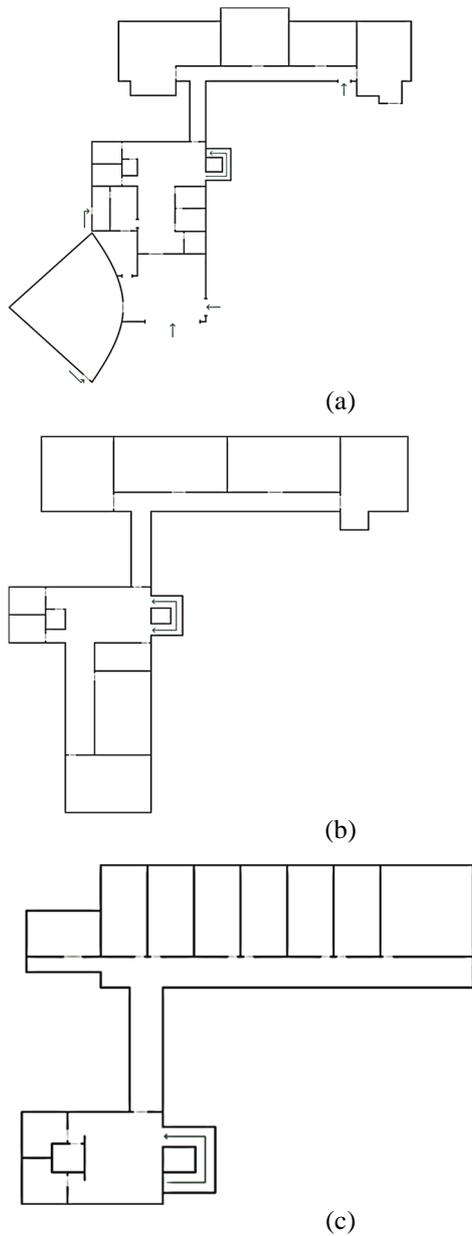


Fig. 7 Planta baixa do CEAMAZON com (a) Térreo, (b) 1º andar, e (c) 2º Andar.

3.2 Análise do Sistema

3.2.1 Análise do Funcionamento do Sistema

Para validar a aplicação web foi construído um circuito no *proto board* que simula a sala da biblioteca existente no CEAMAZON, onde há dois circuitos de luminárias representados por dois LEDs, como mostrado na Fig. 8.

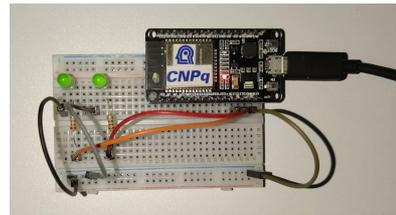


Fig. 8 Circuito simulando a biblioteca do CEAMAZON.

A Fig. 9 mostra a página da biblioteca na aplicação web, onde é possível visualizar o *layout* com a disposição das luminárias presentes no local e os interruptores dos dois circuitos de luminárias, ambos desligados (cor vermelha).

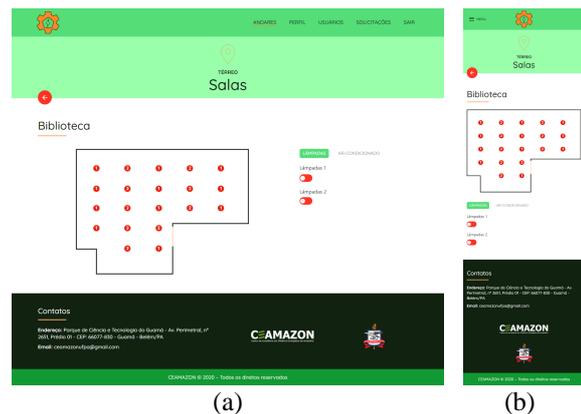


Fig. 9 Página da biblioteca do CEAMAZON visto em um (a) *desktop* e em um (b) *smartphone*.

Ao clicar no interruptor que corresponde ao circuito nomeado como “Lâmpadas 1”, a cor deste é modificada para verde, bem como a representação das luminárias pertencentes a este circuito, como mostra a Fig. 10. O LED que aqui representa este circuito também é acionado, como mostrado na Fig. 11.

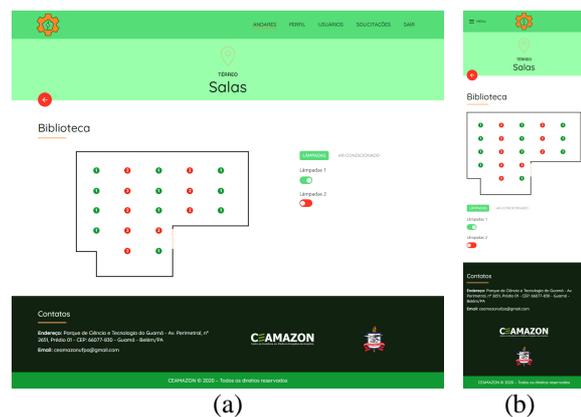


Fig. 10 Página da biblioteca com o circuito “Lâmpadas 1” acionado em um (a) *desktop* e em um (b) *smartphone*.

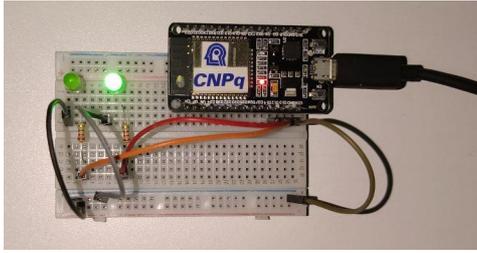


Fig. 11 LED do circuito “Lâmpadas 1” aceso.

Para acendimento de todo conjunto de iluminação, clicando no interruptor do circuito nomeado como “Lâmpadas 2” na mesma página, as cores são modificadas para verde, assim como a representação das luminárias pertencentes a estes circuitos, como mostra a Fig. 12. Os LEDs que representam os circuitos “Lâmpadas 1” e “Lâmpadas 2” também são acionados, como mostrado na Fig. 13.

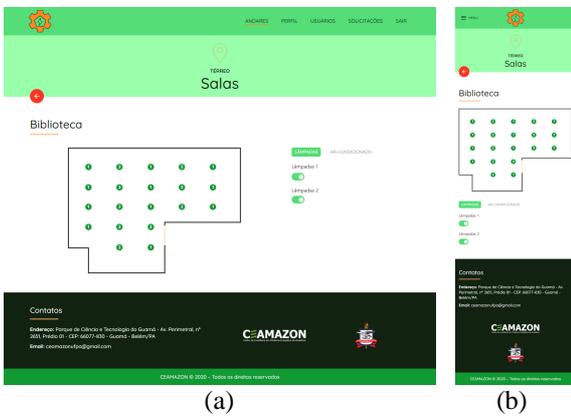


Fig. 12 Página da biblioteca com os circuitos acionados em um (a) *desktop* e em um (b) *smartphone*.

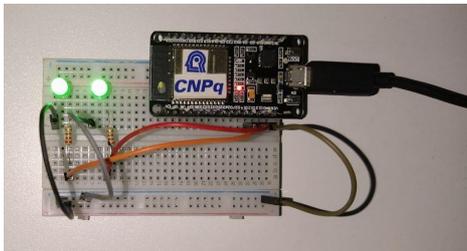


Fig. 13 LEDs dos dois circuitos acesos.

Para os demais ambientes do prédio, foram realizados testes em suas respectivas páginas e todos funcionaram como o previsto, mostrando que a aplicação atende ao seu objetivo inicial.

3.2.2 Análise Financeira de Implantação do Sistema

Os componentes para instalação do sistema de iluminação do CEAMAZON estão listados na Tabela 1 deste artigo. O valor orçado para a compra dos componentes é de R\$ 4.688,40. O valor do serviço de instalação do sistema automatizado nos 23 locais do prédio foi orçado em R\$ 2.760,00. Logo, o custo total de implantação do sistema de automatização é de aproximadamente R\$ 7.448,40. Como não se tem o sistema automatizado instalado em todo o prédio, o cálculo da

economia de energia após a instalação do sistema automatizado foi estimado, utilizando as Equações de (1) a (6), com uma economia estimada de 20% (Williams et al., 2012) (Delvaeye et al., 2016).

$$E_{Iluminação} (kWh) = P_{LI} \times h \quad (1)$$

$$\%E_{Iluminação} = \frac{E_{Iluminação}}{E_M} \quad (2)$$

$$E_{SA} = E_{HP} + E_{HFP} \quad (3)$$

$$\%E_{SA} = \frac{E_{LSA}}{E_{Iluminação}} \quad (4)$$

$$\begin{aligned} \text{Custo}_{SA} (R\$) &= E_{HP} (kWh) \times C_{HP} (R\$/kWh) \\ &+ E_{HFP} (kWh) \times C_{HFP} (R\$/kWh) \\ &+ P_{DSA} (kW) \times C_{PDSA} (R\$/kW) \end{aligned} \quad (5)$$

$$\begin{aligned} \text{Economia}_{Custo_{SA}} (R\$) &= \text{EconomiaEstimada} \times E_{HP} (kWh) \times C_{HP} (R\$/kWh) \\ &+ \text{EconomiaEstimada} \times E_{HFP} (kWh) \times C_{HFP} (R\$/kWh) \\ &+ \text{Economia}_{PDSA} \times P_{DSA} (kW) \times C_{PDSA} (R\$/kW) \end{aligned} \quad (6)$$

Onde $E_{Iluminação}$ é a energia consumida em iluminação; P_{LI} a potência das luminárias instaladas; h o número de horas de operação; E_M a energia média consumida no prédio em um ano; E_{SA} a energia consumida pelo sistema automatizado; E_{LSA} a energia consumida pelas luminárias do sistema automatizado; E_{HP} e E_{HFP} a energia consumida pelo sistema automatizado nos horários de ponta e fora de ponta, respectivamente; C_{HP} e C_{HFP} o preço do kWh nos horários de ponta e fora de ponta, respectivamente; P_{DSA} a potência de demanda máxima para o sistema automatizado; C_{PDSA} o preço da demanda máxima; e Economia_{PDSA} a economia com a demanda máxima para o sistema automatizado.

Os valores calculados nas Equações de (1) a (6) são mostrados na Tabela 2.

Tabela 2. Valores calculados nas equações (1) a (6).

Variável	Valor
$E_{Iluminação}$	1.051,60 kWh
$\%E_{Iluminação}$	16,23%
$\%E_{SA}$	80,52%
Custo_{SA}	R\$ 687,74
$\text{Economia}_{Custo_{SA}}$	R\$ 126,96

As Figs. 14 e 15 apresentam as economias mensal e anual, respectivamente, projetadas com a implementação do sistema proposto. Atualmente, as luminárias projetadas para o sistema automatizado impactam aproximadamente R\$ 8.252,89 na despesa anual do CEAMAZON com iluminação e a economia projetada no mesmo período após implantação do sistema é de aproximadamente R\$ 1.523,48.

Os custos estimados de implementação e a economia anual esperada também foram considerados. Com base nos resultados, um *payback* de aproximadamente 4 anos e 11 meses é obtido, com uma vida útil estimada de 10 anos.

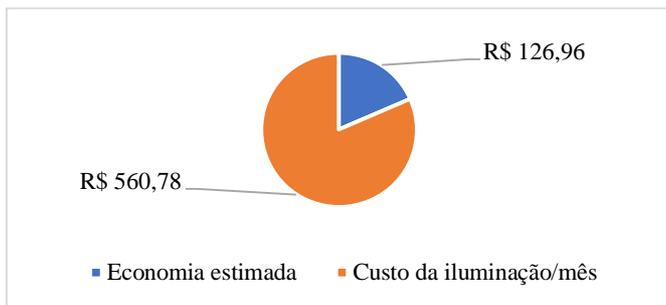


Fig. 14 Projeção mensal da economia após automatização.

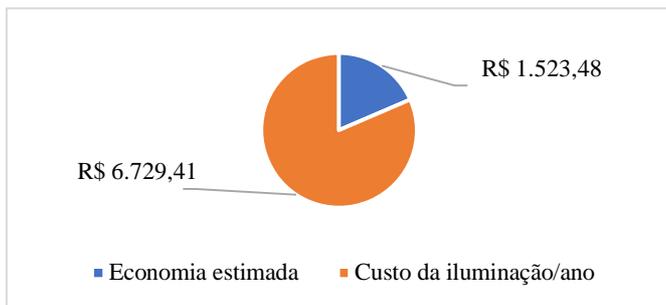


Fig. 15 Projeção anual da economia após automatização.

4. CONCLUSÕES

A adoção de aplicações web tem se mostrado uma forma eficaz para a execução de tarefas cotidianas, devido este poder ser executado em qualquer plataforma. Aliado à boas práticas de usabilidade e segurança, os aplicativos web proporcionam uma melhor experiência para o usuário, por meio de uma interface projetada utilizando o *design* responsivo.

Este trabalho apresentou o desenvolvimento de uma aplicação web para a gestão do sistema de iluminação automatizado implementado em um prédio educacional. Os testes mostraram que a aplicação web se comporta como esperado, sendo possível fazer o acionamento de todos os locais do prédio onde o sistema automatizado for implantado. A implementação de uma interface que se adapte em dispositivos com diferentes resoluções facilita não apenas o acionamento das luminárias, mas também a visualização de seu *status*, resultando uma boa experiência e operabilidade ao usuário.

Por último, o estudo de caso apresentado mostrou que a implementação de sistemas automatizados para o gerenciamento de dispositivos elétricos possibilita a redução do consumo de energia elétrica, trazendo benefícios financeiros aos cofres de instituições, além de reduzir o impacto ambiental e o desperdício de energia no setor predial.

AGRADECIMENTOS

Os Autores agradecem o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001; e do CNPq para a realização do presente trabalho, no âmbito do projeto "Sistema inteligente para determinação dos impactos harmônicos de múltiplos consumidores nas redes de distribuição de energia elétrica", aprovado na Chamada Universal - MCTI/CNPq/CT - ENERGNº 14/2016.

REFERÊNCIAS

- CETIC.br. (2019). Pesquisa sobre o uso das tecnologias de informação e comunicação nos domicílios brasileiros: TIC domicílios 2018 = Survey on the use of information and communication technologies in brazilian household: ICT households 2018. 392.
- CloudMQTT. (n.d.). CloudMQTT - Documentation. CloudMQTT - Documentation. Retrieved March 5, 2020, from <https://www.cloudmqtt.com/docs/index.html>
- Composer. (2020). Composer - Getting Started. <https://getcomposer.org/doc/00-intro.md>
- Delvaeye, R., Ryckaert, W., Stroobant, L., Hanselaer, P., Klein, R., & Breesch, H. (2016). Analysis of energy savings of three daylight control systems in a school building by means of monitoring. *Energy and Buildings*, 127, 969–979. <https://doi.org/10.1016/j.enbuild.2016.06.033>
- Eclipse. (n.d.). Eclipse Paho JavaScript Client. Eclipse Paho JavaScript Client. Retrieved April 1, 2020, from <https://www.eclipse.org/paho/clients/js/>
- FGVcia. (2019). Resumo de Notícias: 30ª Pesquisa Anual do FGVcia da FGV/EAESP, 2019 - Mercado Brasileiro de TI e Uso nas Empresas. 2.
- Gabardo, A. C. (2017). *Laravel Para Ninjas* (1st ed.). Novatec.
- GetUIKit. (n.d.). UIKit. GetUIKit. Retrieved January 3, 2020, from <https://getuikit.com/>
- Heroku. (n.d.). Heroku Solutions. Heroku Solutions. Retrieved March 13, 2020, from <https://www.heroku.com/solutions>
- Laravel. (2020). Laravel - URL Generation. <https://laravel.com/docs/7.x/urls>
- MongoDB. (n.d.). MongoDB Atlas. MongoDB Atlas. Retrieved April 1, 2020, from <https://www.mongodb.com/cloud/atlas>
- Monteiro, A. F. de O. (2017). *Sistemas Inteligentes para Melhoria da Eficiência Energética em Edifícios* [Dissertação de Mestrado, Instituto Superior de Engenharia de Lisboa]. <https://repositorio.ipl.pt/bitstream/10400.21/7980/1/Disserta%C3%A7%C3%A3o.pdf>
- Sommerville, I. (2012). *Engenharia de Software* (9th ed.). Pearson Prentice Hall.
- Souza Lobato, E. P. de. (2019). *Desenvolvimento de um Sistema IoT para o Controle de Iluminação Residencial Baseado nos Princípios Indústria 4.0* [Trabalho de Conclusão de Curso]. Universidade Federal do Pará.
- Williams, A., Atkinson, B., Garbesi, K., Page, E., & Rubinstein, F. (2012). Lighting Controls in Commercial Buildings. *LEUKOS*, 8(3), 161–180. <https://doi.org/10.1582/LEUKOS.2012.08.03.001>
- Zemel, T. (2015). *Web Design Responsivo: Páginas adaptáveis para todos os dispositivos*. Casa do Código.