

SISTEMA AUTOMATIZADO DE CONTROLE DE OCUPAÇÃO DE ASSENTOS EM SALAS DE APRESENTAÇÃO

AUTOMATED SEAT OCCUPATION CONTROL SYSTEM FOR PRESENTATION ROOMS

Ricardo Rall¹Davi Rodrigo de Miranda²Arthur Américo de Lima Ferreira²

RESUMO

Internet das coisas (IoT) é o termo moderno que descreve a interconectividade de objetos, devido às novas tecnologias e apresenta-se como uma área de expansão para negócios otimizar seus processos e oferecerem novas experiências. Sistemas automatizados de controle de ocupação de assentos nos cinemas permitem um gerenciamento mais eficiente dos assentos disponíveis em cada sessão. Isso ajuda a otimizar a ocupação das salas, evitando superlotação ou assentos vazios. A alocação adequada de assentos também contribui para uma experiência mais agradável para os espectadores, reduzindo a confusão e o estresse na busca por lugares. Neste sentido, o objetivo deste trabalho foi desenvolver um sistema IoT que controle de ocupação de assentos em uma sala de cinema, a fim de promover uma melhor experiência ao negócio dos cinemas, a partir do uso de sensores instalados em uma cadeira que transmite esses dados via internet a um website desenvolvido em Node.js. O sistema desenvolvido mostrou como IoT pode ser usada em um negócio e providencia ao gerente informações relevantes que podem ser usadas para otimização do negócio através de dados sobre ocupação de cadeiras em uma determinada sala de apresentação e ao cliente, uma experiência dinâmica e informativa ao fazer uso dos dados obtidos pelos sensores.

Palavras-chave: Internet das coisas. Otimização. Sensor.

ABSTRACT

Internet of things (IoT) is the modern term that describes the interconnectivity of objects due to new technologies presenting itself as an area of expansion for businesses to optimize their processes and offer new experiences. This paper aims to develop a complete IoT system to promote a better experience for theater and cinema businesses. This is carried out through sensors installed on a chair that transmits data by internet to a website developed in Node.js. The developed system shows how IoT can be used in a business and provides the manager with relevant information that can be used for optimization through data of seat occupancy in a specific presentation room and the customer with a dynamic and informative experience when making use of the data obtained by the sensors.

Key Words: Internet of things. Optimization. Sensor.

¹ Docente do curso de Análise e Desenvolvimento de Sistemas Fatec – Botucatu. Av. José Ítalo Bacchi, s/n - Jardim Aeroporto, Botucatu - SP, 18606-851.e-mail: ricardo.rall@fatec.sp.gov.br.

² Graduando curso de Análise e Desenvolvimento de Sistemas – Fatec Botucatu.

1. INTRODUÇÃO

Internet das coisas (*Internet of Things*, IoT) é um termo que descreve a conexão entre pessoas, dados, processos e objetos físicos com máquinas e aparelhos, a fim de apresentar melhores experiências e oportunidades econômicas para empresas ou indivíduos. O termo vem sendo usado desde 1999, como um exemplo da mudança que a tecnologia traz e nos últimos anos vem ganhando popularidade como o termo que descreve a interconectividade do mundo moderno, graças às aplicações modernas do conceito (KILARI, 2017).

Segundo a associação *Economist Intelligence Unit* (EIU, 2020), IoT é uma rede de objetos físicos interagindo entre si através da internet, podendo ser descrita em suas cinco principais capacidades: conectar dispositivos encurtando a diferença entre o mundo físico e o digital, coletando dados no objeto através de sensores, monitoramento remoto que permite uma visão do objeto em tempo real, monetização através da coleta e análise de dados que apresenta novas oportunidades de lucro para empresas seja agregando valor ou com novas funcionalidades para o cliente e a otimização, aumentando eficiência ao apresentar possível formas de poupança em gastos, tempo ou trabalho.

É estimado que IoT aplicada por empresas irá gerar 1.1 trilhão de dólares em receita adicional em empresas no mundo todo até 2025.

A arquitetura de um sistema IoT em sua forma mais básica é composto por 3 camadas de ação. A primeira camada de percepção ou de sensores é onde dados reais são coletados por sensores, sejam esses parâmetros físicos ou a detecção de outros dispositivos no ambiente; a camada de *network*, responsável pelo envio dos dados coletados na primeira camada, dispositivos e protocolos de rede fazem parte dessa camada; e a camada de aplicativo que define os serviços provido ao usuário através do processamento dos dados coletados como por exemplo, casas inteligentes, relógios inteligentes, etc (SETHI; SARANGI, 2017).

Em IoT, a *gateway* (ou porta) é o dispositivo responsável pela coleta e despacho dos dados coletados pelos sensores e para esta função, microcontroladores podem ser usados, fazendo a conexão com a internet e integrando lógica e processamento de dados no processo, para um envio inteligente, aumentando eficiência do sistema (WU; QIU; ZHANG, 2020).

O número de aplicações para IoT em diversos tipos de indústrias é grande e continua a crescer em fabricas, hospitais, fazendas, varejo, casas e veículos. Por exemplo, no caso das fabricas BMW, segundo estudo realizado pela empresa McKinsey (2021), o uso da interconectividade de um sistema IoT nos robôs em sua linha de produção reduziu tempo ocioso não planejado de suas máquinas em 25%. Conforme o avanço da tecnologia e

crescimento da internet, é aparente que empresas continuarão a buscar novas maneiras de melhorar seus processos, isso inclui a maior adoção da IoT que por consequência continuará a influenciar nossas vidas, cada vez mais no futuro.

Segundo a ANCINE (2020), sistemas de controle de ocupação de assentos nos cinemas permitem um gerenciamento mais eficiente dos assentos disponíveis em cada sessão. Isso ajuda a otimizar a ocupação das salas, evitando superlotação ou assentos vazios.

A alocação adequada de assentos também contribui para uma experiência mais agradável para os espectadores, reduzindo a confusão e o estresse na busca por lugares.

Esse artigo teve como objetivo apresentar um protótipo IoT de uma cadeira inteligente que capta dados por meio de sensores e os manda para um website via WiFi, sendo um sistema desenvolvido para teatros ou auditórios que provem dados de ocupação em salas de apresentação e um website desenvolvido para o negócio com informação de filmes e sessões e uma página dinâmica mostrando os assentos em tempo real pelo website.

2. MATERIAL E MÉTODOS

materiais usados podem ser divididos em duas categorias, hardware para materiais físicos e software, para digitais.

Os hardwares usados foram Desktop Asus X570 com processor Ryzen 7 3800X e Windows 10, quatro células de carga 50kg, um módulo conversor HX711, um microcontrolador ESP-12E NodeMCU V3, cabos para conexões, uma chapa de madeira 25cm por 30cm, fita isolante e quatro peças de plástico de suporte para as células de carga. Os softwares usados foram bibliotecas e pacotes diversos, entre os quais, *Visual Studio Code*, *Git Bash*, *Node.js*, *MongoDB*, *Arduino IDE*, *Express*, *Mongoose*, *Bootstrap*, *Joi*, *Ejs*, *Nodemon*, *Passport*, *Mapbox*, *Cloudinary*, *Paginate*, *HX711_ADC* e *Esp8266 Arduino Core*.

2.1 SOFTWARE

2.1.1 Linguagens

As principais linguagens usadas no desenvolvimento web desde projeto foram *HyperText Markup Language (HTML)*, *Cascading Style Sheets (CSS)* e *JavaScript*. *HTML* define os elementos que existem em uma página web; *CSS* define a aparência e como esses

elementos são estruturados na tela do usuário, atribuindo propriedades a eles, a fim de modificar sua aparência e JavaScript, que atribui lógica e ação aos elementos (EIS, 2015). Por exemplo, uma atribuição *on click* que executa algum código JS quando um elemento é clicado, esse tipo de atribuição é uma forma de se fazer páginas web dinâmicas e responsivas, através do *Documento Object Model* (DOM), que é a representação dos elementos HTML em objetos JS. O DOM é criado pelo browser quando lê um documento HTML. Neste projeto, JS foi usado também como a linguagem utilizada pelo servidor e não somente como *scripting*, para as páginas web.

2.1.1 Visual Studio Code

Visual Studio Code foi o editor de código utilizado e permite o uso de extensões e temas para a customização a gosto do usuário. Por padrão Visual Studio Code possui suporte para as linguagens CSS, HTML e JavaScript, que foram usadas no projeto assim como completador de código inteligente para JavaScript e *debug* para Node.js (NEGI, 2023).

2.1.2 Git Bash

Git Bash é o terminal que foi usado no projeto para navegação de arquivos, instalação de pacotes e a execução do projeto. É um programa que vem incluído no pacote Git para Windows. Git é um conjunto de programas utilitários de linha de comando projetado para ambientes de linha de comando baseados em UNIX. Em ambientes Windows, o Git Bash fornece uma camada de emulação UNIX para que seja possível utilizar um terminal Git fazendo o uso de comandos de linha no Windows (KEAN, 2022).

2.1.3 Node.JS

Node.js foi o ambiente de execução usado para o servidor, é uma plataforma interpretadora de código JavaScript para executar aplicações que faz o uso da linguagem JavaScript para toda a montagem *back-end*, incluindo todas as rotas e controladores *get*, *post*, *put*, *delete* e a respectiva lógica para cada um, a requisição e uso de todos os módulos assim como os middlewares de validação, autenticação, autorização e de inicialização e

configuração de módulos, específico ou não à alguma rota, tratamento de erros, conexão com a base de dados e inicialização do servidor web (HAHN, 2016).

2.1.4 Mongodb

Mongodb é a base de dados que foi utilizada neste projeto para a persistência dos dados de usuários, filmes e sessões do projeto localmente. É uma base de dados NoSQL, composta por documentos, que é o nome que se dá a estrutura de dados do banco e por coleções, que é conjunto desses dados estruturados. Em Mongodb, os documentos são salvos e lidos no banco de dados como objetos JavaScript oferecendo uma maior performance e flexibilidade nas operações, quando usado o mesmo formato no aplicativo. Mongodb é projetado sem esquemas, ou seja, não precisa seguir uma estrutura de dados rígida, somente aquelas definidas a nível de aplicação permitindo o desenvolvimento ágil a se adequar a mudanças na estrutura dos dados (D'MELLO, 2017).

2.1.5 Arduino IDE

Arduino Integrated Development Environment (IDE) é um ambiente de desenvolvimento para microcontroladores Arduino usado para a criação e upload de código no microcontrolador. Através do uso de bibliotecas de código, o software pode ser adaptado para uso em diversos outros tipos de microcontroladores e arquiteturas (FEZARI; DAHOUD, 2018). A biblioteca esp8266 pode ser usado para programar e fazer upload em placas baseadas no chip ESP8266 como é o NodeMCU, usado nesse projeto.

2.1.6 Pacotes e Bibliotecas

Os pacotes usados no website foram instalados fazendo o uso do *Node Package Manager* (NPM), que é um recurso instalado junto ao Node.js, gerenciando e facilitando a instalação de pacotes em projetos modulares baseados em Node, pelo livre e fácil acesso a código submetido por outros desenvolvedores através de linhas de comando, sendo uma forma rápida e conveniente para programadores compartilhar código, apresentando, aproximadamente, 3 milhões de downloads na plataforma por semana, em 2018 (LEBLANC, 2018).

Express, um *framework* para Node.js, corresponde a URL e o método de um pedido HTTP a funções a serem executadas em sequência pelo servidor, e depende dessas funções para determinar como será a resposta ao pedido feito, essas funções são chamadas de *middleware* significando uma função que é executada no “meio” de um pedido, entre o começo, onde ele é feito pelo cliente e o fim, onde o servidor dá a resposta ao cliente. Uma aplicação Express em si podendo ser simplificada como uma série de funções *middlewares* que realizam ações e analisam dados no servidor até o fim do pedido quando é dada a resposta ao cliente geralmente na forma de um documento HTML a ser renderizado pelo browser do cliente (SUBRAMANIAN, 2017, p.75).

Mongoose é um *Object Data Mapper* (ODM) para Node.js e Mongo. ODMs são responsáveis pela conexão entre uma aplicação e um banco de dados através da modelagem dos dados que são inseridos e lidos do banco de dados, modelando esses dados para que possam ser trabalhados dentro da aplicação Node.js, permitindo adicionar métodos e validações na forma de esquemas, que é nome da estrutura para os dados dentro de um documento. O princípio chave do mongoose é permitir trabalhar com um modelo dos dados dentro da aplicação e não diretamente no banco de dados. O modelo sendo o esquema compilado dentro da aplicação, em MongoDB, os dados são guardados em conjuntos chamados de documentos e coleções, sendo uma coleção o conjunto de documentos. O mongoose define os dados de documentos em esquema, sendo cada dado definido no esquema, chamado de caminho (HOLMES; HARBER, 2019).

Bootstrap é um *framework* de CSS, contendo um conjunto de *templates* CSS e HTML para estilizar de páginas HTML e plugins JavaScript adicionais que adicionam interatividade a esses componentes com um sistema grade baseado em colunas e linhas para o layout de componentes na tela, desenvolvido para acomodar uma página web para telas de diversos tamanhos (CHIARETTA, 2017, p.50). Bootstrap é utilizado através de classes que são atribuídas aos elementos que atribuem a aquele elemento o design específico do Bootstrap. Neste projeto, grande parte do CSS e parte do HTML usados foram adaptados dos modelos disponíveis na documentação do Bootstrap.

Joi é uma ferramenta de validação de dados através de um esquema joi de validação, como por exemplo se é número ou *string*, seu tamanho mínimo e máximo, se é obrigatório ou não, entre outros (HARRISON, 2016). Aqui foi usado para verificar dados de formulário que seriam inseridos no banco de dados.

Embedded JavaScript (Ejs) é um *template engine* para JavaScript, responsável por criar código HTML através de um *template* com lógica implementada capaz de multiplicar

ou modificar a *template* a fim de criar a página final que o usuário vê na tela. O fato de ser integrado permite adicionar a linguagem JS junto ao HTML para executar lógica a partir de tags como `<% %>` ou `<%= %>` (OLUSOLA, 2021).

Nodemon é uma ferramenta para aplicações Node.js que reinicializa a aplicação toda vez que mudanças nos arquivos forem detectadas, facilitando o desenvolvimento e automatizando esse processo de reinicialização (CERON, 2021).

Passport é uma biblioteca para autenticação e login persistente, é responsável por criptografar e descriptografar a senha de usuários junto com um salt. Pode ser configurado com diferentes tipos de estratégias, como login com conta Facebook ou Twitter. Foi usado junto com as bibliotecas `passport-local` para estratégia local e `passport-local-mongoose` que faz a integração `passport` com `mongoose` adicionando senha e usuário ao esquema `mongoose` automaticamente (WEXLER, 2019). É salvo na base de dados o usuário, o salt e a senha criptografada com o algoritmo de criptografia `pbkdf2`.

Cloudinary é um serviço *cloud* para imagem e vídeos, que possibilita o upload e o subsequente acesso a imagens pelo website.

Mapbox é uma empresa provedora de mapas customizados para websites e seus serviços de *geocoding* foram usados para converter dados em coordenadas, com a biblioteca JavaScript `Mapbox GL JS` para renderizar mapas em páginas web.

Paginate é um módulo que facilita a implementação de paginação realizando a lógica necessária para a paginação através de uma função que retorna HTML pronto para ser implementado na página.

HX711_ADC é uma biblioteca de livre uso desenvolvida por Olav Kallhovd disponível para download no GitHub ou no gerenciador de bibliotecas do ArduinoIDE, sendo usada para buscar e calibrar dados do conversor HX711. Essa biblioteca dispõe de alguns exemplos de código para teste e calibração de uma ou várias células de carga com opção de guardar os dados de calibração na memória EEPROM do microcontrolador.

Esp8266 Arduino Core é o nome de um conjunto de bibliotecas para Arduino IDE que faz suporte para microcontroladores baseados no chip ESP8266 e vem incluso com uma variedade de bibliotecas para se fazer comunicação WiFi e criação de servidor HTTP, DNS, entre outros (SANTOS, [2015?]). Para este projeto, foram usadas as bibliotecas `ESP8266WiFi`, `ESP8266HTTPClient` e `WiFiClient` desse conjunto para se fazer a conexão via WiFi com o servidor do website, o conjunto é de livre uso e disponibilizado pela comunidade `esp8266` pelo GitHub.

2.2 Hardware

2.2.1 NodeMCU

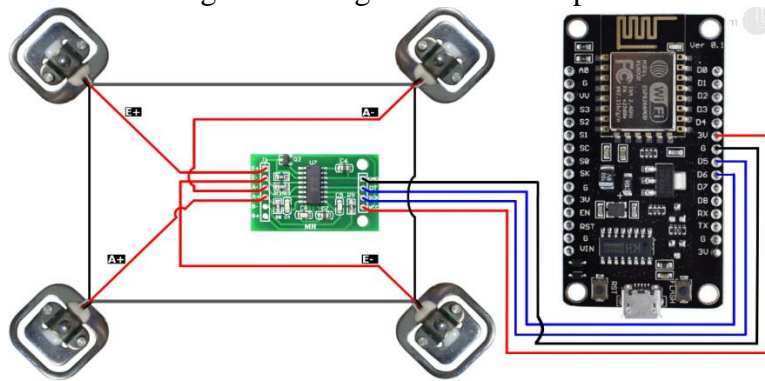
NodeMCU é uma placa de prototipagem, contendo o módulo ESP12 que possui o chip microcontrolador ESP8266 e outros componentes necessários para o seu funcionamento, como uma antena para captar frequências WiFi, sendo o sistema todo baseado no microcontrolador ESP8266 que possui capacidade para network TCP/IP e capacidades de processamento. NodeMCU é uma placa que se destaca para projetos IoT devido ao seu suporte integrado a redes WiFi, baixo custo, baixo consumo de energia e tamanho reduzido, tudo isso com a capacidade de ser programada em Arduino IDE (OLIVEIRA, 2016).

2.2.1 HX711 e células de carga

HX711 é um módulo conversor analógico para digital e amplificador de sinal para uso em conjunto com células de carga fazendo interface dos sensores para o microcontrolador. Sensor de carga é o dispositivo capaz de converter força mecânica em sinais elétricos possibilitando a medição do peso sendo aplicado sobre ele (AGNIHOTRI, [2022?]). Foram usadas quatro células de carga tipo meia ponte de 50kg, conectando as quatro células em um loop pelos fios pretos e brancos e a fios intermediários vermelhos completando uma ponte *Wheatstone*, permitindo medir uma diferença na resistência dos sensores de peso resistivos que é então convertida a um valor digital no HX711.

A Figura 1 mostra as células de carga, o módulo HX711 e a placa NodeMCU e como a conexão entre eles foi feita através do diagrama, com as células conectando entre si em loop e os fios intermediários vermelhos conectando ao módulo HX711 com fios para transmissão de energia (E+, E-) e fios para transmissão de dados (A+, A-) e os fios de saída do HX711 conectando ao microcontrolador como VCC para 3V, GND para G, DT e SCK para D6 e D5 respectivamente.

Figura 1 - Diagrama do Protótipo

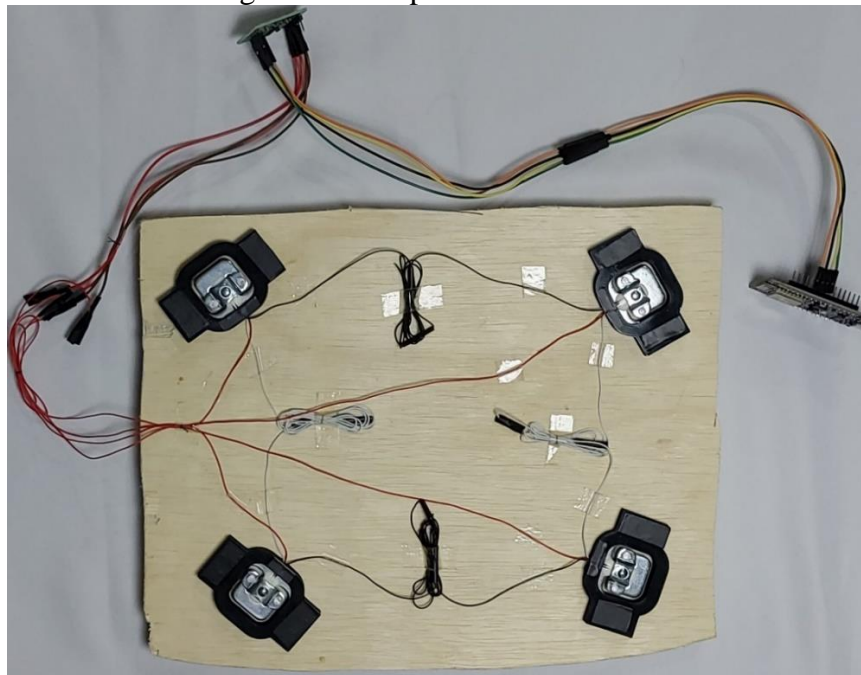


Fonte: Adaptado de Luuk (2020).

2.2.1 COMPONENTES DIVERSOS

Foram usados cabos e solda extras para conectar os componentes eletrônicos. Quatro peças de plástico feitas para suporte as células de carga foram fabricadas e fixadas com parafusos a uma chapa de madeira com dimensões aproximadas de 30cm de comprimento, 25cm de largura e 2cm de espessura podendo ser usada como assento em uma cadeira desde que os quatro pontos de contato das células tenham suporte estável na cadeira. A Figura 2 mostra os componentes instalados na chapa com as células fixadas, fios amarrados e fixados a chapa e conexões soldadas isoladas com fita isolante.

Figura 2 - Componentes Instalados



Fonte: Elaborado pelo Autor.

3. RESULTADOS E DISCUSSÃO

O website conta com página home de entrada no URL vazio do site, com botões para as principais funcionalidades para o usuário comum, em sessões, login ou criar conta. Criar conta leva para uma página com um formulário a ser preenchido com usuário, e-mail e senha persistindo o login após a conta ser criada. Todas as páginas HTML do website são iniciadas através de um arquivo HTML comum à todas as páginas (*boilerplate*), contendo *header* que contém metadados como *scripts* bootstrap e mapbox e outras informações relevantes para o browser. Logo em seguida se inicia o *body* ou corpo da página, com a barra de navegação e o pé da página, entre eles, é incluída a variável que importa o código da página específica que está sendo aberta.

O website possui duas partes: páginas acessáveis por administradores e páginas acessáveis por usuários, sendo páginas de criação, edição e remoção de sessões e filmes e criação de contas administrativa somente acessíveis por administradores, enquanto página index e de detalhes de todas as sessões são sempre visíveis e a página mostrando disponibilidade dos assentos em uma sessão acessível somente para usuários logados.

A página index é acessada através do botão “Todas as sessões” no *dropdown* “Sessão” a partir da barra de navegação, nela podem ser vistas as sessões com o nome, poster e gênero do filme, dia e hora de início e fim da sessão e um botão para a página de detalhes da sessão. Cada sessão no index com suas informações agrupadas em um elemento *div* e listadas verticalmente na página e paginadas com no máximo 5 sessões por página.

A página de detalhes de uma sessão oferece dados extras na forma de uma sinopse do filme, o tempo de execução e local do cinema com um mapa interativo marcado com o local assim como acesso a página de assentos da sessão para usuários logados, conforme Figura 3.

Figura 3 - Página de Detalhes da Sessão

Interestelar

Uma equipe de exploradores viaja através de um buraco de minhoca no espaço, na tentativa de garantir a sobrevivência da humanidade.

Aventura, Drama, Ficção científica

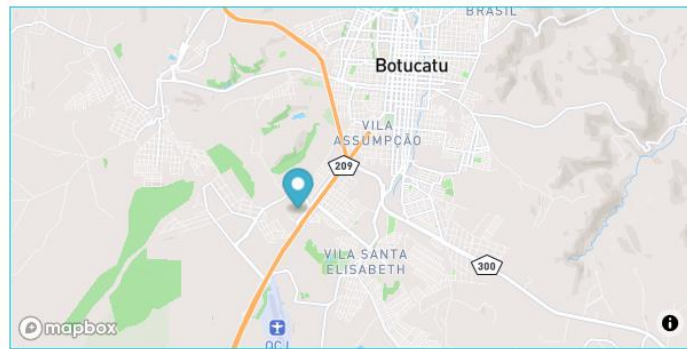
Runtime: 75 min

Data: Quarta-feira, 21 de junho de 2023

Início da Sessão: 10:10:00

Fim da Sessão: 11:45:00

Local: Avenida Marginal Duzentos 1050, 18606294



Assentos Disponíveis

Fonte: Elaborado pelo Autor.

A página com os assentos da sessão mostra todos os assentos de uma sessão e sua numeração através de imagens, com legenda, onde uma cadeira preta está livre, amarela alugada, ou seja, relacionada a um usuário dentro do site, e a vermelha ocupada, ou seja, os sensores na cadeira estão detectando peso. Por ser uma página dinâmica, ela checa as atualizações dos estados das cadeiras a cada 15 segundos através de uma chamada ao servidor. A página permite a um usuário logado “alugar” uma cadeira ao clicar nela e nesse caso, por ser apenas uma versão de desenvolvimento, essa ação apenas associa o id do usuário com aquela cadeira, conforme Figura 4 que mostra a página de assentos sobreposta com o monitor serial do Arduino IDE, mostrando os dados sendo detectados pelos sensores, onde a captar 63000, ou seja, 63kg nos sensores o pedido HTTP é feito com a resposta 200 do servidor, significando que ocorreu conforme esperado, com a cadeira então ficando vermelha como resultado.

Figura 4 - Página de Assentos da Sessão e Terminal Serial



Os sensores do assento detectam e transmitem os dados para o módulo conversor que os transmitem para o microcontrolador que então faz uma chamada POST ao servidor quando houver uma mudança maior que 25kg nos sensores e fizer mais de 30 segundos desde a última chamada, transmitindo dados de peso, número da cadeira, id da sessão e chave de segurança para dar acesso no servidor. Quando houver um peso maior que 25kg detectado, o servidor muda o estado da cadeira para “ocupado”. Toda mudança no estado de uma cadeira é arquivada no banco de dados com o tempo da mudança e para qual estado foi mudado. Esse tipo de informação pode ser útil para um gerente visto que mostra o nível de ocupação dos assentos em determinado momento e quais assentos são mais populares, segundo Figura 5.

Figura 5 - Código NodeMCU

```

if (millis() > t + serialPrintInterval) {
  float i = LoadCell.getData();
  Serial.print("Load_cell output val: ");
  Serial.println(i);
  Serial.println(data);
  if (millis() > (changedData * 1000)) {
    if (abs(data - i) > 25000) { //calcula a diferença absoluta do peso da ul
      if (WiFi.status() == WL_CONNECTED) { //checagem de conexao wifi
        data = i; //peso detectado na ultima transmissao
        if(data<1000||data>200000){data = 1;}
        changedData = (millis() / 1000) + 30; // tempo da ultima transmissao +
        WiFiClient client; //inicia conexao a internet como cliente atraves d
        HTTPClient http; //inicia conexao ao servidor web pelo protocolo HTTP
        http.begin(client, serverName); //conecta a URL especificada a partir
        http.addHeader("Content-Type", "application/x-www-form-urlencoded");
        // Dados a serem enviados
        String httpRequestData = "chairNum=";
        httpRequestData += String(1);
        httpRequestData += "&sensorData=";
        httpRequestData += String(i);
        httpRequestData += "&key=Vg_gZr8cQ5$nK#kbUq56e|l(17?AV0<8!9";
        httpRequestData += "&id=645cd0ef2c6da1e55c7bf725";
        int httpResponseCode = http.POST(httpRequestData); // Enviar pedido H
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        http.end(); // Libera recursos
      }
    }
  }
}

```

Fonte: Elaborado pelo Autor.

A Figura 6 mostra como os sensores são instalados em uma cadeira, sendo um suporte rígido o único requerimento em todos os quatro pontos de contato com as células de carga e uma fonte de energia para o microcontrolador, seja via micro USB ou pelo pino Vin do microcontrolador possibilitando uma configuração “sem fio”, sem conexões externas para energia, com energia fornecida por uma bateria. O microcontrolador possui também a capacidade de ter seu código modificado remotamente sem fio, caso seja configurado para isso.

Figura 6 - Sensor Instalado no Assento com Estofamento



Fonte: Elaborado pelo Autor.

4 CONCLUSÕES

O desenvolvimento desse projeto demonstrou o uso de um sistema IOT completo, dos sensores e networking e suas aplicações. O website desenvolvido apresentou funcionalidades para potenciais consumidores e para gerentes, divulgando sessões de cinema aos clientes assim como uma página dinâmica e informativa para escolha de assentos e ao gerente de negócio, deu o controle necessário para fazer mudanças no conteúdo apresentado ao cliente e um novo leque de informações apresentados na forma dos dados de ocupação, abrindo uma nova área para otimização do negócio a partir do uso de sensores e internet.

REFERÊNCIAS

AGNIHOTRI, Nikhil. *How load cells work*. **Engineersgarage**, [07 jan. 2022?]. Disponível em: <<https://www.engineersgarage.com/load-cells-types-working/>>. Acesso em: 23 jun. 2023.

AGÊNCIA NACIONAL DO CINEMA (ANCINE). **Distribuição em Salas - 2020**. 202121. Disponível em: <https://www.gov.br/ancine/pt-br/oqa/publicacoes/arquivos.pdf/distribuicao2020.pdf>. Acesso em: 08/04/2024.

CERON, Vitor. Nodemon: como configurar e utilizar. **Programando Soluções**, 27 abr. 2021. Disponível em: <<https://programandosolucoes.dev.br/2021/04/27/configurar-utilizar-nodemon/>>. Acesso em: 26 jun. 2023.

CHIARETTA, Simone. **Front-end Development with ASP-NET Core, Angular, and Bootstrap**. Wrox Press, 2017. 50 p.

D'MELLO, Bruno; SATHEESH, Mithun; KROL, Jason. **Web Development With MongoDB and Node**. 3. ed. Packt Publishing, 2017. 17 p.

EIS, Diego. **Guia Front-End: O caminho das pedras para ser um dev Front-End**. 1. ed. Casa do Código, 2015. 7-40 p.

FEZARI, M.; DAHOUD, A. **INTEGRATED-DEVELOPMENT ENVIROMENT “IDE” FOR ARDUINO. WSN applications**. Amman, p.1-12, out. de 2018.

HAHN, E. **Express in Action**. Manning Publications, 2016.

HARRISON, M. **Hapi.js in Action**. Primeira Edição. Manning Publications, 2016.

HOLMES, S.; HARBER, C. **Getting Mean with Mongo, Express Angular, and Node**. Segunda Edição. Manning Publications, 2019. 9 p.

KEAN, Kadeisha. What is Git and How Do You Use It?. **Makeuseof**, 24 jun. 2022. Disponível em: <<https://www.makeuseof.com/git-bash-what-how-use/>>. Acesso em: 21 jun. 2023.

KILARI, N. *Internet of Things for Business*. **International Journal Of Engineering And Computer Science**, Bangaluru, março de 2017. Disponível em: <https://www.researchgate.net/publication/315936935_Internet_of_Things_for_Business>. Acesso em: 2 abr. 2023.

LEBLANC, Daniel. *Getting Started with Node.JS and Node Package Manager NPM*. **Academia**, 2018. Disponível em: <https://www.academia.edu/37093010/Getting_Started_with_Node_JS_and_Node_Package_Manager_NPM>. Acesso em: 12 abr. 2023.

LUUK, Indrek. *50kg Load Cells with HX711 and Arduino. 4x, 2x, 1x Diagrams*. **Circuit Journal**, 17 abr. 2020. Disponível em: <<https://circuitjournal.com/50kg-load-cells-with-HX711>>. Acesso em: 27 abr. 2023.

MCKINSEY & COMPANY. *The internet of things: Catching up to an accelerating opportunity*. São Francisco, novembro de 2021.

NEGI, Vikas. *How to run Node.js files in VS Code: A Step-by-Step Guide*. **Codeforgeek**, 26 de fev. de 2023. Disponível em: <<https://codeforgeek.com/run-node-js-files-in-vs-code/>>. Acesso em: 22 jun. 2023.

OLIVEIRA, Greici. NODEMCU – Uma plataforma com características singulares para o seu projeto IOT. **MasterWalker**, 16 ago. 2016. Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-uma-plataforma-com-caracteristicas-singulares-para-o-seu-projeto-iot>>. Acesso em: 23 abr. 2023.

OLUSOLA, Samuel. *5 How to use EJS to template your Node.js application*. **Logrocket**, 24 dez. 2021. Disponível em: <<https://blog.logrocket.com/how-to-use-ejs-template-node-js-application/>>. Acesso em: 23 jun. 2023.

SANTOS, R. *Installing ESP8266 Board in Arduino IDE (Windows, Mac OS X, Linux)*. **Random nerd tutorials**, [21 set. 2015?]. Disponível em: <<https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/>>. Acesso em: 25 jun. 2023.

SETHI, P.; SARANGI, S.; *Internet of Things: Architectures, Protocols, and Applications*. **Journal of Electrical and Computer Engineering**, New Delhi, janeiro de 2017. Disponível em: <https://www.researchgate.net/publication/312957467_Internet_of_Things_Architectures_Protocols_and_Applications>. Acesso em: 4 abr. 2023.

SUBRAMANIAN, Vasan. **Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node**. Bangaluru: Apress, 2017. 75 p.

THE ECONOMIST INTELLIGENCE UNIT – EIU. *The internet of things: Applications for business*. Londres, maio de 2020. Disponível em: <https://euagenda.eu/upload/publications/18062020_cte-20report_final.pdf.pdf>. Acesso em: 4 abr. 2023.

WEXLER, J. *Get Programming with Node.js*. Manning Publications, 2019.

WU, Z.; QIU, K.; ZHANG, J.; *A Smart Microcontroller Architecture for the Internet of Things*. *Sensors*, Nanjing, março de 2020. Disponível em: <https://www.researchgate.net/publication/340182181_A_Smart_Microcontroller_Architecture_for_the_Internet_of_Things>. Acesso em: 2 abr. 2023.