

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Bruna Michele Correia Ribeiro, 2003356

Cynthia Mayumi W. Yamaoto, 2005192

Isael Rodrigues De Freitas, 2015634

Marcos Ribeiro De Oliveira, 2005638

Matheus Messias Borges De Brito, 2011463

Renato Nogueira Da Silva, 2009044

Tatiana Akemi Okumura, 2011129

Thyago Lima Temoteo Pereira, 2005293

Framework web de informação de disponibilidade de vagas em centros de acolhimento

Vídeo de apresentação do Projeto Integrador

< https://youtu.be/z8Y2Dcw9_cE >

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Framework web de informação de disponibilidade de vagas em centros de acolhimento

Relatório Técnico-Científico apresentado na disciplina
Projeto Integrador IV para o curso de Ciência de Dados e
Engenharia de Computação da Universidade Virtual do
Estado de São Paulo (UNIVESP).

Osasco – SP

2023

RIBEIRO, Bruna Michele Correia; YAMAOTO, Cynthia Mayumi Watanabe; FREITAS, Isael Rodrigues De; OLIVEIRA Marcos Ribeiro De; BRITO Matheus Messias Borges De; SILVA, Renato Nogueira Da; OKUMURA, Tatiana Akemi; PEREIRA, Thyago Lima Temoteo. **Framework web de informação de disponibilidade de vagas em centros de acolhimento.** XXf. Relatório Técnico-Científico. Ciência de Dados e Engenharia de Computação – Universidade Virtual do Estado de São Paulo Tutor: Alexandre Ceconello Marinho. Polo Osasco, 2023.

RESUMO

O presente projeto surgiu a partir da aspiração de auxiliar o atendimento à população em situação de rua, principalmente em época de baixas temperaturas, com o objetivo de desenvolver um framework web interoperável com um dispositivo eletrônico IoT físico para informar-lhes sobre a localização e a disponibilidade de vagas nos centros de acolhimento e de atendimento. Além de haver um botão para acionamento em caso de emergências que proporcionará a análise da demanda de situações extremas. Pretende-se realizar testes de indicação de temperatura e conectividade à internet com o protótipo; avaliar a funcionalidade do framework web, do dispositivo IoT e do botão de emergência junto à comunidade externa e colher feedbacks visando buscar adequações e melhorarias; implantar e disponibilizar o framework web para a área de assistência social da prefeitura de Osasco que fornece amparo a essa população. O Design Thinking foi utilizado como metodologia, já que essa auxilia no processo de escolha e possui a característica de colocar o ser humano no centro dos processos.

PALAVRAS-CHAVE: Assistência social, População em situação de rua; *Design Thinking*, *framework web*; Internet das coisas, Dispositivo *IoT*, Botão de Emergência, Dashboard.

LISTA DE ILUSTRAÇÕES

Figura 1. Ciclos da revolução industrial e tecnológica.....	7
Figura 2. Arquitetura de um framework web.....	11
Figura 3. Fluxograma ao implantar uma aplicação acessível.....	13
Figura 4. Etapas do Design Thinking.....	16
Figura 5. Modelo de placa ESP32.....	23
Figura 6. Modelo de sensor DHT11.....	23
Figura 7. Barramento I2C.....	24
Figura 8. IDE Arduino.....	25
Figura 9. Display LCD 16x2 e sensor DHT11.....	26
Figura 10. DHT11-sensor de temperatura e umidade.....	26
Figura 11. Display LCD 16x2 com módulo I2C.....	26
Figura 12. Disposição da placa ESP32 no dispositivo IoT.....	27
Figura 13. Placa de ensaio ou matriz de contatos.....	27
Figura 14. Padrão de acesso ao sistema.....	29
Figura 15. Acompanhamento de vagas nos centros de acolhimento.....	29
Figura 16. Cadastramento de centros e pessoas em situação de rua.....	30
Figura 17. Canais de comunicação com os centros de acolhimento.....	30
Figura 18. Estrutura de comunicação do sistema.....	31
Figura 19. Modelo de mensagem apresentada no dispositivo IoT.....	31
Figura 20. Entrada (login e cadastro) ao sistema do framework web.....	31
Figura 21. Padrão de acesso ao sistema do framework web.....	32
Figura 22. Cadastramento de usuário no sistema do framework web.....	32
Figura 23. Opção para visualizar características dos abrigos.....	33
Figura 24. Opção para atualizar dados de um abrigo.....	33
Figura 25. Opção para visualizar a distância entre os dispositivos e os centros.....	34
Figura 26. Atualização de senha.....	34

Figura 27. Ilustração do projeto.....	35
Figura 28. Códigos desenvolvidos no Github.....	36
Figura 29. Diagrama de interligação.....	37
Figura 30. Layout da vista frontal do dispositivo.....	38
Figura 31. Visualização API.....	39
Figura 32. Funcionamento do dashboard para análise de dados.....	40
Figura 33. Código e execução dos Testes.....	41
Figura 34. Visão geral do protótipo conectado via rede wi-fi.....	48
Figura 35. Sensor DHT de temperatura.....	48
Figura 36. Visão após o botão ser acionado.....	49
Figura 37. Em caso de sucesso, o LED piscará por 40s.....	49
Figura 38. Em caso de falha, o LED piscará por 40s.....	50
Figura 39. Visão do Dashboard.....	50

SUMÁRIO

1	INTRODUÇÃO.....	1
2	DESENVOLVIMENTO.....	4
2.1	Objetivos.....	4
2.2	Justificativa e delimitação do problema.....	4
2.3	Fundamentação teórica.....	6
2.3.1	Internet das coisas.....	7
2.3.2	Cidades inteligentes.....	8
2.3.3	Ferramentas Aplicadas ao Framework Web.....	9
2.3.4	Acessibilidade nas aplicações.....	11
2.3.5	Diretrizes de acessibilidade na Web.....	11
2.3.6	WCAG.....	12
2.4	Aplicação das disciplinas estudadas no Projeto Integrador.....	13
2.4.1	Fundamentos de Internet e Web (COM130).....	13
2.4.2	Desenvolvimento Web (COM320).....	13
2.4.3	Algoritmos e Programação de Computadores I e II (COM110/120).....	14
2.4.4	Protocolos de Comunicação IoT (COM380).....	14
2.4.5	Engenharia de Software (COM390).....	15
2.5	Metodologia.....	15
2.5.1	Ouvir e Interpretar o Contexto.....	16
2.5.2	Criar e Prototipar.....	17
2.5.3	Implementar e Testar.....	18
3	SOLUÇÃO INICIAL.....	22
3.1	Componentes técnicos para a construção do dispositivo IoT.....	22
3.1.1	ESP32 (WiFi / Bluetooth).....	22
3.1.2	DHT11 – Sensor de Temperatura e Humidade.....	23

3.1.3	Inter-Integrated Circuit (I2C) – Módulo e Protocolo.....	24
3.1.4	Integrated Development Environment (IDE) - Arduino.....	24
3.2	Montagem do dispositivo IoT.....	25
3.3	Framework Web.....	27
3.3.1	Interoperabilidade framework web e IoT.....	30
4	SOLUÇÃO COMPLEMENTAR.....	35
4.1	Desenvolvimento e Performance.....	35
4.1.1	Dispositivo IoT.....	36
4.1.2	API (Back-end).....	38
4.1.3	Dashboard (Front-end).....	39
4.1.4	Testes para Validação e Performance.....	40
5	CONCLUSÕES.....	42
	REFERÊNCIAS.....	43
	APÊNDICE – DISPOSITIVO <i>IoT</i>.....	47

1 INTRODUÇÃO

Dias inteiros sem tomar banho, a roupa suja ou rasgada - que muitas vezes fica grande ou pequena para eles - a fome, os vícios, a exposição à violência e um denominador comum: um olhar de dor que a gritos pedem ajuda. Em uma observação simples, a rua parece igualar a dor dos que vivem e dormem nela, contudo atrás de cada um se esconde uma história diferente, uma razão que revela porque a vida os deixou ao desamparo, sem um teto que os proteja, e porque é tão necessário que a sociedade e as autoridades públicas trabalhem para recuperar a sua dignidade.

De acordo com Andrade, Costa e Marquetti (2014, p. 1249), no Brasil há pessoas em situação de rua desde o período colonial, com o início de processo de libertação dos escravos. Esse número aumentou com a chegada dos imigrantes camponeses advindos da Europa.

Nessa época, era costume classificar esses novos moradores do perímetro urbano como “vagabundos”, distinguindo-os dos demais não pela questão do domicílio, e sim pela integração ou não no mercado de trabalho. A vagabundagem tornou-se crime, no século XIX, e a lei era utilizada para classificar os moradores de rua. (JUSTO, 2008).

Tal fato histórico tornou propícias as condições de desigualdade que se tem materializado nas situações de exclusão, invisibilidade, discriminação e marginalização, assim como na construção de preconceitos e estereótipos dessa parcela da sociedade.

A invisibilidade histórica e a marginalização a que esses indivíduos ou grupos populacionais são submetidos tem-se tentado reverter paulatinamente mediante a realização e implantação de políticas públicas, legislações e ações específicas que revertam as condições de desigualdade e discriminação que enfrentam, ao mesmo tempo que atendam suas necessidades particulares e assegurem as condições mínimas para o exercício pleno dos seus direitos.

O Ministério do Desenvolvimento Social e Combate à Fome define em termo geral a população em situação de rua da seguinte maneira:

Grupo populacional heterogêneo, caracterizado por sua condição de pobreza extrema, pela interrupção ou fragilidade dos vínculos familiares e pela falta de moradia convencional regular. São pessoas compelidas a habitar logradouros públicos (ruas, praças, cemitérios etc.), áreas degradadas (galpões e prédios abandonados, ruínas etc.) e, ocasionalmente, utilizar abrigos e albergues para pernoitar. (BRASIL, 2008, p.8)

O Decreto 7.053 de 23 de dezembro de 2009 define a população em situação de rua como:

Grupo populacional heterogêneo que possui em comum a pobreza extrema, os vínculos familiares interrompidos ou fragilizados e a inexistência de moradia convencional regular, e que utiliza os logradouros públicos e as áreas degradadas como espaço de moradia e de sustento, de forma temporária ou permanente, bem como as unidades de acolhimento para pernoite temporário ou como moradia provisória. (BRASIL, 2009, p.1).

Nesse projeto integrador, voltou-se especificamente para pensar a necessidade dos moradores de rua da cidade de Osasco, uma vez que grande parte dos integrantes do grupo reside nessa região. É escassa a literatura a respeito desses moradores no que tange essa região, por isso buscou-se informações em jornais e sites do município. Em 2020, o *Jornal Giro*¹, publicou que a Prefeitura de Osasco avalia que existam 500 moradores de rua no município, sendo Osasco a cidade com mais moradores de rua da região.

O olhar do grupo voltou-se para os moradores de rua desse município, em especial nos meses de inverno intenso, nos quais as dificuldades de quem não tem um teto se tornam ainda mais severas na questão de sobrevivência devido às baixas temperaturas.

De acordo com informações publicadas no site da prefeitura de Osasco², no inverno, a Secretaria de Assistência Social realiza abordagens noturnas, a partir das 20h, de pessoas em situação de rua sempre que a previsão do tempo indica que fará 10 graus ou menos.

Como tal Secretaria já conhece os locais da cidade onde se concentra essa população e sempre verifica outros que venham a surgir, as visitas são específicas nesses locais e são distribuídos cobertores, roupas e chocolate quente para aqueles que rejeitarem ir a um dos três centros de acolhimento da cidade. Além disso, qualquer residente da cidade de Osasco pode entrar em contato com a prefeitura pela Central 156 para comunicar sobre pessoas em situação de rua que necessitam de ajuda.

As abordagens são realizadas por duas equipes, uma na Zona Norte e outra na Zona Sul da cidade, sendo composta por um assistente social e dois bombeiros civis.

A cidade conta com três serviços de acolhimento (dois no Rochdale, sendo um para homens e outros para mulheres com filhos ou sem filhos, e outro no Centro, apenas para homens) e com o Centro Pop (Centro de Referência Especializado para a População em Situação de Rua), que funciona durante o dia, contudo nos dias de frio intenso abre à noite e disponibiliza colchões e cobertores.

¹“Região: Mais de 500 pessoas moram nas ruas de Osasco”. *Giro*, 2020. Disponível em: <<https://girosa.com.br/regiao-mais-de-500-pessoas-moram-nas-ruas-de-osasco/>>. Acesso em: 15 mar. 2023.

² “Osasco terá operação para abrigar pessoas em situação de rua”. *Marco Borba*, 2022. Disponível em <https://osasco.sp.gov.br/osasco-tera-operacao-para-abrigar-pessoas-em-situacao-de-rua/>. Acesso em: 15 mar. 2023.

Observando esses serviços públicos municipais e o distanciamento, que muitas vezes existe entre eles e os moradores de rua, o grupo questionou como a assistência a esses moradores poderia acontecer de forma mais efetiva. Sendo assim, pensou-se no seguinte problema: Como melhorar e ampliar o canal de comunicação entre a população em situação de rua e os centros de acolhimento no sentido de fornecer publicamente informações sobre a localização e a disponibilidade de vagas nesses centros?

Dessa forma, esse projeto pretende auxiliar o atendimento às pessoas em situação de rua, por meio do desenvolvimento de um framework web com o uso de um dispositivo físico aplicando técnicas, conceitos e recursos relativos ao cenário da Internet das Coisas (IoT – Internet of Things), uma vez que este tipo de tecnologia pode fazer a diferença, no contexto em que é inserido:

A Internet das Coisas é um paradigma que tem como objetivo criar uma ponte entre acontecimentos do mundo real e as suas representações no mundo digital. O objetivo é integrar o estado das Coisas que constituem o nosso mundo em aplicações de software, beneficiando do contexto onde estão instaladas (VALENTE, 2011, p. 1).

O intuito de IoT é aprimorar o ambiente em que está inserido, uma vez que os benefícios de usar essa tecnologia em prol de pessoas que vivem em vulnerabilidade motivou a escolha da temática desse projeto.

2 DESENVOLVIMENTO

2.1 Objetivos

O objetivo geral é desenvolver um framework web interoperável com um dispositivo eletrônico IoT para informar publicamente e em tempo real, a temperatura ambiente, a localização e disponibilidade de vagas nos centros de acolhimento à população em situação de rua. Adicionalmente será implantado um botão para atendimento emergencial que além de auxiliar prontamente, também colherá estatísticas de sua utilização.

Os objetivos específicos buscam definir e delimitar as ações a serem realizadas para que o objetivo geral seja atingido e, conseqüentemente, a conclusão do projeto. Portanto, foram estipulados os seguintes objetivos específicos:

- Analisar as possíveis linguagens de programação a serem utilizadas para as camadas de *front-end* e *back-end*;
- Fazer um levantamento dos requisitos gerais para definir as funcionalidades do framework web e dispositivo IoT;
- Integrar o website com um framework para facilitar em melhorias futuras;
- Analisar os dispositivos e objetos mais adequados para a montagem do protótipo IoT;
- Levantar informações sobre serviços de hospedagem na nuvem;
- Hospedar o website em um servidor na nuvem para reduzir o tempo de processamento dos dados nas páginas;
- Utilizar uma API para a comunicação com o website desenvolvido;
- Realizar testes de indicação de temperatura e conectividade à Internet com o protótipo;
- Avaliar a funcionalidade do framework web e do dispositivo IoT junto a comunidade externa e colher *feedbacks* com o objetivo de buscar adequações e melhorias;
- Garantir que as aplicações estejam de acordo com as regras de acessibilidade;
- Gerar estatísticas do acionamento do dispositivo “botão de emergências”.
- Fornecer uma interface para visualização dos dados gerados pelo dispositivo (*dashboard*).

2.2 Justificativa e delimitação do problema

Como mencionado, a população em situação de rua emerge como um reflexo da exclusão social, que se caracteriza por vários fatores políticos, sociais e econômicos que se

manifestam em um cenário de desigualdades e vulnerabilidades sociais. As condições de vida dessa população em situação de rua são bastante precárias: as necessidades básicas de sobrevivência não são atendidas, acarretando um amplo processo de precariedade, discriminação, vulnerabilidade e relacionamento, aspectos considerados primordiais de aceitação e adaptação às regras estabelecidas para a convivência em uma sociedade, configurando-se no processo de desigualdade e exclusão social (SCHWEITZER; TOLFO, 2022; MENDES; RONZANI; PAIVA, 2019; RESENDE; MENDONÇA, 2019).

Silva et al. (2018) ressaltam que uma característica marcante da população em situação de rua relaciona-se aos aspectos itinerantes, onde em uma trajetória de viagens, percorrida de cidade em cidade, são construídos histórias, memórias e cenários que são vivenciados durante a sua jornada de vida.

Com a finalidade de ajudar a amenizar essas condições precárias algumas entidades de assistência social, públicas e privadas, oferecem alguns serviços de acolhimento e assistência a essa população marginalizada. Todavia, pode-se inferir que ainda há um déficit de políticas socioassistenciais para essa população, mas também de soluções criativas e inovadoras que possam contribuir, mesmo que minimamente, para melhorar o acesso às condições básicas de sobrevivência como moradia, alimentação e higiene pessoal, por exemplo.

A partir de uma ideia embrionária que foi explorada nas reuniões virtuais realizadas pelo grupo e em uma conversa inicial com as Sras. Danielle e Jessaminy, do Departamento de Proteção Social Especial da Secretaria de Assistência Social da Prefeitura de Osasco, verificou-se que a quantidade de vagas nos centros de acolhimento não é suficiente para atender a toda a população em situação de rua, que não há um controle adequado dessas vagas e a divulgação delas junto a essa população não é efetiva, o que pode causar alguns embargos. Contudo, uma das condições essenciais para que a assistência possa ser executada de forma mais efetiva é que a população em situação de rua busque atendimento em algum centro de acolhimento. A Secretaria de Assistência Social contata essa população em situação de rua por meio de abordagens em locais de alta concentração e, também, a Central 156 é disponibilizada para que qualquer munícipe possa informar sobre essa população (BORGA; GOBATTI, 2022; LIOTTA; CAZARINI, 2022). A título de informação, a cidade de Osasco dispõe de três centros de acolhimento efetivos e em períodos de inverno novos centros são disponibilizados (BORGA; GOBATTI, 2022; LIOTTA; CAZARINI, 2022).

Com base no contexto apresentado e para que a assistência possa ser mais organizada e efetiva nos centros de acolhimento, levantou-se o seguinte problema:

Como melhorar e ampliar o canal de comunicação entre a população em situação de rua e os centros de acolhimento no sentido de fornecer publicamente informações sobre a localização e a disponibilidade de vagas nesses centros? Além disso, providenciar o rápido socorro em uma situação de emergencial?

Pereira e Silva (2010) contextualizam que a tecnologia da informação e comunicação (TIC) é uma ferramenta que tem gerado mudanças significativas nos processos de desenvolvimento social, cultural e econômico. A diminuição do custo de aquisição dos recursos tecnológicos, principalmente, aqueles atrelados à Internet, têm gerado um grande impacto nas formas de acessibilidade e no desenvolvimento da sociedade.

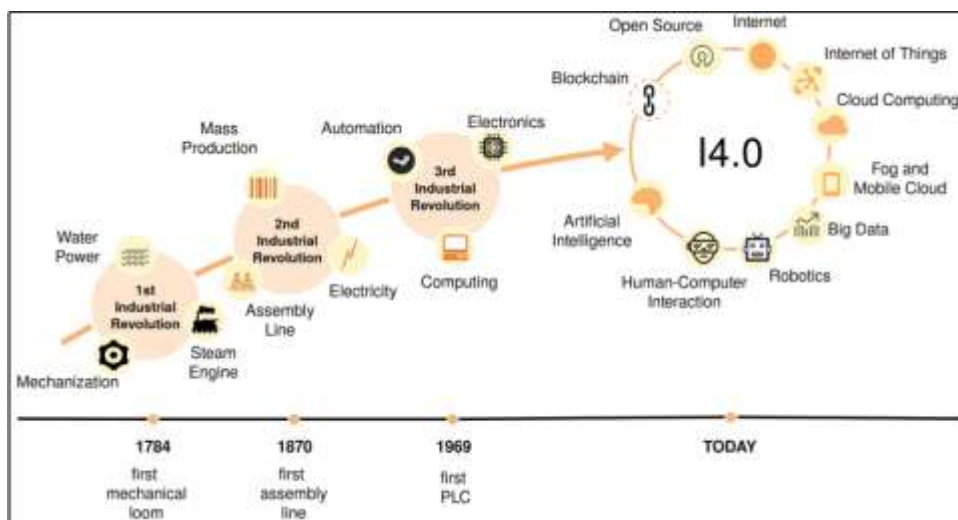
A proposta desse projeto de PI é desenvolver um framework web incorporando recursos de banco de dados, Javascript e API, integrado com elementos (dispositivos) baseados em Internet das Coisas (IoT - Internet of Things) para informar publicamente a população em situação de rua sobre a localização e a disponibilidade de vagas nos centros de acolhimento e atendimento, com a implantação de um botão de acionamento para emergências. Consoante ao tema norteador do PI para esse semestre, outras características que serão abordadas no desenvolvimento do projeto de PI incluem a disponibilização em nuvem, recursos de acessibilidade e integração contínua que poderá abordar mecanismos de controle de versão e política de testes.

2.3 Fundamentação teórica

A tecnologia da informação e comunicação (TIC) tem, nos últimos tempos, causado uma verdadeira revolução na sociedade. Como indutora de uma revolução tecnológica estabelece avanços extraordinários que moldam e criam paradigmas que impactam veementemente a maneira como as pessoas e as empresas irão se adaptar e usufruir dos cenários de transformações culturais, sociais, políticas e econômicas.

Atualmente, este avanço na TIC é propagado como a “quarta revolução industrial ou tecnológica” ou Indústria 4.0 que tem como seus pilares a conectividade e o processamento de informações. Aplicação de conceitos de IoT, Internet, computação em nuvem, inteligência artificial, entre outros, são fundamentais para que essa revolução seja explorada e apresente os resultados esperados. A Figura 1 apresenta uma visão geral dos períodos e das tecnologias desenvolvidas e empregadas desde a primeira revolução industrial. Observa-se no cenário atual as novas tecnologias que serão aperfeiçoadas e disponibilizadas para a sociedade e as empresas. O avanço tecnológico é extraordinário, assim como, os desafios e as responsabilidades que serão imputados à sociedade.

Figura 1. Ciclos da revolução industrial e tecnológica



Fonte: Aceto, Persico e Pescapé (2019).

Esse PI tem como objetivo desenvolver um framework web integrado a um dispositivo IoT. Resumidamente, podemos concluir que essa integração entre framework web e IoT corresponde a um sistema. Diante das várias definições apresentadas em Lemos (2011), pode-se extrair que um sistema é uma combinação de partes unidas ou interdependentes que interagem para formar uma unidade complexa dentro de uma estrutura organizada, com a finalidade de executar tarefas para atingir um objetivo.

Desta forma, esse PI explora o conceito de sistema que abrange um conjunto de elementos inerentes à quarta revolução tecnológica, tendo como exemplos: IoT, Internet, *open source* e interação humano-computador. O desafio é interessante, motivador e, ao mesmo tempo, inovador, visto que, essas tecnologias serão aplicadas em um cenário de assistência social de amparo à população em situação de rua na cidade de Osasco, remetendo, a um outro conceito que é muito empregado, atualmente e futuramente, que é o de cidades inteligentes.

2.3.1 Internet das coisas

A Internet das Coisas, conhecida também como *IoT*, que é a sigla em inglês de “*Internet Of Things*”, tem conseguido, nos últimos anos, mudar o cotidiano das pessoas, inclusive das que não têm conhecimento tecnológico começaram a aceitar a comodidade, o conforto e os valiosos conhecimentos que essa oferece. Desde os dispositivos domésticos conectados às fechaduras das portas, a vigilância à distância e todos os aparelhos

controlados por aplicações, muitos já conhecem e reconhecem a importância dessa tecnologia tanto para o uso industrial e comercial, como para o uso doméstico.

Essa é uma tecnologia em evolução que tem o objetivo de conectar objetos do mundo real a pessoas, como ressaltam os autores Mota e Batista (2013):

Em um futuro próximo, qualquer “coisa” (thing) poderá ser endereçada na grande rede. A Internet, então, abará um conjunto de aparelhos e dispositivos, ampliando o seu escopo para o universo da Internet das coisas. As comunicações serão concebidas não apenas entre humanos, mas também entre humanos e coisas e entre coisas sem a interação com seres humanos (MOTA; BATISTA, 2013, p. 297).

Assim sendo, *IoT* pode ser definida como a rede de objetos físicos - “coisas” - que levam incorporados sensores, software e outras tecnologias com a finalidade de conectar-se e transferir dados com outros dispositivos e sistemas através da Internet, sem que seja necessária a interação interpessoal ou de pessoa a computador.

O ecossistema *IoT* está formado por dispositivos inteligentes habilitados para a web que utilizam sistemas integrados, como processadores, sensores e hardware de comunicação para recolher, enviar e atuar os dados que adquirem de seus ambientes.

Tais dispositivos *IoT* compartilham dados dos sensores que reúnem se conectando a uma entrada de IoT ou a outro dispositivo periférico onde os dados são enviados à nuvem para ser analisados ou se analisam localmente. Também há dispositivos que se comunicam com outros dispositivos relacionados e atuam sobre a informação que obtém uns dos outros.

Os protocolos de conectividade, rede e comunicação utilizados com esses dispositivos habilitados para a web dependem em grande medida das aplicações específicas em que a *IoT* que são usadas.

As aplicações das tecnologias *IoT* são múltiplas, uma vez que ajustadas a quase qualquer tecnologia que seja capaz de proporcionar informação relevante sobre seu próprio funcionamento, sobre o desempenho de uma atividade e inclusive sobre as condições do ambiente que se deseja monitorar e controlar a distância. Atualmente, muitas empresas de diferentes setores ou ramos estão adotando esta tecnologia para simplificar, melhorar, automatizar e controlar diferentes processos.

2.3.2 Cidades inteligentes

Um dos usos surpreendentes da IoT são as cidades inteligentes, já que a vigilância inteligente, o transporte automatizado, os sistemas mais inteligentes de gestão de energia, a distribuição da água, a segurança urbana e a supervisão do meio ambiente são exemplos de aplicações de Internet das Coisas para essas cidades.

O desenvolvimento das cidades inteligentes surge a partir da tecnologia como fator chave para melhorar a vida das pessoas, criando um ambiente mais humano e sustentável. O avance da

transformação digital permite reivindicar os serviços dos governantes conseguindo uma maior eficácia e eficiência, redesenhando a matriz produtiva, avançando no cuidado com o meio ambiente e promovendo uma cultura de inovação aberta e de colaboração entre diferentes atores da sociedade.

Um exemplo da aplicabilidade dessa tecnologia é, basicamente, diversos sensores espalhados pela cidade com o propósito de monitorar e analisar um meio, como por exemplo, no Rio de Janeiro onde foi instalado um centro de operações (COR) que reúne informações da cidade e, antes de começar a chover em locais com possibilidade de alagamento, são mobilizados agentes locais e de suporte para estas áreas.

As cidades do futuro serão inteligentes, digitais, sustentáveis e inovadoras. Assim como a tecnologia está cada vez mais presente na vida cotidiana, também estará em todos os aspectos da gestão de territórios e de cidades. Contudo, na busca de soluções para problemas da cidade, o mais relevante é a capacidade de identificá-los, analisá-los e criar ou selecionar a ferramenta ou tecnologia mais adequada, fazendo o melhor uso dos recursos disponíveis.

2.3.3 Ferramentas Aplicadas ao Framework Web

A ideia para o desenvolvimento do PI caracteriza-se por empregar um conjunto de ferramentas tecnológicas baseadas em código aberto e software livre. No mercado há uma gama de linguagens de programação que utilizam a filosofia de código aberto e software livre.

Para o desenvolvimento dos componentes atrelados ao *front-end* será utilizada um framework denominado Bootstrap. De acordo com a W3Schools³, o Bootstrap reúne três aspectos:

- Um framework gratuito de desenvolvimento do *front-end* para aplicações web que é rápido e fácil de usar;
- Incluir modelos de design baseados em *HTML* e *CSS* para tipografia, formulários, botões, tabelas, navegação, modais, carrosséis de imagens e muitos outros, bem como plugins *JavaScript*;
- Oferecer a capacidade de criar facilmente designs de websites responsivos.

Seguindo essa analogia, pode-se interpretar que o Bootstrap é um *framework open source* que opera com o *front-end* das aplicações utilizando *HTML*, *CSS* e *JavaScript*. Possibilita que sejam criadas aplicações web responsivas de forma rápida e simples, ou seja, aplicações web que se ajustam automaticamente aos diversos tamanhos, padrões e resoluções de tela.

³Bootstrap Get Started. *W3Schools*. Disponível em https://www.w3schools.com/bootstrap/bootstrap_get_started.asp. Acesso em: 20 mar. 2023.

As bibliotecas podem ser entendidas como um conjunto de diversos trechos específicos de códigos que tratam de um assunto específico e que podem ser reutilizáveis de diversas maneiras em diversos códigos. Os *frameworks* são bibliotecas de código reutilizável que surgem no cenário da computação através do desenvolvimento do paradigma de linguagens de programação orientadas a objetos. Seu intuito é evitar que os programadores tenham que trabalhar de forma extensiva em todo o uso de trecho de um código que já existe, resultando, com isso, em uma maior produtividade.

Python é a linguagem de programação adotada para esse PI. É uma linguagem de programação poderosa e fácil de aprender. Possui estruturas de dados de alto nível eficientes e uma abordagem simples, mas eficaz para programação orientada a objetos. A sintaxe elegante e a tipagem dinâmica do *Python*, juntamente com sua natureza interpretada, a torna ideal para *scripts* e desenvolvimento rápido de aplicativos em muitas áreas na maioria das plataformas (SWAROOP, s.d.).

As principais características da linguagem *Python* incluem:

- Simples e fácil de aprender;
- Adota a filosofia de código aberto e livre;
- Linguagem de alto nível que portável e interpretada;
- Orientada a objetos;
- Extensível e incorporável, ou seja, permite conjugar a linguagem Python com aplicações desenvolvidas em outras linguagens.

A estrutura desse PI requer, também, a utilização de uma base de dados. Neste caso, optou-se pelo *MySQL* que é um dos mais populares sistemas gerenciadores de banco de dados relacional (SGBDR ou *RDBMS - Relational Database Management Systems*).

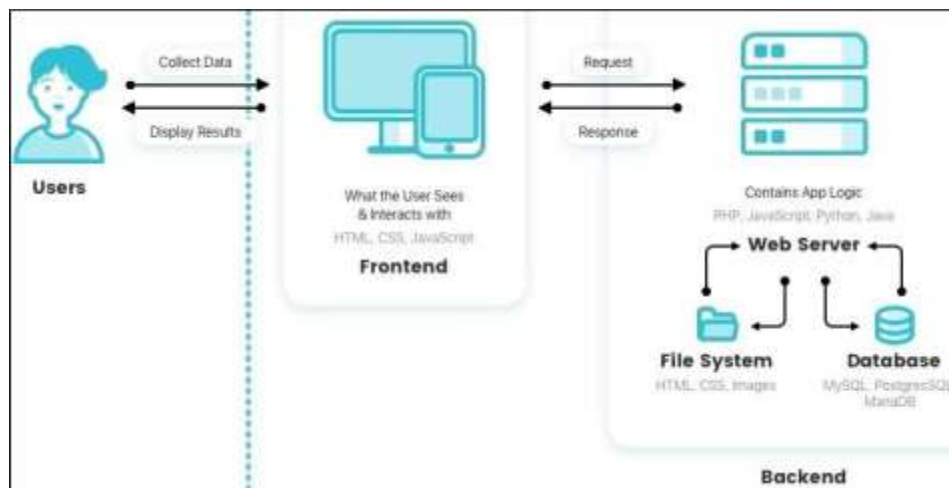
MySQL é um SGBDR (registrado pela *Oracle Corporation*) muito rápido, *multithread*, multiusuário e robusto, que destina-se a sistemas de produção de carga pesada de missão crítica, bem como para incorporação em *software* implantado em massa (MYSQL, 2022).

Django é um *framework* que auxilia no desenvolvimento rápido de aplicações para *web*. Inclui um conjunto de objetos que auxiliam com as tarefas comuns de desenvolvimento web, como: autenticação do usuário, administração de conteúdo, mapas do site, *feeds RSS* e muitas outras tarefas (DJANGO, 2022). Facilita e evita que o programador *web* recrie diversas vezes a mesma coisa. Com Django, é possível criar estruturas esquematizadas de bancos de dados, obter dados de dentro ou fora do banco de

dados, analisar *URLs* (*Uniform Resource Locator*), limpar a entrada de dados, fornecer ferramentas de edição de conteúdos e atender aos requisitos de segurança e usabilidade.

A Figura 2 apresenta uma visão geral dos recursos tecnológicos e linguagens que são utilizadas no desenvolvimento de *framework web*.

Figura 2 Arquitetura de um framework web



Fonte: DABBS (2019)

2.3.4 Acessibilidade nas aplicações

Acessibilidade é tornar qualquer coisa disponível para uso de qualquer pessoa, indiferente de suas habilidades e/ou deficiências.

A acessibilidade tem que ser pensada desde a concepção do projeto. Se desde o início a acessibilidade for considerada, menor será o retrabalho no futuro. É importante que acessibilidade seja contemplada nas regras de negócio.

2.3.5 Diretrizes de acessibilidade na Web

São documentos que guiam o desenvolvimento da web para que pessoas com deficiência tenham acesso. Geralmente, são documentos técnicos, mas que deixam claro as barreiras de acesso, definindo assim as medidas a serem tomadas. Ou seja, as principais diretrizes orientam o desenvolvedor a seguir corretamente a documentação técnica.

Para esse projeto foi priorizada a questão da acessibilidade. Rachadel (2017, p. 26), define a acessibilidade como:

Acessibilidade é a qualidade atribuída àquilo que é acessível, ou seja, àquilo que é alcançável, àquilo que possui acesso fácil no trato e na aquisição. Nas últimas décadas ela está diretamente ligada às preocupações de arquitetos e urbanistas em prol do fornecimento de ensejos às pessoas portadoras de deficiência ou com mobilidade reduzida para o direito de aproveitamento dos espaços públicos com total segurança e autonomia. (RACHADEL, 2017, p. 26).

Essa é uma definição de acessibilidade de uma forma ampla. Já, em se tratando de acessibilidade na rede, Henry (2018) a define como “acessibilidade na *Web* significa que sites, ferramentas e tecnologias são projetados e desenvolvidos para que pessoas com deficiência possam usá-los”. Dessa forma, um sistema *Web* necessita ser desenvolvido valendo-se da acessibilidade como uma premissa e não simplesmente como um recurso adicional para que as pessoas com deficiência possam acessá-las.

O sistema criado nesse projeto encontra-se de acordo com as diretrizes do *WCAG* (*Web Content Accessibility Guidelines*) na versão 2.1, as quais buscam verificar e nortear a acessibilidade ao conteúdo das páginas Web, seja um elemento textual ou audiovisual. Como também no que tange as normas *UAAG* (*User Agent Accessibility Guidelines*), que visam a acessibilidade nos navegadores, dos reprodutores multimídia ou qualquer outro componente que faça parte da renderização e exibição da página ou sistema *Web* (HENRY, 2018).

A preocupação de que o sistema atenda as normas de acessibilidade é porque se espera que os usuários que possuam qualquer tipo de deficiência, de natureza física, intelectual, mental ou sensorial, não encontrem dificuldades em utilizá-lo.

2.3.6 WCAG

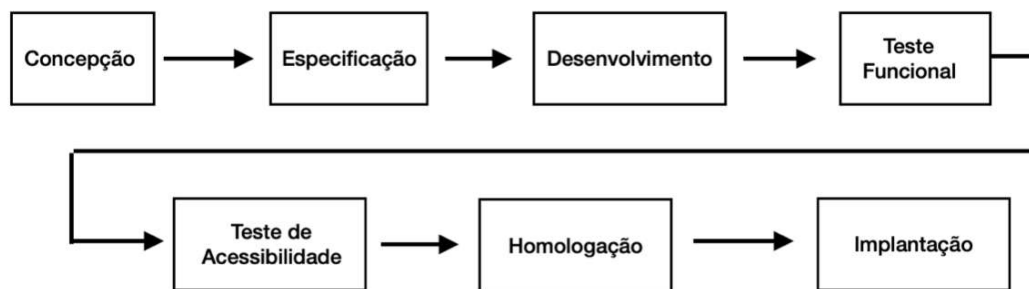
WCAG (*Web Content Accessibility Guidelines*), atualmente na versão 2.1, pode ser considerado o principal documento de acessibilidade na Web, no qual o desenvolvedor se baseia para tornar o conteúdo acessível.

Ela possui 3 níveis de conformidade - A, AA, AAA -, onde quanto mais se eleva o nível, mais se remove barreiras de acesso para as diversas deficiências. É um documento que não foca em tecnologia, ou seja, mesmo sendo considerado técnico, ele é filosófico para fazer refletir sobre determinada necessidade. Foca em conteúdo: informações naturais e semântica.

É uma norma flexível, com conceitos que podem ser aplicados em diversas plataformas. Por exemplo: em uma página *HTML*, não vai falar que é um botão se não houver uma *tag button*, mas vai dizer que: se a tecnologia suporta semântica específica, esta semântica deve ser aplicada.

O WCAG não é o único documento, há normas irmãs como o UAAG, que regem os agentes de usuários, ATAG como ferramentas de edição e os padrões HTML (*WAI-ARIA* e as especificações HTML 5). É importante saber que elas existem e que os desenvolvedores devem seguir.

Figura 3 Fluxograma ao implantar uma aplicação acessível



Fonte: Elaborado pelo autor.

2.4 Aplicação das disciplinas estudadas no Projeto Integrador

É natural que algumas disciplinas apresentaram uma contribuição maior devido a sua aderência às especificidades deste trabalho de PI e dos anteriores, porém, outras disciplinas, mesmo que indiretamente, também tiveram a sua importância. Diante deste contexto, pode-se inferir que todas as disciplinas estudadas, até o momento, tiveram uma participação bastante expressiva para a conclusão dos trabalhos de PI. Abaixo são relacionadas algumas disciplinas, cujo conteúdo e conceitos foram essenciais.

2.4.1 Fundamentos de Internet e Web (COM130)

Temas fundamentais para o desenvolvimento do PI inerentes ao *front-end*, como *HTML5 (Hypertext Markup Language)*, *CSS (Cascading Style Sheets)* e *Javascript* foram introduzidos na disciplina. Também, apresentou uma introdução aos protocolos de comunicação, *TCP (Transmission Control Protocol)*, *UDP (User Datagram Protocol)* e *IP (Internet Protocol)*, que são usados em conexões baseadas na *WWW (World Wide Web)* e fundamentais para o desenvolvimento de aplicações *web*.

A disciplina, também, apresentou e tratou de outros aspectos fundamentais da *web*, como o protocolo *HTTP (Hypertext Transfer Protocol)* que é um protocolo cliente/servidor que permite reconhecer páginas ou documentos HTML e CSS, que são partes essenciais para uma formatação mais adequada e atraente das páginas que serão apresentadas para os usuários (clientes) em um navegador ou *browser* (Chrome, Firefox, Opera, etc).

2.4.2 Desenvolvimento Web (COM320)

Os métodos e as ferramentas inerentes ao processo de desenvolvimento de um framework web reúne um conjunto de características e particularidades que devem ser observadas. É neste contexto, que a disciplina de Desenvolvimento Web teve um papel importante em ajudar a compreender os vários aspectos envolvidos no processo de desenvolvimento do framework web.

A disciplina Desenvolvimento Web proporcionou conhecimentos e habilidades que possibilitaram compreender, de forma introdutória, as metodologias, componentes e estruturas de uma aplicação web, as tecnologias empregadas no front-end e *back-end*, tipos de banco de dados voltados para o desenvolvimento web e, por fim, o conceito, a aplicabilidade e as tecnologias disponíveis e utilizadas na computação em nuvem.

2.4.3 Algoritmos e Programação de Computadores I e II (COM110/120)

O framework web é um programa baseado na Web que para ser desenvolvido deve utilizar uma linguagem de programação. Portanto, não se exige um exercício aprofundado para perceber a importância dessa disciplina no desenvolvimento de um framework. Como mencionado, as características, sintaxes, estruturas e componentes que compõem a metodologia de orientação a objetos são fundamentais para desenvolver os componentes do que se denomina *back-end*, ou seja, aspectos relacionados ao acesso e armazenamento em banco de dados, bem como a aplicação de regras de negócios definidas durante a fase de projeto do framework.

As duas disciplinas trataram de conceitos e exemplos simples de aplicações utilizando a linguagem Python. Este trabalho de PI empregou outra linguagem de programação, contudo, os fundamentos e as estruturas básicas inerentes à maioria das linguagens atuais são semelhantes, e, portanto, o aprendizado de uma determinada linguagem pode ser expandido para a compreensão dos conceitos abordados em outras linguagens.

2.4.4 Protocolos de Comunicação IoT (COM380)

Essa disciplina apresentou um panorama sobre os principais conceitos, definições e contextualizações sobre o tema *IoT*. Dentro desse panorama foi introduzido aspectos relacionados às aplicações e a importância da tecnologia para o presente e futuramente, assim como, a arquitetura e as tecnologias de comunicação que são utilizadas em um ambiente *IoT*.

Outro aspecto importante que foi introduzido é que o termo “coisas” refere-se

a qualquer dispositivo que possa ser capaz de processar informações e interagir com um conjunto de dispositivos, como celulares, TV, cafeteiras, geladeiras, dispositivos de iluminação (lâmpadas, por exemplo), entre outros.

Por fim, ressaltou que o desenvolvimento de soluções *IoT* requer processos bem definidos acerca dos *hardwares*, *softwares* e linguagens que serão utilizadas.

2.4.5 Engenharia de Software (COM390)

Essa disciplina orientou algumas formas para a definição de padrões de projeto. Dentre esses padrões destacam-se quem são os participantes, propósito e os objetivos, motivação e a aplicabilidade e aspectos de colaboração e implementação.

Também, proporcionou subsídios relacionados às estratégias e técnicas empregadas para a realização de testes de software. Testes estruturais, funcionais e de segurança são alguns testes que foram abordados nessa disciplina e serviram de alicerce para a estratégia de testes a ser adotada no PI.

2.5 Metodologia

A definição do contexto e o desenvolvimento do PI devem basear-se em métodos e ferramentas propostas no Design Thinking.

O Design Thinking é um modelo de pensamento centrado no ser humano, pluralista e sistêmico, cujo objetivo é construir soluções inovadoras para problemas complexos (ESCOLA DE DESIGN THINKING, s.d.; THORING; MÜLLER, 2011). Essa abordagem inovadora para a solução de problemas tem atraído o interesse de profissionais e acadêmicos, entretanto, não há um consenso sobre sua aplicabilidade e resultados ((MICHELI et al., 2019)).

Augusto de Lahoz (vídeo⁴ disponibilizado na Univesp) menciona cinco estágios, definidos conforme um documento da DSchool da Universidade de Stanford, para construir um projeto baseado em Design Thinking:

- Empatia: criar empatia e interagir com as pessoas para compreender as suas necessidades;
- Definição: definir o problema a ser resolvido com base nas necessidades levantadas durante o processo de empatia;
- Ideação: concepção de ideias, ou seja, deixar que os envolvidos exponham suas ideias livremente para a resolução do problema. É um processo de

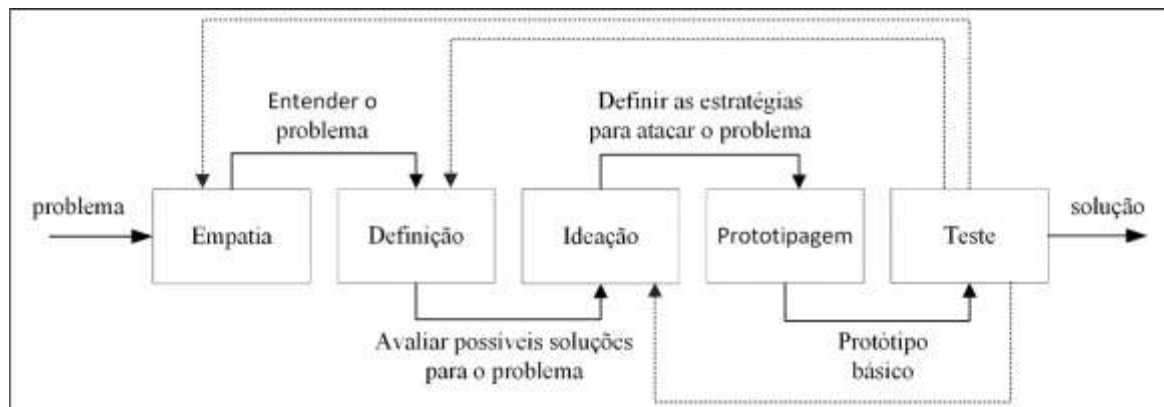
⁴Afinal, o que é Design Thinking? E quais são suas etapas fundamentais? *Canal Na Prática*. Disponível em: <https://www.youtube.com/watch?v=acHRiZZptSs&t=31s>. Acesso em: 21 mar. 2023

brainstorming. Prototipagem: gerar um protótipo com base em uma ou mais ideias apresentadas no processo de ideação;

- Teste: testar o protótipo que foi criado para avaliar o quanto resolve o problema e, em caso contrário, identificar lacunas que precisam ser resolvidas ou melhoradas.

Observa-se uma abordagem mais sintetizada e funcional sobre as etapas do Design Thinking. Entretanto, esse não é um processo linear, visto que, os resultados alcançados na etapa de teste podem apontar a necessidade de ajustes e revisões nas etapas anteriores do ciclo do projeto (DAM; SIANG, 2018).

Figura 4 Etapas do Design Thinking



Fonte: adaptada de Taratukhin; Yadgarova e Becker (2018); Dam e Siang (2018).

Com uma abordagem amparada na metodologia do *Design Thinking* as estratégias metodológicas adotadas para o desenvolvimento do PI fundamentaram-se nas etapas relacionadas abaixo.

2.5.1 Ouvir e Interpretar o Contexto

A escolha da comunidade externa para a aplicação do projeto de PI baseou-se em discussões virtuais realizadas pelos integrantes do grupo. A técnica de *brainstorming* foi utilizada para o compartilhamento de ideias com a finalidade de auxiliar no mapeamento de cenários para escolha e consolidação da comunidade externa. O grupo, por consenso, decidiu desenvolver o projeto de PI com o escopo de atenção à população em situação de rua, que além do aspecto social inerente a essa demanda, também pesou a perspectiva de desenvolver um dispositivo físico aplicando técnicas, conceitos e recursos relativos ao contexto da Internet das Coisas (IoT – Internet of Things).⁵

⁵ A ideia de desenvolver um dispositivo abordando as técnicas da IoT foi sugerida por uma integrante do grupo, a qual teve uma experiência com esse cenário em uma das disciplinas cursadas na Univesp.

O primeiro contato com a Secretaria de Assistência Social (SAS) da Prefeitura Municipal de Osasco foi realizado por meio de e-mail, onde a Sra. Danielle Bueno, Diretora do Departamento de Proteção Social Especial, se prontificou em nos atender. O Departamento de Proteção Social Especial (DPSE) é responsável pelo acolhimento de pessoas adultas de ambos os sexos em situação de rua e desabrigo por abandono, migração e ausência de residência ou pessoas em trânsito e sem condições de autossustento. O serviço de acolhimento também é estendido para crianças e adolescentes, cujas famílias ou responsáveis encontram-se temporariamente impossibilitados de cumprir sua função de cuidado e proteção.

O processo de coleta dos dados foi realizado por meio de entrevista presencial. As Sras. Danielle e Jessaminy, Gerente Administrativa do DPSE fizeram inicialmente uma exposição geral sobre o escopo do DPSE e apresentaram alguns dados relacionados à quantidade de centros de acolhimento e atendimento, de vagas disponibilizadas nesses centros e uma previsão do total de pessoas em situação de rua na região de Osasco; ressaltando que o número de vagas de moradia nos centros é insuficiente. Entretanto, em períodos de inverno e se a previsão do tempo registrar temperaturas abaixo de 10°C há uma ampliação no número de vagas nos próprios centros de acolhimento e, também, em instituições privadas (hotéis, por exemplo) por meio de parcerias. Foi relatado, também, que algumas pessoas em situação de rua amparam algum animal de estimação - cães em sua maioria. Neste caso, o centro de acolhimento também deve dispor de condições para oferecer um alojamento para o animal.

Por fim, as entrevistadas sugeriram se o dispositivo IoT poderia dispor de algum mecanismo onde a pessoa em situação de rua ou qualquer munícipe pudesse estabelecer um canal direto de comunicação com algum centro de atendimento para informar sobre as condições físicas e de saúde da pessoa em situação de rua e, consequentemente, agilizar o processo de atendimento. Atualmente, esse canal de comunicação está centralizado na Central 156, da prefeitura, o que pode acarretar uma demora para o atendimento, em função da alta demanda de serviços oferecidos e atendidos por essa central.

Assim, a ideia de desenvolver um framework web com acesso a um dispositivo eletrônico (IoT) para informar publicamente sobre a localização e disponibilidade de vagas nos centros de acolhimento foi muito bem aceita pelas representantes da comunidade externa, as quais se propuseram até mesmo em buscar viabilizar recursos materiais e financeiros para a implantação de um projeto em escala experimental. Adicionalmente, estuda-se a possibilidade do dispositivo IoT dispor de um canal direto de comunicação com o centro de atendimento ou bombeiros do município pelo acionamento de um botão de emergência, o qual permitiria o rápido socorro ao cidadão em situação de rua.

2.5.2 Criar e Prototipar

A fase “criar e prototipar” representa o momento em que as ideias geradas na fase anterior são concretizadas para que outras pessoas possam contribuir para o seu refinamento. É o momento de avaliar se a solução proposta é viável e atende aos requisitos propostos, ou seja, representa a etapa onde as ideias são traduzidas do papel para a vida real (ESCOLA DE DESIGN THINKING, s.d.). De acordo com Ruffo (2016), um protótipo ajuda a materializar a ideia fazendo com que as pessoas tenham uma familiaridade com a solução desenvolvida, ou seja, uma interação direta com a ideia, para que pontos fortes e fracos possam ser identificados.

Nesta etapa, será elaborado um diagrama de contexto e definida uma lista de eventos ou funcionalidades que o framework web, ou seja, a solução a ser desenvolvida, deve contemplar com base nos requisitos (ideias/problemas) que foram levantados junto à comunidade externa.

Para que a solução possa ser desenvolvida, esta etapa, também, deverá contemplar uma série de atividades e/ou estratégias, relacionadas ao campo da TIC, e que são essenciais para que os objetivos traçados para o desenvolvimento do PI possam ser alcançados. As principais atividades e/ou estratégias são:

Esboçar e definir a arquitetura do sistema com base nas linguagens de programação, no banco de dados, nas API's e na estrutura do dispositivo IoT;

- Realizar a modelagem do banco de dados;
- Elaborar o protótipo das telas do sistema e do dispositivo IoT;
- Definir, iniciar a construção e programação dos elementos e objetos do dispositivo IoT;
- Definir e desenvolver o *front-end* do framework web, considerando as linguagens HTML, CSS, Javascript e Python;
- Definir e desenvolver o *back-end* do framework web, que contempla a estrutura do banco de dados adotado e do framework Django;
- Estabelecer a interoperabilidade entre o dispositivo IoT e o framework web.

Por fim, os códigos-fontes, arquivos de configuração, gráficos, planilhas e todos os documentos relacionados às etapas de desenvolvimento do PI deverão ser registrados e atualizados na plataforma de controle de versão Git/Github para que outros integrantes possam acompanhar e contribuir com o desenvolvimento.

2.5.3 Implementar e Testar

O desenvolvimento de um sistema baseado na TIC envolve um grau de complexidade elevado que exige dos profissionais da área conhecimentos e habilidades para enfrentar os desafios e problemas que surgem durante todas as etapas do desenvolvimento. Não é incomum surgirem problemas em qualquer uma das etapas do ciclo de desenvolvimento. Portanto, as atividades de validação, verificação e teste não se restringem ao produto final e devem ser conduzidas desde a concepção inicial do projeto (DELAMARO; MALDONADO; JINO, 2016, p. 1).

A etapa de testes representa uma extensão da etapa de prototipação onde os desenvolvedores do projeto buscam coletar feedbacks dos potenciais usuários do sistema para avaliar a qualidade da solução apresentada (O'CALLAGHAN; CONNOLLY, 2020; FILA; MCKILLIGAN; GUERIN, 2018). Os integrantes do grupo, nesta etapa, deverão interagir com a comunidade externa para validar se o sistema está funcionando corretamente e se os requisitos especificados foram atendidos. É importante ressaltar que vários testes são conduzidos na etapa de prototipagem para que erros ou falhas técnicas sejam identificados e corrigidos.

A estratégia de testes, adaptada para este PI, abrangerá três fases (DELAMARO; MALDONADO; JINO, 2016, p. 2):

- Teste de unidade: identificar erros de programação, de estruturas de dados e relacionados a algoritmos incorretos ou mal desenvolvidos;
- Teste de integração: verificar se a interação entre as diversas partes do sistema está funcionando de maneira adequada e não apresenta erros ou falhas quando combinadas;
- Teste de sistema: verificar se os eventos e/ou funcionalidades especificadas no levantamento de requisitos foram contemplados. Requisitos não funcionais como segurança, desempenho e robustez também são avaliados. Em resumo, é um teste que verifica se o sistema como um todo está em conformidade com o que foi idealizado.

Como a proposta do PI é desenvolver um framework web, outros testes direcionados especificamente para aplicações web devem ser considerados. Visto que, uma aplicação web é um sistema composto, basicamente, por uma base de dados (*back-end*) e páginas web (*front-end*) com os quais os usuários interagem utilizando um navegador web em uma rede (LI; DAS; DOWE, 2014). A heterogeneidade das aplicações web faz com que diferentes componentes de software, geralmente, utilizem diferentes linguagens de programação e

apresentação como Javascript, CSS e HTML, e isso acaba introduzindo dificuldades para a atividade de teste (DELAMARO; MALDONADO; JINO, 2016, p. 228).

Além das estratégias de testes mencionadas anteriormente, Ricca & Tonella (2001) propuseram duas abordagens de testes para aplicações web que também serão adotadas para este PI:

- 1) Estática: que analisa caminhos de navegação ou fluxo de informações fornecidas pelo usuário e busca identificar:
 - Páginas inacessíveis: páginas disponíveis no servidor, porém, não podem ser acessadas por nenhum caminho a partir da página inicial;
 - Páginas fantasmas: páginas associadas a links com páginas inexistentes;
 - Frames alcançados: inconsistências na formatação e organização dos frames (quadros) das páginas;
 - Dependência de dado: apontar incongruência no uso de variáveis;
 - Páginas cruzadas: páginas que são acessadas por meio de outras páginas (hiperlinks são formas de navegar por várias páginas cruzadas);
 - Caminho mais curto: avaliar o número mínimo de páginas a serem acessadas para chegar à página desejada.

- 2) Dinâmica: avalia a aplicação por meio da execução de casos de teste. Busca revelar inconsistências relacionadas ao fluxo de controle e de dados entre as páginas da aplicação web (DELAMARO; MALDONADO; JINO, 2016, p. 228). Os seguintes critérios serão abordados para os testes:
 - Teste de páginas: cada página do website deve ser visitada pelo menos uma vez em determinado caso de teste;
 - Teste de hiperlink: cada hiperlink de cada página no website deve ser percorrido pelo menos uma vez;
 - Teste de definição-uso: todos os caminhos de navegação associados a um conjunto definição-uso devem ser verificados;
 - Teste de todos-usos: pelo menos um caminho de navegação associado a um conjunto definição-uso deve ser verificado;
 - Teste de todos-caminhos: cada caminho no website deve ser percorrido em algum caso de teste pelo menos uma vez.

Testes relacionados ao dispositivo IoT também, serão encaminhados. Nestes testes, busca-se verificar a consistência do dispositivo IoT em ambientes heterogêneos, a interoperabilidade com o framework web e aspectos relacionados meios de comunicação físico e lógico, acampando o sistema como um todo.

Por fim, após a validação de todos os testes por parte da comunidade externa, o sistema estará apto para ser implementado e disponibilizado, a princípio, como um “projeto piloto” e sugere-se um acompanhamento pela Secretaria de Assistência Social da prefeitura, coletando *feedbacks* junto à população em situação de rua, para que, de fato, tenha-se uma percepção da viabilidade e usabilidade desse recurso tecnológico.

3 SOLUÇÃO INICIAL

Na disciplina Projeto Integrador III o sistema foi desenvolvido e agora servirá de base para as soluções que são alvo da disciplina Projeto Integrador IV. Nesta seção do relatório será apresentada a estrutura existente e na sequência sua complementação.

3.1 Componentes técnicos para a construção do dispositivo IoT

A IoT é uma tecnologia inovadora que vem revolucionando a forma de interconexão e interoperabilidade de diversos dispositivos do cotidiano, desde um simples aparelho doméstico a um automóvel, permitindo que esses objetos possam ser monitorados e gerenciados à distância, por exemplo.

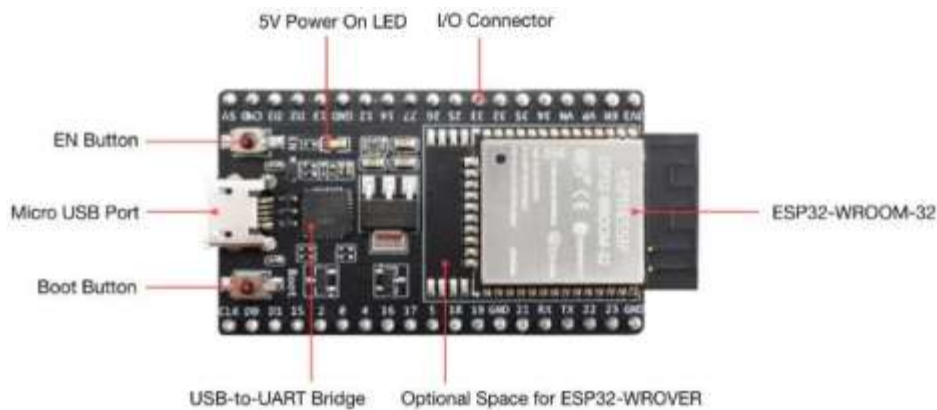
Um dos objetivos centrais desse projeto fundamenta-se no desenvolvimento de um dispositivo IoT. Para tanto, para a construção da solução inicial alguns componentes foram necessários para que o dispositivo tornasse operacional e interoperável.

3.1.1 ESP32 (WiFi / Bluetooth)

O ESP32 foi projetado para dispositivos móveis e aplicativos de Internet das Coisas (IoT). Apresenta todas as características de última geração de chips de baixa potência, incluindo clock gating de granulação fina, vários modos de energia, e escala de potência dinâmica (ESP32 SERIES, 2022).

Ainda segundo as especificações técnicas ESP32 SERIES (2022) é uma solução altamente integrada para aplicações WiFi (802.11 b/g/n) e Bluetooth (v4.2 BR/EDR e LE) e integra um interruptor de antena, balun de RF, amplificador de potência, amplificador de recepção de baixo ruído, filtros, e módulos de gerenciamento de energia. A Figura 5 apresenta uma imagem de um padrão de uma placa ESP32 com algumas especificações internas.

Figura 5. Modelo de placa ESP32



Fonte: Elaborado pelo autor.

3.1.2 DHT11 – Sensor de Temperatura e Umidade

Com base nas descrições da Elprocus (s.d.), o *DHT11* (Figura 6) é um sensor digital de baixo custo para detecção de temperatura e umidade. Este sensor consiste de um elemento de detecção de umidade capacitiva e um termistor para detecção de temperatura e pode ser conectado a microcontroladores, como *Arduino*, *Raspberry*, etc., para medir umidade e temperatura em tempo real

Figura 6. Modelo de sensor DHT11



Fonte: Elaborado pelo autor.

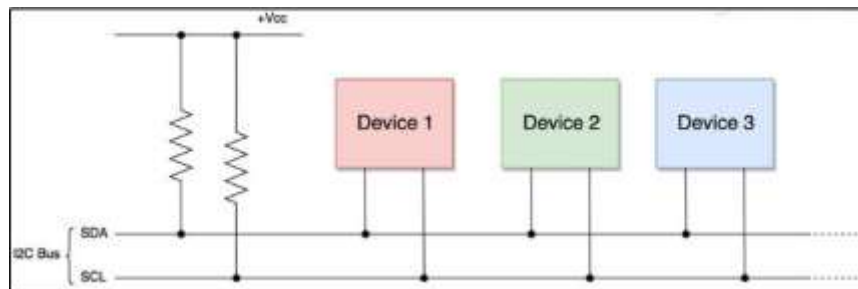
Para medição de temperatura este sensor utiliza um termistor de coeficiente de temperatura negativo, que causa uma diminuição em seu valor de resistência com o aumento da temperatura. Para obter maior valor de resistência mesmo para a menor mudança de temperatura, esse sensor geralmente é feito de cerâmicas ou polímeros semicondutores. A faixa de temperatura do *DHT11* é de 0 a 50 graus Celsius com precisão de 2 graus. A faixa de umidade deste sensor é de 20 a 80% com 5% de precisão. A taxa de amostragem deste sensor é de 1 Hz, ou seja, dá uma leitura para cada segundo. *DHT11* é pequeno em tamanho com tensão operacional de 3 a 5 volts. A corrente máxima usada durante a medição é de 2,5mA.

3.1.3 Inter-Integrated Circuit (I2C) – Módulo e Protocolo

O Inter-Integrated Circuit (I2C) é um protocolo de comunicação comum, muito adequado para comunicações entre periféricos on-board para transferir dados de baixa e/ou média velocidade. O módulo I2C é amplamente utilizado em vários controladores, sensores e alguns outros circuitos integrados (LIU et al, 2019).

O I2C foi introduzido pela empresa Philips e seu barramento é composto por dois dispositivos: master e slave. O processo de comunicação entre o master e o slave é estabelecido por dois esquemas: SDA (serial data) que é responsável pela transmissão de dados e o SCL (serial clock) que se encarrega em compartilhar o sinal de clock gerado no barramento (THANGAVEL, 2018). A Figura 6 mostra um exemplo de um barramento I2C típico, onde o “Device 1” pode ser considerado um master e os demais “Devices”, slaves.

Figura 7. Barramento I2C



Fonte: Elaborado pelo autor.

3.1.4 Integrated Development Environment (IDE) - Arduino

Basicamente, Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software usada para a construção de projetos eletrônicos que consiste em uma placa de circuito impressa programável e um software *IDE (Integrated Development Environment)*, executado no computador e usado para escrever e fazer upload de código de computador para a placa física. Por meio de um conjunto de instruções baseadas em versão simplificada da linguagem C++, as placas Arduino são capazes de ler sinais de entrada analógicos ou digitais de diferentes sensores e transformá-los em instruções de saída, que possibilita ativar um motor, ligar/desligar um LED, conectar-se à nuvem entre muitas outras ações (ARDUINO, 2018; GALADIMA, 2014; NUSSEY, 2013; KUMAR; KUMAR, 2013).

A tela de apresentação da interface IDE Arduino é exibida na Figura 8.

Figura 8. IDE Arduino



Fonte: Elaborado pelo autor.

Segundo a comunidade do Arduino (ARDUINO, 2018) algumas vantagens da utilização da IDE Arduino incluem:

- Ambiente multiplataforma: compatível com os sistemas operacionais Windows, Macintosh e Linux;
- Ambiente de programação mais amigável: IDE Arduino é um ambiente amigável para iniciantes, porém, flexível o suficiente para usuários avançados também aproveitarem todos os seus recursos;
- Software de código aberto e extensível: se baseia em ferramentas de código aberto, disponíveis para extensão por programadores experientes, que podem utilizar um conjunto de bibliotecas C++ e a linguagem de programação AVR C, na qual se baseia, para ampliar as suas funcionalidades.

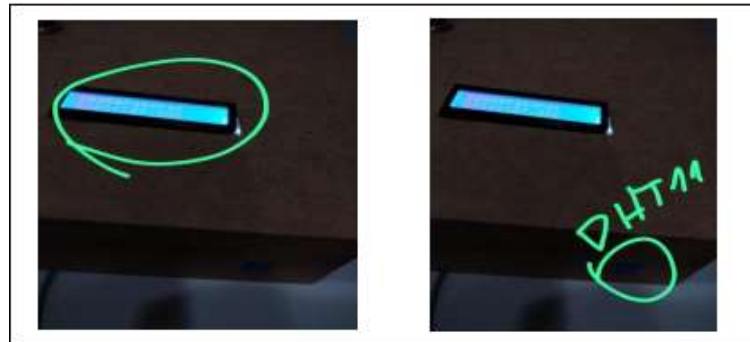
Neste PI, não foi utilizada a placa de circuito impresso programável da plataforma Arduino. Portanto, a interface IDE Arduino foi utilizada especificamente para a programação e operação da placa ESP32.

3.2 Montagem do dispositivo IoT

Todos os componentes mencionados no item anterior foram utilizados para a montagem do dispositivo IoT. Cada figura abaixo mostra e assinala a disposição e as conexões de cada componente no dispositivo IoT.

A Figura 9 demonstra a disposição do display LCD 16x2 juntamente com o sensor de temperatura e umidade DHT11, ao passo que a Figura 10 indica a disposição do sensor DHT11 na montagem do dispositivo IoT.

Figura 9. Display LCD 16x2 e sensor DHT11



Fonte: Elaborado pelo autor.

Figura 10. DHT11 - sensor de temperatura e umidade.



Fonte: Elaborado pelo autor.

A Figura 11 indica a disposição do módulo I2C integrado ao display LCD 16x2. A função do display é exibir as mensagens conforme foram programadas. Neste PI, essas mensagens se referem a informações sobre endereço, disponibilidade de vagas e outras, sobre um determinado centro de acolhimento.

Figura 11. Display LCD 16x2 com módulo I2C.

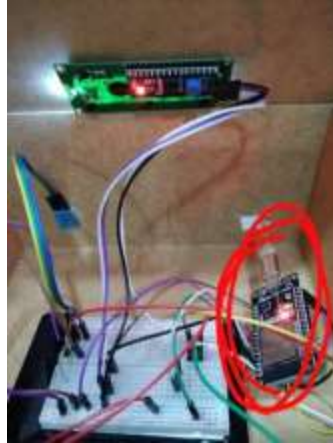


Fonte: Elaborado pelo autor.

A disposição da placa ESP32 no dispositivo é apresentada na Figura 12 como mencionado anteriormente, a ESP32 é uma placa que desempenha uma função plural, pois, é

programável e caracteriza-se por controlar as conexões e processar as instruções dos códigos escritos em uma linguagem de computação.

Figura 12. Disposição da placa ESP32 no dispositivo IoT.



Fonte: Elaborado pelo autor.

Por fim, a Figura 13 mostra uma placa de ensaio ou matriz de contatos (em inglês, breadboard ou protoboard), é uma placa com furos e conexões condutoras utilizadas para a montagem de protótipos e projetos em estado inicial, facilitando à inserção de componentes, uma vez que não necessita soldagem (WIKIPEDIA, 2022).

Figura 13. Placa de ensaio ou matriz de contatos



Fonte: Elaborado pelo autor.

3.3 Framework Web

O desenvolvimento do *front-end* e do *back-end* fundamentaram-se nas tecnologias HTML5, CSS, JavaScript, Bootstrap, PostgreSQL, Django e Heroku.

A estrutura que reúne os códigos e objetos utilizados para o desenvolvimento do front-end do framework web pode ser observada no GitHub, cujo endereço eletrônico é:

- <https://github.com/isapro77/ProjetoIntegrador3>

A Tabela 1 apresenta uma descrição resumida das características de cada uma dessas tecnologias.

Tabela 1. Tecnologias empregadas no desenvolvimento do front-end e back-end. Fonte: Elaborado pelo autor..

HTML5	<i>HiperText Markup Language 5</i> é uma linguagem de marcação utilizada para construir elementos de página como parágrafos, <i>links</i> , títulos, tabelas, imagens, etc.
CSS	<i>Cascading Style Sheets</i> é definida como uma linguagem de estilos utilizada para definir cores, fontes, tamanhos, posicionamento e qualquer outro valor estético para os elementos da página.
<i>JavaScript</i>	É uma linguagem de programação que permite a implementação de recursos mais elaborados e dinâmicos em páginas <i>web</i> , como multimídia, imagens animadas, etc.
BootStrap	É um <i>framework</i> de código aberto que auxilia o desenvolvimento de aplicações <i>web</i> com características responsivas de forma rápida e simples. Utiliza recursos que facilitam a utilização de HTML5, CSS e <i>JavaScript</i> .
<i>Django</i>	Um <i>framework web</i> de alto nível, desenvolvido em Python, que facilita e agiliza o processo de criação de sites.
<i>PostgreSQL</i>	Sistema gerenciador de banco de dados relacional, de código aberto e “ <i>free</i> ”, que apresenta bastante recursos avançados, muito utilizado no mercado e academicamente.
<i>Heroku</i> ⁷	Plataforma orientada por serviço <i>PaaS (Platform as a Service)</i> e que possibilita aos desenvolvedores construir, executar e operar aplicações totalmente na nuvem.

Utilizando os recursos mencionados acima, foi desenvolvido um protótipo do *front-end* que permite estabelecer o controle de vagas nos centros de acolhimento às pessoas em situação de rua. O acesso ao sistema é permitido somente aos usuários que tenham um “login” e uma “senha”, ou seja, às pessoas que tenham algum vínculo empregatício com o centro de acolhimento, conforme pode ser observado na Figura 14.

Figura 14. Padrão de acesso ao sistema

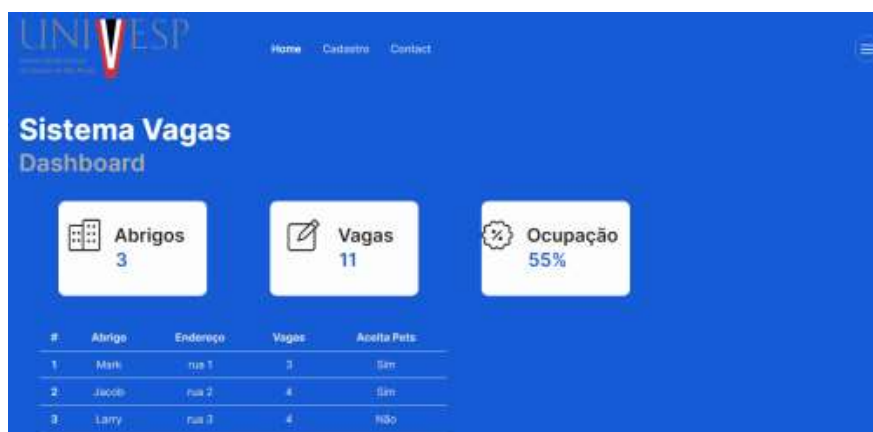


The image shows a login form for 'Sistema Vagas' on a blue background. At the top left is the 'UNIVESP' logo. The title 'Sistema Vagas' is in large white letters. Below it, the word 'Login' is in smaller white letters. There are two white input fields: the first is labeled 'Login' and the second is labeled 'Password'. Below these fields is a black button with the word 'ENVIAR' in white capital letters.

Fonte: Elaborado pelo autor.

Os usuários habilitados, para acesso ao sistema, poderão acompanhar a quantidade de centros de acolhimento (abrigos), o total de vagas disponíveis e a taxa de ocupação e desocupação nos centros. Esses valores são exemplificados na Figura 15. Também é responsabilidade dos usuários habilitados realizar o cadastro das características inerentes a cada centro de acolhimento e do total de vagas que são oferecidas, assim como, das características referente às pessoas em situação de rua, que porventura, poderão utilizar as dependências de um determinado centro. Cabe ressaltar que o sistema fará o controle de vagas nos centros e uma pessoa somente será alojada em um determinado centro se ele possuir vagas disponíveis. A Figura 16 apresenta o modelo de cadastramento para ambas as situações.

Figura 15. Acompanhamento de vagas nos centros de acolhimento



The image shows a dashboard for 'Sistema Vagas' on a blue background. At the top left is the 'UNIVESP' logo. To the right of the logo are links for 'Home', 'Cadastro', and 'Contact'. Below the logo is the title 'Sistema Vagas' and the subtitle 'Dashboard'. There are three white boxes with icons and text: 'Abrigos 3' (with a building icon), 'Vagas 11' (with a notepad icon), and 'Ocupação 55%' (with a percentage icon). Below these boxes is a table with 5 columns: '#', 'Abrigo', 'Endereço', 'Vagas', and 'Aceita Pets.'.

#	Abrigo	Endereço	Vagas	Aceita Pets.
1	Mark	rua 1	3	Sim
2	Jacobi	rua 2	4	Sim
3	Larry	rua 3	4	Não

Fonte: Elaborado pelo autor.

Figura 16. Cadastramento de centros e pessoas em situação de rua

Realizar Cadastro de Vagas

Cadastro Abrigo

Nome abrigo
Nome:

Endereço abrigo
Endereço:

Vagas disponíveis
Vagas disponíveis:

Accepta animais de estimação (PETI)
☐ Sim
☐ Não

ENVIAR **CLEAR**

Cadastro Pessoas

Nome Pessoa
Nome:

Sexo
☐ Masculino
☐ Feminino
☐ Não Definido

Data nascimento
dd/mm/aaaa

Tem animal de estimação(PETI)
☐ Sim
☐ Não

Qual Abrigo

ENVIAR **CLEAR**

Fonte: Elaborado pelo autor.

Por fim, a Figura 17 exibe os canais que poderão ser utilizados para comunicar ou direcionar qualquer evento relacionado às pessoas em situação de rua.

Figura 17. Canais de comunicação com os centros de acolhimento

Contato em caso de ajuda.

Contato
 0084537276023
 younmail@gmail.com

Endereço
 Av. Marechal Rondon, 263 -
 Centro, Osasco - SP, 06083-020
 SP-Brasil

Fonte: Elaborado pelo autor.

3.3.1 Interoperabilidade framework web e IoT

O processo de interação entre os vários “agentes” em toda a estrutura proposta para o PI pode ser observado na Figura 18. O usuário interage com o *front-end* para enviar e receber requisições relacionadas às particularidades de cada centro de acolhimento. O *front-end*, estabelece uma comunicação com o *back-end* para recuperar ou armazenar as informações referentes a cada centro como, nome, endereço, vagas disponíveis e se aceita animais de estimação. Cabe ao *front-end*, também, disponibilizar as informações para o dispositivo IoT, conforme apresentado na Figura 19.

Figura 18.Estrutura de comunicação do sistema.



Fonte: Elaborado pelo autor.

Figura 19.Modelo de mensagem apresentada no dispositivo IoT

```
{ "abrigo": "Santo Antonio", "endereço": "Snt Amaro 1000", "vagas": "2", "Pet" : "Sim" }
```

Fonte: Elaborado pelo autor.

Ressalta-se que os centros de acolhimento às pessoas em situação de rua são referidos no *framework* como abrigos . Classificado, ainda, como um protótipo e com um embasamento acadêmico, o framework está disponível na plataforma do Heroku e o acesso pode ser realizado por meio do link:

- <https://proj-int-iii-univesp.herokuapp.com/>.

A Figura 20 apresenta a tela inicial do framework web, onde é possível observar o percentual de vagas disponíveis nos centros de acolhimento cadastrados. O acesso ao sistema é realizado por meio de uma conta de e-mail e senha, e caso não possua uma, deve-se efetuar o cadastramento, como apresentado nas Figura 20 e Figura 21, respectivamente. Observa-se, ainda, na Figura 20, a opção “Esqueceu a senha?”, a qual possibilita ao usuário cadastrar uma nova senha no caso de esquecimento. Destaca-se que a operacionalização do sistema do framework web é bastante intuitiva e amigável.

Figura 20.Entrada (login e cadastro) ao sistema do framework web

Fonte: Elaborado pelo autor.

Figura 21. Padrão de acesso ao sistema do framework web

Controle de Abrigos Entrar Cadastrar-se

Entrar

E-mail*

Endereço de e-mail

Senha*

Senha

Entrar [Esqueceu a senha?](#)

Fonte: Elaborado pelo autor.

Figura 22. Cadastramento de usuário no sistema do framework web

Abrigo	Vacância
Abrigo Santa Maria	100%
Abrigo São José	95%

Fonte: Elaborado pelo autor.

Os usuários habilitados para acesso ao sistema, após efetuarem o processo de autenticação serão redirecionados para tela apresentada na Figura 22. Observa-se que esses usuários poderão visualizar os centros de acolhimento cadastrados e a disponibilidade de vagas em termos percentuais. Poderão, também, cadastrar, acompanhar e atualizar características relacionadas aos abrigos, aos abrigados e aos dispositivos *IoT*, bem como, a distância que esses dispositivos estão em relação aos abrigos.

Na opção “Abrigos” é possível visualizar especificamente as características referentes a cada abrigo cadastrado no sistema, conforme apresentado na Figura 23. A atualização de algum dado referente a um determinado abrigo pode ser efetuada por meio da opção “Editar”, ampliada na Figura 24 e o cadastramento de um novo abrigo está contemplada na opção “Cadastrar”. Por fim, há a possibilidade de exclusão de um centro de acolhimento do sistema.

Figura 23. Opção para visualizar características dos abrigos

#	Nome	Endereço	CEP	Bairro	Cidade	Pct	Total	Vagas	Ocupação	Ação
2	Abrigo Santa Maria	Rua Santa Maria, 100	08100-100	Vila Santa Maria	Osasco	São	30	30	100%	Editar Excluir
1	Abrigo São José	Rua São José, 100	08100-100	Vila São José	Osasco	São	20	19	95%	Editar Excluir
3	Abrigo Centro	Rua Mariana L.M. Farias, 418	02051-010	Centro	Osasco	São	40	19	47%	Editar Excluir

Fonte: Elaborado pelo autor.

Figura 24. Opção para atualizar dados de um abrigo

Nome*
Abrigo Centro

Endereço*
Rua Mariana L.M. Farias, 418

CEP*
02051-010

Bairro*
Centro

Cidade*
Osasco

Pct*
São

Total*
40

Vagas*
19

Ocupação*
47%

[Salvar](#)

Fonte: Elaborado pelo autor.

Os dispositivos *IoT* que indicarão a disponibilidade de vagas nos centros de acolhimento deverão ser registrados no sistema do *framework web*. A finalidade é cadastrar, visualizar e atualizar os dados sobre os dispositivos no sistema para que as informações sobre o endereço e as vagas disponíveis em um determinado centro possam ser redirecionadas por meio de uma conexão *Wi-Fi*, por exemplo, a esses dispositivos, os quais estão instalados fisicamente em localizações estratégicas que deverão ser definidas pelos órgãos responsáveis.

Para que o dispositivo *IoT* indique o centro de acolhimento mais próximo, o sistema do *framework web* necessita que além da localização física, seja indicada a distância, em quilômetros, em que esses dispositivos estão em relação aos centros de acolhimento. Essa distância somente é registrada entre os dispositivos e centros que estão cadastrados no

sistema. As figuras abaixo apresentam o escopo e o contexto em que esses procedimentos estão inseridos.

Figura 25. Opção para visualizar a distância entre os dispositivos e os centros



#	Distância	Abrigo	Dispositivo	Ação
1	2 Km	Abrigo Santa Maria	ID1	Editar Excluir
2	3 Km	Abrigo São José	ID1	Editar Excluir
3	5 Km	Abrigo Centro	ID1	Editar Excluir

Fonte: Elaborado pelo autor.

Por fim, por questões de segurança é possível que um usuário cadastrado no sistema efetue, periodicamente, uma atualização, ou seja, troca da senha cadastrada anteriormente

Figura 26. Atualização de senha



Trocar Senha

Senha Atual*

Nova Senha*

Nova Senha (novamente)*

[Trocar Senha](#) [Esqueceu a senha?](#)

Fonte: Elaborado pelo autor.

4 SOLUÇÃO COMPLEMENTAR

A partir da solução inicial foi proposta a complementação do dispositivo *IoT* com um botão de emergência e o desenvolvimento de um *dashboard* para visualização dos dados coletados do dispositivo.

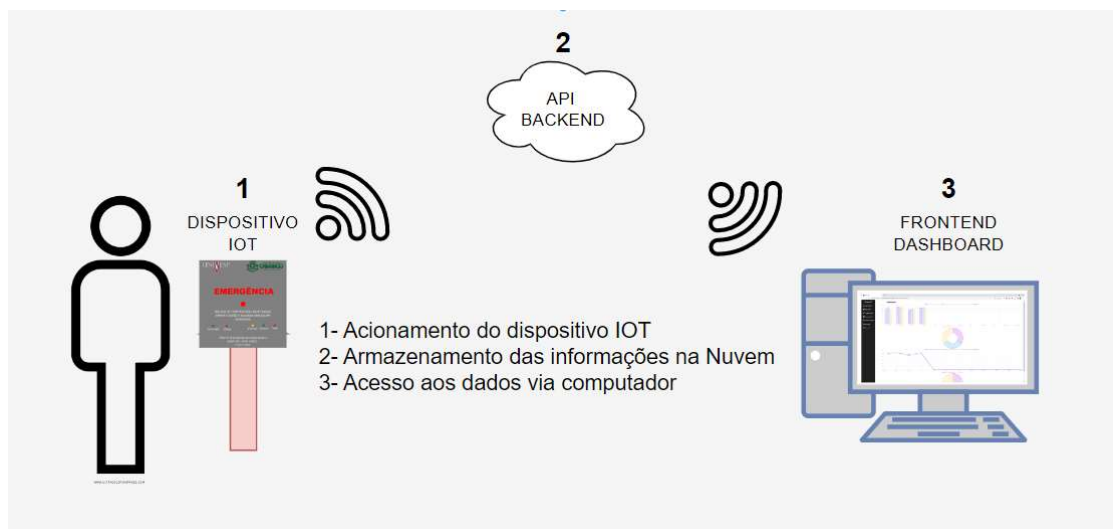
4.1 Desenvolvimento e Performance

Nesta etapa, o projeto foi subdividido em três itens:

- Desenvolvimento do Dispositivo *IoT* incluindo o botão de emergência;
- Desenvolvimento da API (*Application Programming Interface*), utilizada como *back-end*;
- Desenvolvimento do *Front-end*, com implementação de um *dashboard* para visualização de dados;

Quando o botão de emergência é acionado o dispositivo *IoT* envia uma requisição para a API (localizada na nuvem), que por sua vez registra no banco de dados a temperatura e o local no instante do acionamento. Através de um terminal (computador ou celular) é possível acessar o *front-end* e verificar todos os acionamentos registrados, bem como os gráficos gerados no *dashboard*.

Figura 27. Ilustração do projeto



Fonte: Elaborado pelo autor.

4.1.1 Dispositivo IoT

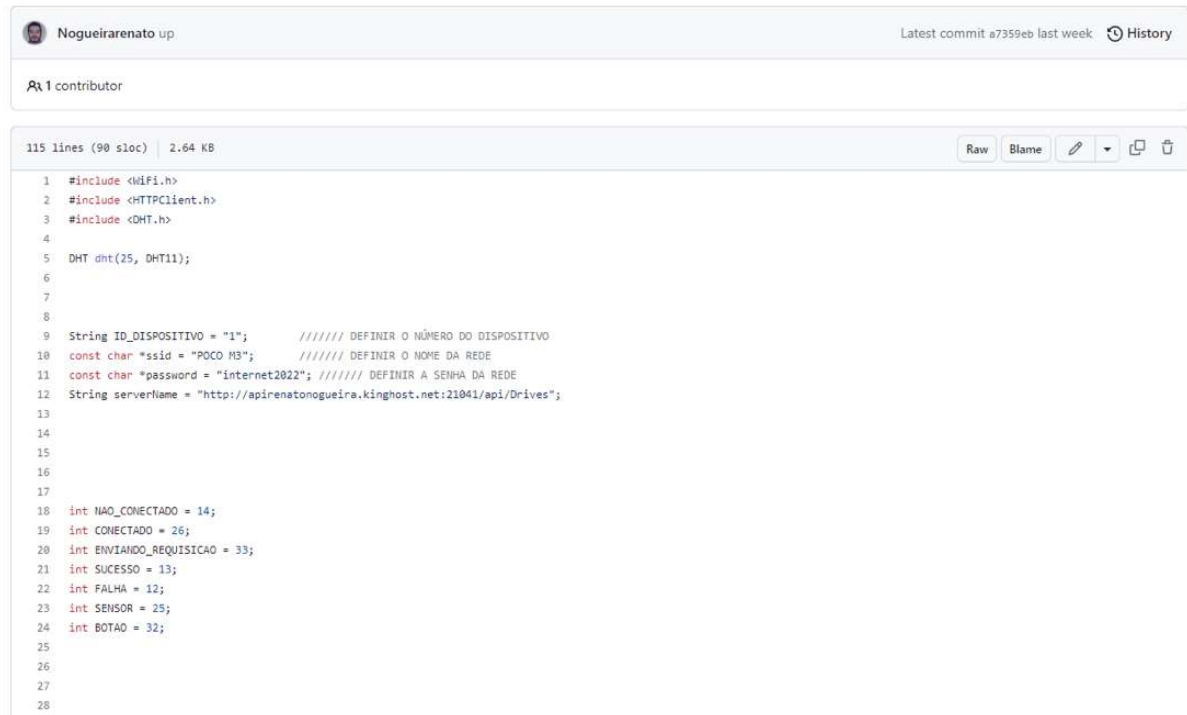
O dispositivo *IoT* foi desenvolvido utilizando uma placa ESP32 conectado à internet pelo WI-FI, e quando o botão é acionado o sensor de temperatura e umidade (DHT11) faz a leitura da temperatura enviando uma requisição POST, no formato JSON, para a API.

A programação do dispositivo foi feita utilizando a linguagem C++.

O código fonte encontra-se no seguinte link:

- <https://github.com/Nogueirarenato/ProjetoIntegradorIV/blob/main/ESP32/driver.c/driver.c.ino>

Figura 28. Códigos desenvolvidos no Github



```

1  #include <WiFi.h>
2  #include <HTTPClient.h>
3  #include <DHT.h>
4
5  DHT dht(25, DHT11);
6
7
8
9  String ID_DISPOSITIVO = "1";          // Definir o número do dispositivo
10 const char *ssid = "POCO M3";         // Definir o nome da rede
11 const char *password = "internet2022"; // Definir a senha da rede
12 String serverName = "http://api.nogueira.ghost.net:21041/api/Drives";
13
14
15
16
17
18 int NAO_CONECTADO = 14;
19 int CONECTADO = 26;
20 int ENVIANDO_REQUISICAO = 33;
21 int SUCESSO = 13;
22 int FALHA = 12;
23 int SENSOR = 25;
24 int BOTAO = 32;
25
26
27
28

```

Fonte: Elaborado pelo autor.

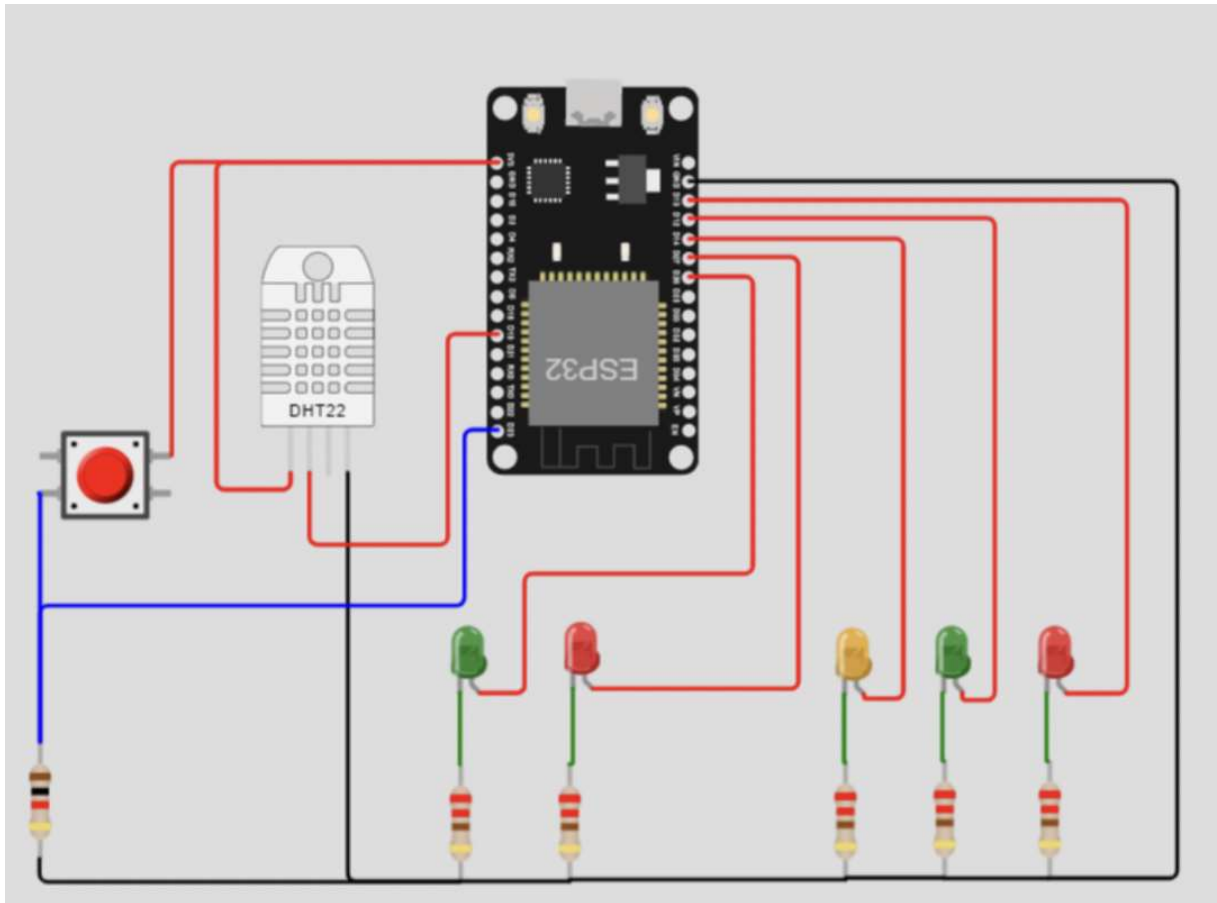
Para a montagem do dispositivo foram utilizados os componentes da Tabela 2, montados de acordo com o diagrama exibido na Figura 29.

Tabela 2. Componentes utilizados no dispositivo IoT. Fonte: Elaboração do autor.

Quantidade	Componentes
1	Microcontrolador ESP32
1	Sensor de temperatura e umidade (DHT11)
2	LED's vermelhos
2	LED's verdes
1	LED's laranja
1	Botão push
5	Resistores de 220 Ohms

1	Resistor de 1000 Ohms

Figura 29. Diagrama de interligação



Fonte: Elaborado pelo autor.

Figura 30. Layout da vista frontal do dispositivo



Fonte: Elaborado pelo autor.

A figura 30 mostra o layout dos LED's no dispositivo *IoT* e abaixo está descrito o comportamento dos LED's de acordo com o funcionamento do sistema:

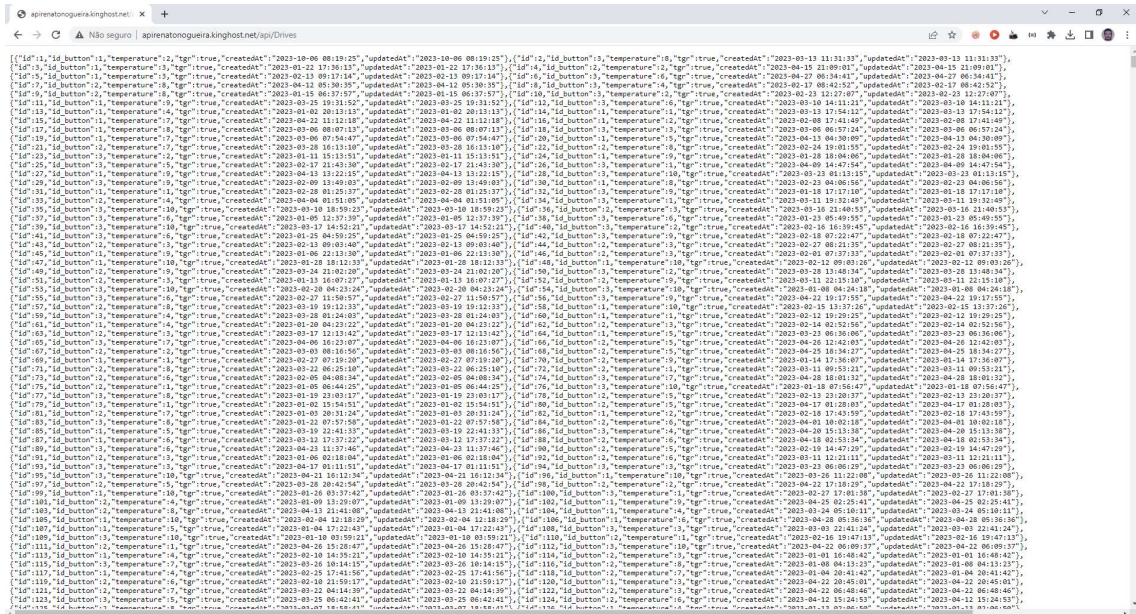
- LED “Conectado” aceso, significa que o dispositivo está conectado à internet
- LED “Offline” aceso, significa que o dispositivo não está conectado à internet;
- LED “Enviando” aceso, significa que, após o botão de emergência ter sido acionado, o dispositivo está tentando se comunicar com a API.
- LED “Sucesso” piscando, significa que a API recebeu com êxito a requisição enviada pelo dispositivo.
- LED “Falha” piscando, significa que houve alguma falha no processo de envio da requisição para a API.

4.1.2 API (Back-end)

A API foi desenvolvida utilizando Node.js (*javascript* voltado para o *back-end*), e foi hospedada no servidor em nuvem *Kinghost*. O *endpoint* para envio de requisições é:

- <http://apirenatonogueira.kinghost.net/api/Drives>

Figura 31. Visualização API



Fonte: Elaborado pelo autor.

Através de requisições do tipo POST (exemplificadas pela Figura 31) os dados são salvos no banco de dados e através de requisições do tipo GET os dados são listados no formato JSON. O código fonte da API encontra-se no link:

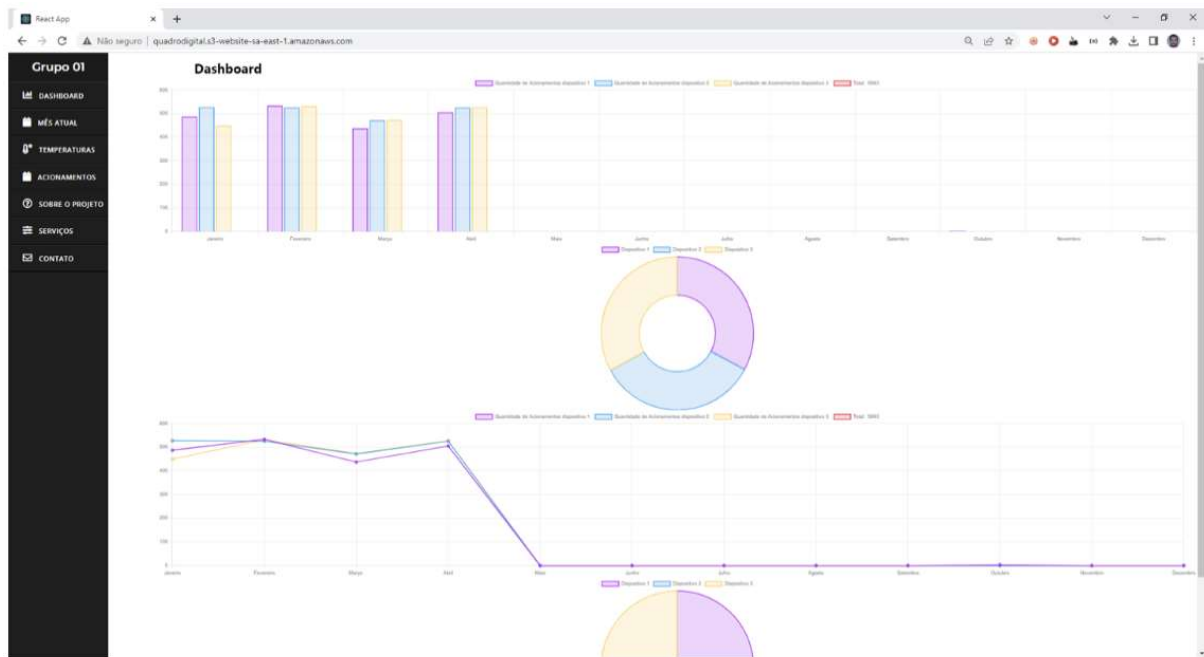
- <https://github.com/Nogueirarenato/ProjetoIntegradorIV/tree/main/ApiAcionador>

4.1.3 Dashboard (Front-end)

O *front-end* foi desenvolvido utilizando a biblioteca REACT (que utiliza a linguagem *Javascript*), e foi armazenado na nuvem da AWS (Amazon Web Service). Através do *front-end*, é possível fazer a visualização de todos os acionamentos e a análise de dados através de gráficos do *dashboard*, conforme exibido pela Figura 32. O código para do *front-end* está disponível em:

- <http://quadrodigital.s3-website-sa-east-1.amazonaws.com/>

Figura 32. Funcionamento do dashboard para análise de dados



Fonte: Elaborado pelo autor.

O código fonte do *front-end* está disponível no link:

- <https://github.com/Nogueirarenato/ProjetoIntegradorIV/tree/main/dash>

4.1.4 Testes para Validação e Performance

Para a execução de testes do sistema foi desenvolvido um programa auxiliar utilizando Linguagem C, o programa gera registros que podem ser enviados em lote para a API, o código fonte do programa auxiliar é exibido na Figura 33 e pode ser acessado através do link:

- <https://github.com/Nogueirarenato/ProjetoIntegradorIV/blob/main/ProgramasAuxiliares/geradorDeRegistros.c>

Figura 33.Código e execução dos Testes

```
36 lines (27 sloc) | 980 Bytes
Raw Blame
1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4
5 void main(){
6
7     int maximo = 0;
8     int botao;
9     float temperatura;
10    int mes;
11    int dia;
12    int hora;
13    int minuto;
14    int segundo;
15
16    printf("INSERT INTO drives(id_button, temperature, tgr, createdAt, updatedAt) VALUES (1, 2.0, true, '2023-10-6 8:19:25', '2023-10-6 8:19:25');");
17
18    while(maximo <= 3000){
19
20        botao = 1+rand()%3;
21        temperatura = 1+ rand()%10;
22        mes = 1+rand()%12;
23        dia= 1+rand()%28;
24        hora= 1+rand()%23;
25        minuto= 1+ rand()%59;
26        segundo = 1+ rand()%59;
27
28        printf("(%d, %.1f, true, '2023-%d-%d %d:%d:%d', '2023-%d-%d %d:%d:%d');", botao, temperatura, mes, dia, hora, minuto, segundo, mes, dia, hora, minuto, segundo );
29        //printf("(2, %.1f, true, '2023-%d-%d %d:%d:%d', '2023-%d-%d %d:%d:%d');", temperatura, mes, dia, hora, minuto, segundo, mes, dia, hora, minuto, segundo ); //forçando o bo
30
31        maximo++;
32    }
33
34    printf("(1, 2.0, true, '2023-10-6 8:19:25', '2023-10-6 8:19:25');");
35
36 }
```

Fonte: Elaborado pelo autor.

5 CONCLUSÕES

As questões relacionadas à população em situação de rua vão muito além de ser um problema majoritariamente social. A privação de direitos básicos que faz com que as condições de vida e de sobrevivência sejam bastante precárias amplia o enfoque para outras questões importantes que se caracterizam pela exclusão social revelando a marginalização, a discriminação e a falta de oportunidades em uma sociedade onde a desigualdade é bastante evidente.

Diante dessa indubitável carência de direitos básicos, algumas entidades de assistência social buscam desempenhar um papel para minimizá-las oferecendo meios que possam reconduzir essas pessoas marginalizadas a um convívio social razoavelmente aceitável.

Este trabalho de PI buscou atender especificamente a uma demanda da população em situação de rua. Por meio de discussões virtuais realizadas com os integrantes do grupo definiu-se o escopo do projeto e a escolha da comunidade externa, requisitos essenciais para a elaboração e desenvolvimento do PI. Neste contexto, surge o Departamento de Proteção Social Especial (DPSE) da Secretaria de Assistência Social da Prefeitura de Osasco que é responsável pelo acolhimento de pessoas adultas, crianças e adolescentes, de ambos os sexos, em situação de rua e desabrigo por abandono, migração e ausência de residência ou pessoas em trânsito e sem condições de se auto-sustentar.

Desta forma, para que a assistência oferecida pelo DPSE possa ser mais efetiva este PI buscou apresentar uma solução para os seguintes problemas:

“Como melhorar e ampliar o canal de comunicação entre a população em situação de rua e os centros de acolhimento no sentido de fornecer publicamente informações sobre a localização e a disponibilidade de vagas nesses centros?”

“Como atender em caráter emergencial a população em situação de rua para os casos onde as pessoas nesta condição estão impedidas de se direcionar aos centros de acolhimento devido a impedimento físico ou condição de saúde agravante?”

O escopo escolhido pelo grupo, que se orientou pelo tema norteador da Univesp, foi desenvolver um framework web incorporando recursos de banco de dados e Javascript e análise de dados com dashboard para auxiliar os colaboradores dos centros de acolhimento a melhorarem continuamente os processos de trabalho para o melhor atendimento às pessoas na situação descrita, além de uma integração com elementos baseados em Internet das Coisas para informar publicamente a população em situação de rua sobre a localização e a

disponibilidade de vagas nos centros de acolhimento, bem como ser um dispositivo de alerta aos centros de acolhimento para atender às situações emergenciais que as pessoas em situação de rua venha a passar, através de um botão de emergência que pode ser acionado diretamente por estas pessoas no dispositivo, acarretando uma diligência das equipes dos centros de acolhimento para socorrer essas pessoas.

Essa proposta de solução para o problema foi bem recebida pelas responsáveis pelo DPSE que vislumbraram uma oportunidade de ampliar o uso da tecnologia para melhorar a gestão operacional dos processos de acolhimento e atendimento a essa população de rua.

Em função da originalidade e inovação deste PI, a construção da solução inicial e final necessita de alguns ajustes e testes mais detalhados para que o protótipo se torne sólido e consistente para a validação por parte da comunidade externa. Contudo, o arcabouço metodológico e as informações mapeadas e catalogadas podem servir de alicerce referencial para a conclusão deste projeto ou, mesmo, para o desenvolvimento de projetos futuros.

Por fim, outra característica importante que norteou todo o desenvolvimento do PI foi uma abordagem fundamentada em técnicas empregadas pelo método Design Thinking para a solução de problemas. O método pressupõe que a solução do problema deve ser norteada pelos seguintes pilares: trabalho coletivo e colaborativo, aprendizagem ativa, identificação de problema em contextos reais e empatia onde as soluções são centradas no ser humano.

REFERÊNCIAS

ACETO, Giuseppe; PERSICO, Valerio; PESCAPÉ, **Antonio**. **A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges**. IEEE Communications Surveys & Tutorials, v. 21, n. 4, p. 3467-3501, 2019.

ARDUINO: **What is Arduino?** Arduino Community. 5 feb. 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 19 nov. 2022.

BARROS, A. J. da S.; LEHFELD, N. A. de S. **Fundamentos de metodologia científica**. São Paulo: Atlas, 2008.

BORBA, Marco; GOBATTI, Sérgio. **Osasco terá operação para abrigar pessoas em situação de rua**. Mai. 2022. Disponível em: <https://osasco.sp.gov.br/osasco-tera-operacao-para-abrigar-pessoas-em-situacao-de-rua/>. Acesso em: 24 set. 2022.

BRASIL. Ministério do Desenvolvimento Social e Combate a Fome (MDS). **Política**

nacional para inclusão social da população em situação de rua. Brasília, DF: MDS, 2008.

_____. Decreto nº 7.053 de 23 de dezembro de 2009. **Institui a política nacional para a população em situação de rua e seu comitê intersetorial de acompanhamento e monitoramento, e dá outras providências.** Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d7053.htm. Acesso em: 25 out. 2022.

DABBS, Mark. **The Fundamentals of Web Application Architecture.** In: BLOG Reinvently. [S.I.], 29 jul. 2019. Disponível em: <https://reinvently.com/blog/fundamentals-web-application-architecture/>. Acesso em: 20 out. 2022.

DAM, Rikke; SIANG, Teo. **What is design thinking and why is it so popular.** Interaction Design Foundation, p. 1-6, 2018.

DANTAS, C. M.; CORDULA, F. R.; ARAÚJO, W. J. **Análise da representação da informação em modelos entidade relacionamento com base em metadados.** Archeion Online, João Pessoa, v. 4, n. 1, p. 40-63, 2016.

DELAMARO, Marcio; MALDONADO, José; JINO, Mário. **Introdução ao teste de software.** 2 ed. Rio de Janeiro: Elsevier Editora Ltda., 2016.

DJANGO: Why Django? **Django Software Foundation.** 2022. Disponível em: <https://www.djangoproject.com/start/overview/>. Acesso em: 20 out. 2022.

ELETRONICS HUB: **Basics of I2C Communication | Hardware, Data Transfer, Configuration.** 8 feb. 2018. Disponível em: <https://www.electronicshub.org/basics-i2c-communication/>. Acesso em: 17 nov. 2022.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados.** São Paulo: Pearson, 2005.

KONNORATE, G. C.; IANKOSKI, L. C.; DE ANDRADE, V. P.; MOREIRA, J. P. A

ELPROCUS: **DHT11 Sensor and Its Working.** [s.d]. Disponível em: <https://www.elprocus.com/a-brief-on-dht11-sensor/>. Acesso em 17 nov. 2022.

ESP32 SERIES: Datasheet. Versão 4.0. Espressif Systems, 2022. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 16 nov. 2022.

FILA, Nicholas D.; MCKILLIGAN, Seda; GUERIN, Kelly. **Design thinking in engineering course design.** In: 2018 ASEE Annual Conference & Exposition. Jun. 2018.

GALADIMA, Ahmad Adamu. **Arduino as a learning tool**. In: 2014 11th International Conference on Electronics, Computer and Computation (ICECCO). Proceedings [...] Abuja, Nigéria: IEEE, 2014. p. 1-4. DOI <https://doi.org/10.1109/ICECCO33708.2014>.

Importância do Controle de Versões no Desenvolvimento de Software. Seminário De Tecnologia Gestão E Educação, v. 1, n. 2, p. 1-4, 2019.

JUSTO, M. G. **Vida nas ruas de São Paulo e alternativas possíveis um enfoque sócioambiental**. InterfacEHS, São Paulo, v. 3, n. 1, p. 1-27, 2008.

KUMAR, Prakash; KUMAR, Pradeep. **Arduino based wireless intrusion detection using IR sensor and GSM**. International Journal of Computer Science and Mobile Computing, v. 2, n. 5, p. 417-424, 2013.

LI, Yuan-Fang; DAS, Paramjit K.; DOWE, David L. **Two decades of Web application testing - A survey of recent advances**. Information Systems, v. 43, p. 20-54, 2014.

LIOTTA, Olga; CASARINI, Fernanda. **Prefeitura começa a abrigar pessoas em situação de rua em hotéis parceiros**. Jun. 2022. Disponível em: <https://osasco.sp.gov.br/prefeitura-comeca-a-abrigar-pessoas-em-situacao-de-rua-em-hoteis-parceiros/>. Acesso em: 24 set. 2022.

MEDEIROS, D. G. de F. **Um framework de agentes de recomendação para sistemas Web**. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte, 2011.

MENDES, Kíssila Teixeira; RONZANI, Telmo Mota; PAIVA, Fernando Santana de. **População em situação de rua, vulnerabilidades e drogas: uma revisão sistemática**. Psicologia & Sociedade, v. 31, 15 p., 2019.

MOTA, R. P. B.; BATISTA, D. **Um mecanismo para garantia de qos na “internet das coisas” com rfid**. In: Brazilian Symposium on Computer Networks and Distributed Systems - SBRC 2013, Brasília. Proceedings [...] Brasília: SBRC, 2013. p. 297–310.

MILETO, E. M.; BERTAGNOLLI, S. de C. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP**. Porto Alegre: Bookman, 2014.

MYSQL: **MySQL 8.0 Reference Manual**. Oracle. 2022. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/introduction.html>. Acesso em: 19 out. 2022.

NIEDERAUER, Juliano. **Desenvolvendo Websites com PHP**. 3. ed. São Paulo: Novatec Editora Ltda, 2017.

NUSSEY, John. **What Is Arduino and Where Did It Come From?** In: Arduino for dummies. Chichester: John Wiley & Sons, 2013. Disponível em: <https://books.google.com.br/books?id=LRWcJH5SPokC&printsec=copyright&hl=pt-BR#v=onepage&q&f=false>. Acesso em: 19 nov. 2022.

O'CALLAGHAN, Gemma; CONNOLLY, Cornelia. **Developing creativity in computer science initial teacher education through design thinking.** In: United Kingdom & Ireland Computing Education Research Conference. 2020. p. 45-50.

PHP: **Manual do PHP.** PHP Documentation Group. 2022. Disponível em: <https://www.php.net/docs.php>. Acesso em: 17 out. 2022.

REGIÃO: **Mais de 500 pessoas moram nas ruas de Osasco.** Giro S/A. Digital, Osasco, 07 de maio de 2020, Metrópole. Disponível em: <https://girosa.com.br/regiao-mais-de-500-pessoas-moram-nas-ruas-de-osasco/>. Acesso em: 18 out. 2022.

RESENDE, Viviane de Melo; MENDONÇA, Daniele Gruppi de. **População em situação de rua e políticas públicas: representações na Folha de São Paulo.** DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada, v. 35, 2019. DOI <https://doi.org/10.1590/1678-460X2019350413>.

RICCA, Filippo; TONELLA, Paolo. **Analysis and testing of web applications.** In: 23RD INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING. 2001, Toronto. Proceedings [...]. Toronto: IEEE Computer Society, 2001. p. 25-34.

RUFFO, Ricardo. **Prototipação Design Thinking: 5 motivos para prototipar qualquer coisa.** Echos School, 23 mar. 2016. Disponível em: <https://escoladesignthinking.echos.cc/blog/2016/03/5-motivos-para-voce-prototipar-qualquer-coisa-2/>. Acesso em: 04 out. 2020.

SCHWEITZER, Lucas; TOLFO, Suzana da Rosa. **Rualização: vivências de pessoas em situação de rua.** ECOS-Estudos Contemporâneos da Subjetividade, v. 12, n. 1, p. 54-65, 2022.

SWAROOP, C.H. **About Python.** [s.d.]. Disponível em: https://python.swaroopch.com/about_python.html. Acesso em: 17 out. 2022.

SILVA, Carla Regina; SILVESTRINI, Marina Sanches, VON POELLNITZ, Jéssica Cristina, PRADO, Ana Carolina da Silva Almeida, LEITE JUNIOR, Jaime Daniel. **Estratégias criativas e a população em situação de rua: terapia ocupacional, arte, cultura e**

deslocamentos sensíveis. Cadernos Brasileiros de Terapia Ocupacional, v. 26, p. 489-500, 2018.

TARATUKHIN, Victor; YADGAROVA, Yulia; BECKER, Joerg. **The Internet of things prototyping platform under the design thinking methodology.** In: 2018 ASEE Annual Conference & Exposition. Jun. 2018.

THANGAVEL, Pramoth. **I2C Communication between two Arduino Boards.** 2018. Disponível em: <https://circuitdigest.com/microcontroller-projects/arduino-i2c-tutorial-communication-between-two-arduino>. Acesso em: 17 nov. 2022.

THORING, Katja; MULLER, Roland M. **Understanding design thinking: A process model based on method engineering.** In: 13TH INTERNATIONAL CONFERENCE ON ENGINEERING AND PRODUCT DESIGN EDUCATION - DS 69. 2011, London. Proceedings [...]. London: E&PDE, Sep. 2011. p. 493-498.

VALENTE, Bruno A. L. **Um middleware para a Internet das coisas.** 2011. Dissertação (Mestrado em Informática) – Faculdade de Ciências, Departamento de Informática, Universidade de Lisboa, Lisboa, 2011. Disponível em: <https://repositorio.ul.pt/handle/10451/9211>. Acesso em: 20 out. 2022.

WIKIPEDIA: **Placa de ensaio.** 27 out. 2022. Disponível em: https://python.swaroopch.com/about_python.html. Acesso em: 29 nov. 2022.

APÊNDICE – DISPOSITIVO *IoT*

Abaixo é apresentada a última versão do dispositivo *IoT* desenvolvida no presente trabalho, que por ser um projeto piloto e acadêmico, está sujeito a atualizações.

Figura 34. Visão geral do protótipo conectado via rede wi-fi



Fonte: Elaborado pelo autor

Figura 35. Sensor DHT de temperatura



Fonte: Elaborado pelo autor

Figura 36. Visão após o botão ser acionado



Fonte: Elaborado pelo autor

Figura 37. Em caso de sucesso, o LED piscará por 40s



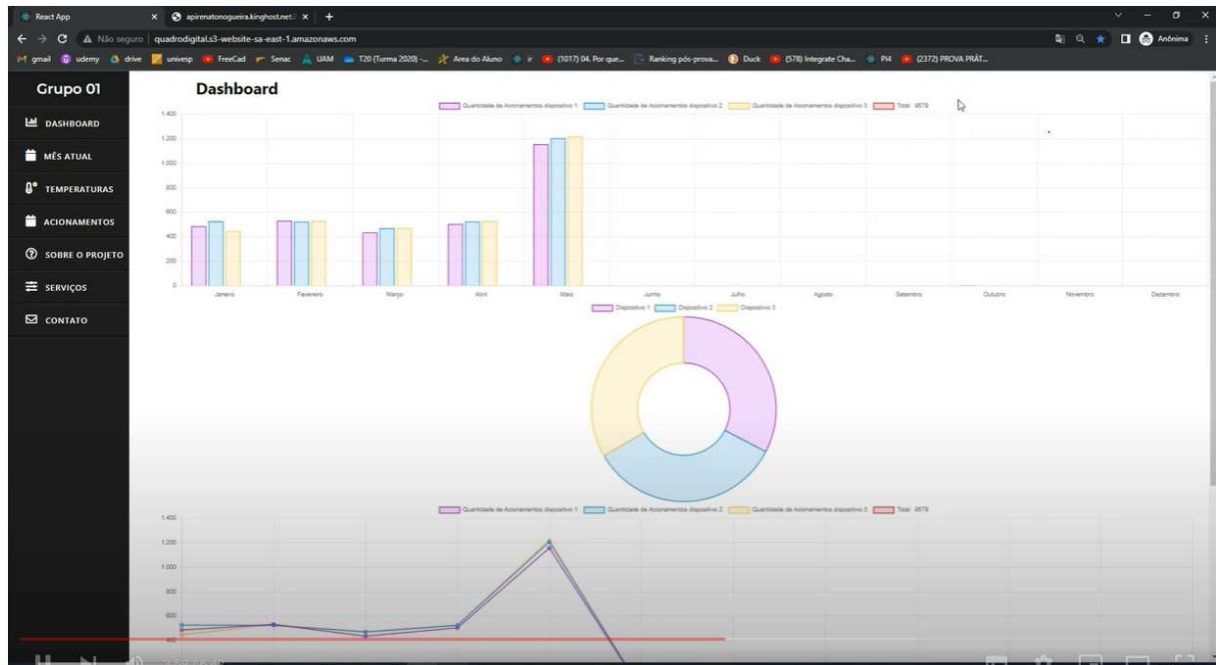
Fonte: Elaborado pelo autor

Figura 38. Em caso de falha, o LED piscará por 40s



Fonte: Elaborado pelo autor

Figura 39. Visão do Dashboard



Fonte: Elaborado pelo autor