

AQUÁRIO VIRTUAL: SIMULADOR DE ECOSSISTEMA UTILIZANDO INTERFACE DE USUÁRIO TANGÍVEL

Flávio Omar Losada, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

flavio.losada@outlook.com, dalton@furb.br

Resumo: Este artigo apresenta o desenvolvimento do projeto de construção de um Aquário Virtual como um simulador de ecossistema que tem por objetivo simular o ecossistema de um aquário real utilizando IUT – Interface de Usuário Tangível para manipulação das propriedades do simulador. O usuário utiliza dos sensores e atuadores disponibilizados para modificar o ecossistema com objetivo de mantê-lo em funcionamento. Foi desenvolvido utilizando o módulo NodeMCU com microcontrolador ESP8266 para construção do módulo de IUT e Unity3D para desenvolvimento do simulador do aquário. Os objetivos foram atingidos, sendo que houve sucesso na conexão entre o módulo IUT e o simulador e a sua utilização se deu de forma dinâmica e interativa para controle do ecossistema. Extensões também são existentes, uma vez que foi desenvolvido toda a comunicação entre as plataformas, sendo possível incrementar novas funcionalidades e melhorias.

Palavras-chave: Aquário virtual. Interface de usuário tangível. Simulador de ecossistema. Simuladores na educação.

1 INTRODUÇÃO

Nos dias de hoje, com avanço significativo da tecnologia e smartphones, tablets e computadores pessoais muito mais acessíveis, nota-se que estas tecnologias estão muito mais presentes no crescimento das crianças. De acordo com Fernandes (2018, p. 1) “Logo após o nascimento, muitas crianças em contexto urbano são inseridas em um espaço tecnológico, desenvolvendo-se nesse ambiente povoado por aparatos digitais.”. Uma vez que este tipo de tecnologia é acessível às crianças, tem-se buscado cada vez mais utilizá-las de forma educacional, para auxílio em atividades de ensino. Conforme Mattei (2013, p. 2), o “[...] uso adequado, oportuniza o desenvolvimento e a organização na construção do pensamento, bem como, desperta o interesse e a curiosidade dos alunos. [...]”.

Uma das formas de trabalhar no ambiente educacional utilizando tecnologia é abordar o recurso de IUT - Interface de Usuário Tangível - do inglês, TUI - Tangible User Interface. A IUT permite que o ambiente real e o ambiente virtual estejam diretamente conectados. Nunes, Radicchi e Botega (2011, p. 27) descrevem IUT como “aquelas que compreendem interações realizadas em artefatos físicos, como estímulos para interferir no contexto e representações de informação digital”, ou seja, interações realizadas no ambiente real impactam no ambiente virtual em questão. As vantagens de se utilizar IUT é que as crianças aprendem em sua forma natural, utilizando vários sentidos (audição, tato e visão) em um processo construtivo. Aumenta a acessibilidade para crianças mais novas e instiga o trabalho em equipe, pois permite que mais de uma criança possa interagir (ZUCKERMAN; ARIDA; RESNICK, 2005).

Ainda no ambiente educacional, pode-se utilizar de jogos e simuladores para uma aprendizagem diferenciada e dinâmica. Um jogo pode gerar diversão e motivar as pessoas a aprender. Através de jogos baseados em regras, direcionados a objetivos, interativos e desafiadores em ambientes de simulação dinâmica e participativa é possível atingir a motivação e diversão (HSIAO, 2007). Portanto o desenvolvimento de um simulador de ecossistema permite a exploração de diversos acontecimentos em um único local. Um simulador baseado em aquário proporciona ideias sobre como manter os peixes, mudanças de temperatura e luminosidade, poluição, entre outros itens que instigam a atenção e motivação.

O objetivo deste projeto consistiu em construir um simulador de ecossistema baseado em um aquário onde os controles sobre este simulador são feitos a partir de recursos de IUT, utilizando sensores e atuadores como entradas para modificar o estado do aquário virtual. Os objetivos específicos são: aprimorar um aquário virtual desenvolvido na plataforma Unity, incluindo interações do usuário e comportamentos reativos; facilitar a comunicação entre o kit de IUT e o aquário virtual por meio de classes que podem ser disponibilizadas; permitir mudanças dinâmicas no ecossistema do aquário instigando a percepção e reação através das interações com o simulador.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresentará os conceitos para entendimento da pesquisa realizada. Na seção 2.1 é comentado sobre Interface de Usuário Tangível – IUT e suas aplicações no meio educacional. Na seção 2.2 é abordado o tema de jogos e simuladores aplicados à educação e como estes podem influenciar no ensino e aprendizagem, na seção 2.3 são abordados

os componentes eletrônicos utilizados no projeto. Por fim, na seção 2.4 são listados e comentado sobre os trabalhos correlatos a este projeto.

2.1 INTERFACE DE USUÁRIO TANGÍVEL NA EDUCAÇÃO

Existem diversas formas de aplicar tecnologia na educação de crianças e adolescentes, uma dessas formas é utilizando Interface de Usuário Tangível (IUT). O recurso de IUT permite interagir com um ambiente virtual por meio de ações realizadas no ambiente real. Como descrito por Falcão e Gomes (2007, p. 579) “A ideia é embutir elementos computacionais em materiais concretos, criando um novo grupo de recurso didático que une as vantagens da manipulação física à interação e multimídia providas pela tecnologia.”. Ou seja, resultando em um ambiente mais palpável para interação com a máquina.

É possível explorar uma forma de ensino e aprendizagem diferenciada quando se utiliza IUT, “exploração e manipulação de objetos físicos é a componente chave para o aprendizado das crianças.” (REVELLE et al., 2015, p. 2051, tradução nossa). Considerando que “as crianças aprendem de forma natural, usando vários sentidos (toque, visão, audição) em um processo construtivo que aumenta a retenção e transferência do conteúdo” (ZUCKERMAN; ARIDA; RESNICK, 2005, p. 860, tradução nossa), entende-se que o uso de uma tecnologia palpável, composta por objetos físicos pode auxiliar no aprendizado. Portanto, a utilização de tecnologias na educação, mais especificamente IUT, podem trazer a atenção do usuário para o aprendizado, “a associação com o lúdico e a possibilidade de manipulação de objetos concretos costumam gerar grande interesse e engajamento por parte das crianças” (FALCÃO; GOMES, 2007, p. 586).

2.2 JOGOS E SIMULADORES APLICADOS À EDUCAÇÃO

Jogos didáticos e simuladores são também uma forma de explorar o ensino e aprendizagem utilizando a tecnologia. Segundo Savi e Ulbricht (2008) os jogos educacionais proporcionam práticas educacionais atrativas e inovadoras, onde o aluno pode aprender de forma ativa, dinâmica e motivadora.

Portanto, procura-se utilizar jogos e simuladores na educação de forma que as crianças e adolescentes sintam-se mais engajados a participar das atividades. Silva e Passerino (2007, p. 3) defende que “[...] ao utilizar meios lúdicos, cria um ambiente gratificante para o desenvolvimento integral da criança.”. É cada vez mais comum que crianças e adolescentes recebam informações de forma rápida e dinâmica, logo, utilizar-se de jogos no ambiente educacional pode tomar a atenção para o aprendizado.

Sendo assim, utilizando um simulador ou jogo educacional, é possível entreter jovens e crianças ao ensino, uma vez que a informação será recebida de uma maneira mais natural. Pelo fato de que jogos educacionais muitas vezes relacionam aprendizado a um universo lúdico, é possível que sejam criados cenários além do que existe no mundo real, dando uma perspectiva diferenciada ao usuário.

[...] metáforas, analogias, figuras de linguagem, resultados inesperados e exemplos elucidativos podem auxiliar no processamento da informação, instigando o usuário e facilitando o entendimento com o resgate de processos já conhecidos. Em animação e jogos educativos, o uso de recursos gráficos, elementos visuais e simbólicos comuns ao aluno podem ser uma forma de atender a essa categoria, assim como o uso de conceitos de usabilidade e ergonomia. (ALVES; BATTIOLA, 2011, p. 4).

Desta forma, percebe-se que a inclusão de jogos ou simuladores digitais no meio educacional pode proporcionar uma experiência dinâmica e entreter os jovens e crianças no aprendizado.

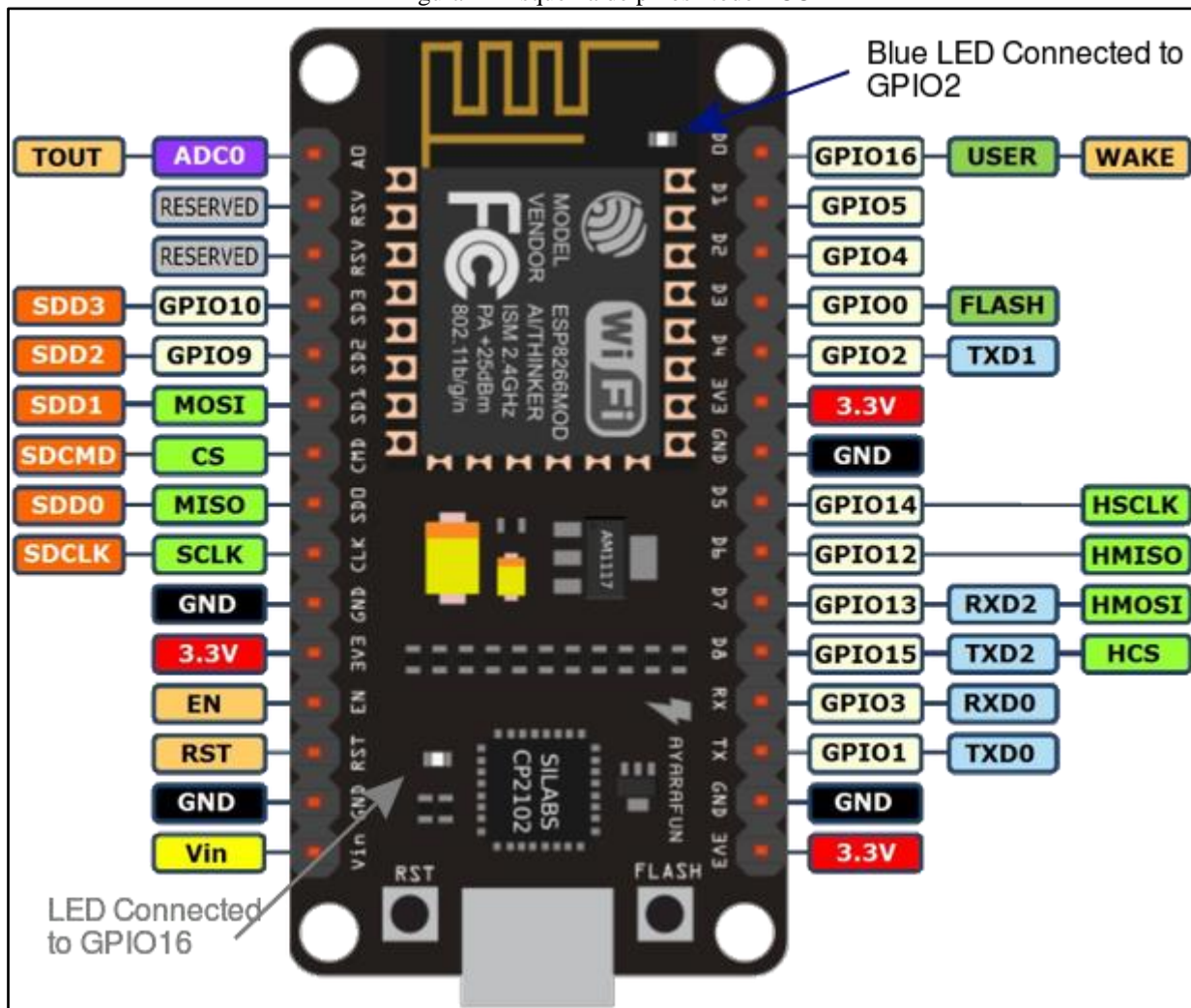
2.3 COMPONENTES ELETRÔNICOS

Para trabalhar com Interface de Usuário Tangível uma das possibilidades é a utilização de componentes eletrônicos conectados a um microcontrolador que pode conectar-se a um ambiente virtual para alteração dos elementos deste ambiente. O microcontrolador ESP8266 acoplado ao módulo NodeMCU pode proporcionar esta conexão. Conforme Yuan (2019) o módulo NodeMCU “[...] é um ambiente de desenvolvimento de software e hardware livre criado em torno de um System-on-a-Chip (SoC) muito econômico chamado ESP8266.”, portanto trata-se de um módulo para desenvolvimento e de baixo custo, que utiliza o microcontrolador ESP8266. O ESP8266 é fabricado pela empresa Espressif Systems (SYSTEMS, 2019), é um System-On-Chip com WiFi embutido, tem conectores GPIO, barramentos I2C, SPI, UART, entrada ADC, saída PWM e sensor interno de temperatura, CPU que opera em 80MHz entre outras especificações. (CURVELLO, 2015).

Por meio dos conectores GPIO o ESP8266 permite a conexão de componentes eletrônicos como sensores e atuadores. O componente LDR é um sensor de luz que modifica a resistência de acordo com a intensidade de luz incidida sobre ele (FILIPEFLOP, 2019), e pode ser conectado ao ESP8266 por meio de GPIO, onde o microcontrolador faz a leitura da tensão e permite trabalhar com valores de 0 a 1023, variável de acordo com a resistência do componente. Desta forma, este componente funciona como um resistor. Resistores são “elementos de um circuito elétrico projetados para transformar energia elétrica em calor.” (SANTOS, 2019), portanto este tipo de componente alteram a tensão que é

transferida para o microcontrolador. Assim como o LDR, temos o potenciômetro, que também se trata de um resistor e tem sua resistência variável mecanicamente. De acordo com Filipeflop (2019) “O Potenciômetro Linear 10K é um componente muito utilizado em circuitos eletrônicos seja para regular a tensão de uma fonte, o contraste de um display LCD ou como um simples divisor de tensão.”.

Figura 1 - Esquema de pinos NodeMCU



Fonte: CIRCUITS4YOU (2017).

As GPIOs numeradas como: 0, 2, 4, 5, 12, 13, 15 e 16 são disponibilizadas para conexão de componentes eletrônicos digitais, já a porta ADC0 permite a conexão de componentes analógicos, que são o caso dos resistores.

2.4 TRABALHOS CORRELATOS

Nesta seção serão descritos os trabalhos correlatos que abordam assuntos de interface de usuário tangível ou jogos e simuladores na educação. No Quadro 1 é apresentado o trabalho desenvolvido por Piske (2015) que aborda um aquário virtual que simula um ecossistema utilizando animação comportamental. No Quadro 2 é apresentado o trabalho TaPrEc – Ambiente de Programação Tangível, desenvolvido por Carbajal e Baranauskas (2015) com objetivo de ensinar programação utilizando recursos de IUT. Por fim, no Quadro 3 é apresentado o Nintendo Labo: Variety Kit desenvolvido pela Nintendo (2019) com intuito de construir jogos ou brinquedos infantis com o conceito DIY – Do It Yourself, em tradução livre: faça você mesmo, utilizando de Interface de Usuário Tangível para interação com mundo virtual.

Piske (2015) desenvolveu um aquário virtual 3D utilizando animação comportamental com objetivo de simular uma cadeia alimentar pequena, tendo um tubarão como predador e sardinha como presa, além de plânctons sendo presas para sardinhas. O simulador de aquário permite que sejam incluídos predadores e presas no ambiente com seus próprios comportamentos e objetivos, possibilitando assim verificar o impacto na cadeia alimentar e no ambiente do aquário, como

mudança da cor d'água e mudança na quantidade de predadores de acordo com o número de presas existentes. O trabalho faz-se correlato a este artigo pelo uso de um simulador de ecossistema com viés educacional.

Quadro 1 – VISEDU – Aquário Virtual: Simulador de ecossistema utilizando animação comportamental

| | |
|--------------------------------|--|
| Referência | Piske (2015). |
| Objetivos | Simulador de ecossistema de aquário marinho utilizando animação comportamental para os personagens. |
| Principais funcionalidades | Aquário virtual simulando ecossistema de um aquário marinho, permitindo a análise do comportamento dos personagens em relação a cadeia alimentar e sobrevivência. Permite adicionar personagens ao aquário como tubarões ou sardinhas. Também é possível alterar a velocidade de movimento dos personagens, bem como o campo de visão. |
| Ferramentas de desenvolvimento | O projeto foi desenvolvido na plataforma VISEDU, utilizando a biblioteca ThreeJS e HTML5. Além disso utiliza a linguagem AgentSpeak para interpretação dos modelos de animação comportamental. |
| Resultados e conclusões | Os objetivos do trabalho foram atingidos. Conforme Piske (2015) “O objetivo principal de criar um aquário virtual que simulasse um ecossistema marinho através de animação comportamental foi contemplado, simulando uma cadeia alimentar pequena que possui o tubarão como predador e a sardinha como presa.”. |

Fonte: elaborado pelo autor.

No projeto Tangible Programming Environment for Children (TaPrEC), Carbajal e Baranauskas (2015) desenvolveram um ambiente para o ensino de programação às crianças utilizando Interface de Usuário Tangível (IUT). O projeto consiste em um ambiente que ^{permite} criar algoritmos utilizando blocos de madeira, criando laços de repetição e blocos condicionais. Os blocos de madeira identificam as funções da linguagem e trazem o conceito de IUT, que faz este projeto se correlacionar a este artigo, que também utiliza IUT com objetivo educacional.

Quadro 2 – TaPrEc: Ambiente de Programação Tangível

| | |
|--------------------------------|---|
| Referência | Carbajal e Baranauskas (2015). |
| Objetivos | Ambiente para ensino de programação utilizando Interface de Usuário Tangível e baixo custo. |
| Principais funcionalidades | Permite desenvolver algoritmos com recursos de linguagens de programação como laços de repetição e blocos condicionais utilizando blocos de madeira como recurso de Interface de Usuário Tangível. |
| Ferramentas de desenvolvimento | Foi utilizado Raspberry Pi para interpretação dos programas desenvolvidos. Cada bloco é associado a uma função da linguagem Scratch. Também é utilizado RFID para identificação de cada bloco de madeira. |
| Resultados e conclusões | Os resultados indicam que o ambiente de programação tangível é atraente para as crianças, fácil e interagir e explorar o pensamento algorítmico. |

Fonte: elaborado pelo autor.

Nintendo Labo: Variety Kit trata-se do produto desenvolvido pela Nintendo (2019) que permite a criação de jogos ou brinquedos utilizando cartões de papelão e um videogame Nintendo Switch. O kit disponibiliza diversos tutoriais e tem o conceito de DIY – Do It Yourself ou Faça Você Mesmo, em tradução livre. Pode ser utilizado para fins educativos desde que supervisionado e traz o conceito de Interface de Usuário Tangível, quando relaciona ambientes do mundo real com ambientes do jogo virtual, relacionando-se ao projeto desenvolvido neste artigo que também relaciona estes elementos.

Quadro 3 – Nintendo Labo: Variety Kit

| | |
|--------------------------------|--|
| Referência | Nintendo (2019). |
| Objetivos | Criar jogos ou brinquedos utilizando o conceito DIY – Do It Yourself, ou Faça Você Mesmo em tradução livre. |
| Principais funcionalidades | As principais funcionalidades do kit Nintendo Labo consistem em permitir a criação de brinquedos utilizando apenas cartões de papelão. Além disso há a necessidade do Nintendo Switch. |
| Ferramentas de desenvolvimento | As ferramentas de desenvolvimento não são acessíveis por ser um produto proprietário. |
| Resultados e conclusões | O projeto possui viés comercial, portanto não foram encontrados resultados e conclusões sobre tal. |

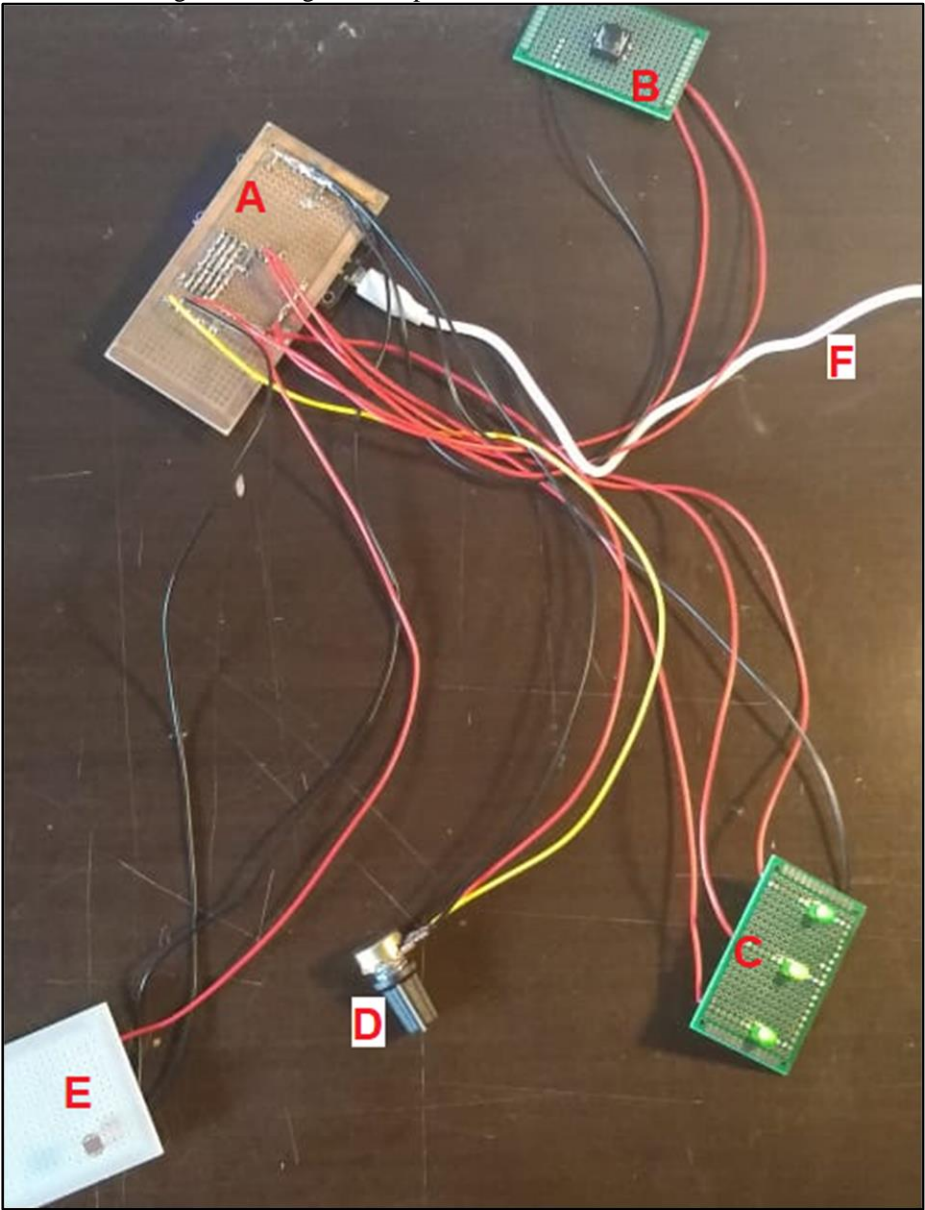
Fonte: elaborado pelo autor.

3 DESCRIÇÃO DO PROJETO

O protótipo se divide em três partes: o módulo de IUT – Interface de Usuário Tangível, que contempla os equipamentos utilizados como sensores e atuadores; a interface de comunicação, que é responsável por transmitir as

informações entre o módulo e o simulador e o Aquário Virtual, que representa um ecossistema de aquário simulado em um ambiente virtual. A Figura 2 apresenta o diagrama simplificado do módulo IUT e o Quadro 4 explica cada item.

Figura 2 - Diagrama simplificado do módulo IUT



Fonte: arquivo pessoal.

Quadro 4 - Especificação dos componentes

| SIGLA | COMPONENTE | ESPECIFICAÇÃO |
|-------|------------------------------|---|
| A | Módulo principal | Trata toda a comunicação entre os demais componentes e o simulador do aquário. É composto por NodeMCU e conversor analógico/digital MCP 3008. |
| B | Botão | Botão utilizado para alimentar os peixes no simulador. Ao pressioná-lo, derruba comida no aquário. |
| C | LEDs de indicação | LEDs utilizados para indicar o status do módulo. Os status são: <u>Ligado</u> , <u>Conectado a WiFi</u> e <u>Conectado ao Simulador</u> . |
| D | Potenciômetro | Resistor potenciômetro utilizado para alterar a temperatura do aquecedor/resfriador do Aquário Virtual. |
| E | LDR – Sensor de Luminosidade | Sensor de luminosidade responsável por captar a intensidade de luz externa e comunicar ao simulador de forma a aplicá-la no ambiente. |
| F | Fonte de alimentação | Fonte de alimentação utilizada para ligar o módulo IUT. Trata-se de um carregador de celular simples com porta micro USB. |

Fonte: elaborado pelo autor

Além dos itens especificados, o Aquário Virtual é composto por um simulador desenvolvido na plataforma Unity3D, o qual se comunica com o módulo principal por meio de comunicação WiFi. As próximas seções deste capítulo descrevem de forma mais detalhada cada parte do projeto.

3.1 MÓDULO DE INTERFACE DE USUÁRIO TANGÍVEL

O módulo de IUT é composto por seus atuadores e sensores, um ADC - Analog/Digital Converter ou Conversor Analógico/Digital e um microcontrolador onde estes equipamentos são conectados. O sensor utilizado foi o LDR – Light Dependent Resistor ou Resistor Dependente de Luz em português, cuja resistência altera de acordo com a intensidade de luz que é incidida sobre ele (FILIPEFLOP, 2019). Como atuadores, foram utilizados um botão e um potenciômetro, que se trata de um resistor com resistência ajustável.

O microcontrolador utilizado foi o módulo NodeMCU, que conta com o processador ESP8266 e módulo Wi-Fi. O motivo do uso deste microcontrolador é justamente a disponibilidade do Wi-Fi e o baixo custo. Houve a necessidade de utilizar um ADC, pois o NodeMCU possui apenas um pino de entrada analógica, enquanto haveria a necessidade de utilizar o LDR e o potenciômetro como fornecedores de valores analógicos ao microcontrolador. O ADC utilizado foi o MCP3008, que ocupa 4 portas digitais do módulo NodeMCU para seu funcionamento, mas disponibiliza 8 entradas analógicas.

Além destes equipamentos, foram utilizadas placas universais para fixação dos componentes, fios para conexão dos componentes, LEDs para indicação de estado do módulo IUT e um carregador de smartphone com cabo microUSB para alimentação do módulo NodeMCU. A Tabela 1 apresenta os componentes utilizados, as quantidades adquiridas e seus valores, considerando que todos foram adquiridos na loja Proesi (PROESI, 2019), com exceção do conversor analógico/digital MCP3008, adquirido no MercadoLivre (MERCADOLIVRE, 2019).

Tabela 1 - Tabela de preços

| Componente | Quantidade | Preço unidade (R\$) | Preço total (R\$) |
|-------------------------------------|------------|---------------------|-------------------|
| Carregador smartphone 2,5 A | 1 | 23,70 | 23,70 |
| Chave tátil 12x12x8,5 mm (botão) | 1 | 0,59 | 0,59 |
| Conversor analógico/digital MCP3008 | 1 | 21,99 | 21,99 |
| Diodo LED Difuso verde 5 mm | 3 | 0,14 | 0,42 |
| Fios flexíveis 0,32 mm | 6 | 0,43 | 2,58 |
| Knob p/ potenciômetro | 1 | 1,89 | 1,89 |
| NodeMCU | 1 | 39,70 | 39,70 |
| Placa PCI Ilhada 10 x 5 | 2 | 3,90 | 7,80 |
| Placa protoboard 40x60 mm Arduino | 2 | 3,89 | 7,78 |
| Potenciômetro | 1 | 0,99 | 0,99 |
| Resistor carbono | 5 | 0,02 | 0,10 |
| Total | | | 107,54 |

Fonte: elaborado pelo autor.

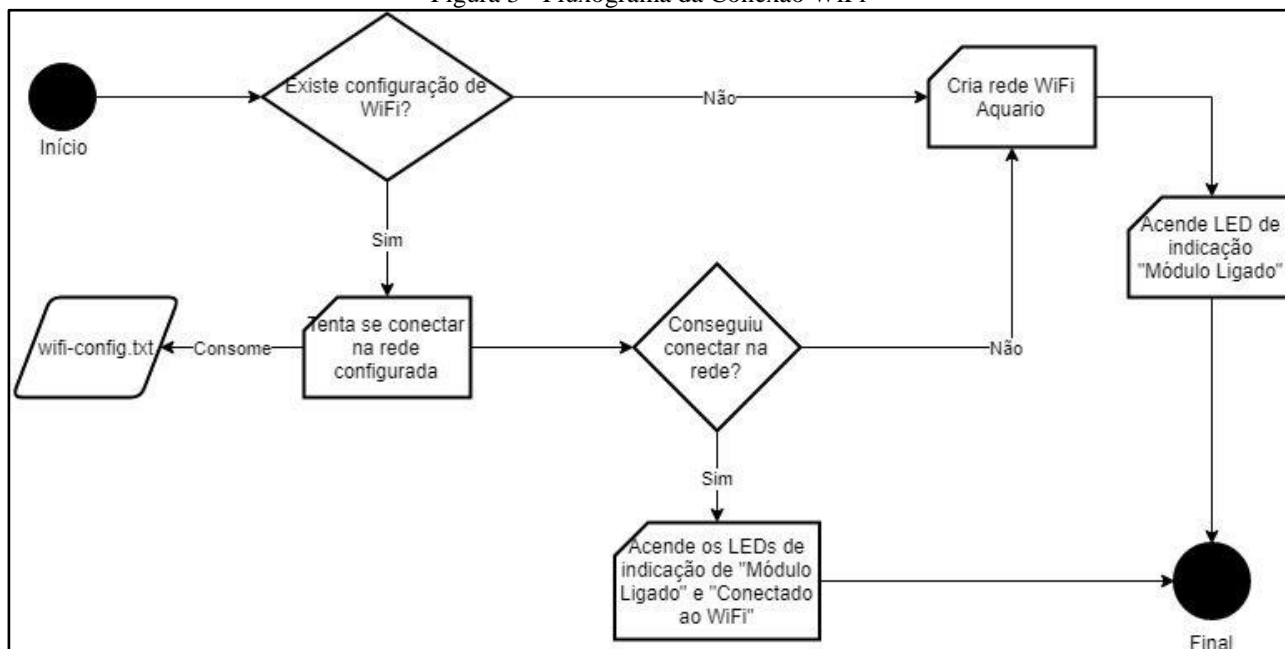
3.2 INTERFACE DE COMUNICAÇÃO

A interface de comunicação trata da transmissão de dados entre o módulo IUT e o Aquário Virtual, portanto é composta por implementação nas duas aplicações e pode ser dividida em três partes: conexão do módulo IUT em uma rede Wi-Fi; conexão entre módulo IUT e Aquário Virtual; e troca de dados com o simulador.

3.2.1 Conexão Wi-Fi

A conexão em uma rede Wi-Fi é feita de forma dinâmica por meio de uma configuração. Quando o módulo IUT é iniciado a classe `WiFiConnect` verifica a existência do arquivo `wifi-config.txt`. Este arquivo contém o nome da rede Wi-Fi (SSID) e senha. Caso o arquivo exista, é obtida a configuração e realizada a tentativa de conexão na rede Wi-Fi configurada. São feitas 3 tentativas de 10 segundos, se não for possível realizar a conexão ou o arquivo não existir, será iniciada uma nova rede Wi-Fi pelo módulo NodeMCU, denominada `Aquario` com a senha `aquario-virtual`, com propósito apenas de configuração. A Figura 3 demonstra como funciona a conexão Wi-Fi no módulo IUT.

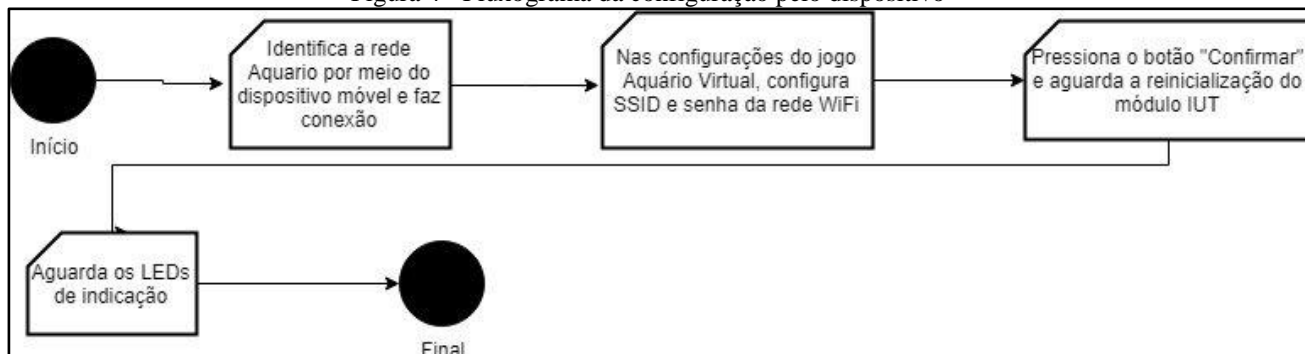
Figura 3 - Fluxograma da Conexão WiFi



Fonte: elaborado pelo autor.

Quando o módulo IUT conectar-se à rede Wi-Fi, o respectivo LED de indicação acenderá. Caso não seja possível realizar a conexão na rede, deverá por meio do dispositivo móvel com a aplicação do Aquário Virtual instalada, conectar-se à rede criada pelo módulo IUT (rede Aquário) e através da tela de configurações do simulador, configurar a conexão Wi-Fi. A Figura 4 apresenta um fluxograma da conexão caso ainda não esteja configurada.

Figura 4 - Fluxograma da configuração pelo dispositivo



Fonte: elaborado pelo autor

Após os LEDs de indicação de módulo ligado e conexão WiFi acenderem, indica que o dispositivo está pronto para conexão no simulador Aquário Virtual.

3.2.2 Conexão entre módulo IUT e Aquário Virtual

A conexão somente acontece após o dispositivo com Aquário Virtual e o módulo IUT conectarem-se a mesma rede Wi-Fi (desde que não seja a rede criada pelo próprio NodeMCU). Para a conexão é utilizado o protocolo Multicast, portanto a rede Wi-Fi deve ter suporte a IGMP – Internet Group Management Protocol.

A utilização do protocolo Multicast tem o objetivo de fazer com que as duas aplicações se identifiquem em uma rede Wi-Fi de forma dinâmica, sem a necessidade de utilizar endereço de IP fixo. O módulo IUT inscreve-se no IP Multicast 227.55.77.99, na porta 5000 e aguarda a mensagem de conexão. O Aquário Virtual envia uma mensagem no mesmo endereço Multicast, com a intenção de iniciar a conexão.

Para evitar que existam conflitos caso haja mais de um módulo IUT conectado na rede Wi-Fi, foi utilizada a estratégia de enviar um identificador na mensagem. Cada módulo IUT deverá ter seu identificador que fica fixado no código-fonte do programa e é disponibilizado junto ao módulo IUT. Este identificador deve ser inserido nas configurações do simulador.

Diferença com protocolo anterior

Justificar com parágrafo anterior

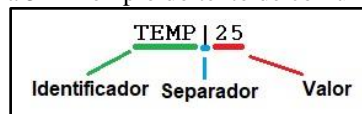
O módulo IUT, ao receber a mensagem Multicast retorna uma mensagem de confirmação ao simulador para que este consiga identificá-lo por meio de um endereço IP.

3.2.3 Comunicação e troca de dados

A comunicação e troca de dados se dá por meio de socket TCP. Após a inicialização, o módulo IUT inicia um servidor socket TCP no qual o Aquário Virtual fará conexão quando descobrir o endereço IP do módulo. Foi decidido implementar socket TCP por ser uma maneira simples de trocar mensagens e que garante a entrega dos pacotes.

Ao conectar-se como cliente no servidor socket TCP mantido no módulo IUT, o Aquário Virtual passa a receber por meio desta comunicação valores dos atuadores e sensores conectados ao módulo IUT. Os dados são transferidos em forma de texto dividido em partes: identificador da informação, caractere separador e valor. A Figura 5 demonstra um exemplo do texto.

Figura 5 - Exemplo do texto de comunicação



Fonte: elaborado pelo autor.

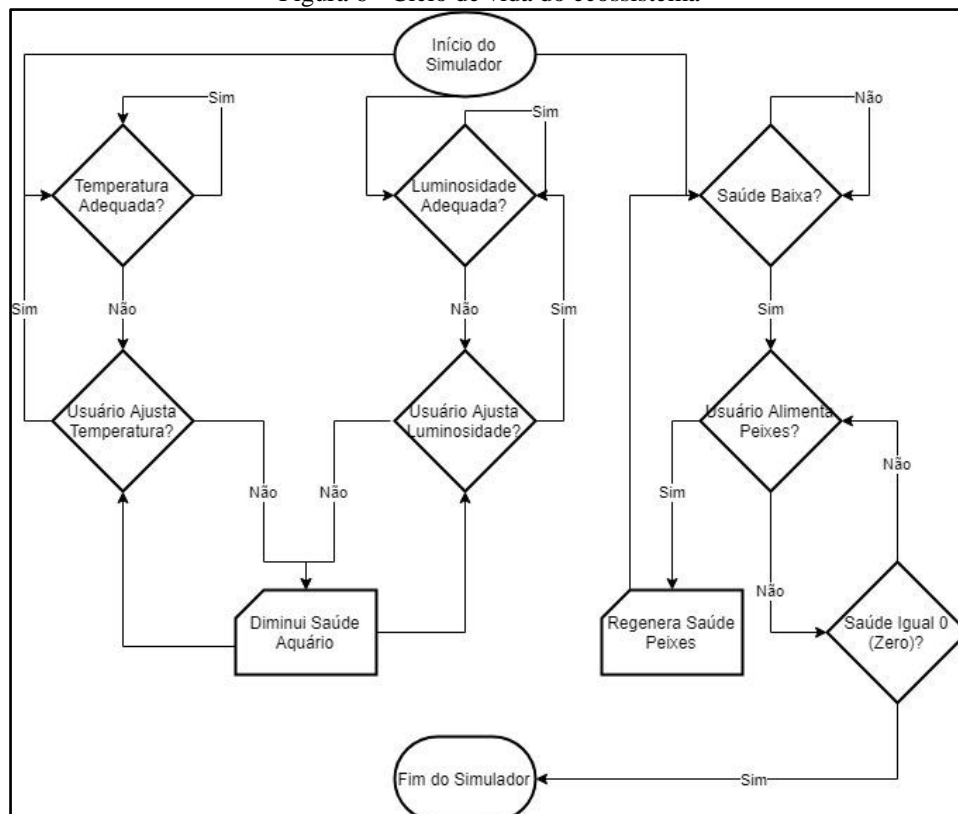
A estratégia de criar o servidor socket TCP no módulo IUT foi utilizada para evitar a criação de um servidor no Aquário Virtual, que já demanda processamento gráfico ao dispositivo.

3.3 AQUÁRIO VIRTUAL – SIMULADOR

O Aquário Virtual foi desenvolvido na plataforma Unity3D, utilizando a linguagem de programação C# para implementação e aprimoração de comportamentos. Para desenvolvimento do simulador foi utilizado o asset AIFishes (UNITY, 2019) como base, obtendo assim os peixes em 3D com alguns comportamentos padrão.

Algumas características de ecossistema foram implementadas para que o Aquário Virtual possua dinamismo. Os peixes no aquário têm pontos de saúde que vão de 0 a 100, iniciando em 100 para todos os peixes. Para reestabelecer os pontos de saúde os peixes precisam se alimentar, para isto foi desenvolvido um sistema de comida, onde uma porção de ração é despejada aos peixes de acordo com o acionamento do botão no módulo IUT. A Figura 6 apresenta o ciclo de vida do simulador com as interações do usuário.

Figura 6 - Ciclo de vida do ecossistema



Fonte: elaborado pelo autor.

O simulador inicia com o item Início do Simulador, onde posteriormente será verificado pelo usuário as etapas sobre Temperatura Adequada, Luminosidade Adequada e Saúde Baixa. Caso a luminosidade ou temperatura estejam diferentes do que é adequado, o usuário deverá ajustar estes valores por meio de manipulação do módulo IUT. Caso a saúde do aquário esteja baixa o usuário deve alimentar os peixes por meio do botão para esta finalidade presente no módulo IUT. O Quadro 5 apresenta os valores adequados para manter o ecossistema saudável.

Quadro 5 - Valores saudáveis para o ecossistema

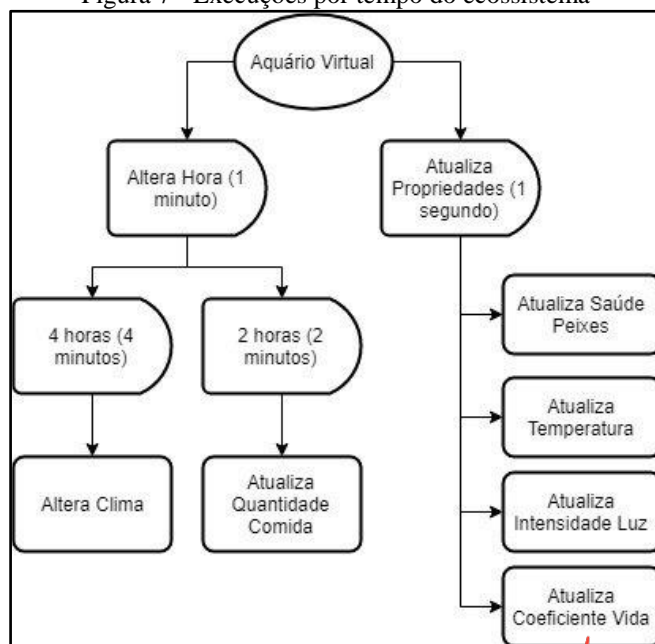
| PROPRIEDADE | VALOR ESPERADO |
|-----------------------|--|
| Temperatura Adequada | Entre 22,5°C e 25,5°C |
| Luminosidade Adequada | Entre 0% e 50% para noite. Entre 50% e 100% para dia. |
| Saúde Baixa | Abaixo de 70%. |

Fonte: elaborado pelo autor. *Aquecedor/resfriador*

A saúde é considerada baixa abaixo de 70% pois é quando os peixes estão aptos a se alimentar. A temperatura deve ser mantida por meio de equilíbrio com o *aquecedor/resfriador* controlado pelo módulo IUT. A luminosidade pode alterar de acordo com o clima ou por meio do módulo IUT, bem como a temperatura.

O aquário possui sistema de clima que influencia na temperatura e luminosidade do aquário. Existem cinco estados no aquário que podem alterar a temperatura externa e a luminosidade. Além disso, o aquário possui um tempo próprio, passando as horas de maneira mais rápida. A Figura 7 demonstra as execuções do ecossistema de acordo com o tempo.

Figura 7 - Execuções por tempo do ecossistema

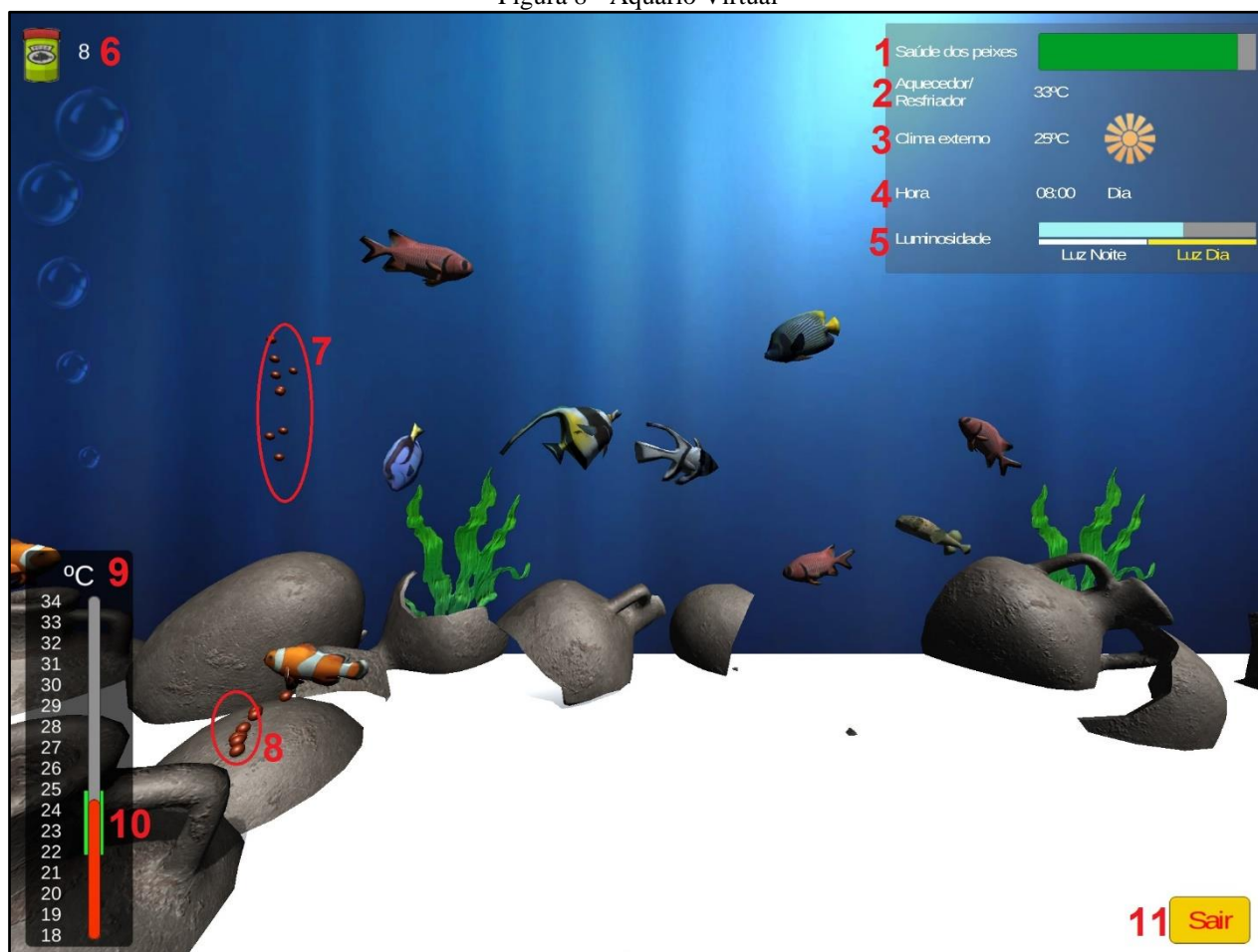


Fonte: elaborado pelo autor.

O item Aquário Virtual trata-se de todo o ecossistema do simulador. *N*o item Altera Hora *o* indica-se que a hora do aquário é alterada a cada minuto, que cria um efeito em cadeia conforme o tempo passa. Os itens Altera Clima e Atualiza Quantidade Comida são executados em função das horas passadas, o primeiro tem a função de modificar o clima do aquário enquanto o segundo recarrega a quantidade de porções de comida que podem ser utilizadas, incrementando em 1.

A execução do item Atualiza Propriedades faz a atualização de diversas informações do Aquário Virtual: o item Atualiza Saúde Peixes é responsável por atualizar a barra de saúde; Atualiza Temperatura faz a atualização da temperatura do aquário no termômetro; o item Atualiza Intensidade Luz modifica a intensidade de luz que é percebida no aquário; o item Atualiza Coeficiente Vida trata-se da rotina responsável por relacionar os itens de temperatura e luminosidade do aquário e calcular qual valor de vida será removido dos peixes nas próximas execuções. A Figura 8 apresenta o ecossistema do Aquário Virtual com os itens destacados e explicados no Quadro 6.

Figura 8 - Aquário Virtual



Fonte: arquivo pessoal.

Quadro 6 - Especificação do Aquário Virtual

| NÚMERO | PROPRIEDADE | ESPECIFICAÇÃO |
|--------|----------------------------|--|
| 1 | Saúde dos peixes | Apresenta a média da saúde de todos os peixes. Quando chega em 0, significa que todos os peixes morreram. |
| 2 | Aquecedor/Resfriador | Apresenta o valor em graus celsius da temperatura do aquecedor/resfriador, controlado pelo módulo IUT. |
| 3 | Clima externo | Apresenta a temperatura e o clima externo, explicado no Quadro 7. |
| 4 | Hora | Hora do Aquário Virtual. Cada hora equivale a 1 minuto do tempo real. |
| 5 | Luminosidade | Apresenta a luminosidade atual do Aquário Virtual, utilizando a luminosidade fornecida pelo Clima externo e a luminosidade do módulo IUT. As legendas Luz Noite e Luz Dia indicam como a luminosidade deve estar em cada um desses períodos. |
| 6 | Comidas | Exibe a quantidade de porções de ração que ainda está disponível para ser utilizada. |
| 7 | Porção de ração | Uma porção de ração caindo, ainda não habilitada para ser consumida. |
| 8 | Porção de ração habilitada | Após colidirem com o solo, as porções estão habilitadas para serem consumidas pelos peixes. |
| 9 | Termômetro | Indica a temperatura atual do aquário. |
| 10 | Temperatura saudável | Indica a faixa de temperatura ideal do Aquário Virtual. |
| 11 | Sair | Botão para finalizar o simulador. |

Fonte: elaborado pelo autor.

Os estados são alterados de tempo em tempo, sempre alternando para um estado aleatório, seguindo algumas regras: se a hora do aquário for maior que 05:00 e menor que 19:00, então os estados que remetem a noite não serão selecionados, caso a hora do aquário seja maior que 18:59 e menor que 05:01, os estados que remetem a dia não serão selecionados. O Quadro 7 apresenta os possíveis estados do aquário e algumas de suas especificações.

Quadro 7 - Especificação dos estados do aquário

| ÍCONE | ESTADO | TEMPERATURA MÍNIMA (°C) | TEMPERATURA MÁXIMA (°C) | LUMINOSIDADE (%) |
|-------|------------------|----------------------------|----------------------------|-------------------------------|
| | Dia/Sol | 25 | 43 | 25-50 |
| | Sol entre nuvens | 18 | 33 | 15-30 |
| | Neve | -2 | 5 | Noite: 0 – 15 Dia: 10 – 25 |
| | Chuva | 10 | 24 | Noite: 0 – 15 Dia: 10 – 25 |
| | Noite/Lua | 8 | 21 | 0 |

Fonte: elaborado pelo autor

Com estes elementos, o aquário possui um ciclo de vida dinâmico, fazendo com que a temperatura e luminosidade sejam ajustadas pelos comandos do módulo IUT, alterando a temperatura com o potenciômetro, que se comporta como um aquecedor e resfriador e o sensor LDR que se busca a luz externa para aplicar ao interior do simulador.

O asset AIFishes (UNITY, 2019) disponibiliza 3 comportamentos: Patrulhar ou Nadar, Ficar e Nadar para longe ou Fugir, além de disponibilizar algumas configurações como velocidade dos peixes, tamanho do campo de visão e quantidade de peixes. Os comportamentos ficam descritos na classe Fish, sendo que o comportamento inicial é Nadar, caso dois peixes colidam, um deles adotará o comportamento de Fugir e em momentos aleatórios é assumido o comportamento de Ficar.

Foram desenvolvidos dois novos comportamentos, Alimentar e Morrer. O comportamento Alimentar é ativado sempre que for identificado que há comida para os peixes. Este comportamento é ativado individualmente em cada peixe e somente em um peixe de cada vez. Seu objetivo é direcionar o peixe ao local onde está a comida. O comportamento Morrer é ativado quando a saúde do peixe chega a 0, fazendo com que o peixe se mova até o topo do aquário e desapareça.

4 RESULTADOS

O projeto desenvolvido neste artigo teve os resultados esperados alcançados, atingindo o objetivo de desenvolver um aquário virtual como simulador de ecossistema controlado por um módulo de interface de usuário tangível. Os testes do projeto foram realizados no decorrer do desenvolvimento, desde a conexão à rede WiFi até a comunicação entre o Aquário Virtual e o módulo IUT. Os resultados foram divididos em quatro seções, sendo a seção 4.1 para o módulo de Interface de Usuário Tangível, a seção 4.2 para a comunicação entre o módulo e o simulador e a seção 4.3 para o simulador Aquário Virtual. Além disso, a seção 4.4 trata dos testes realizados com alunos do curso de Ciência da Computação, da disciplina de Dispositivos Móveis.

4.1 MÓDULO INTERFACE DE USUÁRIO TANGÍVEL

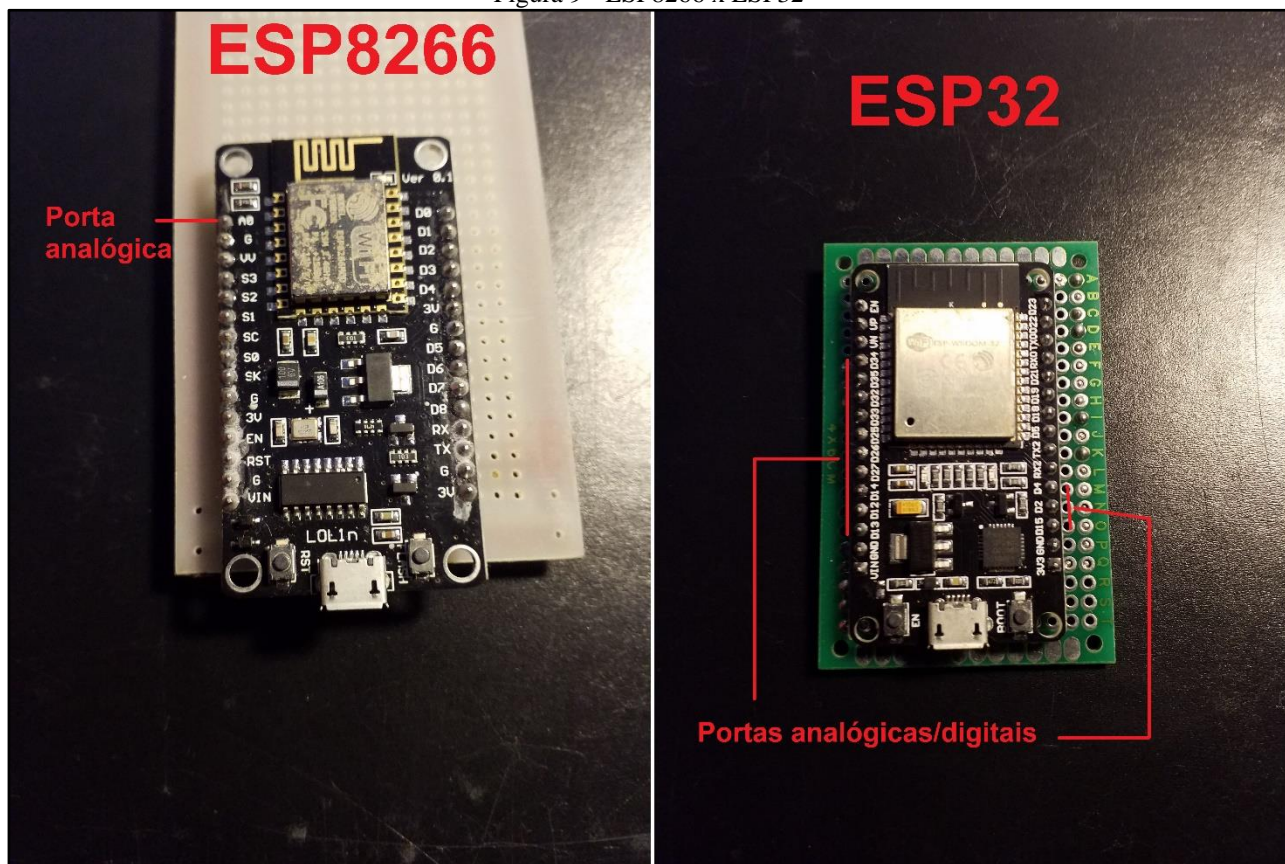
Inicialmente, para desenvolvimento do módulo IUT decidiu-se quais seriam as tecnologias utilizadas. Para tanto, definiu-se que seria utilizado o componente NodeMCU com o microcontrolador ESP8266, visando o baixo custo e a facilidade de manutenção, pois pode ser programado com as mesmas ferramentas utilizadas no Arduino.

Para construção do módulo, houve a necessidade de adquirir os demais componentes utilizados para serem sensores de atuadores do Aquário Virtual, portanto foram utilizados sensor de luz (LDR), potenciômetro, botão e LEDs. Para conexão destes componentes houve a necessidade de soldagem eletrônica em uma placa universal.

Uma das dificuldades de utilizar o módulo NodeMCU envolve a utilização de componentes com valores analógicos, que é o caso do potenciômetro e do sensor de luz. Estes componentes retornam ao módulo NodeMCU um valor variável de 0 até 1023, de acordo com a resistência que é incidida, seja por luz, no caso do sensor de luz ou por alteração de valor de resistência (potenciômetro). O NodeMCU disponibiliza apenas uma porta do tipo analógica para entrada de dados, sendo assim, as possibilidades seriam trocar de módulo ou adquirir um conversor analógico digital.

Desta forma, trocou-se para o microcontrolador ESP32, que dispõe de mais portas analógicas de entrada, mas houve a necessidade de adaptar parte do código-fonte já desenvolvido para funcionar com este novo microcontrolador. A Figura 9 apresenta os dois módulos, um com ESP8266 e outro com ESP32.

Figura 9 - ESP8266 x ESP32



Fonte: arquivo pessoal.

Após a alteração para utilizar o ESP32 como microcontrolador, encontrou-se a dificuldade na utilização das bibliotecas que fazem parte da comunicação entre o módulo IUT e o simulador Aquário Virtual. A biblioteca AsyncUDP utilizada na inicialização da conexão entre o módulo IUT e o simulador possuía um problema já conhecido, mas que ainda não estava corrigido. No Quadro 8 é apresentado o erro relatado.

Quadro 8 - Erro na biblioteca do ESP32

```
assertion "pbuf_free: p->ref > 0" failed: file
"/Users/ficeto/Desktop/ESP32/ESP32/esp-idf-
public/components/lwip/lwip/src/core/pbuf.c", line 765,
function: pbuf_free
abort() was called at PC 0x400d725b on core 1

Backtrace: 0x4008c7e0:0x3ffc8ce0 0x4008ca11:0x3ffc8d00
0x400d725b:0x3ffc8d20 0x400f0473:0x3ffc8d50
0x400d2bbf:0x3ffc8d70 0x400d2c0f:0x3ffc8e00
0x4008877d:0x3ffc8e30
```

Fonte: GitHub (2019b).

Após a identificação deste problema, foi feito um comentário na *issue* já aberta, porém, para continuação do desenvolvimento deste projeto, decidiu-se retornar ao microcontrolador ESP8266, utilizando um conversor analógico/digital. O conversor utilizado trata-se do MCP3008 (MICROCHIP, 2019), este componente ocupa 4 portas digitais de entrada e saída disponíveis no ESP8266 e disponibiliza 8 portas analógicas de entrada, permitindo que mais sensores ou atuadores analógicos fossem conectados.

Contudo, ao consumir 4 das portas de entrada e saída digitais, houve a necessidade de conectar um botão, que é um componente com valores digitais, em uma porta de entrada analógica, pois as portas digitais de saída foram ocupadas pelos LEDs de indicação. Desta forma, na implementação do código-fonte foi feito um ajuste para interpretar o momento

em que o botão fosse pressionado. Uma vez que as portas analógicas retornam valores entre 0 e 1023, quando um botão é pressionado, o valor lido pelo módulo é 1023 e não 1, como em uma porta digital.

Para os testes, após a soldagem eletrônica de todos os componentes na placa universal e conexão entre eles, foi utilizada leitura de valores impressos na porta serial por meio do módulo NodeMCU. Assim para cada valor lido de um componente, era impresso no console da IDE utilizada e comparado com a ação realizada.

4.2 CONEXÃO WIFI E COMUNICAÇÃO

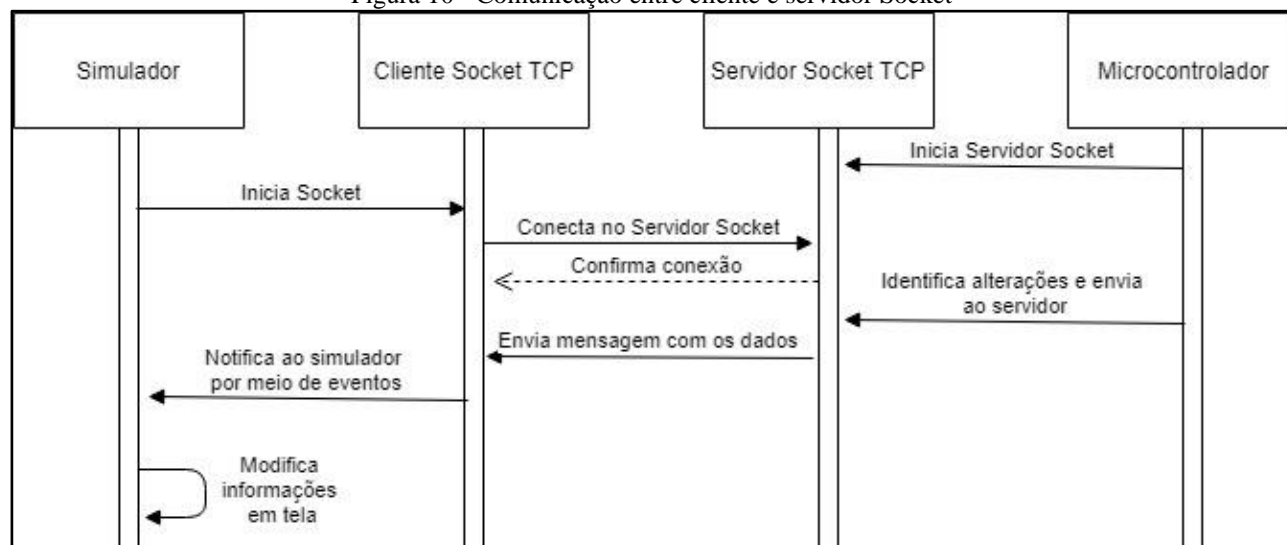
O desenvolvimento da conexão WiFi e comunicação entre módulo IUT e simulador foi independente do desenvolvimento do módulo IUT e componentes eletrônicos, pois ainda era possível trabalhar somente com o ESP8266 conectando-se a rede WiFi e enviando dados por meio deste. Também houve independência da conclusão do desenvolvimento do próprio simulador. Os primeiros testes de comunicação foram feitos utilizando um servidor em Node.js como receptor, uma vez que foram utilizadas técnicas como protocolo HTTP e Socket TCP.

Para desenvolvimento da rotina de conexão com uma rede WiFi, foi utilizada a linguagem C para Arduino, juntamente com os recursos de conexão que o próprio ESP8266 disponibiliza. No início existia a dificuldade de como o microcontrolador encontraria uma rede de forma dinâmica, sem a necessidade de manter apenas uma rede conectada. Este impasse foi resolvido utilizando o recurso de criação de Access Point disponibilizado pelo ESP8266. Este recurso permite que o próprio ESP8266 crie uma rede WiFi. A rede criada pelo microcontrolador foi denominada de *Aquario* com a senha padrão *aquario-virtual*. Esta rede é utilizada somente para configuração do módulo, com objetivo de conectá-lo em uma rede WiFi com acesso a internet. Desta forma, como já descrito no capítulo 3, o módulo verifica a existência de um arquivo de configuração, caso este arquivo não exista ou a rede configurada no arquivo seja inválida (fora de alcance, senha incorreta etc.), será criada a rede *Aquario* pelo microcontrolador.

Após a etapa de conexão com WiFi concluída, deu-se início para fase de comunicação entre o módulo e o simulador. Em um primeiro momento, o objetivo era utilizar protocolo HTTP para realizar tal comunicação. Para tal ideia, existiam duas opções: criar um servidor HTTP dentro do simulador ou tornar o módulo IUT um agente passivo, recebendo requisições HTTP de tempos em tempos e retornando ao simulador. A opção de tornar o módulo IUT passivo foi descartada pois perderia o aspecto de tempo real da aplicação, além da dificuldade de ler valores de botões por exemplo, que poderiam não estar pressionados no momento da requisição. A opção de criar um servidor HTTP no simulador chegou a ser considerada, porém, também foi descartada devido ao consumo de tempo e de recursos que o servidor HTTP causaria para o simulador.

A solução encontrada tratou-se de criar um servidor Socket TCP no microcontrolador, por ser simples de implementar, uma vez que utiliza dos mesmos recursos que o protocolo HTTP, sendo apenas uma camada abaixo. O objetivo foi trabalhar com a ideia de notificações. O simulador se conecta no servidor criado pelo microcontrolador, assim recebe todas as mensagens que são enviadas pelo servidor. Desta forma o simulador é apenas o cliente, portanto os recursos utilizados são baixos. O módulo IUT identifica sempre que um componente tem seu valor alterado e envia uma mensagem por meio do servidor Socket TCP. No simulador, o recebimento das mensagens funciona de forma assíncrona, não interrompendo a execução da aplicação para ler os valores. O diagrama da Figura 10 apresenta como funciona a comunicação.

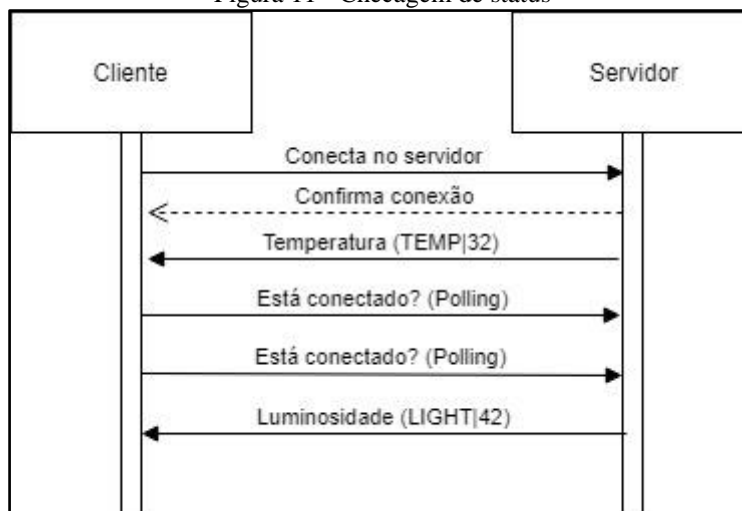
Figura 10 - Comunicação entre cliente e servidor Socket



Fonte: elaborado pelo autor.

Após os testes iniciais feitos com o servidor Node.js, foram realizados os testes diretamente com o simulador do Aquário Virtual. Durante os testes percebeu-se que após algum tempo de conexão entre o módulo IUT e o simulador a conexão era perdida. A alteração das informações por meio dos componentes eletrônicos era registrada pelo módulo IUT, mas não eram recebidas pelo Aquário Virtual. Para identificar o problema, foram incluídos registros (*logs*) tanto no módulo IUT como no simulador. Por meio destes registros foi identificado que após algum tempo as mensagens deixavam de ser recebidas pelo simulador, mas o módulo IUT ainda estava enviando os dados. Como forma de solução foi implementada uma estratégia de teste de conexão por meio de checagem de status. A Figura 11 demonstra como o processo ocorre.

Figura 11 - Checagem de status



Fonte: elaborado pelo autor.

A checagem de status ocorre a cada um segundo por meio do método Poll disponibilizado pela linguagem. Este método verifica o status da conexão entre o cliente e o servidor de socket, caso a resposta seja negativa, é feita a tentativa de uma nova conexão.

4.3 SIMULADOR

Para desenvolvimento do simulador do aquário, foi utilizado a ferramenta Unity3D em conjunto com o asset AIFishes (UNITY, 2019). O asset disponibiliza os peixes modelados em 3D com alguns comportamentos, além de alguns objetos 3D para construção do cenário. Inicialmente foi trabalhado com os peixes, implementando novos comportamentos para que atendessem ao simulador. O Quadro 9 apresenta os comportamentos disponibilizados pelo asset e quais foram implementados.

Quadro 9 - Comportamentos dos peixes

| COMPORTAMENTO | ORIGEM | ESPECIFICAÇÃO |
|------------------------|-------------------------|---|
| Patrulhar/Nadar | AIFishes | Peixe nada para um ponto aleatório do aquário. |
| Nadar para longe/Fugir | AIFishes | Quando dois peixes colidem, um deles nada mais rápido para outro ponto do aquário. |
| Ficar | AIFishes | Peixe fica parado. |
| Alimentar | Implementado pelo autor | Peixe é direcionado para o ponto onde a porção de comida está habilitada para ser consumida. Ocorre somente se tiver porção disponível. |
| Morrer | Implementado pelo autor | Quando a saúde chega a 0, o peixe sofre uma rotação e é movido ao topo do aquário. |

Fonte: elaborado pelo autor.

Além dos comportamentos e dos modelos 3D, o asset disponibiliza algumas funcionalidades muito úteis, como campo de visão dos peixes, que evita uma colisão com objetos ou outros peixes, velocidade de movimento e velocidade de rotação. Durante o desenvolvimento, foram encontradas dificuldades em controlar os peixes dentro do aquário. Para definição do aquário, os objetos 3D que representam os peixes são posicionados dentro de uma caixa com um objeto de colisão, sempre que os peixes identificam uma parede desta caixa, devem desviar para outro ponto.

Para visualização do aquário, foi mantida a câmera fixada em uma única posição, porém, com visualização 3D. Foram identificados problemas para que os peixes não saíssem do alcance da câmera, principalmente pelas laterais ou na parte superior. Foram feitos ajustes manuais no posicionamento da câmera e da área dos peixes. Outra dificuldade relacionada aos peixes foi ao implementar o comportamento *Alimentar*, onde os peixes se dirigem a um ponto específico

no cenário. O problema estava relacionado em todos os peixes colidindo ao tentarem alcançar o mesmo ponto. Como solução, foi implementado uma regra para somente um peixe por vez se dirigir ao ponto onde está a porção de ração. A movimentação ocorre de forma ordenada, considerando os peixes com saúde mais baixa primeiro.

Após o desenvolvimento dos comportamentos dos peixes e do cenário, iniciou-se a implementação do ecossistema de um aquário, tornando o simulador mais dinâmico. Foram desenvolvidas as propriedades de temperatura, luminosidade e saúde. Para que esse ecossistema funcione com essas propriedades, foram definidas algumas regras: o aquário tem 3 propriedades que o pertencem, essas propriedades impactam diretamente na saúde dos peixes. A primeira propriedade é a própria saúde dos peixes, essa informação é exibida em tela e trata-se da média da saúde de cada peixe individualmente, a segunda trata-se da temperatura do aquário, que é exibida em forma de termômetro, por fim, a luminosidade do aquário, que é exibida com uma barra de luminosidade e também por meio da intensidade de luz que pode ser percebida no aquário.

Essas três características são influenciadas por características externas, existe um fator que trata do clima externo, este clima externo possui uma temperatura e uma luminosidade que impactam nas propriedades do aquário, alterando-as. Portanto, é possível que a temperatura do aquário inicie em 25°C. porém, por influência do clima externo, seja gradativamente alterada para uma temperatura mais alta ou mais baixa. A luminosidade pode sofrer a mesma alteração, ser impactada pelo clima externo e aumentar ou diminuir a intensidade de luz do simulador. O objetivo do clima externo é simular o mundo real, alterações climáticas no ambiente onde o aquário se encontra que podem impactar no ecossistema interno do aquário.

Para o devido controle destas características, o módulo IUT faz o envio de informações de temperatura e luminosidade. Estes valores são calculados com os valores obtidos por meio do clima externo gerando um coeficiente para controlar a temperatura interna do aquário, para que esta propriedade seja modificada de forma gradativa ao longo do tempo. A luminosidade é dada em valor bruto, apenas somando o valor obtido do clima externo com o valor obtido do módulo IUT. Nesta etapa os testes foram realizados com objetivo de encontrar um valor equilibrado para este coeficiente, de forma a deixar o simulador com um ecossistema balanceado. O Quadro 10 demonstra a fórmula utilizada para calcular o coeficiente de temperatura.

Quadro 10 - Cálculo do coeficiente de temperatura

```
private void atualizaTemperatura() {  
    float diferencaAquecedor = tempAquecedor - tempAquario;  
    float diferencaExterno = tempExterno - tempAquario;  
    diferencaAquecedor = diferencaAquecedor / coeficienteTempo;  
    diferencaExterno = diferencaExterno / coeficienteTempo;  
    float coeficienteTemperatura = diferencaAquecedor + diferencaExterno;  
    tempAquario = tempAquario + coeficienteTemperatura;  
}
```

Fonte: elaborado pelo autor.

O valor da `diferencaAquecedor` armazena a diferença da temperatura do aquecedor com a temperatura do aquário e posteriormente é dividido pelo coeficiente de tempo, representado por `coeficienteTempo`. O valor de `coeficienteTempo` é obtido por um cálculo de multiplicação entre multiplicador de tempo do simulador – valor em segundos que representa 1 hora do simulador – e o valor 3, valor obtido após testes de balanceamento do simulador. O `diferencaExterno` passa pelo mesmo processo, apenas com a diferença de que obtém os valores do clima externo. Estes valores são somados para obter-se o valor do coeficiente de temperatura, representado por `coeficienteTemperatura`. Este coeficiente é somado à temperatura atual do aquário e o método `atualizaTemperatura` é executado a cada segundo, assim temos a atualização da temperatura do aquário.

Para que os valores de temperatura e luminosidade do ecossistema interno do aquário influenciem na saúde dos peixes, também foi implementado um cálculo para obtenção de um coeficiente de perda de vida. Para este cálculo são usados os valores de temperatura mínimo e máximo aceitáveis pelo aquário, bem como os valores mínimo e máximo de luminosidade. No caso dos valores mínimo e máximo de luminosidade, implementou-se uma regra para que existam valores mínimo e máximo para quando o período é noite e outro valor mínimo e máximo para quando o período é dia.

A saúde dos peixes é subtraída ao decorrer do tempo em que o simulador está ativo, mas caso as propriedades de temperatura e luminosidade não estejam dentro dos valores aceitáveis, então o valor subtraído é proporcionalmente maior. Ou seja, caso a temperatura do aquário esteja em 40°C enquanto o máximo aceitável é 25,5°C, a saúde dos peixes será reduzida mais rapidamente comparado a quando a temperatura do aquário está com o valor dentro do aceitável ou com um valor muito baixo acima, como por exemplo 27°C. O Quadro 11 demonstra o cálculo que é feito para se obter o coeficiente de perda de vida, que é atualizado a cada segundo.

está certo?

Quadro 11 - Cálculo do coeficiente de redução de saúde

```
private void atualizaCoeficienteReducaoSaude() {
    float coeficienteTemperatura = 0;
    float coeficienteLuminosidade = 0;
    float multiplicadorTemp = 0.3;
    float multiplicadorLuz = 0.2;
    if (tempAquario > tempMaxSuportada) {
        coeficienteTemperatura = (tempAquario - tempMaxSuportada) * multiplicadorTemp;
    } else if (tempAquario < tempMinSuportada) {
        coeficienteTemperatura = (tempMinSuportada - tempAquario) * multiplicadorTemp;
    }
    if (luzAquario > luzMaxSuportada) {
        coeficienteLuminosidade = (luzAquario - luzMaxSuportada) * multiplicadorLuz;
    } else if (luzAquario < luzMinSuportada) {
        coeficienteLuminosidade = (luzMinSuportada - luzAquario) * multiplicadorLuz;
    }
    coeficienteReducaoSaude = coeficienteTemperatura + coeficienteLuminosidade;
}
```

Fonte: elaborado pelo autor.

Os valores representados por `multiplicadorTemp` e `multiplicadorLuz` foram valores obtidos por meio de testes práticos do balanceamento do ecossistema do aquário, o objetivo é que a saúde seja reduzida gradativamente. Os demais valores que não são apresentados nestes blocos estão definidos em outro escopo, mas podem ser interpretados. Com o decorrer do tempo do Aquário Virtual, os peixes tem a saúde reduzida, porém podem recuperá-la por meio das porções de ração. O objetivo das porções de ração é que os peixes possam regenerar sua saúde e continuarem vivos no simulador.

4.4 TESTES COM ALUNOS DA DISCIPLINA DE DISPOSITIVOS MÓVEIS

Os testes foram realizados com alunos do curso de Ciência da Computação da FURB – Fundação Universidade Regional de Blumenau, da disciplina de Dispositivos Móveis. A aplicação dos testes ocorreu em aula, onde os alunos se dirigiram até a mesa e lhes eram explicados os objetivos, juntamente com a apresentação do projeto e aplicação de um questionário. O questionário foi criado por meio da ferramenta Google Forms e as perguntas criadas tem como objetivo analisar a usabilidade da aplicação bem como entender a experiência do usuário quanto ao que foi desenvolvido.

Para aplicação dos testes, o equipamento do módulo de IUT foi acoplado a uma caixa de madeira, juntamente com o tablet utilizado para executar o simulador, com objetivo de se assemelhar a um aquário. A mostra como ficou o resultado do equipamento.

Figura 12 - Aquário Virtual finalizado



Fonte: arquivo pessoal.

Os testes foram realizados com 9 alunos e foram realizadas algumas perguntas para entender o conhecimento prévio sobre simuladores e IUT – Interface de Usuário Tangível. O Quadro 12 apresenta o resultado destas perguntas.

Quadro 12 - Perfil dos alunos

| PERGUNTA | RESPOSTA |
|--|---|
| Nível de escolaridade | Ensino superior incompleto – 77,8% Ensino superior completo – 11,1% Ensino médio completo – 11,1% |
| Já utilizou algum simulador gráfico virtual? | Frequentemente – 22,2% Às vezes – 66,7% Nunca utilizei – 11,1% |
| Já ouviu falar de Interface de Usuário Tangível? | Sim – 55,6% Não – 44,4% |

Fonte: elaborado pelo autor.

Durante o questionário, eram apresentados passos para que fossem realizadas ações no simulador, com objetivo de entender as funcionalidades e ter a experiência de usuário. Posteriormente foram realizadas perguntas para entender qual foi o nível da experiência e como o aluno se sentiu ao utilizar o simulador. As respostas são valores que variam de 1 a 5, sendo 1 a resposta que define a experiência mais baixa e 5 um melhor nível de experiência. O Quadro 13 mostra as perguntas e respostas dos alunos referentes a utilização do simulador de ecossistema.

Quadro 13 - Experiência de usuário

| PERGUNTA | RESPOSTA |
|---|--|
| Você conseguiu concluir os objetivos dessa pesquisa com facilidade? | 100% responderam 5 |
| Quantas tarefas você concluiu sem nenhum auxílio externo? | 88,9% responderam 5 11,1% responderam 4 |
| Como você classifica a experiência de utilizar um equipamento de Interface de Usuário Tangível? | 77,8% responderam 5 22,2% responderam 4 |
| Como você classifica a usabilidade do Aquário Virtual? | 44,4% responderam 5 55,6% responderam 4 |

Fonte: elaborado pelo autor.

Por fim, foram obtidas as opiniões dos alunos sobre o projeto ter cumprido seu objetivo de simular um ecossistema utilizando Interface de Usuário Tangível, além de obter críticas e sugestões por meio de texto. O apresenta as respostas quanto ao objetivo do projeto.

Quadro 14 - Opinião dos alunos

| PERGUNTA | RESPOSTA |
|--|--------------------|
| Você acha que o Aquário Virtual cumpriu seu objetivo de desenvolver um simulador de ecossistemas utilizando Interface de Usuário Tangível? | 100% responderam 5 |

Fonte: elaborado pelo autor.

As sugestões e críticas obtidas por texto também foram registradas, inclusive houve avaliação sobre o equipamento final que envolve a caixa de madeira desenvolvida e a disposição dos equipamentos eletrônicos. Um dos comentários registrados está relacionado a danificação de um equipamento eletrônico. Após o desenvolvimento da caixa de madeira, foi levantada a ideia de cobrir o sensor de luz com algum tipo de proteção, porém, esta proteção poderia influenciar nos valores de intensidade de luz registrados pelo componente. Durante os testes, um aluno pressionou o sensor de luz por acreditar que era um botão, apesar de danificar a estrutura, os valores capturados pelo sensor continuaram funcionando normalmente. Algumas outras sugestões e críticas estão listadas no, mantendo o anonimato de cada aluno.

Quadro 15 - Críticas e sugestões

| |
|--|
| Seria interessante uma legenda indicando a barra de temperatura. A barra verde que indica a temperatura ideal não está totalmente claro. |
| Melhorar o menu inicial. |
| Muito bem implementado, único possível ponto de melhoria seria o aumento da complexidade do meio e das interações. |

Fonte: questionário elaborado pelo autor e respondido por alunos.

As críticas e sugestões foram avaliadas e consideradas como possíveis extensões para o projeto.

5 CONCLUSÕES

Os resultados esperados foram alcançados no desenvolvimento deste projeto, houve o desenvolvimento do Aquário Virtual como um simulador de ecossistema que demonstrou funcionar bem de uma forma dinâmica. As classes utilizadas para comunicação entre as aplicações podem ser reaproveitadas em outros projetos, pois foram desenvolvidas

justificar os parágrafos

com objetivo de construção de uma biblioteca, tanto para o microcontrolador ESP8266 quanto para a ferramenta Unity3D. Os sensores e atuadores utilizados se mostraram eficientes para o fim esperado, o sensor de luz e o potenciômetro atenderam bem para alteração das características do ecossistema. O ecossistema do Aquário Virtual se mostrou dinâmico, sofrendo as alterações ao longo do tempo e instigando o pensamento para o ajuste das propriedades. A utilização de Interface de Usuário Tangível com componentes eletrônicos foi interessante, onde as alterações feitas pelo módulo IUT acontecem rapidamente no ecossistema e traz a sensação de imersão no aquário, visto que os componentes são posicionados estrategicamente para este fim.

A ferramenta Unity3D se mostrou eficiente no desenvolvimento de jogos ou simuladores 3D e de fácil aprendizado, uma vez que boa parte das funcionalidades utilizadas foram aprendidas no decorrer do desenvolvimento do projeto. A forma de comunicação escolhida para conectar o módulo IUT e o dispositivo móvel também foi eficiente, ambos continuam com acesso a rede WiFi e não houve grandes dificuldades para conectá-los a rede WiFi. Para desenvolvimento do código-fonte executado no microcontrolador, foi utilizado o editor Visual Studio Code (CODE, 2019), a qual se mostrou uma ferramenta muito poderosa quando utilizada em conjunto com o plugin PlatformIO (PLATFORMIO, 2019), permitindo a transmissão do programa ao microcontrolador diretamente pela ferramenta, além de funcionalidades como sugestão de métodos para utilização, entre outros.

Apesar do desenvolvimento do projeto ter ocorrido com sucesso, existem extensões que podem ser aplicadas para tornar o ecossistema ainda mais dinâmico e interativo. Algumas delas são:

- Incluir mais atuadores e sensores no módulo de Interface de Usuário Tangível, como um sensor de temperatura para obter a temperatura externa e sensor de proximidade, para alterar o comportamento dos peixes.
- Implementação de inteligência artificial para o comportamento dos peixes, baseando-se no trabalho de Piske (2015), poderia incluir animação comportamental para controle dos peixes.
- Trabalhar com os sensores do próprio dispositivo móvel, como a funcionalidade de toque na tela, e temperatura do dispositivo.
- Trabalhar o ecossistema incluindo novas variações, como poluição, por exemplo.
- Entender as origens de cada espécie de peixe para trabalhar as propriedades de cada um de forma individual, como temperatura suportada e luminosidade.
- Desenvolver elementos de morte dos peixes por tempo e geração de novos peixes, por reprodução.
- Utilizar o recurso de som do dispositivo móvel, trabalhando mais ainda a imersão no simulador.
- Trabalhar na experiência de usuário quanto a indicação das informações que estão em tela, incluindo o menu inicial.

REFERÊNCIAS

ALVES, Marcia Maria; BATTAIOLA, André Luiz. Recomendações para ampliar motivação em jogos e animações educacionais. In: SIMPÓSIO BRASILEIRO DE GAMES, 10., 2011, Salvador. **SBC - Proceedings of SBGames**. Salvador: Sbc, 2011. p. 1 - 5.

CARBAJAL, Marleny Luque; BARANAUSKAS, M. Cecília C.. TaPrEC: Desenvolvendo um ambiente de programação tangível de baixo custo para crianças. In: TISE, 11., 2015, Santiago. **Nuevas Ideas en Informática Educativa**. Santiago: Congreso Internacional de Informática Educativa, 2015. p. 363 - 370.

CIRCUITS4YOU. **NodeMCU pinout**. 2017. Disponível em: <<https://circuits4you.com/2017/12/31/nodemcu-pinout/>>. Acesso em: 02 jul. 2019.

CODE. **Visual Studio Code**. 2019. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 30 jun. 2019.

CURVELLO, André. **Apresentando o módulo ESP8266**. 2015. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 02 jul. 2019.

FALCÃO, Taciana Pontual; GOMES, Alex Sandro. Interfaces Tangíveis para a Educação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 18., 2007, Recife. **Anais do SBIE 2007**. Rio de Janeiro: Mackenzie, 2007. p. 579 - 589.

FERNANDES, Larissa Krüger. **Infância urbana e novas tecnologias: uma análise pela perspectiva da criança**. 2018. xix, 142 f., il. Dissertação (Mestrado em Processos de Desenvolvimento Humano e Saúde)-Universidade de Brasília, Brasília, 2018.

FILIPEFLOP. **Sensor de Luminosidade LDR**. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-luminosidade-ldr-5mm/>>. Acesso em: 09 jun. 2019.

GITHUB. **GitHub**. 2019a. Disponível em: <<https://github.com/>>. Acesso em: 30 jun. 2019.

GITHUB. **GitHub**. 2019b. Disponível em: <<https://github.com/espressif/arduino-esp32/issues/2685/>>. Acesso em: 30 jun. 2019.

HSIAO, Hui-chun. A Brief Review of Digital Games and Learning. In: FIRST IEEE INTERNATIONAL WORKSHOP ON DIGITAL GAME AND INTELLIGENT TOY ENHANCED LEARNING, 1., 2007, Jhongli. **2007 First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL'07)**. Jhongli: Ieee, 2007. p. 57 - 64.

MATTEI, Claudinéia. **O prazer de aprender com a informática na educação infantil**. 2013. 15 f. Monografia (Especialização), Instituto Catarinense de Pós-graduação, Indaial, 2013.

MERCADOLIVRE. **MercadoLivre**. 2019. Disponível em: <<https://www.mercadolivre.com.br/>>. Acesso em: 23 jun. 2019.

MICROCHIP. **Microchip**. 2019. Disponível em: <<https://www.microchip.com/wwwproducts/en/MCP3008>>. Acesso em: 30 jun. 2019.

NINTENDO. **Nintendo Labo**. 2019. Disponível em: <<https://www.nintendo.pt/Nintendo-Labo/Nintendo-Labo-1328637.html>>. Acesso em: 22 jun. 2019.

NUNES, Augusto L. P.; RADICCHI, Adriel O.; BOTEGA, Leonardo C.. Interfaces Tangíveis: Conceitos, Arquiteturas, Ferramentas e Aplicações. In: SIMPÓSIO DE REALIDADE VIRTUAL E AUMENTADA, 8., 2011, Uberlândia. **Livro do Pré-Simpósio**. Uberlândia: Sbc – Sociedade Brasileira de Computação, 2011. p. 26 - 44.

PISKE, Kevin Eduardo. **Aquário virtual**: simulador de ecossistema utilizando animação comportamental. 2015. 113 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Fundação Universidade Regional de Blumenau - Furb, Blumenau, 2015.

PLATFORMIO. **PlatformIO**. 2019. Disponível em: <<https://platformio.org/>>. Acesso em: 30 jun. 2019.

PROESI (Blumenau). **Proesi**. 2019. Disponível em: <<https://proesi.com.br/>>. Acesso em: 09 jun. 2019.

REVELLE, Glenda et al. Tangible user interfaces for children. Extended Abstracts On Human Factors In Computing Systems. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 5., 2005, Portland. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. Portland: Association For Computing Machinery, 2005. p. 2051 - 2052.

SANTOS, José Carlos Fernandes dos. **Resistores e Leis de Ohm**. 2019. Disponível em: <<http://educacao.globo.com/fisica/assunto/eletrmagnetismo/resistores-e-leis-de-ohm.html>>. Acesso em: 02 jul. 2019.

SAVI, Rafael; ULBRICHT, Vania Ribas. Jogos digitais educacionais: benefícios e desafios. **Novas Tecnologias na Educação**, [s. l.], v. 6, n. 2, p.1-10, dez. 2008.

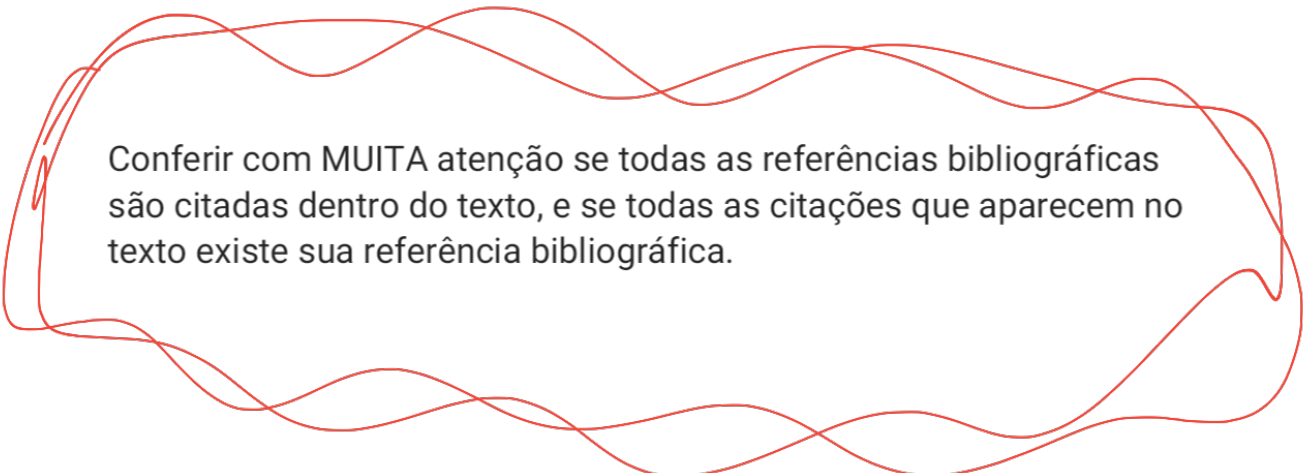
SILVA, Alessandro Antunes; PASSERINO, Liliana Maria. A Fazenda: Software Educativo para a Educação Ambiental. **Novas Tecnologias na Educação**, [s. l.], v. 5, n. 2, p.1-12, dez. 2007.

SYSTEMS. **Espressif Systems**. 2019. Disponível em: <<https://www.espressif.com/>>. Acesso em: 02 jul. 2019.

UNITY. **Unity AssetStore**. 2019. Disponível em: <<https://assetstore.unity.com/packages/3d/characters/animals/ai-fishes-ready-solutions-for-your-project-124645>>. Acesso em: 30 jun. 2019.

YUAN, Michael. **Conhecendo o NodeMCU e sua placa DEVKIT**. 2018. Disponível em: <<https://www.ibm.com/developerworks/br/library/iot-nodemcu-open-why-use/index.html>>. Acesso em: 02 jul. 2019.

ZUCKERMAN, Oren; ARIDA, Saeed; RESNICK, Mitchel. Extending tangible interfaces for education: digital montessori-inspired manipulatives. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 5., 2005, Portland. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. Portland: Association For Computing Machinery, 2005. p. 859 - 868.



Conferir com MUITA atenção se todas as referências bibliográficas são citadas dentro do texto, e se todas as citações que aparecem no texto existe sua referência bibliográfica.