

DQI: uma aplicação web para detecção de queda de idoso

Divana Aparecida Josaphat do Espírito Santo¹, Daniela Marques¹, Leandro C. Ledel¹

¹Instituto Federal de São Paulo - Câmpus Hortolândia

divanajosaphat@gmail.com, {marquesdaniela,ledel}@ifsp.edu.br

Abstract. *The fall of the elderly is a constant concern for family members, for this reason Silva et al. (2019) created a fall detection mechanism with low-cost devices. However, the detector has a user interface. The data necessary to associate the device with the elderly person and the responsible person's data are hard coded. This work proposes the creation of a web application to complement the work developed by Silva et al.. The application will collect the information of the responsible and the elderly and will allow the detector to access the information on the database.*

Resumo. *A queda do idoso é uma preocupação constante para os familiares, por este motivo Silva et al. (2019) criou um detector de queda com dispositivos de baixo custo. Contudo, o sistema deste detector não possui nenhuma interface de usuário. Os dados necessários para associar o dispositivo ao idoso e os dados do responsável são feitos no código. Este trabalho propõe a criação de um aplicativo web para complementar o trabalho desenvolvido. Por meio da aplicação são fornecidas as informações do responsável e idoso e possibilitará que o detector de queda acesse as informações no banco de dados criado.*

1. Introdução

A população idosa no mundo inteiro, de acordo com dados da Organização Mundial de Saúde “OMS” com mas de 60 anos, passará de 2 Bilhões em 2050 e, de acordo ainda com estudos da OMS, em 2020 a população de idosos com mas de 60 anos será maior do que de crianças com até 5 anos (OMS, 2005).

De acordo com a OMS (2005) “em todo mundo há tendência maior de que os idoso vivam sozinhos, especialmente as mulheres mais velhas, a queda de idosos é uma causa crescente de lesões e mortes”. Contudo, somente 5% das quedas resultam em fraturas ou hospitalizações (Oliveira, 2017). Gomes (2014, p. 3544) em seu artigo define que:

A queda é conceituada como um deslocamento não intencional do corpo para um nível inferior à posição inicial, sem correção de tempo hábil e é determinada por circunstâncias multifatoriais que comprometem a estabilidade, ou seja, mecanismos envolvidos com a manutenção da postura.

As causas provocadas por esse evento podem ser agrupadas em fatores intrínsecos, aqueles relacionados com as alterações fisiológicas decorrentes do processo de envelhecimento, como condições patológicas e consumo de medicamentos, e os fatores extrínsecos, que estão ligados aos perigos ambientais, devido às inadequações arquitetônicas e de mobiliário, que a maioria dos idosos está exposta.

Baseado neste cenário, surgiu a ideia de se fazer um sistema de detecção de queda para idoso utilizando dispositivos de baixo custo. A solução proposta foi dividida em duas partes: (i) o desenvolvimento do dispositivo de detecção de queda e (ii) uma aplicação web para gerenciamento dos dados a serem utilizados pelo dispositivo. As soluções foram desenvolvidas por duas analistas, ambas alunas do curso tecnólogo em Análise e Desenvolvimento de Sistemas.

A primeira parte da solução foi feita por Silva (2019) cujo trabalho foi desenvolver um dispositivo de *hardware* e *software* para detectar a queda de idosos e enviar um sinal de alerta remoto a pessoas previamente cadastradas. A Figura 1 mostra o simulador construído para detecção de queda (Figura 1a) e o foco no acelerômetro utilizado na solução (Figura 1b).

Silva (2019 p.13) indica a utilização de hardware de baixo custo para monitoramento de queda e para o envio da notificação da queda pelo e-mail cadastrado:

O nó sensor detecta a queda e realiza a chamada de um arquivo no servidor web php por meio do protocolo HTTP POST. Nesse arquivo tem um comando que chama o Mail Server através do Simple Mail Transfer Protocol (SMTP), que após validar o endereço e a senha da conta de e-mail envia a mensagem para o endereço eletrônico do responsável.

As configurações necessárias para envio de mensagens são feitas diretamente no código.

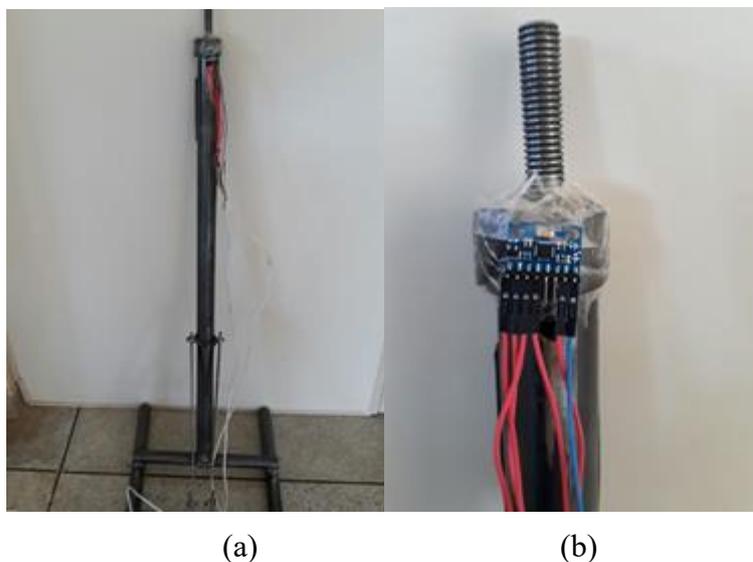


Figura 1. Mecanismo de Simulação de Quedas
[Fonte: (Silva, 2019)]

A segunda parte do trabalho consiste no desenvolvimento de uma aplicação web para facilidade de acesso às informações usadas pelo dispositivo. A crescente utilização da internet e a constante necessidade de serviços rápidos e práticos fez com que cada vez mais usuários recorram a aplicações ou dispositivos que possam melhorar seu dia a dia. Assim, este trabalho desenvolveu uma aplicação web simples do ponto de vista dos usuários e que facilite a sua navegação para realização dos cadastros do responsável e do idoso.

Este trabalho está organizado da seguinte maneira: na Seção 1 foi descrita a Introdução, o Referencial Teórico é mostrado na Seção 2, a Seção 3 mostra as ferramentas utilizadas, a Metodologia é apontada na Seção 4, a Seção 5 mostra o desenvolvimento do trabalho e, por fim, a Conclusão é feita na Seção 6.

2. Referencial Teórico

As subseções a seguir apresentam conceitos que foram utilizados no decorrer deste trabalho.

2.1. Desenvolvimento web

Nos primórdios da *World Wide Web* (por volta de 1990 a 1995), os sites eram formados por nada mais que um conjunto de arquivos de hipertexto linkados que apresentavam informações usando texto e gráficos limitados. Com o tempo, o aumento da HTML (*HyperText Markup Language*), via ferramentas de desenvolvimento (por exemplo, XML, Java), tornou possível

aos engenheiros da Internet oferecerem capacidade computacional juntamente com as informações. Nasceram, então, os sistemas e aplicações baseados na web (Pressman, 2010).

Pelo fato de software, como todo capital, ser conhecimento incorporado, e pelo fato de esse conhecimento ser, inicialmente, disperso, tácito, latente e em considerável medida, incompleto, o desenvolvimento de software é um processo de aprendizado social.

Esse processo é um diálogo no qual o conhecimento, que deverá tornar-se o software, é coletado, reunido e incorporado ao software. Tal processo possibilita a interação entre usuários e projetistas, entre usuários e ferramentas em evolução e entre projetistas e ferramentas em evolução (tecnologia). Trata-se de um processo iterativo no qual a própria ferramenta em evolução serve como meio de comunicação, com cada nova rodada do diálogo extraindo mais conhecimento útil das pessoas envolvidas (Pressman, 2010).

2.1.1 Aplicação web

Uma aplicação web pode conter uma coleção de páginas, porém o conteúdo destas páginas é montado dinamicamente, ou seja, é carregado através de solicitações (requisições) à um banco de dados, que contera armazenado os textos e indicação dos caminhos das imagens ou mídias que a página precisa exibir. Porém um arquivo escrito em HTML não tem acesso direto a um banco de dados, e esta comunicação deve ser feita por uma linguagem de programação de servidor web. Esta aplicação escrita com uma linguagem de servidor que tem o poder de acessar o banco de dados e montar a página HTML conforme o solicitado pelo navegador (Caelum, 2021).

Estas solicitações podem ser feitas de várias maneiras, inclusive utilizando JavaScript. Portanto uma aplicação Web é mais complexa porque precisa de uma linguagem de servidor para poder intermediar as solicitações do navegador, um banco de dados, e muitas vezes (porém não obrigatoriamente) exibir páginas HTML com estes conteúdos (Caelum, 2021).

2.2. Requisitos funcionais

Requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer. Uma forma de documentar os requisitos funcionais do sistema é utilizando os diagramas de Caso de Uso da UML (*Unified Modeling Language*), onde cada funcionalidade é representada pelo caso de uso e os atores representam papéis desempenhados por usuários externos ao sistema (Sommerville, 2011).

Os requisitos funcionais de um sistema descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos. Quando expressos como requisitos de usuário, os requisitos funcionais são normalmente descritos de forma abstrata para serem compreendidos pelos usuários do sistema. No entanto, requisitos de sistema funcionais mais específicos descrevem em detalhes as funções do sistema, suas entradas e saídas, exceções etc (Sommerville, 2011).

2.3 Requisitos não funcionais

Os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área. Uma alternativa a esse cenário seria os requisitos que definem

restrições sobre a implementação do sistema, como as capacidades dos dispositivos de E/S ou as representações de dados usadas nas interfaces com outros sistemas.

Os requisitos não funcionais, como desempenho, proteção ou disponibilidade, normalmente especificam ou restringem as características do sistema como um todo. Requisitos não funcionais são frequentemente mais críticos que requisitos funcionais individuais. Os usuários do sistema podem, geralmente, encontrar maneiras de contornar uma função do sistema que realmente não atenda a suas necessidades. No entanto, deixar de atender a um requisito não funcional pode significar a inutilização de todo o sistema (Sommerville, 2011).

2.4 Internet das coisas

A Internet das Coisas (do inglês *Internet of Things* (IoT)) emergiu dos avanços de várias áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. De fato, a IoT tem recebido bastante atenção tanto da academia quanto da indústria, devido ao seu potencial de uso nas mais diversas áreas das atividades humanas (Santos, 2017).

A Internet das Coisas, em poucas palavras, nada mais é que uma extensão da Internet atual, que proporciona aos objetos do dia-a-dia (quaisquer que sejam), mas com capacidade computacional e de comunicação, se conectarem à Internet. A conexão com a rede mundial de computadores viabiliza primeiro controlar remotamente os objetos e, segundo, permitir que os próprios objetos sejam acessados como provedores de serviços (Santos, 2017).

Estas novas habilidades, dos objetos comuns geram um grande número de oportunidades tanto no âmbito acadêmico quanto no industrial. A Figura 2 mostra uma arquitetura básica a ser utilizada em IoT com 3 camadas: (i) de percepção é a camada de objetos inteligentes que representam os sensores; (ii) de rede representa a camadas com as abstrações das tecnologias de comunicação, e, (iii) de aplicação que fornece os serviços para os clientes (Santos, 2017).

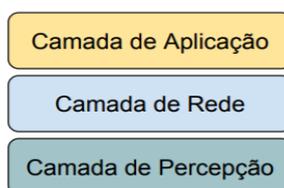


Figura 2 Arquitetura para IoT

[Fonte: (Santos, 2017)]

3. Ferramentas

Para o desenvolvimento deste trabalho utilizamos ferramentas *Open Source* tais quais: banco de dados MySQL, as ferramentas HTML por ser a linguagem de marcação mas utilizada em aplicações web, o PHP, o Bootstrap por ser um *framework* para *front-end*, e por fim o JavaScript.

3.1 HTML

O HTML é o bloco de construção mais básico da web. Ele define o significado e a estrutura do conteúdo da web. HTML não é uma linguagem de programação; é uma ‘linguagem de marcação’ que define a estrutura do seu conteúdo. O HTML consiste em uma série de elementos, que você usa para incluir ou agrupar diferentes partes do conteúdo para que ele apareça de uma certa maneira ou aja de certa maneira. As tags anexas podem criar um hiperlink de palavra ou imagem para outro lugar, colocar itálico nas palavras, aumentar ou diminuir a fonte e assim por diante.

Em sua essência, HTML é uma linguagem bastante simples composta de elementos, que podem ser aplicados a pedaços de texto para dar-lhes significado diferente em um documento e incorporar conteúdo como imagens e vídeos em uma página. Uma das principais funções do HTML é dar significado ao texto (também conhecido como semântica), para que o navegador saiba como exibi-lo corretamente.

3.2 CSS

O CSS (*Cascading Style Sheets*) é um dos principais idiomas da web aberta e é padronizado nos navegadores da web. Como o HTML, o CSS não é realmente uma linguagem de programação. CSS é uma 'linguagem de folha de estilos'. Isso significa que permite aplicar estilos seletivamente a elementos em documentos HTML. CSS é usada para estilizar e arranjar páginas web, por exemplo, para alterar a fonte, cor, tamanho e espaçamento do seu conteúdo, separá-lo em múltiplas colunas, ou então adicionar animações e outras implementações (Silva, 2015).

Sendo assim em cada nova execução na estrutura do HTML foram adicionados os estilos inicialmente no *body* que foi a escolha da fonte, tamanho e cores depois a cada nova adição de código se trabalhou o CSS.

3.3. PHP

PHP é uma linguagem que permite criar sites web dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e *links*. A diferença de PHP com relação a linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas o HTML puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente.

Isso pode ser útil quando o programa está lidando com senhas ou qualquer tipo de informação confidencial. PHP também tem como uma das características mais importantes o suporte a bancos de dados, como dBase, Interbase, mSQL, MySQL, Oracle, Sybase, PostgreSQL e vários outros. (Barreto, 2000).

3.4. MySql

O MySQL é um sistema de gerenciamento de bancos de dados relacional. Um banco de dados relacional armazena dados em tabelas separadas em vez de colocar todos os dados num só local. MySQL, o mais popular sistema de gerenciamento de banco de dados *SQL Open Source*, é desenvolvido, distribuído e tem suporte. Um banco de dados é uma coleção de dados estruturados.

Para adicionar, acessar, e processar dados armazenados em um banco de dados de um computador, você necessita de um sistema de gerenciamento de bancos de dados como o Servidor MySQL. Como os computadores são muito bons em lidar com grandes quantidades de dados, o gerenciamento de bancos de dados funciona como a engrenagem central na computação, seja como utilitários independentes ou como partes de outras aplicações (Carvalho, 2015).

3.5. Javascript

JavaScript (frequentemente abreviado como JS) é uma linguagem de programação leve, interpretada e orientada a objetos com funções de primeira classe, conhecida como a linguagem

de *scripting* para páginas web, mas também utilizada em muitos ambientes fora dos navegadores. Ela é uma linguagem de *scripting* baseada em protótipos, multi-paradigma e dinâmica, suportando os estilos orientado a objetos, imperativo e funcional (MDM, 2021).

3.6 Bootstrap

Bootstrap é um poderoso *framework front-end* voltado para o desenvolvimento de sites e aplicações web responsivos e alinhados com a filosofia *mobile-first*. É indicado para dispositivos e projetos de todos os tamanhos e destinado a desenvolvedores com a utilização de JavaScript, HTML e CSS para desenvolvimento de sites e aplicações web responsivas e alinhadas com a filosofia *mobile first*. Torna o desenvolvimento *front-end* muito mais rápido. (Silva, 2015).

4. Metodologia

Este trabalho dá continuidade ao desenvolvimento feito por Silva (2019), onde foi criado o monitor e detector de quedas por meio do desenvolvimento do hardware de baixo custo. O primeiro passo deste trabalho foi o entendimento do funcionamento do monitor e detector de queda para identificar quais informações seriam necessárias.

Posteriormente, os casos de uso foram elaborados para delimitar o escopo do sistema. Com base no diagrama de caso de uso e nas informações necessárias para o detector de queda, foi então criado o banco de dados.

A abordagem incremental foi utilizada na implementação, onde cada incremento referenciava um caso de uso. O sistema se tornou completo quando os testes foram finalizados com sucesso.

Por se tratar de uma aplicação web foram feitas diversas iterações para melhorar o *front-end* da aplicação.

5. Desenvolvimento do Trabalho

O simulador de detecção de queda é um dispositivo acoplado ao idoso para detectar uma possível queda, uma tendência considerando o conceito de IoT. Uma vez identificado um evento de queda, um e-mail é enviado ao responsável pelo idoso. Atualmente as informações do e-mail do responsável são mapeadas diretamente no código, não mostrando ser amigável a uma pessoa leiga.

A Figura 3 apresenta a arquitetura para IoT adaptada ao contexto deste trabalho. Silva et al. (2019) focou nas camadas de percepção (com NodeMCU e acelerômetro) e na camada de rede (onde foi feita a comunicação e leitura dos dados dos dispositivos e enviado um e-mail para o responsável pelo idoso. Esse trabalho é focado na camada da aplicação, onde será oferecido o serviço dos cadastros de responsável e idoso e na camada de rede, onde estará disponível o servidor do banco de dados.

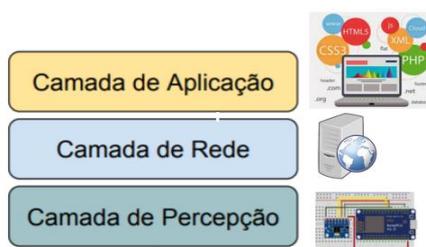


Figura 3. Arquitetura IoT adaptada

No início deste trabalho foi necessário fazer uma pesquisa de quais ferramentas poderiam ser utilizadas para tornar a aplicação web uma realidade. Neste estudo procurou-se entender como funciona e o que compõe o desenvolvimento web para daí escolher qual linguagem utilizar. Este é um trabalho meramente acadêmico e não uma aplicação com robustez e partindo desses estudos decidiu-se usar o HTML que é a principal base para para qualquer aplicação web. As subseções a seguir mostram todas as etapas do ciclo de desenvolvimento de software realizado.

As subseções a seguir mostram todas as etapas do ciclo de desenvolvimento de software realizado.

5.1. Especificação dos requisitos

Após o estudo e compreensão do trabalho desenvolvido por Silva (2019), os seguintes requisitos funcionais foram identificados:

- RF1: Dados dos responsáveis;
- RF2: Dados do idoso;
- RF3: Associar dados do responsável e idoso;
- RF4: Acesso ao sistema.
- RF5: Possibilidade de entrar em contato com os desenvolvedores da aplicação.

Após análise dos requisitos funcionais, o diagrama de caso de uso foi elaborado (Figura 4). O ator principal é o Responsável pelo Idoso, o qual tem a função de logar ao sistema, realizar o seu cadastro como responsável, realizar o cadastro do idoso (que será associado ao responsável) e recuperar a senha quando necessário. O dispositivo de queda, então, usará informações do banco de dados da aplicação que são adicionadas nos cadastros.

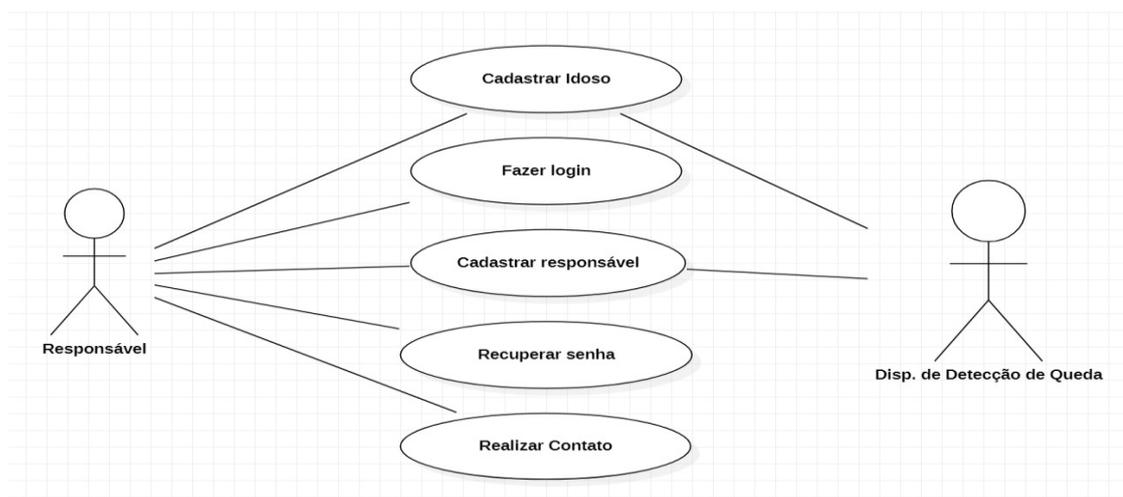


Figura 4. Diagrama de Caso de Uso

[Fonte: autoria própria]

5.2. Implementação

As subseções a seguir apresentam os 3 incrementos realizados para este trabalho. A cada incremento feito, foi entendido o requisito funcional, implementado o *front-end* e depois o *back-end* da aplicação considerando a arquitetura proposta.

O banco de dados da aplicação foi realizado à medida que os incrementos aconteciam e a Figura 5 mostra o Diagrama de Banco de Dados da aplicação. Pode-se notar 4 tabelas:

- Login: armazena os dados relacionados ao login no sistema;
- Responsável: guarda os dados relativos ao cadastro do responsável pelo idoso;
- Idoso: armazena os dados do idoso;
- Contato: caso o usuário queira entrar em contato com os programadores do sistema, essa tabela reúne alguns atributos como email, nome, assunto, observação.

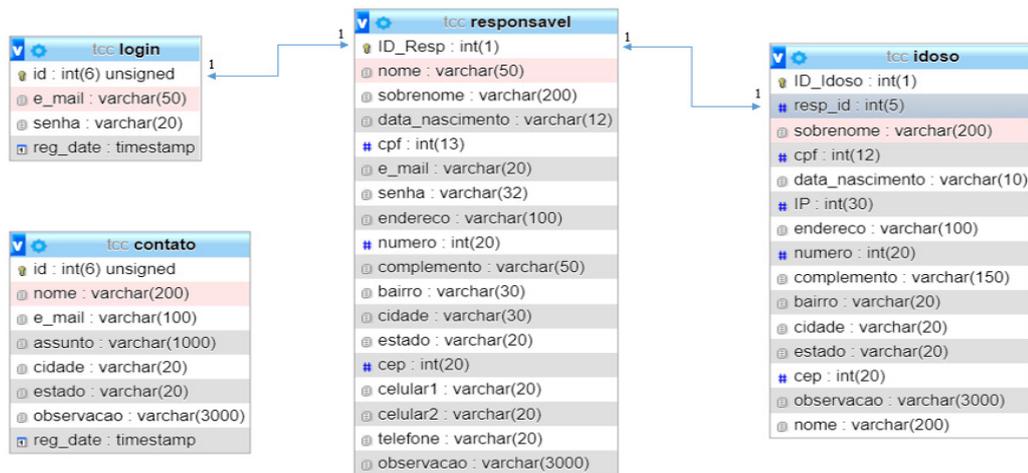


Figura 5. Diagrama do Banco de Dados

[Fonte: autoria própria]

5.2.1 Incremento 1 - Cadastrar responsável

O primeiro incremento é relacionado ao cadastro do responsável, a pessoa que receberá a notificação caso alguma queda com o idoso ocorra. Neste cadastro pensou-se em guardar informações pessoais referente ao responsável pelo idoso para que fosse facilmente localizado. A principal informação do cadastro do responsável é o e-mail que será usado pelo dispositivo para alertar sobre possíveis quedas.

Pelo aprendizado na linguagem HTML, na disciplina de Desenvolvimento Web foi possível iniciar toda a aplicação web para fazer a comunicação entre as ferramentas. A aplicação foi iniciada com a utilização do HTML depois do CSS nas páginas. A Figura 6 apresenta a marcação padrão que se utiliza no HTML com seus elementos para estruturar o código onde foram utilizadas as seguintes tags na organização da estrutura principal <DOCTYPE>, <html>, <head> e <body>. A Figura 7 mostra um trecho do CSS aplicado no desenvolvimento.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head> ...
</head>

  <style>

    @import url('https://fonts.googleapis.com/css?family=Numans');
  > html, body { ...
  }

</style>
  > <body> ...
  > <form method="POST" action="../servlet/Cadastro_Responsavel.php"> ...
  </div>

  > </form> ...
  </body>
</html>
```

Figura 6. Trecho HTML da aplicação

```
✓ html, body {
  background-image: url('Resources/img/violeta.png');
  background-color: linear-gradient;
  background-repeat: no-repeat;
  height: 78%;
  font-family: 'Numans', sans-serif;
  background-size: cover;
}
```

Figura 7. Exemplo do CSS aplicado

A Figura 8 mostra a primeira versão da tela desenvolvida para o cadastro do responsável. Pelo desenvolvimento web ser um processo iterativo e buscar a evolução para que facilite o acesso dos usuários, essa primeira tela passou por um processo de melhoria no *front-end*. Foi aplicado o CSS para dar forma a tela com cores, tabelas e fontes.

Com a ampliação do meu conhecimento iniciei os testes nas ferramentas e introduzi outras linguagens como PHP para a interação com o banco de dados (Figura 9) e para proporcionar mais dinâmica a aplicação web, e com avanço da pesquisa comecei a ter contato com Bootstrap que veio a agregar bastante valor no formato de minha estrutura por ser um *framework* para *front-end*.

Home
Quem Somos
Área do
Cadastro Responsável
Cadastro Idoso
Área do

CADASTRO RESPONSÁVEL

*Nome Completo:
*CPF:
*Data de Nascimento: dd/mm/aaaa *Sexo: Masculino Feminino
*Endereço: *Bairro:
*Cidade: Estado:
Telefone:
*Celular 1:
Celular 2:
*Criar Senha: *Confirmar Senha:
Imagem de perfil: Nenhum arquivo selecionado
Observações:

*Campos de preenchimento obrigatório
© Este é um trabalho escolar reservado a aluna do IFSP-JITO 2015 - Todos os direitos reservados.
Desenvolvido por Divana Josephat

Figura 8. Primeira versão do *design* do cadastro do responsável pelo idoso

```
servlet > Cadastro_Responsavel.php
1 <?php
2 // Inicia sessões
3 session_start();
4 require_once("../util/Conexao.php");
5 require_once("../dao/ResponsavelDAO.php");
6
7 ?>
8 <?php
```

Figura 9. Trecho do código em PHP com ligação ao banco de dados

A Figura 10 apresenta um exemplo para a criação da tabela do responsável utilizando da comunicação com o PHP em sua criação com seus tipos de dados e restrições.

```
$sql = "CREATE TABLE Responsavel.php (  
nome VARCHAR(50) NOT NULL,  
sobrenome VARCHAR(200) NOT NULL,  
data_nascimento VARCHAR(12) NOT NULL,  
cpf int (13) NOT NULL,  
e_mail VARCHAR(20) NOT NULL,  
senha VARCHAR (20) NOT NULL,  
endereco VARCHAR(100) NOT NULL,  
numero int (20) NOT NULL,  
complemento VARCHAR (50) NOT NULL,  
bairro VARCHAR(30) NOT NULL,  
cidade VARCHAR(30) NOT NULL,  
estado VARCHAR(20) NOT NULL,  
cep int (20) NOT NULL,  
celular1 VARCHAR(20) NOT NULL,  
celular2 VARCHAR(20) NOT NULL,  
telefone_fixo VARCHAR(20) NOT NULL,  
observacao VARCHAR(3000) NOT NULL  
);
```

Figura 10. Estrutura da tabela para o banco de dados em PHP

Por último foi incluído o JavaScript para validação dos formulários. A Figura 11 mostra o uso do JavaScript para validação do CEP e a Figura 12 para a validação da data de nascimento.

```
<div class="form-group col-md-3">  
  <label for="cep">*CEP</label>  
  <input type="number" id="cep" name="cep" class="form-control" id="cep">  
</div>  
  
<script>  
  var cep = '01234-050';  
  
  function isValidCEP(str){  
    var regex = /\d{5}-\d{3}/gi;  
    return regex.test(str);  
  }  
</script>
```

Figura 11. Aplicação do Script para validação do CEP

```
<div class="form-group col-md-2">  
  <label for="data_nascimento">Data Nascimento</label>  
  <div class="input-group date" style="width: 200px;" id="data_nascimento">  
    <input type="date" id="data_nascimento" name="data_nascimento" class="form-control" style="height: 35px; padding-top: 1px; padding-bottom: 1px;">  
    <span class="input-group-addon">  
      <span class="glyphicon glyphicon-calendar"></span>  
    </span>  
  </div>  
</div>  
  
<script type="text/javascript">  
  $(function () {  
    $('#datetimepicker1').datetimepicker();  
  });  
</script>
```

Figura 12. Aplicação do Script para validação da data de nascimento

O Bootstrap oferece utilização de contêineres e *grids* (Figura 13) para uniformizar os *layouts* fazendo com que colunas fiquem com suas larguras idênticas nos formulários criados

no projeto e possibilita a quebra para separação de todos os formulários. Componentes do Bootstrap precisam de *plugins* (Figura 14) do JavaScript para funcionar e a utilização dos *links* entre o CSS e Bootstrap (Figura 15) na estilização dos elementos da página HTML.

```

<div id="main" class="container-fluid" style="padding-right: 2px; padding-left: 2px; width: 999px; height: 700px"
  <div class="retangulo endResp" style="height: 600px; margin-top: 5px;"

    <div class="container">
      <h3>Dados Pessoais</h3>

      <div class="form row">
        <div class="form-group col-md-4">
          <label for="nome">*Nome</label>
          <input type="text" id="nome" name="nome" minlength="2" class="form-control" required>
        </div>

        <div class="form-group col-md-5">
          <label for="sobrenome">*Sobrenome</label>
          <input type="text" name="sobrenome" id="sobrenome" class="form-control" placeholder=" " >
        </div>

```

Figura 13. Código estruturado no Bootstrap com os *grids Container, form row e form-group*

```

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCn

```

Figura 14. *Plugins*

```

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
  integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

```

Figura 15. *Links do Bootstrap com CSS*

Após as melhorias aplicadas na tela, a segunda versão para o cadastro do responsável é mostrada na Figura 16, onde foram utilizados princípios para uma boa usabilidade, com uma tela simples para que o usuário não se perca na navegação ou fique buscando informações na tela. Ao se comparar a primeira (Figura 8) e a segunda versão (Figura 16) do *design* das telas é possível verificar cores mais brandas, alinhamento dos campos da tela, sem rolagento na tela para clicar no botão ‘Enviar’, divisão entre dados pessoais e dados cadastrais e botão ‘Voltar’ disponível no canto superior direito.

Detailed description of Figure 16: The image shows a web form titled 'CADASTRO DO RESPONSÁVEL'. At the top right is a 'Voltar' button. The form is divided into two main sections: 'Dados Pessoais' and 'Dados Cadastrais'. 'Dados Pessoais' includes fields for *Nome, *Sobrenome, Data Nascimento (dd/mm/aaaa), *CPF (somente número), *Email, and *Senha. 'Dados Cadastrais' includes fields for Endereço (rua, av. pass etc...), *Número, Complemento (apto), *Bairro, *Cidade, *Estado (dropdown), *CEP, *Celular, *Celular, and Telefone. There is also an 'Observação' text area and an 'Enviar' button at the bottom left. A footer contains a small copyright notice.

Figura 16. Segunda versão do *design* do cadastro do responsável pelo idoso

5.2.2 Incremento 2 - Cadastrar idoso

Do mesmo modo em que foi criado o cadastro do responsável, o próximo passo foi criar o formulário de cadastro do idoso utilizando as mesmas técnicas com algumas diferenças no que diz respeito ao conteúdo do formulário. A principal informação no cadastro do idoso é o IP destinado ao dispositivo que está associado ao idoso e responsável pela detecção da queda. Após o cadastro do idoso é feita a associação do mesmo ao responsável.

A Figura 17 mostra a primeira versão da tela do cadastro do idoso que após passar pelo processo de evolução resultou na segunda versão da tela (Figura 18). Observa-se novamente pontos de melhoria em termos de cor, disposição das informações, alinhamento dos campos e facilidade de acesso aos botões sem necessidade de rolar a tela.

CADASTRO DO IDOSO

*Nome Completo:

*CPF:

*Data de Nascimento: *Sexo: Masculino Feminino

*Endereço: *Bairro:

*Cidade: Estado:

Observações:

*Campos de preenchimento obrigatório

© Este é um trabalho escolar reservado a aluna do IFSP-HTO 2018 - Todos os direitos reservados. Desenvolvido por Diviana Josephat.

Figura 17. Primeira versão do *design* do cadastro do idoso

CADASTRO DO IDOSO

*Nome *Sobrenome Data Nascimento

*CPF IP

Endereço *Número Complemento *Bairro

*Cidade *Estado *CEP

Observação

© Este é um trabalho escolar reservado a aluna do IFSP-HTO 2018 - Todos os direitos reservados. Desenvolvido por Diviana Josephat.

Figura 18. Segunda versão do *design* cadastro do idoso

5.2.3 Incremento 3 - Tela Inicial da aplicação

No terceiro incremento foi desenvolvido a página inicial da aplicação com as seguintes opções: “Quem somos”, onde o usuário poderá saber um pouco sobre o detector de quedas; “Contato”

(caso de uso ‘Realizar contato’) para que as pessoas possam entrar em contato com os desenvolvedores e “Login” (caso de uso ‘Fazer login’) para acessar a aplicação. Para o primeiro acesso deve-se selecionar o *link* “Cadastre-se aqui” que apresenta a tela de cadastro de responsável e em caso de esquecimento da senha, deve-se clicar no *link* “Esqueceu sua senha?” (Figura 19).



Figura 19. Página inicial da aplicação

5.3 Testes

Os primeiros testes foram realizados em cada incremento de cada página para verificar se o acesso ao banco estava correto. Depois da aplicação finalizada, foram feitos testes na página de login depois de sua conclusão no *link* do “Cadastre-se aqui” para saber se estava direcionando corretamente para a página do Responsável onde deverá exibir o formulário de cadastro.

Na sequência foram realizados testes no preenchimento do formulário do responsável para verificar se o mesmo estava sendo direcionado para o banco corretamente, para isso o botão de enviar precisava estar apontando corretamente para a tabela criada no banco, o mesmo teste foi realizado na tela de cadastro do idoso e depois ao sair da página se ao tentar logar se estava trazendo as informações informadas pelo responsável.

6. Conclusão

O desenvolvimento da aplicação web proposto para esse trabalho foi alcançado. Em conclusão a todo esse processo posso dizer que diante desse desafio as maiores dificuldades que encontrei foi em como organizar corretamente as tags do HTML, a escolha do formato do formulário, o qual mudei diversas vezes a medida que fui compreendendo melhor como utilizar o CSS e Bootstrap. O PHP e JavaScript no começo foram também desafiantes para saber como iria utilizá-los e como eles poderiam melhorar a dinâmica dentro de todo o processo da construção da aplicação. Posso dizer que após a finalização do trabalho aprendi bastante em relação ao desenvolvimento web.

A aplicação proposta atendeu aos objetivos que eram realizar as telas de cadastro do Responsável, Idoso e o Login para ser utilizado depois dos cadastros com as informações necessárias para terem acesso ao simulador de quedas. Por este trabalho ser de cunho acadêmico, reconhece a importância de questões referente aos requisitos não funcionais como segurança, confiabilidade e robustez, porém não foram tratados no decorrer deste desenvolvimento.

A limitação deste trabalho é que não houve alteração no simulador de queda para acesso a este banco de dados, fazendo que a solução integrada fosse testada. Sendo assim, fica como proposta de trabalho futuro a manutenção no código do simulador de queda para adequar a busca das informações no banco de dados deste trabalho.

Por fim as principais disciplinas que contribuíram para o desenvolvimento desse projeto apresentadas foram Desenvolvimento web onde aprendemos como desenvolver e estruturar uma aplicação web; Interação Humano Computador que nos possibilitou a base necessária de como organizaríamos a estrutura da aplicação; Algoritmo e Lógica de Programação; Banco de Dados; Análise Orientada ao Objeto; e Engenharia de Software para definição dos requisitos funcionais e processo de desenvolvimento de software. A participação no GEPIC (Grupo de Estudo para Internet das Coisas) foi importante para o entendimento de IoT.

Referências

- Barreto, Vivas de Souza. Curso de Linguagem PHP 2000. Disponível em: <<http://www.etelg.com.br/paginaete/downloads/informatica/php.pdf>>. Acesso em: 5 jun. 2021.
- Caelum. Estruturação de páginas usando HTML e CSS. Disponível em: <<https://www.caelum.com.br/apostila/apostila-html-css-javascript.pdf>>. Acesso em 20 ago. 2021.
- Carvalho, Vinicius. MySQL Comece com o principal banco de dados open source do mercado 2015. Disponível em: <https://www.academia.edu/23820095/MySQL_Comece_com_o_principal_banco_de_dados_open_source_do_mercado>. Acesso em: 5 jun. 2021.
- Macêdo, Diego. Entendendo as aplicações web 2017. Disponível em: <<https://www.diegomacedo.com.br/entendendo-as-aplicacoes-web/>>. Acesso em: 5 jun. 2021.
- MDM Web Docs. Disponível em: <<https://developer.mozilla.org/en-US/>> Acesso em: 5 jun. 2021.
- MDM Web Docs. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>> Acesso em: 5 jun. 2021.
- MDM Web Docs. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>> Acesso em: 5 jun. 2021.
- MDM Web Docs. Sobre JavaScript. O que é JavaScript? Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript>. Acesso em: 31 jul. 2021.
- Pressman, Roger S.; Maxim, Bruce R.. Engenharia de software: uma abordagem profissional. 7 ed. Porto Alegre: McGraw, 2010.
- Santos, Bruno P. et al. (2017). Internet das Coisas: da Teoria à Prática. Disponível em: <<https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>> Acesso em: 5 jun. 2021.

Silva, Maurício Samy. Bootstrap 3.3.5. Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos 2015. Disponível em: <https://www.academia.edu/36086750/Livro_bootstrap>. Acesso em: 5 jun. 2021.

Silva, Renata Gomes da. et. al. Sistema Domiciliar de Cuidados em Saúde: Mecanismo de Detecção de Quedas de Pessoas, 2019.

Sommerville, Ian. Engenharia de Software. 9 ed. Prentice Hall, São Paulo, 2011.

World Health Organization. Envelhecimento ativo: uma política de saúde 2005. Disponível em: <https://bvsms.saude.gov.br/bvs/publicacoes/envelhecimento_ativo.pdf>. Acesso em ago. 2021.

Documento Digitalizado Público

Anexo I - Entrega do TCC final

Assunto: Anexo I - Entrega do TCC final
Assinado por: Daniela Marques
Tipo do Documento: Projeto
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- Daniela Marques, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 19/08/2021 18:49:38.

Este documento foi armazenado no SUAP em 19/08/2021. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 748442

Código de Autenticação: b09c1f475a

