

**INSTITUTO FEDERAL DO SUDESTE DE MINAS GERAIS**  
**CAMPUS AVANÇADO BOM SUCESSO**  
**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DANIEL TEIXEIRA NOGUEIRA**

**PORTAL CAMPO DAS VERTENTES: UMA PLATAFORMA PARA A DIVULGAÇÃO  
DE SERVIÇOS E PRODUTOS DA MICRORREGIÃO DO CAMPO DAS  
VERTENTES — MG**

**BOM SUCESSO, MG**

**2022**

**DANIEL TEIXEIRA NOGUEIRA**

**PORTAL CAMPO DAS VERTENTES: UMA PLATAFORMA PARA A DIVULGAÇÃO  
DE SERVIÇOS E PRODUTOS DA MICRORREGIÃO DO CAMPO DAS  
VERTENTES — MG**

Trabalho de Conclusão de Curso apresentado ao Instituto Federal do Sudeste de Minas Gerais - Campus Avançado Bom Sucesso, como parte das exigências do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, para obtenção do título de Tecnólogo.

Orientador: Prof. Pedro Henrique de Oliveira e Silva

Coorientador: Prof. Graziany Thiago Fonseca

**BOM SUCESSO, MG**

**2022**

Dados internacionais de catalogação na publicação (CIP)  
Bibliotecária responsável Maria de Lourdes Cardoso CRB-6/3242

---

N788p Nogueira, Daniel Teixeira 1997 -

Portal Campo das Vertentes: uma plataforma para divulgação de serviços e produtos da microrregião do Campo das Vertentes-MG / Daniel Teixeira Nogueira. -- 2022.

48 f.: il.; 30 cm.

Orientador: Pedro Henrique de Oliveira e Silva

Coorientador: Graziany Thiago Fonseca

Monografia (Graduação) - Instituto Federal do Sudeste de Minas Gerais, Campus Avançado Bom Sucesso, Coordenadoria de Curso de Tecnologia em Análise de Desenvolvimento de Sistemas, Bom Sucesso, MG. 2022.

1. Plataforma aberta da web. 2. Tecnologia – Serviços de informação. 3. Trabalhadores autônomos. I. Silva, Pedro Henrique de Oliveira e. II. Fonseca, Graziany Thiago. III. Instituto Federal do Sudeste de Minas Gerais, Campus Avançado Bom Sucesso. IV. Título.

CDD: 006.3

---



**MINISTÉRIO DA EDUCAÇÃO**  
**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO SUDESTE DE MINAS GERAIS**

**COMPROVANTE DE APROVAÇÃO DO PROJETO POR BANCA EXAMINADORA Nº 7 / 2022 BSCCCADS  
(11.01.10.01.01.02.01)**

Nº do Protocolo: NÃO PROTOCOLADO

Juiz de Fora-MG, 02 de Agosto de 2022

**TERMO DE APROVAÇÃO**

**DANIEL TEIXEIRA NOGUEIRA**

**PORTAL CAMPO DAS VERTENTES: UMA PLATAFORMA PARA A DIVULGAÇÃO  
DE SERVIÇOS E PRODUTOS DA MICRORREGIÃO DO CAMPO DAS VERTENTES  
- MG**

Este Trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais - *Campus* Avançado Bom Sucesso.

*(Assinado digitalmente em 05/08/2022 11:32 )*

**GRAZIANY THIAGO FONSECA**  
**PROFESSOR ENS BASICO TECN TECNOLOGICO**  
*Matrícula: 1966904*

*(Assinado digitalmente em 02/08/2022 20:24 )*

**LARISSA CARVALHO SOARES AMARAL**  
**PROFESSOR ENS BASICO TECN TECNOLOGICO**  
*Matrícula: 1033178*

*(Assinado digitalmente em 25/08/2022 10:40 )*

**PEDRO HENRIQUE DE OLIVEIRA E SILVA**  
**PROFESSOR ENS BASICO TECN TECNOLOGICO**  
*Matrícula: 1758559*

Para verificar a autenticidade deste documento entre em <https://sig.ifsudestemg.edu.br/documentos/> informando seu número:  
7, ano: 2022, tipo: **COMPROVANTE DE APROVAÇÃO DO PROJETO POR BANCA EXAMINADORA**, data de  
emissão: **02/08/2022** e o código de verificação: **7f1c216b5a**

## RESUMO

Com o crescimento constante da internet, a cada ano mais e mais pessoas estão se conectando, e assim como as pessoas procuram mais produtos no e-commerce, em busca de produtos mais variados e preços mais competitivos, elas também devem procurar por prestadores de serviços. Conforme a economia de um país sofre, seja caindo ou estagnando, muitas empresas e pequenos negócios sofrem e tendem a demitir funcionários na esperança do prejuízo não os levar à falência. A pandemia gerada pelo Covid-19 é um exemplo mais recente, e que afetou o mundo todo. Em momentos assim, muitos buscam uma forma de manter as contas em dia e muitas acabam encontrando no mercado informal uma solução, trabalhando por conta própria. O termo “trabalho informal” pode assustar algumas pessoas, mas este possui pontos positivos e pode ajudar aqueles em situação difícil, ou servir como um complemento de renda. Sendo assim, foi decidido pela criação de uma plataforma na qual as pessoas possam publicar os produtos e os serviços que tem a oferecer, ganhando assim uma maior visibilidade. E também facilitando a vida de quem procura por tais profissionais.

**Palavras-chave:** *Freelance*; plataforma; serviços; autônomos.

## **ABSTRACT**

With the constant growth of the internet, more and more people are getting connected every year, and just as people are looking more for products in ecommerce, looking for more variety of products and more competitive prices, they must also look for service providers. As a country's economy suffers, whether it is declining or stagnating, many companies and small businesses suffer and end up having to lay off employees in the hope that the damage will not drive them out of business. The pandemic generated by Covid-19 is the most recent example, and it has affected the whole world. In times like these, many people look for a way to keep up with their bills, and many end up finding a solution in the informal market, working for themselves. The term "informal work" may frighten some people, but it has positive points and can help those in difficult situations, or serve as an income supplement. Therefore, it was decided to create a platform on which people can publish the products and services they have to offer, thus gaining greater visibility.

And also making life easier for those looking for such professionals.

**Keywords:** *Freelance*; plataform; services; autonomous.

## LISTA DE QUADROS

Quadro 1 - Requisitos funcionais do sistema.....	33
Quadro 2 - Requisitos não funcionais do sistema.....	34

## LISTA DE FIGURAS

Figura 1 - Curva de defeitos.....	16
Figura 2 - Método de cascata.....	18
Figura 3 - Método de prototipação.....	18
Figura 4 - Método de desenvolvimento incremental.....	19
Figura 5 - Método em espiral.....	20
Figura 6 - Projeto MVC.....	24
Figura 7 - Exemplo de casos de uso.....	26
Figura 8 - Exemplo de diagrama de classe.....	27
Figura 9 - Exemplo de relação entre classes.....	28
Figura 10 - Sistema de banco de dados.....	29
Figura 11 - Diagrama de caso de uso para o visitante.....	35
Figura 12 - Diagrama de caso de uso para o profissional.....	35
Figura 13 - Diagrama lógico do banco de dados.....	36
Figura 14 - Sitemap do site.....	36
Figura 15 - Exemplo de código para função migrate.....	38
Figura 16 - Sistema de arquivos do projeto.....	39
Figura 17 - Página Inicio.....	40
Figura 18 - Página de visualização do profissional.....	41
Figura 19 - Página Mural.....	42
Figura 20 - Página para cadastro do profissional.....	43
Figura 21 - Página para cadastro dos dados pessoais.....	43
Figura 22 - Página para cadastro de serviços.....	44
Figura 23 - Página para visualização de mensagens.....	45



## LISTA DE SIGLAS

UML	Unified Modeling Language
SGBD	Sistemas de Gestão de Banco de Dados
VSCODE	Visual Studio Code
MVC	Model, View, Controller
DAO	Data Access Object
RUP	Rational Unified Process
IoC	Inversion of Control
CRUD	Create, Update, Delete
OMG	Object Management Group
RDBMS	Relational Database Management System
JS	JavaScript

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	11
1.1 OBJETIVOS.....	12
<b>1.1.1 Objetivo geral</b> .....	<b>12</b>
<b>1.1.2 Objetivos específicos</b> .....	<b>12</b>
1.2 JUSTIFICATIVA.....	12
1.3 ORGANIZAÇÃO DO TRABALHO.....	13
<b>2 METODOLOGIA</b> .....	14
<b>3 REVISÃO BIBLIOGRÁFICA</b> .....	15
3.1 O USO DA INTERNET NA DIVULGAÇÃO DE PRESTAÇÃO DE SERVIÇOS.....	15
3.2 DESENVOLVIMENTO DE SOFTWARE.....	16
3.3 MODELOS PRESCRITIVOS DE PROCESSO DE SOFTWARES.....	17
3.4 PROCESSO UNIFICADO DE SOFTWARE.....	20
<b>3.4.1 Fases do Processo Unificado</b> .....	<b>21</b>
3.5 PADRÕES DE PROJETOS.....	22
<b>3.5.1 Padrões de projeto MVC</b> .....	<b>23</b>
<b>3.5.2 Padrões de projeto DAO</b> .....	<b>25</b>
3.6 LINGUAGEM DE MODELAGEM UNIFICADA– UML.....	25
<b>3.6.1 Diagrama de caso de uso</b> .....	<b>26</b>
<b>3.6.2 Diagrama de classes</b> .....	<b>27</b>
<b>3.6.3 Diagrama lógico de banco de dados</b> .....	<b>28</b>
<b>3.6.4 Sitemap</b> .....	<b>28</b>
3.7 BANCO DE DADOS.....	29
<b>3.7.1 MySQL</b> .....	<b>30</b>
3.8 LARAVEL.....	30
3.9 TRABALHOS RELACIONADOS.....	31
<b>4 DESENVOLVIMENTO</b> .....	32
4.1 SISTEMA PROPOSTO.....	32
4.2 ANÁLISE, LEVANTAMENTO E MODELAGEM DOS REQUISITOS.....	32
<b>4.2.1 Diagrama de caso de uso</b> .....	<b>34</b>
<b>4.2.2 Diagrama lógico de banco de dados</b> .....	<b>36</b>
<b>4.2.3 Sitemap</b> .....	<b>36</b>

4.3 O SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS.....	37
4.4 DESENVOLVIMENTO DO SISTEMA .....	38
<b>5 RESULTADOS .....</b>	<b>400</b>
5.1 MURAL DIVULGAÇÃO DE PRESTADORES DE SERVIÇOS: PÁGINA PRINCIPAL .....	400
5.2 MURAL DIVULGAÇÃO DE SERVIÇOS PRESTADOS .....	422
5.3 CADASTRO DE PRESTADORES DE SERVIÇOS.....	422
5.4 MÓDULO PRESTADOR DE SERVIÇOS .....	433
<b>5.4.1 Cadastro de serviços .....</b>	<b>433</b>
<b>5.4.2 Visualizar mensagens .....</b>	<b>44</b>
<b>6 CONCLUSÃO E TRABALHOS FUTUROS .....</b>	<b>455</b>
<b>REFERÊNCIAS .....</b>	<b>4747</b>

## 1 INTRODUÇÃO

A economia dos países sofreu com a presença do vírus COVID-19, que acompanhada de medidas de isolamento e restrição, acabaram levando muitos estabelecimentos e comércios a fecharem as suas portas, conseqüentemente acarretando na demissão de muitos trabalhadores. Sem trabalho e sem expectativa de quando conseguiriam um novo, as pessoas tiveram que encontrar outros meios de ganhar dinheiro e manter as contas de casa em dia.

Montar o seu próprio negócio e trabalhar por conta própria é uma boa opção, seja para quem quer trabalhar no conforto de casa, para quem quer ganhar um dinheiro a mais, para quem quer fazer os próprios horários, ou simplesmente para aquelas pessoas que estão desempregadas e não querem, ou não podem ficar esperando uma oportunidade de emprego surgir.

O trabalho como *freelancer* não é uma novidade gerada pela pandemia, mas com certeza foi impulsionada por ela. Vários novos trabalhadores surgiram, e dentre aqueles que já trabalhavam desta forma, muitos viram a demanda por seus serviços aumentarem. A contratação de *freelancers* não se dá apenas por pessoas, mas também por empresas, que encontraram nessa forma de vínculo, um jeito de suprir as suas necessidades.

Segundo Nader (2021) no segundo trimestre de 2021 o Brasil chegou a 24,8 milhões de trabalhadores autônomos, essa quantia corresponde a 28,3% da população ativa no mercado, segundo o IBGE (IBGE, 2021, citado por NADER, 2021). Esses resultados apontam um crescimento de 4,2% comparados ao primeiro trimestre de 2021. O principal crescimento foi em atividades relacionadas à alojamento e alimentação (9,1%), seguida por construção (5,7%), e serviços domésticos (4%).

No passado as pessoas que trabalhavam por conta própria precisavam contratar serviços de divulgação como rádio, jornal, ou torcer para que os seus clientes satisfeitos comentassem sobre eles. No entanto, o alcance destas formas de propagandas era limitado, o que não acontece hoje graças à internet, que além de oferecer preços bem mais competitivos, ainda possui um alcance muito superior.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

O objetivo geral deste trabalho é propor uma plataforma *on-line* para a divulgação de prestação de serviços da microrregião Campos das Vertentes – MG.

### 1.1.2 Objetivos específicos

Com o intuito de atingir o objetivo geral, foi necessário realizar os seguintes objetivos específicos:

- Definição de uma metodologia de desenvolvimento de *software* para a implementação da plataforma *on-line*;
- Definição das tecnologias e ferramentas que foram utilizadas para o desenvolvimento da plataforma *on-line*;
- Definição das etapas necessárias para o desenvolvimento do sistema;
- Levantamento, análise e modelagem dos requisitos do sistema;
- Desenvolvimento do mural de divulgação dos prestadores de serviço;
- Desenvolvimento do mural de divulgação dos serviços cadastrados;
- Desenvolvimento do cadastro de prestadores de serviços;
- Desenvolvimento do módulo para o gerenciamento de informações dos prestadores de serviços;

## 1.2 JUSTIFICATIVA

A internet facilitou a vida das pessoas, dando a elas acesso fácil à informação, e por consequência aproximando clientes e prestadores de serviços. Em momentos em que o desemprego afeta tantas pessoas (o que pode ser agravado em momentos como o de uma pandemia), elas acabam buscando outras formas de ganhar dinheiro, uma dessas formas é prestando serviços como autônomo, e a internet é uma grande vitrine para tal. Partindo disso, surge a necessidade da criação de uma ferramenta

que consiga reunir vários destes trabalhadores em um mesmo local, aumentando sua visibilidade, e facilitando a vida de quem procura por tais serviços.

### 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho é composto por 6 capítulos. O primeiro trata da introdução, os objetivos e a justificativa. No segundo capítulo encontra-se a metodologia usada para a construção deste projeto. No terceiro está o referencial teórico, abordando assuntos do trabalho e de sua construção. O quarto retrata o desenvolvimento do mesmo. No quinto são apresentados os resultados alcançados ao final. O sexto contém a conclusão e ideias para trabalhos futuros.

## 2 METODOLOGIA

O presente trabalho foi realizado uma pesquisa bibliográfica, levantamento de requisitos, observação direta e estudo de caso, com o objetivo de coletar as informações necessárias para e efetuar a análise, a modelagem e o desenvolvimento do sistema proposto. A pesquisa bibliográfica foi realizada através de consultas de artigos científicos, bem como livros. Para o levantamento dos requisitos foi realizado uma pesquisa de sistemas semelhantes bem como conversas informais com prestadores de serviços, o estudo de caso foi realizado através da observação direta e pesquisa de plataformas *on-line* de divulgação de serviços, a fim de se obter os requisitos necessários para o desenvolvimento do sistema e os principais problemas enfrentados pelos prestadores de serviços da microrregião Campos das Vertentes - MG.

Após a coleta e análise dos requisitos foi realizado a modelagem dos mesmos através da Linguagem de Modelagem Unificada (UML), o que possibilitou a criação dos diagramas de caso de uso, e classes, diagrama lógico de banco de dados e o sitemap, que auxiliaram num melhor entendimento do sistema, tornando o desenvolvimento do mesmo mais funcional e completo. Juntamente com a modelagem dos requisitos, foi criado o Banco de Dados, onde todas as informações dos sistemas serão armazenadas de forma segura e íntegra.

O Sistema de gerenciamento de banco de dados (SGBD) escolhido para o desenvolvimento do Sistema foi MySQL. Para o desenvolvimento do Sistema, foi usado o Laravel, framework PHP utilizado para o desenvolvimento *web*, que utiliza a arquitetura MVC e tem como principal característica ajudar a desenvolver aplicações seguras e performáticas de forma rápida, com código limpo e simples. Para a codificação do sistema foi utilizado a ferramenta Visual Studio Code (VSCODE), editor desenvolvido pela Microsoft, baseado em electron, um framework que é usado para desenvolver aplicativos Node.js para o *desktop* rodando no motor de *layout* Blink. O VSCODE inclui suporte para depuração, controle de versionamento Git incorporado, complementação inteligente de código, snippets e refatoração de código, podendo ser customizado, além de ser um *software* livre e de código aberto.

### 3 REVISÃO BIBLIOGRÁFICA

Neste capítulo será realizado uma revisão bibliográfica dos conteúdos utilizados para o desenvolvimento da plataforma *on-line* para a divulgação de prestação de serviços da microrregião Campos das Vertentes - MG, bem como mostrar algumas plataformas existentes relacionadas com a divulgação de prestação de serviços.

#### 3.1 O USO DA INTERNET NA DIVULGAÇÃO DE PRESTAÇÃO DE SERVIÇOS

Segundo Nogueira (2019) o número de pessoas conectadas à internet no mundo cresceu 1114% desde 2010. Segundo dados da Netcraft (NETCRAFT, 2019), até 2019, havia 1,45 bilhão de sites em atividade na rede, número 7 vezes maior do que os 215 milhões que haviam dez anos antes.

Durante a pandemia por COVID-19 o número de pessoas conectadas à rede aumentou substancialmente, segundo a União Internacional de Telecomunicações (ITU) 4,9 bilhões de pessoas se conectaram a internet no ano de 2021, em parte esse crescimento se resultou devido às medidas de distanciamento social adotadas pelos países (INTERNATIONAL TELECOMMUNICATION UNION, 2021). Com o fechamento de empresas, escolas, e outros locais, muitas das tarefas cotidianas realizadas presencialmente pelas pessoas passaram a ser executadas de forma *on-line*. Alunos passaram a assistir às aulas pelo computador, trabalhadores cujas empresas foram completamente, ou parcialmente, fechadas começaram a trabalhar de forma remota, entre outras.

Dados obtidos por uma pesquisa realizada pela empresa TIC Domicílios (2020) mostraram que cerca de 81% da população do Brasil está conectada à internet, este número equivale a 152 milhões de pessoas. Segundo a mesma pesquisa, a proporção de domicílios com acesso à internet aumentou principalmente na classe C, onde aumentou para 91%; nas classes D e E houve um aumento para 64%.



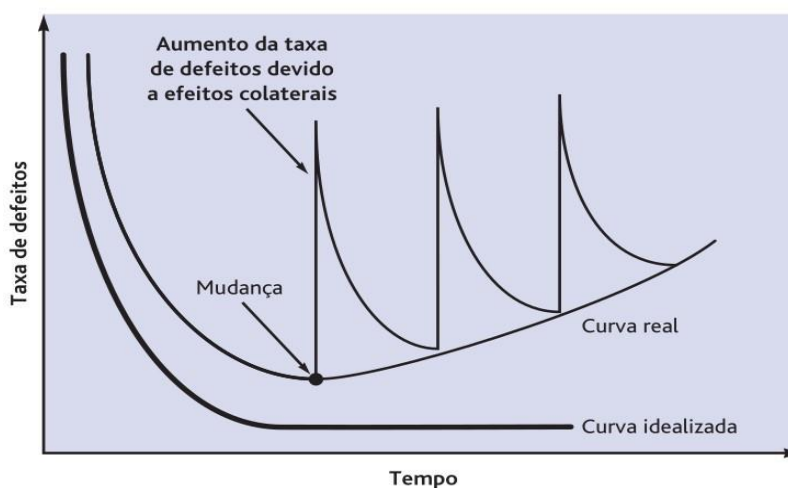
### 3.2 DESENVOLVIMENTO DE SOFTWARE

Segundo Pressman (2016) um *software* consiste em: (1) instruções (programas de computador) que quando executadas, fornecem características, funções e desempenho desejados; (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente; e (3) informação descritiva, tanto na forma impressa quanto na virtual, descrevendo a operação e o uso dos programas.

Sem dúvida, poder-se dar outras definições mais completas, mas, provavelmente, uma definição mais formal não melhoraria, consideravelmente, a compreensão do que é *software*. Para conseguir isso, é importante examinar as características do *software* que o tornam diferenciado de outras coisas que os seres humanos constroem. *Software* é mais um elemento de sistema lógico do que físico. Portanto, o *software* tem uma característica fundamental que o torna consideravelmente diferente do *hardware*: o *software* não “se desgasta” (PRESSMAN, 2016).

*Software* não é suscetível aos fatores maléficos do ambiente que fazem com que o *hardware* se desgaste. Portanto, teoricamente, a curva da taxa de defeitos para *software* deveria assumir a forma da “curva idealizada”, mostrada na Figura 1. Defeitos ainda não descobertos irão resultar em altas taxas logo no início da vida de um programa. Entretanto, esses serão corrigidos, e a curva se achata, como mostrado. A curva idealizada é uma simplificação grosseira de modelos de defeitos reais para *software*. Porém, a implicação é clara: o *software* não se desgasta, mas deteriora (PRESSMAN, 2016).

Figura 1 - Curva de defeitos



Fonte: Pressman (2016)

Quando se fala sobre a qualidade do *software* profissional, devemos levar em conta que o *software* é usado e alterado pelas pessoas, além de seus desenvolvedores. A qualidade, portanto, implica não apenas o que o *software* faz. Ao contrário, ela tem de incluir o comportamento do *software* enquanto ele está executando, bem como a estrutura e a organização dos programas do sistema e a documentação associada. Isso se reflete nos atributos de *software* chamados não funcionais ou de qualidade. Exemplos desses atributos são o tempo de resposta do *software* a uma consulta do usuário e a compreensão do código do programa. Um conjunto específico de atributos que você pode esperar de um *software* obviamente depende da aplicação. Portanto, um sistema bancário deve ser seguro, um jogo interativo deve ser ágil, um sistema de comutação de telefonia deve ser confiável, e assim por diante (SOMMERVILLE, 2011).

### 3.3 MODELOS PRESCRITIVOS DE PROCESSO DE SOFTWARES

Um processo de *software* é um conjunto de atividades relacionadas que levam à produção de um produto de *software*. Essas atividades podem envolver o desenvolvimento de *software* a partir do zero em uma linguagem padrão de programação. No entanto, aplicações de negócios não são necessariamente desenvolvidas dessa forma. Em 2011, novos *softwares* de negócios eram desenvolvidos por meio da extensão e modificação de sistemas existentes ou por meio da configuração e integração de prateleira ou componentes do sistema (SOMMERVILLE, 2011).

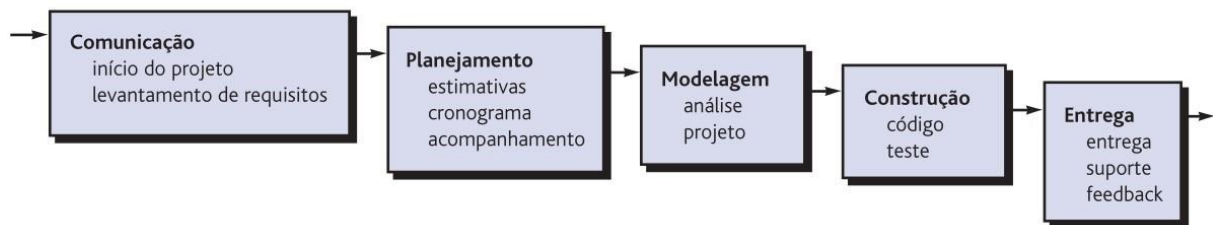
Os processos têm evoluído de maneira a tirarem melhor proveito das capacidades das pessoas em uma organização, bem como das características específicas do sistema em desenvolvimento. Para alguns sistemas, como sistemas críticos, é necessário um processo de desenvolvimento muito bem estruturado; para sistemas de negócios, com requisitos que se alteram rapidamente, provavelmente será mais eficaz um processo menos formal e mais flexível (SOMMERVILLE, 2011).

Dentre os modelos prescritivos, também conhecidos como modelos tradicionais, pode-se citar quatro que são mais conhecidos: o modelo em cascata, o modelo de prototipação, o modelo de desenvolvimento incremental, e o modelo em espiral. Segundo Sommerville (2011) esses modelos não são mutuamente exclusivos e muitas vezes são usados em conjunto, especialmente para o desenvolvimento de

sistemas de grande porte. Para sistemas de grande porte, faz sentido combinar algumas das melhores características do modelo em cascata e dos modelos de desenvolvimento incremental.

O modelo em cascata, apresentado na Figura 2, propõe uma abordagem organizada em sequência, assim, para que se possa passar para a próxima etapa, deve-se primeiro concluir a anterior. Segundo Sommerville (2011) o modelo em cascata deve ser usado apenas quando os requisitos são bem compreendidos e pouco provavelmente venham a ser radicalmente alterados durante o desenvolvimento do sistema.

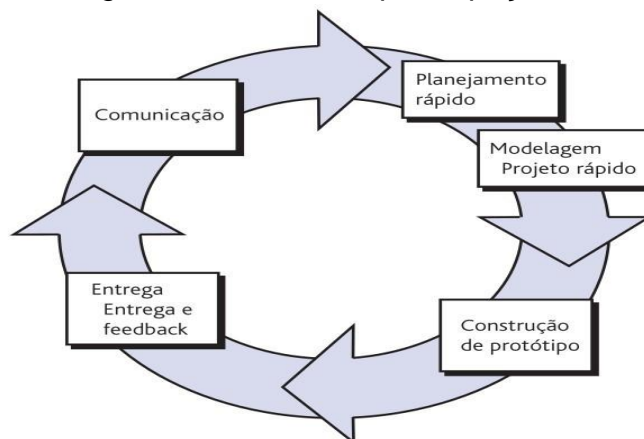
Figura 2 - Método de cascata



Fonte: Pressman (2016)

Frequentemente, o cliente define uma série de objetivos gerais para o software, mas não identifica, detalhadamente, os requisitos para funções e recursos. Em outros casos, o desenvolvedor se encontra inseguro quanto à eficiência de um algoritmo, quanto à adaptabilidade de um sistema operacional ou quanto à forma em que deve ocorrer a interação homem-máquina. Em situações como essas, e em muitas outras, o paradigma da prototipação (Figura 3) pode ser a melhor abordagem (PRESSMAN, 2016).

Figura 3 - Método de prototipação



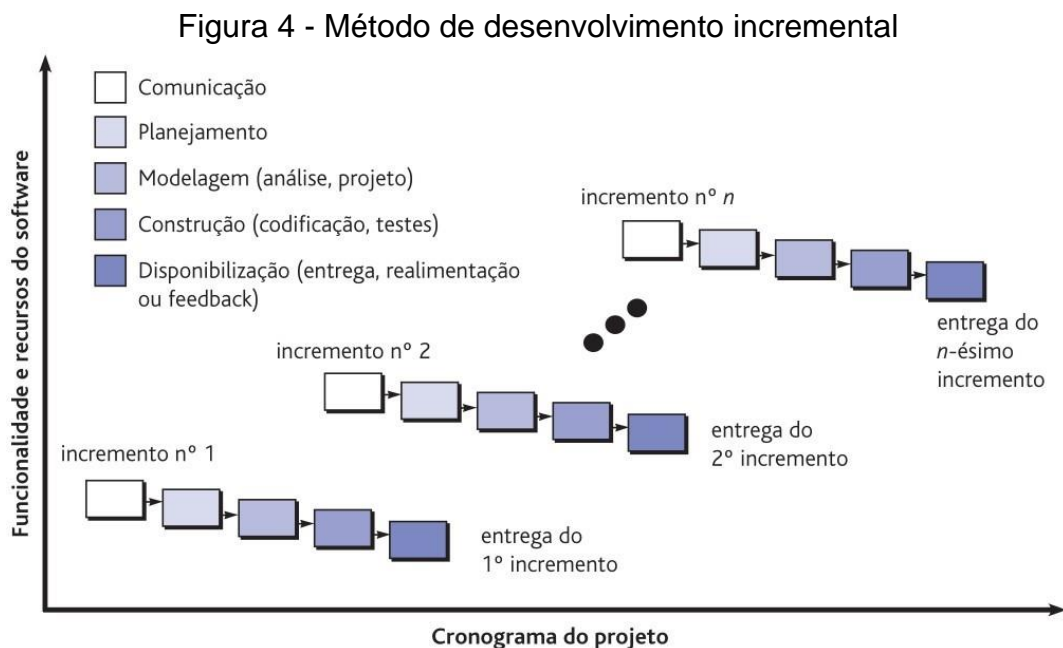
Fonte: Pressman (2016)

Embora a prototipação possa ser utilizada como um modelo de processo isolado, ela é mais comumente utilizada como uma técnica a ser implementada no contexto de qualquer um dos modelos de processo citados neste capítulo.

Independentemente da forma como é aplicado, quando os requisitos estão obscuros, o paradigma da prototipação auxilia os envolvidos a compreender melhor o que está para ser construído (PRESSMAN, 2016).

O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido (Figura 4). Atividades de especificação, desenvolvimento e validação são intercaladas, e não separadas, com rápido feedback entre todas as atividades (SOMMERVILLE, 2011).

Este modelo foi criado para ser uma evolução do modelo em cascata. Segundo Pressman (2016) prazos apertados, determinados pelo mercado, tornam impossível concluir um produto de software abrangente, porém uma versão limitada tem de ser introduzida para aliviar e/ou atender às pressões comerciais ou da concorrência. Modelos evolucionários são iterativos. Apresentam características que possibilitam desenvolver versões cada vez mais completas do software.

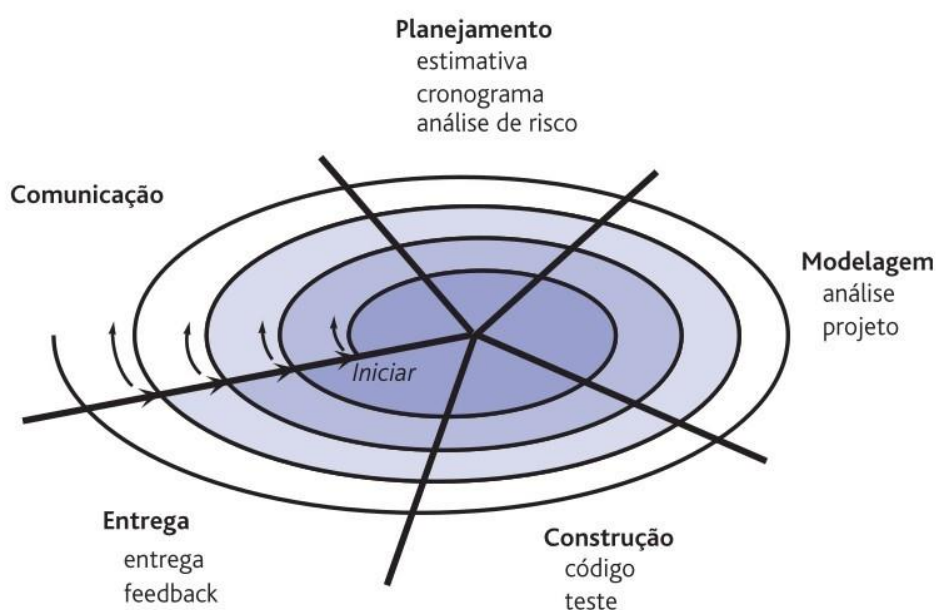


Fonte: Pressman (2016)

Originalmente proposto por Barry Boehm (1988, apud PRESSMAN, 2016), o modelo espiral, apresentado na Figura 5, é um modelo de processo de software evolucionário que une a natureza iterativa da prototipação aos aspectos sistemáticos e controlados do modelo cascata. Tem potencial para o rápido desenvolvimento de versões cada vez mais completas do software (PRESSMAN, 2016).

Cada volta na espiral representa uma fase do processo de software. Dessa forma, a volta mais interna pode preocupar-se com a viabilidade do sistema; o ciclo seguinte, com definições de requisitos; o seguinte com o projeto do sistema, e assim por diante. O modelo em espiral combina prevenção e tolerância a mudanças, assume que mudanças são um resultado de risco de projeto e inclui atividades explícitas de gerenciamento de riscos para sua redução (SOMMERVILLE, 2011).

Figura 5 - Método em espiral



Fonte: Pressman (2016)

### 3.4 PROCESSO UNIFICADO DE SOFTWARE

Segundo Falbo (2000) o Processo Unificado proposto pela Rational (*Rational Unified Process – RUP*) foi criado para apoiar o desenvolvimento orientado a objetos, fornecendo uma forma sistemática para se obter reais vantagens no uso da Linguagem de Modelagem Unificada (*Unified Modeling Language – UML*).

Ainda segundo Falbo (2000) o RUP está fundamentado em três princípios básicos: orientação a casos de uso, arquitetura e iteração. Ele é dito dirigido a casos

de uso, pois são os casos de uso que orientam todo o processo de desenvolvimento. Com base no modelo de casos de uso, são criados uma série de modelos de análise, projeto e implementação, que realizam estes casos de uso. É centrado em arquitetura, pois defende a definição de um esqueleto para a aplicação (a arquitetura), a ganhar corpo gradualmente ao longo do desenvolvimento. Finalmente, o RUP é iterativo e incremental, oferecendo uma abordagem para particionar o trabalho em porções menores ou miniprojetos.

### 3.4.1 Fases do Processo Unificado

O ciclo de vida adotado no RUP é tipicamente evolutivo. Contudo, uma forma de organização em fases é adotada para comportar os ciclos de desenvolvimento, permitindo uma gerência mais efetiva de projetos complexos. Ao contrário do tradicionalmente definido como fases na maioria dos modelos de ciclo de vida – planejamento, levantamento de requisitos, análise, projeto, implementação e testes, são definidas fases ortogonais a estas (FALBO, 2000):

- **Concepção:** nesta fase, é estabelecido o escopo do projeto e suas fronteiras, determinando os principais casos de uso do sistema. Esses casos de uso devem ser elaborados com a precisão necessária para se proceder estimativas de prazos e custos. As estimativas devem ser globais para o projeto como um todo e detalhadas para a fase seguinte. Assim, a ênfase nesta etapa recai sobre o planejamento e, por conseguinte, é necessário levantar requisitos do sistema e preliminarmente analisá-los. Ao término dessa fase, são examinados os objetivos do projeto para se decidir sobre a continuidade do desenvolvimento;
- **Elaboração:** o propósito desta fase é analisar mais refinadamente o domínio do problema, estabelecer uma arquitetura de fundação sólida, desenvolver um plano de projeto para o sistema a ser construído e eliminar os elementos de projeto que oferecem maior risco. Embora o processo deva sempre acomodar alterações, as atividades da fase de elaboração asseguram que os requisitos, a arquitetura e os planos estão suficientemente estáveis e que os riscos estão suficientemente mitigados, de modo a se poder prever com precisão os custos e prazos para a conclusão do desenvolvimento.

- **Construção:** durante esta fase, um produto completo é desenvolvido de maneira iterativa e incremental, para que esteja pronto para a transição à comunidade usuária.
- **Transição:** nesta fase, o software é disponibilizado à comunidade usuária. Após o produto ter sido colocado em uso, naturalmente surgem novas considerações que vão demandar a construção de novas versões para permitir ajustes do sistema, corrigir problemas ou concluir algumas características que foram postergadas.

### 3.5 PADRÕES DE PROJETOS

Os padrões de projeto foram obtidos a partir das ideias apresentadas por Christopher Alexander (ALEXANDER *et al.*, 1977, apud SOMMERVILLE, 2011), que sugeriu haver padrões comuns de projeto de prédios que eram inerentemente agradáveis e eficazes. O padrão é uma descrição do problema e da essência de sua solução, de modo que a solução possa ser reusada em diferentes contextos. O padrão não é uma especificação detalhada. Em vez disso, você pode pensar nele como uma descrição de conhecimento e experiência, uma solução já aprovada para um problema comum (SOMMERVILLE, 2011).

Os padrões tiveram um enorme impacto no projeto de software orientado a objetos. Além de serem soluções já testadas para problemas comuns, tornaram-se um vocabulário para falar sobre um projeto. Você pode, portanto, explicar seu projeto por meio de descrições dos padrões que você usou. Isso é particularmente verdadeiro para os padrões de projeto mais conhecidos que foram originalmente descritos pela 'Gangue dos Quatro' em seu livro de padrões (GAMMA *et al.*, 1995, citado por SOMMERVILLE, 2011). Outras descrições de padrões particularmente importantes são as publicadas em uma série de livros de autores da Siemens, uma grande empresa europeia de tecnologia (BUSCHMANN *et al.*, 1996; BUSCHMANN *et al.*, 2007a; BUSCHMANN *et al.*, 2007b; KIRCHER e JAIN, 2004; SCHMIDT *et al.*, 2000, apud SOMMERVILLE, 2011).

Os padrões podem ter finalidades de criação, comportamento e estrutural. Os padrões de criação abstraem a forma como os objetos são criados. Um padrão de criação de classes usa herança para definir qual classe será instanciada; um padrão de criação de objetos delega esta tarefa para outro objeto. Os padrões estruturais

cuidam de como classes e objetos se compõem. Para compor interfaces e implementações são usados os de classes, e para obter novas funcionalidades a partir da composição de objetos é usado o padrão de objetos. Já os padrões de comportamento descrevem o comportamento de objetos, classes, e também a comunicação entre os objetos. Os padrões comportamentais de classes utilizam herança, enquanto os padrões de objetos utilizam a composição de objetos (LEITE, 2005).

Os principais padrões de projetos são (LEITE, 2005):

- **Abstract Factory:** usada quando se deseja isolar a aplicação da implementação da classe concreta, e fornece uma interface para a criação de famílias de objetos.
- **Factory Method:** este transfere para as subclasses a tarefa de decidir qual classe será instanciada. É usado quando há a necessidade de encapsular a criação de uma classe se isolando do conhecimento da classe concreta.
- **Singleton:** produz uma instância única, e esta estará disponível para toda a aplicação. Usada em aplicações que precisam manter um controle sobre a quantidade de instâncias criadas.
- **Adapter:** possibilita que classes com interfaces diferentes trabalhem em conjunto ao converter a interface de uma classe por outra.
- **Template Method:** permite que subclasses redefinam um algoritmo sem alterar sua composição original. Este conduz a uma inversão de controle (IoC), onde a classe-mãe chama as subclasses, e não o contrário.

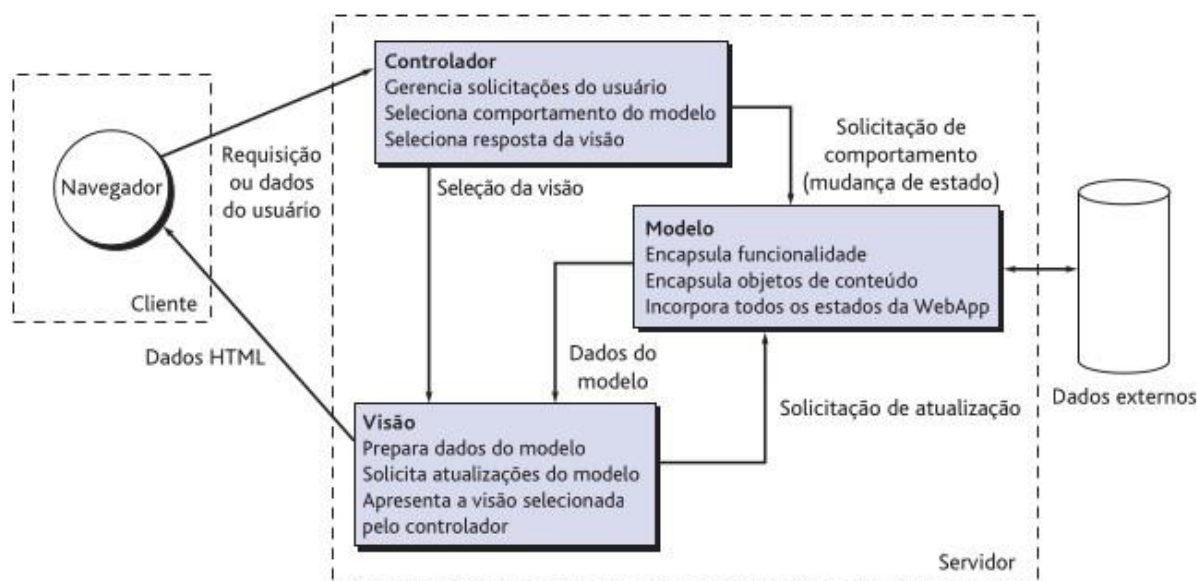
### 3.5.1 Padrões de projeto MVC

A arquitetura Modelo-Visão-Controlador (MVC, *Model-View-Controller*) (KRASNER, 1988, apud PRESSMAN, 2016), apresentado na Figura 6, é uma de vários modelos de infraestrutura sugeridos para WebApps que separam a interface do usuário da funcionalidade e do conteúdo de informações de uma WebApp. O modelo (algumas vezes conhecido como “objeto-modelo”) contém todo o conteúdo e a lógica de processamento específicos à aplicação, inclusive todos os objetos de conteúdo, acesso a fontes de dados/ informações externas e toda a funcionalidade de processamento específica para a aplicação. A visão contém todas as funções



específicas à interface e possibilita a apresentação do conteúdo e lógica de processamento, inclusive todos os objetos de conteúdo, acesso a fontes de dados/informações externas e toda a funcionalidade de processamento exigida pelo usuário. O controlador gerencia o acesso ao modelo e à visão e coordena o fluxo de dados entre eles; ele também seleciona o objeto visão aplicável, de acordo com a solicitação do usuário. Uma vez determinado o tipo de solicitação, é transmitida uma solicitação de comportamento ao modelo, que implementa a funcionalidade ou recupera o conteúdo necessário para atender à solicitação. O objeto-modelo pode acessar dados armazenados em um banco de dados corporativo, como parte de um repositório de dados local ou de um conjunto de arquivos independentes. Os dados desenvolvidos pelo modelo devem ser formatados e organizados pelo objeto de visão apropriado e transmitidos do servidor de aplicações de volta para o navegador instalado no cliente para exibição na máquina do usuário (PRESSMAN, 2016).

Figura 6 - Projeto MVC



Fonte: Pressman (2016)

Empresas de TI gastam mais com manutenção de softwares do que com a criação de novos, e por esse motivo as consultorias de arquitetura recomendam o uso de padrões, mesmo se o projeto for pequeno.

### 3.5.2 Padrões de projeto DAO

Segundo Dalepiane (2014), o padrão de projeto DAO surgiu com a necessidade de separarmos a lógica de negócios da lógica de persistência de dados. Este padrão permite que seja possível mudar a forma de persistência sem que isso influencie em nada na lógica de negócio, além de tornar nossas classes mais legíveis.

Classes DAO são responsáveis por trocar informações com o SGBD e fornecer operações CRUD e de pesquisas, elas devem ser capazes de buscar dados no banco e transformar esses em objetos ou lista de objetos, fazendo uso de listas genéricas, também deverão receber os objetos, converter em instruções SQL e mandar para o banco de dados (DALEPIANE, 2014).

Toda interação com a base de dados se dará através destas classes, nunca das classes de negócio, muito menos de formulários.

Se este padrão for aplicado corretamente ele vai abstrair completamente o modo de busca e gravação dos dados, tornando isso transparente para aplicação e facilitando muito na hora de fazermos manutenção na aplicação ou migrarmos de banco de dados. Também consegue-se centralizar a troca de dados com o SGBD, teremos um ponto único de acesso a dados, tendo assim nossa aplicação um ótimo design orientado a objeto (DALEPIANE, 2014).

### 3.6 LINGUAGEM DE MODELAGEM UNIFICADA– UML

Segundo Pressman (2016), a Linguagem de Modelagem Unificada (UML, *Unified Modeling Language*) é “uma linguagem-padrão para descrever/documentar projeto de software. A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software intensivo” (BOOCH, 2005, apud PRESSMAN, 2016). Em outras palavras, assim como os arquitetos criam plantas e projetos para serem usados por uma empresa de construção, os arquitetos de software criam diagramas UML para ajudar os desenvolvedores de software a construir o software. Se você entender o vocabulário da UML (os elementos visuais do diagrama e seus significados), poderá facilmente entender e especificar um sistema e explicar o projeto desse sistema para outros interessados.

Grady Booch, Jim Rumbaugh e Ivar Jacobson desenvolveram a UML na década de 1990, com muitas opiniões da comunidade de desenvolvimento de

software. A UML combinou um grupo de notações de modelagem concorrentes usadas pela indústria do software na época. Em 1997, a UML 1.0 foi apresentada ao OMG (Object Management Group), uma associação sem fins lucrativos dedicada a manter especificações para serem usadas pela indústria de computadores. A UML 1.0 foi revisada tornando-se a UML 1.1 e adotada mais tarde naquele ano. O padrão atual é a UML 2.32 e agora é um padrão ISO. Como esse padrão é novo, muitas referências mais antigas, como (GAMMA, 1995, apud PRESSMAN, 2016), não usam a notação UML (PRESSMAN, 2016).

### 3.6.1 Diagrama de caso de uso

Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação. Essa é, então, suplementada por informações adicionais que descrevem a interação com o sistema. A informação adicional pode ser uma descrição textual ou um ou mais modelos gráficos, como diagrama de sequência ou de estados da UML. Os casos de uso são documentados por um diagrama de casos de uso de alto nível. O conjunto de casos de uso representa todas as possíveis interações que serão descritas nos requisitos de sistema. Atores, que podem ser pessoas ou outros sistemas, são representados como figuras 'palito'. Cada classe de interação é representada por uma elipse. Linhas fazem a ligação entre os atores e a interação. Opcionalmente, pontas de flechas podem ser adicionadas às linhas para mostrar como a interação se inicia. Essa situação é ilustrada na Figura 7, que mostra alguns dos casos de uso para o sistema de informações de pacientes (SOMMERVILLE, 2011).

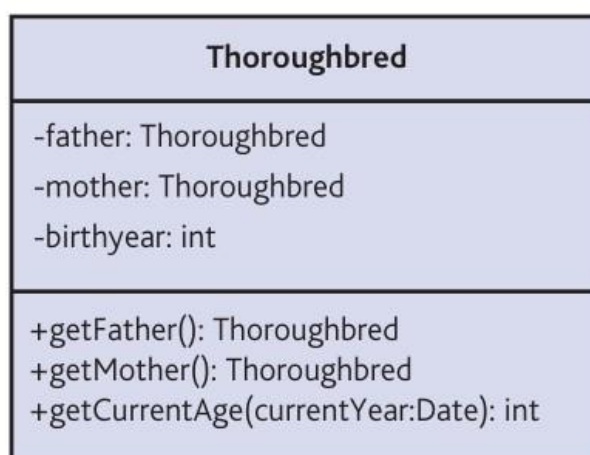


Fonte: Sommerville (2011)

### 3.6.2 Diagrama de classes

Um diagrama de classe (Figura 8) fornece uma visão estática ou estrutural do sistema. Ele não mostra a natureza dinâmica das comunicações entre os objetos das classes no diagrama; Seus elementos principais são caixas, ou seja, ícones usados para representar classes e interfaces. Cada caixa é dividida em partes horizontais. A parte superior contém o nome da classe. A seção do meio lista os atributos da classe. Os atributos podem ser valores que a classe calcula a partir de suas variáveis de instância ou valores que a classe pode obter de outros objetos dos quais é composta. A terceira seção do diagrama de classes contém as operações ou comportamentos da classe. Uma operação refere-se ao que os objetos da classe podem fazer. Usualmente é implementada como um método da classe. Vale mencionar que o ícone que representa uma classe pode ter outras partes opcionais. Por exemplo, uma quarta seção na parte inferior da caixa de classe pode ser usada para listar as responsabilidades da classe (PRESSMAN, 2016).

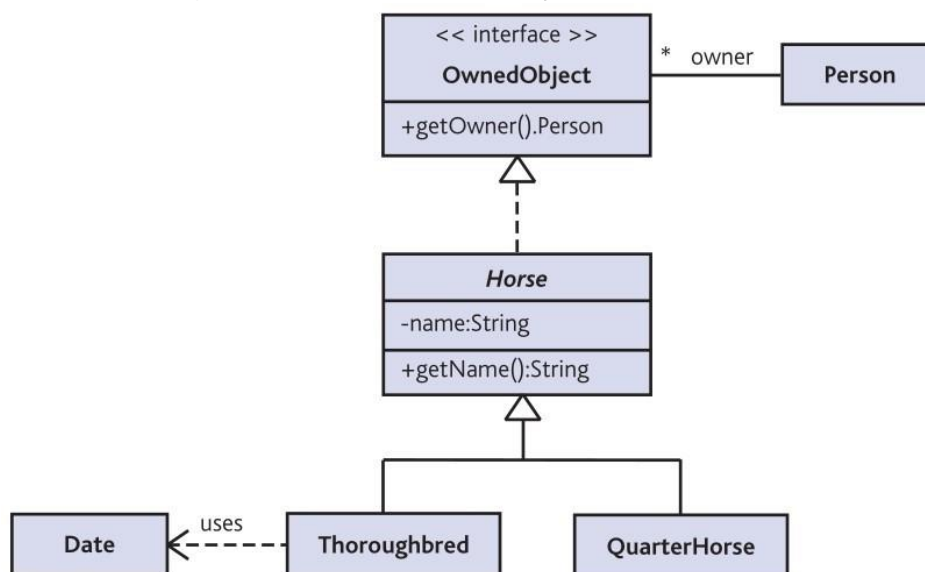
Figura 8 - Exemplo de diagrama de classe



Fonte: Pressman (2016)

Os diagramas de classe também podem exibir relações entre classes (Figura 9). Uma classe que seja subclasse de outra classe é conectada a ela por uma seta com uma linha cheia como eixo e com uma ponta triangular vazia. A seta aponta da subclasse para a superclasse. Em UML, uma relação como essa é chamada de generalização (PRESSMAN, 2016).

Figura 9 - Exemplo de relação entre classes



Fonte: Pressman (2016)

### 3.6.3 Diagrama lógico de banco de dados

O diagrama lógico, segundo Alves (2014), compreende a descrição das estruturas que serão armazenadas no banco de dados e que resulta numa representação gráfica dos dados de maneira lógica, inclusive já nomeando os componentes e as ações que exercem um sobre o outro.

Nessa etapa também se define a abordagem de banco de dados que será utilizada: hierárquica, de rede ou relacional. Veja o seguinte exemplo (ALVES, 2014):

Médico(CRM, NomeMedico ,Especialidade)

Paciente(CodigoPaciente, NomePaciente, Endereço)

Consulta(DataConsulta, HoraConsulta, CodigoPaciente, NomeMedico)

### 3.6.4 Sitemap

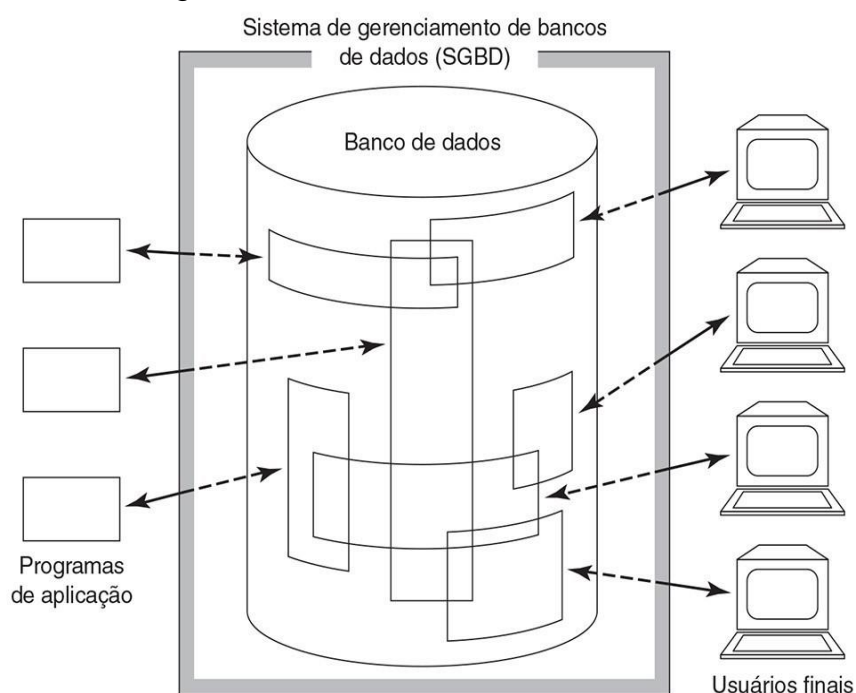
O Sitemap ou mapa do site é um ficheiro que inclui toda a informação do site: páginas, imagens, vídeos, outros ficheiros existentes no site e a correlação entre estes. Resumindo, este organiza a informação de um site num mesmo ficheiro para que os crawlers dos motores de busca tenham a tarefa mais facilitada e, por conseguinte, consigam melhores indexações. Além disso, também informa o motor de busca, como o Google, de quais são as páginas e ficheiros mais relevantes do site,

juntamente com alguma informação adicional, como as atualizações de um página, a frequência com que é alterada e, ainda, quaisquer versões em idiomas alternativos de uma mesma página (VENÂNCIO, 2021).

### 3.7 BANCO DE DADOS

Segundo Date (2004), um sistema de banco de dados (Figura 10) é basicamente um sistema computadorizado de manutenção de registros; em outras palavras, é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar. As informações em questão podem ser qualquer coisa que tenha algum significado ao indivíduo ou à organização a que o sistema deve servir – ou seja, qualquer coisa que seja necessária para auxiliar no processo geral das atividades desse indivíduo ou dessa organização.

Figura 10 - Sistema de banco de dados



Fonte: Date (2004)

É comum referir-se aos dados em um banco de dados como “persistentes” (embora, na verdade, não persistam por muito tempo!). Por persistente, queremos sugerir intuitivamente que os dados desse banco de dados diferem em espécie de outros dados mais efêmeros, como dados de entrada, dados de saída, filas de trabalho, blocos de controle de *software*, instruções SQL, resultados intermediários e,

de modo geral, quaisquer dados que tenham natureza transitória. Mais precisamente, dizemos que os dados no banco de dados “persistem” porque, uma vez aceitos pelo SGBD para entrada no banco de dados em primeiro lugar, eles só podem ser removidos do banco de dados mais tarde por alguma requisição explícita ao SGBD, e não como um mero efeito colateral de (por exemplo) algum programa concluindo sua execução (DATE, 2004).

### 3.7.1 MySQL

Uma empresa sueca chamada MySQL AB desenvolveu o MySQL em 1994. Então, a companhia norte-americana Sun Microsystems obteve controle total do software ao comprar a MySQL AB em 2008. Já em 2010, a gigante Oracle, também norte-americana, por sua vez comprou a Sun Microsystems, e o MySQL MySQL tem sido da Oracle desde então. Quanto a sua definição, MySQL é um Banco de Dados relacional (RDBMS – *Relational Database Management Systems*) com um modelo de cliente-servidor. RDBMS é um software de código aberto ou serviço usado na criação e gerenciamento de bancos de dados baseados no modelo relacional. Agora vamos analisar cada termo (LONGEN, 2021).

Banco de dados é um local onde dados são armazenados e gerenciados. A palavra “relacional” significa que os dados armazenados estão organizados em tabelas. Cada tabela está relacionada de alguma maneira. Caso o software não suporte o modelo relacional, então chamamos DBMS (LONGEN, 2021).

### 3.8 LARAVEL

É um *framework* gratuito, de código aberto, com suporte a recursos avançados e facilidade na construção do código, simples e legível: essas são as principais vantagens do Laravel, o que torna ele o framework preferido de vários desenvolvedores PHP (JOUKOVSKI, 2021).

Sua primeira versão foi lançada em 2011 como uma alternativa ao CodeIgniter, um dos frameworks mais populares da época. Tyler B. Otwell foi o desenvolvedor responsável e, desde então, manteve o conjunto de funções disponíveis em código aberto, recebendo atualizações constantes de melhoria pelas

mãos do criador e da comunidade até hoje. Outro ponto funcional é que o framework é baseado na arquitetura MVC, ou Model-View-Controller, que é um padrão de arquitetura de software para reutilização de código (JOUKOVSKI, 2021).

### 3.9 TRABALHOS RELACIONADOS

Todo mundo já passou pela experiência de precisar contratar um serviço e não saber por onde começar. No caso, pode ser desde um passeador de cachorro, professor particular, designer até encanador, cabeleireiro e por aí vai. Afinal, nem sempre é fácil achar um profissional e, mesmo se isso ocorrer, ainda há a questão de saber se o trabalho é bom ou não (BORRELLI, 2017).

Ao mesmo tempo, do outro lado da moeda, profissionais autônomos e freelancers também encontram dificuldades ao buscar clientes e serviços. É aqui que entra o GetNinjas, criado com o objetivo de unir ambas as necessidades. O *marketplace* atua tanto encontrando profissionais para o serviço que você deseja, quanto facilitando que você seja encontrado por clientes potenciais (BORRELLI, 2017).

Com seis anos de vida, a startup já conseguiu o lugar de maior classificado de serviços *on-line* do país, com cerca de 250 mil profissionais cadastrados na plataforma. Além disso, no último ano ela teve 2 milhões de serviços solicitados e gerou R\$ 300 milhões em transações realizadas (BORRELLI, 2017).

Inegavelmente a internet tem muito a oferecer para todos os tipos de negócio! Exemplo disso são as plataformas criadas nas últimas semanas para reunir pequenas empresas com a finalidade de anunciar e comercializar os seus serviços (PEIXOTO, 2020).

A Plataforma Eu Festa, por exemplo, cadastra gratuitamente prestadores de serviços do segmento de comemorações, desde doceiras, equipe de recreação, decoradores, fotógrafos e isso está para muito além de crise gerada pelo Coronavírus. Gerar praticidade a quem contrata e reduzir custos a quem anuncia representa um grande passo rumo a uma nova maneira de pensar os negócios no Brasil. Outro fator importante quanto à Eu Festa é que poderão anunciar tanto aqueles que já possuem CNPJ, como microempreendedores que ainda estão iniciando ramo, ainda informalmente! (PEIXOTO, 2020).



## 4 DESENVOLVIMENTO

Neste capítulo será apresentado o processo utilizado para o desenvolvimento do sistema web proposto, apresentando todo o processo de desenvolvimento dele, bem como os diagramas criados, métodos, técnicas, tecnologias e ferramentas utilizadas.

### 4.1 SISTEMA PROPOSTO

O sistema *web* Campo das Vertentes é uma plataforma on-line voltada para a divulgação de prestadores de serviços da microrregião Campos das Vertentes MG. Através do sistema as pessoas terão acesso a informações sobre os serviços, produtos e/ou projetos realizados por prestadores de serviços, bem como realizar contatos com os mesmos.

Por ser um Sistema de Informação (SI) baseado na arquitetura web, foi necessário a utilização de importantes técnicas e ferramentas. A tecnologia utilizada no desenvolvimento do sistema proposto foi baseada em código livre, com utilização das linguagens HTML, PHP e *JavaScript* (JS) e um banco de dados MYSQL, além do uso do padrão de projeto MVC e a utilização de frameworks, tais como Laravel e Tailwindcss. Para o desenvolvimento do sistema foi utilizado Visual Studio Code para a codificação.

### 4.2 ANÁLISE, LEVANTAMENTO E MODELAGEM DOS REQUISITOS

Diante da proposta de desenvolvimento de um sistema para a divulgação de prestadores de serviços da microrregião Campos das Vertentes - MG foi realizado uma análise de sistemas correlacionados ao tema já existentes, bem como outras formas de divulgação, tais como jornais regionais, mídias sociais, revistas, entre outros. Após a análise, foi realizada a coleta de requisitos, sendo possível identificar os requisitos funcionais e os não funcionais necessários para o desenvolvimento do sistema proposto. Os requisitos funcionais do sistema, apresentado no Quadro 1, foram divididos em duas categorias, a categoria comum a todos os usuários, a categoria específica para os usuários prestadores de serviços.

Quadro 1 - Requisitos funcionais do sistema

Identificação	Requisito	Prioridade
<b>Área comum a todos os usuários</b>		
RF01	Visualizar perfil dos prestadores de serviços	Alta
RF02	Visualizar serviços postados	Alta
RF03	Enviar mensagem para os profissionais	Média
RF04	Fazer pesquisas	Alta
RF05	Entrar em contato com suporte	Alta
<b>Área prestadores de serviço</b>		
RF05	Realizar cadastro	Alta
RF06	Realizar login	Alta
RF07	Adicionar dados pessoais	Alta
RF08	Editar dados pessoais	Alta
RF09	Adicionar serviços prestados	Alta
RF10	Remover serviços prestados	Alta
RF11	Visualizar mensagens recebidas	Média
RF12	Excluir conta	Alta

Fonte: Elaborado pelo autor (2022)

A fim de obter um melhor entendimento a respeito do uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e das tecnologias que seriam utilizadas durante o desenvolvimento do projeto, foi necessário realizar o levantamento dos requisitos não funcionais. O Quadro 2 apresenta todos os requisitos não funcionais identificados na etapa de análise e coleta de requisitos.

Quadro 2 - Requisitos não funcionais do sistema

Identificação	Requisito	Prioridade
RNF01	Desenvolvimento com HTML, CSS, JS e PHP	Alta
RNF02	Comunicação com o banco de dados MySQL	Alta
RNF03	Utilizar padrão MVC	Alta
RNF04	Possuir máscara de <i>inputs</i>	Baixa
RNF05	Ser compatível com diversos navegadores	Alta
RNF06	Possuir design responsivo	Alta
RNF07	Navegação e <i>layout</i> simples	Alta
RNF08	Agilidade	Alta

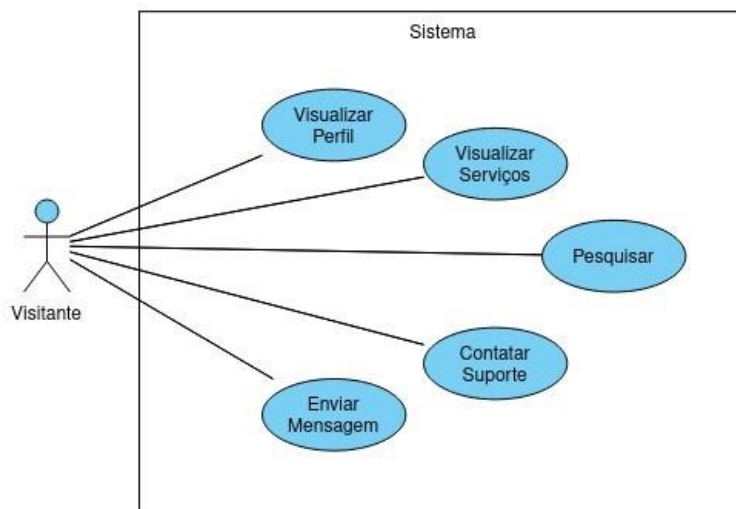
Fonte: Elaborado pelo autor (2022)

A etapa de Análise e levantamento de requisitos possibilitou que fosse realizado a modelagem dos mesmos através da criação de diagramas da UML, visando obter um melhor entendimento do funcionamento do sistema e desenvolvê-lo de forma completa e funcional.

#### 4.2.1 Diagrama de caso de uso

As Figuras 11 e 12 mostram os diagramas de caso de uso que foram utilizados para a modelagem do sistema. A Figura 11 apresenta como usuário do sistema o visitante, que representa qualquer pessoa que acesse o site, seu acesso é limitado a visualização dos prestadores de serviços, visualizar os serviços cadastrados, bem como a realização de contatos com os prestadores de serviços e contatar o suporte técnico do site.

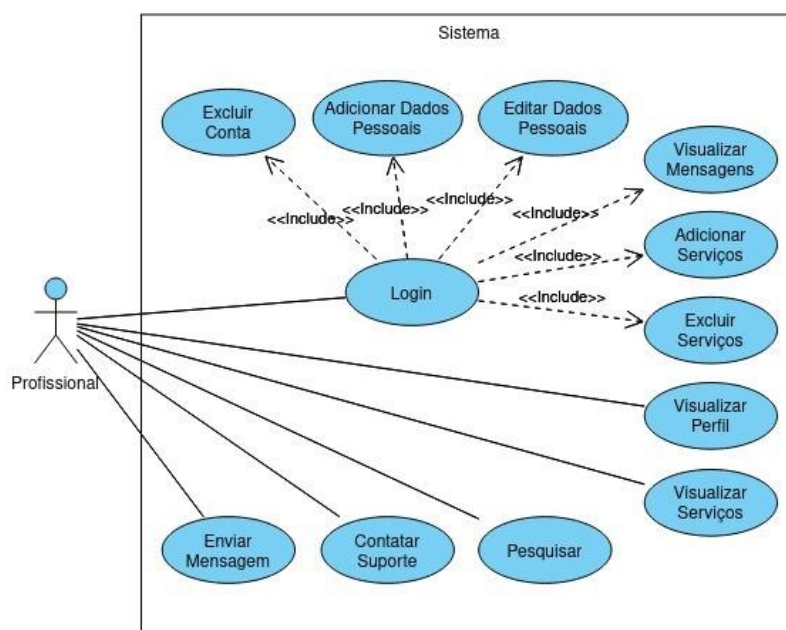
Figura 11 - Diagrama de caso de uso para o visitante



Fonte: Elaborado pelo autor (2022)

A Figura 12 apresenta como usuário o profissional (prestador de serviço) que possui cadastro no sistema, este além de possuir as mesmas funcionalidades do visitante, conta com ferramentas para adicionar e remover seus conteúdos do site e a visualização de mensagens recebidas.

Figura 12 - Diagrama de caso de uso para o profissional

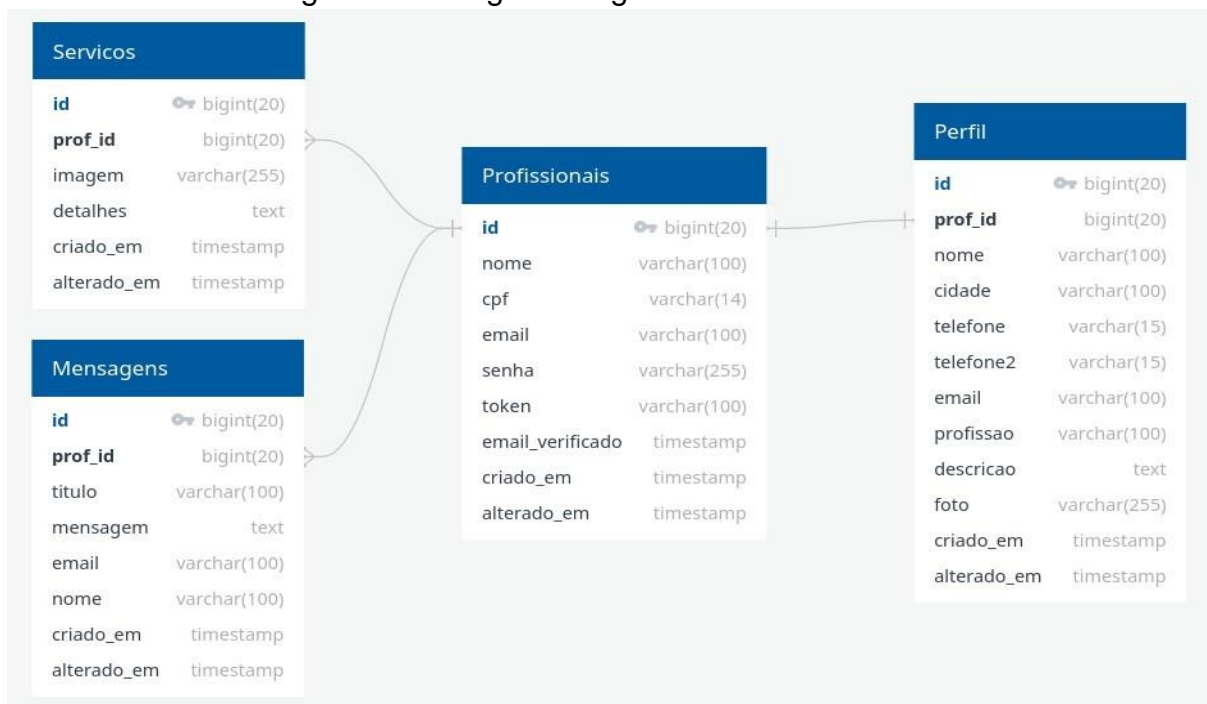


Fonte: Elaborado pelo autor (2022)

#### 4.2.2 Diagrama lógico de banco de dados

O diagrama lógico de banco de dados apresentado na Figura 13 apresenta as tabelas, bem como suas colunas (nome e tipo), chaves (primárias e secundárias), e relações entre as mesmas.

Figura 13 - Diagrama lógico do banco de dados

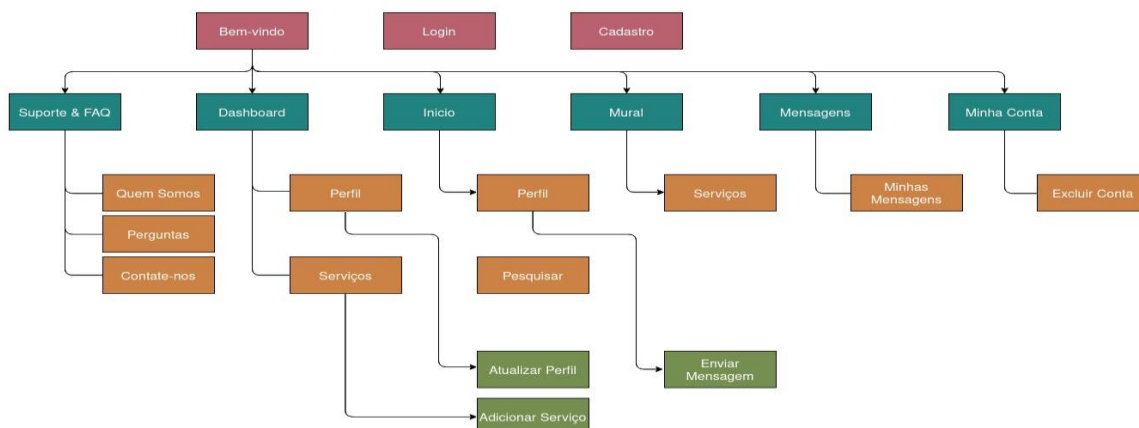


Fonte: Elaborado pelo autor (2022)

#### 4.2.3 Sitemap

A Figura 14 apresenta o *sitemap* do sistema proposto, nele estão contidas informações que mostram o caminho correto a ser percorrido pelos usuários do site.

Figura 14 - Sitemap do site



Fonte: Elaborado pelo autor (2022)

### 4.3 O SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Todos os dados do sistema são armazenados e manipulados através de um sistema de gerenciamento de banco de dados relacional MySQL, utilizando linguagem não estruturada.

A criação do banco de dados intitulado “campo” foi feita através de linha de comando pelo terminal, assim como alterações necessárias ou qualquer outro tipo de acesso. Já a criação das tabelas que o compõe foi feita por intermédio do *framework* PHP de código aberto Laravel o qual através da função “migrate”<sup>1</sup> construiu as tabelas da forma com que foram especificadas dentro de arquivos PHP específicos; a mesma função também permite atualizar as tabelas caso haja alguma alteração necessária.

<sup>1</sup> A interface Laravel Schema fornece suporte agnóstico de base de dados para a criação e manipulação de tabelas em todos os sistemas de base de dados suportados pelo Laravel. Normalmente, as migrações utilizarão esta interface para criar e modificar tabelas e colunas de bases de dados.

Figura 15 - Exemplo de código para função migrate

```
return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('messages', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->constrained();
            $table->string('title');
            $table->text('message');
            $table->string('email', 100);
            $table->string('name', 100);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('messages');
    }
};
```

Fonte: Elaborado pelo autor (2022)

#### 4.4 DESENVOLVIMENTO DO SISTEMA

Para o desenvolvimento do sistema foi utilizada linguagem de marcação HTML para a estruturação das páginas e a estilização foi feita com folha de estilo em cascata (CSS), mais especificamente através de um *framework* gratuito chamado *Tailwindcss*, o qual fornece classes pré-definidas de forma dinâmica à medida que o programador as chama no HTML, fazendo com que o arquivo “app.css” contenha apenas as classes utilizadas, reduzindo assim o tamanho total do arquivo.

O padrão de projeto utilizado foi o MVC, que aliás, já compõe por padrão a estrutura do *framework* Laravel, ajudando a manter os arquivos bem organizados e suas funções bem definidas. Quanto ao *JavaScript*, este foi utilizado na criação de máscaras para os *inputs* através do *Cleave*, uma ferramenta gratuita e de código

aberto que fornece diversas máscaras, além de permitir a customização das mesmas, este pode ser encontrado no repositório do Github.

Na Figura 16 é apresentado a estrutura do projeto “Campo”. Sua estrutura segue exatamente o formato fornecido pelo Laravel, não tendo sido alterada em nada.

Dentro da pasta “*app*” são encontrados os *Controllers* e os *Models* da aplicação, as *Views* serão encontradas na pasta “*resources*”, assim como o CSS e o JS usado, porém estes dois não serão utilizados quando alguém acessar a página, e sim suas cópias que estarão na pasta “*public*”. Os arquivos que permitem a criação das tabelas do banco de dados se localizam dentro da pasta “*database*”.

As imagens que forem enviadas pelos usuários ficaram na pasta “*storage*”, a qual não pode ser acessada por visitantes, logo um *link* simbólico é criado na pasta “*public*” para que os mesmo possam ter acesso tanto as imagens enviadas, como de ícones que compõem as páginas.

Em “*routes*” ficam as funções que indicam para onde o usuário deve ser direcionado quando um determinado *link* é acessado, em casos simples há uma função dentro do próprio arquivo que retorna uma *View*, e em outros casos é chamada uma função dentro do *Controller*.

Figura 16 - Sistema de arquivos do projeto



Fonte: Elaborado pelo autor (2022)



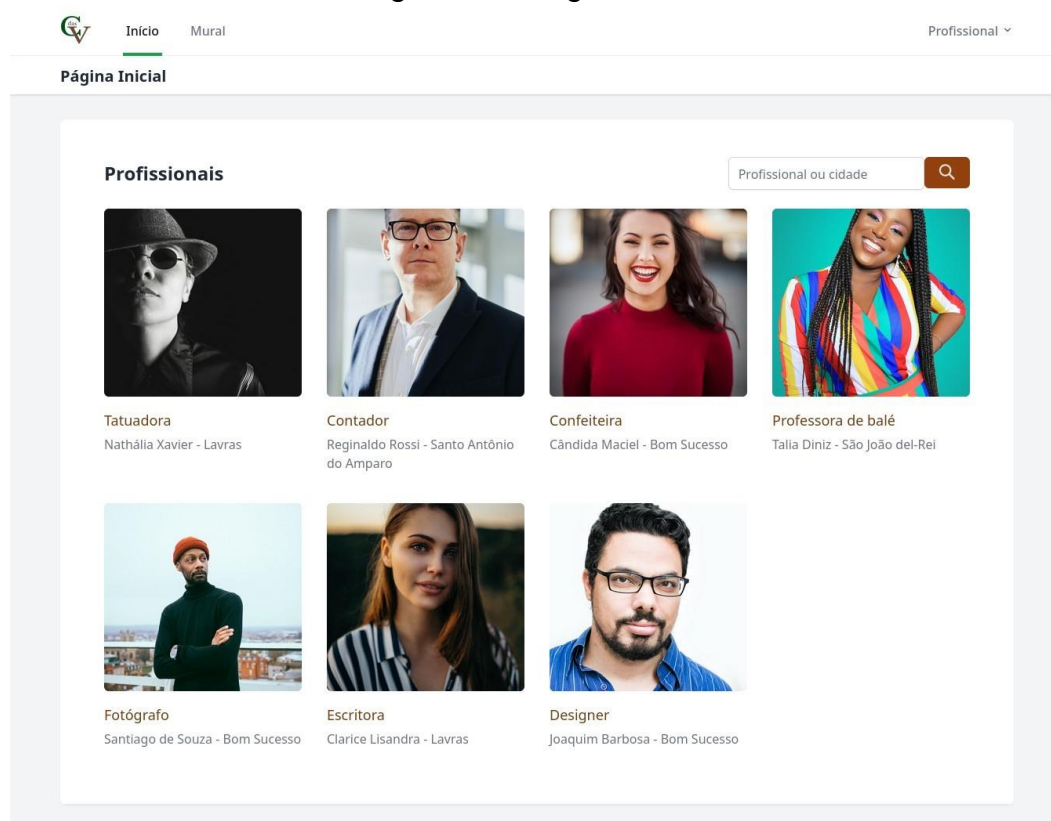
## 5 RESULTADOS

A finalidade deste capítulo é apresentar os resultados obtidos a partir do desenvolvimento do sistema web Campo das Vertentes.

### 5.1 MURAL DIVULGAÇÃO DE PRESTADORES DE SERVIÇOS: PÁGINA PRINCIPAL

A tela inicial (Figura 17) fará a apresentação de todos os prestadores de serviços da plataforma ordenando pelos últimos a serem cadastrados. Será exibido primeiramente a profissão do mesmo, seguido por seu nome e sua cidade. No canto superior direito o usuário tem a possibilidade de pesquisar por um profissional específico, ou por uma cidade.

Figura 17 - Página Início




© 2022 - Campo das Vertentes. Todos os direitos reservados.  
[Suporte & Perguntas frequentes](#)

Fonte: Elaborado pelo autor (2022)

Após um profissional ser selecionado, o usuário será direcionado para a página pessoal do profissional (Figura 18), onde será apresentado todos os seus


dados para contato, junto de todos os serviços que o mesmo já postou na plataforma. Caso o usuário deseje, poderá entrar em contato diretamente com o profissional através da plataforma deixando uma mensagem direta para ele.

Figura 18 - Página de visualização do profissional


Início Mural
Profissional ▾

Vizualizar perfil

### Perfil



**Nome**  
Santiago de Souza

**Cidade**  
Bom Sucesso

**Telefone**  
(35) 13212-1321


**E-mail**  
vbn@vbn.com

**Profissão**  
Fotógrafo

**Descrição**  
Nec dui nunc mattis enim. Blandit turpis cursus in hac habitasse platea dictumst quisque. Purus in massa tempor nec feugiat nisl. Massa id neque aliquam vestibulum morbi blandit cursus risus at.


[ENTRAR EM CONTATO](#)

### Serviços




Felis eget velit aliquet sagittis id consectetur purus. Sed augue lacus viverra vitae congue eu consequat ac felis.

**RS 49,90**



Gravida in fermentum et sollicitudin ac orci.

**RS 49,90**



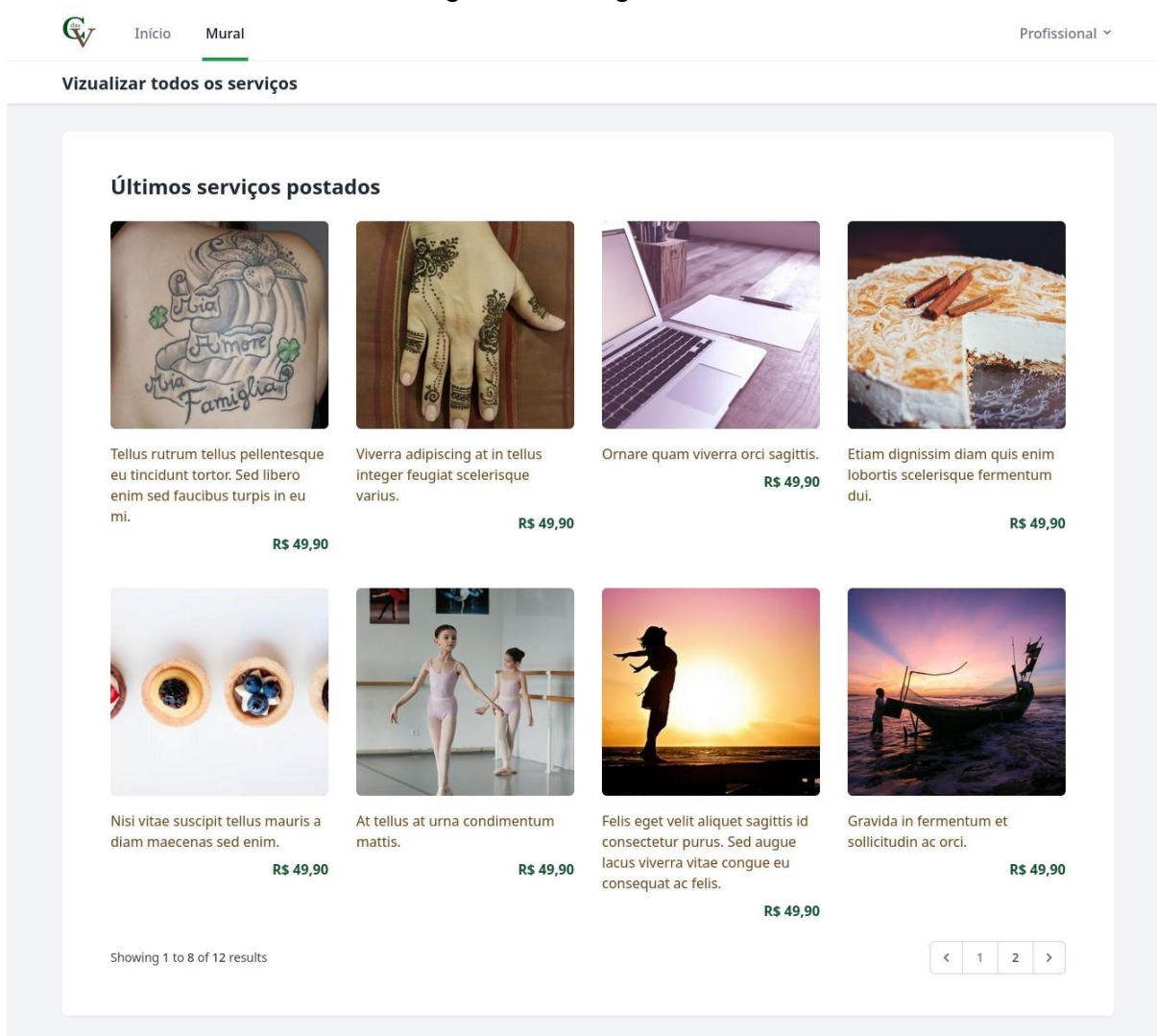
Diam quis enim lobortis.

**RS 49,90**

## 5.2 MURAL DIVULGAÇÃO DE SERVIÇOS PRESTADOS

A Figura 19 apresenta a página onde serão exibidos todos os serviços cadastrados na plataforma e, assim como com os profissionais, os serviços também são exibidos por ordem decrescente à data de publicação. Ao selecionar um serviço o usuário será redirecionado para a página do profissional ao qual aquela postagem pertence.

Figura 19 - Página Mural



© 2022 - Campo das Vertentes. Todos os direitos reservados.  
[Suporte & Perguntas frequentes](#)

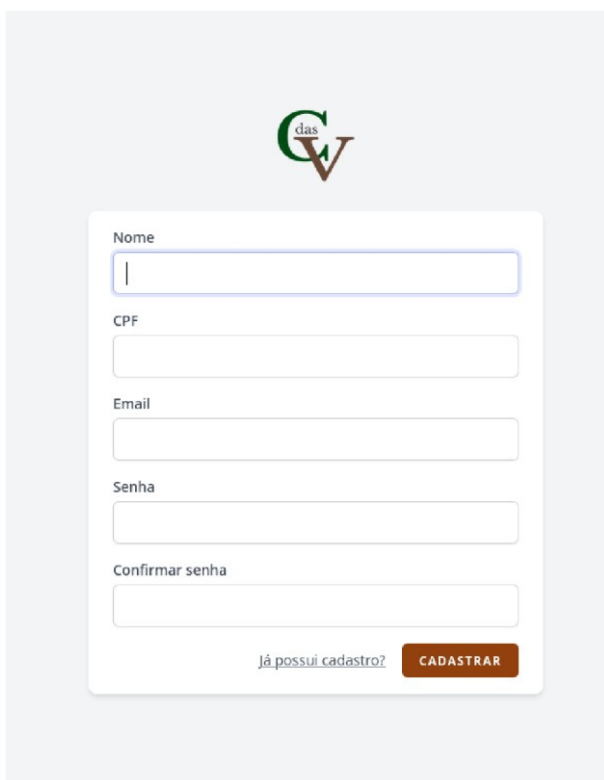
Fonte: Elaborado pelo autor (2022)

## 5.3 CADASTRO DE PRESTADORES DE SERVIÇOS

As Figura 20 e 21 mostram como é feito o cadastro de um profissional na plataforma. Primeiro o usuário faz o cadastro para entrar na plataforma, este contendo

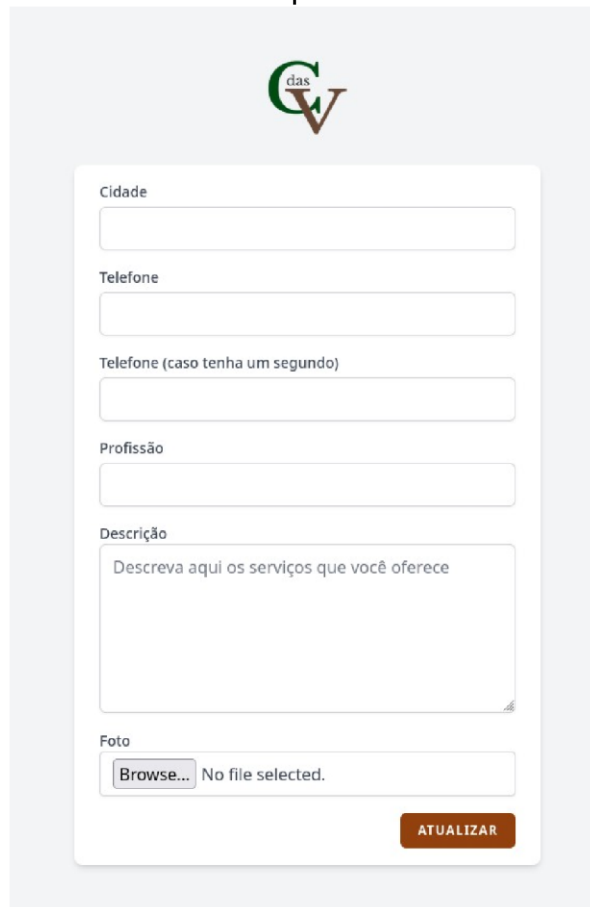
nome, cpf, e-mail e senha. Após acessar o site pela primeira vez com sua conta ele irá se deparar com um perfil em branco e terá a opção de cadastrar seus dados pessoais como cidade, telefones, profissão, uma descrição mais completa dos serviços que serão prestados por ele, e uma foto de si mesmo.

Figura 20 - Página para cadastro do profissional

The screenshot shows a registration form for a professional. At the top center is a logo with the letters 'das' in a green circle and 'V' in a brown circle. Below the logo is a white form with the following fields: 'Nome' (text input), 'CPF' (text input), 'Email' (text input), 'Senha' (password input), and 'Confirmar senha' (password input). At the bottom left of the form is a link that says 'já possui cadastro?' and at the bottom right is a brown button labeled 'CADASTRAR'.

Fonte: Elaborado pelo autor (2022)

Figura 21 - Página para cadastro dos dados pessoais

The screenshot shows a form for updating personal data. At the top center is the same 'das V' logo. The form contains the following fields: 'Cidade' (text input), 'Telefone' (text input), 'Telefone (caso tenha um segundo)' (text input), 'Profissão' (text input), and 'Descrição' (a larger text area with the placeholder text 'Descreva aqui os serviços que você oferece'). At the bottom is a 'Foto' section with a 'Browse...' button and the text 'No file selected.'. A brown button labeled 'ATUALIZAR' is located at the bottom right of the form.

Fonte: Elaborado pelo autor (2022)

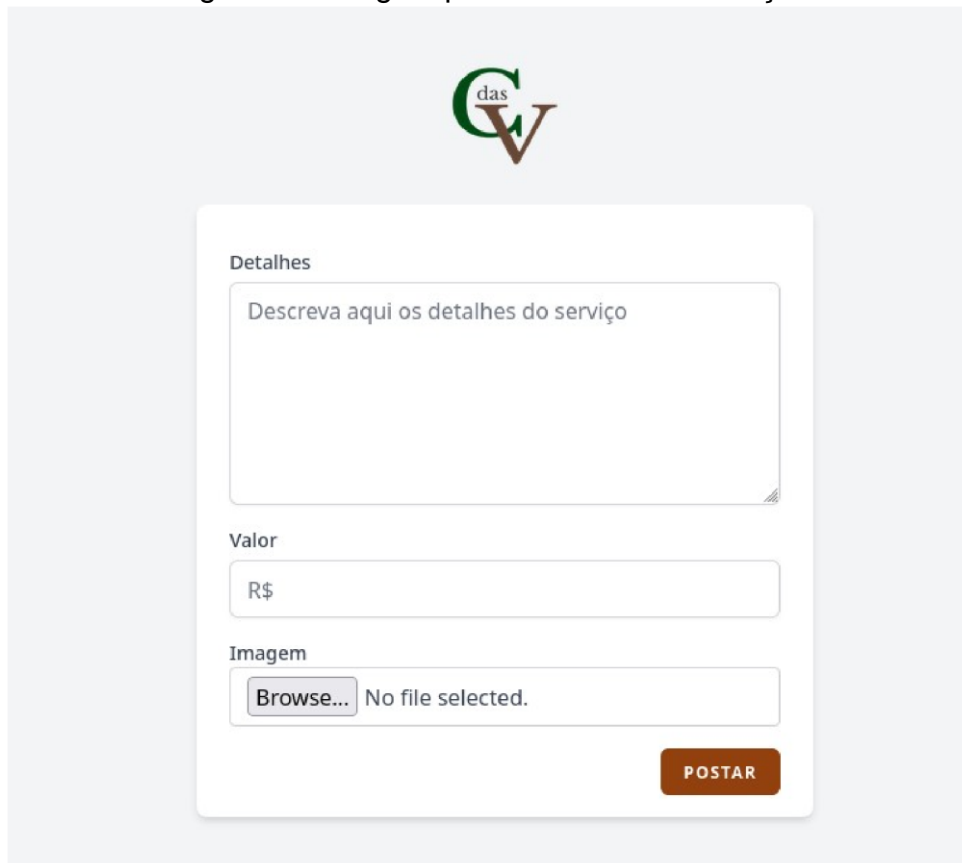
## 5.4 MÓDULO PRESTADOR DE SERVIÇOS

### 5.4.1 Cadastro de serviços

A Figura 22 apresenta o formulário a ser preenchido durante a postagem de um serviço, este contém uma área para que o profissional detalhe o que fora feito, o processo, material, ou quaisquer detalhe que ele julgue relevante para o compartilhamento. Uma segunda área para a atribuição do valor daquele serviço permitindo assim que os visitantes da página tenham uma noção do preço médio dos

serviços, sem a necessidade de entrar em contato apenas para obter essa informação. E uma imagem para que os visitantes possam ver o resultado daquele trabalho.

Figura 22 - Página para cadastro de serviços



Detalhes

Descreva aqui os detalhes do serviço

Valor

R\$

Imagem

Browse... No file selected.

POSTAR

Fonte: Elaborado pelo autor (2022)

#### 5.4.2 Visualizar mensagens

O profissional pode receber mensagens diretas dentro da plataforma, como apresentado na Figura 23. As mensagens recebidas são acompanhadas do nome e do e-mail de quem as enviou, após ler as mensagens o profissional pode apagá-las.

Figura 23 - Página para visualização de mensagens

The screenshot displays a web interface for viewing messages. At the top, there is a navigation bar with a logo on the left, links for 'Dashboard', 'Início', and 'Mural', and the user's name 'Cândida Maciel' on the right. Below this, the main heading is 'Mensagens'. A sub-section titled 'Minhas mensagens' contains three message items, each in a light gray box. The first message is from 'Camila Orleans e Bragança' (ggg@mail.com) with the subject 'Que bolos lindos' and a body of Lorem Ipsum text, followed by an 'EXCLUIR MENSAGEM' button. The second message is from 'Billie Jean' (rrr@mail.com) with the subject 'Um teste' and another body of Lorem Ipsum text, also with an 'EXCLUIR MENSAGEM' button. The third message is from 'Samantha Sanderson' (ddd@mail.com) with the subject 'Preços e descontos' and a body of Lorem Ipsum text, with an 'EXCLUIR MENSAGEM' button. At the bottom of the page, there is a footer with the text '© 2022 - Campo das Vertentes. Todos os direitos reservados.' and a link for 'Suporte & Perguntas frequentes'.

© 2022 - Campo das Vertentes. Todos os direitos reservados.  
[Suporte & Perguntas frequentes](#)

Fonte: Elaborado pelo autor (2022)

## 6 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo geral deste trabalho era propor uma plataforma *on-line* para a divulgação de prestação de serviços da microrregião Campos das Vertentes - MG, bem como apresentar todas as etapas realizadas para o seu desenvolvimento. Acredita-se que o sistema apresentado poderá atender ao que fora proposto, apresentando-se como uma ferramenta simples e eficaz para a divulgação de prestadores de serviços, oferecendo maior visibilidade e interação entre possíveis novos clientes.

A utilização dos frameworks (Laravel e Tailwindcss) e padrão de projeto MVC durante o desenvolvimento do sistema foi extremamente importante, facilitando a implementação de todas as funcionalidades propostas, tornando seu código bem estruturado e eficiente, facilitando futuras alterações, bem como tornando-o bem estruturado e eficiente facilitando sua manutenção.

Para trabalhos futuros pretende-se implementar um método para a avaliação dos profissionais, sendo exibidas em forma de estrelas, que serão calculadas sobre a pontuação média daquele profissional. Como os clientes não se cadastraram na plataforma, para que a avaliação de alguém seja contabilizada no sistema, esta pessoa receberá um e-mail de confirmação no momento da avaliação.

Junto da avaliação dos prestadores de serviços, também poderá ser implementado uma forma de pesquisa mais completa, permitindo que na página inicial seja pesquisado não apenas por profissão e cidade, como permitirá filtro por aqueles melhor avaliados.

## REFERÊNCIAS

ALVES, William P. **Banco de dados**. São Paulo: Érica, 2014. *E-book*. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788536518961/>. Acesso em: 18 mai. 2022.

BORRELLI, Isabela. Estudo de caso: GetNinjas, o marketplace que acha o profissional ideal para você. **StartSe**, 2017. Disponível em: <https://www.startse.com/noticia/conteudo-memoravel/estudo-de-caso-getninjas/>. Acesso em: 17 jul. 2022.

DALEPIANE, Filipe. DAO Pattern: persistência de dados utilizando o padrão DAO. **DevMedia**, 2014. Disponível em: <https://www.devmedia.com.br/dao-patternpersistencia-de-dados-utilizando-o-padrao-dao/30999>. Acesso em: 17 maio 2022.

DATE, C.J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Elsevier, 2003. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788595154322/>. Acesso em: 12 maio 2022.

FALBO, Ricardo de Almeida. A experiência na definição de um processo padrão baseado no processo unificado. Anais do II SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE, SIMPROS, 2., 2000. **Anais [...]** v. 200, p. 63-74, 2000.

INTERNATIONAL TELECOMMUNICATION UNION. **Measuring digital development**. Geneva: ITU, 2021. Disponível em: <https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2021.pdf>. Acesso em: 22 fev. 2022.

JOUKOVSKI, R. V. Cezar. O que é Laravel? Conheça o framework de PHP mais utilizado. **Tecmundo**, 2021. Disponível em: <https://www.tecmundo.com.br/software/223718-laravel-conheca-o-framework-phputilizado.htm>. Acesso em: 17 maio 2022.

LEITE, Alessandro. Conheça os padrões de projeto. **DevMedia**, 2005. Disponível em <https://www.devmedia.com.br/conheca-os-padroes-de-projeto/957>. Acesso em: 28 fev. 2022.

LONGEN, Andrei. O que é MYSQL? Guia para iniciantes. **Hostinger**, 2021. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-mysql>. Acesso em: 17 maio 2022.

NADER, Danielle. Número de trabalhadores autônomos bate recorde em 2021. **Contábeis**, 2021. Disponível em: <https://www.contabeis.com.br/noticias/48632/numero-de-trabalhadores-autonomosbate-recorde-em-2021/>. Acesso em: 22 fev. 2022.



NETCRAFT. **April 2019 Web Server Survey**. 2019. Disponível em: <https://news.netcraft.com/archives/2019/04/22/april-2019-web-server-survey.html>. Acesso em 22 fev. 2022.

NOGUEIRA, Luiz. Dados mostram o crescimento impressionante da internet em 10 anos. **Olhar Digital**, 2019. Disponível em: <https://olhardigital.com.br/2019/05/17/noticias/dados-mostram-o-crescimentoimpressionante-da-internet-em-10-anos/>. Acesso em: 22 fev. 2022.

PEIXOTO, Bruna. Como a internet pode ajudar os prestadores de serviços durante a crise?. **SigeCloud**, 2020. Disponível em: <https://blog.sigecloud.com.br/como-a-internet-pode-ajudar-os-prestadores-deservicos-durante-a-crise/>. Acesso em: 17 jul. 2022.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software**. Trad. João Eduardo Nóbrega Tortello. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. Trad. Ivan Bosnic e Kalinka G. 9a ed. São Paulo: Pearson Prentice Hall, 2011.

COMITÊ GESTOR DA INTERNET NO BRASIL. **TIC domicílios**: pesquisa sobre o uso das tecnologias de informação e comunicação nos domicílios brasileiros. São Paulo: Comitê Gestor da Internet no Brasil, 2021. Disponível em: [https://cetic.br/media/docs/publicacoes/2/20211124201233/tic\\_domicilios\\_2020\\_livro\\_eletronico.pdf](https://cetic.br/media/docs/publicacoes/2/20211124201233/tic_domicilios_2020_livro_eletronico.pdf). Acesso em: 22 fev. 2022.

VENÂNCIO, Rafael. Afinal que é um Sitemap?. **Van**, 2021. Disponível em: <https://van.pt/afinal-o-que-e-um-sitemap/>. Acesso em: 17 maio 2022.