

**INSTITUTO FEDERAL
SANTA CATARINA**

**CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA
TECNOLOGIA DA INFORMAÇÃO**

ALISSON DA SILVA

Soluções de aplicação web para simulação de jardim vertical

Florianópolis - SC

2021

**INSTITUTO FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA
INFORMAÇÃO**

ALISSON DA SILVA

**PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO:
SOLUÇÕES DE APLICAÇÃO WEB PARA SIMULAÇÃO DE JARDIM
VERTICAL.**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Tecnólogo em Gestão da Tecnologia da Informação.

Professor Orientador:
Prof. Herval Daminelli, Esp.

Professor Coorientador:
Prof. Júlio César da Costa Ribas, Dr.

Florianópolis – SC

2021

Ficha de identificação

Silva, Alisson da
Soluções de aplicação web para simulação de jardim vertical /
Alisson da Silva;
Orientação de Herval Daminelli - Florianópolis, SC, 2021.
90p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de
Educação, Ciência e Tecnologia de Santa Catarina, Campus
Florianópolis. CST em Gestão da TI. Departamento Acadêmico de
Saúde e Serviços.
Inclui Referências.

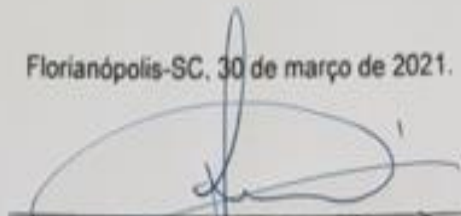
1. Aplicação Web 2. Paisagismo 3. Simulação de jardins verticais
I. Daminelli, Herval II. Instituto Federal de Educação, Ciência e
Tecnologia de Santa Catarina. Departamento Acadêmico de Saúde e
Serviços III. Soluções de aplicação web para simulação de jardim
vertical.

SOLUÇÕES DE APLICAÇÃO WEB PARA SIMULAÇÃO DE JARDIM
VERTICAL.

ALISSON DA SILVA

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis-SC, 30 de março de 2021.

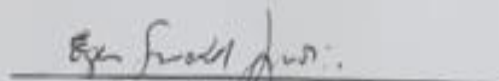


Prof. Felipe Cantório Soares, Me.
Coordenador do CST em Gestão da Tecnologia da Informação
Instituto Federal de Santa Catarina

Banca Examinadora:



Prof. Herval Daminelli, Esp.
Orientador
Instituto Federal de Santa Catarina



Egon Sewald Junior, Dr.
Instituto Federal de Santa Catarina



Felipe Cantório Soares, Me.
Instituto Federal de Santa Catarina

Dedico este trabalho à minha família, em especial, à minha esposa, Jaqueline, pelas palavras de inspiração e incentivo durante toda minha trajetória, e nossa filha, Isabela, que veio trazer luz e sentido à nossa vida.

AGRADECIMENTOS

Agradeço, primeiramente aos meus familiares e amigos, que sempre me apoiaram durante esta jornada; aos meus professores, que no decorrer destes anos ajudaram a construir uma base sólida de conhecimento; minha esposa Jaqueline pela paciência e incentivo; em especial, ao meu orientador professor Herval Daminelli, por toda dedicação, conhecimento e compreensão, todos essenciais para realização de tal projeto.

RESUMO

O presente trabalho apresenta um estudo sobre as diversas soluções disponíveis de aplicação *web* para simulação de jardim vertical. Tem como objetivo agregar à lacuna de ferramentas gráficas atualmente existentes para simulação de projetos de paisagismo, em especial os projetos de jardins verticais. O trabalho utiliza pesquisa bibliográfica sobre as diversas aplicações disponíveis, com um estudo aprofundado de cada ferramenta e suas particularidades, dissecando seus recursos e analisando seu potencial para uso em apresentação e simulação de jardim vertical. Ao mesmo tempo, busca tecnologia e conceitos já validados em projetos *web*, bem como a fundamentação necessária para modelar e desenvolver um protótipo *web* que atenda os requisitos propostos pelo projeto.

Palavras-chave: Aplicação *Web*. Paisagismo. Simulação de jardins verticais.

ABSTRACT

The present work presents a study on the different available web application solutions for vertical garden simulation. It aims to add to the gap of graphic tools currently existing for simulation of landscaping projects, in particular vertical garden projects. The work uses bibliographic research on the various applications available, with an in-depth study of each tool and its particularities, dissecting its resources and analyzing its potential for use in presentation and simulation of vertical garden, at the same time, it seeks technology and concepts already validated in web projects, as well as the necessary foundation to model and develop a web prototype that meets the requirements proposed by the project.

Keywords: Web application. Landscaping. Simulation of vertical gardens.

LISTA DE ILUSTRAÇÕES

Figura 1 - Representação gráfica de projeto de jardim vertical	18
Figura 2 - Representação dos jardins suspensos da Babilônia.....	21
Figura 3 – Jardim romano	22
Figura 4 - Jardim oriental	23
Figura 5 - Jardins de Versalhes.....	24
Figura 6 - Sofitel Palm Jumeirah, Dubai.....	25
Figura 7 - Caixa Forum, Madrid.....	25
Figura 8 - Esboço manual de projeto concluído em Paris por Patrick Blanc.	27
Figura 9 - Esboço manual e projeto concluído em Paris por Patrick Blanc.	27
Figura 10 - Imagem exemplo de software de paisagismo CAD.....	28
Figura 11 - Técnica de fotomontagem para representação de jardins.	29
Figura 12 - Exemplos de jardins verticais.....	30
Figura 13 - Fachada verde com trepadeiras se desenvolvendo no muro.....	31
Figura 14 - Fachada Verde com suporte de treliças instalada em Vancouver.	32
Figura 15 - Parede viva com sistema de painéis geotêxteis modulares.....	33
Figura 16 - Sistema aberto de irrigação automatizada.....	34
Figura 17 - Sistema fechado de fertirrigação automatizada.	35
Figura 18 - Luxímetro.	37
Figura 19 - Diagramas <i>UML</i>	43
Figura 20 - Estrutura de uma página <i>HTML5</i>	45
Figura 21 - Exemplo de estrutura de componentes Angular.	50
Figura 22 - Estrutura do <i>DOM</i>	51
Figura 23 - Representação de um SGBD.....	54
Figura 24 - Apresentação de projeto com DynaScape Design e Horticopia Pro.	61
Figura 25 - Navegando com realidade virtual no projeto com Edificius Land.	63
Figura 26 - Exemplo de projeto criado com PhotoLANDSCAPE.....	64
Figura 27 - Exemplo de utilização com o aplicativo FloraMe.	66
Figura 28 - Diagrama de casos de uso da aplicação.	71
Figura 29 - Arquivo de configuração Wamp Server/Apache.	79

LISTA DE TABELAS

Tabela 1 – Documentação do caso de uso “Visualizar catálogo de plantas”.....	72
Tabela 2 – Documentação do caso de uso “Visualizar projetos executados”.	73
Tabela 3 – Documentação do caso de uso “Fazer upload de imagem para cenário de simulação”.....	74
Tabela 4 – Documentação do caso de uso “Selecionar, arrastar e soltar plantas sobre o cenário”.	75
Tabela 5 – Documentação do caso de uso “Salvar imagem da simulação”.....	76

LISTA DE ABREVIATURAS

ASP	<i>Active Server Pages</i> (Páginas do servidor ativo)
CAD	<i>Computer-Aided Design</i> (Desenho Assistido por Computador)
CERN	<i>European Council for Nuclear Research</i> (Conselho Europeu de Pesquisa Nuclear)
CSS	<i>Cascading Style Sheets</i> (Folhas de Estilo em Cascata)
DOM	<i>Document Object Model</i> (Modelo de Documento por Objetos)
HTML	<i>Hyper Text Markup Language</i> (Linguagem de marcação de hipertexto)
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
JS	<i>JavaScript</i>
MIT	<i>Massachusetts Institute of Technology</i>
MVC	<i>Model-View-Controller</i>
NoSQL	<i>Not Only SQL</i>
OMT	<i>Object Modelling Technique</i> (Técnica de modelagem de objetos)
PHP	<i>Hypertext Preprocessor</i> (Pré-processador de hipertexto)
PPC	Plano Pedagógico do Curso
SGBD	Sistemas de Gestão de Base de Dados
SQL	<i>Structured QueryLanguage</i> (Linguagem de Consulta Estruturada)
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i> (Linguagem Unificada de Modelagem)

SUMÁRIO

1. INTRODUÇÃO	15
1.1. JUSTIFICATIVA	16
1.2. DEFINIÇÃO DO PROBLEMA	18
1.3. OBJETIVOS.....	19
1.3.1. Objetivo geral	19
1.3.2. Objetivos específicos	19
1.4. ESTRUTURA DO TRABALHO	19
2. JARDINS VERTICAIS E A ENGENHARIA DE SOFTWARE	21
2.1. HISTÓRIA E EVOLUÇÃO DOS JARDINS.....	21
2.2. TECNOLOGIAS DE SOFTWARE EM PROJETOS DE JARDINS VERTICAIS	26
2.2.1. Desenho manual.....	26
2.2.2. Softwares CAD.....	28
2.2.3. Fotomontagem	29
2.3. JARDINS VERTICAIS, SEUS COMPONENTES E MANUTENÇÃO.....	30
2.3.1. Fachadas Verdes.....	30
2.3.2. Paredes Vivas	32
2.3.3. Irrigação	33
2.3.4. Nutrição e variedades de plantas.....	35
2.3.5 Iluminação.....	36
2.4. ENGENHARIA DE SOFTWARE	37
2.4.1. Engenharia de requisitos.....	38
2.4.2. Requisitos funcionais	39
2.4.3. Requisitos não funcionais	39
2.4.4. Requisitos de domínio	40
2.4.5. Levantamento de requisitos.....	41
2.4.6. Modelagem <i>UML</i>	42
2.5. TECNOLOGIAS DE DESENVOLVIMENTO WEB	43
2.5.1. <i>HTML</i>	44
2.5.2. <i>CSS</i>	45
2.5.3. JavaScript	45
2.5.4. <i>PHP</i>	46

2.6. FRAMEWORKS DE DESENVOLVIMENTO	47
2.6.1. Bootstrap	48
2.6.2. <i>Angular</i>	49
2.6.3. <i>React</i>	50
2.6.4. <i>CakePHP</i>	52
2.7. BANCO DE DADOS	53
2.7.1. Sistemas de Gestão de Banco de Dados.....	54
3. PROCEDIMENTOS METODOLÓGICOS	56
3.1. CARACTERIZAÇÃO DA PESQUISA.....	56
3.2. DELIMITAÇÃO E POPULAÇÃO DA PESQUISA	57
3.3. ETAPAS DA PESQUISA	57
4. DESENVOLVIMENTO DO TRABALHO E APRESENTAÇÃO DE RESULTADOS	59
4.1. REQUISITOS	59
4.1.1. Requisitos funcionais	59
4.1.2. Requisitos não funcionais	60
4.2. ANÁLISE DAS FERRAMENTAS JÁ EXISTENTES NO MERCADO.....	60
4.2.1. DynaScape	61
4.2.2. Edificius LAND	62
4.2.3. AuE PhotoLANDSCAPE.....	63
4.2.4. App FloraMe	65
4.2.5. Conclusão da análise das ferramentas.....	66
4.3. TECNOLOGIAS UTILIZADAS	67
4.4. FERRAMENTAS UTILIZADAS	68
4.4.1. Wamp Server.....	69
4.4.2. Astah UML	69
4.4.3. Visual Studio Code	70
4.5. DIAGRAMA DE CASO DE USO.....	70
4.6. MODELAGEM E DESENVOLVIMENTO DO PROTÓTIPO	77
5. CONSIDERAÇÕES FINAIS.....	80
5.1. EM RELAÇÃO AO OBJETIVO GERAL	80
5.2. EM RELAÇÃO AOS OBJETIVOS ESPECÍFICOS	80
5.3. TRABALHOS FUTUROS	81
REFERÊNCIAS	83

APÊNDICE A – TELA PRINCIPAL DA APLICAÇÃO	88
APÊNDICE B – TELA PRINCIPAL DA APLICAÇÃO COM MENU EXPANDIDO ..	88
APÊNDICE C – MENU EXPANDIDO COM LISTA DE PLANTAS.....	89
APÊNDICE D – PALETA VEGETAL SOL.....	89
APÊNDICE E – PALETA VEGETAL SOMBRA	90
APÊNDICE F – EXEMPLO DE SIMULAÇÃO EXECUTADA COM A APLICAÇÃO	90

1. INTRODUÇÃO

Ao longo da história os jardins sempre estiveram presentes, refletindo o bem-estar humano, a cultura, o poder ou a religiosidade de seus povos. Ao longo dos tempos, eles foram evoluindo e transformando os cenários, influenciando e contribuindo com a arte em uma forma única de retratar a sociedade. (QUINTADELLARTE, 2017)

O ramo do paisagismo, por muitos séculos, ficou restrito à utilização esporádica e doméstica, com raros registros ou documentos, que acabaram por não se popularizar rapidamente. Outros fatores colaboraram para a falta de popularidade dos jardins verticais, entre eles o custo alto e a escassez de profissionais habilitados. (I LOVE DÉCOR, 2012)

Segundo Bergamasco (2002), foi no início dos anos de 1990 que começou uma maior valorização dos projetos de paisagismo, devido a um movimento de duas décadas atrás onde o homem busca priorizar seu bem-estar e qualidade de vida. Assim, tende a ter um retorno às coisas da natureza. A partir deste marco, o paisagismo começou a ser valorizado nos empreendimentos imobiliários, nas grandes cidades e até mesmo nas residências.

No cenário atual dos grandes centros urbanos, onde áreas verdes como praças, parques arborizados, jardins, bosques estão cada vez mais escassos, percebe-se o concreto cada vez mais e mais presente em nosso dia a dia, composto por uma grande aglomeração de arranha-céus, condomínios, casas, muros, estradas, entre outras construções. Amparado, principalmente, pelo acelerado crescimento da construção civil, este cenário torna nosso cotidiano cada vez mais cinza e menos vivo. Consequentemente, acaba contribuindo significativamente por gerar alguns problemas ambientais e, ao nos afastar da natureza, diminui a qualidade de vida e o bem-estar humano. (BARBOSA; FONTES, 2016)

Pensando em amenizar a falta de áreas verdes nas grandes cidades, integrando os centros urbanos com a natureza, trazendo diversos benefícios ao homem e possibilitando decorar pequenos espaços com uma roupagem totalmente nova, floresce o conceito de jardim vertical. A verticalização na disposição das plantas

torna possível criar diversas combinações e modificar ambientes corporativos e residenciais, evidenciando ambientes vivos e com grande variedade botânica. (SCHERER; FEDRIZZI, 2014)

Neste contexto de expansão paisagística, de grandes transformações e evolução dos jardins, onde, segundo Barbosa e Fontes (2016, p. 115) “a partir do desenvolvimento tecnológico, os jardins verticais tomaram novas proporções e hoje estão muito além de simples trepadeiras direcionadas sobre muros”, percebe-se uma lacuna de ferramentas que auxiliem na elaboração e utilização dos projetos de jardins verticais.

Pensando no crescimento exponencial deste ramo, é essencial que as ferramentas tecnológicas também estejam em constante evolução e supram as necessidades dos profissionais paisagistas, proporcionando maior realismo na elaboração e simulação dos jardins, atendendo aos requisitos do projeto, agregando valor ao produto, permitindo que o próprio cliente faça suas simulações e experiencie suas criações paisagísticas.

1.1. JUSTIFICATIVA

Devido ao crescimento do paisagismo e, em especial, dos projetos de jardins verticais, inúmeras são as técnicas, soluções, ferramentas e variedade botânica utilizada na elaboração e construção de jardins verticais.

Além da valorização imobiliária, os jardins proporcionam qualidade de vida ao homem, aliado ao contato com a natureza. Segundo o botânico francês Blanc (2008), muitos são os benefícios que geramos ao criarmos jardins verticais trazendo natureza e biodiversidade para a vida cotidiana, pois, devido ao efeito de isolamento térmico, o jardim vertical ajuda a reduzir o consumo de energia, tanto no inverno (protegendo o edifício do frio) quanto no verão.

Outro benefício que os jardins verticais proporcionam, é o efeito de limpeza do ar com as folhas das plantas e o feltro que é o pano de fundo do jardim vertical, pois nele ficam retidas as impurezas do ar e, posteriormente essas impurezas acabam por mineralizar-se e fertilizar o jardim. “O Jardim Vertical é, portanto, uma ferramenta eficiente para remediar o ar e a água sempre que superfícies planas já são

amplamente utilizadas por atividades humanas retirando as impurezas do ar.” (BLANC, 2008, p.4).

Entretanto, percebe-se uma lacuna neste mercado inovador e promissor, onde algumas poucas ferramentas de simulação gráfica para este propósito estão disponíveis no mercado, tentando aproximar a projeção da realidade e, demonstrar de uma forma mais palpável o resultado do projeto.

Neste sentido, este trabalho busca romper as barreiras tecnológicas que se impõem à elaboração, experimentação e visualização gráfica de um projeto de jardim vertical e, propõe buscar uma solução, ou hipóteses que amenizam a lacuna de ferramentas gráficas existentes no mercado do paisagismo de duas formas: por meio da adoção de ferramentas e tecnologias já validadas e disponíveis no mercado, ou modelando um protótipo de aplicação que atenda aos requisitos pretendidos.

A aplicação objetiva que tanto um profissional do ramo, quanto um usuário comum possam simular jardins verticais em ambientes reais através de fotomontagens com variedade de plantas possíveis de utilização no projeto, proporcionando ao projetista ou arquiteto agregar valor ao produto e, proporcionar ao usuário maior interação, inspiração e identificação com seu projeto de jardim vertical.

O alinhamento deste trabalho com o PPC do curso de Gestão da Tecnologia da Informação fica explícito no ponto em que lista as competências do egresso, citando, entre elas, “prospectar novas tecnologias na área de TI, bem como facilitar sua incorporação a organização e selecionar adequadamente métodos, técnicas e ferramentas para o desenvolvimento de soluções em TI.” (PLANO PEDAGÓGICO DO CURSO, 2017, p.6)

Ao mesmo tempo, atinge um dos objetivos do curso, de proporcionar ao aluno “desenvolver competências para a tomada de decisões estratégicas sobre a adoção de tecnologias da informação de modo alinhado às necessidades do negócio”. (PLANO PEDAGÓGICO DO CURSO, 2017, p.5)

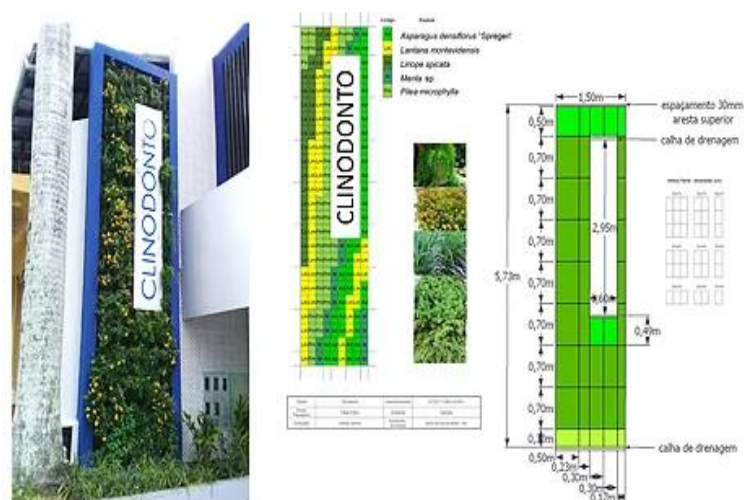
No sentido particular, este trabalho busca analisar e adotar a melhor solução tecnológica, que auxilie e esteja alinhada com o ramo de negócio de um grande amigo, que além de desenvolver projetos na área do paisagismo, oferece cursos de capacitação para interessados em trabalhar com jardins verticais.

1.2. DEFINIÇÃO DO PROBLEMA

O projeto tem como objetivo solucionar uma dor ou dificuldade encontrada no dia a dia de uma empresa que projeta, constrói e instala jardins verticais. Dentre as atividades da empresa, destacam-se as ações de marketing com o objetivo de angariar novos clientes e buscar novas parcerias. Nesse contexto de oferecer e demonstrar seus serviços aos clientes, a instituição parceira dessa proposta de trabalho sente a necessidade de aprimorar ou desenvolver uma ferramenta gratuita, que possibilita simular e demonstrar, de uma forma prática e realista, seus projetos para seus clientes, proporcionando, assim, uma melhor visualização gráfica da simulação do jardim.

Atualmente, é utilizada, para simulação e apresentação do projeto, uma paleta vegetal contendo todas as plantas contempladas no projeto, além de uma matriz do tamanho do jardim idealizado, identificando a posição de cada planta e suas cores, possibilitando ao cliente uma pequena visualização do resultado do projeto, como mostra a figura 1. Desta forma, evidencia-se a lacuna de ferramentas gratuitas de simulação de jardins verticais que atenda suas necessidades.

Figura 1 - Representação gráfica de projeto de jardim vertical



Fonte: página *web* da empresa parceira¹

¹Disponível em: <<https://www.verticajardins.com.br/>> Acesso em: 22 fev. 2020.

Embasado por este cenário, o presente trabalho busca responder ao seguinte questionamento: De que forma podemos agregar uma nova opção à lacuna de ferramentas gráficas gratuitas atualmente disponíveis para simulação de jardim vertical?

1.3. OBJETIVOS

1.3.1. Objetivo geral

O presente projeto objetiva adoção e implementação de soluções ou tecnologias que preencham a lacuna de ferramentas gráficas gratuitas utilizadas para simulação de jardim vertical no mercado do paisagismo.

1.3.2. Objetivos específicos

- Fazer o levantamento de requisitos para a aplicação;
- Descrever as ferramentas disponíveis em relação ao uso de *softwares* na simulação de jardins verticais, analisando se atendem os requisitos do projeto;
- Modelar um protótipo que atenda os objetivos propostos neste trabalho, caso a análise das soluções existentes revele que elas não atendem plenamente os requisitos do presente projeto;
- Desenvolver o protótipo requerido.

1.4. ESTRUTURA DO TRABALHO

Este trabalho está estruturado em cinco capítulos: no primeiro, definimos o escopo do tema abordado com suas nuances e características próprias, com a definição do problema identificado, e a justificativa para o desenvolvimento do trabalho.

No segundo capítulo, temos a fundamentação teórica. Neste capítulo, são apresentados os tópicos fundamentais para o entendimento do contexto inserido e os conceitos necessários para o desenvolvimento do projeto.

No capítulo seguinte, determinamos a metodologia de pesquisa, pontuando os limites da pesquisa e determinando sua população.

O capítulo 4 apresenta algumas das etapas no desenvolvimento da proposta, abordando estudo e análise das ferramentas disponíveis, além de conceitos e modelos utilizados para o desenvolvimento de aplicações *web*.

O capítulo final explicita e analisa os resultados alcançados com o desenvolvimento do projeto, apresentando as considerações finais obtidas em relação aos objetivos propostos e à trabalhos futuros.

2. JARDINS VERTICAIS E A ENGENHARIA DE SOFTWARE

2.1. HISTÓRIA E EVOLUÇÃO DOS JARDINS

Desde o início da civilização, os jardins sempre estiveram presentes na sociedade, carregando consigo histórias de poder, riqueza e religiosidade, tendo diferentes significados, de acordo com sua sociedade e época de civilização. (MATTIUZ, 2010)

Percebe-se, na cultura cristã, logo no início, antes mesmo das civilizações, que o conceito de jardim já existia, pois, segundo a Bíblia, o Jardim do Éden era um lugar de paz, fartura e aconchego para Adão e Eva.

Os primeiros documentos históricos de jardins que se têm registro são dos Assírios na Mesopotâmia. Segundo historiadores, eles dominavam as técnicas de irrigação e drenagem, criando diversos jardins com hortas e pomares. Um dos jardins mais falados da história, os Jardins Suspensos da Babilônia (605-652a.C.) - figura 2 - é considerado por muitos como uma das maravilhas da humanidade, devido à beleza e grandeza dos elementos arquitetônicos. Apesar de alguns historiadores contestarem sua existência, outros dizem que esse jardim foi construído por Nabucodonosor, como forma de homenagear a sua esposa. (RENATO RODRIGUES, 2016)

Figura 2 - Representação dos jardins suspensos da Babilônia



Fonte: página *web* clube de jardinagem²

² Disponível em: < <http://www.clubedejardinagem.com.br/index.php/jardinagem-paisagismo/32-o-jardim-egito-babilonia-persia-grecia-e-roma-antigas>> Acesso em: 09 abr. 2020.

Os jardins persas tinham grande influência do islamismo e contemplavam elementos da natureza em quadrantes. Nele, eram cultivadas plantas frutíferas, ornamentais e aromáticas. Mais tarde, acabaram por influenciar os jardins árabes. Na Grécia antiga, os jardins eram vistos como locais sagrados.

Figura 3 – Jardim romano



Fonte: página web clube de jardinagem³

O jardim romano foi bem valorizado com as artes gregas, tendo variedade de elementos arquitetônicos, cultivando plantas úteis, muros com trepadeiras, sempre valorizando a arquitetura das casas e completando-as de forma útil em todo o tempo do dia. (MATTIUZ, 2010)

No renascimento, os jardins assumem características próprias em cada país. Na Europa, segundo relato da professora Mattiuz (2010) “[...] o culto da forma fazia com que as plantas fossem interpretadas como esculturas que se integravam à imponência das construções”. Neste tipo de jardim havia muitos elementos arquitetônicos, como estátuas e fontes.

Os primeiros jardins orientais datam de 2000 a.C. e têm características de contemplação e imobilidade, voltados para a meditação e o silêncio, composto por elementos que assumiam significados simbólicos. Entre os elementos, podemos citar pedras, fontes, cascalhos, água, pontes e lanternas.

³ Disponível em: < <http://www.clubedejardinagem.com.br/index.php/jardinagem-paisagismo/32-o-jardim-egito-babilonia-persia-grecia-e-roma-antigas>> Acesso em: 09 abr. 2020.

Figura 4 - Jardim oriental

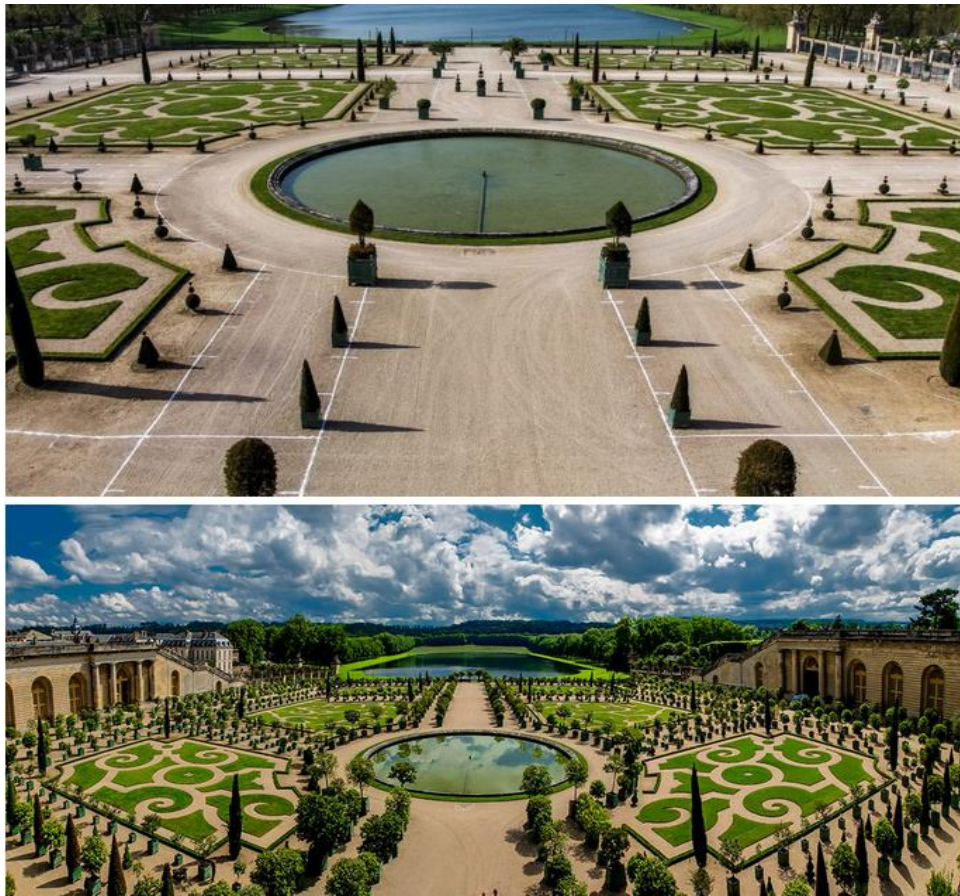
Fonte: página web clube de jardinagem⁴

Os jardins franceses foram influenciados inicialmente pelos italianos, mas acabaram desenvolvendo seu estilo próprio, com formas geométricas precisas em planos monumentais. Um exemplo é o famoso Jardim de Versalhes onde,

os jardins foram estruturados em uma série de terraços abertos onde eram construídos canteiros elaborados com topiaria em buxinhos. Os espaços internos dos canteiros eram preenchidos com flores coloridas, e os externos por pedras brancas trituradas, para contrastar com o verde da topiaria. (CLAUDIA FABRINO MACHADO MATTIUZ, 2010, p.5)

A topiaria consiste na arte ou técnica de jardinagem, onde as plantas são podadas de forma artística ou ornamental. Geralmente, essa poda é feita com tesouras ou equipamentos especializados. Abaixo, temos imagens do Jardim de Versalhes.

⁴ Disponível em: < <http://www.clubedejardinagem.com.br/index.php/jardinagem-paisagismo/31-o-jardim-oriental> > Acesso em: 09 abr. 2020.

Figura 5 - Jardins de Versalhes

Fonte: *p\u00e1gina web* Unitur.⁵

Outro exemplo de grande paisagista franc\u00eas \u00e9 o bot\u00e2nico Patrick Blanc, sendo considerado o pai dos jardins verticais, desenvolvendo e difundindo suas t\u00e9cnicas de jardins pelo mundo, realizando diversos projetos e estudos em in\u00fameros pa\u00edses, como por exemplo o projeto de jardins verticais feitos no Sofitel em Dubai, conforme visto na figura 6, e o projeto Caixa Forum, em Madrid, na figura 7.

⁵ Dispon\u00edvel em: <https://www.unitur.com.br/jardins-do-palacio-de-versalhes-a-arte-da-simetria-entenda-por-que/>. Acesso em: 09 maio 2020.

Figura 6 - Sofitel Palm Jumeirah, Dubai



Fonte: página *web* do botânico Patrick Blanc⁶

Figura 7 - Caixa Forum, Madrid



Fonte: página *web* do botânico Patrick Blanc⁷

Nos jardins contemporâneos, nenhum novo estilo de jardim foi criado, mas todas as tendências do passado influenciam sua expressão. Buscam, geralmente, atender uma necessidade específica, seja de clima ou época.

No Brasil, o marco do paisagismo teve início por conta dos trabalhos de Roberto Burle Marx no século XX. Ele projetou e executou os primeiros jardins públicos no Recife. Sempre utilizava plantas nativas para suas construções paisagísticas. Trabalhou em diversos países e deixou um grande legado proveniente de suas expedições, que podem ser conferidas no Instituto do Patrimônio Histórico e Artístico Nacional, buscando transmitir a riqueza das plantas que cultivava e o encanto pela criação de jardins. (MATTIUZ, 2010)

⁶ Disponível em: <<https://www.verticalgardenpatrickblanc.com/realisations/dubai/sofitel-palm-jumeirah-dubai>> Acesso em: 22 fev. 2020.

⁷ Disponível em: <<https://www.verticalgardenpatrickblanc.com/node/1414>> Acesso em: 22 fev. 2020.

2.2. TECNOLOGIAS DE SOFTWARE EM PROJETOS DE JARDINS VERTICAIS

No universo paisagístico, diversos fatores contribuem para o sucesso de um projeto. Além de conhecimentos sobre botânica e engenharia, é essencial que o profissional, projetista ou arquiteto esteja amparado por ferramentas que auxiliem no planejamento, elaboração e composição dos ambientes. (AUE SOFTWARE, 2016)

No mercado cada vez mais competitivo, já é consolidada como uma necessidade a redução de tempo na execução dos processos das organizações. O desenvolvimento de projetos paisagísticos é um trabalho que demanda tempo e conhecimento. No entanto, algumas ferramentas podem auxiliar esta tarefa. Com a contribuição de ferramentas gráficas de visualização, que minimizam os riscos de projeto, é possível maior agilidade nos processos, precisão e realismo nos resultados apresentados.

Outro fator a ser considerado é que, utilizando tecnologias de *software* nos projetos de jardim vertical, aumenta-se a produtividade e, conseqüentemente, proporciona condições de reduzir os prazos propostos ao cliente, aumentando a credibilidade.

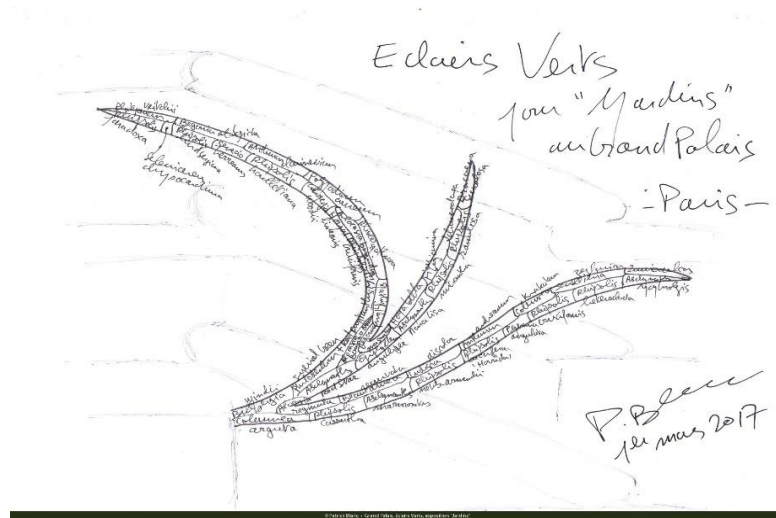
No ramo de paisagismo, existem algumas opções de representação gráfica ou simulação de projetos. No entanto, algumas lacunas ainda são visíveis, seja por conhecimento técnico ou valores monetários.

2.2.1. Desenho manual

Alguns profissionais mais relutantes à tecnologia ainda preferem fazer a representação de jardins em desenho manual. Desta forma, reproduz a realidade de acordo com a qualidade do profissional e, também, acaba por não manter a padronização nos projetos.

Um exemplo de utilização desta técnica está exposto nas figuras abaixo, onde o botânico francês Patrick Blanc, mundialmente conhecido por seus fabulosos projetos de jardins verticais em diversos países, faz a representação de um jardim vertical com um desenho manual, construído sobre um desenho técnico do ambiente.

Figura 8 - Esboço manual de projeto concluído em Paris por Patrick Blanc.



Fonte: página web do botânico Patrick Blanc.⁸

Figura 9 - Esboço manual e projeto concluído em Paris por Patrick Blanc.



Fonte: página web do botânico Patrick Blanc.⁹

⁸ Disponível em: < <https://www.verticalgardenpatrickblanc.com/node/12476>> Acesso em: 22 fev. 2020.

⁹ Disponível em: < <https://www.verticalgardenpatrickblanc.com/node/12476>> Acesso em: 22 fev. 2020.

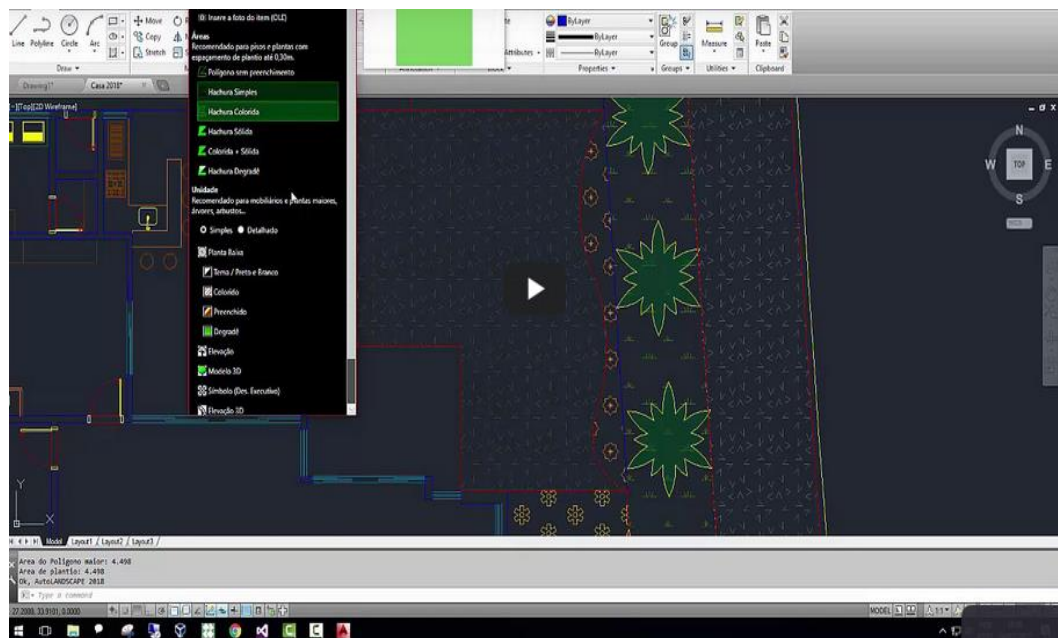
2.2.2. Softwares CAD

Muitos profissionais utilizam-se de softwares *CAD* – que é um tipo de *software* para desenho auxiliado por computador. Nele, é possível desenhar tudo que se pode imaginar, desde um prego até árvores, além de possuir um excelente sistema de medidas.

No entanto, este tipo de *software* demanda conhecimento técnico e tempo e, assim, acabam por reduzir a produtividade, ao menos em um primeiro momento. Outro problema deste tipo de representação é a falta de padronização dos desenhos, pois, segundo Oliveira et al. (2007)

em alguns casos, recebemos desenhos tão mal feitos e sem critério que pode ser mais rápido desenhar novamente, no lugar de corrigi-lo e prepará-lo para o trabalho. Ainda não há uma norma técnica da ABNT (Associação Brasileira de Normas Técnicas) para disciplinar a forma de desenhar em CAD.

Figura 10 - Imagem exemplo de software de paisagismo CAD



Fonte: página web AuE software.¹⁰

¹⁰ Disponível em: < <https://auesoftware.com/?id=6-AutoLANDSCAPE&ln=null> > Acesso em 22 fev. 2020.

2.2.3. Fotomontagem

Outra técnica utilizada para simulação gráfica de jardins, é fazer a representação dos jardins verticais em planilhas, onde as cores tentam produzir o efeito das plantas do jardim, proporcionando uma breve projeção do resultado. Esta é uma técnica que vem sendo utilizada pela empresa parceira, não demanda muito tempo ou conhecimento técnico especializado. No entanto, produz baixo impacto visual na apresentação do projeto ao cliente.

Primando pelo realismo nas apresentações e simulações, temos a fotomontagem como opção. Com esta técnica busca-se harmonizar diversos elementos do jardim fazendo colagens e montagens com sobreposição de imagens, simulando e transformando ambientes de forma intuitiva e prática, criando jardins para experimentação, com rapidez e realismo. (AuE SOFTWARE, 2016)

É uma técnica que produz um resultado bem próximo da realidade com colagens modernas de fotos ou sobreposição de imagens e encanta pela qualidade e rapidez com que é feita a montagem dos jardins em *softwares* especializados.

Figura 11 - Técnica de fotomontagem para representação de jardins.



Fonte: página web AuE software.¹¹

¹¹ Disponível em: <<https://auesoftware.com/figuras/36-028.jpg>> Acesso em 22 fev. 2020.

2.3. JARDINS VERTICAIS, SEUS COMPONENTES E MANUTENÇÃO

Os jardins verticais, por definição, são formas de revestimento de alvenarias ou estruturas verticais, com o desenvolvimento e crescimento de uma superfície com vegetação auto aderente ou com o auxílio de estruturas e suportes diversos, com as raízes em contato com solo ou substrato. (SCHERER; FEDRIZZI, 2014)

Figura 12 - Exemplos de jardins verticais



Fonte: página web *Vertical Garden*¹²

Compreendendo o conceito tradicional, podemos subdividir os jardins verticais em dois tipos principais, de acordo com sua estruturação: as fachadas verdes ou cortinas verdes, também conhecidas como sistemas extensivos, e as paredes vivas.

2.3.1. Fachadas Verdes

Os sistemas de fachadas verdes caracterizam-se por revestir alvenarias ou estruturas verticais através de um revestimento de plantas, que acabam por dar uma impressão de pele verde sobre a edificação ou muro. Geralmente, utilizam plantas trepadeiras autoaderentes à superfície ou pendentes, tornando possível cobrir toda a estrutura com o crescimento e expansão das plantas. (SCHERER; FEDRIZZI, 2014)

¹² Disponível em: <<https://www.verticalgarden.com.br/post/jardins-verticais-10-razoes-para-amar>> 09 maio 2020.

Figura 13 - Fachada verde com trepadeiras se desenvolvendo no muro.



Fonte: página *web* casa e construção.¹³

Nos sistemas tradicionais, as mudas são plantadas diretamente no chão na base das estruturas. Desta forma, elas vão se alastrando através de raízes ou ramos autoaderentes. Comumente, vemos este tipo de jardim em muros ou casas com trepadeiras.

Outra forma de construção de uma fachada verde é com o auxílio de uma estrutura modular ou treliças, que proporcionem sustentação às plantas, criando uma espécie de cortina verde em frente à edificação. Também, são conhecidas como dupla fachada verde. Dentre as vantagens deste tipo de jardim vertical, está a possibilidade de instalação em partes com aberturas ou vidros do edifício, pois a estrutura não fica diretamente fixa na parede, dispensando, também, uma impermeabilização da parede. Abaixo, temos um exemplo de dupla fachada verde com treliças. (SCHERER; FEDRIZZI, 2014)

¹³ Disponível em: < <https://casaeconstrucao.org/paisagismo/trepadeiras/> > Acesso em: 15 mar. 2020.

Figura 14 - Fachada Verde com suporte de treliças instalada em Vancouver.



Fonte: página web da empresa GSKY.¹⁴

2.3.2. Paredes Vivas

Neste sistema de construção de jardins verticais, as plantas são alocadas em vasos ou pequenos blocos com cavidades para alocar o substrato e as plantas, ou em painéis geotêxteis - painéis desenvolvidos geralmente em módulos, feitos com fibra naval ou outro material impermeável em sua estrutura, com uma camada sobreposta de feltro ou manta geotêxtil, possuem pequenos berços ou bolsinhos para as plantas, criando uma estrutura de contato, proporcionando sustentação, filtragem e substrato. Entre as principais aplicações das mantas têxteis, além do uso em sistemas de jardinagem, podemos citar: filtração e drenagem, no controle de erosão, em proteção costeira de margem de rios e canais, como proteção em sistemas de impermeabilização e depósitos de resíduos sólidos.

Neste modelo de implantação, as plantas não possuem contato direto com o solo na base da estrutura, podendo utilizar substrato nas cavidades ou bolsinhos. Porém, o mais comum é adotar o sistema hidropônico, sem a utilização de substrato,

¹⁴ GSKY PLANT SYSTEM. Disponível em: <<https://gsky.com>>. Acesso em 09 abr. 2020.

apenas a irrigação com fertilizantes solúveis. (SCHERER; FEDRIZZI, 2014)

Figura 15 - Parede viva com sistema de painéis geotêxteis modulares.



Fonte: página *web* da empresa parceira¹⁵

As paredes vivas podem ser instaladas tanto no exterior quanto no interior de ambientes, desde que tenham iluminação natural ou artificial suficiente. Causam um grande impacto visual e atualmente, são bastante difundidas em empreendimentos de grandes corporações. Abaixo, temos um exemplo de parede viva com o sistema de painéis geotêxteis, instalado em ambiente corporativo, interno e com iluminação artificial.

2.3.3. Irrigação

Diversos componentes e fatores se inter-relacionam e contribuem para o sucesso de um jardim vertical. A necessidade hídrica é um dos principais itens deste

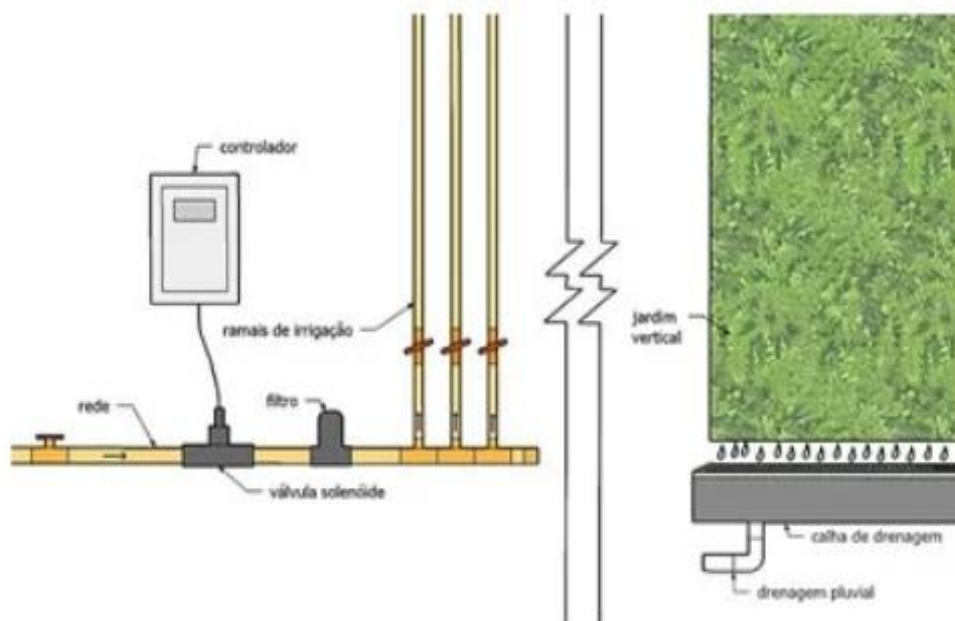
¹⁵ Disponível em: < <https://www.verticajardins.com.br/portfolio-1?pgid=jqe48zdu-fbb91abd-6bff-4200-8c5a-db7909a70743> > Acesso em: 22 fev. 2020.

sistema e deve ser adequada ao tipo de jardim e às espécies plantadas.

Deve-se levar em consideração, para a instalação de um jardim vertical, que as plantas ficam em um ambiente artificial, raramente em contato direto com o solo, geralmente com pouco substrato e utilizando nutrientes líquidos para sua nutrição. Cabe notar, ainda, que, quanto mais alto for o jardim, mais sofrerá com a ação do vento e do sol. (BARBOSA; FONTES, 2016)

Alguns sistemas de jardins verticais, geralmente mais simples, localizados em áreas externas, utilizam-se de irrigação com um sistema aberto. Neste sistema, a rega é feita por uma pessoa de forma manual ou de forma automática com um temporizador. O excesso de água que escorre do jardim vertical é direcionado e conduzido por uma calha na parte inferior para uma área de escoamento. Desta forma, não há reutilização da água e nutrientes, como mostra a figura 16.

Figura 16 - Sistema aberto de irrigação automatizada



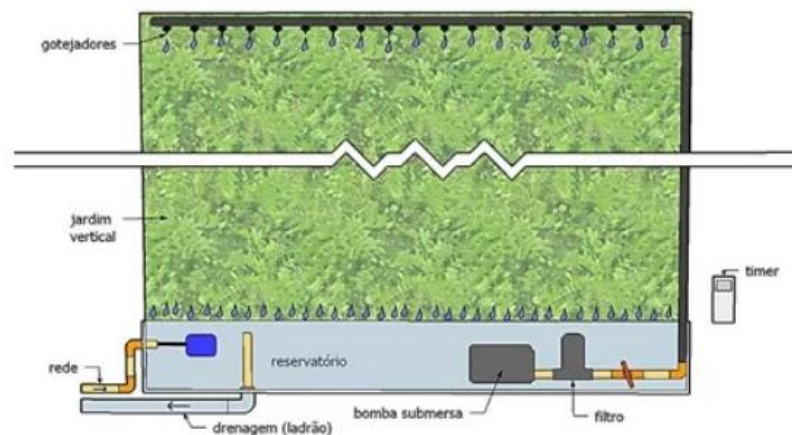
Fonte: página web da empresa parceira¹⁶

Jardins verticais profissionais utilizam-se de sistemas automáticos de irrigação. Nestes jardins, geralmente é implantado um sistema de fertirrigação ou hidroponia, também conhecido como sistema fechado, utilizando-se da técnica de gotejamento.

¹⁶ Disponível em: < <https://www.verticajardins.com.br/vertica-painel> > Acesso em: 22 fev. 2020.

Nele, a água é distribuída em mangueiras de polietileno, onde são fixados gotejadores. Estes gotejadores, por sua vez, distribuem a água na zona radicular das plantas. O excedente é coletado em um reservatório para nova utilização, que fica localizado na parte inferior do jardim, como mostra a figura 17.

Figura 17 - Sistema fechado de fertirrigação automatizada.



Fonte: página *web* da empresa parceira.¹⁷

De acordo com Cheng; Cheung; Chu (2010, apud BARBOSA; FONTES 2016, p. 6), “A irrigação pode estar presente em todos os tipos de jardins verticais. Ao projetá-la, deve-se considerar o fator gravidade”. Se um jardim tiver a irrigação feita de uma forma uniforme e pelo mesmo tempo, por gravidade a água escorrerá e a parte inferior ficará muito mais encharcada que a parte superior do jardim. De acordo com Irwin (2015 apud BARBOSA; FONTES 2016, p. 6), o excesso de água prejudica a respiração e absorção de nutrientes, além de aumentar os compostos tóxicos na planta, sendo um dos principais motivos de mortes de plantas em jardins verticais.

Desta forma, cada sistema de irrigação precisa ser planejado de maneira adequada para atender ao tamanho, tipologia e vegetação do jardim.

2.3.4. Nutrição e variedades de plantas

A escolha da vegetação deve ser planejada levando em consideração fatores

¹⁷ Disponível em: < <https://www.verticajardins.com.br/vertica-painel> > Acesso em: 22 fev. 2020.

como a necessidade hídrica, exigências do clima, nutrição, características quanto ao porte e crescimento e tempo de exposição à radiação solar. Em relação ao clima, as espécies precisam estar adaptadas quanto ao tempo de exposição ao sol e a necessidade de pleno sol, meia-sombra ou sombra-plena, além das demais condições locais. (BARBOSA; FONTES, 2016)

Diferentes espécies possuem variados portes e hábitos, e todos estes fatores devem ser considerados na escolha das plantas para harmonização com a estrutura do projeto. Algumas áreas podem necessitar de vegetação rasteira, trepadeiras, ou samambaias, pendentes. Já outro ambiente necessita de flores e temperos. Cada espaço com suas peculiaridades e especificidades.

Alguns tipos de substratos diversos são utilizados para a fixação das raízes das plantas em diferentes tipos de jardins, de acordo com a sua estrutura. São eles: o solo preparado, húmus e o feltro.

Outro fator importante é entender a exigência nutricional de cada espécie. Segundo Biscaro e Oliveira (2014 apud BARBOSA; FONTES 2016, p. 7), em fachadas verdes com espécies de trepadeiras plantadas diretamente no solo, ou paredes vivas com sistemas de vasos ou jardineiras, é possível fazer adubação e reposição de nutrientes com adubos granulados. Já, em paredes vivas contínuas ou jardins mais amplos, este tipo de nutrição acaba sendo inviável. Para solucionar este problema, utiliza-se fertirrigação ou hidroponia, onde são colocados os nutrientes solúveis em água e distribuídos em um sistema automático por gotejamento.

2.3.5 Iluminação

Os jardins verticais possibilitam criar ambientes incríveis e a iluminação é um elemento importante neste cenário, pois, além de poder acrescentar requinte e elegância com efeitos noturnos, ela também pode ser utilizada em ambientes que carecem de iluminação natural. Desta forma, vem complementar o conjunto de fatores que levarão ao sucesso do jardim vertical.

Os jardins necessitam de iluminação natural proveniente do sol para seu pleno desenvolvimento. Em locais com *déficit* de iluminação natural, a iluminação artificial torna-se imprescindível para o sucesso dos jardins. Em paredes vivas de ambientes

internos, a iluminação é muito funcional, pois, dependendo da vegetação contemplada no jardim, é necessário um estudo técnico com a quantia de iluminação necessária (medida em lux) para o pleno crescimento das espécies.

Segundo Kohler (2008 apud BARBOSA; FONTES 2016, p. 7), as fachadas verdes geralmente são mais utilizadas em ambientes externos, pois contemplam plantas de sol pleno e meia sombra, como espécies trepadeiras ou pendentes.

A iluminação é compreendida e determinada pelos lúmens que incidem em um metro quadrado. A unidade de medida é o lux, e a iluminância pode ser medida com um aparelho chamado luxímetro como mostra a figura 18, que, por sua vez, deve ser operado por profissional qualificado para tal atividade. (GRUPO MB, 2015)

Figura 18 - Luxímetro.



Fonte: página *web* da empresa Grupo MB.¹⁸

2.4. ENGENHARIA DE SOFTWARE

Projetar, modelar e construir *software* é uma tarefa desafiadora, que demanda criatividade e conhecimento técnico. Um dos fatores que demonstram a complexidade da tarefa é que cada projeto é único, não existe uma solução pronta e nem ao menos um único modo para a resolução. Outro ponto é que no universo do desenvolvimento de *software*, os projetos são planejados e executados por pessoas, condicionando, de certa forma, o sucesso do projeto à competência e integração da equipe e, nem sempre, os processos que apoiam o negócio são bem definidos. (ÁVILA; SPÍNOLA, 2007)

Um dos pontos cruciais para o sucesso de um projeto é a definição do escopo,

¹⁸ Disponível em: <https://grupomb.ind.br/mbobras/economia-de-energia/como-e-por-que-medir-a-iluminacao-no-ambiente-de-trabalho/>. Acesso em: 12 abr. 2020.

ou levantamento e modelagem dos requisitos. Neste ponto, pensando em aprimorar seus processos com redução de custos, prazos e retrabalho, as empresas têm se movimentado em direção a aprimorar e definir muito bem seus processos no apoio ao desenvolvimento.

Podemos considerar engenharia de *software* como sendo “a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, operação e manutenção de *software*.” (ÁVILA; SPÍNOLA, 2007, p.46).

Neste sentido, podemos compreender como fundamental para o desenvolvimento de aplicação ou *software*, o conhecimento e o uso adequado das ferramentas e técnicas que amparam um time de desenvolvimento, utilizando-se de uma abordagem sistemática em todos os processos, evitando retrabalhos, reduzindo custos e prazos, além de permitir maior controle de qualidade e entrega de valor ao produto.

2.4.1. Engenharia de requisitos

Segundo Pressman (2011, p.127) “o amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos é denominado engenharia de requisitos.”

No cenário de desenvolvimento de *software*, a elucidação dos requisitos do projeto definindo seu escopo é uma ação determinante para o sucesso dele. Inicia-se na comunicação com o cliente definindo as suas necessidades e as necessidades do usuário do sistema, verificando a viabilidade, buscando soluções razoáveis, documentando processos, validando soluções e gerenciando as necessidades do projeto. (PRESSMAN, 2011)

Com a engenharia de requisitos, busca-se aumentar a produtividade nos processos de desenvolvimento, reduzindo retrabalhos, manutenções de sistema, além de permitir aos profissionais maior controle sobre o projeto, em questão de custos, prazos e qualidade. (ÁVILA; SPÍNOLA, 2007)

Na literatura técnica, existem diversas definições para requisitos. Segundo Ávila e Spínola (2007, p.48), “podemos entender requisitos como sendo o conjunto de necessidades explicitadas pelo cliente que deverão ser atendidas para solucionar um

determinado problema do negócio do qual o cliente faz parte”. Geralmente, são subdivididos em: requisitos funcionais, requisitos não funcionais e requisitos de domínio.

Os requisitos de um *software*, quando bem especificados na fase inicial, são determinantes para o sucesso do projeto. Afinal, eles apoiam todo o processo de validação e desenvolvimento da ideia. Quando não estão bem definidos, podem causar muito retrabalho e, conseqüentemente, atrasos no desenvolvimento. Por vezes, podem até inviabilizar o projeto.

2.4.2. Requisitos funcionais

Os requisitos funcionais são aqueles que estão diretamente ligados às funcionalidades do *software*, demonstrando como o *software* deve se comportar em determinadas situações, como reagir à entrada e saída de dados específicos. Enfim, descreve ações que o sistema deve realizar e as que não deve fazer. (ÁVILA; SPÍNOLA, 2007)

É importante verificar a consistência dos requisitos no momento do seu levantamento, buscando evitar definições abstratas ou ambíguas que podem gerar dúvidas posteriores ou retrabalhos.

Dentre os requisitos funcionais de um *software*, podemos citar exemplos como:

- O sistema deve permitir cadastro de usuário;
- O sistema deve permitir impressão de relatório de vendas;
- O sistema deve permitir que o usuário faça pesquisa em todo ou determinado banco de dados;
- O sistema deve permitir editar imagens.

2.4.3. Requisitos não funcionais

Os requisitos não funcionais de um sistema podem ser compreendidos por definirem condições, propriedades e restrições do sistema. São requisitos mais críticos que os funcionais.

Se os requisitos funcionais mostram o que o *software* deve fazer, com os

requisitos não funcionais determinamos como ele deve fazer, ou seja, todo requisito não funcional deve ser uma premissa ou restrição do sistema. Um ponto importante a ser ressaltado, é que esses requisitos precisam ser mensuráveis, ou seja, tem como você verificar se ele está sendo atendido pelo software ou não. (ALFF, 2018)

Exemplos de requisitos não funcionais:

- O sistema deve ser multiplataforma: *Windows, Mac, Linux*;
- O tempo de resposta de processamento não pode ser superior a 10 segundos;
- O sistema deve ser desenvolvido em linguagem *Python*;

Deve-se ter cuidado, na fase de levantamento de requisitos, para que não se faça definições ambíguas ou imprecisas, como, por exemplo: o sistema precisa ser rápido, ou o *software* deve ser seguro. Nestes casos específicos, são requisitos que não são mensuráveis. Para ser requisito não funcional, precisa ser mensurável, passível de controle, como, por exemplo, o sistema poder ser capaz de criptografar os dados de cadastro de clientes, ou o sistema proporcionar tempo de resposta de consultas inferior a 5 segundos. (ALFF, 2018)

Por serem essencialmente mensuráveis, os requisitos não funcionais necessitam de algumas métricas para controle essenciais para seu entendimento. Por exemplo, métricas de velocidade como: tempo de atualização de tela, tempo de resposta a processamentos, métricas de uso como tempo gasto em treinamento, em relação à confiabilidade do sistema, disponibilidade do sistema, taxa de ocorrência de falhas, entre outras.

2.4.4. Requisitos de domínio

Segundo Ávila e Spínola (2007, p.48) requisitos de domínio são

derivados do domínio da aplicação e descrevem características do sistema e qualidades que refletem o domínio. Podem ser requisitos funcionais novos, restrições sobre requisitos existentes ou computações específicas.

Este tipo de requisito geralmente é definido por fatores ligados à legislação, órgãos de controle ou regras de negócio. Podemos citar, como exemplo, o cálculo do

valor de horas extras em determinado dia de feriado para um funcionário, onde, naquele dia a hora trabalhada tem o valor acrescido por um percentual X, determinado pela legislação trabalhista, ou o valor da média de terminado aluno, sendo calculado pela fórmula $((nota1*2 + nota2*3) /5)$. (ÁVILA; SPÍNOLA, 2007)

2.4.5. Levantamento de requisitos

A atividade denominada levantamento de requisitos é fundamental no processo de desenvolvimento de *software*. Ela objetiva a descoberta, elucidação e documentação dos requisitos do sistema, definindo o escopo do projeto. Para o sucesso de tal atividade,

analistas e engenheiros de software trabalham com clientes e usuários finais para descobrir o problema a ser resolvido, os serviços do sistema, o desempenho necessário, restrições de hardware e outras informações. (ÁVILA; SPÍNOLA, 2007, p.49)

Existem algumas ferramentas e técnicas que amparam o trabalho do analista de requisitos. Entre estas técnicas, temos a observação de cenário, onde o analista precisa estar atento às entradas e saídas do sistema, verificar as necessidades do cliente e usuários a fim de buscar uma solução de negócio. Neste sistema de observação, pode-se fazer anotações ou rascunhos com observações relevantes, para posterior análise e integração com os demais requisitos.

Outra solução é uma entrevista com *stakeholders* (cliente, patrocinadores, usuários etc.). O objetivo desta técnica é identificar o problema principal, pontuando as regras de negócio, e delimitar as demais necessidades do *software*. A entrevista pode ser informal ou estruturada, desde que objetiva e devidamente documentada, para que possa gerar artefatos posteriores relevantes ao desenvolvimento, como cenários ou casos de uso. (ALFF, 2018)

Opção bem difundida para execução do levantamento de requisitos, é fazer testes de mesa ou prototipação. Desta forma, faz-se a verificação do fluxo proposto pelo sistema, buscando todos os caminhos possíveis para verificar sua viabilidade, dando uma breve amostra de como o sistema se comportará. Outra forma de descoberta e levantamento de requisitos pode ser concebida através de um estudo

de mercado, detalhando quais ferramentas já existem disponíveis no mercado, suas principais funcionalidades e deficiências, buscando entender quais soluções já existem e como é possível gerar uma nova solução ainda mais eficiente e eficaz, reduzindo custos ou prazos. (ALFF, 2018)

2.4.6. Modelagem UML

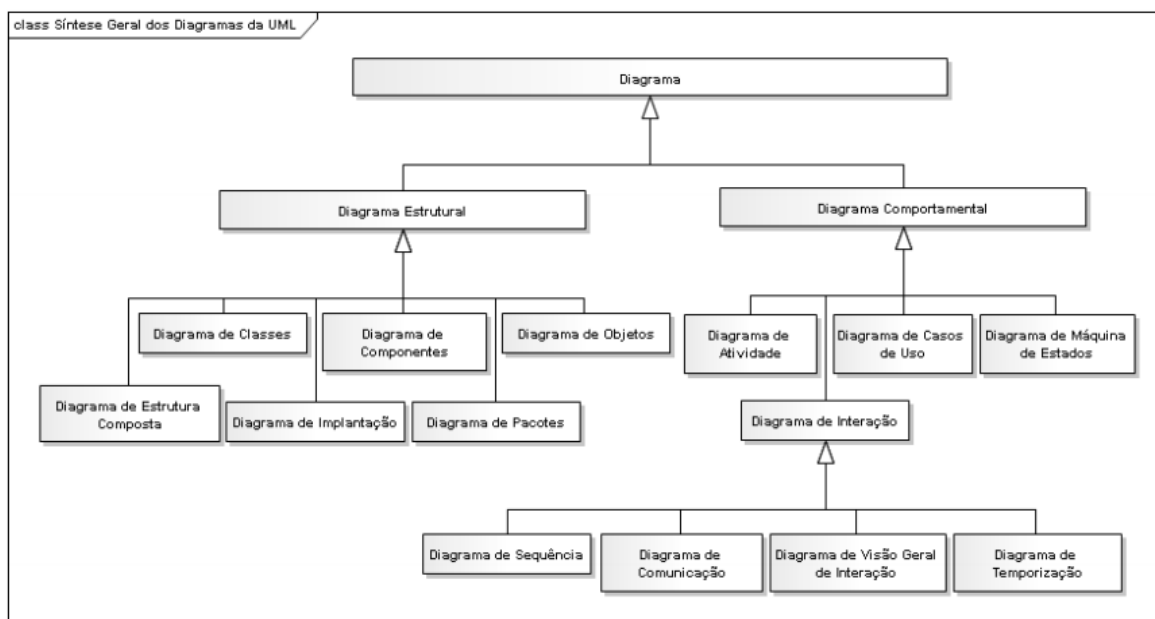
A *UML (Unified Modeling Language)*, que significa Linguagem Unificada de Modelagem é uma linguagem para especificação, documentação, visualização e construção de artefatos no desenvolvimento de software. Ela surgiu da fusão de três grandes métodos: *BOOCH*, *OMT* (Rumbaugh) e *OOSE* (Jacobson), e vem da necessidade de estabelecer um padrão, que facilite a comunicação entre os analistas de requisitos e o time de desenvolvimento. (MARTINEZ, 2006)

Atualmente, é a linguagem padrão internacional adotada na modelagem de software. Segundo Guedes (2011, p. 19), a *UML* “é uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação a objetos.”

Alguns profissionais podem afirmar que podem construir todo um sistema de cabeça, sem necessitar de nenhuma diagramação de elementos ou requisitos, ou que documentar o processo é demorado e muitas vezes causa apenas desperdício de tempo. No entanto, sabemos que, quanto mais complexo o sistema for e mais partes envolvidas no mesmo, mais difícil será alcançar metas, reduzir custos, cumprir prazos, levantar e gerenciar requisitos sem uma comunicação clara, documentada e padronizada das etapas e premissas do projeto. (GUEDES, 2011)

A *UML* é composta por vários diagramas (representações gráficas do modelo parcial de um sistema). São usados em combinação, separados ou em conjunto, com a finalidade de obter todas as visões e aspectos do sistema, podendo ser subdivididos em comportamentais e estruturais. (MARTINEZ, 2006)

Figura 19 - Diagramas UML.



Fonte: Guedes (2011, p. 41)

Entre os principais diagramas da *UML*, destacam-se:

- Diagrama de Casos de Uso;
- Diagrama de Classes;
- Diagrama de Objetos;
- Diagrama de Sequência;
- Diagrama de Implantação;

2.5. TECNOLOGIAS DE DESENVOLVIMENTO WEB

Com o avanço do uso da internet, ficou cada vez mais comum e necessário o uso de aplicações com base na *web*, principalmente, devido à acessibilidade e portabilidade, além da facilidade do uso.

Juntamente com o crescente mercado de aplicações *web*, também surgiram diversas linguagens e ferramentas que auxiliam os profissionais no desenvolvimento destas aplicações, cada uma com suas particularidades e recursos. Torna-se importante que o desenvolvedor conheça cada uma das linguagens, para que possa, assim, fazer a escolha adequada de qual tecnologia utilizar. (CAMARGO, 2010)

Conforme a literatura, aplicação *web* pode ser definida como sistema projetado para utilização através de um navegador, na internet ou em redes privadas. Portanto, é um conjunto de programas sendo executados em um servidor de HTTP, sendo acessado por diferentes usuários. (GONÇALVES, 2007)

Os sistemas *web* geralmente são baseados em informação com conteúdo e as aplicações *web* são fundamentadas no paradigma do hipertexto, onde os elementos básicos são nós e elos (*links*). Os nós mostram o conteúdo e os *links* permitem navegar entre os nós. (JACYNTHO, 2008)

Nos primórdios na internet, as aplicações *web* eram apenas hipertexto com páginas estáticas. Com a evolução das ferramentas e das tecnologias de desenvolvimento, hoje as aplicações *web* trazem páginas dinâmicas, com inúmeros recursos e interações e, por vezes, ainda está integrada com sistemas legados, demonstrando a versatilidade e potencial das aplicações. (CAMARGO, 2010)

Dentro do espectro das tecnologias relacionadas ao desenvolvimento *web*, podemos destacar: *HTML*, *CSS*, *JavaScript* e *PHP*, ressaltando que cada uma possui suas características bem definidas, com recursos e elementos próprios.

2.5.1. *HTML*

Segundo Camargo (2010, p.435)

HTML (Hyper Text Markup Language) é uma linguagem de formatação muito simples baseada em marcadores (*tags*) para a criação de documentos hipertexto. A principal característica do hipertexto é a possibilidade de se interligar a outros documentos da *web*.

A linguagem *HTML* foi criada em 1991 no *CERN (European Council for Nuclear Research)* na Suíça, pelo inglês Tim Berners-Lee, com o objetivo de interligar computadores do laboratório de Física e interligar instituições de pesquisa, facilitando a exibição e acesso aos documentos científicos.

As versões anteriores da linguagem *HTML* são: *HTML*, *HTML+*, *HTML 2.0*, *HTML 3.0*, *HTML 3.2*, *HTML 4.0*, *HTML 4.01*, *HTML 5* (versão atual).

A popularidade da linguagem cresceu junto com a internet, tendo como uma das principais vantagens a portabilidade e a facilidade com que usuários leigos podem

aprender seu funcionamento. Para que um navegador possa renderizar arquivos com comandos *HTML*, é necessário que este possua as *tags* da linguagem e o arquivo precisa ter a extensão *.html* ou *.htm*. (CAMARGO, 2010)

Figura 20 - Estrutura de uma página *HTML5*.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8"/>
5   <title>Document</title>
6 </head>
7 <body>
8   <!-- Conteúdo -->
9 </body>
10 </html>
```

Fonte: Elaborado pelo autor.

2.5.2. CSS

O CSS determina a estilização de uma página *HTML*. Segundo Ribeiro et al. (2006, p.3)

Cascading Style Sheets (CSS - Folhas de Estilo em Cascata), é uma tecnologia que nos permite criar páginas *web* de uma maneira mais exata. Esta tecnologia permite fazer muitas coisas que não era possível utilizando somente *HTML*, como incluir margens, tipos de letra, fundos, cores etc.

O CSS permite aplicar estilos seletivamente em elementos *HTML*, fazendo da sua combinação de uso com o *HTML5* uma excelente ferramenta de estilização de conteúdo. O CSS pode ser escrito dentro do arquivo *HTML*, ou pode ser escrito em outro arquivo separado e, este, por sua vez, “linkado” com o arquivo principal. O CSS3 é a atual versão e, proporcionando inúmeros recursos como por exemplo, animações em 2 e 3 dimensões.

2.5.3. JavaScript

Apesar da semelhança pelo nome, ela não possui nenhuma ligação com a

linguagem *Java*. O *JavaScript* é muito comumente utilizado em conjunto com *HTML* e *CSS*, atuando na manipulação do comportamento das páginas, sendo suportada por praticamente todos os navegadores atuais.

A linguagem *JavaScript* “é uma linguagem de programação *web* que é executada do lado cliente, desenvolvida pela *Netscape*. Essa linguagem possibilita adicionar recursos dinâmicos às páginas *HTML*.” (RIBEIRO et al., 2006)

De acordo com Ribeiro et al. (2006, p.4)

O fato de o *JavaScript* não exigir a instalação de *softwares* especiais para execução (assim como ocorre com linguagens mais sofisticadas, como *PHP* e *ASP*) o fez ter grande popularidade, já que qualquer página *HTML* pode conter recursos em *JavaScript*.

2.5.4. *PHP*

O *PHP* surgiu inicialmente como o nome de *PHP/FI*. Foi criado em 1994 por Rasmus Lerdof. Segundo Barreto (2000, p. 6), “a primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como “*Personal Home Page Tools*” (ferramentas para página pessoal)”. Ainda, segundo o autor supracitado,

o “FI” veio de um outro pacote escrito por Rasmus que interpretava dados de formulários *HTML* (*Form Interpreter*). Ele combinou os scripts do pacote *Personal Home Page Tools* com o FI e adicionou suporte a *mySQL*, nascendo assim o *PHP/FI*, que cresceu bastante, e as pessoas passaram a contribuir com o projeto.

O fato de, em 1995, o código ter sido disponibilizado aos desenvolvedores, permitiu correções de bugs e aperfeiçoamentos diversos. Em 1997, o *PHP/FI* encontrava-se ineficiente para algumas aplicações e com poucos recursos. Foi então que Andi, Rasmus e Zeev resolveram fazer uma reescrita do interpretador. Como o *PHP/FI 2.0* trazia uma impressão de uso pessoal, eles decidiram renomear o novo projeto somente como *PHP: Hypertext Preprocessor*. (SILVA, 2018)

A linguagem, posteriormente, passou por outras versões como: *PHP4.0*, *PHP5*, *PHP 7* e atualizações em busca de melhores performances em aplicações complexas.

É uma linguagem *open source*, geralmente utilizada para desenvolvimento *web*. O *PHP* é relativamente fácil para um iniciante, e oferece muitos recursos aos

profissionais. O código *PHP* é marcado pelas instruções ou *Tags*, delimitando o início e fim do “modo” *PHP* e o código pode ser embutido dentro do *HTML*. O que distingue o *PHP* de algo como o *JavaScript* é que o código é executado no servidor e transformado em um *HTML*, que é enviado ao navegador.

O *PHP* pode fazer muitas coisas, como páginas *web* dinâmicas, pode coletar dados de formulários, enviar e receber *cookies*, entre outras aplicações. É focado em *scripts* do lado do servidor, e tem compatibilidade com a maioria dos sistemas operacionais, além de suporte a variedade de banco de dados, como, por exemplo, *MySQL*, *Oracle*, *PostgreSQL*, entre outros, sendo este um forte diferencial da linguagem. (BARRETO, 2000)

Atualmente, diversas empresas utilizam *PHP* como sua base principal para o desenvolvimento de seus sistemas, entre elas o Facebook, que iniciou utilizando *PHP*, e depois desenvolveu uma linguagem própria, baseada em *PHP*. Outros exemplos são o Twitter e Yahoo. (SILVA, 2018)

2.6. FRAMEWORKS DE DESENVOLVIMENTO

Desenvolver aplicações para a *web* do zero é uma tarefa árdua e que demanda tempo. Pensando em reduzir esse tempo e complexidade da tarefa, surgiram os *frameworks* que, com as suas particularidades, prometem ajudar no desenvolvimento rápido e seguro das aplicações. (GUIMARÃES, 2019)

De forma genérica, podemos dizer que *frameworks* funcionam como bibliotecas de arquivos, com diversas funções básicas para o desenvolvimento de aplicações.

Um *framework* ou arcabouço é uma estrutura de suporte definida em que outro projeto de software pode ser organizado e desenvolvido, quando se analisa o conceito no âmbito do desenvolvimento de software. Um *framework* pode incluir programas de suporte, bibliotecas de código, linguagens de script e outros *softwares* para ajudar a desenvolver e juntar diferentes componentes de um projeto de software. (WILLEMANN E IBARRA, 2007, p. 41)

As vantagens do uso de *frameworks* estão relacionadas ao aproveitamento e

reutilização de código, aumentando a produtividade, segurança, além de padronização nos processos e códigos. Já as desvantagens se dão pelo fato de a curva de aprendizado e domínio de um *framework* ser lenta, podendo, também, gerar vícios no programador, negligenciando o conhecimento da linguagem nativa, e introduzir problemas de compatibilidade entre *frameworks* distintos. (GUIMARÃES, 2019)

2.6.1. Bootstrap

Criado com o objetivo de otimizar o desenvolvimento de aplicações, adotando uma estrutura única e, assim, reduzindo inconsistências na forma de programar, em meados de 2010 foi desenvolvido o *Bootstrap*, por Jacob Thorton e Mark Otto, engenheiros do Twitter. O projeto deu tão certo, que a ferramenta foi disponibilizada no *GitHub* como *software* livre e lançado oficialmente em agosto de 2011. (CAMPOS, 2017)

Bootstrap é um *framework* desenvolvido para otimizar a criação do *frontend*. Em outras palavras, é um kit de ferramentas de código aberto para desenvolvimento em *HTML/CSS/JS*. Hoje, é o *framework* mais popular, intuitivo e com uma comunidade superativa, o que possibilita constantes atualizações. (MOTA, 2014)

Atualmente, na versão 4.1.3, a principal aplicação do *Bootstrap* é atuar na criação de sites responsivos (*mobile*), otimizando o tempo do desenvolvedor, atuando na reutilização de código com uma estrutura única, além de possuir vários plugins em *JavaScript (jQuery)*. Também, possui diversas bibliotecas prontas para serem incluídas em qualquer projeto.

Entre alguns recursos oferecidos, podemos citar: *slideshow, menu dropdown, carousel, modal, cards, layouts* de tabelas, listas, entre outros. São estruturas que tornam o processo de desenvolvimento mais rápido e, ao mesmo tempo, mantêm um padrão na codificação e qualidade dos projetos.

Como o *framework* é *open source*, pode ser baixado no seu site oficial. Oferece um aprendizado rápido e toda documentação é disponibilizada gratuitamente. Ainda oferece, no próprio site, um espaço onde é possível visualizar alguns exemplos de aplicação dos códigos, como *layouts* de tabelas, menus, barras de navegação etc.,

justificando, desta forma, o porquê de ser, atualmente, um dos mais populares e intuitivos *frameworks* de desenvolvimento *web*.

2.6.2. Angular

Angular é um *framework* de código aberto. Inicialmente, foi lançado em 2012 com o nome de *AngularJS*. O *framework JavaScript* é um conceito de *Misko Hevery* e *Adam Abrons*, engenheiros da Google, para solucionar problemas internos de desenvolvimento. Já que a ferramenta deu certo, foi disponibilizada a todos os desenvolvedores. (LONGEN, 2020)

Contando com uma comunidade ativa e disponibilizando constantes atualizações e correções, em sua segunda versão,

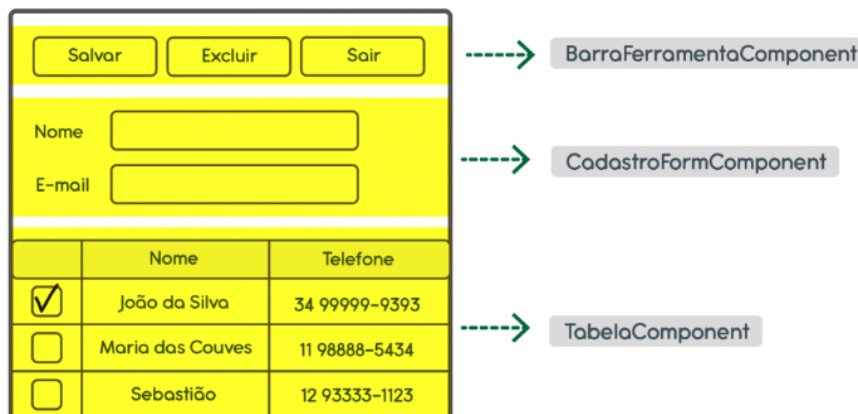
lançada em 2016, um *framework* completamente novo foi disponibilizado, e adotando a linguagem *TypeScript*, criada pela *Microsoft*, o *JS* foi retirado de seu nome. Então, em versões superiores, é chamado apenas de *angular*. (NALESSO, 2018)

Entre as vantagens do uso deste *framework*, podemos citar o aumento de produtividade, permitindo criações com excelente qualidade, com grande facilidade de entendimento, além de possuir várias bibliotecas com diversos componentes. Conforme cita Afonso (2018), entre as principais características do Angular, “podemos destacar os componentes, *templates*, diretivas, roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas, como a de executar os testes unitários de uma aplicação.”

Uma aplicação em Angular é baseada em componentes, ou seja, é possível encapsular comportamentos e interfaces, o que pode possibilitar melhor entendimento e maior reaproveitamento do código, tornando possível, inclusive, encapsular componentes dentro de outros componentes. (AFONSO, 2018)

Um exemplo do uso de componentes pode ser a criação de uma barra de ferramentas com alguns botões. Esta mesma barra de ferramentas pode ser reutilizada em outras telas da aplicação, tornando o processo de criação muito mais rápido e criativo, ou, ainda, projetar um componente de tela que inclui vários outros componentes dentro, como demonstra a figura 21.

Figura 21 - Exemplo de estrutura de componentes Angular.



Fonte: Página web *Blog Algaworks*¹⁹

2.6.3. React

Muitas das ferramentas e tecnologias de desenvolvimento surgem dentro de grandes corporações para depois ganhar o mundo. O *React* é mais um exemplo deste tipo de criação. Desenvolvido em 2011 dentro do *Facebook*, começou a ser usado no *feed* de notícias da rede social. No ano seguinte, já fazia parte das tecnologias do *Instagram* e de outras ferramentas. Oficialmente, em 2013, foi disponibilizado para toda a comunidade, ganhando notoriedade e popularização, devido a sua eficiência e flexibilidade para criação de interfaces de usuário (UI). (UDACITY, 2018)

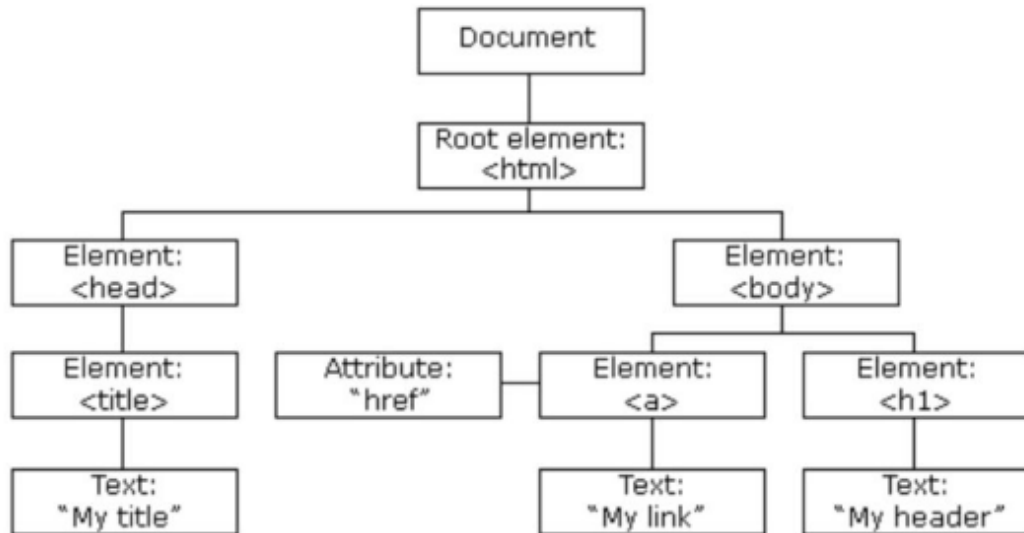
Um diferencial do *React* é o *DOM* virtual, o Modelo de Documento por Objetos (do inglês *Document Object Model*). É uma representação do documento, ou seja, quando ele é carregado no navegador, cria o *HTML DOM*, documento com estrutura em árvore, onde seus objetos são nós nesta estrutura como mostra a figura 22.

React é uma biblioteca *JavaScript open source*, mantida pelo *Facebook*, e com uma comunidade gigante. Sua estrutura também é baseada em componentes, e a codificação declarativa facilita o entendimento e manutenção do *software*. A principal função do *React* é organizar o conteúdo exposto ao usuário. “Sendo assim, ele veio para resolver problemas da camada visual, facilitando trabalhar com componentes

¹⁹ Disponível em: <https://blog.algaworks.com/o-que-e-angular/>. Acesso em: 09 maio 2020.

interativos e reutilizáveis para interfaces de usuário.” (FREIRES, 2019)

Figura 22 - Estrutura do *DOM*.



Fonte: Página web *Blog Geekhunters*.²⁰

Nesta estrutura de *DOM*, quando algum elemento precisa ser alterado, é necessário percorrer toda a árvore, ou procurar um nó específico. O problema é que o *DOM* não foi feito para isso e, quando for preciso atualizar documentos grandes, isto pode ser complexo e demorado. Neste sentido, o conceito de *DOM* virtual vem otimizar esta tarefa,

o *React* então compara o *Virtual DOM* com uma imagem do *DOM* feita antes da atualização e descobre o que realmente mudou, atualizando somente os componentes que mudaram de estado. Há um enorme ganho de performance aqui. criando uma cópia do *DOM* e, desta forma só altera o *DOM* original depois de finalizadas todas as alterações definitivas. (UDACITY, 2018)

Portanto, vemos no *React* uma ferramenta flexível, eficiente e prática, que se utiliza do conceito de software baseado em componentes para criação de interfaces de usuário, tornando a reutilização de código e a manutenção da aplicação mais

²⁰ Disponível em: <https://blog.geekhunter.com.br/um-guia-para-usar-react-js/>. Acesso em: 09 maio 2020.

fáceis. É, por este motivo, hoje, uma das bibliotecas *JavaScript* mais populares e requisitadas no universo do desenvolvimento *web*.

2.6.4. *CakePHP*

O *CakePHP* é um *framework* de desenvolvimento PHP, com a promessa de ser ágil. Surgiu quando Michal Tatarynowicz, um programador polaco, escreveu, apenas para uso pessoal, uma versão mínima para desenvolvimento em *PHP*, carinhosamente apelidou de “*cake*”. Tempos depois, ele publicou seu *framework* com a licença *MIT* e, desta forma, abriu o código para a comunidade de desenvolvedores, disseminando o *framework* e, ao mesmo, tempo propiciando atualizações e correções. (PORTAL GSTI, 2020)

O principal objetivo da ferramenta é proporcionar ao desenvolvedor uma forma estruturada de trabalhar, utilizando padrões de projetos de uma forma ágil e flexível. O *framework* segue o padrão MVC e possui uma vasta documentação.

O *CakePHP* pode ser instalado com um gerenciador de pacotes como o Composer, que auxilia a organizar e gerenciar dependências, deixando para o desenvolvedor o foco nas regras de negócio de sua aplicação.

Para ele rodar e iniciar, é preciso atender alguns requisitos básicos, que é o servidor *web*, *PHP*, e quatro extensões como *mbstring*, *intl*, *simplexml* e PDO. Tem boa integração com banco de dados como *MySQL*, *MariaDB*, *PostgreSQL*, além de *SQL Server* e *SQLite 3*. (PINTO, 2019)

Após o processo de instalação, é de responsabilidade do desenvolvedor configurar os dados de acessos ao banco de dados. Entres suas vantagens e características, podemos citar:

- Compatível com PHP5;
- Ferramentas que auxiliam gerar *JavaScript*, *AJAX*, *Forms HTML*;
- Padrão MVC;
- Licença flexível;
- Uso de *templates*;
- Validação;

- Funciona em qualquer subdiretório *web*, com pouca configuração Apache.

Todas estas características citadas, evidenciam sua segurança e agilidade no desenvolvimento *web*, além de manter um padrão de projeto estruturado. (TEIXEIRA, 2013)

2.7. BANCO DE DADOS

Nos dias atuais, cada vez mais nos vemos imersos em um mar de dados e informações e, mesmo sem perceber, convivemos e trabalhamos diariamente com inúmeros dados, como por exemplo: agenda do celular, conta do banco, planilhas de orçamentos, lista do supermercado etc. Todos estes dados e informações essenciais em nossas vidas particulares ou profissionais, dependem de uma estruturação e de alguns preceitos para que possam atender os requisitos básicos a que se designam.

Com a entrada dos computadores e dos sistemas de informação no meio empresarial, e com o crescente volume de dados, começaram a surgir os primeiros sistemas para atender as necessidades de processamento e manipulação dos dados. Neste sentido, os bancos de dados “foram concebidos com o objetivo de possibilitar posteriores acesso e manipulação de informações, de forma organizada e estruturada.” (CAYRES, 2015, p.2)

Segundo Cayres (2015, p.1), banco de dados pode ser definido como uma “coleção de dados relacionados que podem ser inseridos, atualizados e recuperados e que possuem um significado implícito.”

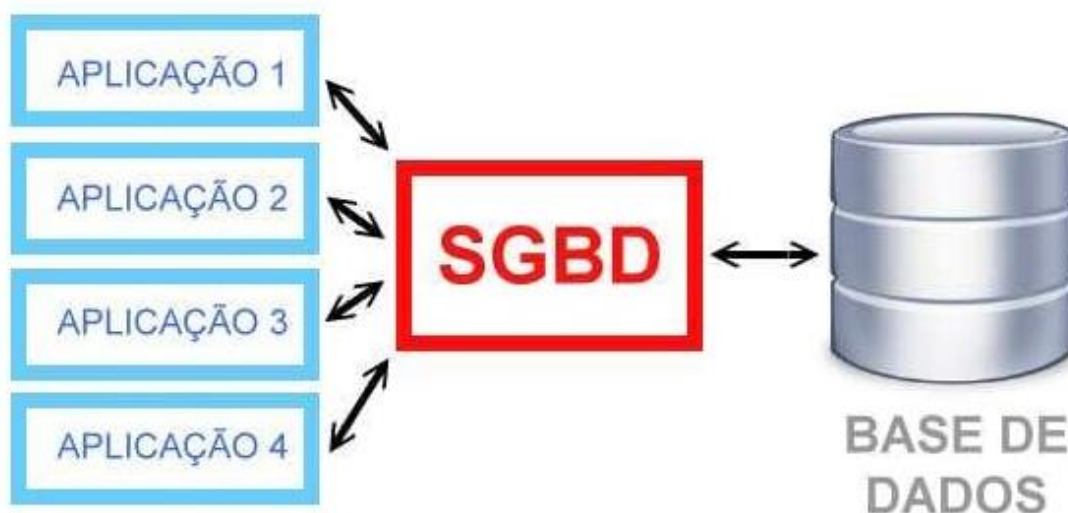
Entre os principais objetivos a serem alcançados por um sistema de banco de dados estão: gerenciar dados e informações, independente da complexidade do sistema e garantir a integridade dos dados, além de evitar redundância ou inconsistência dos dados, pois a duplicidade de dados é um problema comum em alguns sistemas de arquivos. Outro propósito é garantir a segurança dos dados, seja através de usuários e senhas, ou outras formas, além de oferecer facilidade no acesso a esses dados, através de *softwares* específicos. O acesso a estes dados deve se dar por meio de diversos recursos para facilitar o gerenciamento e a manutenção dos bancos de dados, prevendo possível migração, pois, às vezes, torna-se necessária a

troca de informações entre bancos de dados distintos. (CAYRES, 2015)

2.7.1. Sistemas de Gestão de Banco de Dados

Os SGBD's, Sistemas de Gestão de Base de Dados, oriundo do inglês *DBMS (Data Base Management System)*, são *softwares* responsáveis por criar e gerenciar um banco de dados. Estes dados podem ser oriundos de uma ou mais aplicações, conforme representado abaixo na figura 23.

Figura 23 - Representação de um SGBD.



Fonte: Página web Oficina da net.²¹

Os primeiros SGBD's surgiram no final dos anos 1960, juntamente com a necessidade de evolução do armazenamento e gerenciamento dos dados. Entre as características que um SGBD precisa oferecer, podemos destacar:

- Gerenciamento de grande quantidade de dados;
- Segurança dos dados;
- Controle de acesso;
- Evitar redundância e inconsistências;

²¹ Disponível em: <https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd>. Acesso em: 09 maio 2020.

- Esquematização, facilitando a compreensão dos dados;
- Interfaceamento, disponibilizando acesso de forma gráfica, e não somente texto;
- Garantir a integridade dos dados;
- Facilidade na migração;
- Sistemas de *backup*;
- Exemplos: *OracleDB*, *MySQL*, *PostgreSQL*, *MongoDB*. (OLIVEIRA, 2019)

Em relação aos SGBD's, utilizamos ferramentas visuais gráficas e linguagens de programação para facilitar o processo de construção e manipulação dos dados. Segundo Cayres (2015, p.3)

de forma simplificada, um *SGBD* faz a interface entre a camada física de armazenamento dos dados (discos, *storage*, métodos de acesso, *clustering* de dados etc.) e a sua organização lógica (instâncias) através de um determinado modelo de organização (esquema ou subesquema).

No que diz respeito à organização e estruturação dos *SGBD's*, atualmente, existem 6 modelos: modelo hierárquico, em rede, relacional, orientado a objetos, modelo objeto-relacional e *NoSQL (Not Only SQL)*. Para organizar, manipular e administrar um banco de dados, é utilizada uma linguagem padrão, *SQL (Structured Query Language – Linguagem de Consulta Estruturada)*, sendo aceita em quase todos os *SGBD's*. (OLIVEIRA, 2019)

Percebe-se que a forma como uma base de dados será construída e administrada, é essencial para o sucesso de um projeto de aplicação com banco de dados. Entretanto, compreender as diversas ferramentas, recursos e estruturas de dados, conhecendo as peculiaridades e características de cada modelo de *SGBD*, torna possível a melhor tomada de decisão em relação à organização e utilização dos dados.

3. PROCEDIMENTOS METODOLÓGICOS

Para que uma pesquisa científica obtenha resultado satisfatório, é necessário seguirmos um planejamento cuidadoso, além de possuir bases sólidas de conhecimento, que nos permitam reflexões que possivelmente solucionem as indagações propostas.

Neste sentido, segundo Silva e Menezes (2005, p. 9), “a metodologia tem como função mostrar a você como andar no ‘caminho das pedras’ da pesquisa”, proporcionando ao pesquisador um novo olhar sobre o mundo, com novas perspectivas e reflexões criativas.

3.1. CARACTERIZAÇÃO DA PESQUISA

Segundo abordagem proposta por Silva e Menezes, este estudo caracteriza-se como:

- Do ponto de vista de sua natureza, uma pesquisa aplicada, pois pretende buscar solução para a lacuna de ferramentas gráficas de simulação de jardins verticais. De acordo com a literatura “objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos.” Silva e Menezes (2005, p. 20)
- Quanto à forma de abordagem do problema, é considerada uma abordagem qualitativa, não sendo possível medir suas relações apenas em números. Não tem o foco na utilização de dados probabilísticos e utiliza diversos conceitos e experiências práticas vividas pela empresa parceira para o seu desenvolvimento.
- Quanto aos procedimentos técnicos adotados, esta pesquisa pode ser considerada bibliográfica, utilizando-se de conceitos e técnicas já descritos e validados em livros, artigos e materiais de pesquisa na internet, principalmente no estudo das ferramentas disponíveis no mercado do paisagismo, como também para modelagem e desenvolvimento de um protótipo *web*. Também, é um estudo de caso, pois “envolve o estudo profundo e exaustivo de um ou poucos objetos de maneira que se permita o seu amplo e detalhado

conhecimento.” Silva e Menezes (2005, p. 20)

- Em relação aos objetivos, esta pesquisa é exploratória. Busca dissecar o problema a fundo de forma a torná-lo claro, possibilitando avaliar os ganhos com as soluções já disponíveis no mercado, ao mesmo tempo que busca criar hipóteses ou desenvolver um protótipo de aplicação adequado para a resolução do problema.

3.2. DELIMITAÇÃO E POPULAÇÃO DA PESQUISA

A pesquisa consiste em um estudo bibliográfico acerca das ferramentas gráficas atualmente disponíveis para simulação de jardins verticais e, ao mesmo tempo, é um estudo de caso relacionado à modelagem e desenvolvimento de um protótipo de aplicação *web* para simulação de jardins verticais.

Abrange um estudo dos *softwares* gráficos disponíveis no mercado para simulação de jardins verticais, desenvolvendo uma análise preliminar sobre suas características, recursos disponíveis e valor de mercado, além de apropriar-se de alguns conceitos e conhecimentos em boas práticas de desenvolvimento, para se necessário modelar e desenvolver um protótipo de uma aplicação que simule jardins verticais, atendendo aos requisitos do projeto e, agregando às ferramentas disponíveis.

As ferramentas e técnicas utilizadas no desenvolvimento da pesquisa foram escolhidas de acordo com a aptidão do desenvolvedor, buscando equilibrar o conhecimento prévio e as necessidades do projeto.

3.3. ETAPAS DA PESQUISA

Para atingir os objetivos da pesquisa, fez-se uma subdivisão dos processos em etapas, de forma a proporcionar otimização e qualidade nos resultados.

Entendendo o problema de pesquisa e os objetivos propostos, primeiramente tratamos da fundamentação teórica, trazendo conceitos e técnicas fundamentais para o real entendimento do contexto em que está inserido o protótipo, abordando as ferramentas e tecnologias necessárias para seu desenvolvimento.

Na etapa seguinte, buscamos fazer um levantamento dos requisitos funcionais e não funcionais do protótipo, descrevendo suas características e funcionalidades desejáveis, assim como os recursos e ferramentas que serão oferecidos ao usuário da aplicação.

Após a etapa de definição dos requisitos, fez-se necessário verificar as ferramentas e *softwares* já existentes no mercado. Nesta etapa, também se faz necessário um estudo sobre cada ferramenta disponível, verificando suas características e peculiaridades e, se atendem todas as funcionalidades do protótipo requerido.

Após análise das ferramentas gráficas disponíveis atualmente, e na hipótese de não atenderem todos os requisitos do projeto proposto, definimos o escopo e as tecnologias necessárias para a modelagem e desenvolvimento da aplicação.

Finalmente, concluímos a pesquisa desenvolvendo e descrevendo etapas da modelagem e desenvolvimento do protótipo requerido, citando alguns pontos de destaque na escolha das ferramentas utilizadas, assim como o desenvolvimento da codificação.

4. DESENVOLVIMENTO DO TRABALHO E APRESENTAÇÃO DE RESULTADOS

4.1. REQUISITOS

A aplicação pretendida com este projeto busca apresentar uma possível solução para a simulação gráfica de jardins verticais em ambientes reais. A aplicação tem como principal escopo trazer maior realismo nas simulações de jardins verticais, proporcionando, de uma forma prática e rápida, melhor visualização e projeção de possíveis criações e projetos de jardins verticais.

O protótipo não tem foco em armazenar dados de clientes ou logins de acesso. Prioriza o uso irrestrito de usuários sem necessidade de instalação de *software*, licenças ou recursos avançados de hardware. Objetiva a experimentação e simulação de jardins verticais, bem como a disseminação desta técnica de paisagismo.

No campo do paisagismo, os projetos de jardins verticais tornaram-se uma vertente importante de grande expansão no mercado. Proporcionando beleza, requinte e sofisticação, os jardins verticais redecoram ambientes, além de proporcionar benefícios térmicos e de bem-estar ao ser humano. Desta forma, as tecnologias e ferramentas gráficas precisam estar em consonância e constante evolução para dar suporte ao desenvolvimento das atividades do paisagista.

Com base nas necessidades apresentadas pela empresa parceira, na apresentação do problema e nos objetivos propostos, alguns requisitos são fundamentais para o sucesso da aplicação pretendida e, portanto, assim definidos.

4.1.1. Requisitos funcionais

A aplicação deve atender os seguintes requisitos funcionais:

- Permitir ao usuário visualização do catálogo de plantas disponíveis com suas respectivas características para uso nas simulações;
- Contar com uma aba para visualização de exemplos de jardins verticais já executados, proporcionando maior inspiração aos usuários;
- Permitir ao usuário fazer upload de imagens com ambientes reais e utilizá-las na projeção e simulação de jardins verticais;

- O sistema deve contar com a funcionalidade de selecionar as plantas desejadas, arrastá-las e soltar sobre o ambiente de simulação através de sobreposição de imagens ou fotomontagens;
- Salvar a imagem da simulação concluída com a aplicação em formato PNG;

4.1.2. Requisitos não funcionais

A aplicação deve atender os seguintes requisitos não funcionais:

- A aplicação deve ser compatível com os principais navegadores atualmente no mercado;
- O sistema deve ser responsivo, facilitando sua utilização em dispositivos móveis.
- A aplicação não deve solicitar instalação de qualquer software ou componente por parte do cliente;
- Sabendo dos benefícios gerados pelos jardins, buscando difundir sua utilização e ciente das dificuldades de projeção e simulação na criação de jardins verticais, a ferramenta deve ser gratuita, sem coleta de dados do usuário e passível de utilização tanto pelo projetista quanto por um usuário comum.

4.2. ANÁLISE DAS FERRAMENTAS JÁ EXISTENTES NO MERCADO

Com base nos requisitos pretendidos para a aplicação, algumas ferramentas já disponíveis no mercado foram analisadas. A maior parte das aplicações são pagas e necessitam de recursos de hardware para seu pleno funcionamento.

As soluções analisadas possuem versão em português e, mesmo não sendo objetivo do projeto, também se analisou aplicativo *mobile* gratuito disponível para fins paisagísticos.

Todos as cotações de preços, assim como a avaliação das características, recursos e performance das ferramentas citadas abaixo na análise individual, foram feitas em 21 de janeiro de 2021.

4.2.1. DynaScape

Os *softwares DynaScape* trazem soluções de aplicação paga baseada em CAD, necessitando-se de licença de uso para cada estação de trabalho para sua utilização. Possui um pacote de soluções para paisagismo, podendo ser adquirido individualmente ou em módulos.

Entre as soluções estão o *Horticopia Pro*, focado em projetos paisagísticos. O *software* é um catálogo com enorme banco de dados de plantas com diversas informações referentes de cada planta como: cores, formas, origem etc. Também permite a inclusão de novas espécies pelo usuário. Entretanto, para ter melhor aproveitamento, pode ser usado com outro módulo, o *DynaScape Design*, que é um software CAD que proporciona desenhos ricos em recursos e apresentação em planta baixa.

Figura 24 - Apresentação de projeto com DynaScape Design e Horticopia Pro.



Fonte: Página web DynaScape.com.²²

²² Disponível em: <https://www.dynascape.com/solutions/design/horticopia-pro/>. Acesso em: 21 jan. 2021.

Os preços de licenças variam de acordo com o módulo adquirido. Também possui edição para estudante por um valor de \$187,50 e, sendo contemplado com todas as funcionalidades do *software* profissional. No entanto, esta licença tem prazo de 12 meses, após este período a licença expira e torna-se necessário a compra da licença profissional. A licença profissional do *DynaScape Design* em pacote promocional está pelo preço de \$43,95 mensais. As soluções *DynaScape* necessitam de licença individual para cada estação de trabalho, além de manter uma assinatura do *software*. (DYNASCAPE.COM, 2021)

Outra solução apresentada pela *DynaScape* é o *Shaderlight*, *software* de renderização 3D para *SketchUp*. É direcionado para profissionais paisagistas, oferecendo diversos recursos para apresentação foto realística, como: alteração da localização do sol para alterar sombras e sombreamento, efeitos de iluminação noturna, permite criação de texturas de água para piscinas, entre outros recursos.

O *DynaScape* e suas soluções não se coadunam com os objetivos pretendidos no projeto proposto, pois, apesar de possuir diversos recursos e apresentação realista com imagens 3D, as aplicações são extremamente profissionais, tornando sua utilização pelo usuário comum complexa, demandando tempo e treinamento para domínio da ferramenta, além de não ser gratuita e não ser uma aplicação *web*, ou seja, necessita de instalação em computador pessoal.

4.2.2. Edificius LAND

O *software* desenvolvido pela empresa ACCA *Software* tem como público-alvo arquitetos e engenheiros. Busca, por meio de suas soluções, aprimorar o setor de construção com aumento de produtividade e ganho de qualidade.

O *Edificius Land* otimiza o trabalho do arquiteto ou projetista, integra o projeto arquitetônico e espaços exteriores como projetos arquitetônicos de edifícios, parques urbanos, jardins, entre outros exemplos, e possibilita projetar com elevado nível de detalhes e realismo.

Dentre suas características e recursos, destacam-se a possibilidade de modelar terrenos diretamente do *Google Maps* ou com arquivos no formato DWG.

Possui uma rica biblioteca de objetos e plantas em 2D e 3D, fornece toda a documentação do projeto como planimetrias, cortes; gera legendas automaticamente a partir do modelo 3D; disponibiliza orçamento automático dos valores do projeto; fornece renderizações foto-realistas para apresentação do projeto e, ainda, possibilita navegar no projeto com a realidade virtual. Abaixo, temos um exemplo de projeto criado com o *software* (ACCASOFTWARE, 2021)

Figura 25 - Navegando com realidade virtual no projeto com Edificius Land.



Fonte: Página web DynaScape.com.²³

A aplicação é extremamente completa para uso de profissionais de engenharia e arquitetura. Funciona com licença de uso e a solução mais básica custa €27/mensal. O *software* exige alguns recursos de *hardware* como processador Core i5 ou superior, 8GB de RAM e placa de vídeo, itens estes que tornam seu uso mais restrito a profissionais do ramo e a aplicação um pouco mais complexa para o usuário comum.

4.2.3. AuE PhotoLANDSCAPE

O programa de paisagismo focado em apresentação de projetos com fotomontagem é uma das soluções apresentadas pela AuE *Software*. A aplicação

²³ Disponível em: <https://www.accasoftware.com/ptb/programa-para-paisagismo>. Acesso em: 21 jan. 2021.

disponibiliza recursos de alto padrão visual com praticidade e rapidez, proporcionando qualidade e agilidade na criação e apresentação dos projetos de paisagismo.

Os principais recursos que a aplicação oferece são: catálogo com mais de 2.300 plantas ornamentais, com detalhes e características de cada planta, todos os itens do banco de dados, sem fundo, permitindo posicionar os itens sobre a foto do terreno e aplicar o efeito desejado. Possui integração com outro *software* da empresa, *CalcLANDSCAPE* (vendido separadamente) que, por sua vez possibilita gerar o orçamento do projeto com todos os itens escolhidos, com efeitos noturno e diurno, jardim vertical, efeitos de perspectiva, sombreamento etc. (AUESOFTWARE, 2018)

Abaixo, temos um exemplo de projeto criado com o programa, e disponibilizado no site da empresa.

Figura 26 - Exemplo de projeto criado com PhotoLANDSCAPE.



Fonte: Página web auesoftware.com.²⁴

A ferramenta não funciona de forma online, sendo necessário sua instalação

²⁴ Disponível em: <https://auesoftware.com/figuras/53-013.jpg>. Acesso em: 21 jan. 2021.

em um computador pessoal. O custo de licença anual é de R\$ 1.690,00 à vista. Se desejar comprar um curso preparatório online de 8 horas-aula, para melhor conhecimento e desenvolvimento dos projetos utilizando a ferramenta, o investimento sobe para R\$ 2.300,00. Entre os principais requisitos para o funcionamento estão: Processador *Quad Core* ou superior, placa gráfica 256MB ou superior e sistema operacional *Windows 7, 8, 8.1 e 10*.

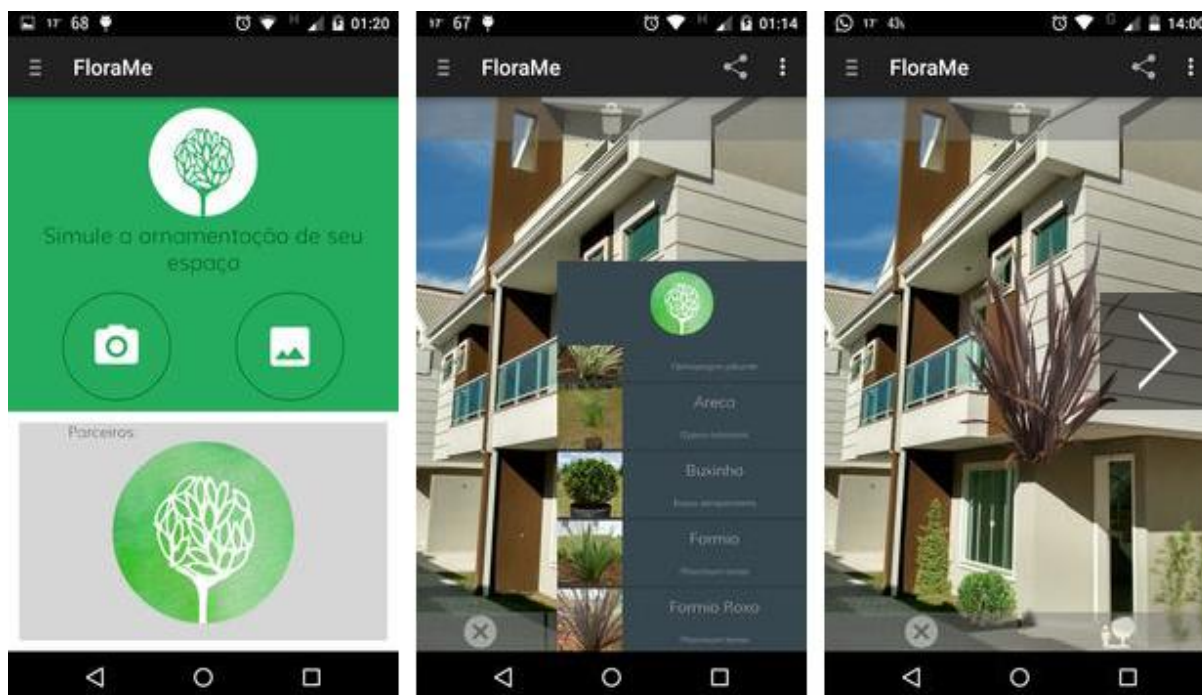
4.2.4. App FloraMe

O aplicativo está disponível gratuitamente para *download* na *Play Store* e funciona somente em dispositivos com sistema *Android*.

Com a ferramenta é possível simular diversas combinações de plantas ornamentais em ambientes internos e externos. Apesar de ocorrer alguns travamentos, o aplicativo permite utilizar imagens reais ou arquivadas no dispositivo como plano de fundo e conta com uma relação de plantas ornamentais para uso nas simulações divididas em grupos: forrações, arbustos, árvores, palmeira, ciprestes, touceiras, dracenas, cercas-vivas, agaves e cactos, orquídeas e bromélias, internas e vasos. (XUBNAHA CO, 2019)

Abaixo, temos um breve exemplo das telas do aplicativo e recursos disponíveis com o aplicativo.

Figura 27 - Exemplo de utilização com o aplicativo FloraMe.



Fonte: Página web apkpure.com.²⁵

Após a conclusão da simulação com as plantas que o usuário deseja adicionar na montagem, o aplicativo também permite salvar ou compartilhar a criação.

4.2.5. Conclusão da análise das ferramentas

Com base nos testes e experimentação das ferramentas disponíveis, notamos a escassez de ferramentas gráficas gratuitas para simulação de jardins verticais.

As soluções *DynaScape* e *Edificius Land* são aplicações bastante robustas e com inúmeras funcionalidades, voltadas principalmente para profissionais de engenharia, arquitetura e paisagismo. Exigem alguns requisitos de *hardware* para seu funcionamento. Também possuem integrações CAD com outros *softwares*, além de um grande banco de dados. No entanto, exatamente por estas mesmas características, torna seu uso mais complexo para o usuário comum, além de ser um

²⁵ Disponível em: <https://apkpure.com/br/florame-landscaping-made-easy/com.xubnaha.florame>. Acesso em: 21 jan. 2021.

software pago com um custo relativamente alto, principalmente, se considerarmos que a licença é individual para cada estação de trabalho.

A solução apresentada pela AuE *Software* atende parcialmente os requisitos pretendidos pelo projeto. O *software* PhotoLANDSCAPE permite a apresentação de projetos e simulação de jardins verticais com grande realismo por meio da técnica de fotomontagem, possuindo um vasto banco de dados de espécies de plantas disponíveis, efeitos de sombreamento e efeito 3D.

Já entre os pontos negativos da aplicação, podemos citar o fato de não ser gratuita e não ser uma aplicação *web*, ou seja, necessita de instalação em uma estação de trabalho, com alguns requisitos de *hardware* e licença individual por máquina. Desta forma, inviabiliza seu uso irrestrito pelos clientes ou usuários finais que pretenderem fazer suas simulações e experimentações de jardins verticais antes da definição de um projeto.

No caso do aplicativo FloraMe, este apresenta algumas funcionalidades desejadas pelo projeto, é gratuito e permite utilizar fotos reais para simulação por meio de fotomontagens com as plantas disponíveis. Apesar de não ser específico para simulação de jardins verticais, não impossibilita sua utilização. No entanto, a aplicação tem o foco voltado para simulação de ambientes com plantas ornamentais, funciona somente em dispositivos *Android* e, não contempla a maior parte das plantas utilizadas em projetos de jardins verticais pela empresa parceira, além de apresentar alguns travamentos durante sua utilização.

Como o objetivo do projeto é uma solução *web* que funcione nos principais navegadores de internet, sem a necessidade de instalação de aplicativo ou *software* pelo cliente, que seja de fácil utilização, gratuita, e que foque na simulação de jardins verticais com fotomontagens, contemplando as plantas disponíveis para os projetos de jardins verticais utilizadas pela empresa parceira, notamos que nenhuma solução atende plenamente os requisitos pretendidos, donde conclui-se que há a necessidade de desenvolver um protótipo de aplicação específico para este fim.

4.3. TECNOLOGIAS UTILIZADAS

Partindo da premissa de uma aplicação de pequeno porte e, sem utilização de

banco de dados, decidimos por utilizar um *framework* para desenvolvimento *web*, que facilite e agilize o desenvolvimento da aplicação pelo arcabouço de suporte que fornece ao desenvolvedor, constituindo-se com um esqueleto para as demais funcionalidades do protótipo pretendido.

O *Bootstrap* é um *framework* bastante intuitivo, uma das principais ferramentas gratuitas, juntando HTML, CSS e JavaScript para desenvolvimento *web* responsivo. Por já ter sido utilizada em outros projetos pelo desenvolvedor, ser uma ferramenta gratuita e que possibilita uma rápida curva de aprendizado mesmo para iniciantes, decidimos por utilizá-lo como base para codificação do protótipo.

Os demais *frameworks* citados na fundamentação teórica, mesmo que com grande potencial e recursos, foram descartados para utilização neste projeto, pois demandariam, primeiramente, de um estudo da ferramenta, para posterior utilização por parte do desenvolvedor, o que poderia gerar atraso na entrega.

As linguagens de programação utilizadas foram HTML, que é a linguagem mais utilizada para hipertexto, CSS para estilização da página (como por exemplo, cor das letras, design dos botões etc.), e, ainda, utilizamos JavaScript para criar interações com o usuário, dotando a aplicação de comportamentos dinâmicos, como na funcionalidade desejada de fazer upload de imagens ou selecionar e arrastar as plantas no cenário de simulação.

Desta forma, analisando os requisitos desejáveis juntamente com a aptidão do desenvolvedor, não utilizamos nenhum *framework* ou linguagem PHP, nem banco de dados ou SGBD, já que a aplicação não tem foco em armazenar dados ou imagens, apenas rodar o software como serviço pelo lado do cliente sem utilização de formulários, coleta de dados ou logins de acesso.

4.4. FERRAMENTAS UTILIZADAS

Algumas ferramentas foram necessárias e fundamentais para a modelagem e desenvolvimento ágil do protótipo. Todas as ferramentas utilizadas são gratuitas e estão disponíveis na internet.

4.4.1. Wamp Server

O Wamp Server é uma aplicação ou um pacote de *softwares* para desenvolvimento *web*. A sigla é um acrônimo de *Windows, Apache, MySQL, PHP*, ou seja, instalando o WAMP no computador, automaticamente é instalado esse pacote de aplicações que permite tornar sua máquina um ambiente de desenvolvimento *web*.

O WAMP é um conjunto derivado do LAMP (o L é de Linux). A única diferença entre eles é que o WAMP é usado para o Windows. Enquanto isso, o LAMP é usado para sistemas operacionais baseados em Linux (Ubuntu, por exemplo). (LONGEN, 2019)

Com o WampServer, é possível criar aplicações *web* de uma forma bastante intuitiva, além de disponibilizar a ferramenta PhpMyAdmin utilizada para o gerenciamento do seu banco de dados MySQL.

Agindo como uma máquina virtual em seu computador, o Wamp permite que você crie e experimente diversas opções em seu site ou aplicação antes de colocar tudo online na *web*, além de agilizar o processo de trabalho para designers e desenvolvedores sem necessitar ficar fazendo uploads de arquivos para seus sites. (LONGEN, 2019)

4.4.2. Astah UML

Astah Community é um *software* para modelagem UML (Unified Modeling Language – Linguagem de Modelagem Unificada) com suporte a UML 2, desenvolvido pela Change Vision, Inc. Anteriormente conhecido por JUDE, um acrônimo de *Java and UML Developers Environment* (Ambiente para Desenvolvedores UML e Java). (LIMA, 2016)

Astah está disponível em versão gratuita para estudante e paga na versão Professional, permitindo a diagramação UML com uma interface amigável e prática, e tornando-se intuitivo com um pouco de conhecimento em UML. Disponibiliza para desenvolvimento diversos diagramas como: diagrama de classes, casos de uso, implantação, componentes etc. Também permite exportar os diagramas desenvolvidos.

4.4.3. Visual Studio Code

Disponível para *Windows, Linux e Mac*, o *Visual Studio Code* é um editor de código gratuito com foco em construção de aplicações *web*.

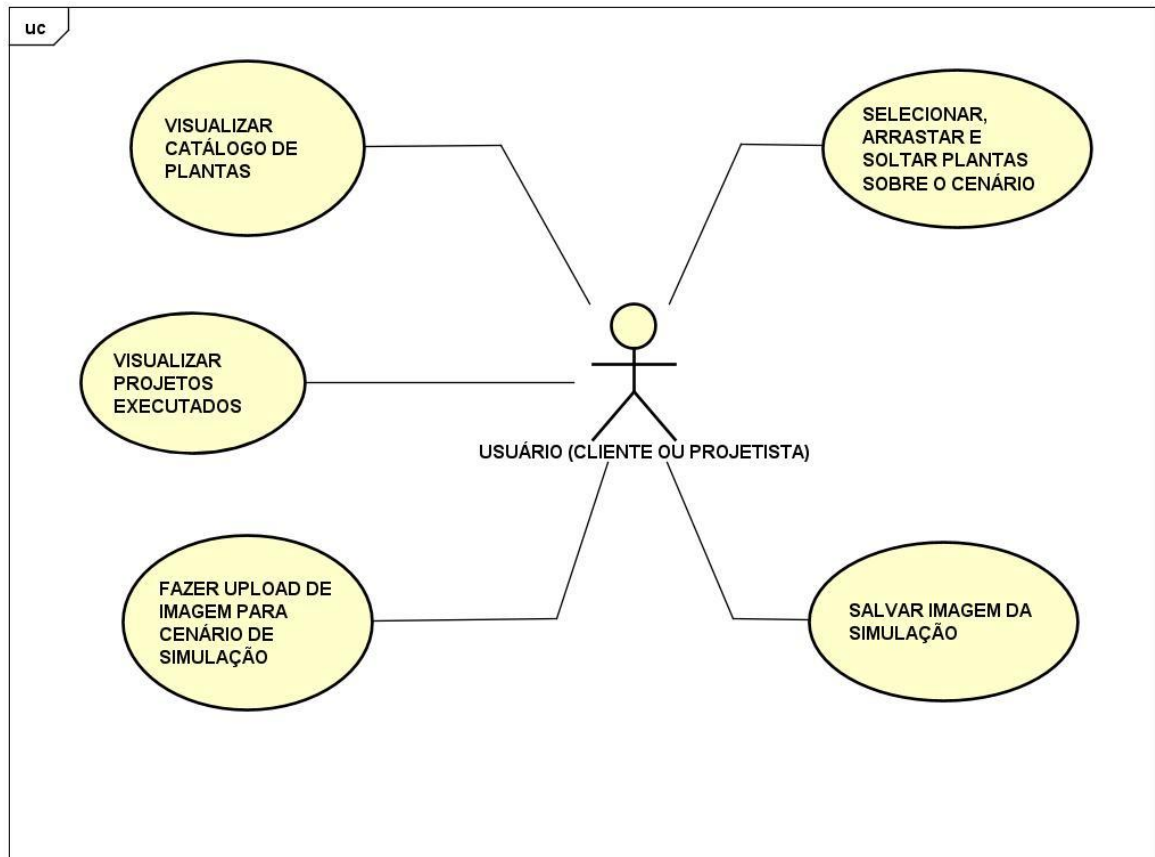
É um editor com excelente performance, sem lentidão ou travamentos e, ainda oferece diversos recursos importantes como: autocompletar de código, comandos git integrados, *software* passível de personalização de idiomas e temas. Também oferece opção de adicionar extensões para serviços adicionais. (EVANGELISTA, 2018)

4.5. DIAGRAMA DE CASO DE USO

Dentre todos os diagramas da UML, o diagrama de caso de uso é o mais comum, simples e prático para análise e levantamento de requisitos, e proporciona uma visão ampla das funcionalidades do sistema e a interação com os atores ou usuários desse sistema. Por isso mesmo, também facilita a comunicação entre o cliente e os desenvolvedores de *software*. (GUEDES, 2011)

Para elaboração dos casos de uso do protótipo, foram aplicados os conceitos de UML definidos por Guedes (2011) e utilizada a ferramenta Astah. Abaixo, temos o diagrama com os principais casos de uso e suas tabelas de documentação descritivas.

Figura 28 - Diagrama de casos de uso da aplicação.



Fonte: Elaborado pelo autor.

Tabela 1 – Documentação do caso de uso “Visualizar catálogo de plantas”.

VISUALIZAR CATÁLOGO DE PLANTAS / UseCase Description	
ITEM	VALUE
UseCase	VISUALIZAR CATÁLOGO DE PLANTAS
Summary	Este caso de uso permite ao usuário visualizar todo o catálogo de plantas disponíveis que podem ser utilizadas na simulação de jardim vertical.
Actor	USUÁRIO (CLIENTE OU PROJETISTA)
Precondition	Acessar a aplicação.
Postcondition	
Base Sequence	1 - Ação do ator: Clica no ícone paleta vegetal (sombra ou sol); 2 - Ação do sistema: A aplicação abrirá uma aba com cards contendo as plantas disponíveis e suas respectivas características;
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Fonte: Elaborado pelo autor.

Tabela 2 – Documentação do caso de uso “Visualizar projetos executados”.

VISUALIZAR PROJETOS EXECUTADOS / UseCase Description	
ITEM	VALUE
UseCase	VISUALIZAR PROJETOS EXECUTADOS
Summary	Este caso de uso permite ao usuário visualizar projetos já executados e, que tiveram grande destaque, permitindo ao usuário inspirar-se nestes projetos para posteriormente experimentar suas próprias criações de jardim vertical.
Actor	USUÁRIO (CLIENTE OU PROJETISTA)
Precondition	Acessar a aplicação.
Postcondition	
Base Sequence	1 - Ação do ator: Clicar no ícone inspirações; 2 - Ação do sistema: A aplicação abrirá uma aba com cards colapsáveis contendo os jardins verticais de maior destaque já executados pela empresa parceira, é possível visualizar o conteúdo de cada card clicando sobre ele e, ou escondendo seu conteúdo clicando novamente sobre ele.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Fonte: Elaborado pelo autor.

Tabela 3 – Documentação do caso de uso “Fazer upload de imagem para cenário de simulação”.

FAZER UPLOAD DE IMAGEM PARA CENÁRIO DE SIMULAÇÃO / UseCase Description	
ITEM	VALUE
UseCase	FAZER UPLOAD DE IMAGEM PARA CENÁRIO DE SIMULAÇÃO
Summary	Este caso de uso permite que o usuário faça upload de imagens reais, com o intuito de utilizá-las como cenário da simulação.
Actor	USUÁRIO (CLIENTE OU PROJETISTA)
Precondition	Acessar a aplicação.
Postcondition	
Base Sequence	1 - Ações do ator: clicar na área "clique e selecione sua imagem"; 2 - Ações do sistema: Abrirá o sistema de arquivos do dispositivo utilizado, para que o usuário o selecione o arquivo de imagem desejado para fazer upload; 3 - Ações do ator: Seleciona a imagem desejada e clica em ok; 4 - Ações do sistema: automaticamente a imagem selecionada aparecerá como cenário de fundo na tela de simulação;
Branch Sequence	
Exception Sequence	Caso o usuário mude de ideia e, decida trocar a imagem do cenário de simulação, ele deverá clicar no botão "REMOVER", desta forma a imagem é excluída e, o usuário deverá recomençar novamente no passo 1 do fluxo principal descrito acima.
Sub UseCase	
Note	

Fonte: Elaborado pelo autor.

Tabela 4 – Documentação do caso de uso “Selecionar, arrastar e soltar plantas sobre o cenário”.

SELECIONAR, ARRASTAR E SOLTAR PLANTAS SOBRE O CENÁRIO / UseCase Description	
ITEM	VALUE
UseCase	SELECIONAR, ARRASTAR E SOLTAR PLANTAS SOBRE O CENÁRIO
Summary	Este caso de uso permite ao usuário selecionar as plantas desejadas e adicionar na sua simulação de jardim vertical.
Actor	USUÁRIO (CLIENTE OU PROJETISTA)
Precondition	Acessar a aplicação de simulação e, ter feito upload da imagem que servirá de cenário para a simulação.
Postcondition	
Base Sequence	1 - Ação do ator: Após fazer upload da imagem de cenário de simulação, clicar no ícone de plantas de sol ou plantas de sombra; 2 - Ação do sistema: O sistema abrirá uma lista dropdown com as plantas disponíveis; 3 - Ação do ator: O usuário clica na planta desejada; 4 - Ação do sistema: A planta clicada irá surgir no cenário de simulação; 5 - Ação do ator: O ator poderá selecionar a planta com 1 clique, arrastá-la até a posição desejada no cenário e soltá-la, montando o jardim vertical de acordo com suas preferências.
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Fonte: Elaborado pelo autor.

Tabela 5 – Documentação do caso de uso “Salvar imagem da simulação”.

SALVAR IMAGEM DA SIMULAÇÃO / UseCase Description	
ITEM	VALUE
UseCase	SALVAR IMAGEM DA SIMULAÇÃO
Summary	Este caso de uso permite ao usuário salvar a imagem da simulação executada na aplicação.
Actor	USUÁRIO (CLIENTE OU PROJETISTA)
Precondition	Acessar a aplicação.
Postcondition	
Base Sequence	1 - Ações do ator: clicar no botão salvar; 2 - Ações do sistema: O sistema deverá fazer um "print" da simulação executada e mostrar abaixo da área de simulação; 3 - Ações do ator: clicar no print com botão direito do mouse e selecionar salvar como, determinando local e o nome do arquivo que será salvo. (O arquivo será salvo em formato PNG).
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Fonte: Elaborado pelo autor.

4.6. MODELAGEM E DESENVOLVIMENTO DO PROTÓTIPO

Inicialmente, foram utilizados alguns modelos formatados do *Bootstrap*, com o objetivo de dar agilidade ao desenvolvimento do protótipo, ao mesmo tempo em que proporciona uma aplicação responsiva.

Na página principal do protótipo, onde deve ocorrer a simulação, priorizamos uma tela *clean*, de forma a não poluir visualmente o ambiente de trabalho com muitas informações ou ícones desnecessários. Desta forma, criamos uma barra de menu lateral onde ficam alocados os ícones da aplicação.

No centro da página criamos um espaço onde o usuário poderá clicar e fazer upload da imagem do ambiente de simulação para a página da aplicação. Após selecionar a foto ou imagem, esta aparecerá na tela de simulação e, poderá ser utilizada na simulação do jardim vertical, utilizando da técnica de sobreposição de imagens, deslocando as plantas selecionadas pelo usuário sobre a imagem do cenário.

Também, é possível remover ou trocar a imagem do ambiente de simulação a qualquer momento, clicando no botão “remover” e selecionando outra imagem. Outro ponto de destaque no protótipo, é que tornamos possível com um clique fazer um print da simulação executada pelo usuário e baixá-la em formato PNG, proporcionando salvar as diversas criações ou experimentações do usuário, gerando maior familiaridade com o projeto de jardim vertical e interação com suas criações.

Percebemos que, durante o processo de simulação, o usuário pode querer consultar o catálogo de plantas disponíveis e adequadas para o ambiente de simulação em questão. Para atender esta demanda, na barra de menu lateral, a paleta vegetal, quando clicadas, abre uma nova aba com as informações e imagens de cada planta, proporcionando ao usuário navegar na simulação e alternar, quando quiser, na aba de paleta vegetal.

Na barra de menu constam os seguintes ícones: plantas de sol, plantas de sombra e plantas medicinais. Clicando nestes ícones, abre-se uma lista *dropdown*, com suas respectivas plantas, que podem ser utilizadas na sobreposição de imagens para a simulação do jardim vertical. Também estão contemplados nesta barra de menu o ícone paleta vegetal, inspirações, *refresh*, ajuda, e um botão de *WhatsApp*

que permite ao usuário compartilhar o *link* da aplicação através do *WhatsApp*, fomentando seu uso e divulgação pelo maior número possível de usuários.

A paleta vegetal ou catálogo de plantas que podem ser utilizadas nos projetos, foi desenvolvida em aba sobre modelos de *cards Bootstrap*. Em cada *card* são apresentadas as principais características de cada planta, como nome científico, origem, tamanho e desenvolvimento, além de sua imagem de exemplo.

No protótipo, também determinamos uma aba “Inspirações” para demonstrar exemplos de jardins verticais já executados pela empresa parceira, proporcionando, assim, maior inspiração na criação de novos projetos e simulações pelos novos usuários, além de divulgar os trabalhos já executados.

Para a construção da aba “inspirações”, cada projeto de jardim vertical é apresentado em um item colapsável. Para chegarmos a este resultado, criamos um grupo de itens colapsáveis com diversos projetos já executados e finalizados com suas respectivas informações.

O ícone de ajuda, quando clicado, abre uma aba com um vídeo explicativo de como utilizar a ferramenta de simulação e seus componentes.

No desenvolvimento do código do protótipo, foram utilizados trechos de códigos abertos e disponíveis na *web*, além de alguns componentes *Bootstrap* e alguns *scripts* necessários para desenvolver os comportamentos da página.

Para tornar a aplicação disponível na *web*, foram feitas pequenas configurações no WAMP, determinando o arquivo que será exibido na página principal e as portas de comunicação da aplicação com a internet. Como a aplicação é um protótipo, e, não temos nenhum domínio registrado na internet, optamos por acessá-la através do IP e porta de comunicação configurada (150.162.162.144:8080), como mostra figura abaixo, com parte do arquivo de configuração do Apache/WAMP.

Figura 29 - Arquivo de configuração Wamp Server/Apache.

```
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName 150.162.162.144:8080

HostnameLookups Off

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:\wamp\www\Verticaliza"
```

Fonte: Elaborado pelo autor.

5. CONSIDERAÇÕES FINAIS

A presente pesquisa teve como premissa agregar outras possibilidades à lacuna de ferramentas gráficas utilizadas para simulação de jardim vertical existentes no mercado. Para tanto, foram definidos os requisitos desejáveis da aplicação e, posteriormente, foi executada uma pesquisa bibliográfica das ferramentas de paisagismo pré-existentes, fazendo uma análise das aplicações com suas funcionalidades básicas e requisitos necessários para sua utilização.

5.1. EM RELAÇÃO AO OBJETIVO GERAL

Notoriamente, há uma escassez de ferramentas gráficas gratuitas para paisagismo, especialmente das ferramentas gráficas voltadas para projetos de jardins verticais. Neste sentido, a pesquisa buscou agregar mais uma opção às poucas ferramentas existentes. Afinal, dentre as ferramentas analisadas, não foram encontradas ferramentas gratuitas para simulação de jardins verticais, inviabilizando seu uso irrestrito, além de que grande parte das ferramentas necessita de alguns recursos de *hardware*, aumentando consideravelmente os custos com sua adoção ou implementação. Sendo assim, modelamos e desenvolvemos uma aplicação *web* gratuita que proporciona simular projetos de jardins verticais, sem necessitar de licença de uso, instalação de *software* ou recursos mais avançados de *hardware*, podendo ser utilizada tanto por projetistas paisagistas, quanto pelo usuário comum.

5.2. EM RELAÇÃO AOS OBJETIVOS ESPECÍFICOS

Com o auxílio da empresa parceira, foi possível determinar os principais requisitos de uma aplicação que auxilie na simulação de jardins verticais. A pesquisa mostrou a evolução do paisagismo e suas nuances, destacando a necessidade de consonância e evolução das técnicas e ferramentas que auxiliam o profissional, como de suma importância para alcançar o sucesso nos projetos de jardins verticais.

O levantamento das ferramentas disponíveis no mercado do paisagismo,

mostrou que poucas são as ferramentas voltadas ao propósito de simular jardins verticais. A grande maioria das aplicações são pagas e apresentam um custo relativamente alto, levando em consideração que possuem licenças individuais por estação de trabalho. Outro ponto a ser destacado é que são ferramentas robustas, voltada para profissionais, e que necessitam de recursos consideráveis de hardware, dificultando sua utilização em larga escala, com equipamento computacional mínimo, e exigindo um conhecimento considerável por parte do usuário.

O desenvolvimento do protótipo gratuito fez-se necessário devido aos custos e às limitações das ferramentas existentes, e possibilitou o uso de alguns conhecimentos adquiridos durante o curso de CSTGTI, destacando-se o conhecimento apresentado nas unidades curriculares de programação *web*, processos de desenvolvimento de sistemas, gerenciamento de projetos e lógica de programação.

O curso tem como objetivo proporcionar diversas habilidades ao egresso, entre elas: “Prospectar novas tecnologias na área de TI, bem como facilitar sua incorporação a organização e selecionar adequadamente métodos, técnicas e ferramentas para o desenvolvimento de soluções em TI” (PLANO PEDAGÓGICO DO CURSO, 2017, p.6), conhecimentos e habilidades que foram amplamente utilizados neste projeto.

É de se supor que a utilização deste protótipo irá facilitar a simulação gráfica de projetos de jardins verticais, proporcionando maior familiaridade com os jardins, maior conhecimento sobre as espécies utilizadas, aumentando o poder de tomada de decisão sobre projetos futuros, permitindo pré-ajustes do projeto antes da execução, além de fomentar a disseminação desta técnica do paisagismo.

O protótipo será implementado pela empresa parceira a fim de se aprimorar, por meio de testes, as funções definidas e validar a aplicabilidade destas funcionalidades, bem como identificar novos requisitos que se façam necessários.

5.3. TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, podemos considerar o desenvolvimento de novos requisitos que possam aprimorar a experiência do usuário, entre eles: um

sistema de medidas que traga mais informações sobre o tamanho do jardim vertical a ser executado, juntamente com a quantidade de plantas utilizadas na simulação. Desta forma, amplia-se o leque de informações do projeto. Outra possibilidade seria melhorar a interface gráfica com uma interface visual mais polida, visto que o protótipo foi desenvolvido de forma genérica, sem avaliar todos os aspectos do design.

REFERÊNCIAS

ACCASOFTWARE (Italia). **Programa para Paisagismo**: edificios land. Edificius LAND. 2021. Disponível em: <<https://www.accasoftware.com/ptb/programa-para-paisagismo>>. Acesso em: 01 fev. 2021

AFONSO, Alexandre. **O que é Angular?** 2018. Disponível em: <<https://blog.algaworks.com/o-que-e-angular/>>. Acesso em: 09 maio 2020.

ALFF, Francilvio. **O que são requisitos funcionais e requisitos não funcionais?** 2018a. Disponível em: <<https://analisederequisitos.com.br/requisitos-funcionais-e-requisitos-nao-funcionais-o-que-sao/>>. Acesso em: 12 abr. 2020.

ALFF, Francilvio. **5 técnicas eficientes de levantamento de requisitos de software.** 2018b. Disponível em: <<https://analisederequisitos.com.br/5-tecnicas-para-levantar-requisitos-de-software/>>. Acesso em: 12 abr. 2020.

AUE SOFTWARE (Juiz de Fora). **AuE PhotoLANDSCAPE.** 2018. Disponível em: <<https://auesoftware.com/?id=7-PhotoLANDSCAPE&ln=pt-BR>>. Acesso em: 01 fev. 2021.

AUE SOFTWARE (Juiz de Fora). **Porque você precisa começar a utilizar um software para paisagismo agora.** 2016.

BARRETO, Maurício Vivas de Souza. **Curso de linguagem PHP.** 2000. Disponível em: <<http://www.etelg.com.br/paginaete/downloads/informatica/php.pdf>>. Acesso em: 09 abr. 2020.

BLANC, P. **The Vertical Garden: A scientific and artistic approach.** 2008. Disponível em: <<http://www.verticalgardenpatrickblanc.com/documents>>. Acesso em: 22 fev. 2020.

CAMARGO, Danieli A. de Oliveira. **ESTUDO DAS TÉCNICAS DE DESENVOLVIMENTO WEB E VALIDAÇÃO DESTE ESTUDO COM UM PORTAL PARA RECICLADORES E PRODUTORES DE RECICLADOS.** 2010. 26 f. TCC (Graduação) - Curso de Ciências da Computação, Faculdade Anhanguera de Santa Bárbara, Santa Bárbara, 2010. Disponível em: <<http://repositorio.pgsskroton.com.br/bitstream/123456789/1116/1/artigo%2032.pdf>>. Acesso em: 12 abr. 2020.

CAMPOS, Toni. **O QUE É BOOTSTRAP E PARA QUE SERVE? Início » O Que é Bootstrap e Para Que Serve?** Cia Websites. Disponível em: <<https://www.ciawebsites.com.br/dicas-e-tutoriais/o-que-e-bootstrap/>>. Acesso em: 09 maio 2020.

CASAECONSTRUCAO.ORG. **Plantas trepadeiras: espécies e fotos de muros e pérgolas com plantas!** 2016. Disponível em: <<https://casaconstrucao.org/paisagismo/trepadeiras/>>. Acesso em: 15 mar. 2020.

CAYRES, Paulo Henrique. Modelagem de Banco de Dados. Rio de Janeiro: RNP/ESR, 2015.

CELSO BERGAMASCO (São Paulo). *Associação Nacional de Paisagismo (Ed.). O mercado de paisagismo*. Disponível em: <http://www.paisagismobrasil.com.br/index.php?system=news&news_id=1159&action=read>. Acesso em: 22 fev. 2020.

CLUBE DE JARDINAGEM (São Paulo). **O Jardim – Egito, Babilônia, Pérsia, Grécia e Roma (antigas)**. 2020. Disponível em: <<http://www.clubedejardinagem.com.br/index.php/jardinagem-paisagismo/32-o-jardim-egito-babilonia-persia-grecia-e-roma-antigas>>. Acesso em: 09 abr. 2020.

CRUCIOL BARBOSA, Murilo; FONTES, Maria Solange G. de C. Jardins verticais: modelos e técnicas. **PARC Pesquisa em Arquitetura e Construção**, Campinas, SP, v. 7, n. 2, p. 114-124, jun. 2016. ISSN 1980-6809. Disponível em: <<http://periodicos.sbu.unicamp.br/ojs/index.php/parc/article/view/8646304>>. Acesso em: 20 fev. 2020. doi:<http://dx.doi.org/10.20396/parc.v7i2.8646304>.

DAVI GOMES SEVERIANO. AuE Paisagismo. **PhotoLANDSCAPE: Fotomontagem passo-a-passo parte 5**. 2012. Disponível em: <<https://auepaisagismo.com/default.aspx?id=photolandscape:-fotomontagem-passo-a-passo-parte-5&in=1060>>. Acesso em: 09 maio 2020.

DYNASCAPE.COM (Canadá) (ed.). Hortiscopía Pro: expandir sua seleção de planta de design de paisagem. EXPANDIR SUA SELEÇÃO DE PLANTA DE DESIGN DE PAISAGEM. 2021. Disponível em: <<https://www.dynascape.com/solutions/design/hortiscopía-pro/>>. Acesso em: 01 fev. 2021.

EVANGELISTA, Leonéia (ed.). **Editor de código gratuito: 7 opções para você escolher o seu**: microsoft visual studio code. Microsoft Visual Studio Code. 2018. Disponível em: <<https://king.host/blog/2018/06/editor-de-codigo-gratuito-7-opcoes-para-voce-escolher-o-seu/>>. Acesso em: 01 fev. 2021.

FÁBIO FIALHO (Florianópolis). **Cursos e Treinamento**. 2016. Disponível em: <<https://www.verticajardins.com.br/>>. Acesso em: 22 fev. 2020.

FREIRES, Naélio. **Um guia para usar React JS**. 2019. Disponível em: <<https://blog.geekhunter.com.br/um-guia-para-usar-react-js/>>. Acesso em: 09 maio 2020.

GRUPO MB (Balneário Piçarras). Grupo Mb (ed.). **A luz, o lux e o luxímetro**: Como e por que medir a iluminação no ambiente de trabalho. 2015. Disponível em: <<https://grupomb.ind.br/mbobras/economia-de-energia/como-e-por-que-medir-a-iluminacao-no-ambiente-de-trabalho/>>. Acesso em: 12 abr. 2020.

GONÇALVES, Edson. **Desenvolvendo Aplicações web com NetBeans IDE 5.5**. Editora Ciência Moderna, 2007.

GSKY PLANT SYSTEM. Disponível em: <<https://gsky.com>>. Acesso em 09 abr. 2020.

GUEDES, Gilleanes T. A. **UML 2 - Uma abordagem prática**, 2ª ed., São Paulo: Novatec, 2011

GUIMARÃES, Ana Luiza. **O que são frameworks para desenvolvimento WEB. (Conheça os melhores para PHP, Python e Ruby).** 2019. Disponível em: <<https://configr.com/blog/frameworks-para-desenvolvimento-web/>>. Acesso em: 12 abr. 2020.

I LOVE DÉCOR (Brasil). **Jardins Verticais, uma história da evolução.** 2012. Disponível em: <<https://ilovedecor.wordpress.com/2012/07/27/jardins-verticais-uma-historia-da-evolucao/>>. Acesso em: 22 fev. 2020.

INSTITUTO FEDERAL DE SANTA CATARINA, 2018, Florianópolis. **Plano Pedagógico do Curso (PPC) - 2018.** Florianópolis: Ifsc, 2017. 123 p. Disponível em: <https://moodle.ifsc.edu.br/pluginfile.php/58896/mod_resource/content/1/PPC_GTI_2016_19_09_2017.pdf>. Acesso em: 20 fev. 2020.

LIMA, Davi de (ed.). **Astah Community: modele softwares com astah community.** Modele softwares com Astah Community. 2016. Disponível em: <<https://www.techtodo.com.br/tudo-sobre/astah-community.html#:~:text=Astah%20Community%20%C3%A9%20um%20software,sistemas%20operacionais%20Windows%2064%20bits>>. Acesso em: 01 fev. 2021.

LONGEN, Andrei. **O Que é Angular? Guia para Iniciantes.** 2020. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-angular/>>. Acesso em: 09 maio 2020.

LONGEN, Andrei. **O Que é WAMP? Um Guia Amigável para Iniciantes.** 2019. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-wamp>>. Acesso em: 02 fev. 2021.

MARTINEZ, Marina. **UML.** 2006. Disponível em: <<https://www.infoescola.com/engenharia-de-software/uml/>>. Acesso em: 12 abr. 2020.

MATTIUZ, Claudia Fabrino Machado. **HISTÓRIA E EVOLUÇÃO DOS JARDINS.** In: MATTIUZ, Claudia Fabrino Machado. **Disciplina de Floricultura e Paisagismo.** São Paulo: Usp, 2010. p. 1-1. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/1880778/mod_resource/content/1/Texto%20Alunos%20Evoluc%CC%A7a%CC%83o%20Paisagismo-1.pdf>. Acesso em: 05 abr. 2020.

MOTA, Fernando Jorge. **Bootstrap – Um framework HTML/CSS/JS para você começar bem o seu projeto.** 2014. Disponível em: <<https://fjorgemota.com/bootstrap-um-framework-htmlcssjs-para-voce-comecar-bem-o-seu-projeto/>>. Acesso em: 09 maio 2020.

NALESSO, Amanda. **Construindo aplicações front-end com Angular.** Disponível em: <<https://movile.blog/aplicacoes-front-angular/>>. Acesso em: 09 maio 2020

OFICINA DA NET (Rio Grande do Sul). **O que é um SGBD?** 2016. Disponível em: <<https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd>>. Acesso em: 09 maio 2020.

OLIVEIRA, Guilherme Motta de et al. **Representação gráfica digital em paisagismo com os softwares autolandscape e photolandscape.** 2007. Disponível em: <<https://ornamentalhorticulture.emnuvens.com.br/rbho/article/viewFile/1964/1523>>. Acesso em: 12 abr. 2020.

OLIVEIRA, Marcos. **O que é SGBD?** 2019. Disponível em: <<https://terminalroot.com.br/2019/08/o-que-e-sgbd.html>>. Acesso em: 09 maio 2020.

PINTO, Maicon Silva. **Apresentando o CakePHP**. 2018. Disponível em: <<https://www.dbccompany.com.br/apresentando-o-cakephp/>>. Acesso em: 09 maio 2020.
PORTAL GSTI. **O que é CakePHP?** Origem do CakePHP. 2020. Disponível em: <<https://www.portalgsti.com.br/cakephp/sobre/>>. Acesso em: 09 maio 2020.

QUINTADELLARTE (RJ). Blog Quintadellarte (ed.). **História dos jardins no Brasil: quais são suas influências?** 2017. Disponível em: <<https://blog.quintadellarte.com.br/historia-dos-jardins-no-brasil-quais-sao-suas-influencias/>>. Acesso em: 02 mar. 2020

RENATO RODRIGUES (Minas Gerais). Aprenda Fácil Editora. **O surgimento dos jardins**. 2016. Disponível em: <<https://www.afe.com.br/artigos/o-surgimento-dos-jardins>>. Acesso em: 05 abr. 2020.

RIBEIRO, Hudson Simões et al. **Integração de Tecnologias para Desenvolvimento de Sistemas Web, utilizando a metodologia AJAX**. 2006. Disponível em: <https://www.aedb.br/seget/arquivos/artigos06/304_Artigo_SEGET.pdf>. Acesso em: 12 abr. 2020.

SCHERER, Minéia Johann; FEDRIZZI, Beatriz Maria. JARDINS VERTICAIS: POTENCIALIDADES PARA O AMBIENTE URBANO. **Revista Latino-americana de Inovação e Engenharia de Produção**, Curitiba, v. 2, n. 2, p. 49-61, jun. 2014. Disponível em: <<https://revistas.ufpr.br/relainep/article/view/37883/23495>>. Acesso em: 05 maio 2020.

SILVA, E. L. da; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. Florianópolis: UFSC, 2005. 138 p. Disponível em: <www.posarq.ufsc.br/download/metPesq.pdf>. Acesso em: 04 set. 2007.

SILVA, Fernando. **História do PHP – Curiosidades por trás dos 23 anos da linguagem**. 2018. Disponível em: <<https://king.host/blog/2018/06/historia-do-php/>>. Acesso em: 09 maio 2020.

TEIXEIRA, José Ricardo. **Apresentando o framework CakePHP**. 2013. Disponível em: <<https://www.devmedia.com.br/apresentando-o-framework-cakephp/27455>>. Acesso em: 09 maio 2020.

UDACITY BRASIL. **React: o que é e como funciona essa ferramenta?** 2018. Disponível em: <<https://tableless.com.br/react-o-que-e-e-como-funciona-essa-ferramenta/>>. Acesso em: 09 maio 2020.

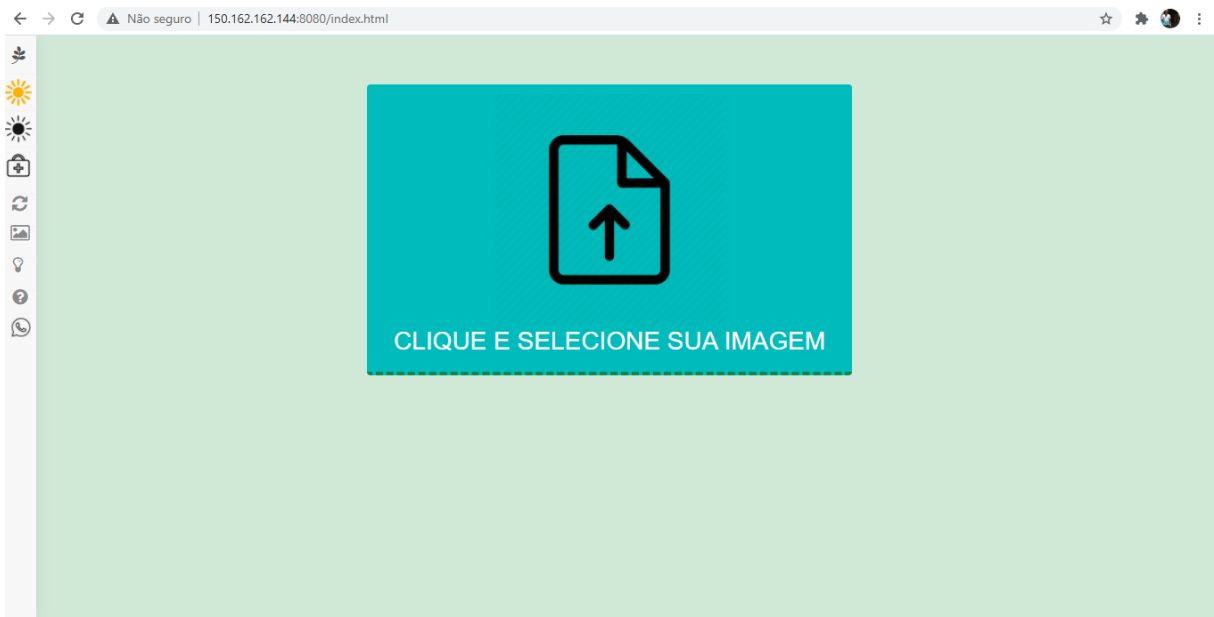
UNITUR (Porto Alegre). **JARDINS DO PALÁCIO DE VERSALHES: A ARTE DA SIMETRIA! ENTENDA POR QUÊ**. 2019. Disponível em: <<https://www.unitur.com.br/jardins-do-palacio-de-versalhes-a-arte-da-simetria-entenda-por-que/>>. Acesso em: 09 maio 2020.

VERTICAL GARDEN (São Paulo) (org.). **10 razões para amar Jardins Verticais**. 2019. Disponível em: <<https://www.verticalgarden.com.br/post/jardins-verticais-10-razoes-para-amar>>. Acesso em: 09 maio 2020.

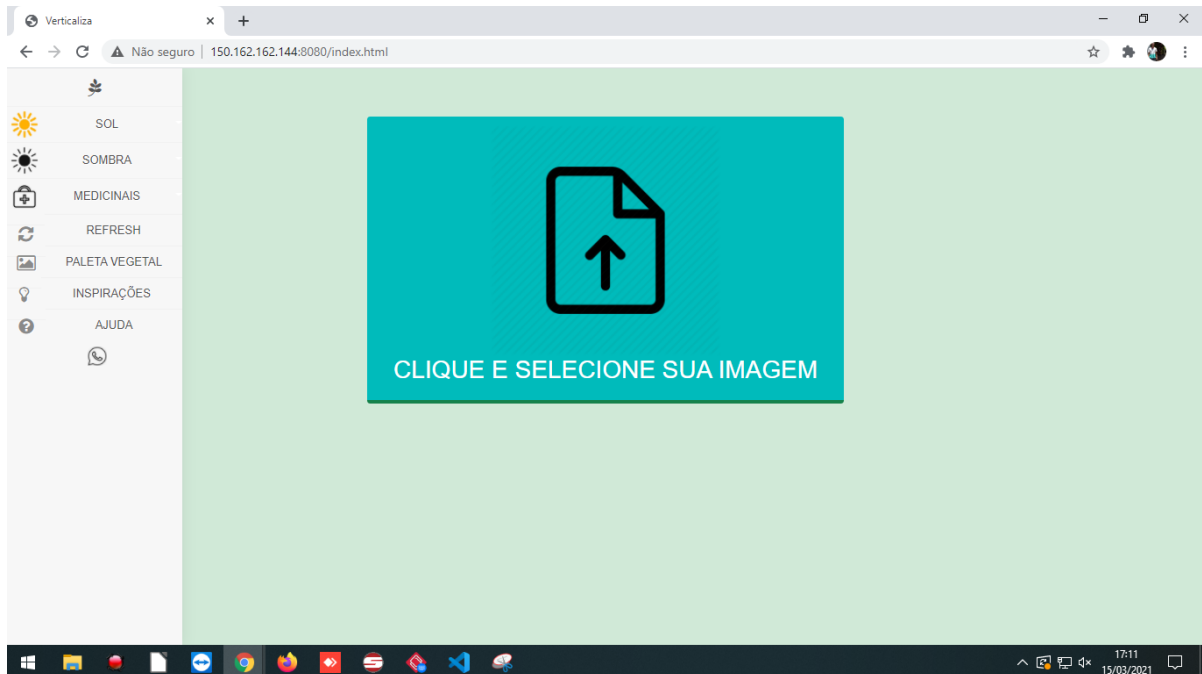
XUBNAHA CO (Brasil). **FloraMe: paisagismo fácil**. Paisagismo Fácil. 2019. Disponível em: <<https://apkpure.com/br/florame-landscaping-made-easy/com.xubnaha.florame>>. Acesso em: 01 fev. 2021.

WILLEMANN, David Pedro; IBARRA, Gustavo Bestetti. **Framework Java de Apoio ao Desenvolvimento de Aplicações Web com Banco de Dados, utilizando Struts, Tiles e Hibernate**. 2007. 155 f. TCC (Graduação) - Curso de Ciências da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2007. Disponível em:
<<https://repositorio.ufsc.br/bitstream/handle/123456789/183987/TCC%20-%20Framework%20Web%20Java%20-%20Gustavo%20%26%20David%20-%202007-1.pdf?sequence=1&isAllowed=y>>. Acesso em: 12 abr. 2020.

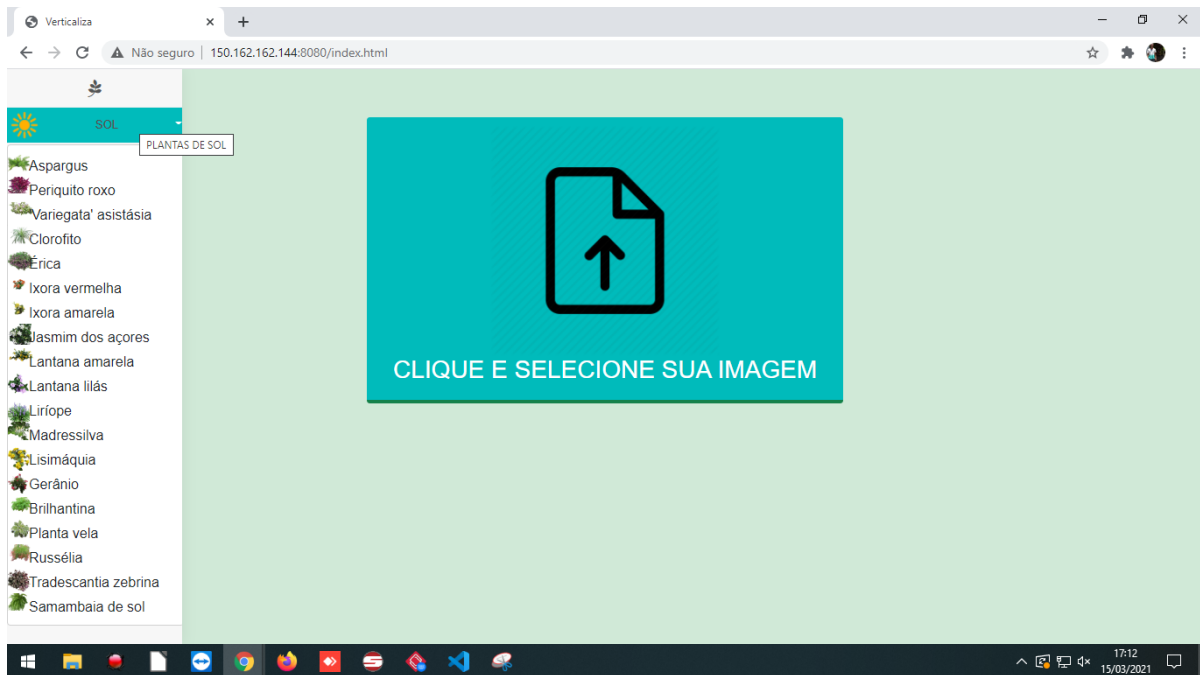
APÊNDICE A – TELA PRINCIPAL DA APLICAÇÃO



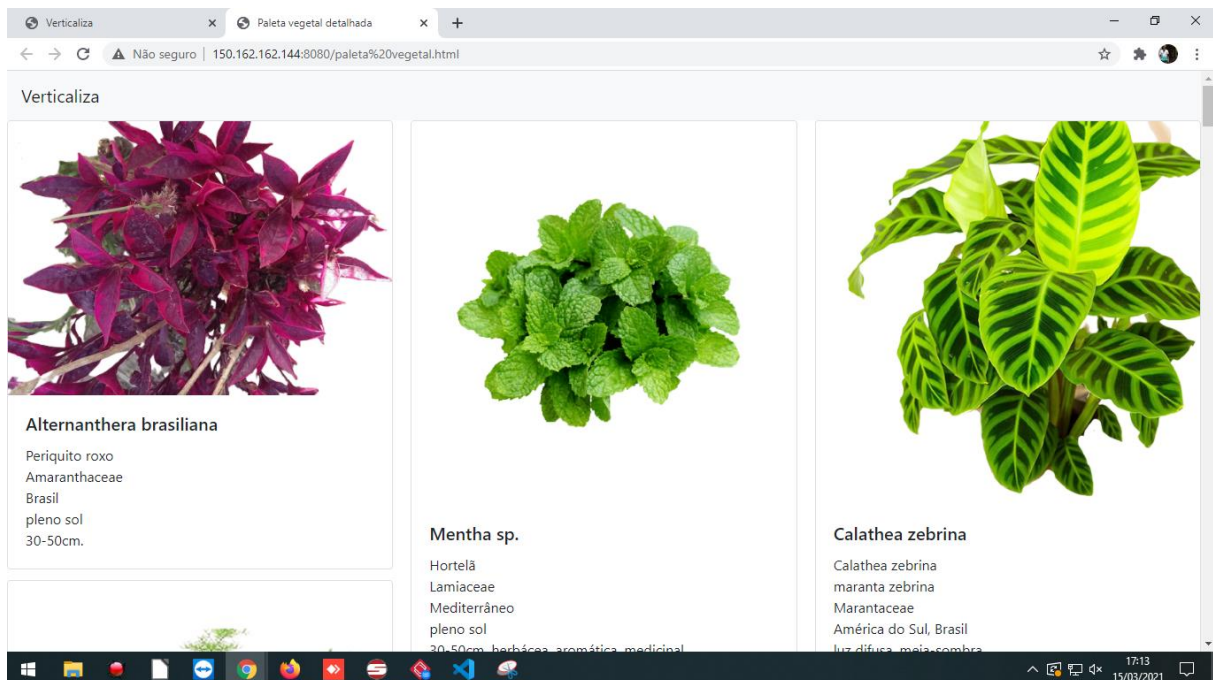
APÊNDICE B – TELA PRINCIPAL DA APLICAÇÃO COM MENU EXPANDIDO



APÊNDICE C – MENU EXPANDIDO COM LISTA DE PLANTAS

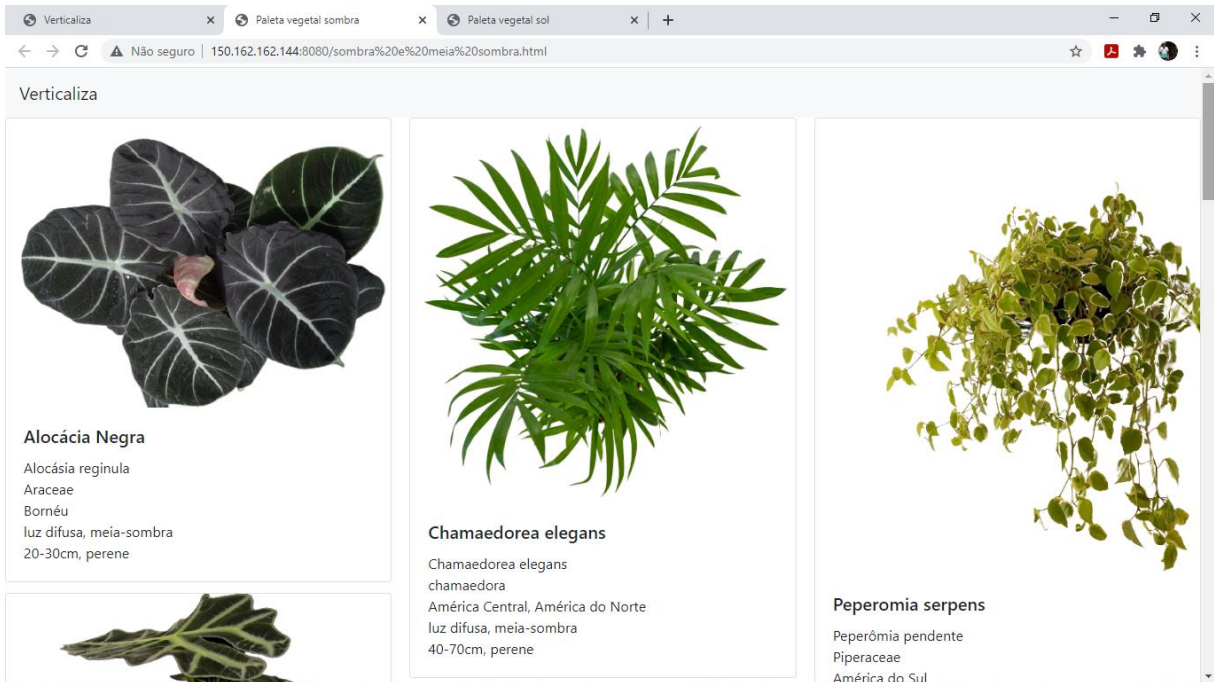


APÊNDICE D – PALETA VEGETAL SOL



APÊNDICE E – PALETA VEGETAL SOMBRA

Verticaliza



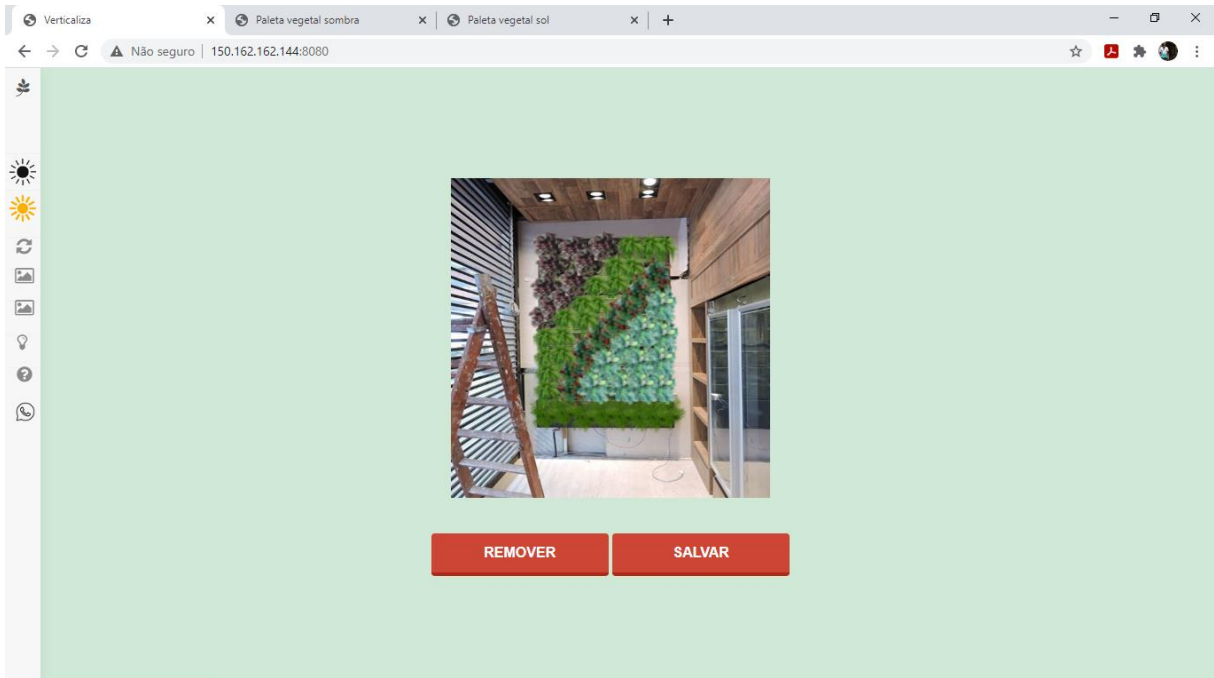
Alocácia Negra
Alocásia reginula
Araceae
Bornéu
luz difusa, meia-sombra
20-30cm, perene

Chamaedorea elegans
Chamaedorea elegans
chamaedorea
América Central, América do Norte
luz difusa, meia-sombra
40-70cm, perene

Peperomia serpens
Peperômia pendente
Piperaceae
América do Sul

APÊNDICE F – EXEMPLO DE SIMULAÇÃO EXECUTADA COM A APLICAÇÃO

Verticaliza



REMOVER SALVAR