

Faculdade Senac Jaraguá do Sul
Análise Desenvolvimento De Sistemas

Marcos Miguel Regis
Oilson Soares Guimarães
João Vitor Lago
Guilherme Willian Goldacher

PROTÓTIPO DE E-COMMERCE CLEANEX

Jaraguá do Sul

2019

Marcos Miguel Regis
Oilson Soares Guimarães
João Vitor Lago
Guilherme Willian Goldacher

PROTÓTIPO DE E-COMMERCE CLEANEX

Trabalho apresentado a Faculdade SENAC como requisito parcial para obtenção de créditos da Trabalho de Conclusão Semestral de caráter interdisciplinar de Desenvolvimento de Aplicação no curso de Análise Desenvolvimento de Sistemas.

Orientador: Emerson Pellis

Jaraguá do Sul

RESUMO

A Equipe de Alunos do Serviço Nacional de Aprendizagem Comercial (SENAC), do curso Superior de Análise e Desenvolvimento de Sistemas (ADS) está iniciando o projeto definido como um E-commerce de um lojista da cidade de Jaraguá do Sul – SC. O Projeto de Nome CLEANEX tem como objetivo inicial trazer aos usuários de produtos de limpeza em grandes quantidades, tais como: Condomínios Residenciais, Repartições Públicas, Conjuntos Empresariais, Parques Fabris, etc..., O projeto contará com a tecnologia atual e com tudo pode oferecer melhores resultados ao clientes de produtores locais, uma forma de aquisições desses produtos via Internet em um Site concentrando produtores locais, trazendo assim maior rentabilidade pro comercio local. O Site www.cleanex.com.br terá como lojista principal Alana Produtos de Limpeza – Fornecedor de Produtos de Limpeza em Jaraguá do Sul localizado na Rua 25 de Julho, 662, Bairro Vila Nova, Cep: 89259-000.

Palavras-chave: Lojista, Tecnologia, E-commerce.

LISTAS DE ILUSTRAÇÕES

Figura 1 - Um dos Primeiros Processadores de imagens de interface(IMP)	13
Figura 2 - Tim Berners Lee, o inventor da web	17
Figura 3 - Tim Berners Lee, o inventor da web	18
Figura 4 - Navegador Opera	19
Figura 5 - Exemplo de Script CSS.	20
Figura 6 - Casos de uso	23
Figura 7 - Atores	24
Figura 8 - Ator (representação tradicional)	24
Figura 9 - Associação entre atores e casos de uso	25
Figura 10 - Generalização de casos de uso	26
Figura 11 - Extensão de caso de uso	27
Figura 12 - Inclusão de casos de uso	27
Figura 13 - Arquitetura cliente-servidor para uma biblioteca de filmes.	29
Figura 14 - Exemplo de um diagrama entidade relacionamento	31
Figura 15 - Exemplos de tabelas de um Banco de Dados relacional	32
Figura 16 - Tela inicial VScode	37
Figura 17 - Caso de uso Cleanex	37
Figura 18 - Pagina Home	38
Figura 19 - Pagina Cadastro de produtos	39
Figura 20 - Pagina Contato	40
Figura 21 - Pagina Sobre	40
Figura 22 - Pagina Carrinho de compras	41
Figura 23 - Pagina Login	42

LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

ADS – Análise e Desenvolvimento de Sistemas

API- Application Programming Interface

ARP-Advanced Research Projects Agency

CEP – Código De Endereçamento Postal

CERN- Organização Europeia de Pesquisas Nuclear

CSAIL- Laboratório da Ciência da Computação e Inteligência Artificial

CSNET - Computer Science Network

DARPA –Defense Advanced Reseach Projects

DNS – Domain Name System

ENIAC- Electronic Numerical Integrator And Computer

HTML – Hyper TextMarkup Language

IBM- International Business Machines

IDÉ - Integrated Development Environment

IMPsI -interface message processors

IP – Internet Protocol

LAN- local area network, sigla

MIT- Instituto da Tecnologia de Massachusetts

MIT-Massachusetts Institute of Technology

MySQL- Structured Query Language

NCP- network-control protocol

OMC – Organização Mundial do Comércio

PC – Personal computer

PHP- Personal Home Page

SENAC – Serviço Nacional de Aprendizagem Comercial

SQL – Structured Query Language

SRI- Stanford Research Institute

TCP – Transmission Control Protocol

W3C- World Wide Web Consortiun

WWW – World Wide Web

XHTML-Extensible Hyper TextMarkup Language

SUMÁRIO

1 INTRODUÇÃO	9
2 OBJETIVOS	9
2.1 OBJETIVOS GERAL	9
2.2 OBJETIVOS ESPECÍFICOS.....	9
3 JUSTIFICATIVA	10
4 FUNDAMENTAÇÃO TEÓRICA.....	10
4.1 HISTÓRIA DA INTERNET.....	11
4.1.1 DESENVOLVIMENTO DA COMUTAÇÃO DE PACOTES 19961-1972	11
4.1.2 REDES PROPRIETÁRIAS E TRABALHO EM REDE 1972-1980.....	13
4.1.3 PROLIFERAÇÃO DE REDES 1980-1990.....	15
4.1.4 A EXPLOSAÇÃO DA INTERNET A DÉCADA DE 1990	16
4.1.5 WEB	16
4.1.6 HTML.....	17
4.1.7 XHTML.....	17
4.1.8 NAVEGADOR.	18
4.1.9 DEFINIÇÕES E CONCEITOS DE CSS.....	19
4.1.10 JAVASCRIPT	20
4.1.11 HISTÓRIA DO PHP	21
4.1.12 ANÁLISES DE SISTEMAS	22
4.1.13. UML – UNIFIED MODELING LANGUAGE.	22
4.1.14. DIAGRAMA DE CASO DE USO.	22
4.1.15. ATORES.....	23
4.1.16. RELACIONAMENTOS ENTRE ATORES E CASOS DE USO.	25
4.1.17. RELACIONAMENTOS ENTRE CASOS DE USO	25
4.1.18. ARQUITETURAS DE SISTEMAS	27
4.1.19. BANCO DE DADOS.....	30
4.1.20. MODELAGEM DE BANCO DE DADOS.....	30
4.1.21. MODELO CONCEITUAL.....	31
4.1.22. MODELO LÓGICO.	32
4.1.23. MYSQL.....	33
5. ESPECIFICAÇÃO DO SISTEMA	33
5.1 LEVANTAMENTO DE INFORMAÇÕES.....	33
5.2 ESPECIFICAÇÃO DO BANCO DE DADOS	35
5.3 IDE – AMBIENTES DE DESENVOLVIMENTO	35
5.4 LINGUAGEM DE PROGRAMAÇÃO	37
6 ESPECIFICAÇÃO DA ARQUITETURA DO SISTEMA	38
6.1 ESPECIFICAÇÃO DA APLICAÇÃO	38
6.1.1. TELA PRINCIPAL DO E-COMMERCE.....	38
6.1.2. CADASTRO DE PRODUTOS.....	38
6.1.3. CONTATOS	39

6.1.4 <i>SOBRE</i>	40
6.1.5. <i>CARRINHO</i>	40
6.1.7. <i>LOGIN</i>	41
REFERÊNCIAS	43
<i>APÊNDICE A – TÍTULO DO APÊNDICE</i>	44
ANEXO A – TÍTULO DO ANEXO	45

1 INTRODUÇÃO

O Projeto de e-commerce chamado CLEANEX – Soluções em produtos de limpeza em atacado tem como objetivo facilitar a comunicação e vendas entre o fabricante de produtos de limpeza, e o consumidor final que deseja usufruir dos produtos em forma de atacado. Esta solução irá envolver a empresa Alana Produtos de Limpeza que é o fornecedor desses produtos, e outras empresas que serão consumidores, no caso restaurantes, hotéis postos de combustíveis entre outras empresas.

2 OBJETIVOS

O Projeto “Cleanex” que é um e-commerce está sendo implementado com o intuito de facilitar a venda online de produtos de limpeza, facilitando a comunicação entre o produtor e o consumidor final. Os principais consumidores finais são empresas que utilizam produtos de limpeza em grande quantidade.

2.1 Objetivos geral

Fazer a venda online de produtos de limpeza através de um e-commerce, de maneira que o produtor consiga expor e vender seus produtos, de maneira mais rápida e eficiente.

2.2 Objetivos específicos

Como Forma de detalhar todos os objetivos Específicos desse projeto, está listado abaixo:

- Identificar e-commerce de venda de produtos de limpeza existentes.
- Pesquisar produtores de produtos de limpeza artesanais.
- Explorar formas de viabilizar vendas em grande escala dos produtos de limpeza.
- Procurar saber como funciona um e-commerce.

3 JUSTIFICATIVA

O Cleanex é um e-commerce de produtos de limpeza do tipo B2B ou seja, de empresa para empresa, o fornecedor que será Alana Produtos de Limpeza que utilizara desse software para expandir suas vendas, onde poderá expor toda sua linha de produtos, desse modo todos que tiverem uma conexão com a internet e um navegador web poderão acessar e realizar compras a qualquer momento, além do custo de um e-commerce ser bem mais barato do que montar uma loja física, pois a demanda de funcionários diminui, energia elétrica, entre outros custos, desse modo com custos reduzidos a empresa poderá atuar de uma forma mais competitiva com a concorrência.

Com o cleanex os clientes poderão realizar compras sem precisar sair do seu local, poderá ver uma vasta linha de produtos com diversos preços, e realizar uma compra de forma inteligente, de modo que se possa simular a compra, avaliar preços produtos de acordo com sua necessidade, as compras serão 100% online e as formas de pagamento poderão ser cartão de credito, boletos, podendo ser parcelado, assim que o cliente realizar a compra e só aguardar no seu endereço, pois a Alana Produtos de limpeza, se encarregará da distribuição dos produtos.

4 FUNDAMENTAÇÃO TEÓRICA

Vamos dar uma rápida e resumida cronologia sobre a Internet até o surgimento do e-commerce. No ano de 1945 surgiu o ENIAC, primeiro computador eletrônico, em 1950 surgiu o primeiro modem digital, já em 1963 surge o primeiro “mouse”, em 1969 a ARPANET dá início a Internet, em 1971 forma criadas as primeiras redes LAN sem fios (wireless) e também o primeiro e-mail foi enviado, em 1974 a primeira rede ARPANET comercial foi criada, no ano de 1975 surgiu o primeiro “spam” e em 1985 o primeiro “vírus”, daí até o ano de 1995, ano da criação do e-commerce, muitas inovações tecnológicas positivas surgiram, e não param mais.

O comercio eletrônico ou e-commerce representa parte do presente e do futuro do comercio existem várias oportunidades de negócio espalhadas pela internet além de muitas que são criadas a todo momento.

De acordo com GOMES (2011) o comércio pela internet, é uma pratica e que envolve, mídias da propaganda, sejam institucionais, sejam comerciais, canal de serviço de atendimento ao consumidor, telemarketing ativo, telemarketing passivo prosperarão de novos clientes,

recuperação de clientes inativos, incremento de vendas a clientes existentes pela exploração das potencialidades do cliente, etc.

Alguns cuidados deveram ser tomados. Deve-se ter um mínimo de estrutura, mas o mais importante é saber o que se quer e espera da internet, qual o principal objetivo de se utilizar a internet, de se colocar nossa empresa, marca, produtos/serviços à disposição dos internautas. Mais ainda quem são os internautas:

Segundo GOMES (2011) os internautas são públicos bastante variado, de diversas faixas etárias, diversificadas posição social, de nível cultural um pouco acima da média, de diversas profissões, empresários, desempregados, bem empregados, compradores, vendedores, pesquisadores, e também muitos curiosos.

O comercio eletrônico ou e-commerce representa parte do presente do futuro do comércio. Existem várias oportunidades de negócios espalhadas pela internet, além de muitas que são criadas a todo momento. (TEIXEIRA, 2015)

4.1 História da internet

A seguir falaremos um pouco da história da internet desde o início até os dias de hoje.

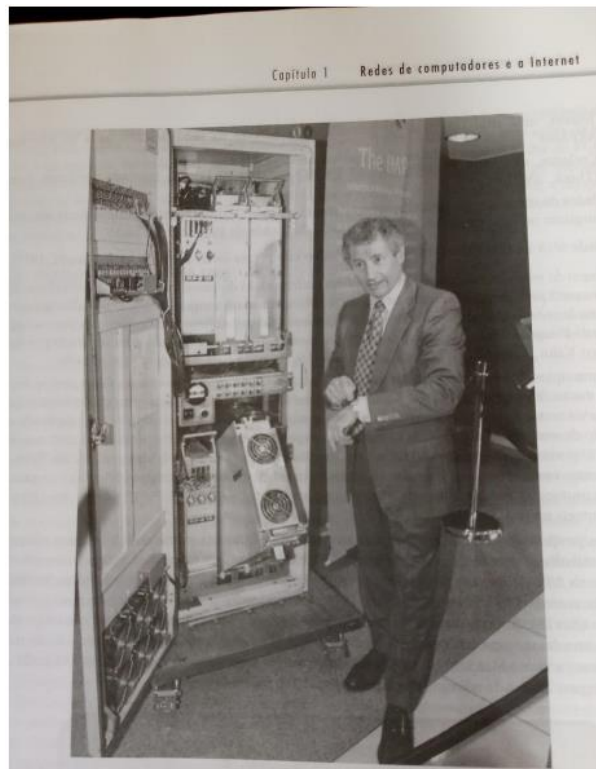
4.1.1 Desenvolvimento da comutação de pacotes 1961-1972

Os primeiros passos da rede de computadores e da internet podem ser traçados desde o início da década de 1990, quando a rede telefônica era a rede de comunicação dominante no mundo inteiro. A rede telefônica usa comutação de circuitos para transmitir informações entre uma origem e um destino, uma escolha certada já que a voz é transmitida a uma taxa constante entre a origem e o destino. Dada a importância cada vez maior (e o alto custo) dos computadores no início da década de 1960 e o advento de computadores com multiprogramação (time-sharing), nada seria mais natural (agora que temos uma visão do passado) do que considerar a questão de como interligar computadores para que pudessem ser compartilhados entre usuários distribuídos em localizações geográficas diferentes. O trafego gerado por esses usuários provavelmente era intermitente, por rajadas, período de atividades, como o envio de um comando a um computador remoto, seguidos de períodos de inatividade, como a espera por uma resposta ou o exame de uma resposta recebida (KUROSE, 2006).

Três grupos de pesquisadores ao redor do mundo, sem que nenhum tivesse conhecimento do trabalho do outro, começaram a inventar a comutação de pacotes como uma alternativa poderosa e eficiente a de circuitos. O primeiro trabalho publicado sobre técnicas de comutação de pacotes foi o de Leonard Kleinrock (Kleinrock, 1961, 1964), que naquela época era um doutorando do MIT. Usando a teoria de filas, o trabalho de Kleinrock demonstrou, com elegância, a eficácia da abordagem da comutação de pacotes para fontes de tráfegos intermitentes (em rajadas). Em 1964, Paul Baran (Baran, 1964), do Rand Institute, começou a investigar a utilização de comutação de pacotes na transmissão segura de voz pelas redes militares, ao mesmo tempo que Donald Davies e Roger Scantlebury desenvolviam suas ideias sobre esse assunto no National Physical Laboratory, na Inglaterra.

Os trabalhos desenvolvidos no MIT, no Rand Institute e no National Physical Laboratory foram os alicerces do que hoje é a internet. Mas a internet também tem uma longa história de atitudes do tipo “construir e demonstrar”, que também data no início da década de 1960. J.C.R. Licklider (DEC, 1960), e Lawrence Roberts, ambos colegas de Kleinrock no MIT, foram adiante e lideraram o programa de ciência de computadores na ARP (Advanced Research Projects Agency– Agência de Projetos de Pesquisa Avançada), nos Estados Unidos. Roberts publicou um plano geral para a ARPAnet (Roberts, 1967), a primeira rede de computadores por comutação de pacotes e uma ancestral direta da internet pública de hoje. Os primeiros comutadores de pacotes eram conhecidos processadores de mensagens de interface (Interface message processors IMPs), e o contrato para fabricação desses comutadores foi entregue a empresa BBN. Em 1969, no dia do trabalho nos Estados Unidos, foi instalado o primeiro IMP na UCLA (Universidade da Califórnia em Los Angeles) sob a supervisão de Kleinrock. Logo em seguida foram instalados três IMPs, adicionais no Stanford Research Institute (SRI), na Universidade da Califórnia em Santa Barbara e na Universidade de Utah (figura 1).

Figura 1 Um dos Primeiros Processadores de imagens de interface(IMP)



Fonte Kuroser, 2006.

O incipiente precursor da Internet tinha quatro nós fundamentais de 1969. Kleinrock recorda que a primeiríssima utilização da rede foi fazer um login remoto entre UCLA e o SRI, derrubando o sistema (KUROSE, 2006).

Em 1972, a ARPAnet tinha aproximadamente 15 nós e foi apresentada publicamente pela primeira vez por Robert Kahn na Conferência Internacional sobre Comunicação por Computadores (International Conference on Computer) daquele ano. O primeiro protocolo fim-a-fim entre sistemas finais da ARPAnet, conhecido como protocolo de controle de rede (network-control protocol – NCP), estava concluído (RFC 001) e a partir desse momento a escrita de aplicações tornou-se possível. Em 1972, Ray Tomlinson, da BBN, escreveu o primeiro programa de e-mail (KUROSE, 2006).

4.1.2 Redes proprietárias e trabalho em rede 1972-1980

A ARPAnet inicial era uma rede isolada, fechada. Para se comunicar com uma máquina da ARPAnet, era preciso estar ligado a um outro IMP dessa rede. Do início a meados de 1970, surgiram novas redes de comutação de pacotes.

ALOHAnet, uma rede de microondas ligando universidades das ilhas do Havaí (Abramson, 1970), bem como as redes de pacotes por satélite [RFC 819] e por rádio (Kahn, 1978) da DARPA.

Telenet, uma rede comercial de comutação de pacotes da BBN Fundamentada na tecnologia ARPAnet.

Cyclades, uma rede de comutação de pacotes pioneira na França, montada por Louis Pouzin (Think, 2002).

Redes de tempo compartilhando como a Tymnet e a rede GE information Services, entre outras que surgiram no final da década de 1960 e início da década de 1970 (Schwarz, 1977).

Rede SNA da IBM (1969-1974), cujo trabalho comparava-se ao da ARPAnet (Schwartz, 1977).

O número de redes estava crescendo. Hoje, com perfeita visão do passado, podemos perceber que aquele era a hora certa para desenvolver uma arquitetura abrangente para conectar redes. O trabalho pioneiro de interconexão de redes, sob o patrocínio da DARPA (Defense advanced Reseach Projects – Agencia de Projetos de Pesquisa Avançada de defesa), criou em essência uma rede de redes e foi realizado por Vinton Cerf e Robert Kahn, o termo internetting foi cunhado para descrever esse trabalho.

Esses princípios de arquitetura foram incorporados ao TCP. As primeiras versões desse protocolo, contudo, eram muito diferentes do modelo TCP de hoje. Aquelas versões combinavam uma entrega sequencial confiável de dados via retransmissão por sistemas final (que ainda faz parte do TCP de hoje) com funções de envio que hoje são desempenhadas pelo IP. As primeiras experiências com o TCP, combinadas com o reconhecimento da importância de um serviço de transporte fim-a-fim não confiável, sem controle de fluxo, para aplicações como voz e pacotes, levaram a separação entre IP e TCP e ao desenvolvimento do protocolo UDP. Os três protocolos fundamentais da internet que temos hoje – TCP, UDP, e IP – estavam conceitualmente disponíveis no final da década de 1970 (KUROSE, 2006).

Além das pesquisas sobre a internet realizadas pela DARPA, muitas outras atividades importantes relacionadas ao trabalho em rede estavam em cursos. No Havaí, Norman Abramson estava desenvolvendo a ALOHAnet uma rede de pacotes por rádio que permitia que vários lugares remotos das ilhas havaianas se comunicassem entre si. O ALOHAnet foi o primeiro protocolo de acesso múltiplo que permitiu que usuários distribuíssem em diferentes localizações geográficas compartilhassem um único meio de comunicação broadcast (uma frequência de rádio). O trabalho de Abramson sobre protocolo de múltiplos acesso foi

aprimorado por Metcalfe e Boggs com o desenvolvimento do protocolo Ethernet para redes compartilhadas de transmissão broadcast por fio.

O interessante é que o protocolo Ethernet de Metcalfe e Boggs foi motivado pela necessidade de conectar vários PCs, impressoras e discos compartilhados. Há 25 anos, bem antes da revolução do PC e da explosão das redes, Metcalfe e Boggs estavam lançando as bases para as LANs de PCs de hoje. A tecnologia Ethernet representou uma etapa importante para o trabalho em redes interconectadas. Cada rede Ethernet local era, em si, uma rede, e, à medida que o número de LANs aumentava, a necessidade de interconectar essas redes foi se tornando cada vez mais.

4.1.3 Proliferação de redes 1980-1990

Ao final da década de 1970, aproximadamente 200 máquinas estavam conectadas a ARPAnet. Ao final da década de 1980, o número de máquinas ligadas a internet pública, uma confederação de redes muito parecida com a Internet de hoje, alcançaria cem mil. A década de 1980 seria uma época de formidável crescimento.

Grande parte daquele crescimento foi consequência de vários esforços distintos para criar redes de computadores para interligar universidades. A BITNET processava e-mails e fazia transferência de arquivos entre diversas universidades do norte dos Estados Unidos. A CSNET (computer Science network – rede da ciência de computadores) foi formada para interligar pesquisadores de universidades que não tinham acesso a ARPAnet. Em 1986, foi criada a NSFNET para prover acesso a centros de supercomputação patrocinados pela NSF. Partindo de uma velocidade inicial de 56 kbps, ao final da década o backbone da NSFNET estaria funcionando a 1,5 Mbps e servindo como backbone primário para interligação de redes regionais.

Na comunidade da ARPAnet, já estavam sendo encaixados muitos dos componentes finais da arquitetura da Internet de hoje. No dia 1 de janeiro de 1983, o TCP/IP foi adotado oficialmente como o novo padrão de protocolo de máquinas para a ARPAnet (em substituição do protocolo NCP). Devido a importância do evento, o dia da transição do NCP para o TCP/IP foi marcado como antecedência a partir daquele dia todas as máquinas tiveram de adotar o TCP/IP. No final da década de 1980, foram agregadas importantes extensões ao TCP para implementação do controle de congestionamento assíncrono em hospedeiros (Jacobson, 1988). Também foi desenvolvido o sistema de nomes de domínios (DNS) utilizado para mapear nomes da internet fáceis de entender (por exemplo. gaia.cs.umass.edu) para seus endereços IP de 32 bits. Paralelamente ao desenvolvimento da ARPAnet que em sua maior parte deve-se aos

Estados Unidos, no início da década de 1980 os franceses lançaram o projeto Minitel, um plano ambicioso para levar as redes de dados para todos os lares. Patrocinado pelo governo francês, o sistema Minitel consistia em uma rede pública de comutação de pacotes (baseada no conjunto de protocolos X.25, que usava circuitos virtuais), servidores Minitel e terminais baratos como modems de baixa velocidade embutidos. O Minitel se tornou um sucesso em 1984, quando o governo francês forneceu, gratuitamente, um terminal para toda residência francesa que quisesse. O sistema Minitel incluía sites de livre acesso como o da lista telefônica e também sites particulares, que cobravam uma taxa de cada usuário baseada no tempo de utilização. No seu auge, em meados de 1990, o Minitel oferecia mais de 20 mil serviços, que iam desde home banking até bancos de dados especializados para pesquisas. Era usado por mais de 20 por cento da população da França, gerava receita de mais de um bilhão de dólares por ano e criou dez mil empregos. Estava presente em grande parte dos lares franceses dez anos antes de a maioria dos norte-americanos ouvir falar de Internet (KUROSE, 2006).

4.1.4 A explosão da Internet a década de 1990

A década de 1990 esteou com vários eventos que simbolizaram a evolução continua da Internet. A ARPAnet, a progenitora da Internet, deixou de existir. Durante a década de 1980, a MILNET e a Defense Data Network (Rede de Dados de Defesa) cresceram e passaram a carregar a maior parte do trafego do Departamento de Defesa dos Estados Unidos e a NSFNET começou a servir como uma rede de backbone conectando redes regionais nos Estados Unidos com nacionais no exterior. Em 1991, a NSFNET extinguiu as restrições que impunham a sua utilização com finalidades comerciais, mas, em 1995, perderia seu mandato quando o trafego de backbone da Internet passou a ser carregado por provedores de serviços de Internet.

O principal evento da década de 1990, no entanto, foi o surgimento da World Wide Web, que levou a internet para os lares e as empresas de milhões de pessoas no mundo inteiro. A Web serviu também como plataforma para a habilitação e a disponibilização de centenas de novas aplicações, inclusive negociação de ações e serviços bancários-online, serviços de multimídia em tempo real e serviços de recuperação de informações

4.1.5 Web

A Web foi inventada no CERN (European Center for Nuclear Physics – Centro Europeu para Física Nuclear por Tim Berners Lee entre 1989 e 1991, com base em ideias originadas de trabalhos anteriores sobre hipertexto realizados por Bush, na década de 1940, e por Ted Nelson,

da década de 1960. Berners Lee e seus companheiros desenvolveram versões iniciais de HTML, HTTP, um servidor Web e um browser (KUROSE, 2006)

Segundo SILVA (2008), Web é uma palavra inglesa que significa teia, e em internet é usada para designar abreviadamente a rede mundial de computadores, cujo funcionamento assemelha-se a uma imensa teia de aranha, integrando computadores do mundo inteiro.

4.1.6 HTML

HTML é uma das linguagens que utilizamos para desenvolver websites. O acrônimo HTML vem do inglês e significa Hypertext Markup Language ou em português Linguagem de Marcação de Hipertexto.

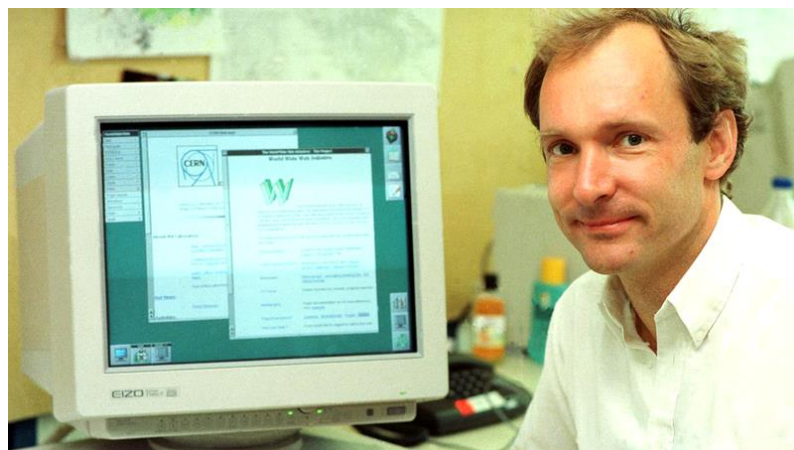
O HTML é a linguagem base da internet. Foi criada para ser de fácil entendimento por seres humanos e também por máquinas, como por exemplo o Google Chrome, Safari, Mozilla Firefox, ou outros sistemas conhecidos como navegadores (SILVA, 2008).

4.1.7 XHTML

Segundo SILVA (2008). Chamados de agentes de usuário. Um navegador, um leitor de tela, e um robô busca são exemplos de agentes de usuários.

A web foi inventada em 1992 por Sir Tim Berners-Lee cujo a foto é mostrada na figura 1. Atualmente Tim é diretor da W3C- World Wide Web Consortium, pesquisador sênior do CSAIL- Laboratório da Ciência da Computação e Inteligência Artificial do MIT- Instituto da Tecnologia de Massachusetts- e professor de Ciência da Computação na Universidade de Southampton, na Inglaterra.

Figura 2 Tim Berners Lee, o inventor da web



Fonte: Kuroser, (2006).

Tim Berners Lee, trabalhava na Seção de Computação do CERN- Organização Europeia de Pesquisas Nuclear com sede em Genebra, na Suíça, quando iniciou pesquisas visando descobrir um método que possibilitasse aos cientistas do mundo compartilhar eletronicamente seus textos e pesquisas que tivesse a funcionalidade de interligar os documentos. Estava criada a nossa web de links que como são conhecidas atualmente.

Em 1990, Tim criou o protótipo de um navegador para rodar em computadores da NeXT, uma companhia fundada em 1985 por Steve Jobs, que também criou a Apple. O computador usado por Tim Berners Lee é mostrado na figura 2.

Figura 3 Tim Berners Lee, o inventor da web



Fonte: Kuroser, (2006).

Inicialmente, o termo navegador foi chamado de WordWildWeb e, posteriormente recomendado para Nexus, a fim de evitar confusão com a Word Wild Web.

Desde a invenção da web por Tim Berners-Lee, a HTML evolui por sete versões, que são elas HTML, HTML+, HTML2.0, HTML3.0, HTML4.0, HTML5.0 sendo a mais atual.

4.1.8 Navegador.

Também conhecido como browser (em inglês), é um programa destinado a visualizar documentos desenvolvidos com linguagem de marcação, ou, em sentido mais restrito, sites na internet (SILVA, 2008).

O termo em inglês browser é usado no jargão da internet para designar um programa capaz de ler e apresentar ao usuário os conteúdos de um documento web escrito em linguagem de marcação. Browser vem do verbo to browser, que significa folhar casualmente as páginas de um livro e foi traduzido para o português como navegador. Gerando a tão bem conhecida expressão “navegar na internet”. São

exemplos de navegadores o Internet Explore, o Firefox, o Opera, e o Safari, (SILVA, 2008).

Figura 4 Navegador Opera



Fonte: Silva, (2008)

4.1.9 Definições e conceitos de CSS

Folhas de estilo em cascata são a tradução para os termos inglês Cascading Style Sheet ou, abreviadamente CSS,

Segundo a W3C "um mecanismo simples para adicionar estilos (por exemplo: fontes, cores e espaçamentos) aos documentos web".

A finalidade primordial, a razão da própria existência das CSS é devolver a linguagem da marcação HTML sua verdadeira função, aquela para a qual foi criada, ou seja, estruturar um documento web marcando com o elemento apropriado cada tipo de conteúdo que compõe o documento. Convém lembrar que esse conceito está intimamente ligado ao conceito de semântica.

Figura 5 Exemplo de Script CSS.

```

<head>
...
<style type="text/css">
  p.estilo {
    font-size: large;
    font-family:Georgia, Serif;
    color:red
  }
</style>
</head>

```

Fonte: Silva, (2008)

4.1.10 JavaScript

A linguagem de marcação HTML destina-se a estruturar uma página web, não se devendo empregá-la para adicionar estilos ou apresentação visual aos elementos que consistem a página, sendo tais tarefas funções das folhas de estilos em cascata CSS. O HTML, também não possui funcionalidades que permitam adicionar interatividade avançadas a página, sendo tal tarefa das linguagens de programação, como o JavaScript

JavaScript foi criada pela Netscape em parceria com a Sun Microsystems, com a finalidade de fornecer um meio de adicionar interatividade a uma página web. A primeira versão, denominada 1.0, foi lançada em 1995 e implementada em março de 1996 no navegador Netscape Navigator 2.0 quando o mercado era dominado pela Netscape SILVA (2010).

Logo a seguir, veio a época da chamada guerra dos browsers, cujos efeitos nocivos se fazem sentir até os dias atuais. Para não fugir à regra, a Microsoft, em resposta a Netscape, criou a linguagem JScript baseada em Visual Basic cuja primeira versão denominada JScript 1.0 foi lançada com o navegador Internet Explore 3.0. Não há como fazer funcionar um formulário HTML com o uso do elemento HTML. A HTML limita-se a criar os rótulos e campos de um formulário para serem preenchidos pelo usuário e nada mais. Com HMTL, não conseguimos processar os dados nem mesmo enviá-lo ao servidor ou a outra máquina qualquer. Para cumprir essas tarefas, é necessário utilizar um programa que consiga manipular e processar dados. Entre as várias linguagens de programação destinadas a adicionar e processar dados em páginas web, destacam-se PHP, ASP, Java< Ruby, Python entre outras SILVA (2010).

As linguagens de programação como as citadas anteriormente foram desenvolvidas para rodar do lado do servidor, isto é, dependem de uma máquina remota onde estão hospedadas as funcionalidades capazes de interpretar e fazer funcionar os programas.

JavaScript é uma linguagem desenvolvida para rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível por que existem um interpretador JavaScript hospedado no navegador SILVA (2010).

Tanto a Netscape como a Microsoft desenvolveram interpretadores JavaScript para serem hospedados no servidor, tornando possível rodar JavaScript no lado do servidor. Esses interpretadores foram disponibilizados para uso público e podem ser usados pelos desenvolvedores para serem embutidos em aplicações gerais. A organização Mozilla lançou, sob a bandeira de código aberto e livre duas versões do interpretador: Spider Monkey escrita em C Rhino escrita em Java

Em tese, precisamos apenas de um navegador para fazer funcionar scripts desenvolvidos com a linguagem JavaScript. Ao contrário, programas escritos em PHP, por exemplo, precisam ser hospedados em um servidor remoto configurado para rodar PHP ou visualizados localmente em uma máquina na qual tenha sido instalado um servidor local com suporte para PHP.

4.1.11 História do PHP

Todo software teve seu início e como o PHP não foi diferente. Aconteceu que, no ano de 1994, um consultor de empresas, chamado Rasmus Lerdorf, enviava o endereço eletrônico de sua página que continha seu currículo para os mais diversos potenciais empregadores. Curioso em saber quem era esses que estavam consultando sua página, desenvolveu um script, feito com a linguagem Perl, que colhia informações sobre os seus visitantes e media o número de acessos que sua página pessoal tinha (ANSELMO, 2002, p. 5).

Como não poderia deixar de ser, e provavelmente para impressionar ainda mais esses empregadores, ele resolveu mostrar esses dados. Chamando todo esse código desenvolvido de Personal Home Page Tools, ou seja, conjunto de ferramentas da página pessoal nasceu aqui a primeira versão batizada de PHP Tools.

Segundo (ANSELMO, 2002, p. 6) inúmeras pessoas demonstraram o interesse nesse código, então Lerdorf resolveu finalmente disponibilizá-lo. Conseguindo assim ser empregado pela Universidade de Toronto, no Canadá, ele seria o responsável por desenvolver um sistema dial-up de onde os alunos pudessem acessar a Web.

Com esse projeto, ele teve tempo para melhorar ainda mais seu script e finalmente transformou o PHP Tools numa nova ferramenta, agora feita em código C, em que ele adicionou a conectividade com banco de dados.

Aqui nasceu a conexão com o MySQL e com os famosos Form Interpreters _IF que são tags para inserção do código HTML. Nascia assim, em 1995, o que se pode chamar de PHP como o conhecemos, mas que Rasmuns e a comunidade conhece por PHP_FI.

Os desenvolvedores de páginas passaram então a utilizar o PHP para realizar coisas mais complexas e, por consequência, passaram a adicionar novas funcionalidades ao produto. Nesse ponto, já era óbvio que o desenvolvimento não poderia se apoiar em apenas Lendorf e foi então que nasceu um grupo desses desenvolvedores organizados em uma comunidade formada através da Web.

4.1.12 Análises de sistemas

De acordo com Pressman (2006) análise de sistemas é a atividade que integra grande parte das tarefas que chamamos coletivamente de engenharia de sistemas de computadores, de certo modo a análise de sistemas concentra-se em todos os elementos do sistema, não apenas em software, tem como objetivo identificar necessidades do usuário, considerar a concepção do sistema quanto à sua exequibilidade, executar a análise técnica e econômica, atribuindo funções ao software, ao hardware, estabelecendo as restrições de prazo e custo. Tanto a perícia em hardware quanto em software é exigida para que se possa atingir com sucesso os objetivos relacionados.

4.1.13. UML – Unified Modeling Language.

Uma linguagem visual utilizada para modelar sistemas de informação baseado no paradigma Orientado a Objetos. Oferece visualização, especificação, construção e documentação de artefatos do sistema.

Tem como objetivo auxiliar: requisitos, comportamento, estrutura lógica, dinâmica dos processos, necessidade física de implantação. A UML tornou-se uma norma industrial para o desenvolvimento de software OO, e não é uma linguagem de programação e sim uma linguagem de modelagem. (PÁDUA,2000.)

4.1.14. Diagrama de caso de uso.

Um caso de uso representa uma unidade coerente de funcionalidade, oferecida por um classificador. Casos de usos são mais usados para descrever funções completas de um sistema, aplicações, produtos, mas também pode ser usado no nível de subsistemas e até de classes. Os serviços de caso de uso devem oferecer valores para objetos externos que interagem com ele.

No nível de sistema, um caso de uso realiza um aspecto maior da funcionalidade do produto, devendo gerar um ou mais benefícios para o cliente, os usuários ou outros sistemas. Na fig. 3 são mostrados os casos de uso que representam a funcionalidade relativa à gestão de compras, de um produto de informatização de mercearias. O conjunto dos casos de uso sobre toda a funcionalidade do produto, e cada caso de uso representa uma fatia independente de funcionalidade, (PADUA, 2000).

Figura 6 Casos de uso

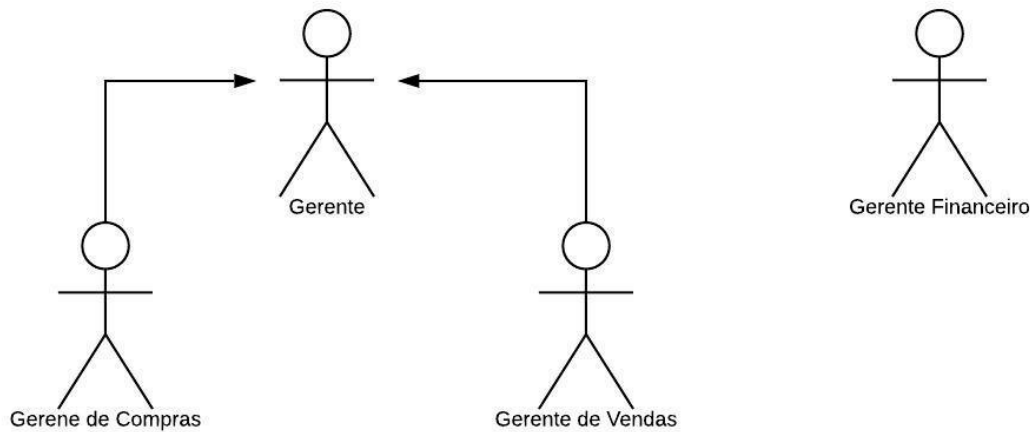


Fontes: Pádua (2009).

4.1.15. Atores.

Um ator é um classificador que representa os objetos externos que interagem com um caso de uso. Os papéis dos usuários de um produto são modelados por intermédio dos atores figura 4. Cada ator representa uma classe de usuários. Os atores modelam os papéis, e não as pessoas dos usuários; por exemplo, o mesmo usuário físico pode agir com gerente de vendas ou Gerente de compras.

Figura 7 Atores



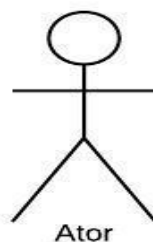
Fonte: Pádua (2009).

Pode-se também definir atores não humanos para modelar outros sistemas que devam interagir com o produto em questão, exemplo Sistema Financeiro.

Caso exista grande número de atores, pode-se agrupá-los em atores genéricos, que representem características comuns a vários grupos de usuários de comportamento semelhante em relação ao produto. Atores genéricos e específicos são ligados por relacionamentos de heranças. Na figura 4 indica-se que gerente de vendas e gerente de compras têm alguns aspectos em comum, que são modelados pelo ator abstrato gerente.

Os atores são sempre representados por figuras humanas estilizadas, mesmo quando modelam sistemas externos. As ferramentas de modelagem podem adotar ícones especiais, mas é comum o uso da tradicional notação de pauzinhos herdada da UML, como na figura (5).

Figura 8Ator (representação tradicional)



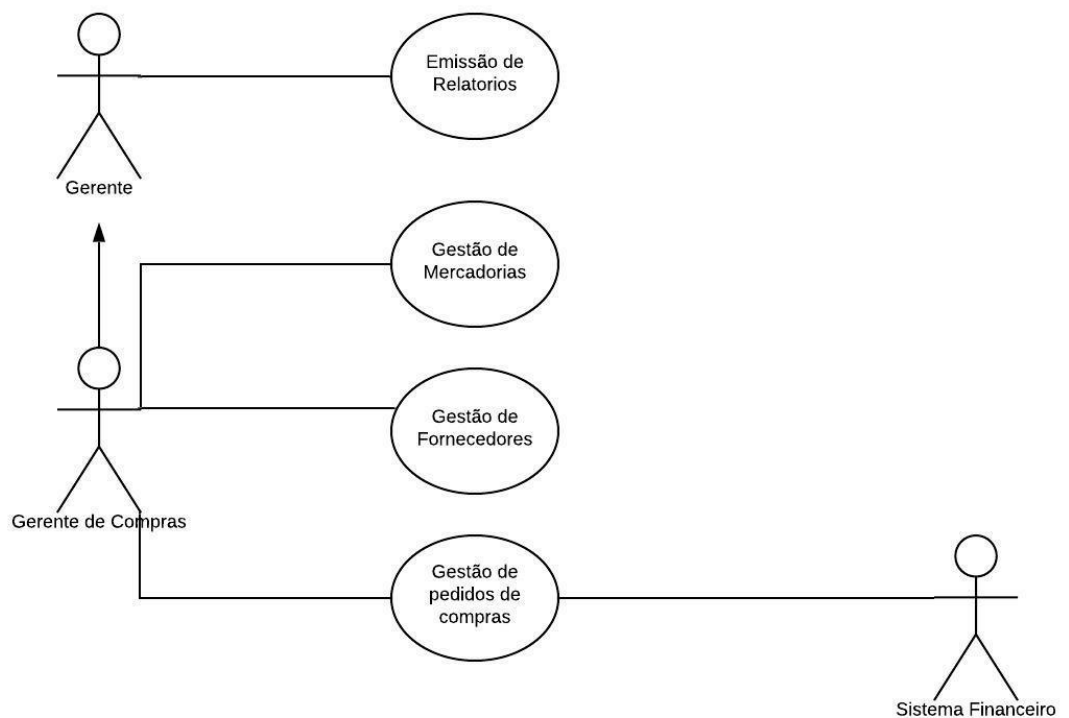
Fonte: Pádua (2009)

4.1.16. Relacionamentos entre atores e casos de uso.

De acordo com PÁDUA (2009), Diagramas de casos de uso podem especificar os relacionamentos de associação entre casos de uso e atores. As associações indicam a existência de comunicação entre atores e casos de uso. Um caso de uso pode estar associado a mais de um ator, quando a sua execução requer a participação de diferentes atores. A figura 6 indica que o gerente de compras se comunicando de forma direta com a gestão de mercadorias, e a gestão de fornecedores e a gestão de pedidos de compras, e de forma indireta com emissão de relatório, por ser uma especialização de gerentes. Gestão de pedidos de compra se comunica também com o ator sistema financeiro.

Diversos aspectos dos relacionamentos de associação, como nomes de pontas de associação, direções de navegação e multiplicidades, podem ser usados para modelar aspectos mais avançados do relacionamento entre atores e casos de uso. Entretanto, a utilização desses recursos não é usual, e deve ser reservada para modelos mais complexos.

Figura 9 Associação entre atores e casos de uso



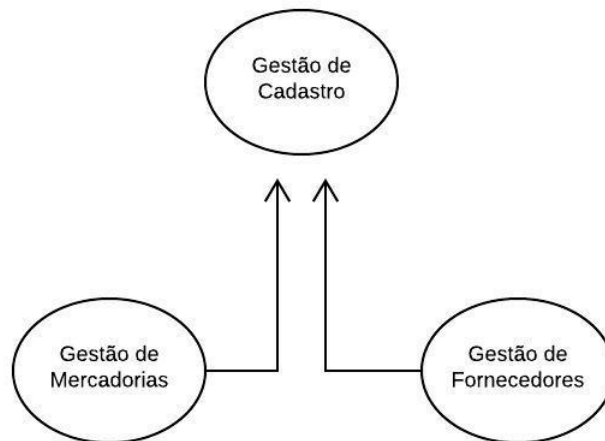
Fonte: Pádua (2009).

4.1.17. Relacionamentos entre casos de uso

Notações especiais são utilizadas para facilitar a descrição de funcionalidades mais complexa. Tal como acontece nas descrições estruturais, as funções são decompostas em casos de uso de comportamentos mais simples, combinados por meio de relacionamentos PÁDUA (2009).

Segundo Pádua (2009) relacionamentos de generalização indicam que um caso de uso mais especializado herda comportamentos de um caso de uso mais genérico, acrescentando ou redefinindo detalhes. Como acontece com as generalizações entre classes, A figura 7, gestão de cadastro é um caso de uso abstrato que representa aspectos de comportamento comuns aos casos de uso gestão de mercadorias e gestão de fornecedores.

Figura 10 Generalização de casos de uso



Fonte: Pádua (2009).

No ponto de vista de Pádua (2009) diz que o caso de uso B estende o caso de uso A, ou que existe um relacionamento de extensão de B por A, quando B representa um comportamento adicional (por exemplo, uma situação opcional ou de exceção), que normalmente não ocorre durante a execução de A. O comportamento adicional é inserido em um ponto de extensão, definido dentro do fluxo de comportamentos do caso de uso estendido. Um caso de uso pode ter vários pontos de extensão, cada ponto de extensão pode ter vários casos de uso extensores, que são executados ou não, dependendo de condições avaliadas quando o ponto de extensão é atingido, na figura 8, o caso de uso gestão de pedidos de compras tem um ponto de extensão que é atingido quando um novo pedido é completado. Os dois casos de uso extensores podem ser invocados nesse ponto, conforme se deseje uma emissão manual, on-line ou ambas.

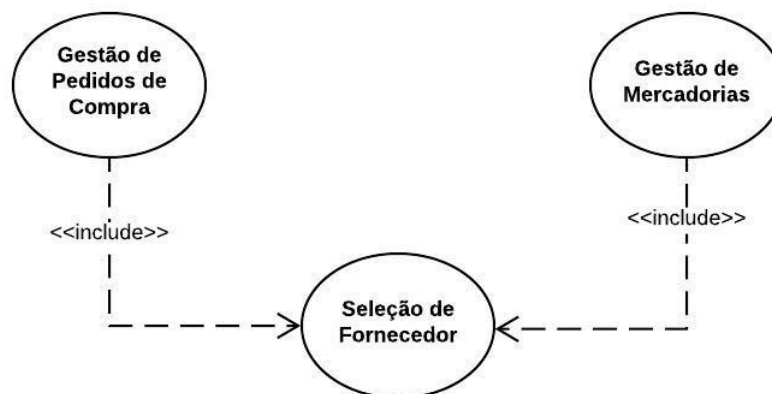
Figura 11 Extensão de caso de uso



Fonte: Pádua (2009)

Diz-se que o caso de uso A inclui o caso de uso B, ou que existe um relacionamento de inclusão de A para B, quando B representa uma atividade complexa, invocada dentro de fluxo de comportamento de A. Essa atividade pode ser comum a vários casos de uso. O caso de uso incluído é referenciado explicitamente no comportamento do caso de uso base, ao contrário do que acontece na extensão. A inclusão funciona como uma expansão de macro: é equivalente à cópia dos passos do caso de uso incluído no fluxo do caso de uso base, na figura 9, seleção de fornecedor representa um fragmento de comportamento, comum aos casos de uso gestão de pedidos de compra e gestão de mercadorias, (PÁDUA, 2000).

Figura 12 Inclusão de casos de uso



Fonte: Pádua (2009).

4.1.18. Arquiteturas de Sistemas

De acordo com Sommerville (2011), o projeto de arquitetura de software está preocupado com a compreensão de como um sistema deve ser organizado e com a estrutura geral desse sistema. No modelo de processo de desenvolvimento de software, o projeto de

arquitetura é o primeiro estágio no processo de projeto de software. É o elo crítico entre o projeto e a engenharia de requisitos, pois identifica os principais componentes estruturais de um sistema e os relacionamentos entre eles.

Segundo Sommerville (2011), pode-se projetar a arquitetura de software em dois níveis de abstração, o que ele chama de arquitetura em pequena e grande escala:

1. A arquitetura em pequena escala está preocupada com a arquitetura de programas individuais. Nesse nível, estamos preocupados com a maneira como um programa individual é decomposto em componentes e com as arquiteturas de programas.

2. A arquitetura em grande escala preocupa-se com a arquitetura de sistemas corporativos complexos que incluem outros sistemas, programas e componentes de programas. Estes sistemas empresariais estão distribuídos por diversos computadores, que podem pertencer e ser geridos por diferentes empresas.

A arquitetura de software é importante, pois afeta o desempenho e a robustez, bem como a capacidade de distribuição e de manutenibilidade de um sistema (BOSCH, 2000).

Bass et al.(2003) discutem três vantagens de projetar e documentar, explicitamente, a arquitetura de software:

- Comunicação de stakeholders. A arquitetura é uma apresentação de alto nível do sistema e pode ser usada como um foco de discussão por uma série de diferentes stakeholders.
- Análise de sistema. Tornar explícita a arquitetura de sistema, em um estágio inicial de seu desenvolvimento, requer alguma análise. As decisões de projeto de arquitetura têm um efeito profundo sobre a possibilidade de um sistema atender ou não aos requisitos críticos, como desempenho, confiabilidade e manutenibilidade.
- Reuso em grande escala. Um modelo de uma arquitetura de sistema é uma descrição compacta e gerenciável de como um sistema é organizado e como os componentes interrompem. A arquitetura do sistema geralmente é a mesma para sistemas com requisitos semelhantes e, por isso, pode apoiar o reuso de software em grande escala.

Exemplo: Cliente-Servidor

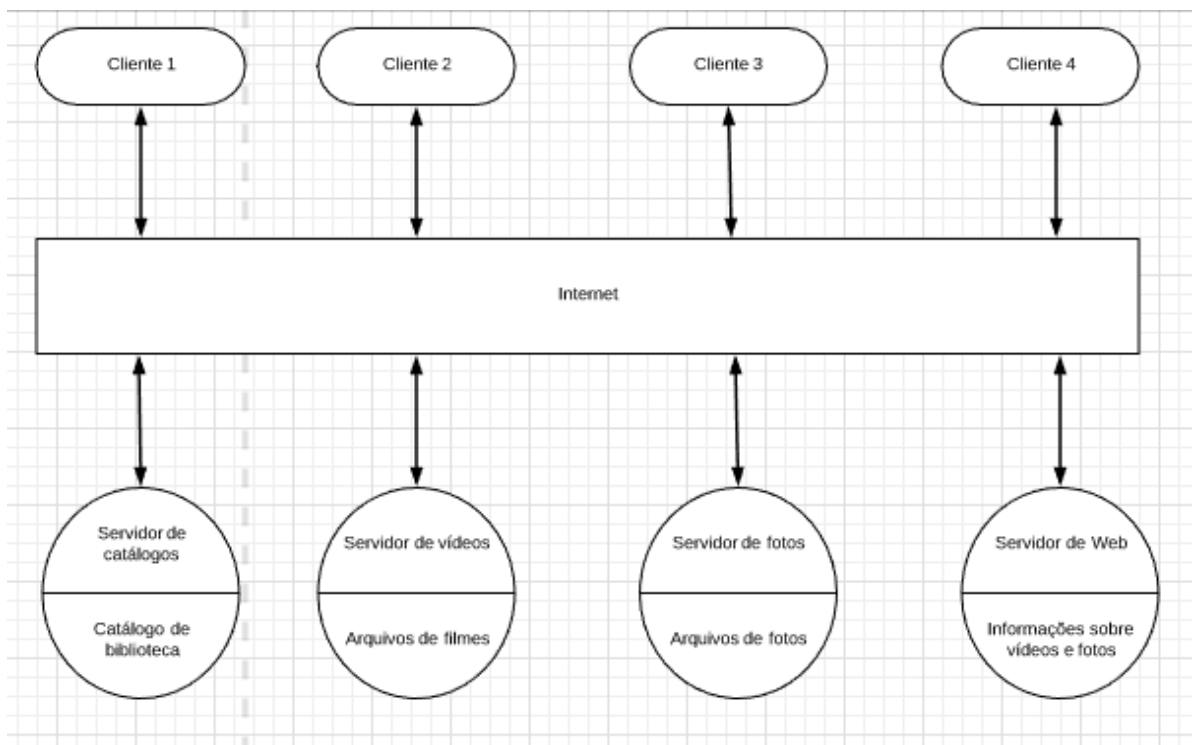
Somerville (2011) diz que um sistema que segue o padrão cliente-servidor é organizado como um conjunto de serviços e servidores 0associados e clientes que acessam e usam os serviços.

Arquiteturas cliente-servidor são normalmente consideradas arquiteturas de sistemas distribuídos, mas o modelo lógico de serviços independentes rodando em servidores separados pode ser implementado em um único computador. Novamente, um benefício é a separação e a

independência. Serviços e servidores podem ser modificados sem afetar outras partes dos sistemas.

Os clientes podem ter de saber os nomes dos servidores e serviços que eles fornecem. No entanto, os servidores não precisam conhecer a identidade dos clientes ou quantos clientes estão acessando seus serviços. Os clientes acessam os serviços fornecidos por um servidor por meio chamadas de procedimento remoto usando um protocolo de solicitação-resposta, tal como o protocolo HTTP usado na internet. Essencialmente, um cliente faz uma solitação a um servidor e espera uma resposta.

Figura 13 Arquitetura cliente-servidor para uma biblioteca de filmes.



Fonte: Somerville, (2011).

4.1.19. Banco de dados.

De acordo com Machado (2008), um banco de dados pode ser definido como um conjunto de dados devidamente relacionados. Podemos compreender como dados os objetos conhecidos que podem ser armazenados e que possuem um significado implícito, porém o significado do termo banco de dados é mais restrito que simplesmente a definição dada anteriormente. Um banco de dados possui as seguintes propriedades.

- É uma coleção lógica coerente de dados com um significado inerente; uma disposição desordenada dos dados não pode ser referenciada como banco de dados.
- Ele é projetado, construído e preenchido com valores de dados para um propósito específico; um banco de dados possui um conjunto predefinido de usuários e de aplicações.
- Ele representa algum aspecto do mundo real, o qual é chamado de minimundo; qualquer alteração efetuada no minimundo é automaticamente refletida no banco de dados.

4.1.20. Modelagem de Banco de Dados

Modelar significa criar um modelo que explique as características de funcionamento e comportamento de um software a partir do qual ele será criado facilitando o seu entendimento, diante disso é importante ressaltar que:

Modelagem de dados é o estudo das informações existentes em um contexto sob observação para a construção de um modelo de representação e entendimento de tal contexto. A modelagem de dados minera as informações que representam um contexto, estruturando-as em conjunto que denominamos de modelo lógico. (MACHADO,2008, p, 16)

Uma das principais características da modelagem de banco de dados é que ela fornece níveis de abstração de dados que omitem do usuário final detalhes sobre o armazenamento dos dados. Não existe a preocupação com um banco de dados tecnologicamente falando. O modelo de dados é um conjunto de conceitos que podem ser utilizados para descrever as estruturas lógicas e físicas de um banco de dado, (MACHADO,2008).

O objetivo de um modelo de dados é ter certeza de que todos os objetos de dados existentes em determinado contexto e requeridos pela aplicação e pelo banco de dados estão completamente representados e com precisão. Pelo fato de os dados modelados usarem

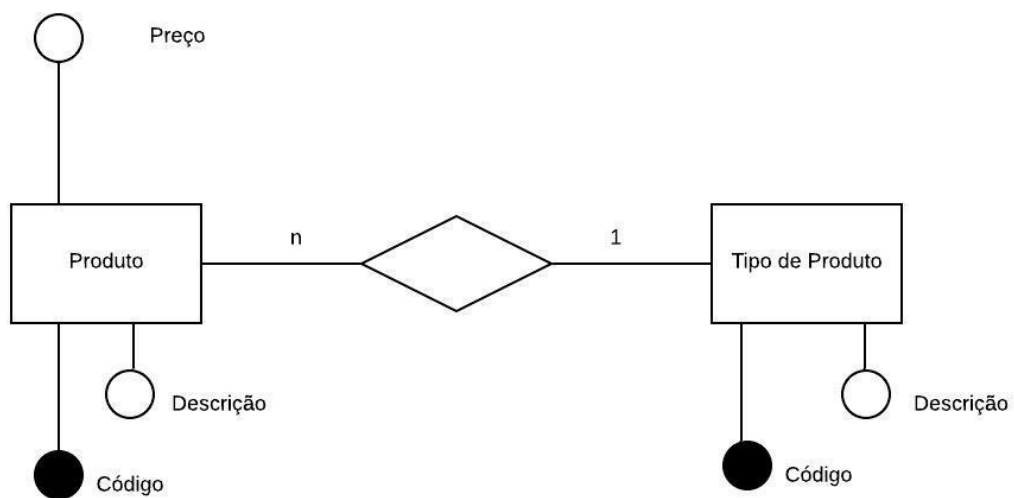
notações facilmente compreendidas e em idioma natural, eles podem ser revisados e verificados pelos usuários finais. O modelo de dados também deve ser detalhado o bastante para ser usado pelo implementador (DBA) Administrador do Banco de Dados do banco de dados. (MACHADO, 2008).

4.1.21. Modelo conceitual.

Segundo HEUSER (1998), modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um SGBD. O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD.

A técnica mais difundida de modelagem conceitual é a abordagem entidade-relacionamento (ER). Nesta técnica, um modelo conceitual é usualmente representado através de um diagrama, chamado diagrama entidade-relacionamento (DER). A Figura 1.3 apresenta um DER parcial para o problema da fábrica, HEUSER (1998).

Figura 14 Exemplo de um diagrama entidade relacionamento



Fonte: Heuser, (1998).

Entre outras coisas, este modelo informa que o banco de dados contém dados sobre produtos e sobre tipo de produtos. Para cada tipo de produto, o banco de dados armazena o código, a descrição, preço, bem como o tipo de produto ao qual está associado. Para cada tipo de produto, o banco de dados armazena o código, a descrição, bem como os produtos daquele tipo.

4.1.22. Modelo lógico.

Na opinião Heuser (1998, p.17) “um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD. Assim, o modelo lógico é dependente do tipo particular de SGBD que está sendo usado”. A seguir trataremos apenas modelos lógicos referente a SGBD relacional, em um SGBD relacional os dados são organizados na forma de tabelas.

A Figura 12 mostra um exemplo de BD relacional projetado a partir do modelo conceitual mostrado na Figura 11. Um modelo lógico para o BD acima deve definir quais as tabelas que o banco contém e, para cada tabela, quais os nomes das colunas.

Figura 15 - Exemplos de tabelas de um Banco de Dados relacional

CodTipoProd	DescrTipoProd
1	Computador
2	Impressora

CodProd	DescrProd	PrecoProd	CodTipoProd
1	PC desktop modelo X	2.500	1
2	PC notebook ABC	3.500	1
3	Impressora jato de tinta	600	2
4	Impressora laser	800	2

Fonte: Heuser (1998).

O modelo lógico descreve a estrutura do banco de dados, conforme vista pelo usuário do SGBD. Detalhes de armazenamento interno de informações, que não tem influência sobre a programação de aplicações no SGBD, mas podem influenciar a performance da aplicação (por exemplo, as estruturas de arquivos usadas no acesso às informações) não fazem parte do modelo lógico. Estas são representadas no modelo físico. Modelos físicos não são tratados neste livro. Eles são usados apenas por profissionais que fazem sintonia de banco de dados, procurando otimizar a performance. As linguagens e notações para o modelo físico não são padronizadas e variam de produto a produto. A tendência em produtos mais modernos é esconder o modelo físico do usuário e transferir a tarefa de otimização ao próprio SGBD

4.1.23. Mysql.

O MySQL teve origem quando os desenvolvedores David Axmark, Allan Larsson e Michael "Monty" Widenis, na década de 90, precisaram de uma interface SQL compatível com as rotinas ISAM que utilizavam em suas aplicações e tabelas. Em um primeiro momento, tentaram utilizar a API mSQL, contudo a API não era tão rápida quanto eles precisavam, pois utilizavam rotinas de baixo nível (mais rápida que rotinas normais). Utilizando a API do mSQL, escreveram em C e C++ uma nova API que deu origem ao MySQL (MILANI, 2006)

Com o ótimo resultado gerado por essa nova API, o MySQL começou a ser difundido e seus criadores fundaram a empresa responsável por sua manutenção que é a MySQLAB.

Partir dessa fase, o MySQL tornou-se mais conhecido por suas características de rápido acesso e cada vez mais utilizado. Novas versões foram lançadas, contemplando novas necessidades e firmando, assim, sua posição no mercado. Sua mais recente versão é a 5.1, a qual conta com novos recursos, estabelecendo sua capacidade para competir com os bancos de dados privados de maior popularidade (SQL Server e Oracle).

5. ESPECIFICAÇÃO DO SISTEMA

5.1 Levantamento de informações

Um sistema que atenda a grande maioria das necessidades dos usuários tem grandes chances de ser um sistema de sucesso.

O Projeto Cleanex utiliza a técnica Brainstorming para levantamento de requisitos, que basicamente é reunir um grupo de pessoas e gerar ideias, discutir, e colaborar para o desenvolver a mesma. Afirma COUTINHO, BOTTENTUIT (2007):

O brainstorming é uma técnica de recolha de informação muito utilizada na investigação em Ciências Sociais e Humanas com o objetivo de explorar novas ideias sobre um tema ou alternativas de solução para problemas de mais diversa índole seja em organizações, empresas, negócios, etc. (p.107)

Requisitos Funcionais

- RF00 – Senha e Login
- RF001 – Cadastrar Usuários
- RF002 – Alterar Cadastro
- RF003 – Excluir Cadastro
- RF004 – Consultar Usuário
- RF005 – Cadastrar Produtos

RF006 – Excluir Produtos
 RF007- Ver Detalhes dos Produtos
 RF008 – Adicionar Produtos ao carrinho
 RF009- Excluir Produtos do carrinho
 RF0010-Escolher Forma de Pagamento
 RF011- Confirmar Compra

Requisitos não funcionais

RNF001 – Interface Gráfica Materialize, CSS, HTML5
 RNF002 – JavaScript, PHP.
 RNF003 – Banco de Dados MySQL
 RNF004 – Servidor Xampp.
 RNF005 – Navegadores, Google Chrome, Mozilla Firefox, Safari, Opera

Tabela 1. Requisitos Funcionais

ID	Descrição	Prioridade	Categoria
RF00	Utilização de login e senha para confirmar a compra	Alta	Segurança
RF001	Cadastrar usuários no sistema e dados pessoais	Alta	Usabilidade
RF002	Alterar cadastro e dados pessoais de usuários	Alta	Usabilidade
RF003	Excluir cadastro e dados de usuários	Médio	Usabilidade
RF004	Consultar usuários cadastrados no sistema	Médio	Usabilidade
RF005	Cadastra Produtos	Alto	Usabilidade
RF006	Excluir Produtos	Alto	Usabilidade
RF007	Ver detalhes dos Produtos	Alto	Usabilidade
RF008	Adicionar Produtos ao carrinho	Alto	Usabilidade
RF009	Excluir Produtos do carrinho	Alto	Usabilidade
RF010	Escolher Forma de Pagamento	Alto	Usabilidade
RF011	Confirmar Compra	Alto	Usabilidade

Fonte: elaborado pelos Autores

Tabela 2. Requisitos Não Funcionais

ID	Descrição	Prioridade	Categoria
RNF001	Utilização do Materialize, CSS, HTML5	Alta	Usabilidade
RNF002	Linguagem de programação JavaScript, PHP	Médio	Portabilidade
RNF003	Banco MySQL usado pra guarda dados	Alto	Segurança
RNF004	Servidor Xampp	Alto	Portabilidade
RNF005	Navegadores, Google Chrome, Mozilla Firefox, Safari, Opera	Médio	Usabilidade

Fonte: elaborado pelos Autores

5.2 Especificação do Banco de Dados

O projeto Cleanex – e-commerce de produtos de limpeza, utilizou o Gerenciador de banco de dados MySQL, para fazer a manipulação dos dados, o MySQL é algumas dependências da linguagem PHP, necessárias para o desenvolvimento da aplicação. Como Data base foi nomeado de “SITE” Abaixo mostraremos o script do banco de dados.

5.3 IDE – Ambientes de Desenvolvimento

IDE, ou Ambiente Integral de Desenvolvimento em tradução livre, é um software criado com a finalidade de facilitar a vida dos programadores. Neste tipo de aplicação estão todas as funções necessárias para o desenvolvimento desde programas de computador a aplicativos mobile, assim como alguns recursos que diminuem a ocorrência de erros nas linhas de código (NOVAES, 2014).

Se no passado os desenvolvedores precisavam apenas de um editor de texto e de um navegador para criar um software, agora, com os IDEs, eles possuem mais opções para otimizar o tempo gasto com os códigos. Imagine os IDEs como as calculadoras. Logicamente

you learn to do mathematical operations in school, but rarely do it manually when you need it.

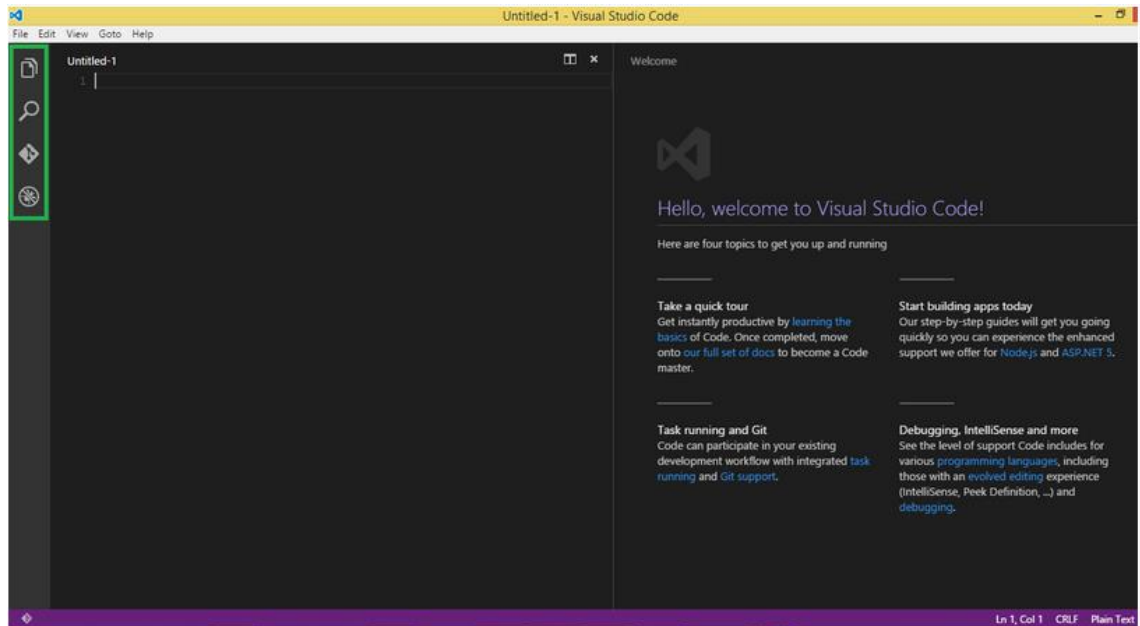
One of the main advantages of IDEs is the ability to compile complete language libraries. Another function quite common in this type of program are the debuggers, which point out errors that occasionally can occur when writing code. Some IDEs also have auto-completion.

Already the disadvantage is due to the need for a reasonable knowledge of programming. Users with little experience – or who are taking the first steps in software development – can get confused with the excess of resources that some IDEs have.

In the Cleanex project – we use an IDE known as Visual Studio Code. In 2015 it was launched by Microsoft as an editor of code designed for the development of web applications called Visual Studio Code, or simply VSCode. Announced during the Build, event aimed at developers that occurs in the United States annually. It is a light and multiplatform tool that is available both for Windows, as well as for Mac OS and Linux and serves a wide range of projects, not only ASP.NET, but also Node.js. Additionally, the editor has support for the syntax of various languages such as Python, Ruby, C++.

Besides being completely free, even in the second semester of the year of launch, during the event Connect(), the editor was announced as open source, with code available on GitHub, which allows the technical community to contribute to its development and facilitate the creation of extensions and new functionalities.

Figura 16 Tela inicial VScode



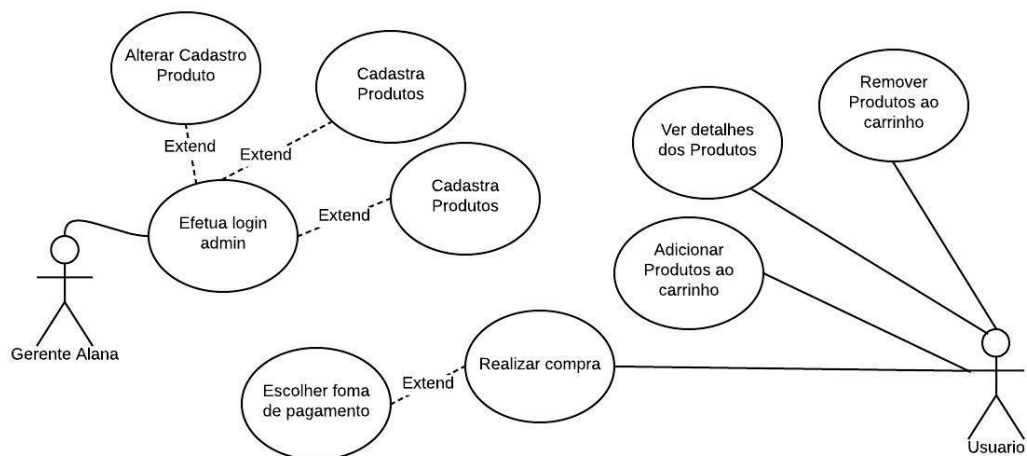
. Fonte Devmedia, (2017).

5.4 Linguagem de Programação

O Projeto Cleanex – e-commerce de produtos de limpeza foi realizado a partir das linguagens HTML, PHP, CSS, JavaScript, que foi escolhida pela equipe de desenvolvimento Devido a ser do conhecimento de todos.

Diagrama de Caso de Uso - onde descrevemos a funcionalidade propostas do nosso sistema.

Figura 17 Caso de uso Cleanex



Fonte: Soares, (2019).

6 ESPECIFICAÇÃO DA ARQUITETURA DO SISTEMA

O projeto Cleanex é um e-commerce de produtos de limpeza, feito para a empresa Alana produtos de limpeza, que vai expor seus produtos, com o intuito de vender para empresas que consomem uma grande quantidade de produtos de limpeza.

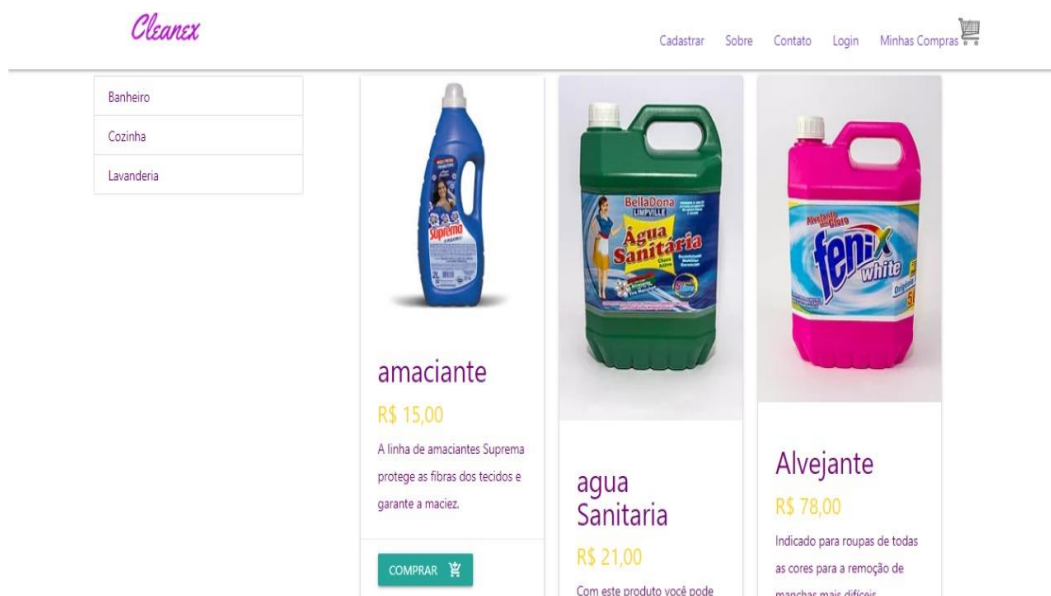
6.1 Especificação da aplicação

Aqui mostraremos alguns detalhes do sistema e suas principais funcionalidades.

6.1.1. Tela Principal do e-commerce

A tela principal exibe os produtos para a venda as suas categorias do tipo banheiro, cozinha, lavanderia, ao clicar nos produtos você será redirecionado para uma outra pagina onde verá mais detalhes dos produtos e poderá adicionar ao carrinho de compras, também tem a opção de navegar, em outras páginas como sobre, contato, login meu carrinho, cadastrar.

Figura 18 Pagina Home



Fonte: Alunos Projeto Cleanex, (2019).

6.1.2. Cadastro de Produtos

Na tela cadastro de produtos o usuário administrador do sistema, no caso Alana Produtos de Limpeza, poderá cadastrar toda a sua linha de produtos que será exposto na página principal.

Figura 19 Pagina Cadastro de produtos

Cleanex

Cadastrar Sobre Contato Login Minhas Compras

Nome

Preço

Categoria

Nome da Imagem

Descrição

ENVIAR

Acompanhe as novidades nas redes sociais!

f t g+

Copyright © Cleanex.com.br

Fonte: Alunos Projeto Cleanex, (2019).

6.1.3. Contatos

A tela contatos é feita para os clientes que quiserem entrar em contato sobre alguma dúvida, e que queira dar sugestão para melhorias, deixar seu feedback, e bem simples de usar basta colocar o nome e um e-mail válido e escrever sua mensagem e enviar.

Figura 20 Pagina Contato

Cleanex

Cadastrar Sobre Contato Login Minhas Compras

Contato

Nome Email

Mensagem

ENVIAR

Acompanhe as novidades nas redes sociais!

f t g+

Fonte: Alunos Cleanex, (2019).

6.1.4 Sobre

A tela sobre, contém informações da empresa contando um pouco da sua história do começo até os dias de hoje

Figura 21 Pagina Sobre

Cleanex

Cadastrar Sobre Contato Login Minhas Compras

CLEANEX

O e-commerce Cleanex tem o intuito de facilitar a venda online de produtos de limpeza artesanais, facilitando a venda entre o produtor e o consumidor final. Os principais consumidores finais são clientes MEI, ou condomínios.

Endereço: esquina com 4ª avenida, R. 2300, 1100 - Centro, Balneário Camboriú - SC

Telefone: (47) 3228-0141

Acompanhe as novidades nas redes sociais!

f t g+

Copyright © Cleanex.com.br

Fonte: Alunos Projeto Cleanex, (2019).

6.1.5. Carrinho

Na tela carrinho de compras voce podera ver todos os produtos que voce escolheu antes de confirmar sua compra, mais para confirmar a compra o usuario devera fazer o login caso não esteja cadastrado devera realizar o cadastro, e depois podera confirmar sua compra e escolher forma de pagamento calcular frete.

Figura 22 Pagina Carrinho de compras

The screenshot displays the Cleanex shopping cart interface. At the top, the Cleanex logo is on the left, and navigation links for 'Cadastrar', 'Sobre', 'Contato', 'Login', and 'Minhas Compras' are on the right. The main content area lists three items:

- amaciante**: Price \$15, with a 'REMOVE' button and quantity controls (minus, 1, plus).
- agua Sanitaria**: Price \$21, with a 'REMOVE' button and quantity controls (minus, 1, plus).
- Alvejante**: Price \$78, with a 'REMOVE' button and quantity controls (minus, 1, plus).

On the right side, a 'DETALHES DA COMPRA' table shows the following details:

Preço (3 itens)	R\$114
Frete	Gratuito
Total	R\$114


The footer features social media icons for Facebook, Twitter, and Google+, the text 'Acompanhe as novidades nas redes sociais!', and the copyright notice 'Copyright © Cleanex.com.br'.


Fonte: Alunos Projeto Cleanex (2019).

6.1.7. Login

A tela login é uma tela de autenticação, ou seja, serve para se identificar, o usuário que não efetuar login não poderá confirmar compra, se não for cadastrado basta se cadastra com nome e-mail e senha.

Figura 23 Pagina Login

Cleanex [Cadastrar](#) [Sobre](#) [Contato](#) [Login](#) [Minhas Compras](#) 


Cleanex
Soluções em Produtos de Limpeza por Atacado

E-mail

E-mail

E-mail

Senha

[Esqueceu sua senha?](#)

[LOGIN](#)

[Cadastrar-se](#)

Acompanhe as novidades nas redes sociais!

[f](#) [t](#) [g+](#)

Copyright © Cleanex.com.br

Fonte: Alunos Projeto Cleanex, 2019

REFERÊNCIAS

ANSELMO, Fernando. **PHP e MySQL o maior e melhor sem cortes**. Florianópolis SC 2. ed. Visual Books. 2002.

COUTINHO, Clara Pereira; BOTTENTUIT Junior, João Batista. **Utilização da técnica do brainstorming na introdução de um modelo de e/b-learning numa escola profissional portuguesa: a respectiva de professores e alunos**. São Paulo 2007.

GOMES, Carlos Roberto. **Marketing, Telemarketing & E-commerce**: Viena. 3. ed. São Paulo: 2011

HEUSER, Carlos Alberto. **Projeto de Banco de dados**. 4. ed. Sagra Luzzatto, São Paulo.

KUROSE, James. **Redes de computadores e a Internet**. 3 ed. Person Addison Wesley, São Paulo, 2006.

MILANI, André: **MySQL: Guia do Programador**. São Paulo: Novatec,2006. E-Book. Disponível em: <<https://books.google.com.br/books?hl=pt-BR>>. Acesso em: 22 nov. 2019.

NOVAES, Rafael. **IDÉ: O que é e para que serve IDÉ**. ago. 2014. Disponível em: <<https://www.psafte.com/blog/o-que-serve-ide/>> 24 nov. 2019.

PÁDUA, Wilson. Engenharia de software: fundamentos métodos e padrões, março 2000. Disponível em: <http://aulasprof.6te.net/arquivos-aulas/07-proces-desen-soft/livro-eng-soft-fund-met-padroes.pdf> > Acesso em: 20 nov 2019.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Pearson Makron Books, 2006.
SILVA, Maurício Samy. **Construindo Sites com CSS e (X) HTML: Sites controlados por folhas de estilos em cascata**. São Paulo: Novatec,2008. E-Book. Disponível em: <<https://books.google.com.br/books?hl=pt->>. Acesso em: 24 nov. 2019.

SILVA, Maurício Samy. **Criando sites com HTML**. Sites de alta qualidade com HTML e CSS. 3 ed. São Paulo. 2014.

SILVA, Maurício Samy: **JavaScript: Guia do Programador**. São Paulo: Novatec,2010. E-Book. Disponível em: <<https://books.google.com.br/books?hl=pt->>. Acesso em: 26 nov. 2019.

SOMMERVILLE, Lan. Engenharia de Software. 9. ed. São Paulo: Person, 2011.

TEIXEIRA, Tarciso. **Comercio eletrônico: Conforme o Marco Civil da Internet e a regulamentação do e-commerce no Brasil**. São Paulo: Saraiva, 2015. E-Book. Disponível em: <<https://books.google.com.br/books?hl=pt->>. Acesso em: 24 nov. 2019.

VSCODE: Introdução ao Visual Studio Code. Devmedia, 2016. Disponível em: <https://www.devmedia.com.br/introducao-ao-visual-studio-code/34418> . Acesso em: 26 nov. 2019.

APÊNDICE A – Título do apêndice

Textos e/ou documentos elaborados pelo autor, a fim de completar a sua argumentação, caso seja necessário.

ANEXO A – TÍTULO DO ANEXO

Textos e/ou documentos elaborados por outros autores, a fim de completar a sua argumentação, caso seja necessário.