



**INSTITUTO POLITÉCNICO INDUSTRIAL DE LUANDA -
IPIL/MAKARENCO ÁREA DE FORMAÇÃO DE INFORMÁTICA CURSO
GESTÃO DE SISTEMA INFORMÁTICA**

SISTEMA PARA AGENDAMENTO DE CONSULTAS

REALIZADO POR:

EMÍLIA DA COSTA MANUEL

Luanda – LDA, Ano 2021

**INSTITUTO POLITÉCNICO INDUSTRIAL DE LUANDA -
IPIL/MAKARENCO ÁREA DE FORMAÇÃO DE INFORMÁTICA CURSO
GESTÃO DE SISTEMA INFORMÁTICA**

TEMA:

PÁGINA WEB

SUBTEMA: SISTEMA PARA AGENDAMENTO DE CONSULTAS

REALIZADO POR:

EMÍLIA DA COSTA MANUEL

Trabalho de conclusão de curso apresentado ao Instituto Politécnico Industrial de Luanda – Área de Informática, como requisito parcial para obtenção do título de Técnico Médio, Curso de Gestão de Sistema Informática sob as orientações do Orientador: Prof. Eng.^º Nelson Mário Tito.

Orientador: Prof. Nelson Mário Tito

Luanda – LDA, Ano 2021

DEDICATÓRIA

Este trabalho é dedicado primeiramente a Jeová Deus, como Originador da vida, meu guia e Ajudador em todos os momentos de aflições.

Dedico também esse trabalho para os meus pais e para todos aqueles me ajudaram direita ou indiretamente na realização desse Projeto.

AGRADECIMENTOS

Agradeço a Deus primeiramente pela saúde e força que tive para a realização desse projeto. Agradeço também aos meus pais pelo apoio e por me darem apoio físico, emocional, espiritual e financeiro.

Em seguida o meu agradecimento vai para os meus amigos, irmãos, professores que deram-me bastante força para a elaboração desse projeto.

Apesar das circunstâncias difíceis que nos encontramos por causa da pandemia COVID-19 agradeço ao meu orientador “**Nelson Mário Tito**” pelo apoio e o esforço que teve de mantermos em contato e tirar tempo para orientar-me e compartilhar sua experiência profissional e instruções que me ajudaram no melhoramento desse projeto.

RESUMO

A criação das páginas web têm ajudado muito e facilitado a vida das pessoas nos últimos anos. O uso de tecnologias modernas é um dos aspectos que países desenvolvidos possuem.

Quanto mais usuários dispostos a fazer uso das páginas web, mais dispositivos eletrônicos serão usados em função do tempo.

Sistema para Agendamento de consultas é o foco principal deste presente trabalho, e está dividido em três partes respetivamente: Fundamentação Teórica, falou-se propriamente sobre páginas web, em seguida foi Programação em camada onde abordou-se conceitos teóricos sobre o mesmo e por último a implementação onde foi apresentada uma série de ilustrações relacionadas ao funcionamento da aplicação

ABSTRACT

LISTA DE ILUSTRAÇÕES

Fig1-Exemplo de uma página web – youtube.com

Fig2- logo HTML

Fig3-logo CSS

Fig4-logo JavaScript

Fig5-logo PHP

Fig6- logo Xamp

Fig7-logo Wampserver

Fig8-logo MsQL WorkBench

Fig9-logo Font- Awoseme

Fig10-logo Visual Studio Code

Fig11-logo Chrome

LISTA DE TABELAS

Tabela 1. Requisitos Funcionais da página web

Tabela 2. Requisito Não Funcionais da Página web

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

HTML HyperText Markup Language

SQL Structured Query Language

PHP Personal Home Page ou Hypertext Preprocessor

UML Unified Modeling Language

HTTP Hypertext Transfer Protocol

DML Data Manipulation Language

Sumário

INTRODUÇÃO.....	Erro! Marcador não definido.
CAPÍTULO I – FUNDAMENTAÇÃO TEÓRICA	Erro! Marcador não definido.
1. PAGINAS WEB.....	Erro! Marcador não definido.
1.1. Desenvolvimento WEB	Erro! Marcador não definido.
1.2. Tipos de Páginas Web.....	Erro! Marcador não definido.
1.3. Desenvolvimento Front- End e Back-End	Erro! Marcador não definido.
1.4. Exemplos de framework's front-end e back-end	Erro! Marcador não definido.
1.5. Principais linguagens Front-End.....	Erro! Marcador não definido.
1.6. História sobre HTML.....	Erro! Marcador não definido.
1.6.1. CSS	Erro! Marcador não definido.
1.6.2. CSS	Erro! Marcador não definido.
1.6.3. Aplicação do Selectores em CSS.....	Erro! Marcador não definido.
1.6.4. Sintaxe do CSS	Erro! Marcador não definido.
1.7. JavaScript.....	Erro! Marcador não definido.
1.7.1. Utilização do JavaScript	Erro! Marcador não definido.
1.7.2. O uso do javaScript em HTML	Erro! Marcador não definido.
1.7.3. Principais Linguagens Beck-End.....	Erro! Marcador não definido.
1.8. História sobre o PHP	Erro! Marcador não definido.
1.8.1. Alguns Suportes do PHP	Erro! Marcador não definido.
1.8.2. Funcionamento do PHP no servidor.....	Erro! Marcador não definido.
2.1. Exemplos de bases de dados	Erro! Marcador não definido.
2.2. Classificação das bases de dados	Erro! Marcador não definido.
As bases de dados classificam-se em:	Erro! Marcador não definido.

- a) Quanto ao Modelo de Dados..... **Erro! Marcador não definido.**
 - b) Quanto ao Número de Usuários **Erro! Marcador não definido.**
 - c) Quanto á Localização **Erro! Marcador não definido.**
- 2.3. Aplicações de base de dados..... **Erro! Marcador não definido.**
 - 2.4. Segurança em bases de dados **Erro! Marcador não definido.**
 - 2.5. Sistemas Gerenciadores de bases de dados (SGBD)**Erro! Marcador não definido.**
- 2.2. O Modelo Relacional **Erro! Marcador não definido.**
- 2.2.1. Entidades ou tabelas **Erro! Marcador não definido.**
 - 2.2.3. Normalização..... **Erro! Marcador não definido.**
- 2.3. Formas Normais **Erro! Marcador não definido.**
- 2.3.1. Primeira Forma Normal..... **Erro! Marcador não definido.**
 - 2.3.2. Segunda Forma Normal..... **Erro! Marcador não definido.**
 - 2.3.3. Terceira forma Normal **Erro! Marcador não definido.**
 - 2.3.4. Terceira Forma Normal **Erro! Marcador não definido.**
 - 2.3.5. Quarta Forma Normal..... **Erro! Marcador não definido.**
 - 2.3.6. Structured Query Language (SQL)..... **Erro! Marcador não definido.**
 - 2.3.7. Sublinguagens do SQL **Erro! Marcador não definido.**
- 2.4. Aplicações de páginas web **Erro! Marcador não definido.**
- 2.4.1. Xamp **Erro! Marcador não definido.**
 - 2.4.2. Wampserver..... **Erro! Marcador não definido.**
 - 2.4.3. MySql Workbench..... **Erro! Marcador não definido.**
 - 2.4.4. Font Awesome..... **Erro! Marcador não definido.**
 - 2.4.5. Visual Studio Code..... **Erro! Marcador não definido.**
- 2.5. Navegadores web **Erro! Marcador não definido.**
- 2.5.1. Google Chrome **Erro! Marcador não definido.**
- 3. Modelo em quatro Camadas..... **Erro! Marcador não definido.**

- 3.1. Modelo em três camadas..... **Erro! Marcador não definido.**
- 3.2. Aplicações monolíticas (uma camada) **Erro! Marcador não definido.**
- 3.3. Aplicações em duas camadas..... **Erro! Marcador não definido.**
- 3.4. Aplicações em três camadas **Erro! Marcador não definido.**
 - 3.4.1. MVC – Model, View, Controller..... **Erro! Marcador não definido.**
 - 3.4.2. POO **Erro! Marcador não definido.**
- 4.1. Objeto de estudo **Erro! Marcador não definido.**

INTRODUÇÃO

As páginas web's estão presentes em nossas vidas, sendo possível acessar em vários dispositivos como (computadores, telefones, tablets e outros). Com a facilidade de acesso a internet as aplicações páginas/web, clínicas, restaurantes, instituições de ensino e outros, podem publicitar e dar a conhecer ao público em geral o seu produto e apresentar o seu negócio e a maior vantagem é que os interessados poderão saber disso tudo sem sair de casa ou sua zona de conforto.

DEFINIÇÃO DO PROBLEMA

Baseado em resultados obtidos através de Pesquisas em algumas clínicas e Centros Médicos notou-se a grande dificuldade por parte dos Doutores e Pacientes em fazer a marcação de consultas atualmente devido a situação que nos encontramos.

Além disso, existe ainda a dificuldade por parte de muitas clínicas para fazer o controlo de marcação de consultas. Também notou-se as dificuldades por parte dos Pacientes e Doutores em cumprir os horários de marcações de consultas.

Justificativa:

Para a resolução deste problema, decidi criar um site eficiente, e seguro, para fazer agendamento de consultas em online.

Isso eliminará todas as falhas na gestão de marcação de consultas, facilitará o processo de atendimento e certamente tornará seguro toda a gestão.

Ele será simples a nível visual, mas complexo nas funcionalidades, facilitará ao máximo o trabalho de todos os usuários.

O Site permitirá fazer o gerenciamento, e marcação de consultas em online que facilitará todos os pacientes marca as consultas de modo fácil sem deslocamento desnecessário evitando também aglomeração de pacientes no centro médico Sol.

OBJETIVOS

Objetivo geral:

- ❖ Elaborar uma aplicação web para agendamento de consultas em qualquer clínica.

Objetivos específico

- ❖ Dar acesso e permitir a marcação de consultas em online.
- ❖ Fazer gerenciamento das consultas como, aceitar consulta, negar, anular, fechar consulta marca falta caso o usuário agende uma consulta e não apareça na clínica.

CAPÍTULO 1 – FUNDAMENTAÇÃO TEÓRICA

1. Páginas web

Uma página web, também conhecida pelo no inglês webpage, é uma “pagina” na world wide web, geralmente em formato HTML e com ligações de hipertexto que permitem a navegação de uma pagina, ou secção, para outra. As páginas web usam com frequência ficheiros gráficos associados para fins de ilustração, e também estes ficheiros podem ser ligações clicáveis. Uma página web é apresentada com o recurso a um navegador, ou browser, e pode ser construída por forma a recorrer a applets (subprogramas que ocorrem dentro da página), que muitas vezes fornecem gráficos em movimento, interação com o utilizador e som.

As páginas web podem ser maiores do que aquilo que cabe no ecrã. Exceto em casos muito especiais, uma página mais larga do que ecrã requerendo deslocamento (scrolling) horizontal, é pouco prática e por isso evitada. Uma página que é mais alta do que o ecrã é mais comum e não levanta problemas; requer deslocamento (scrolling) vertical para ser toda vista.

Uma coleção de páginas web armazenadas numa única pasta ou em subpastas relacionadas de um servidor web é conhecida como site. Um sitio web costuma incluir uma página principal chamada índice.html.

Uma dificuldade no desenho e teste de páginas web é torná-las adequadas a muitos navegadores e definições, e a diferentes resoluções de ecrã.

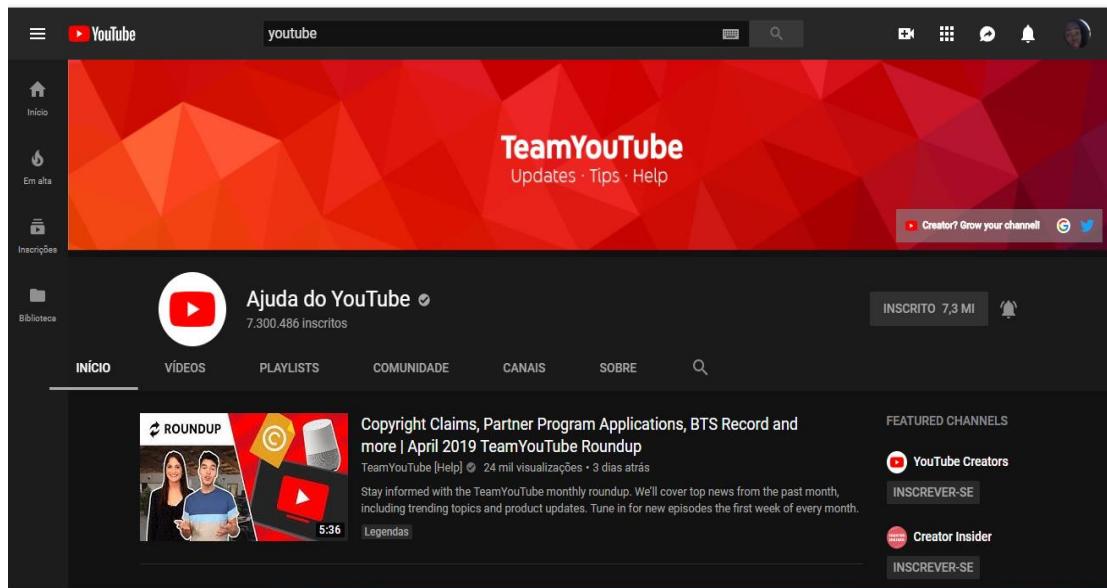


Fig1-Exemplo de uma página web – youtube.com

1.2. Desenvolvimento WEB

Desenvolvimento web é o termo utilizado para descrever o desenvolvimento de sites na internet ou numa intranet. Estes são o profissional que trabalha desenvolvendo websites, podendo ser um web Designer (Desenvolvedor do Layout), ou web Developer (Desenvolvendo de sistemas). O desenvolvimento refere-se a um processo de construção e testes do software específico para a web, com a finalidade de se obter um conjunto de programas, que satisfazem as funções pretendidas, quer em termos de usabilidade dos usuários ou compatibilidade com outros programas existentes. O desenvolvimento web pode variar desde simples páginas estáticas ricas, comércios eletrônicos ou redes sociais.

1.2.1. Tipos de Páginas Web

- ❖ Página Web estática também é chamada de Página Web Side.
- ❖ Página Web dinâmica também chamada de Página Server Side.

Uma página web é estática quando ela é interpretada pelo browser. Na página estática qualquer alteração que precisa ser feita, terá de se recorrer ao código fonte da mesma. Ela não contém mecanismo de auto gerenciamento, não contém interação com usuário, não acessa nem manipula banco de dados.

Uma página diz-se dinâmica quando ela é interpretada pelo servidor. Se o usuário quiser fazer alterações na mesma, essas poderão ser feitas dentro da própria pois ela possui mecanismo de auto gerenciamento, acesso e manipula banco de Dados.

1.3. Desenvolvimento Front- End e Back-End

Front-end, interface frontal ou parte frontal e back-end, parte secundária, parte de suporte ou parte de retaguarda são termos generalizados que se referem ás etapas inicial e final de um processo. O front- end é responsável por coligir a entrada do usuário em varias formas e processa-la para adequá-la a uma especificação em o back-end a possa utilizar.

Em compiladores, o front-end traduz um código fonte de programa de computador em uma representação intermediária e o beck-end trabalha com a representação intermediaria para produzir código em uma linguagem de saída. O back-end normalmente otimiza-se para produzir código que execute mais rápido. A distinção de front-end/back-end pode separar a seção parser que lida com o código fonte e o back-end que gera código e o optimiza. Alguns projetos, como GCC, oferecem escolhas entre vários front-ends (analizando linguagens fonte diferentes) ou back-ends (gerando código para processadores de alvo diferentes).

Em síntese de voz, o front-end refere-se á parte do sistema de síntese que converte o texto de entrada em uma representação fonética simbólica e o back-end converte a representação fonética simbólica em sons reais.

No contexto de aplicações WWW, um mediador é um serviço que funciona simultaneamente com um servidor em seu front-end e com um cliente em seu back-end.

1.3.1. Exemplos de framework's front-end e back-end

Front-End

❖ ReactJS

❖ HTML5

❖ CSS3

Back-end

- ❖ PHP
- ❖ Python

1.3.2. Principais linguagens Front-End

HTML (abreviação para a expressão inglesa Hyper Text Markup Language, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretadas por navegadores. A tecnologia é fruto da junção entre os padrões Hytime e SGML.

A verdade é que o HTML é uma linguagem que serve para formatações, ela apenas serve para definir conteúdo da página e descrever o significado de cada elemento. Não cabe ao HTML fornecer informações sobre apresentação dos elementos cores, tamanhos dos textos e posicionamento.

Se escrevemos um HTML, simples num arquivo html basta abri-lo no navegador e pronto, já podemos visualizar a página. Mas quando envolvemos um servidor o processo não é tão simples.

HyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.) conectados por hiperligações. O padrão é independente de outros padrões de processamento de texto em geral.

SGML é um padrão de formatação de textos. Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objetos para descrever as ligações.

1.4. História sobre HTML

Tim Berners-Lee (físico britânico) criou o HTML original

(e outros protocolos associados como o HTTP) em uma estação NexTcube usando o ambiente de desenvolvimento NexTSTEP.

Na época a linguagem não era uma especificação, mas uma coleção de ferramentas para resolver um problema de Tim: a comunicação e disseminação das pesquisas entre ele e seu grupo de colegas. Sua solução, combinada com a então emergente internet pública (que tornar-se-ia a internet) ganhou atenção mundial.

As primeiras versões do HTML foram definidas com regras sintáticas flexíveis, o que ajudou aqueles sem familiaridade com a publicação na Web. Através do tempo, a utilização de ferramentas para autoria de HTML aumentou, assim como a tendência em tornar a sintaxe cada vez mais rígida. Apesar disso, por questões históricas (retrocompatibilidade), os navegadores ainda hoje conseguem interpretar páginas web que estão longe de ser um código HTML válido.

A linguagem foi definida em especificações formais na década de 1990, inspirada nas propostas originais de Tim Berners-Lee em criar uma linguagem baseada em SGML para a internet. A primeira publicação foi esboçada por Berners-Lee e Dan Connolly, e publicada em 1993 na IETF como uma publicação formal para o SGML (com uma DTD em SGML definida a gramática). A IETF criou um grupo de trabalho para o HTML no ano seguinte, e publicou o HTML 2.0 em 1995. Desde 1996, as especificações HTML vêm sendo mantidas, com o auxílio de fabricantes de software, pelo World Wide Web Consortium (W3C). Apesar disso, em 2000 a linguagem tornou-se também uma norma internacional (ISO/IEC 15445:2000). A recomendação HTML 4.01 foi publicada no final de 1999 pelo W3C,. Uma errata ainda foi lançada em 2001.

Estrutura básica do HTML:

```
<html>  
<head><<head>  
<body><body>  
</html>
```



Fig2- logo HTML

1.5 CSS

CSS (Cascading Style Sheets) que em português significam folhas de estilo em cascadas, é um mecanismo para adicionar estilo (cores, fontes, espaçamentos, tamanhos textos posicionamento etc.) a um documento web. CSS cuida da estilização da página. Tim Berners-Lee e a w3C lançaram a primeira versão de CSS1-O CSS1.

1.5.1. CSS

O código CSS pode ser aplicado diretamente nas tags ou ficar contido dentro das tags <style>. Também é possível, em vez de colocar a formatação dentro do documento, criar um link para um arquivo CSS que contem os estilos. Assim, quando se quiser alterar a aparência dos documentos vinculados a este arquivo CSS, basta modifica-lo.

Com a variação de atualização dos navegadores, o suporte ao CSS pode variar. A interpretação dos navegadores pode ser avaliada com o teste Acid2, que se tornou uma forma de revelar quão eficiente é o suporte de CSS, fazendo com que a nova Versão em desenvolvimento do Firefox seja totalmente compatível a ele, assim como o Opera já é.

O Doctype informado, ou a ausência dele, determina o quirks mode ou o strict mode, modificando o modo como o CSS é interpretado e a página desenhada.

1.5.2. Aplicação do Selectores em CSS

Em CSS, seletores são usados para declarar a quais elementos de marcação um estilo se aplica, uma espécie de expressão correspondente. Os seletores podem ser aplicados a todos os elementos de um tipo específico, ou apenas aqueles elementos que correspondam a um determinado atributo; elementos podem ser combinados, dependendo de como eles são colocados em relação uns aos outros no código de marcação, ou como eles estão aninhados dentro do objeto de documento modelo.

Pseudoclasse é outra forma de especificação usada em CSS para identificar os elementos de marcação, e, em alguns casos, ações específicas de usuário para o qual um bloco de declaração especial se aplica. Um exemplo frequentemente utilizado é o : hover

psedoclasse que se aplica um estilo apenas quando o usuário aponta para o elemento visível, normalmente mantendo o cursor do mouse sobre ele é anexado a um seletor como em a:hover ou #elementid:hover. Outras psedoclasses e pseudoelementos são, por exemplo, :first-line, :visited ou :before. Uma pseudoclasses especial é :lang(c), “c”.

Uma psedoclasse seleciona elementos inteiros, tais como :link ou :visited, considerando que um pseudoelemento faz uma seleção que pode ser constituída por elementos parciais, tais como :first-line ou :first-letter.

Seletores podem ser combinados de outras formas também, especialmente em CSS 2.1 alcançar uma maior especificidade e flexibilidade.

Usa-se muito a seguinte frase que resume muito bem a função de cada linguagem. HTML para estruturar e CSS para apresentar.

1.5.3. Sintaxe do CSS

CSS tem uma sintaxe simples, utiliza uma série de palavras em inglês para especificar os nomes de diferentes propriedades de estilo de uma página.

Uma instrução CSS consiste em um seletor e um bloco de declaração. Cada declaração contém uma propriedade e um valor, separados por dois pontos (:). Cada declaração é separada por ponto e vírgula (;).

Estrutura básica do CSS:

```
Selector{  
Propriedade:Valor;}
```



Fig3-logo CSS

1.5.4. JavaScript

JavaScript, frequentemente abreviado como JS, é uma linguagem de programação interpretada de alto nível, caracterizada também, como dinâmica, fracamente tipada prototype-based e multi-paradigma. Juntamente com HTML e CSS, o Javascript é uma das três principais tecnologias da World Wide Web. JavaScript permite páginas da Web interativas e, portanto, é uma parte essencial dos aplicativos da Web. A grande maioria dos sites usa, e todos os principais navegadores têm um mecanismo JavaScript dedicado para executá-lo.

JavaScript é a linguagem de programação mais popular no desenvolvimento Web. Suportada por todos os navegadores, a linguagem é responsável por praticamente qualquer tipo de dinamismo que queremos em nossas páginas. Ela foi inventada por Brindam Eich co – fundador do projeto Mozilla da fundação e da corporação Mozilla

Usando todo o poder que ela tem para oferecer, podemos chegar a resultado impressionantes. Excelentes exemplos disso são aplicações web, complexas, como Gmail, GoogleMaps é GoogleDocs.

O termo JavaScript não tem nada a ver com a linguagem de programação Java, esta é uma grande confusão feita pela comunidade de desenvolvimento web.

É atualmente a principal linguagem para programação client-side em navegadores web, ou seja ela é executada no computador do usuário. É também bastante utilizada do lado do servidor através de ambiente como o node.js.

1.5.4.1 Utilização do JavaScript

JavaScript é usada para adicionar interações ao seu site por exemplo: jogos, ações que ocorrem quando botões são formulários, efeitos dinâmicos de estilo, animações e muito mais. É uma linguagem de programação que pode ser aplicada em um documento HTML e usada para criar interações dinâmicas em sites.

Como uma linguagem multi-paradigma, o JavaScript suporta estilos de programação orientados a eventos, funcionais e imperativos (incluindo orientado a objetos e prototype based), apresentando recursos como fechamento (closures) e funções de alta ordem comumente indisponíveis em linguagens populares como Java e C++. Possui APIs para

trabalhar com texto, matrizes, datas, expressões regulares e o DOM, mas a linguagem em si não inclui nenhuma E/S, como instalações de rede, armazenamento ou contando com isso no ambiente host em que está embutido.

Foi originalmente implementada como parte dos navegadores web para que Scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido, porém os mecanismos.

Os termos Vanilla JavaScript e Vanilla JS se referem ao JavaScript não estendido por qualquer estrutura ou biblioteca adicional. Scripts escritos em Vanilla JS são códigos JavaScript simples.

Estrutura básica do JavaScript:

```
<script Language="javascript">  
  <!-- Local do Script-->  
</script>
```



Fig3-logo JavaScript

1.5.4.2. O uso do JavaScript em HTML

O uso primário de JavaScript é escrever funções que são embarcadas ou incluídas em páginas HTML e que interagem com o Modelo de Objeto de Documentos (DOM) da página.

Alguns exemplos deste uso são:

- ❖ Abrir uma nova janela (dialog) com um controle programático sobre seu tamanho, posição e tributos;

- ❖ Validar valores de um formulário para garantir que são aceitáveis antes de serem enviados ao servidor;
- ❖ Mudar imagens á medida que o mouse se movimenta sob elas.

Por ser o JavaScript a única linguagem que a maioria dos navegadores populares suportam, tem se tornado uma linguagem alvo para muitos Frameworks em outras linguagens, a respeito do fato de não ter sido planejado para tal. Não obstante as limitações de desempenho inerente á sua natureza dinâmica, a crescente velocidade das implementações JavaScript tem feito da mesma uma prática linguagem intermediária.

1.6. Principais Linguagens Beck-End

PHP(um acrônimo recursitivo para “PHP:Hypertext Pre-processor”, originalmente personal Home page) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wid Web. Figura entre as primeiras linguagens possíveis de inserção em documentos HTML, dispensado em muitos casos o uso de arquivos externos para eventuais processamentos de dados. O código é interpretado no lado do servidor pelo módulo PHP, que também gera página Web a ser visualizada no lado do cliente.

A linguagem evoluiu, passou a oferecer funcionalidades em linha de comando, e além disso, ganhou características adicionais, que possibilitaram usos adicionais do PHP, não relacionados a web sites. É possível instalar o PHP na maioria dos sistemas operacionais, gratuitamente. Concorrente direito da tecnologia ASP pertencente á Microsoft, o PHP é utilizado em aplicações como o MediaWik, Facebook, Drupal, Joomla, WordPress, Magento e o Oscommerce.

1.6.1.História sobre o PHP

O PHP foi criado por Rasmus Lerdorf em 1995, tem a produção de sua implementação principal, referencia formal da linguagem, mantida por uma organização chamada The PHP Group. O PHP é software livre, licenciado sob a PHP Lincense, uma licença incompatível com a GNU General Public License (GPL) devido a restrições no uso do termo PHP.

A linguagem foi criada em 1994 e o código fonte do PHP só foi criado em 1995, como um pacote de programas CGI criados por Rasmus Lerdorf, com o nome personal Home Page Tools, para substituir um conjunto de scripts perl que ele usava no desenvolvimento de sua página pessoal. Em 1997 foi lançado o novo pacote da linguagem com o nome de PHP/FI, trazendo a ferramenta Forms interpreter, un interpretador de comandos SQL. Mais tarde, Zeev Suraski desenvolveu o analisador do PHP 3 que contava com o primeiro recuso de orientação a objetos, que dava poder de alcançar alguns pacotes, tinha herança e dava aos desenvolvedores somente a possibilidade de implementar a propriedade e métodos.

Trata-se de uma linguagem extremamente modularizada, o que a torna ideal para instalação e uso em servidores Web. Diversos módulos são criados no repositório de extensões PECL (PHP Extension Community Library) e alguns destes módulos são introduzidos como padrão em novas versões da linguagem. É muito parecida, em tipos de dados, sintaxe e mesmo funções, com linguagem C e com a C++. Pode ser, dependendo da configuração do servidor, embarcada no código HTML. Existem versões do PHP disponíveis para seguintes sistemas operacionais: Windows, Linux, FreeBSD, Mac OS, OS/OS2, AS/400, Novell Netware, RISC OS, AIX, IRIX e Solaris.

Construir uma página dinâmica baseada em bases de dados é simples com PHP, (em parte, vale lembrar), este provê suporte a um grande número de bases de dados: Oracle, Sybase, PostgreSQL, InterBase, MySQL, SQLite, MSSQL, Firebird, etc., podendo abstrair o banco com a biblioteca ADODB, entre outras. A Wikipedia funciona sobre um software inteiramente escrito em PHP, usando bases de dados MySQL: o MediaWiki.

1.6.2. Alguns Suportes do PHP

PHP têm suportes aos protocolos: IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC, SOAP. É possível abrir sockets e interagir com outros protocolos. E as bibliotecas de terceiros expandem ainda mais estas funcionalidades. Existem iniciativas para utilizar o PHP como linguagem de programação de sistemas fixos. A mais notável é a PHP-GTK. Trata-se de um conjunto do PHP com a biblioteca GTK, portada do C++, fazendo assim softwares interoperacionais entre windows e Linux. Na prática, essa extensão tem sido muito pouco utilizada para projetos reais. O acrônimo recursivo PHP representam um elefante, que é conhecido como a mascote da linguagem

1.6.3. Funcionamento do PHP no servidor

Ao usar PHP, podemos escolher diversos servidores compatíveis. O mais famoso de todos é o Apache, que provavelmente você vai encontrar em muitas hospedagens no dia a dia. Mas uma novidade das últimas versões do PHP (5.4+) é que ele já vem com um **servidor embutido** simples que dispensa a instalação de um servidor adicional. É ideal para testes e para usar em desenvolvimento.

Estrutura básica do PHP:

```
<?Php
```

Comandos

```
?>
```



Fig4-logo php

2. Base de dados

Banco de dados ou bases de dados: são conjuntos de arquivos relacionados entre si com registros sobre pessoas, lugares ou coisas. São coleções organizadas de dados que se relacionam de forma a criar algum sentido (informação) e dar mais eficiência durante uma pesquisa ou estudo. São de vital importância para empresas e há duas décadas se tornaram a principal peça dos sistemas de informação. Normalmente existem por vários anos sem alterações em sua estrutura.

Um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico, ou seja um assunto.

Segundo Korth, um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico, ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados.

Podemos exemplificar situações clássicas como uma lista telefônica, um catálogo de CDs ou um sistema de controle de RH de uma empresa.

Já um sistema de gerenciamento de banco de dados (SGBD) é um software que possui recursos capazes de manipular as informações do banco de dados e interagir com o usuário. Exemplos de SGBDs são: Oracle, SQL Server, DB2, PostgreSQL, MySQL ou próprio Access ou paradox, entre outros.

Por último, temos que conceituar um sistema de banco de dados com o conjunto de quatro componentes básicos: dados, hardware, software e usuários. Date conceituou que sistema de banco de dados pode ser considerado como uma sala de arquivos eletrônica.

2.1. Exemplos de bases de dados

- ❖ Uma lista Telefônica.
- ❖ Um catálogo de CD's e DVD's.
- ❖ Um sistema de controle de RH de uma empresa.
- ❖ Lista dos clientes de um Banco

2.2. Classificação das bases de dados

As bases de dados classificam-se em:

- a)** Quanto ao Modelo de Dados
- b)** Quanto ao Número de Usuários
- c)** Quanto á Localização

2.3. Aplicações de base de dados

Sistemas gerenciadas de Banco de dados são usadas em muitas aplicações, enquanto atravessando virtualmente a gama inteira de software de computador. Os sistemas gerenciadores de banco de dados são o método preferido de armazenamento/recuperação de dados/informações multiusuárias grandes onde a coordenação entre muitos é necessária. Até mesmo usuários individuais os acham conveniente, entretanto, muitos programas de correio eletrônico e organizadores pessoais estão baseados em tecnologias de banco de dados standard.

2.4. Segurança em bases de dados

Os bancos de dados são utilizados para armazenar diversos tipos de informações, desde dados sobre uma conta de e-mail até dados importantes da Recita Federal. A segurança do banco de dados herda as mesmas dificuldades que a segurança da informação enfrenta, que é garantir a integridade, a responsabilidade e a confidencialidade. Um sistema gerenciador de banco de dados deve fornecer mecanismo que auxiliem nesta tarefa.

Uma forma comum de ataque á segurança do banco de dados é injeção de SQL, em banco de dados que façam uso desta linguagem, mas bancos de dados no SQL também podem ser vítimas. Para evitar estes ataques, o desenvolvedor de aplicações deve garantir que nenhuma entrada possa alterar a estrutura da consulta enviada ao sistema.

Os bancos de dados SQL implementam mecanismos que restringem ou permitem acessos aos dados de acordo com papéis ou roles fornecidos pelo administrador. O comando GRANT concede privilégios específicos para um objeto (tabela, visão, banco de dados,

função, linguagem procedural, esquema ou espaços de tabelas) para um ou mais usuários ou grupos de usuários.

2.5. Sistemas Gerenciadores de bases de dados (SGBD)

Um sistema de gerenciamento de banco de dados (SGBD) – do inglês Data Base Management System (DBMS) – é o conjunto de softwares responsáveis pelo gerenciamento de um banco de dados. Seu principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a persistência, a manipulação e a organização dos dados. O SGBD disponibiliza uma interface para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados. Em bancos de dados relacionais a interface é constituída pelas APIs (Application programming Interface) ou drivers do SGBD, que executam comandos na linguagem SQL (Structured Query Language).

Todas as organizações, por menor que sejam, possuem quantidades cada vez maiores de dados e informações a armazenar. Todavia, a manipulação dessas informações tornou-se impossível de ser realizada manualmente (via papéis, principalmente), pois sua utilização, além de demorada (devido a catalogação dos dados), é impossível de erros, principalmente ocasionados pelo desgaste do operador em conseguir resgatar informações requisitadas.

Dicionário de dados (guarda definições de elementos de dados e respectivas características que descreva os dados, quem os acede, etc.)

Exemplos de SGBD's

- ❖ PostgreSQL
- ❖ MySQL
- ❖ MongoDB
- ❖ Oracle
- ❖ SQL-Server

❖ Mongo DB

2.6. O Modelo Relacional

O modelo relacional é um modelo de dados representativo (ou de implementação), adequado a ser o modelo subjacente de um Sistema Gerenciador de Banco de Dados (SGBD), que se baseia no princípio de que todos os dados estão armazenados em tabelas (ou matematicamente falando, relações). Toda sua definição é teórica e baseada na lógica de predicados e na teoria dos conjuntos.

O conceito foi criado por Edgar Frank Codd em 1970, sendo descrito no artigo “Relational Model of Data for Large Shared Data Banks”. Na verdade, o modelo foi o primeiro modelo relacional, pois o primeiro modelo de dados descrito teoricamente, os bancos de dados já existentes passaram então a ser conhecidos como (modelo hierárquico, modelo em rede ou codasyl e modelo de listas invertidas).

O modelo relacional para gerência de bases de dados (SGBD) é um modelo de dados baseado em lógica e na teoria de conjuntos.

Em definição simplificada, o modelo baseia-se em dois conceitos: conceito de entidade e relação.

2.3. Entidades ou tabelas

Uma entidade ou tabela é um elemento caracterizado pelos dados que são recolhidos na sua identificação vulgarmente designado por tabela. Na construção da tabela identificam-se os dados da entidade. A atribuição de valores a uma entidade constrói um registro da tabela.

A relação determina o modo como cada registro de cada tabela se associa a registro de outras tabelas.

Historicamente ele é o sucessor do modelo hierárquico e do modelo em rede. Estas arquiteturas antigas são até hoje utilizadas em alguns data centres com alto volume de dados, onde a migração é inviável pelo custo que ela demandaria; existem ainda os

novos modelos baseados em orientações ao objeto, que na maior parte das vezes são encontrados como kits em linguagem formal.

Nos modelos de bases de dados relacionais, a tabela é um conjunto de dados em número infinito de colunas e números ilimitado de linhas (tuplas).

As colunas são tipicamente consideradas os campos da tabela, e caracterizam os tipos de dados que deverão constar na tabela (numéricos, falta-numéricos, datas, coordenadas, etc.). O número de linhas pode ser interpretado como o número de combinações de valores dos campos da tabela, e pode conter linhas idênticas, dependendo do objetivo. A forma de referenciar inequivocamente uma única linha é através da utilização de uma chave primária.

Para além do tipo de dados inerente a todas as colunas de uma tabela, algumas posem ter associados restrições: a unidade (SQL: UNIQUE), proibição de valores NULL (SQL NOT NULL), delimitação de valores, etc. Estas restrições impedem que sejam inseridos valores desejados que comprometam a validade e integridade dos dados.

2.3.2 Classificação dos atributos de uma entidade

Uma chave em bases de dados é uma coluna que permite, para cada linha identificar essa linha (caso seja a chave primária), ou relacionar com outra tabela no caso da chave estrangeira.

Chave primária

Atributo da tabela que permite identificar univocamente os registros.

Primary keys ou PK, sob o ponto de vista de um banco de dados relacional, referem-se ao conjuntos de um ou mais campos, cujos valores, considerados a combinação de valores em caso de mais de uma chave primária, nunca se repetem na mesma tabela e, desta forma podem ser usadas como um índice de referências para criar relacionamentos com as demais tabela do banco de dados (daí vem o nome banco de dados relacional). Portanto, uma chave primária nunca pode ter valor nulo nem repetição.

Simplificando, quando a chave primária é simples, ou seja é formada por um único campo da tabela esse campo não pode ter dois ou mais registros de mesmo valor e também não pode conter nenhum registro de mesmo valor e também não pode conter nenhum registro

nulo. A chave primária pelo menos um dos campos da tabela deve ter valor único para cada registro

Chave candidata ou secundária

É uma chave que se candidata a ser primária.

Uma “chave candidata” consiste em um atributo que pode ser aniversário, ou para a feriados, utiliza-se os campos ou grupo de atributos cujo valor identifica unicamente cada dupla em uma relação e para o qual nenhum dos atributos pode ser removido sem destruir a identificação única.

2.3.4. Normalização

Normalização de banco de dados é o processo de depurar as relações dum dado esquema de BD relacional de maneira a eliminar anomalias redundâncias e outros problemas. É o processo que transforma um esquema de dado qualquer em um modelo relacional.

É o conjunto de regras que visa, principalmente, a organização de um projeto de banco de dados para reduzir a redundância de dados, aumentar a integridade de dados e o desempenho. Para normalize dados, deve-se examinar as colunas (atributos) de uma entidade e as relações entre entidades (tabelas), com o objetivo de se evitar anomalias observadas na inclusão, exclusão e alteração de registros.

Atualmente, muitos sistemas de informação ainda não utilizam banco de dados relacionais senão esses chamados de sistema legados. Os dados desses sistemas são armazenados em arquivos de linguagens de terceira geração, com COBOL, ou então, em banco de dados da era pré-relacional.

2.3.5. Formas Normais

Existem pelo menos Seis Formas Normais, denomina-se das Primeira Forma Normal, Segunda Forma etc. Normalmente escrevemos como 1FN, 2FN, 3FN, 3FNBC, 4FN, 5FN.

2.3.6. Primeira Forma Normal

Uma tabela está na 1FN se e somente todos os valores das colunas da tabela forem atómicos (valores indivisíveis). Não há conjuntos de atributos repetidos descrevendo a mesma característica na 1FN.

Assim, podemos dizer que os relacionamentos, como definidos acima, estão necessariamente na 1FN. Uma relação está na 1FN quando os atributos da relação estiveram baseados em um domínio simples, não contendo grupos ou valores repetidos.

Passagem á 1FN

- ❖ Encontre a chave primária da tabela;
- ❖ Fique ciente de quais são as colunas da tabela que apresentam dados repetidos para que sejam removidos;
- ❖ Crie uma tabela para esses dados repetidos, com a chave primária da anterior;
- ❖ Por fim, estabeleça relação entre a nova tabela e a principal.

Outra forma de identificar se a tabela não está na 1FN é verificando se existe tabela aninhadas, ou seja mais de um registro para uma primária.

2.3.7. Segunda Forma Normal

Uma relação (Tabela) está na 2FN se, e somente se, estiver na 1FN e cada atributo não-chave for dependente da chave primária inteira, isto é, todos os atributos não-chave dependem funcionalmente da totalidade de chave.

No caso de tabelas com chave primária composta, se um atributo depende de uma parte da chave primária, então esse atributo deve ser colocado em outra tabela.

Passagem á 2FN

- ❖ Geração de novas tabelas com Dfs (Dependências Funcionais) completas.

- ❖ Análise de dependências funcionais.
- ❖ Funcionais indiretas.

Na segunda forma normal (2FN) são eliminados todas as dependências de atividades não-chave para com a chave primária.

2.3.8. Terceira forma Normal

Uma relação R está na 3FN se ela estiver na 2FN e cada atributo não-chave de R não possui dependência transitiva, para cada chave candidata de R. Todos os atributos dessa tabela devem ser dependentes funcionalmente um dos outros e ao mesmo tempo devem ser dependentes exclusivamente da chave primária da tabela.

Exemplo ilustrativo

A tabela a seguir não está na Terceira Forma Normal porque a coluna Total é dependente, ou é resultado, da multiplicação das colunas preço e quantidade, ou seja, a coluna total tem dependência transitiva de colunas que não fazem parte da chave primária, ou mesmo candidata da tabela. Para que essa tabela passe a Terceira FN o campo Total deverá ser eliminado, a fim de que nenhuma coluna tenha dependência de qualquer outra que não seja exclusivamente chave.

Terceira Forma Normal

Aqui, após o atributo/coluna Total ser excluído da tabela, ela já está na terceira forma normal. Esse atributo pode ser movido para outra tabela referenciando a antiga.

Passagem á 3FN

- ❖ Para estar na 3FN precisa estar na 2FN;
- ❖ Geração de novas com DF diretas;
- ❖ Análise de dependência funcionais entre atributos não chave;
- ❖ Verificar a dependência exclusiva da chave primária;
- ❖ Entidades na 3FN também não podem conter atributos que sejam resultados de algum cálculo de outro atributo.

2.3.9. Quarta Forma Normal

Uma tabela está na 4FN, se e somente se, estiver na 3fn e não existirem dependência multivaloradas.

Exemplo (base de dados sobre livros)

Relação não normalizada: Livros (nrol,(autor) titulo, (assunto), editora ,cid_edit, ano public).

Na 3FN, a base de dados ainda apresenta os seguintes problemas:

Redundância para representar todas as informações;

Representação não-uniforme (repete alguns elementos ou posições nulas).

2.4. Structured Query Language (SQL)

SQL é a sigla para Structured Query Language (Linguagem Estruturada para consulta).

Linguagem de consultas estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Muitas das características originais do SQL foram inspirada na álgebra relacional. Consulta é uma pergunta que eu faço no banco de dados.

A linguagem SQL é usada para comunicar com base de dados. É uma linguagem padrão para base de dados, declarativa, ao contrário das linguagens habituais que são do tipo procedimental. O SQL é uma linguagem de programação que serve para criar, atualizar e responder as questões da informação armazenada num sistema gerenciador de base de dados (SGBD).

A linguagem é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Ela é uma linguagem declarativa em oposição a outras linguagens procedurais. Isto reduz o ciclo de aprendizado aqueles que se iniciam na linguagem.

Sublinguagens do SQL

A linguagem SQL é constituída por sublinguagens de acordo com as operações que queremos efetuar sobre um banco de dados, tais como:

DML – Linguagem de Manipulação de Dados

O primeiro grupo é a DML (Data Manipulation Language – Linguagem de dados). DML é uma sublinguagem da linguagem SQL que é utilizado para realizar inclusões, consultas, alterações e exclusões de diversas tabelas ao mesmo tempo. Os comandos que realizam respetivamente as funções acima referidas são Insert, Select, Update e Delete.

DDL – Linguagem de Definição de Dados

O segundo grupo é a DDL (Data Definition Language – Linguagem de Definição de dados). É a área da SQL que se encarrega da definição dos bancos de Dados. Uma DDL permite a criação dos bancos de dados, criação de tabelas novas e alteração da tabela, dos campos e seus respectivos tipos de Dados. A maioria dos bancos de dados de SQL comerciais tem extensões proprietárias no DDL.

Os comandos básicos da DDL são poucos:

CREATE: cria um objeto, por exemplo uma base de dados ou uma tabela.

DROP: apaga um objeto do banco de dados. E apaga toda a estrutura e o conjunto de valores lá existentes.

ALTER: permite ao usuário alterar um objeto ou a estrutura de uma tabela, por exemplo adicionando uma coluna a uma tabela existente, alterar dados de uma tabela como nome, idade, sexo, nacionalidade etc.

Outros comandos DDL:

- ❖ CREATE TABLE
- ❖ CREATE INDEX
- ❖ CREATE VIEW
- ❖ CREATE DROP

DCL – Linguagem de controlo de Dados

O terceiro grupo é o DCL (Data Control Language - Linguagem de controle de Dados). DCL controla os aspectos de autorização de dados e licenças de usuários para controlar quem tem acesso para ver ou manipular dados dentro do banco de dados.

Duas palavras-chaves da DCL:

- ❖ GRANT - autoriza ao usuário executar ou setar operações.
- ❖ REVOKE – remove ou restringe a capacidade de um usuário de executar operações.

DTL – Linguagem de Transação de Dados

- ❖ BEGIN WORK (ou START TRANSACTION, dependendo do dialeto SQL) pode ser usado para marcar o começo de uma transação de banco de dados que pode ser completada ou não.
- ❖ COMMIT finaliza uma transação dentro de um sistema de gerenciamento de banco de dados.
- ❖ ROLLBACK faz com que as mudanças nos dados existentes desde o último COMMIT ou ROLLBACK sejam descartadas.

COMMIT e ROLLBACK interagem com áreas de controlo como transação e locação. Ambos terminaram qualquer transação aberta e liberam qualquer cadeado ligado a dados. Na ausência de um BEGIN WORK ou uma declaração semelhante, a semântica de SQL é dependente da implementação.

DQL – Linguagem de consulta de dados

Embora tenha apenas um comando, a DQL é a parte da SQL mais utilizada. O comando SELECT permite ao usuário especificar consulta (“query”) como uma descrição do resultado desejado. Esse comando é composto de várias cláusulas e opções, possibilitando elaborar consultas das mais simples as mais elaboradas.

2.5 Aplicações de páginas web

2.5.1.Xamp

É um pacote com os principais servidores de código aberto do mercado, incluindo FTP Banco de Dados MSQSL e Apache Com suporte as linguagens PHP.



Fig6- logo xampp

2.5.2. Wampserver

É o servidor de aplicação onde desenvolvida a base de dados. É um software que efetua a instalação automática de um conjunto de softwares no computador, de modo a facilitar a configuração de um software interpretador de scripts local e um banco de dados no sistema Windows. Conjunto desenvolvido pela PHP Team sob licença GNU General Public License.

É uma ferramenta que auxilia no desenvolvimento, permitindo aos programadores de websites testarem o trabalho em seus próprios computadores, sem necessitar acesso à Internet.



Fig7-logoWampserver

2.5.3. MySql Workbench

O MySQL Workbench é uma ferramenta de design de banco de dados visual que integra desenvolvimento, administração, design de banco de dados, criação e manutenção de SQL em um único ambiente de desenvolvimento integrado para o sistema de banco de dados MySQL.



Fig8- logo Msqlworkbench

2.5.4. Font Awesome

É um conjunto ferramentas de fontes e ícones com base Css e Less foi feita por Dave Gandy para uso com twitter Bootstrap e mais tarde foi incorporado no Bootstrap.

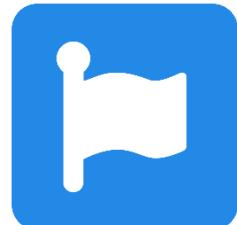


Fig9-logo FontAwesome

2.5.4. Visual Studio Code

É um editor de texto de código – fonte desenvolvido pela Microsoft para Windows, Linux e MacOs, ele inclui Suport para depuração controle Git incorporado realce de sintaxe complementação inteligente de código snippets e refatoração de códigos.

O Visual Studio Code é baseado no Electron, uma *framework* que é usada para desenvolver aplicativos Node.js para o desktop rodando no motor de layout Blink.

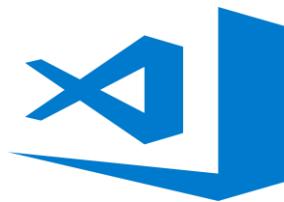


Fig9-logo Visual Studio code

2.6. Navegadores web

Navegador é uma ferramenta para visualização do conteúdo web. Cada conteúdo está indicado por um endereço próprio ao qual o navegador se refere quando necessita de determina informação.

Os navegadores podem ser classificadas quanto ao modo de visualização em:

- ❖ Modo texto
- ❖ Modo gráfico

Modo texto: São navegadores que mostram apenas o texto do conteúdo requerido. São ideias para computadores de baixa capacidade já que são rápidos e leves.

Modo Gráfico: São navegadores com visualização mas agradável já que suporta visualização de imagens e outros recursos multimídia.

2.6.1. Google Chrome

É um navegador de internet desenvolvido pela companhia Google com visual mínima lista. Foi lançada pela primeira vez em setembro de 2008 para o Microsoft Windows e mais tarde portado para o Linux, Mac, IOS e Android. O Google Chrome também é o principal componente do Chrome OS onde funciona como uma plataforma para

executar aplicativos da web. E sua estrutura de desenvolvimento de aplicações_framework.

O navegador atualmente está disponível, em mais de 51 idiomas, para as plataformas Windows, Mac OS X, Android, iOS, Ubuntu, Debian, Fedora e openSUSE.

Atualmente, o Chrome é o navegador mais usado no mundo, com 49,18% dos usuários de Desktop, contra 22,62% do Internet Explorer e 19,25% do Mozilla Firefox, segundo a StatCounter. Além de desenvolver o Google Chrome, o Google ainda patrocina o Mozilla Firefox, um navegador desenvolvido pela Fundação Mozilla. Até julho de 2018, o navegador Google Chrome permanece na liderança como o navegador mais usado no mundo.



Fig11- logoChrome

Existem outros navegadores web como: o Mozilla Opera Mini etc.

CAPÍTULO 2: Programação em camadas

Programação ou desenvolvimento em camadas: é a divisão da funcionalidade, componentes e o código para uma aplicação, seja para web ou para desktop.

Normalmente usado para o desenvolvimento de interfaces de usuário que divide uma aplicação em partes (camadas/componentes) interconectadas. Isto é feito para separar representações de informação internas dos modos como a informação é apresentada e aceita pelo usuário, levando ao desenvolvimento paralelo de maneira eficiente.

2.1. Modelo em quatro Camadas

Em camadas distintas podem ser delineadas da seguinte forma:

Camada de armazenamento dos dados (DS - Data Store): Camada onde os dados são armazenados. Pode ser um banco de dados relacional (RDBMS), um arquivo XML, texto ou qualquer outro sistema de armazenamento de informações;

Camada de acesso aos dados (DAL-Data Access Layer): Camada onde reside o código que trata da obtenção, tratamento e retorno dos dados salvos na camada de armazenamento de dados. É composta pelo repositório das informações e as classes que as manipulam.

Camada da lógica de negócio (BLL-BusinessLogic Layer): Camada onde está o código que obtém os dados retornados pela **DAL** e os repassa ao cliente de uma forma mais intuitiva, efetuando a validação lógica conforme as regras do negócio e verificando se os dados estão no formato adequado. Também chamada de lógica empresarial, regras_de negócio ou funcionalidade. É nela que ficam as funções e regras de todo o negócio;

Camada de apresentação (UI - User Interface) - Camada onde reside o código que define o que será efetivamente apresentado ao usuário, efetuando a formatação dos dados, navegação e ajuda ao usuário final. Esta camada interage diretamente com o usuário, é através dela que são feitas as requisições como consultas;

Camada de Conteúdo: para essa camada o objetivo é fazer a demarcação dos blocos de conteúdo propostos pelo planejamento e pelo layout do website. Para isso a principal linguagem utilizada é o HTML, que está em sua versão 5.

Camada de Formatação: nesta camada o objetivo é formatar o conteúdo do modo como ele foi formatado na criação do layout. Para isso são utilizadas imagens e a linguagem CSS, que está em sua versão 3.

Camada de Comportamento

Nesta camada são incluídos movimentos e recursos de interação do usuário com o conteúdo, para melhorar sua experiência com o site. Para isso é utilizada a linguagem JAVASCRIPT e alguns framework JQUERY e seus plugins.

Camada de Automação: esta camada envolve a automação do gerenciamento das informações do website, tanto por parte de usuários administradores quanto pelos visitantes do site. Para isso são utilizadas linguagens avançadas de programação, como o C#, Java ou PHP e banco de dados, como MsSQL ou MySQL.

2.2. Modelo em três camadas

No modelo 3 camadas, a lógica de apresentação esta separada em sua própria camada lógica e física. A separação em camadas lógicas torna os sistemas mais flexíveis, permitindo que as partes possam ser alteradas de forma independente.

As funcionalidades da camada de negócio podem ser divididas em classes e essas classes podem ser agrupadas em pacotes ou componentes, reduzindo as dependências entre as classes e pacotes, podem ser reutilizadas por diferentes partes do aplicativo e até por aplicativos diferentes. O modelo de 3 camadas tornou-se a arquitetura padrão para sistemas corporativos com base na Web.

As três partes de um ambiente modelo três camadas são: camada de apresentação, camada de negócio e camada de dados.

Camada de apresentação: é a chamada GUI (Graphical User Interface), ou simplesmente interface. Esta camada interage diretamente com o usuário, é através dela que são feitas as requisições como consultas.

Camada de negócio: também chamada de lógica empresarial, regras de negócio ou funcionalidade. É nela que ficam as funções e regras de todo o negócio. Não existe uma interface para o usuário e seus dados são voláteis, ou seja, para que algum dado seja mantido deve ser utilizada a camada de dados.

Camada de Dados: é composta pelo repositório das informações e as classes que as manipulam. Esta camada recebe as requisições da camada de negócios e seus métodos executam essas requisições em um banco de dados. Uma alteração no banco de dados alteraria apenas as classes da camada de dados, mas o restante da arquitetura não seria afetado por essa alteração.

Na arquitetura de distribuição cliente-servidor, é preciso definir que tipo de cliente será utilizado, se um cliente inchado ou magro.

O cliente inchado (*fat client*) é aquele que congrega as funcionalidades de interface com usuário com a lógica da aplicação. Neste caso, o servidor é essencialmente um provedor e armazenador de dados.

O cliente magro (*thin client*) está restrito essencialmente às funcionalidades de interface com usuário. Neste caso, a lógica da aplicação é alocada ao servidor da aplicação.

Em uma arquitetura de três camadas, a aplicação é organizada em três unidades de distribuição:

1. Camada de interface com usuário;
2. Camada de negócios, com a lógica da aplicação;
3. Camada de dados, com funcionalidade de armazenamento e recuperação.

Aplicações monolíticas (uma camada)

Nos tempos antigos do reinado do grande porte e do computador pessoal independente, um aplicativo era desenvolvido para ser usado em uma única máquina (standalone). Geralmente esse aplicativo continha todas as funcionalidades em um único módulo gerado por uma grande quantidade de linhas de código e de difícil manutenção. A entrada do usuário, verificação, lógica de negócio e acesso a banco de dados estavam todos presentes em um mesmo lugar.

Aplicações em duas camadas

A necessidade de compartilhar a lógica de acesso a dados entre vários usuários simultâneos fez surgir as aplicações em duas camadas. Nesta estrutura, a base de dados foi colocada em uma máquina específica, separada das máquinas que executavam as aplicações. Nessa abordagem, temos aplicativos instalados em estações clientes contendo toda a lógica da aplicação (clientes ricos ou gordos). Um grande problema neste modelo é o gerenciamento de versões pois, para cada alteração, os aplicativos precisam ser atualizados em todas as máquinas clientes.

Aplicações em três camadas

Com o advento da Internet, houve um movimento para separar a lógica de negócio da interface com o usuário. A ideia é que os usuários da WEB possam a cessar as mesmas aplicações sem ter que instalar estas aplicações em suas máquinas locais. Como a lógica do aplicativo, inicialmente contida no cliente rico, não mais reside na máquina do usuário, este tipo de cliente passou a ser chamado de cliente pobre ou magro (Thin Client). Neste modelo o aplicativo é movido para o servidor e um navegador web é usado como um cliente magro. O aplicativo é executado em servidores web com os quais o navegador web se comunica e gera o código HTML para ser exibido no cliente.

2.3. MVC – Model, View, Controller

É uma arquitetura de projeto onde seu objetivo é separar seu código em três camadas fazendo com que cada área só trabalhe com itens que competem à elas ou seja, cada um só faz o que foi desenvolvido para fazer. Com o MVC você facilmente transforma seu código de modo à ficar muito mais legível.

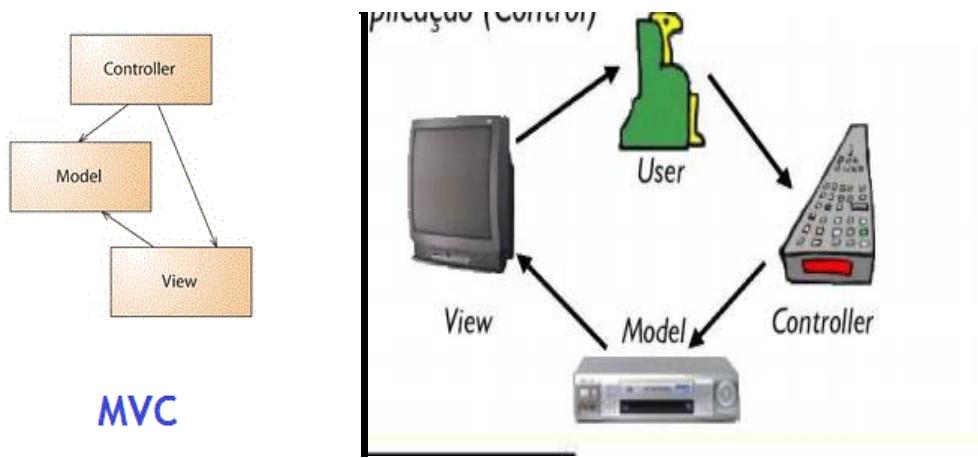
Foca-se no reuso de código e a separação de conceitos em três camadas interconectadas, onde a apresentação dos dados e interação dos usuários (*front-end*) são separados dos métodos que interagem com o banco de dados (*back-end*). Para utilizá-lo você tem que ter em mente que haverá uma separação em seu código, as regras de negócio ficarão separadas da lógica e da interface do usuário.

O model, ou modelo, no padrão MVC serve para armazenar e persistir os dados. Toda comunicação com o banco de dados. Os comandos crud (inserir, alterar, remover, buscar) serão feitas pelas classes deste tipo.

É utilizado para armazenar informações, trabalhando como um espelho da tabela do banco de dados. Como trabalhamos com objetos, os dados serão persistidos como objetos.

O view, ou visão, no padrão MVC servirá APENAS para exibir as informações enviadas pelo controller, aqui não existirá nenhuma lógica ou regra de negócio, apenas a interface do usuário.

O controle faz exatamente o que o nome diz: controla. Ele é o responsável por fazer o intermédio entre o modelo e a visão. É o responsável também por toda lógica do sistema. Retornando somente os itens necessários para a comunicação entre o modelo e a visão. Entre o usuário e a aplicação.



Exemplo MVC

2.3.1. VANTAGENS

1. Facilita a manutenção
2. Facilita o desenvolvimento por horas multidisciplinares

A utilização do padrão MVC trás como benefício isolar as regras de negócios da lógica de apresentação, a interface com o usuário. Isto possibilita a existência de várias interfaces com o usuário que podem ser modificadas sem que haja a necessidade da alteração das regras de negócios, proporcionando assim muito mais flexibilidade e oportunidades de reuso das classes.

Uma das características de um padrão de projeto é poder aplicá-lo em sistemas distintos. O padrão MVC pode ser utilizado em vários tipos de projetos como, por exemplo, desktop, web e mobile.

2.3.2. POO

Programação orientada a objetos (**POO**, ou **OOP** segundo as suas siglas em inglês) é um paradigma de programação baseado no conceito de "objetos", que podem conter dados na forma de campos, também conhecidos como *atributos*, e códigos, na forma de procedimentos, também conhecidos como métodos. Uma característica de objetos é que um procedimento de objeto pode acessar, e geralmente modificar, os campos de dados do objeto com o qual eles estão associados (objetos possuem uma noção de "this" (este) ou "self" (próprio)).

Em POO, programas de computadores são projetados por meio da composição de objetos que interagem com outros. Há uma diversidade significante de linguagens de POO, mas as mais populares são aquelas baseadas em classes, significando que objetos são instâncias de classes, que, normalmente, também determinam seu tipo.

Muitas das linguagens de programação mais utilizadas (como C ++, Object Pascal, Java, Python, etc.) são linguagens de programação multiparadigmas que suportam programação orientada a objetos em maior ou menor grau, tipicamente em combinação com programação procedural imperativa.

CAPÍTULO 3 – IMPLEMENTAÇÃO

3. Página web para o Centro Médico Sol

3.1. Objeto de estudo

A página web para agendamento de consultas online em clínicas permitirá marcar consultas. Ela começou a ser desenvolvida em Julho de 2020 pela programadora Emília Manuel. Ela pode ser utilizada para Pacientes (Clientes).

3.1.2. Objetivos

A página web para agendamento de consultas online em clínicas tem como objetivo ajudar e facilitar a população Angolana nesses últimos dias na marcação de consultas em qualquer clinica sem sair de casa oferecendo os seus serviços de uma forma rápida e fácil.

3.1.3. Principais problemas encontrados

Devido aos grandes problemas que estão surgindo nesses ultimos dias há muitas dificuldades em fazer qualquer tipo de marcação de consultas presencialmente, e também devido ao grande numero da população infelizmente muitos que se dispõem em sair da sua casa para fazer marcação de consulta muitas vezes não são atendidos no mesmo dia as clinicas precisam de uma solução automatizada que faça agendamento de consultas. Para o efeito, me disponibilizei em desenvolver um site ou aplicação Web com objetivo de atender as necessidades da população e que seja realmente algo util na vida das pessoas

3.1.4. Solução Encontrada

Como Solução criou-se a aplicação web que terá como principal foco fazer agendamento de consultas em clínicas, a fim de facilitar todas as clinicas e todos os pacientes na marcação de consulta de uma maneira segura e facil

3.1.5. Ferramentas usadas para construção da aplicação

Visual Code – utilizei para desenvolvimento de códigos em HTML e CSS.

Google chrome - usei para executar o Aplicativo vez á pós vez, depois de alterações.

Linguagem PHP- utilizei-a para dar dinamismo a página de forma a ter uma interatividade mais ampla com o usuário.

Microsoft SQL Server- usei apenas para criar relação entre as tabelas, e foi aqui onde criei diagrama.

WampServer- utilizei para o desenvolvimento da base de dados e teste os códigos php que regista os dados inseridos nos formulários dentro do projeto.

3.1.6. Requisitos

Os requisitos num sistema, “*são funcionalidades ou características consideradas relevantes na ótica do utilizador*”. Normalmente, representam o comportamento esperado do sistema, que na prática consiste num serviço que deve ser disponibilizado a um utilizador.

3.1.7. Requisitos Funcionais

Requisitos funcionais, “*descrevem o que um sistema faz*”. São os requisitos que inicialmente serão levantados, abrangendo a descrição de processamento a efetuar pelo sistema, entradas (inputs) e saídas (outputs) de informação em papel ou no ecrã que derivam da interação com pessoas e outros sistemas. A seguir, é apresentado uma tabela com os requisitos funcionais do sistema.

Tabela 1. Requisitos Funcionais da página web

Identificador	Requisito Funcional
RF01	Login de Usuário
RF02	Cadastra pacientes
RF03	Cadastra Clínica
RF04	Visualiza Clinica
RF05	Fazer Marcação de consulta
RF07	Aceitar consulta
RF08	Fechar consulta
RF09	Marca falta no Cliente
RF10	Cancelar Consultas

3.1.8. Requisitos Não Funcionais

Requisitos não funcionais “está relacionados com as características qualitativas do sistema, descrevendo a qualidade com que o sistema deverá fornecer os requisitos funcionais”. Abrange medidas de desempenho como, por exemplo, tempos de resposta, volume de dados ou considerações de segurança.

Tabela 2. Requisitos Não Funcionais da Página Web

RF01	O sistema Deverá Funcionar somente como web	Portabilidade
RF02	O sistema não precisa ter acesso a internet para funcionar	Usabilidade
RF03	O Sistema confirma o cadastramento dos usuários de uma forma rápida	Usabilidade

3.2. Arquiteturas

3.3. Descrições das Tabelas

3.9. Descrições dos Formulários

3.10. Modelo entidade de relacionamento