

CENTRO PAULA SOUZA

ETEC EURO ALBINO DE SOUZA

Técnico em Informática

Gabriel Coelho Soares

Hermes Vieira Silva

Luís Fernando Ribeiro

João Paulo Camargo Romani

Michell Elvis Christofferson Geovanne Monteiro da Silva

Marc Alder Delma

Richard Glauber Dias

Valterli J. M. Coutinho

Vinícius da Cruz Becaleti

TechTrain Website

MOGI GUAÇU - SP

2017

Gabriel Coelho Soares

Hermes Vieira Silva

Luís Fernando Ribeiro

João Paulo Camargo Romani

Michell Elvis Christofferson Geovanne Monteiro da Silva

Marc Alder Delma

Richard Glauber Dias

Valterli J. M. Coutinho

Vinícius da Cruz Becaleti

TechTrain Website

Trabalho de Conclusão de Curso
apresentado à Etec Euro Albino de
Souza, do Centro Estadual de Educação
Tecnológica Paula Souza, como requisito
para a obtenção do diploma de Técnico
de Nível Técnico de Informática sob a
orientação do Professor Pedro Ramires
da Silva Amalfi Costa.

MOGI GUAÇU - SP

2017

Gabriel Coelho Soares

Hermes Vieira Silva

Luís Fernando Ribeiro

João Paulo Camargo Romani

Michell Elvis Christofferson Geovanne Monteiro da Silva

Marc Alder Delma

Richard Glauber Dias

Valterli J. M. Coutinho

Vinícius da Cruz Becaleti

TechTrain Website

Aprovada em: _____ / _____ / _____

Conceito: _____

Banca de Validação:

Presidente da Banca

Professor Orientador.....

Professor

Professor

MOGI GUAÇU – SP

2017

Agradecimento

Agradecemos a Deus por estar sempre conosco, ao nossos familiares, amigos e professores que sempre nós motivamos a não desistir e, a Diretoria que disponibilizou as ferramentas para o desenvolvimento do Trabalho, e a todos os professores que nos ensinaram e formaram novos técnico de informática.

Epígrafe

“Acho que a Internet pode ajudar as pessoas a comunicar de uma forma mais eficaz.

Tim Berners-Lee

RESUMO

Esse trabalho teve como objetivo o desenvolvimento de um site de aprendizagem técnica onde o usuário possa ser professor ou aluno. Para ser professor, ele deverá realizar um cadastro indicando um canal em alguma plataforma de vídeos *online*, e passará por aprovação prévia, para que seja permitido que ele crie novos cursos. Para alcançar o objetivo proposto, foram realizados ao longo de vários meses e dias, aulas teóricas e práticas de PTCC, pesquisa de campos e na *internet*, documentos textuais e CorelDraw, e também o desenvolvimento do banco de dados, visando entregar uma proposta com uma problemática e solução para a mesma.

PALAVRAS-CHAVE: Desenvolvimento de Site. Pesquisas. Internet. Vídeos.

ABSTRACT

This work aimed to develop a portal website where the common users may learn a course. The website to present some instructions for anyone users to teach some course and others users will may be watching class and increase your knowledge. A mini tutorial on how to use it. In order to reach the proposed objective, theoretical and practical classes of PTCC, field and internet research, textual documents and CorelDraw were carried out over several months and days, and finally the development of the database, aiming to deliver a proposal with a Problem and solution to it.

KEY WORDS: Website Development. Search. Network. Videos.

LISTA DE FIGURAS

Figura 1 – HTML

Figura 2 – CSS

Figura 3 - PHP

Figura 4 - Notepad++

Figura 5 - Visual Studio Code

Figura 6 - Git

Figura 7 - Github

Figura 8 - Banco de Dados

Figura 9 - UML

Figura 10 - Diagrama de Caso de Uso

Figura 11 - MariaDB

Figura 12 - Tree View of the Framework

Figura 13 - ./app/views/index.php

Figura 14 - index.php codes

Figura 15 - Cadastro

Figura 16 - Estrutura da Página Cadastro

Figura 17 - função postRegister

Figura 18 - função login

Figura 19 - função logout

Figura 20 - Página de cursos

Figura 21 - Página de cada Curso

Figura 22 - cursos.php código

Figura 23 - coursePage.php código

Figura 24 - coursePage.php código parte 2

Figura 25 - CourseController.php códigos

Figura 26 - Página de Vídeo

Figura 27 - videoPage.php códigos

LISTA DE TABELAS

Tabela 1 - Receita

LISTA DE SIGLAS

API - Application Programming Interface - Interface de Programação de Aplicativos.

CERN - European Council for Nuclear Research – Conselho Europeu de Pesquisa Nuclear.

CSS - Cascading Style Sheets - Folhas de Estilo em Cascata.

DNS - Domain Name System - Sistema de Nomes de Domínios.

GPL - General Public License,- Licença Pública Geral.

HTML - *HyperText Markup Language* - Linguagem de Marcação de Hipertexto.

IDE - Integrated Development Environment - Ambiente Integrado ao Desenvolvimento.

OMT - Object Modelling Tool – Ferramenta de modelagem de objeto.

SGBD - Data Base Management System - Sistema de Gerenciamento de Bancos de Dados Relacionais.

STL - Standard Template Library em português - Biblioteca Padrão de Gabaritos.

SGML - Standard Generalized Markup Language - linguagem de marcação. Generalizada

SQL - Structured Query Language - Linguagem de consulta estruturada.

UML - Unified Modeling Language - Linguagem Unificada de Modelagem.

URL- Uniform Resource Locator - Localizador Padrão de Recurso.

SUMARIO

INTRODUÇÃO

Para este trabalho foi escolhido o tema: Desenvolvimento de *Website* que consiste numa rede de aprendizagem onde qualquer um pode ser professor, desde que tenha o desejo de ensinar. Os cursos também terão certificados livres para o aluno que o completar.

O desenvolvimento será feito através dos conhecimentos obtidos nas aulas de informática e com a ajuda de livros e de recursos da *internet*.

O grupo pretende, futuramente, realizar parcerias com empresas de desenvolvimento e com outras empresas que tenha interesse no site para lecionar. É um projeto onde o objetivo é demonstrar e despertar curiosidades do usuário leigo em assuntos técnicos e futuramente ele pode ingressar em um emprego com um preparo melhor.

1. TÉCNOLOGIAS APLICADAS

Para o desenvolvimento de um sistema *Web* é necessário aplicar devidas tecnologias. No decorrer deste capítulo algumas delas serão abordadas. Devido a sua finalidade cada uma delas, elas são classificadas em três categorias: tecnologias voltadas à Programação, tecnologias voltadas ao Banco de Dados e tecnologias diversas.

No caso deste projeto elas serão representadas respectivamente pelo PHP (linguagem de programação), MariaDB (banco de dados), e enfim CorelDraw (editor de imagens).

1.1. LÓGICA DE PROGRAMAÇÃO

Alguns criadores de linguagem de programação, Niklaus Wirth concebeu, com o intuito de encorajar o uso de código estruturado, a linguagem Pascal, Alain Colmerauer e Philippe Roussel, foi outra das linguagens de

programação que apareceu nesta década. Enquadra-se no paradigma de programação em lógica matemática, consistindo numa linguagem puramente lógica (INFOESCOLA,2006).

A lógica de programação é muito relevante para desenvolver programas em qualquer linguagem de programação. A finalidade dela é ligar os pensamentos do programador para alcançar as desejadas metas. Para atingir esse objetivo, esses pensamentos devem seguir uma sequência até chegar à concretização, esses passos, ou seja, essa sequência é chamada de sequência Lógica.

1.2. ALGORITMO

Na vida real, para fazer qualquer coisa, desenvolver algo, há de seguir um passo a passo, uma descrição do processo do início ao fim, essa descrição é chamada de algoritmo.

Por exemplo para fazer um bolo primeiramente tem que escolher o tipo desejado para definir a receita. Ao definir a receita precisa compra os ingredientes. A receita está definida, os ingredientes e os instrumentos já estão prontos, agora falta preparar massa. Depois de preparar a massa seguindo as devidas normas o próximo passo é colocá-la no fogão. Ao retirá-la no fogão é obtido um bolo pronto.

Repare que no decorrer da preparação do bolo, três etapas foram passadas: a comprar dos ingredientes, a preparação da massa até chegar ao fogão e finalmente retirada do bolo no fogão. É possível deduzir que o algoritmo é dividido em três etapas: entrada, processamento e saída. Essas etapas são de uma importância capital na linguagem de programação.

1.3. HTML

De acordo com a imagem a seguir (Figura 1 - HTML) a sigla HTML significa (*HyperText Markup Language*, linguagem de marcação de hipertexto). A

primeira versão do HTML foi baseada na linguagem SGML (*Standard Generalized Markup Language*, linguagem de marcação Generalizada).

O SGML era utilizado para a estruturação de documentos e foi dele que o HTML herdou diversas tags tais como: título <h1> ao <h6>, cabeçalho <head> e parágrafo <p>. A maior diferença entre essas duas linguagens de marcação é que o HTML implementava a tag <a> com o atributo href, permitindo assim a ligação (links) de uma página a outra. Esse conceito de interligação entre documentos é a base do funcionamento de toda *Web*.

Figura 1 - HTML



Fonte: (KENDSNYDER - 2017)

O criador da HTML foi o inglês Tim Berners-Lee, em 1991. Sua criação se concebeu no CERN (*European Council for Nuclear Research*, Conselho Europeu de Pesquisa Nuclear), na suíça. A sua intenção era interligar computadores do laboratório a outras instituições de pesquisa e exibir documentos científicos de forma simples e fácil de acessar. (SOUSA,2004).

Para muitos Vannevar Bush é o verdadeiro pai do hipertexto. Em 1945 escreveu um artigo intitulado "As que may think", no qual descrevia um dispositivo

chamado Memex. O seu objetivo era aumentar a memória humana providenciando os meios para organizar a informação associadamente (OTAVIOARC,2017).

Entretanto a contribuição de Ted Nelson para o desenvolvimento do hipertexto e do hipermedia é profunda, estendendo-se até à própria origem dos termos hipertexto e hipermedia. A partir de 1960, desenvolveu um paradigma compreensivo para a implementação de um sistema hipermedia distribuído, que cobria todo o espectro de saídas: dos algoritmos à economia. O resultado deste paradigma é o projecto Xanadu. (OTAVIOARC,2017).

Em 1992, com a liberação da biblioteca de desenvolvimento para WWW (*World Wide Web* - Rede de Alcance Mundial), deu-se a construção de vários *browsers* (navegadores) WWW e servidores que viabilizaram a *WEB*. O Mosaic, foi um dos primeiros (W3C,2017).

Devido ao rápido crescimento na *web*, foi necessário criar um padrão para que todos os navegadores pudessem reconhecer quaisquer versões dos HTML, que são as seguintes: HTML 2.0, HTML 3.2, HTML 4.0 e o HTML 4.01.

1.4. CSS

De acordo com a imagem a seguir (Figura 2 - CSS) a sigla CSS (*Cascading Style Sheets*, folhas de estilos em cascata) é uma linguagem de folhas que definem o padrão da apresentação de um documento ou uma página na *Web* criada, por exemplo, em HTML sua principal utilidade é ajudar na separação do conteúdo, do formato do site.

Figura 2 CSS



Fonte: (SOLOLEARN - 2017)

Para utilizar o CSS o programador precisa criar a formatação em uma página separada do código para Web, ligado a ele por um link, que vai leva-lo a um caminho em que permite a modificação do seu arquivo, atualmente as versões do CSS são CSS2, CSS3.

Quando ouvir falar de acessibilidade, performance e manutenção, tem-se como princípio fazer separação do conteúdo, da interatividade e da apresentação de um site ou aplicação *web*. O CSS desempenha um grande papel na camada da apresentação.

A forma certa de publicar um documento *web* é seguindo uma estrutura semântica. O CSS traz toda a informação do *layout*, isto é, cores, posicionamento, fontes, tamanhos e imagens de fundo, enquanto o HTML deve fornecer uma “arquitetura” para o conteúdo.

O suporte a CSS pelos navegadores de hoje é bem sólido, mas teve um início tímido, sendo inicialmente suportado pelo navegador Netscape. Até 2009 nem todos os navegadores em uso suportavam plenamente seus recursos. Uma nova versão da especificação está em desenvolvimento e felizmente os navegadores mais recentes já estão testando-a (ELIS,2016).

1.5. NAVEGADOR

Um navegador de *Internet* também conhecido pelos termos inglês *web browser* ou simplesmente de *browser* é o *software* (programa) que permite acessar a *internet*, veja vídeos, escute música, jogue e interaja com documentos virtuais da *internet*, também conhecidos como páginas da *web*.

O navegador que usa faz uma enorme diferença na maneira como quem visualiza ou percebe a *Internet*, então ter a capacidade de escolher seu navegador é importante.

Basicamente, os navegadores transformam as páginas codificadas em HTML para uma visualização compreensível para o usuário comum. O HTML é um padrão de marcação de hipertexto (textos, imagem, vídeo e áudio) que define como os elementos de uma página devem ser exibidos. Assim, ao invés dos usuários de *Internet* terem que entender o comando `navegador` (marcação

que faz a palavra aparecer em negrito), o navegador exibe a palavra navegador em negrito, facilitando a compreensão dos usuários.

Atualmente, as páginas exibidas são bem diferentes das páginas primitivas da *Web*, criadas no começo da década de 1990. (TECMUNDO,2009).

A padronização de folhas de estilo em CSS permitiu inovações na formatação de páginas e o isolamento das equipes de desenvolvimento e de design das páginas.

As folhas de estilo permitem definir, para cada elemento da página, suas cores, fontes, tamanhos e definir modificações que ocorrerão quando o usuário fez algo.

Além dos avanços visuais na exibição das páginas da *Web*, os navegadores processam linguagens de programação completas como a linguagem JavaScript, com a qual é possível o usuário interagir com a página da *Web* sem que a comunicação vá até o servidor ou, através dos avanços obtidos com a técnica de Asyncronous Javascript And XML (AJAX), carregando um único elemento da página, sem que toda ela seja carregada.

Recentemente, foi padronizado o HTML5, que traz inovações no que os navegadores apresentam para os usuários, trazendo como principal característica a exibição de vídeos diretamente, sem a necessidade de um *software* adicional, como o Adobe Flash Player.

Como os navegadores, os sistemas operacionais também são pontes que possibilitam a comunicação entre o usuário e o computador. Eles são uma interface entre a linguagem do usuário e a linguagem do computador, expressa em formato binário.

O sistema operacional permite que o usuário execute tarefas como o gerenciamento das aplicações em execução, do acesso à memória e dos arquivos no disco.

1.6. SERVIDOR

Um servidor pode ser considerado como um *software* ou computador, ele consiste em um sistema de computação centralizado que fornece serviços a uma rede de computadores chamada de cliente, os servidores podem oferecer várias funcionalidades, tais como serviços de dados ou de recursos de sistemas.

Com o passar dos anos a virtualização começou a cair no esquecimento devido a criação de novas aplicações cliente/servidor e ao declínio da plataforma mainframe que perdeu força frente a ascensão da plataforma x86. De acordo com a VMWare, a ampla adoção do Windows e Linux como sistema operacional em servidores na década de 1990 acabaram por estabelecer a arquitetura x86 como padrão da indústria (REDAÇÃO,2017).

Então, em 1999, a VMWare Inc. introduziu o conceito de virtualização na plataforma x86 como uma maneira mais eficiente para utilizar o equipamento desta plataforma, aproveitando servidores x86 para fornecer uma estrutura computacional que possibilitasse o total aproveitamento dos recursos computacionais destes servidores. (GOLDBERG,2009).

Os servidores são usados em empresa de grandes e médio porte, embora um servidor possa equivaler a um *software* ou a partes de um sistema computacional, ou até mesmo a uma máquina que não seja necessariamente um computador, existem vários tipos de servidores quais como, servidor de fax, arquivos, *web*, impressão, banco de dados, *DNS* (*Domain Name System* - Sistema de nomes de domínios), *proxy*, imagens, *FTP* (*File Transfer Protocol* - protocolo de transferência de arquivos), *webmail*, virtualização, e por último o servidor de sistema operacional.

1.6.1. Servidor Local

É um conjunto de *hardware* e *software* que permite que computadores individuais estabelecerem comunicação entre si, trocando e compartilhando informações e recursos às tais redes são denominadas locais por cobrirem apenas uma área limitada a 1 km no máximo, além do que passam a ser denominadas de MANs.

As redes em áreas maiores necessitam de tecnologias mais sofisticadas, visto que, fisicamente, quanto maior a distância de uma rede a outra rede, maior a taxa de erros que ocorrerão devido à degradação do sinal.

1.6.2. Servidor Web

O servidor web é a peça mais importante da infraestrutura de um site na internet. Ele é um programa que usa o HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto) para servir os arquivos que formam páginas da web para os usuários, em resposta aos seus pedidos, que são encaminhadas pelos clientes HTTP de seus computadores. Computadores dedicados e equipamentos podem ser referidos como servidores web também.

Servidor web ele é responsável pelo armazenamento de páginas de um determinado site, requisitados pelo cliente através de *browsers*, ele aceita pedidos de HTTPS de clientes, o processo do servidor web começa através de uma conexão entre o computador onde é instalado o servidor web no computador do cliente, a partir disso é processado o pedido cliente e conforme com as restrições de segurança e a existência da informação solicitada, o servidor devolve os dados, e por último os servidores web podem executar os programas de *scripts* interligados com o usuário.

Os servidores web estão disponíveis dia e noite, pois a capacidade de conexão deve estar disponível a qualquer momento. Os pedidos http que se referem às páginas HTML são feitos através de *browsers*. O processo é iniciado com conexão entre o computador onde está instalado o servidor web e o computador do usuário/cliente.

1.7. LINGUAGEM DE PROGRAMAÇÃO

Uma linguagem de programação é utilizar métodos de instruções para se comunicar com um computador, foi na década de cinquenta que as primeiras linguagens de programação vêm aparecendo, as linguagens são projetadas para adotar uma sintaxe de nível baixo, médio e alto.

Foi em 1942 que se originou a 1ª linguagem de programação de alto nível do mundo, chamada de Plankalkul. Esta linguagem foi criada pelo alemão Konrad Zuse que também projetou e construiu o primeiro computador eletromecânico, o Z3, constituído por relés, que efetuava cálculos, era programável e exibia os resultados em fita perfurada. Era possível trabalhar com números de ponto flutuante e em binário. (PETNEWS,2012).

A linguagem Plankalkul foi considerada uma revolução na dinâmica das linguagens de programação pois introduziu técnicas e opções inovadoras, tais como: atribuições, expressões aritméticas, adição dos tipos derivados (inteiros, real, arranjos), laços e repetição, dentre outras. Zuse criou vários algoritmos, dentre eles: algoritmos de ordenação, busca em grafos, entre outros. (PETNEWS,2012).

Mesmo com a ideia da linguagem Plankalkul, programar ainda era uma atividade trabalhosa e bastante suscetível a erros. Na maioria dos casos era mudar a posição de fios ou furar cartões. Era necessário muito tempo, bastante atenção e dedicação por parte dos programadores. Para que se tenha ideia, os números de ponto flutuante deviam ser armazenados em uma certa posição e sempre que fossem manipulados o programador deveria saber a posição e o endereço de destino. Era fundamental que novas formas fossem definidas para tornar a programação mais fácil. (PETNEWS,2012).

Por volta da década de 50, surgiram os primeiros compiladores, o que proporcionou o surgimento de uma nova linguagem de programação que atendia aos anseios dos programadores. Tratava-se da linguagem Fortran que foi desenvolvida a partir de um projeto da IBM liderado por John Backus. Inicialmente não havia a compilação modular e como os programas começaram a crescer bastante, a compilação de todo o código do programa tornou-se um grave problema. Com isso foi lançada a versão Fortran II onde havia a possibilidade de executar módulos, e não programas inteiros. Essa linguagem é utilizada até hoje e foi a primeira a tornasse popular entre a comunidade de programação. (PETNEWS,2012).

Contudo isso descrito acima foi surgindo a linguagem de programação e a cada dia que passa a linguagem vem sendo cada vez mais importante para os técnicos de TI e entre outros.

1.7.1. Linguagem de programação *desktop*

Por definição, uma aplicação *desktop* é qualquer *software* ou sistema que pode ser instalado em um computador e usado para executar tarefas específicas. Essas aplicações estão armazenadas na memória do computador e também não necessitam da conexão de *internet* para funcionarem, programas como: Word, Excel, adobe reader, WinRar, são exemplos de aplicações desenvolvidas em linguagens voltado à plataforma *desktop*.

A história da programação começou com a plataforma *desktop*, pois o primeiro *software* desenvolvido em 1948 na Inglaterra baseado no sistema matemático de John von Neumann. Essa plataforma nunca para de crescer, hoje existe várias linguagens voltadas a ela, por exemplo, C++, C#, Visual Basic, Python, Java, etc (REDAÇÃO MUNDO ESTRANHO, 2016)

1.7.2. Linguagem de Programação *mobile*

O aplicativo *mobile* são destinados a serem executados em aparelhos celulares. São aplicativos desenvolvidos para um certo tipo de plataforma. Essas plataformas envolvem diversas tecnologias, tais como: sistemas operacionais, linguagens de programação e IDEs (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado).

O sistema operacional é responsável por fazer conexão entre *hardwares* e *softwares* do aparelho, as linguagens de programação são usadas no desenvolvimento de *softwares* e os IDEs são conhecidos como ambiente de desenvolvimento, pois fornece ferramentas necessárias para o desenvolvimento de *softwares*. Atualmente, existem vários sistemas operacionais voltados à plataforma *mobile*, tais como: *Androide (google)*, *iOS (Apple Inc.)*, *Windows Mobile (Microsoft Corporation)*.

Cada um dos sistemas operacionais requer uma linguagem de programação específico como, por exemplo, Java para o sistema *Androide*, C++ ou C# para o sistema *Windows Mobile*, etc.

1.7.3. Linguagem de programação Web

Trata-se de sistemas ou aplicativos *web*, todos os aplicativos que depende da conexão de *internet* para se funcionarem. A tecnologia de *internet* surgiu na década de 1960 nos Estados Unidos por estudantes de universidades norte-americanas em meio à Guerra Fria. Seu primeiro nome foi “*ARPANET*”, depois passa a ser chamado de “*INTERNET*”. Foi desenvolvido âmbito de garantir que as informações estariam salvas caso acontecesse invasões da União Soviética, devido às ameaças constantes da *URSS* naquela época. (TECMUNDO,2009).

World Wide Web em português rede mundial de computadores ou WWW foi o primeiro sistema desenvolvido que ligou as universidades entre si para que os trabalhos e pesquisas acadêmicos fossem utilizados mutuamente em um ambiente de contribuição dos lados envolvidos. A invenção do mesmo foi do Berners-Lee. (TECMUNDO,2009).

As linguagens de programação *web* têm como finalidade de desenvolver *web* sites, portais, aplicações *web* em geral.

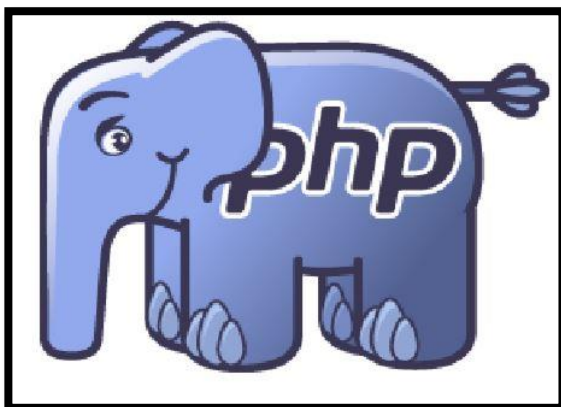
Usavam linguagem de programação para o desenvolvimento de *softwares* ou sistemas voltados à plataforma desktop, ou seja, aplicações (como Word, Excel, Media Player) que funcionam direto no computador, sem precisa de conexão de *internet*, quando surgiu as primeiras linguagens de programação. Com o avanço da *Internet* aplicações *web* passaram a ser muito relevante.

Os aplicativos da *web* ao contrário dos outros aplicativos precisam de um servidor *web* para serem executados. Usa-se várias linguagens de programação para desenvolver *web* sites tais como o Java Script, Java, mas o foco desse trabalho é o PHP.

1.8. PHP

De acordo com a imagem a seguir (Figura 3 – PHP), o PHP (*Personal Home Page* em Português pagina pessoal) é uma linguagem de script embutida no HTML, muito da sua sintaxe é emprestada de C, Java e Perl com algumas características específicas do PHP juntas.

Figura 3 - PHP



Fonte (ARQUIVO PESSOAL – 2015)

O objetivo da linguagem é permitir que os desenvolvedores *web* escrevam páginas geradas dinamicamente de forma rápida, o PHP tem código fonte disponível para qualquer pessoa, sem custos. A licença de uso e edição é *Open Source*, ou seja, ninguém pode comercializar qualquer versão modificada do PHP, e qualquer modificação deve continuar com o código fonte aberto para os usuários explorarem e modificarem. Este sistema de licença não traz lucro aos desenvolvedores, pois eles disponibilizam tudo para o público gratuitamente, e o público, por sua vez, ajuda divulgando erros e ajudando a modificar o código fonte.

Muitas empresas apoiam os desenvolvedores do PHP, pois estes não visam lucro ao criarem e desenvolverem o programa e sim ajudar quem precisa, O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros.

O PHP também é suportado pela maioria dos servidores *web* atualmente. Isso inclui o Apache, o IIS e muitos outros. Também qualquer servidor *web* que pode utilizar o binário FastCGI do PHP, como o *lighttpd* e o *nginx*. O PHP trabalha tanto como módulo quanto como um processador CGI.

Uma das características mais fortes e mais significativas do PHP é seu suporte e sua ampla variedade de banco de dados. Escrever uma página *web* consultando um banco de dados é incrivelmente simples usando uma das extensões específicas de um banco de dados (MySQL), ou usando uma camada de abstração como o PDO ou conectar a qualquer banco de dados que tem suporte ou padrão

"Open Database Connection em português abrir conexão de banco de dados" usando a extensão ODBC.

Outros bancos de dados podem utilizar URL (Uniform Resource Locator, Localizador Padrão de Recurso), como o CouchDB. As versões do PHP são (PHP/FI 2.0 ele é uma versão mais antiga e não mais suportada do PHP. PHP 3 é o sucessor do PHP/FI 2.0, PHP 5 é a geração atual do PHP.

1.9. APLICATIVO NOTEPAD++

De acordo com a imagem a seguir (Figura 4 – Notepad++) Notepad++ é um editor de texto e de código fonte de código aberto sob a licença GPL (*General Public License*, Licença Pública Geral). Suporta várias linguagens de programação rodando sob o sistema Microsoft Windows (possível utilização no Linux).

Figura 4 - Notepad++



Fonte (WIKIMEDIA – 2017)

O Notepad++ é distribuído como um *Software* livre. O projeto foi hospedado no SourceForge.net, onde foi baixado mais de milhões de vezes e ganhou duas vezes o prêmio "SourceForge Community Choice Award" por melhor ferramenta de desenvolvimento. O projeto foi hospedado no TuxFamily desde 2010 até 2015 depois foi movido para o GitHub. Ele é baseado no Scintilla, é escrito em C++ utilizando a API (*Application Programming Interface*, Interface de Programação de Aplicativos) Win32 e usa a STL (Standard Template Library, Biblioteca Padrão de Gabaritos). O objetivo do Notepad++ é oferecer um esguio e

eficiente binário com uma interface gráfica totalmente modificável (TECMUNDO,2017).

Através do Notepad++ será possível editar códigos em diversas linguagens sejam de programação ou não, dentre mais de quarenta linguagens disponíveis. Outra função que pode agradar principalmente *web designers* é a possibilidade de abrir o texto nos navegadores *Firefox*, *Internet Explorer*, *Google Chrome* e *Safari*, além de uma opção para definir qual programa deseja executar (como outro navegador, ou, talvez, um compilador, no caso de linguagens de programação).

1.10. VISUAL STUDIO CODE

De acordo com a imagem a seguir (Figura 5 – VISUAL STUDIO CODE) o Visual Studio Code editor multi plataforma de código da Microsoft foi desenvolvido para uso no Linux, Mac e no Windows (PPL WARE,2015). É de código aberto, leve e suporta diversas linguagens de programação como: C++, C#, ASP.NET, PHP, Python, Java, entre outras além de padrões Web HTML, JavaScript e CSS (TECNOBLOG, 2015).

Figura 5 - Visual Studio Code



Fonte (VISUAL STUDIO CODE - 2017)

O Visual Studio Code traz muita praticidade no momento da codificação que é um ponto muito importante no desenvolvimento do Website (TECNOBLOG, 2015). Outro fator é a interação com o Sistema de Versão Git e o GitHub ferramentas importantes no desenvolvimento da aplicação Portal Website Tech Train.

1.11. GIT

De acordo com a imagem a seguir (Figura 6 - GIT) o Git é um sistema de controle de versão distribuído, desenvolvido pelo Linus Torvalds e usado no desenvolvimento do Kernel do Linux (OFICINA DA NET, 2015). O Git trabalha com mesmo diretório, podendo fazer as alterações do projeto, gravando documentação e comentários (OFICINA DA NET, 2015). O Git é capaz de gravar tudo o que foi feito. E este registro é muito útil para que o desenvolvedor possa voltar a fase anterior quando surgir um possível problema. Além disso, através do Git é possível fazer a criação de áreas separadas entre si para realizar testes ou mesmo projetos diferentes.

Figura 6 - Git



Fonte (SUBSTANCIANDO.BLOGSPOT - 2013)

O Git tem ainda como funcionalidade residir na ramificação e integração de documentos. Com isso é possível trabalhar em arquivos de forma independente sem ter que alterar o arquivo principal antes da publicação oficial (OFICINA DA NET, 2015). O trabalho em equipe é outro ponto forte do Git e neste último quesito é um grande diferencial no desenvolvimento da aplicação web proposta neste trabalho.

1.12. GITHUB

De acordo com a imagem a seguir (Figura 7 - GITHUB) o GitHub foi lançado em 2008, seu uso e aplicabilidade designa em hospedar projetos (OFICINA DA NET, 2015). Para controle de versão é usado o Git.

O GitHub oferece alguns recursos de redes sociais, já que é possível seguir projetos de outros desenvolvedores e ainda comentar sobre todos eles (OFICINA DA NET, 2015). E além de tudo, o GitHub permite compartilhar códigos em blocos é disponível gratuitamente, com limite de armazenamento de 300MB (OFICINA DA NET, 2015). Se caso for necessário um armazenamento à acima de 300MB, torna-se pago.

Figura 7 GitHub



Fonte: (KANBANIZE - 2017)

O projeto pode ser acessado de qualquer local com uso do GitHub, que é de suma importância para a equipe, quanto ao acesso, construção, correção, testes e validações na implementação do Portal Website Tech Train.

1.13. BANCO DE DADOS

De acordo com a imagem a seguir (Figura 8 – BANCO DE DADOS), o banco de dados ou base de dados são armazéns de informações que se relacionam de forma a criar um sentido. São de vital importância para empresas, e há muito tempo se tornaram a principal peça dos sistemas de informação até porque, hoje em dia

todos os programas desenvolvidos fazem a conexão com o banco de dados, isto é uma requisição do mercado de *software*.

Figura 8 - Banco de Dados



Fonte (PORTAL GSTI – 2014)

O conceito de banco de dados não era novidade quando na década de 1950 iniciando década de 1960 os dados começaram a ser armazenados em fitas magnéticas e Decks de cartão perfurado. A leitura era feita sequencialmente, os programas realizavam seu trabalho de forma específica (DICAS DE PROGRAMAÇÃO,2017).

No final de 1960 início da década de 1970 Surgiram os discos rígidos e suas facilidades. Os dados não necessitam de processamento “sequencial”. Surgiram bancos de dados em rede e hierárquico cujos dados eram organizados em lista e arvores.na década de 1980 surgiram os projetos System R (Sistema R) e SQL/DS, banco de dados relacional. Nas mesmas décadas de 80 surgiram outros bancos de dados, a Oracle apresentou o Oracle 2 e a IBM o SQL/DS (que se tornou DB2), ambos sistemas comerciais de bancos de dados. Na sequência vieram SQL Server, MySQL, DBase III, Paradox, etc. (DICAS DE PROGRAMAÇÃO,2017).

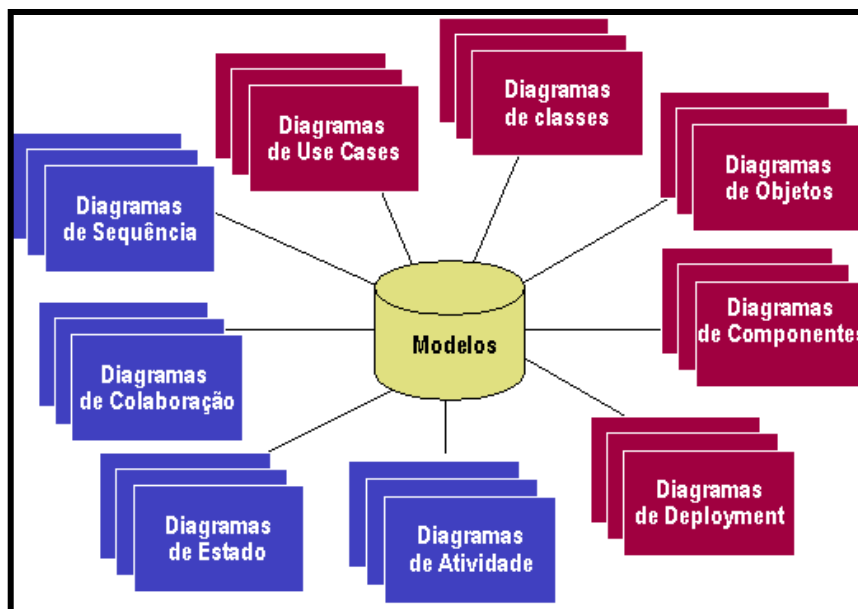
Atualmente existem vários modelos de bancos de dados tais como orientado a objetos, orientado a documentos e o banco de dados relacional, o mais comum. Cabe sempre ao programador de escolher qual tipo de base de dados que atenderia a suas necessidades.

1.14. UML

De acordo com a imagem a seguir (Figura 9 – UML) UML é um acrônimo para a expressão *Unified Modeling Language* que significa Linguagem Unificada de Modelagem. Pelo conceito é possível ver que a UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que se desenvolve.

Ela fornece nove tipos de diagramas conforme a imagem abaixo: diagrama de sequência, diagrama de colaboração, diagrama de estado, diagrama de atividade, diagrama de desenvolvimento, diagrama de componentes, diagrama de classes, diagrama de objetos e diagrama de caso de uso. O último será usado no desenvolvimento desse projeto por isso será mais detalhado nos próximos parágrafos.

Figura 9 - UML



Fonte: (COMPUTAÇÃO UFCG – 2017).

A UML foi inventada em 1994 por dois pesquisadores, analistas de projetos de sistemas orientado a objetos, respeitados na área de desenvolvimento de *software*. O objetivo dos dois investigadores era unificar dois métodos foram

desenvolvidos: o método Booch, que descrevem um conjunto de objetos e as suas relações e da OMT (*Object Modelling Tool* – ferramenta de modelagem de objeto) de Rumbaugh, objeto orientado, combinando notações de modelação orientada ao objeto, dados, componentes e fluxos de trabalho.

O chamado UML vem dos nomes de três participantes do projeto. Esta linguagem tratada vários métodos, técnicas e aspectos em notações comuns, mas usando diferentes; aprendendo desvantagens, aplicações e uso de ferramentas e também entre diferentes abordagens concorrentes, portanto, deve estabelecer uma notação padrão. É o primeiro método para publicar sua própria notação, incluindo a notação para a maioria dos requisitos de informação, análise e design. A *Unified Modeling Language* é usada para entender, design, configurar, manter e controlar sistemas de informação para construir (BLOGSPOT,2012).

A primeira versão da UML foi a 0.8 que surgiu em outubro de 1995. Em junho de 1996, após uma solicitação de proposta realizado pela OMG, as versões 0.9 e 1.0 da UML foram lançadas pelos pesquisadores. Em janeiro de 1997, novas contribuições lançaram o UML 1.1. (BLOGSPOT,2012).

A partir da versão 1.1, a comunidade de desenvolvimento de *software* começa a fazer uma aderência maciça ao UML. Em novembro de 1997, o UML foi adotado como norma pela OMG, que estabeleceu um RTF (Revision Task Force) para aperfeiçoar pequenos detalhes na linguagem. Em junho de 1999 o RTF libera a versão UML 1.3 e em setembro de 2001 é lançado o UML 1.4. Em março de 2003, publica-se a versão 1.5, que combina os detalhes da versão 1.4 com uma semântica de ações. Em julho de 2004, a versão 1.4.2 é criada, sendo simultaneamente publicada como a norma ISO/IEC19501.

Em julho de 2005 publica-se a versão 2.0 da norma, que traz diversas inovações (dentre outras, substituindo o diagrama de colaboração pelo diagrama de comunicações - uma versão aperfeiçoada do mesmo). A versão 2.1 do UML nunca foi publicada como uma especificação formal, mas em agosto de 2007 disponibilizou-se a versão 2.1.1 e em novembro de 2007 a versão 2.1.2. A versão 2.2 da norma foi publicada em fevereiro de 2009 e em maio de 2010 publicou-se a versão 2.3.

Em março de 2011 publicou-se a versão 2.4 e em agosto de 2011 saiu uma versão corrigida, a 2.4.1. Em junho de 2015 foi publicada a versão 2.5, que é a versão mais atual da norma até a data da elaboração deste documento. (RICARDO R. GUDWIN,2015).

1.15. DIGRAMA DE CASO DE USO

O diagrama de caso de uso é um modelo que tem como papel descrever graficamente como diferentes tipos de usuários interagem com o sistema para executar uma determinada tarefa. Ele descreve as metas dos usuários, as interações entre os usuários e o sistema, bem como o comportamento necessário do sistema para satisfazer estas metas. O diagrama de caso de uso cumpre também o papel de unificador, por isso se torna relevante no desenvolvimento de projeto orientado ao objeto. O diagrama de caso de uso, como mostra a imagem 5, é um conjunto de elementos de modelo, sendo os autores, o caso de uso e a relação entre os autores.

O ator representa uma função que uma pessoa ou um sistema externo executa ao interagir com o sistema.

O caso de uso nada mais e que uma sequência de ações executadas pelo sistema que geram resultado para um determinado autor. Os autores se comunicam entre si, as associações são usadas para descrever os relacionamentos entre eles e os casos de uso em que eles participam.

1.16. MODELO ENTIDADE RELACIAMENTO

Na engenharia de *software* usa-se, para descrever os objetos ou entidades envolvidas em um domínio de negócio (universo), como pode se ver na figura 10, com suas propriedades e a maneira cuja elas se relacionam entre si, o Modelo de Entidades e Relacionamentos (MER) nada mais é um paradigma conceitual que representa a estrutura que a base de dados do sistema possuirá.

O relacionamento pode ser simples (envolve duas entidades) ou complexo (envolve três ou mais entidades). E de acordo com a quantidade de objetos envolvidos pode-se classificar em três formas:

Relacionamento 1: 1 (um para um): quando só um objeto de uma entidade é ligado a um outro objeto da outra. Por exemplo, em um sistema que contém entidades como: usuário e currículo, nesse contexto, um usuário do sistema não pode ter vários currículos vice-e-versa.

Relacionamento 1: N (um para muitos): nesse caso vários objetos de uma das entidades envolvidas podem fazer ligação somente com um objeto da outra. Por exemplo, em um sistema que contém entidades como: funcionário e departamento, vários funcionários trabalham em um departamento, mas um funcionário não pode trabalhar em vários departamentos.

Relacionamento N: N (um para muitos): quando vários objetos de uma das entidades envolvidas podem fazer ligação com vários objetos da outra classe vice-e-versa. Por exemplo, em sistema de compras um cliente pode comprar vários produtos assim como um produto pode ser comprado por vários clientes.

1.19. SQL

O SQL (*Structured Query Language* - linguagem de consulta estruturada) é a linguagem de programação utilizada para se armazenar e consultar dados. E conhecê-la pode fazer a diferença na gestão de Tecnologia da Informação. O SQL é fundamental para qualquer sistema que armazena dados, empresas, de qualquer tamanho.

O SQL foi criado na década de 70 pela IBM com base em conceitos de álgebra. E tornou-se a linguagem padrão universal para a manipulação de bancos de dados que podem ser relacionados por meio dos SGBD (*Data Base Management System* - Sistema de Gerenciamento de Bancos de Dados Relacionais). (IBM,2017).

A IBM apresentou em 1974 um protótipo de base de dados relacional (System/R). O projecto terminaria em 1979, tendo o seu sucesso conduzido a resultados significativos. Estava desenvolvida uma linguagem para interrogação a

múltiplas tabelas do System/R com suporte para múltiplos utilizadores. Foi designada Structured English Query Language (SEQUEL). Viria a evoluir para a designação atual. (IBM,2017).

A Oracle Corporation (inicialmente RelationalSoftware, Inc.) viria a desenvolver em 1979 a 1ª versão comercial do SQL. (IBM,2017).

A partir de então o sucesso do SQL conduziu ao aparecimento de SQL/Data System (SQL/DS) posteriormente Database 2 (DB2). O empenhamento da própria IBM determinou efectivamente a consolidação do SQL.

Atualmente existem diversos esforços no sentido da criação de uma definição *standard* para integrar os múltiplos fabricantes. Todas as grandes marcas mundiais aderiram a este *standard*, nomeadamente: Microsoft SQL Server, Oracle, Informix, Sybase e obviamente IBM DB2.

Com o das parcerias a Microsoft lançou a versão 6.0 do SQL Server no ano 1995. (IBM,2017). E o lançamento foi uma das maiores rescritas da tecnologia SQL Server. A versão 6.0 aumentou o desempenho substancialmente provendo mecanismo interno e administração centralizada. Daí em diante o SQL só evoluir.

1996 – A Microsoft lançou a versão 6.5 do SQL Server. Essa versão trouxe melhorias significativas para a tecnologia disponibilizando novas funcionalidades. (TRABALHOSFEITOS,2017).

1997 – A Microsoft lançou a versão Enterprise do SQL 6.5. (TRABALHOSFEITOS,2017).

1998 – A Microsoft lançou a versão 7 do SQL Server o qual foi completamente reescrito. (TRABALHOSFEITOS,2017).

2000 – A Microsoft lançou o SQL Server 2000, o SQL Server 2000 é o lançamento mais importante até hoje. (TRABALHOSFEITOS,2017).

1.20. MARIADB

De acordo com a imagem a seguir (Figura 11 - MariaDB) MariaDB é um binary drop em substituição da mesma versão do MySQL (por exemplo MySQL 5.1 -

> MariaDB 5.1, MariaDB 5.2 e MariaDB 5.3 são compatíveis. O MySQL 5.5 será compatível com o MariaDB 5.5). O que isto quer dizer é que:

- Arquivos de definição de tabelas e dados (.frm) são binariamente compatíveis.
- Todas as APIs clientes, protocolos e estruturas são idênticas.
- Todos os nomes de arquivo, binários, caminhos, portas, sockets, entre outros deveriam ser os mesmos.
- Todos os conectores do MySQL (PHP, Perl, Python, Java, .NET, MyODBC, Ruby, o conector MySQL, etc) funcionam inalterados com MariaDB.

Figura 11 - MariaDB



Fonte: (MARIADB - 2017)

MariaDB é um servidor de banco de dados que oferece a funcionalidade de substituição drop-in para o MySQL. MariaDB é construído por alguns dos autores originais do MySQL, com a ajuda da grande comunidade de desenvolvedores de *software* livre e *Open Source*. Além das funcionalidades básicas do MySQL, MariaDB oferece um rico conjunto de aprimoramentos de recursos, incluindo mecanismos de armazenamento alternativo, otimizações de servidores e patches.

MariaDB é impulsionada principalmente pelos desenvolvedores de Monty Program, uma empresa fundada por Michael “Monty” Widenius, o autor original do MySQL, mas isso não é toda a história sobre MariaDB. Em “Sobre MariaDB” página onde vai encontrar mais informações sobre todos os participantes da comunidade MariaDB, incluindo o armazenamento e motores XtraDB PBXT. (BENTO,2010).

MariaDB.com é organizado pela *Open Database Alliance*, um consórcio independente de fornecedor de desenvolvedores de código aberto do banco de dados e provedores de solução. O *Open Database Alliance* garante que onde vai encontrar produtos, suporte e serviços para MariaDB que são os melhores da indústria do *software* tem para oferecer.

1.21. FRAMEWORK BANCO DE DADOS

Xampp é o *framework* que foi utilizado durante o desenvolvimento desse projeto, pois disponibiliza as ferramentas necessárias, apache, e SQL Server, que são muito importantes para a realização desse projeto. O nome prove da abreviação de X (para qualquer dos diferentes sistemas operativos), Apache, MariaDB (o substituto de MySQL), PHP, Perl.

1.22. CORELDRAW

O CorelDRAW Graphics Suite, ou simplesmente CorelDRAW é o aplicativo pioneiro em diagramação de páginas e edição vetorial para a plataforma Windows.

Foi lançado em 1989 como o primeiro programa capaz de reunir imagens, vetores e textos coloridos dentro de um mesmo arquivo. Dois anos mais tarde, com o lançamento da versão 3, em 1991, trouxe a primeira suíte de edição gráfica da história, que combinava ilustração vetorial, layout de página, edição de fotos e muito mais em um único pacote (LIUTE CRISTIAN,2013).

O aplicativo tem uma história memorável e atualmente se mantém como o programa mais usado no Brasil somando gráficas, bureaus de serviço, profissionais do design e amadores. O CorelDRAW, atualmente em sua versão X8 (18) é intuitivo e é bem mais fácil de se aprender a usar do que seus concorrentes, em especial o Adobe Illustrator, embora sua principal vantagem sobre ele durante um bom tempo tenha sido o preço, o aplicativo cativou principalmente usuários menos experientes

que são capazes de desenvolver projetos usando recursos de maneira menos descomplicada.

Não sendo por isso um aplicativo amador, pelo contrário, a suíte CorelDRAW é usada por grandes profissionais capazes de projetar material complexo e com um nível de detalhe surpreendente.

O CorelDRAW tomou o mundo da computação gráfica de assalto em 1989 com a introdução de um programa de layout e ilustração vetorial totalmente colorida, o primeiro do gênero. Dois anos depois, a Corel revolucionou o setor novamente com a introdução da primeira suíte completa de aplicativos gráficos, a versão 3, que combinava ilustração vetorial, layout de página, edição de fotos e muito mais em um único pacote (BN2 DESIGN,2010).

Versão 2 (1991): ferramenta envelope (para distorção de texto ou objetos usando uma forma primária), mistura, extrusão (para simulação de perspectiva e volume em objetos) e perspectiva (para distorcer objetos ao longo dos eixos X e Y) (GNMILASI,2008).

Versão 3 (1992): Incluía o Corel PHOTO-PAINT* (para edição de imagens bitmap), CorelSHOW (para criação de apresentações on-screen), CorelCHART (para tabelas gráficas), Mosaic and CorelTRACE (para vetorização de bitmaps). A inclusão deste *software* originou as verdadeiras suítes gráficas (GNMILASI,2008).

Versão 5 (1994): Corel Ventura foi incluído no pacote (e logo após vendido como um programa em separado). Era um *software* de *layout* (programação visual, diagramação) como o Adobe PageMaker ou o InDesign (GNMilasi, julho de 2008).

Versão 6 (1995): Interface customizável, ferramentas Polígono, Espiral, Garfo e Borracha. Corel Memo, Corel Presents, Corel Motion 3D, Corel Depth, Corel Multimedia Manager, Corel Font Master e Corel DREAM (para modelagem em 3D) foram incluídos no pacote (GNMILASI,2008).

Versão 7 (1996): Barra de Propriedades sensível ao Contexto, Print Preview com opções de Zoom e Pan, Scrapbook (para visualização de objetos gráficos em arraste-e-solte), opção para Edição em HTML, opções de exibição tipo rascunho e aperfeiçoada, preenchimento interativo e ferramentas de mistura, ferramentas de

transparência, Ferramenta Natural Pen, localizar e substituir, conversor de vetor para bitmap (no CorelDRAW), verificador ortográfico. O pacote incluía os programas Corel Scan e Corel Barista (edição para *web*) (GNMILASI,2008).

Versão 8 (1997): Digger selection, Docker windows, Distorção Interativa, 3D, Envelope e Ferramentas, Realistic Dropshadow tool, Mistura interativa de cores, Editor de Paleta de cores, linhas-guia como objetos, páginas de tamanho customizável, suporte à duotone. O Corel Versions foi incluído neste pacote (GNMILASI,2008).

Versão 9: Mesh fill tool (para preenchimento complexo de cor), Artistic Media tool, Publish to PDF features, embedded ICC colour profiles, Multiple On-screen Color Palettes and Microsoft Visual Basic for Applications 6 support. O pacote incluiu o *software* Canto Cumulus LE, de gerenciamento de mídia (GNMILASI,2008).

Versão 10: Corel R.A.V.E. (aplicativo para animação vetorial), Formas Perfeitas, ferramentas para gráficos *Web* (para criação de elementos interativos como botões), Page sorter, suporte a documento multilíngue, janela Navigator.

Versão 11 (2002): Símbolos de biblioteca, corte de imagem (desenho de *web* de soldado pára-quedista), escovas vetoriais sensíveis a pressão, instrumentos de desenho de 3 pontos (GNMILASI,2008).

Versão 12 (2004): guias dinâmicas, Smart Drawing tools, opção de Exportar para MS Office ou Word, ferramenta para excluir segmento virtual, suporte à texto Unicode. Versão X3 (2006): ferramentas de preenchimento interativo, ferramentas de filete, Vieira e Chanfro, PowerTRACE integrado ao CorelDRAW, laboratório de ajustes de imagem (GNMILASI,2008).

Versão X4 (2008): identificação rápida de fontes, texto espelhado, formatação de texto instantânea, ConceptShare, suporte a arquivos RAW de câmera, camadas independentes de página, criação de tabelas (GNMILASI,2008).

2. PESQUISAS

Para que o projeto pudesse ser concluído de maneira objetiva, uma pesquisa foi feita para analisar as diversas tecnologias necessárias, além do custo e do perfil do cliente.

2.1. MACRO E MICRORREGIÃO

É de sabedoria comum que sites de instruções e/ou cursos *online* existem na internet. Vários e vários sites são utilizados como meio de informação e formação de alunos em diversas áreas, principalmente a área da Tecnologia. Foram denominados estes sites como a Macrorregião do projeto, por serem concorrentes diretos caso o projeto cresça consideravelmente.

Dentro da Microrregião, fazem parte as escolas técnicas do Centro Paula Souza, que poderão indicar aos alunos o site para obter conhecimento maior, e de graça. Além destas, escolas de cunho extensivo-profissional, como People, Microcamp e similares, que também podem ministrar aulas de conhecimento técnico, com a ressalva de serem, normalmente, pagas.

2.2. OBJETIVO DO PROJETO

O projeto tem como objetivo fazer parte deste grupo de websites que liberam informação profissional, porém com o auxílio da comunidade, que indicará cursos de graça que já estão na plataforma YouTube, da Google. Assim, o projeto se diferencia das demais, tratando-se não apenas de um site de cursos *online*, mas também de um *super site* que consiste em agrupar materiais em um lugar fácil de serem encontrados.

Além de ser um portal de treinamento, também teria espaço para um blog para que os desenvolvedores pudessem expor novas funcionalidades, bem como atualizar os usuários com os cursos indicados que foram aceitos, os que ainda estariam em processo avaliativo e os que foram rejeitados. Também é objetivo a

criação de uma avaliação conclusiva do curso, para que um certificado pudesse ser emitido, nominalmente, para o usuário.

2.3. ÁREA DE ATUAÇÃO

A área de atuação dependerá do auxílio da comunidade, mas inicialmente o projeto consistirá de aulas nos cursos de: Informática, Administração, Música e Marketing e qualquer outra gama de conhecimento de nível técnico.

2.4. TECNOLOGIA NECESSÁRIA

A Tecnologia usada no projeto são HTML que é uma linguagem para estruturação e apresentação de conteúdo para a *World Wide Web* em português rede mundial de computadores e é uma tecnologia chave da Internet originalmente proposto por *Opera Software*, CSS para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML (*eXtensible Markup Language*, Linguagem de marcação extensível), HTML e XHTML), PHP (*Personal Home Page*, página pessoal) 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 Wide Web*.

2.5. CONCORRENTES DIRETOS E INDIRETOS

Todos os sites de cursos serão concorrentes diretos, principalmente os mais conhecidos, como Udemy, Curso em Vídeo, Udacity, Lynda;

Como concorrente indireto, pode-se levar em conta as próprias plataformas de upload gratuito de vídeos, como o Youtube, DailyMotion e afins.

2.6. PERFIL DO CLIENTE

Pessoa que deseja compartilhar de seu conhecimento técnico, através de vídeo aulas sobre assuntos diversos. E também qualquer pessoa que deseja ampliar os conhecimentos em diversas áreas que o projeto website puder abranger. Sem restrição de Idade ou de sexo.

2.7. CUSTO E RECEITA

Os custos referentes ao projeto foram calculados a partir de pesquisas feitas no mercado referente ao domínio de um website, servidores e até mesmo foi levado em conta como se o “*TechTrain Group*” fosse realmente uma empresa. Para este último ponto, o grupo precisaria de um computador pessoal cada, além de um espaço físico para trabalho integral. Com isso, chegou-se à conclusão explícita na Tabela 1, onde pode-se ver os gastos com todos os computadores, o aluguel de um estabelecimento para trabalho e gastos com o domínio na internet:

Tabela 1 - Receita

Custo Total		
Item	Quantidade	SubTotal
Ibyte I-ITW10	9	R\$ 26.691,00
Aluguel	1	R\$ 1.200,00
Gastos Locais	1	R\$ 800,00
Domínio	1	R\$ 54,90
Hospedagem	12	R\$ 298,80
Total		R\$ 32.044,70

Fonte: (AUTORIA PRÓPRIA)

Deve-se ter em mente que a hospedagem, por mais que seja considerada anual, é paga mensalmente. As pesquisas foram baseadas na melhor opção para o grupo e não no melhor preço.

Embora não haja o investimento real de R\$ 32.044,70, o grupo tem intenção de manter o projeto *online* por um tempo ainda indeterminado, entretanto todo o código e documentação estarão disponíveis online via “*GNU General Public License v3.0*” acessível pelo link <https://github.com/GabrielCoelho/TechTrain>.

A receita foi feita a partir de cálculos de divulgadores de propaganda, como E-Grana, Google AdSense e similares. O valor estimado é de R\$ 30.000,00 por 250 mil visitas no site, considerando o clique nas propagandas. É de comum entendimento que para um site chegar à esse patamar, leva-se um certo tempo. Foi estimado cerca de um ano e meio para que se chegue à esse patamar, com bons trabalhos de divulgação e marketing do site.

Conclui-se que o investimento seja, tanto através de patrocinadores, empréstimo-investimento bancário e afins, pouco a mais do custo, além de esperar o retorno há um curto prazo, sempre sendo jogado para cima de 1 ano e 6 meses.

2.8. LEIS E REGRAS

Por trata-se de um projeto público e divulgado para toda a comunidade, faz-se necessário o uso do mesmo dentro das leis e regras comuns, além de aplicar uma licença pública *OpenSource* (Código Aberto).

2.8.1. Quanto à Licença Pública

A *General Public License* dá permissão para uso comercial, distribuição, modificação, utilização da patente e uso privado. Para quem se interessar sobre o projeto e quiser utilizá-lo de alguma maneira, deverá cumprir as seguintes condições: Caso haja uma republicação do mesmo, o código deverá ser aberto como o original; quaisquer cópias deverão ser sob a mesma licença e conter no código do projeto o arquivo de licença com o devido *copyright* (direito de cópia); toda e qualquer alteração deverá ser documentada. Quanto às limitações desta licença

deve-se levar em conta que ela não prevê nenhuma garantia perante ao uso do código.

2.8.2. Quanto às Regras Trabalhistas

Não há intenção alguma do grupo, enquanto projeto de conclusão de curso, de contratar profissionais para criar aulas e libera-las para que seja incluído no projeto. Após a conclusão do curso, se o grupo decidir contratar, deverá ser feito através de contrato como previsto pelas leis trabalhistas do país. No contrato, tanto contratado como contratante, deverão ter suas vias assinadas e faz-se necessário que ambas as partes estejam de comum acordo para com as normas do contrato.

É necessário deixar claro que não haverá nenhum vínculo empregatício com os criadores de conteúdo de vídeos retirados das plataformas grátis.

2.8.3. Quanto ao uso de vídeos das plataformas

A única regra que implica na implantação do projeto é a questão dos direitos autorais. Caso o criador de conteúdo não queira que seus vídeos sejam transmitidos no projeto, o site terá uma seção específica para retirada de vídeos, bem como a de inserção dos mesmos.

2.8.4. Quanto à manutenção, segurança e estabilidade

Deve-se sempre manter atento às necessidades da sigla CIDA dentro da segurança de um sistema. A sigla tem como significado: Confidencialidade, Integridade, Disponibilidade e Autenticidade. Ou seja, a troca de informações entre o usuário e o sistema devem ser confidenciais, serem integras, acima de tudo, na internet, elas têm a necessidade de estarem disponíveis, e ser um conteúdo autêntico do site para com o usuário.

Assim, no *login* o usuário tem de se sentir seguro com seus dados para com o site, tratando-se da Confidencialidade; Saber que todo conteúdo ali disponível não é fraudoso, entrando no âmbito da Integridade; O site teria de estar sempre online,

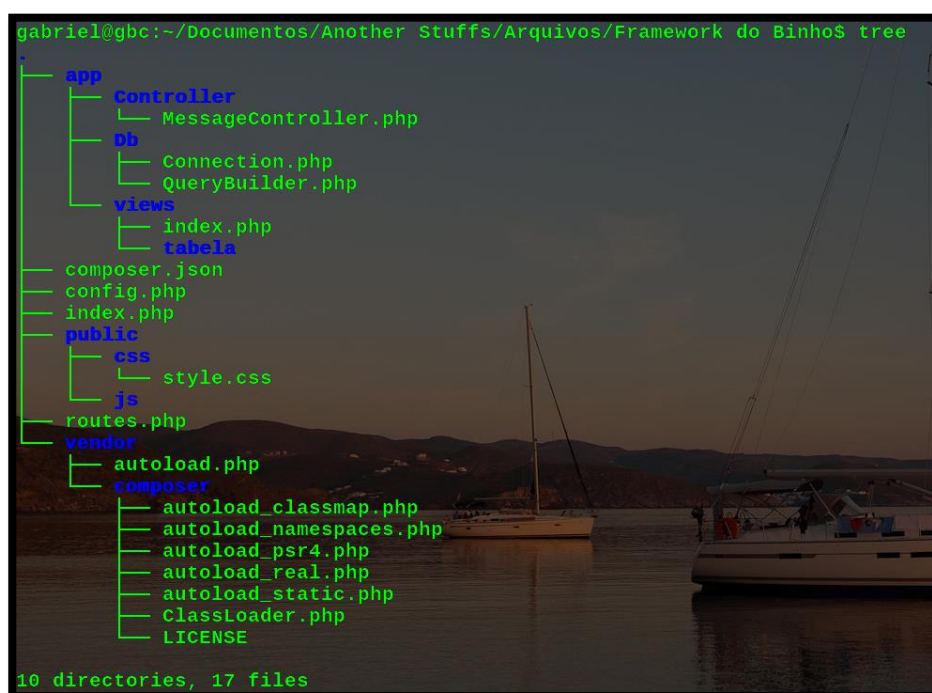
para que esteja disponível sempre que um usuário desejar; e ser autêntico para com a emissão de certificados válidos.

3. DESENVOLVIMENTO

Este capítulo abrangerá toda a documentação dos códigos do projeto e explicação básica sobre o código. Deve-se ter em mente que haverá algumas reutilizações de um mesmo código, sendo este explicado apenas uma vez e citado nas outras. Note que todos os arquivos estarão no repositório online livres para uso e visualização no link: <https://github.com/GabrielCoelho/TechTrain>

3.1 FRAMEWORK “DO-BINHO”

Figura 12: Tree view of the *Framework*



Fonte:(ARQUIVO PESSOAL)

Na figura 12 pode-se ver como é composto o *framework* que foi utilizado e apelidado de “*Framework do Binho*” por se tratar de um *framework* criado pelo professor João Rubens Marchete Filho e disponibilizado através do link <https://github.com/GabrielCoelho/fw-do-Binho>

Dentro do diretório APP existem três pastas, cada uma direcionada a um polo no desenvolvimento do projeto. Além delas, existem mais duas pastas, a *public* e a *vendor*. Por fim, existem os arquivos na pasta raiz do projeto.

3.1.1. APP/Controller

Neste diretório é que serão criadas as classes (orientação a objetos) para controlar, como o nome diz, toda a aplicação. Isto é, toda a programação, toda a lógica estará nesta pasta.

3.1.2. APP/Db

Este diretório é, inicialmente, composto de apenas dois arquivos. Estes são responsáveis por conectar ao banco de dados e executar comandos SQL para busca, inserção e remoção de dados.

3.1.3. APP/Views

Os arquivos PHP que terão conteúdo HTML deverão estar presentes neste diretório. Todo o design deverá estar nesta página, por mais que haja conteúdo de lógica em algumas páginas (que se faz necessário em qualquer projeto desenvolvido em PHP). Não há limitações quanto a criação de novos diretórios dentro deste, até é encorajado para que isso aconteça.

3.1.4. Public

Neste diretório é que haverá as folhas de estilo em cascata (CSS), conteúdos Javascript, imagens e afins, ou seja, tudo que for de interesse público e que possa vir, ou não, a aparecer nas páginas deverão estar nesta pasta. Assim como no diretório views, há um encorajamento para que novas pastas sejam criadas dentro desta.

3.1.5. Vendor

Este diretório é criado a partir do Composer, que é uma ferramenta de gerenciamento de pacotes próprio para o PHP. Pelo fato do *framework* não ser

criado pelo grupo, é preferível que se tenha um conhecimento alto de PHP para que seja alterado qualquer conteúdo desta pasta.

3.1.6. Arquivos soltos no diretório raiz

Parte importante do *framework*, os arquivos PHP que estão na raiz do projeto devem ser alterados apenas em conteúdo, acrescentando e decrementando o que for necessário. No arquivo de rotas pode-se alterar, criar ou excluir uma rota e deixar explícito o que fazer em cada uma das criadas. No arquivo de configuração deverá conter a criação de objetos dos controladores, além de configurar na variável `$_ENV` o nome do banco de dados, a localização host dele, senha e usuário para o mesmo.

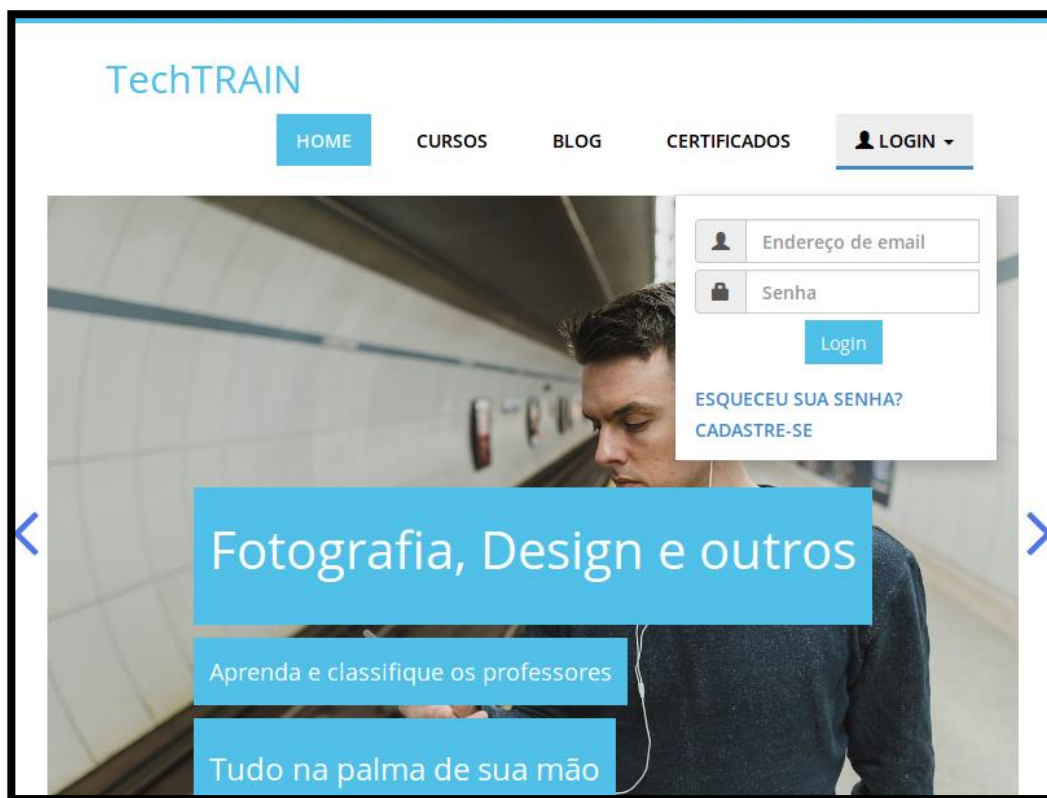
3.2. PÁGINA INICIAL

Para a criação da página inicial foi utilizado o *framework* Bootstrap para o CSS e alguns *snippets* (trechos de código) encontrados na internet de acesso e uso livres, como exemplo, o *Login dropdown* (menu suspenso) e o slider responsivo.

No início do desenvolvimento não foi pensado em nenhuma programação para a página inicial além do formulário de login, que será passado o nome de usuário e senha e, através do SQL será consultado e verificado se há um login idêntico já cadastrado. Sendo o retorno deste verdadeiro, no lugar do menu suspenso de login, haverá outro menu suspenso para mostrar as configurações do usuário.

Na figura 13, a visualização básica da página inicial, com o menu suspenso aberto. Vale como nota que o slider tem 3 campos, “passeando” por eles automaticamente ou através do controle do slider. Na figura, por se tratar de um *screenshot* (captura de tela), apenas uma imagem do slider pode ser vista.

Figura 13: ./app/views/index.php



Fonte: (arquivo pessoal)

3.2.1 Código da Página inicial

Como documentado anteriormente, na página inicial onde há um *slider*, um menu suspenso além de alguns códigos de JavaScript para dar beleza ao site. Dentro destes últimos, pode citar a forma como alguns componentes surgem na tela para o usuário. Assim que chegam a determinada “altura” do site, algumas *divs* (tags para divisão de seções no HTML) aparecem automaticamente. Toda ação documentada acima, pode ser vista na figura 14, onde há o código da página inicial.

Note que no projeto há um diretório dentro de *views* chamado de INC (./app/views/inc) onde existem trechos de códigos HTML repetidos diversas vezes em cada página. Estes são o *Header* e o *Footer*, para isso foi feito que eles carregassem automaticamente em cada início e final de rota respectivamente.

Figure 14: index.php codes

```

GNU nano 2.5.3                               Arquivo: app/views/index.php
#
<!--><start:main-->
<div class="slider">
  <div class="container">
    <div class="row">
      <div class="responsive-slider" data-orig="responsive-slider" data-autoplay="true">
        <div class="slides" data-group="slides">
          <ul>
            <li>
              <div class="slide-body" data-group="slide">
                
                <div class="caption header" data-animate="slideAppearUpToDown" data-delay="500" data-length="300">
                  <button class="btn btn-primary">
                    <h2>Você guia ao seu sucesso</h2>
                  </button>
                  <div class="caption-sub" data-animate="slideAppearDownToUp" data-delay="1200" data-length="300">
                    <button class="btn btn-primary">
                      <h4>
                        <span>Reconheça em um só lugar</span>
                      </h4>
                    </button>
                  </div>
                  <div class="caption-sub" data-animate="slideAppearLeftToRight" data-delay="900" data-length="300">
                    <button class="btn btn-primary">
                      <h3>Encontre Facilmente</h3>
                    </button>
                  </div>
                </div>
              </li>
            <li>
              <div class="slide-body" data-group="slide">
                
                <div class="caption header" data-animate="slideAppearDownToUp" data-delay="500" data-length="300">

```

Fonte (Arquivo Pessoal)

3.3. PÁGINA DE CADASTRO

Para a página de cadastro foi utilizada as propriedades básicas do bootstrap própria para formulário. Assim, foi decidido colocar o formulário como parte central e, abaixo dele, alguns motivos para se cadastrar no site. Para o cadastro, inicialmente, foi elaborado algo simples, porém necessários, como pode ser observado na figura 15. São estes o nome completo, e-mail e criação de uma senha.

Figura 15: Cadastro

Fonte (Arquivo Pessoal)

3.3.1 Estrutura da Página de Cadastro

Como pode ser observado na Figura 16, a estrutura da página de cadastro foi feita com um formulário simples, ainda podendo ser executado o cadastro a partir de 3 diferentes fontes. São elas o Github®, o Facebook® e o Twitter®.

Figura 16: Estrutura da página Cadastro

```

<button class="btn-primary"><h3>Cadastre-se</h3></button>
</div>

<?= $flash? ' <p class="alert alert-danger">' . $flash . '</p>' : ''?>

<form role="form" method="post" action="/post-register">
  <div class="form-group">
    <input type="text" class="form-control" name="fullName" placeholder="Nome Completo" />
  </div>

  <div class="form-group">
    <input type="date" class="form-control" name="dtNasc" placeholder="Data de nascimento">
  </div>

  <div class="form-group">
    <input type="email" class="form-control" name="email" placeholder="Email">
  </div>

  <div class="form-group">
    <input type="text" class="form-control" name="userName" placeholder="Nome de Usuário">
  </div>

  <div class="form-group">
    <input type="password" class="form-control" name="password" placeholder="Senha">
  </div>

  <div class="form-group">
    <input type="password" class="form-control" name="rPassword" placeholder="Repita a Senha">
  </div>

  <h2 class="cadastro-form">Ou...</h2>

  <p>
    <ul class="nav nav-tabs">
      <li role="presentation" class="cert"><a class="btn btn-info"><i class="fa fa-github fa-2x"
    </i> Cadastre-se com Github</a></li>
      <li role="presentation" class=""><a class="btn btn-info"><i class="fa fa-facebook fa-2x"><
    </i> Cadastre-se com Facebook</a></li>
      <li role="presentation" class="login"><a class="btn btn-info"><i class="fa fa-twitter fa-2
    </i> Cadastre-se com Twitter</a></li>
    </ul>
  </p>

  <button type="submit" class="btn btn-default btn-primary">Submit</button>
</form>

```

Fonte: (Arquivo Pessoal)

3.3.2 Código do Cadastro

Para que seja executado o cadastro e salvar os dados no banco de dados, cada um dos componentes do formulário, acima descrito, foi utilizado, principalmente o campo *Password* que verifica as duas senhas digitadas e só confirma o cadastro quando ambas são idênticas.

Assim, pode ser armazenado em variáveis no PHP as informações obtidas através do formulário além de criptografar a senha do usuário para maior segurança. Sendo isso tudo feito, há a necessidade, a partir do *framework* PHP utilizado, de instanciar um objeto da classe *QueryBuilder*. Chamado de “q”, este objeto irá inserir todos os dados no banco de dados autorizando, futuramente, o usuário a executar o login no site, sendo observado na Figura 17.

Figura 17 - função postRegister

```

21 // esse método recebe os dados para registrar um usuário
22 public function postRegister()
23 {
24     //recebe os dados de email e senha
25     $dados['nomeUsuario'] = htmlentities($_POST['fullName'], ENT_QUOTES);
26     $dados['dataUsuario'] = htmlentities($_POST['dtNasc'], ENT_QUOTES);
27     $dados['emailloginUsuario'] = htmlentities($_POST['email'], ENT_QUOTES);
28     $dados['loginUsuario'] = htmlentities($_POST['userName'], ENT_QUOTES);
29     $dados['senhaUsuario'] = htmlentities($_POST['password'], ENT_QUOTES);
30     $rPassword = htmlentities($_POST['rPassword'], ENT_QUOTES);
31
32
33
34     //compara os dois campos de senha, devolvendo uma mensagem flash caso sejam diferentes
35     if ($dados['senhaUsuario'] != $rPassword) {
36
37         Flash::setFlash('As senhas não conferem');
38         header('Location: /register');
39         exit;
40     }
41
42     // criptografa a senha para guardar no banco de dados.
43     // a sequência que passei é bem fraca, mas é um exemplo de salt
44     $dados['senhaUsuario'] = crypt($dados['senhaUsuario'], '123456mad6991ef');
45
46     $q = new QueryBuilder();
47     $cadastrado = $q->insert('usuario', $dados);
48
49     // se não foi possível realizar o cadastro, como por exemplo, email repetido
50     // dispara um mensagem flash
51     if (!$cadastrado) {
52         Flash::setFlash('Dados inválidos');
53         header('Location: /register');
54         exit;
55     }
56
57     // guarda o email do usuário na session
58     $_SESSION['user'] = $dados['loginUsuario'];
59
60     //chama o método de configuração inicial do portal
61     //header('Location: /');
62     header('Location: /');
63
64 }

```

Fonte: (Arquivo Pessoal)

3.3.3. Função *Login* e *Logout*

A função *login* somente pode ser executada depois de um cadastro, assim, ela seleciona, à partir de um objeto “q”, os dados no banco de dados correspondentes aos do *login* e senha inseridos pelo usuário. Caso estes já estejam cadastrados, o sistema redireciona o usuário para a página de Cursos para que o mesmo possa ver os cursos recém adicionados e as categorias mais pertinentes. Caso não seja autorizado o *login*, o sistema retorna a informação de que os dados não são válidos. Toda a função descrita acima, pode ser observada na Figura 18.

Figura 18 - Função Login

```

66 //metodo para realizar o login do usuário
67 public function login()
68 {
69     $dados['emailloginUsuario'] = htmlentities($_POST['loginEntrada'], ENT_QUOTES);
70     $dados['senhaUsuario'] = htmlentities($_POST['loginSenha'], ENT_QUOTES);
71
72     $dados['senhaUsuario'] = crypt($dados['senhaUsuario'], '123456mad6991ef');
73
74     $q = new QueryBuilder();
75
76
77     $usuario = $q->select('usuario', [
78         'emailloginUsuario' => $dados['emailloginUsuario'],
79         'senhaUsuario' => $dados['senhaUsuario']
80     ]);
81
82     // se o usuário não foi encontrado no banco de dados
83     // emite uma mensagem de erro
84     if (!$usuario) {
85         Flash::setFlash("Dados inválidos");
86         header('Location: /');
87         exit;
88     }
89
90     // autentica o usuário
91     // $consulta = $q->select('usuario', [
92         //     'nomeUsuario'
93     // ]);
94     // var_dump($usuario);
95     $_SESSION['user'] = $usuario[0]['nomeUsuario'];
96     $_SESSION['id'] = $usuario[0]['idUsuario'];
97     $_SESSION['senha'] = $usuario[0]['senhaUsuario'];
98     $_SESSION['logged'] = true;
99     header('Location: /courses');
100 }

```

Fonte: (Arquivo Pessoal)

Já a função *logout* é bem mais simples. A função apenas apaga os dados que estão na Sessão atual com o comando *session_unset()* e retorna o usuário para a página inicial, como pode ser visto na Figura 19

Figura 19 - função *logout*

```

182     public function logout()
183     {
184         //remove todas variáveis criadas de sessão
185         session_unset();
186
187         //devolve para a página inicial
188         header('Location: /');
189     }

```

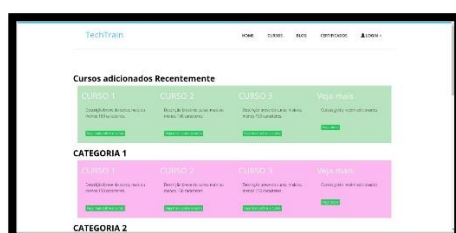
Fonte: (Arquivo Pessoal)

3.4 . A ROTA CURSOS

Para a rota de cursos, foi realizada a criação de uma página de visualização com uma pré-visualização de como deverá aparecer ao usuário os cursos listados no banco de dados. Assim, pode ser observado na figura 20, a disposição dos cursos.

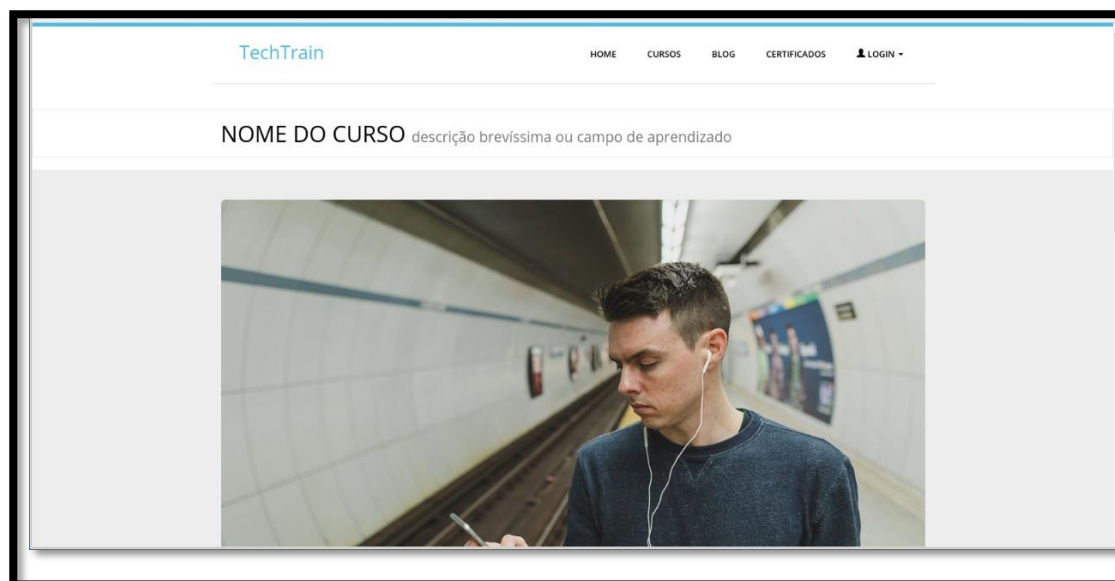
Assim que o usuário decidir o curso que deseja se inscrever, ele poderá clicar no curso em questão para que possa ver mais informações sobre o mesmo, tais como o nome do instrutor, uma ementa básica do curso, cursos semelhantes também oferecidos pela plataforma, além da quantidade de horas que a pessoa passará, aproximadamente, executando o curso. Esta página pode ser vista na figura 21, que é a rota “/coursePage”.

Figura 20 - Página de Cursos



Fonte: (Arquivo Pessoal)

Figura 21 - Página de cada Curso



Fonte: (Arquivo Pessoal)

3.4.1 Códigos das páginas de Curso

Para que as rotas possam selecionar os cursos do banco de dados a partir da *query* executada, precisa-se de alguns arquivos PHP para que seja feita a integração. É necessário notar que, cada rota ou conjunto de rotas semelhantes, existe um controlador (*Controller*) para tal rota ou grupo de rotas. Assim, para a rota de cursos, temos o arquivo "CourseContorller.php" que fará o controle total de todos os componentes vistos na rota de cursos, bem como chamar as páginas de visualização.

Nas figuras 22, 23 e 24, pode-se observar trechos do código da página de visualização das páginas de todos os cursos e de cada curso. Na figura 25, temos os códigos do controlador que irá retornar os valores para a visualização. Deve-se ter como nota que as imagens foram tiradas enquanto o projeto estava em produção.

Figura 22 - cursos.php código

```

<div class="container">
  <div class="row">
    <h2 style="font-weight: bold; color: black;">Cursos adicionados Recentemente</h2>
  </div>
  <div class="row course_1">
    <div class="col-sm-3 col-xs-12">
      <h2>CURSO 1</h2>
      <!-- Exemplo de imagem Descrição breve do curso, mais ou menos 150 caracteres.</p>
      <p><a href="#" class="btn btn-success btn-xs">Veja mais sobre o curso</a></p> <!--
    </div>

    <div class="col-sm-3 col-xs-12">
      <h2>CURSO 2</h2>
      <!-- Aqui vem uma imagem. Precisa de um campo no banco de dados para adicionar a
      <p>Descrição breve do curso, mais ou menos 150 caracteres.</p>
      <p><a href="#" class="btn btn-success btn-xs">Veja mais sobre o curso</a></p> <!--
    </div>

    <div class="col-sm-3 col-xs-12">
      <h2>CURSO 3</h2>
      <!-- Aqui vem uma imagem. Precisa de um campo no banco de dados para adicionar a
      <p>Descrição breve do curso, mais ou menos 150 caracteres.</p>
      <p><a href="#" class="btn btn-success btn-xs">Veja mais sobre o curso</a></p> <!--
    </div>

    <div class="col-sm-3 col-xs-12">
      <h2>Veja mais</h2>
      <p>Cursos grátis recém adicionados</p>
      <p><a href="#" class="btn btn-success btn-xs">Veja todos</a></p> <!-- Aqui é out
    </div>
  </div>
</div>

```

Fonte: (Arquivo Pessoal).

Figura 23 - coursePage.php código - parte 1.

```

<div class="page-header">
  <div class="container">
    <h1 style="color: black;">NOME DO CURSO <small>descrição brevíssima ou campo de aprendizado</small></h1>
  </div>
</div>
<div class="jumbotron">
  <div class="container">
    <div class="col-xs-12">
      
      <!-- Aqui teremos a imagem do curso de tamanho maior que na rota /courses -->
    </div>
  </div>
  <div class="row margeada">
    <p>
      conteúdo explicativo do curso <br>
      Lorem ipsum dolor sit amet,
      consectetur adipiscing elit,
      sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua. <br>
      Ut enim ad minim veniam,
      quis nostrud exercitation
      ullamco laboris nisi ut
      aliquip ex ea commodo consequat. <br>
      Duis aute irure dolor in
      reprehenderit in voluptate
      velit esse cillum dolore eu fugiat nulla pariatur.
      <!-- Aqui um conteúdo simples, apenas para explicar o que é e como é o curso -->
    </p>
  </div>

```

Fonte: (Arquivo Pessoal)

Figura 24 - courePage.php código - parte 2

```
<aside class="signupCourse course_5">
  <div class="col-xs-12">
    
  </div>
  <hr class="style18">
  <ul>
    <li><strong>Nome: </strong> Astolfo Bernardino Camiães</li>
    <li><strong>Canal: </strong> <a href="#">ABC da Programação</a></li>
    <li><strong>Duração do Curso: </strong> 48Mil Horas</li>
    <li><strong>Certificado ao final: </strong> Não</li>
  </ul>
  <hr class="style18">
  <button type="button" class="btn btn-success btn-lg btn-block" name="button">Inscreva-se nesse curso!</button>
</aside>

<div class="row margeada">
  <h2 style="color: rgb(26, 164, 210); margin-left: 10px;">Veja os Vídeos aqui ou no youtube!</h2>
  <p>
    O instrutor ganhará visualização por qualquer uma das maneiras! Lembre-se
    sempre que <strong>o importante é você aprender!</strong>
  </p>
</div>

<div class="row margeada">
  <h2 style="color: black; margin-left: 10px;">Você também pode gostar de: </h2>
  <ul>
    <li>Outros cursos</li>
    <li>Outros cursos</li>
    <li>Outros cursos</li>
  </ul>
</div>
```

Fonte: (Arquivo Pessoal)

Figura 25 - CourseController.php códigos

```
class CourseController
{
    public function searchCourses()
    {
        $q = new QueryBuilder();
        $array = $q->select("curso");
        // print_r($array);
        // die();
        require './app/views/cursos.php';
    }

    public function courseClick()
    {
        require './app/views/coursePage.php';
    }

    public function watchVideo()
    {
        require './app/views/videoPage.php';
    }
}
```

Fonte: (Arquivo Pessoal)

3.5 PÁGINA DE VÍDEO

Para a página de vídeo, por tratar-se de vídeos do curso, foi utilizado o mesmo controlador da rota Cursos. Sendo assim, o controlador fará uma busca pelo id do Curso selecionado e retornará com o link do vídeo proposto. Assim, nas figuras 26 e 27, observa-se a página de vídeo bem como o código da mesma, respectivamente.

Figura 26 - Página de Vídeo



Fonte: (Arquivo Pessoal)

Figura 27 - videoPage.php códigos

```
<div class="container">
  <div class="row">
    <div class="page-header">
      <div class="container">
        <h1 style="color: black;">NOME DO VÍDEO <small>Nome do Curso</small></h1>
      </div>
    </div>
    <div class="video-container">
      <!-- aqui vem o iframe do vídeo -> aqui que incluiremos o php -->
      <iframe src="https://www.youtube.com/embed/kmT5mVBF2tI" frameborder="0" gesture="media"
    </div>
  </div>
  <div class="row">
    <p>
      conteúdo explicativo do vídeo <br> <!-- Pode ser o que está na descrição no YT -->
      Lorem ipsum dolor sit amet,
      consectetur adipiscing elit,
      sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua. <br>
      Ut enim ad minim veniam,
      quis nostrud exercitation
      ullamco laboris nisi ut
      aliquip ex ea commodo consequat. <br>
      Duis aute irure dolor in
      reprehenderit in voluptate
      velit esse cillum dolore eu fugiat nulla pariatur.
    </p>
  </div>
</div>
```

Fonte: (Arquivo Pessoal)

CONSIDERAÇÕES FINAIS

Mesmo que o grupo não tenha atingido cem por cento da visão inicial do projeto, todos ficaram satisfeitos com o andamento e com a versão final entregue. Ainda assim, todo o grupo consentiu em que a visão inicial era demasiadamente grande e necessitava de um conhecimento maior do que o adquirido durante as aulas do curso.

Todas as metas não atingidas, ficam como projetos futuros, tais como a secção de comentários e avaliação do curso e dos professores, o blog para a equipe ou empresa que decidir colocar o site online e, principalmente, a emissão de certificados.

Mesmo com a satisfação geral do grupo, alguns não têm a intenção de manter o projeto ativo. Seja pela dificuldade, pela área do curso não agradar totalmente ou simplesmente por falta de tempo. Mesmo assim, os que têm vontade de manter o projeto atualizado, o farão através do repositório online previamente documentado.

REFERÊNCIAS

Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa. Disponível na internet via <http://historiadaslinguagensdeprogramacao.blogspot.com.br/>. Arquivo capturado em 14 de Abril. 2017

GOLDBERG, Robert P.. Virtualização um pouco de história. {online}. Disponível na Internet via <https://hbueno.wordpress.com/2009/04/29/virtualizacao-um-pouco-de-historia/>. Arquivo capturado em 29 de abri 2017.

GUIMARÃES, Gleyser. As primeiras linguagens de programação {online}. Disponível na internet via http://www.dsc.ufcg.edu.br/~pet/jornal/novembro2012/materias/historia_da_computacao.html. Arquivo capturado em 25 de Julho. 2017

KARAIVANOV, Dimitar. Github to Kanbanize Integration {online}. Disponível na internet via <https://kanbanize.com/blog/kanbanize-github-integration/>. Arquivo capturado em 16 de Novembro. 2017

MORAES, Paulo Sergio de. Logica de Programção. {online}. Disponível na Internet via <http://www.inf.ufsc.br/~vania.bogorny/teaching/ine5231/Logica.pdf>. Arquivo capturado em 14 de abri 2017.

Matera Systems. O que é CSS e qual sua importância {online}. Disponível na internet via <http://www.matera.com/br/2012/07/25/o-que-e-css-e-qual-sua-importancia/>. Arquivo capturado em 25 de julho 2017.

OK CONCURSOS. Linguagem de programação {online}. Disponível na internet via <http://www.okconcursos.com.br/apostilas/apostila-gratis/130-informatica-para-concursos/1689-linguagem-de-programacao#.WcMJt7J96M9>. Arquivo capturado em 14 de Abril. 2017

PACIEVITCH, Yuri. PHP {online}. Disponível na Internet via <http://www.infoescola.com/informatica/php/>. Arquivo capturado em 14 Abril. 2017

PHP Group. Informações Gerais {online}. Disponível na internet via http://php.net/manual/pt_BR/faq.general.php. Arquivo capturado em 14 Abril. 2017

PHP Group. O que o PHP pode fazer? {online}. Disponível na internet via https://secure.php.net/manual/pt_BR/intro-whatcando.php. Arquivo capturado em 14 Abril. 2017

POZZEBOM, Rafaela. O que é Github? {online}. Disponível na internet via <https://www.oficinadanet.com.br/post/14791-o-que-github>. Arquivo capturado em 16 de Novembro. 2017

REDAÇÃO OFICINA. O que é um servidor web. {online}. Disponível na Internet via https://www.oficinadanet.com.br/artigo/servidores/o_que_e_um_servidor_web. Arquivo capturado em 14 de abril. 2017

ROGÉRIO, Pedro. Quais são as versões do html. {online}. Disponível na Internet via <http://www.pinceladasdawe.com.br/blog/2007/10/15/quais-sao-as-versoes-do-html/>. Arquivo capturado em 15 de out. 2017

SIMÕES, Pedro. Visual Studio Code: O novo editor da Microsoft para Linux e Mac {online}. Disponível na internet via <https://pplware.sapo.pt/linux/visual-studio-code-o-novo-editor-da-microsoft-para-linux-e-mac/>. Arquivo capturado em 16 de Novembro. 2017