

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

THALITA ALVES PEREIRA

***WEBSITE* PARA ENSINAR PROGRAMAÇÃO UTILIZANDO JOGOS**
BASEADOS NO GÊNERO RPG

BAURU - SP

2017

THALITA ALVES PEREIRA

**WEBSITE PARA ENSINAR PROGRAMAÇÃO UTILIZANDO JOGOS
BASEADOS NO GÊNERO RPG**

Trabalho de Conclusão de Curso
apresentado à Universidade Estadual
Paulista "Júlio de Mesquita Filho",
Faculdade de Ciências, Campus
Bauru, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.

Orientador: Professor Doutor Marco
Antônio Rahal Sacoman

BAURU - SP

2017

Pereira, Thalita Alves

website para ensinar programação utilizando jogos baseados no gênero RPG – Thalita Alves Pereira, Bauru, 2017.

50 p. il.

Orientador: Marco Antônio Rahal Sacoman.

Trabalho de conclusão de curso (Graduação) -- Universidade Estadual Paulista, Faculdade de Ciências, Bacharelado em Ciência da Computação, Bauru, BR-SP, 2017.

1.Ensino de programação 2.Jogos educativos 3.Programação para web

THALITA ALVES PEREIRA

**WEBSITE PARA ENSINAR PROGRAMAÇÃO UTILIZANDO JOGOS
BASEADOS NO GÊNERO RPG**

Trabalho de Conclusão de Curso
apresentado à Universidade Estadual
Paulista "Júlio de Mesquita Filho",
Faculdade de Ciências, Campus Bauru,
como requisito parcial para obtenção
do título de Bacharel em Ciência da
Computação.

Orientador: Professor Doutor Marco
Antônio Rahal Sacoman

Banca Examinadora

Professor Doutor Marco Antônio. Rahal Sacoman (Orientador)

Departamento de Computação - FC - UNESP

Professora Doutora Simone das Graças Domingues Prado

Departamento de Computação - FC - UNESP

Professor Doutor José Remo Ferreira Brega

Departamento de Computação - FC – UNESP

Bauru, 12 de dezembro de 2017

AGRADECIMENTOS

Agradeço e dedico este trabalho a todos que direta ou indiretamente contribuíram para minha formação acadêmica.

Agradeço à minha família pelo apoio emocional e moral, sempre estando ao meu lado quando tudo parecia ficar mais difícil.

Agradeço aos meus professores que ensinaram muito do que posso aplicar não só neste, mas em futuros trabalhos.

Um agradecimento especial ao meu orientador que esteve sempre à disposição para tirar todas as minhas dúvidas e me deu muitas dicas para este trabalho.

Por último, mas não menos importante, obrigada a todos os amigos que me apoiaram durante essa trajetória na faculdade e me ajudaram muito a aprender mais também, não só sobre as matérias, mas sobre a vida em si.

RESUMO

Com o avanço da tecnologia surgem cada vez mais programas para facilitar a vida do ser humano. Mas toda essa tecnologia requer conhecimento para produzi-la, e neste ponto o Brasil é um país ainda engatinhando no processo de educação já que não faz parte obrigatoriamente do currículo de ensino básico o ensino de programação e lógica. Sendo assim, muitos jovens não têm ideia de como se fazem programas. Como uma tentativa de diminuir a quantidade de leigos no assunto, este trabalho propõe a criação de um *website* em que, através de pequenos jogos com temática baseada no gênero RPG (*Role Playing Game*) procura ensinar conceitos básicos de programação. Para tal serão utilizadas linguagens e ferramentas de desenvolvimento voltadas à Web somadas a um breve estudo sobre jogos educativos e como eles auxiliam o aprendizado, propondo criar um ambiente de imersão em que o jovem como jogador se interesse pelo estudo da programação e esteja preparado para o ensino superior ou para uma carreira na área da computação caso deseje buscar mais conhecimento nesta área.

PALAVRAS-CHAVE: ensino de programação, jogos educativos, programação para web.

ABSTRACT

With the advancement of technology, more and more programs are appearing to make life easier for the human being. But all this technology requires knowledge to produce it, and at this point Brazil is a country still crawling in the process of education since it is not compulsory part of the basic education curriculum teaching programming and logic. So many young people have no idea how to do programs. As an attempt to reduce the number of lay people in the subject, this work proposes the creation of a *website* in which, through small games based on the RPG genre (Role Playing Game), it tries to teach basic programming concepts. This will be used web-based development languages and tools coupled with a brief study on educational games and how they aid learning, proposing to create an immersion environment in which the young person as a player is interested in studying programming and is prepared for the higher education or for a career in the field of computing if it wants to seek more knowledge in this area.

KEYWORDS: teaching programming, educational games, web programming.

LISTA DE ABREVIATURAS E SIGLAS

PC	Personal Computer
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
SQL	Structured Query Language
RPG	Role Playing Game
FPS	First Person Shoot
TPS	Third Person Shoot
MOBA	Multiplayer Online Battle Arena
MMO	Massively Multiplayer Online
NPC	Non-Player Character

LISTA DE FIGURAS

Figura 1 - Documento em HTML puro	23
Figura 2 - Diferença de HTML sem e com CSS aplicado	23
Figura 3 - Parágrafos estilizados	24
Figura 4 - Exemplos feitos com Bootstrap	26
Figura 5 - Modelo Entidade Relacionamento	29
Figura 6 - Protótipo de interface com usuário	30
Figura 7 - Layout básico das páginas de conteúdo	32
Figura 8 - Página inicial (descrição)	32
Figura 9 - Página inicial (cadastro e log in)	33
Figura 10 - Tela do Animate	34
Figura 11 - Tela de jogo com diálogo	35
Figura 12 - Tela de jogo com receita	35
Figura 13 - Tela de jogo com pseudocódigo	36
Figura 14 - Trecho de código PHP da página de log in	38
Figura 15 - Código e imagem com informações de um usuário extraído do banco de dados	39

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.1.1	<i>Objetivo Geral</i>.....	14
1.1.2	<i>Objetivos Específicos</i>	14
1.2	Organização da Monografia	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	<i>Websites</i>.....	16
2.2	Role Playing Game	17
2.3	Imersão de jogadores	17
2.4	Jogos digitais	18
2.5	Jogos educativos	19
2.6	Jogos educativos e programação.....	19
2.7	Servidores Web	20
2.8	Programação e algoritmos	20
3	METODOLOGIA	22
3.1	Ferramentas	22
3.1.1	<i>HTML</i>	22
3.1.2	<i>CSS</i>.....	23
3.1.3	<i>Javascript</i>.....	24
3.1.4	<i>Bootstrap</i>	25
3.1.5	<i>jQuery</i>.....	26
3.1.6	<i>CreateJS</i>.....	26
3.1.7	<i>PHP</i>	27
3.1.8	<i>SQL</i>	27

3.1.8.1	<i>MySQL</i>	28
3.1.9	<i>Apache</i>	28
3.2	Métodos	29
4	DESENVOLVIMENTO	31
4.1	Front End	31
4.2	Animação do jogo	33
4.3	Back End	37
4.3.1	<i>Sistema de cadastro e log in</i>	37
4.3.2	<i>Banco de dados</i>	38
5	CONCLUSÃO	40
5.1	Trabalhos Futuros	40
	REFERÊNCIAS	42
	BIBLIOGRAFIA COMPLEMENTAR	46
	APÊNDICE A – ROTEIRO DA ANIMAÇÃO "TUTORIAL"	47

1 INTRODUÇÃO

Apesar de ser uma ciência considerada recente com relação à história da humanidade, a computação já está presente na maior parte do dia a dia de todas as pessoas, seja nos eletrodomésticos, nos caixas eletrônicos ou até nos grandes datacenters que armazenam os milhares de *websites* que existem na internet. Entretanto o que é necessário para se criar toda essa tecnologia é ainda pouco difundido entre os mais jovens: o conhecimento da programação e como ela é aplicada.

Existem *websites* que ensinam programação, alguns gratuitos e alguns pagos, com o formato de aulas escritas, através de arquivos texto e pdf, e aulas em vídeo. Para exemplificar, tem-se os *websites* *Codecademy*, de origem norte-americana, e o *DevMedia*, que além de cursos possui uma área para quem busca um trabalho em diferentes ramos da computação (CODECADEMY, 2017; DEVMEDIA, 2017). Estas plataformas apresentam, como anteriormente citado, aulas de programação no formato de vídeos e texto, mas esse tipo de ensino não é muito atrativo a jovens que pretendem começar cedo a aprender programação já que o conteúdo é oferecido totalmente de uma vez ao aluno e ele deve ser autodidata, estudando rigorosamente por conta própria.

Este trabalho propõe uma visão que difere de uma sala de aula, transformando o aprendizado da programação em algo divertido para quem está iniciando. Para isso será usado como base outros *websites* que ensinam programação através de jogos, como *CodeCombat*, que utiliza a programação para que o personagem realize ações e vença os desafios propostos, *CodeMonkey*, onde o objetivo é fazer um macaco coletar todas as bananas desviando de obstáculos, dentre outros. (CHAPMAN, 2017)

Foi construído um *website* que possui, como temática, jogos do estilo RPG (Role Playing Game, jogo de interpretação de personagens), que consistem em jogadores que encarnam personagens para participar de aventuras em um universo pré-designado pelo criador do jogo, com uma interface apresentando os itens do menu, os jogos nomeados como missões, o

personagem que representa o jogador e outras informações características destes jogos, como experiência do jogador e moedas adquiridas nas tais aventuras.

Programar não é uma tarefa tão simples, requer certo raciocínio lógico e o conhecimento gramatical de uma linguagem de programação. Para um aluno de ensino técnico ou superior na área da computação aprender e aplicar isso tudo pode não ser uma tarefa tão árdua, porém para um leigo no assunto pode tornar-se um grande desafio.

Considerando que os mais jovens, antes de concluírem o ensino médio, raramente são apresentados ao universo da programação têm-se um público pequeno interessado na área se comparado a outras áreas de conhecimento como ciências humanas e biológicas, por exemplo, tendo, assim, mais indivíduos interessados em serem futuros médicos ou advogados do que desenvolvedores de softwares. A dificuldade de se aprender inicialmente uma linguagem de programação por conta própria somada à falta de interesse e capacidade das escolas em apresentar esse tipo de conteúdo em aulas gera essa pequena parcela de pessoas que detém o conhecimento de programar (MEDEIROS; SILVA; ARANHA, 2013).

Este trabalho se baseia nesta premissa para criar uma plataforma onde aprender programação não precisa ser uma tarefa tão dificultosa, apresentando conceitos básicos de maneira que a pessoa se torne um jogador e acabe por aprender sem perceber, já que estará entretido com o jogo.

1.1 Objetivos

O *website* pretende atingir o objetivo geral descrito a seguir, utilizando os procedimentos dados pelos objetivos específicos aliados a ferramentas de programação.

1.1.1 Objetivo Geral

Ensinar programação de uma maneira lúdica, fazendo com que as pessoas, principalmente os jovens que estão no ensino fundamental ou médio, aprendam enquanto jogam um jogo em estilo RPG.

1.1.2 Objetivos Específicos

Os objetivos específicos estão listados a seguir:

- Fazer a revisão bibliográfica.
- Revisitar programas similares.
- Descrever todo o processo de produção do *website*.
- Basear-se na descrição e anotações além da bibliografia para escrever a monografia.
- Roteirizar as histórias utilizadas como plano de fundo nos jogos.
- Programar o *website* (*front end* e *back end*) de modo a conter os requisitos seguintes.
- Criar sistema de autenticação de usuário.
- Desenvolver uma interface.
- Criar sistema de premiação por missões (que funcionam como "aulas") cumpridas.
- Criar registro de desempenho do jogador, baseado nas missões cumpridas, na forma de experiência do jogador.
- Criar os jogos em si baseados em jogos de raciocínio e lógica unidos a conceitos de programação.
- Encaixar os elementos planejados no roteiro junto aos jogos para formar as missões.

1.2 Organização da Monografia

Os capítulos subsequentes estão divididos conforme seus conteúdos descritos a seguir.

Capítulo 2 - FUNDAMENTAÇÃO TEÓRICA: Revisão bibliográfica com todos os conceitos necessários para entender o desenvolvimento da aplicação e sua finalidade.

Capítulo 3 - METODOLOGIA: Contém as linguagens de programação e as ferramentas descritas com detalhes além do método utilizado para planejar o desenvolvimento da aplicação.

Capítulo 4 - DESENVOLVIMENTO: Lista e explica todos os procedimentos para realizar o desenvolvimento de todo o *website*, separando as etapas por *Front end*, Animação e *Back end*.

Capítulo 5 - CONCLUSÃO: Tem por finalidade analisar o produto final do trabalho, escrever conclusões sobre os procedimentos realizados e sugerir melhorias futuras.

REFERÊNCIAS

BIBLIOGRAFIA COMPLEMENTAR

APÊNDICE A – ROTEIRO DA ANIMAÇÃO "TUTORIAL"

2 FUNDAMENTAÇÃO TEÓRICA

Desenvolver um *website* educativo requer não só conhecimento das linguagens a serem utilizadas na sua programação como também deve-se ter uma base teórica de alguns conceitos pertinentes ao assunto, de modo que se tenha a aptidão de visualizar o processo de criação como um todo e ter uma noção de como seria sua aplicação prática.

2.1 *Websites*

Com a internet, tornou-se necessário um mecanismo que pudesse divulgar através da rede mundial o conteúdo e as informações que fossem necessárias. Esse é o objetivo dos *websites*, aplicações da *web* que contém textos, imagens, vídeos e muito mais, utilizados como forma de divulgar conhecimento, conversar com pessoas que podem estar em outra cidade ou até outro país, realizar trocas e vendas, entre outras funções. "Desde a invenção da Web, a tecnologia para construção de sites vem sendo progressivamente incrementada permitindo o desenvolvimento de aplicações cada vez mais complexas". (WINCKLER; PIMENTA, 2002).

Partindo deste princípio, *websites* são uma importante ferramenta quando se trata de alcance, em outras palavras utilizando uma página na *web* há muito mais chances de atingir pessoas em locais mais distantes do que em uma mídia física como uma revista, por exemplo, por ser mais rápido bastando ter uma conexão com a internet e não depender de limitações geográficas, como ter de percorrer longas distâncias em um ou mais veículos de transporte para levar tal conteúdo a outras cidades, estados ou países.

2.2 Role Playing Game

O RPG, jogo de representação de personagens, surgiu inicialmente na forma do RPG de mesa, sendo um jogo em que é apresentado um mundo fictício, tendo como participantes um mestre que coordena este mundo e dita os passos do jogo e os jogadores que são representados por seres que habitam este lugar e, no decorrer do jogo, têm de tomar decisões que vão moldando o rumo da partida (ROSA, 2008). Esta imersão, sensação de fazer parte do jogo, é o que atrai muita gente para este gênero.

Dado o avanço da tecnologia o RPG foi deixando de ser apenas um jogo de tabuleiro e passou a ser um gênero muito popular de jogos digitais. Graças aos consoles que permitem cada vez mais armazenamento de memória e qualidade gráfica os jogos RPG vêm sempre garantindo seu espaço no mercado já que todo aquele ambiente fantástico pode ser visto com riqueza de detalhes e possibilita que cada jogador possa ter seu personagem muito próximo do que imaginava, através do alto nível de personalização. Esse gênero possui títulos tanto para *single player*, jogos feitos para um único jogador por vez, quanto para *multiplayer*, jogos onde mais de uma pessoa joga simultaneamente.

2.3 Imersão de jogadores

A imersão em um jogo consiste no jogador ter a impressão de fazer parte da história, podendo ser ele mesmo dentro do jogo, como acontece em alguns jogos de realidade virtual, por exemplo, ou ter um personagem que o representa, como é o caso do RPG. A imersão quando bem aplicada leva o jogador a se envolver profundamente com o jogo, provocando uma mudança de foco do mundo real para o mundo do jogo (SAVI et. al., 2010). Essa mudança de foco traz até mesmo uma distorção temporal, em que o jogador pode passar noites em claro jogando sem notar a passagem do tempo (TAKATALO et. al., 2010).

Este ato de sentir-se dentro de outro universo era explorado de maneira mais rasa nos cinemas e na televisão, uma imersão passiva, enquanto que nos jogos atuais é possível se sentir completamente neste novo ambiente, de forma mais palpável (FALCÃO; RIBEIRO, 2009). Isso é possível devido ao avanço das tecnologias que permite a criação de um mundo virtual cada vez mais próximo de algo que poderia se tornar real, com histórias bem construídas e com gráficos cada vez mais realistas.

Entretanto, para que haja a imersão, é necessário que o jogo apresente elementos que induzam o jogador a participar do mundo que ele apresenta. "Quanto mais bem resolvido o ambiente de imersão, mais ativo o jogador desejará ser dentro do game", afirmam Silva e Justino (2009). Ainda segundo estes autores, para se trazer resultados positivos da imersão é preciso despertar e alimentar o desejo da pessoa seguir adiante no jogo, tornando a experiência de descobrir algo novo prazerosa, levando-a a realizar ações que a coloquem em um patamar cada vez melhor.

2.4 Jogos digitais

O primeiro jogo criado para o meio digital foi feito por Alexander S. Douglas, em 1952 sendo uma versão do popular jogo da velha (CONTI, 2015). Desde então todas as faixas etárias têm se aproveitado cada vez mais da facilidade de ter vários jogos em um único aparelho.

Uma das plataformas com vários jogos digitais disponíveis é o PC (*Personal Computer*), podendo armazenar em sua memória múltiplos jogos, sendo que atualmente a maioria destes jogos utiliza a conexão com a internet para torná-los mais dinâmicos. Há vários gêneros dos chamados jogos MMO (*Massively Multiplayer Online*, multijogadores *online* massivamente) que utilizam a internet para conectar vários jogadores ao mesmo tempo para jogarem simultaneamente (FERREIRA, 2006). Alguns deles são FPS (*First Person Shoot*, tiro em primeira pessoa), TPS (*Third Person Shoot*, tiro em terceira pessoa), MOBA (*Multiplayer Online Battle Arena*, arena de batalhas com multijogadores *online*) e RPG.

Apesar da forma mais comum de disseminação desses jogos ser via *download* do conteúdo do jogo existem aqueles que executam direto nos navegadores, sem serem necessários arquivos extras na máquina do jogador. Esses jogos são executados diretamente nos *websites* e podem abranger os mesmos gêneros que os instaláveis, porém com limitações de recursos de processamento e gráficos visto que ficam armazenados no servidor.

2.5 Jogos educativos

Nem todos os jogos têm como objetivo apenas entreter quem os joga, mas também ensinam algum conceito ao jogador. Esses são os jogos educativos, utilizados como forma de somar na educação de crianças, jovens e até adultos oferecendo um conteúdo a ser aprendido pelo jogador em uma interface interativa. "Através dos jogos educacionais, são aperfeiçoados o reconhecimento e entendimento de regras" (PÁDUA, 2008).

Os jogos digitais no processo de ensino-aprendizagem podem ser "um rico instrumento para a construção do conhecimento, transformando o ato de jogar em ato de aprender e ensinar, construindo os objetivos necessários para se alcançar a aprendizagem", segundo Medeiros, Silva e Aranha (2013). Despertar o interesse dos jovens no ensino se torna ainda mais fácil quando eles estão interessados em continuar o processo de aprendizado até o fim, o que ocorre quando estão jogando.

2.6 Jogos educativos e programação

Programar pode ser uma tarefa complexa para alguém que nunca teve contato com códigos antes. Ensinar, portanto, esta função acaba sendo uma tarefa árdua, principalmente para estas pessoas sem base alguma de conhecimento sobre programação. Os jogos educativos vêm para auxiliar esse processo.

Como visto, jogos educativos facilitam o aprendizado, de modo que o jogador aprende enquanto está se divertindo. No aprendizado da programação este fato é muito importante visto que as dificuldades provenientes do desenvolvimento do raciocínio lógico tendem a desanimar quem se dispõe a aprender (FERRANDIN, STEPHANI, 2005).

Existem alguns jogos com o intuito de ensinar programação, como *CodeCombat* e *CodeMonkey*, citados na introdução desta monografia. Ambos os exemplos são jogos de navegador e têm como objetivo ensinar programação de uma maneira mais divertida que a de uma aula comum, utilizando histórias como base do jogo. (CODE COMBAT, 2017; CODE MONKEY STUDIOS INC, 2017).

2.7 Servidores Web

Tratando-se de jogos *online* é necessário um mecanismo para permitir que os jogadores se conectem simultaneamente ao jogo. Essa conexão é feita via internet, através de um sistema cliente-servidor, em que o servidor recebe uma solicitação do navegador, localiza em um sistema local de arquivos o documento solicitado e o envia de volta ao navegador (WINCKLER; PIMENTA, 2002).

O cliente nesse sistema é a máquina onde está o jogador. Este possui uma conexão com uma máquina onde está localizada toda informação do jogo e cabe ao servidor receber a informação de cada cliente e repassar atualizações aos mesmos (CECIN, 2005). Esse é o tipo de conexão mais comum para jogos que executam na web.

2.8 Programação e algoritmos

Um programa de computador é construído por linguagens que a máquina é capaz de ler e interpretar, gerando como resultado o que o programador espera ao programa-lo. A base para que esta programação surta

o efeito para o qual foi feita é o algoritmo, uma sequência de operações que, se executadas da maneira correta, gera a solução do problema que o programa se propõe a resolver.

Além da computação, há outras áreas com cursos que oferecem a disciplina de algoritmos como em engenharias em que se deseja introduzir conceitos de programação. Porém este acaba por se tornar mais um indicativo da dificuldade observada no aprendizado da programação, já que há um alto índice de reprovação e evasão por parte destas disciplinas (CUNHA et. al., 2006).

Além da grande bagagem de conteúdo que cursos superiores na área da computação exigem, fatores como dificuldade do aluno de desenvolver o raciocínio lógico, avaliações tradicionais mal-empregadas nos cursos, falta de motivação criada pelo despreparo do aluno e problemas de didática em que professores não buscam em outras áreas conteúdos afins que colaborariam para o ensino são fatores que aumentam a reprova e a evasão dos cursos nesta área, segundo Costa e Santos (2006). Ainda segundo os mesmos autores,

... o uso de um produto de software educacional, que enfatize animação gráfica, é de extrema importância como facilitador do processo de aprendizagem, visto que a apresentação de conceitos abstratos se torna mais viável e didática, melhorando a qualidade do material das aulas.

3 METODOLOGIA

O ambiente utilizado para executar o projeto é um *website*. Este conterá os elementos citados nos objetivos específicos como sistema de log in, jogo desenvolvido para a plataforma proposta e informações únicas para cada jogador de acordo com seu progresso nas aulas-jogos. Para tal, serão utilizadas linguagens de programação e marcação voltadas para o desenvolvimento web, além de ferramentas que agilizam todo o processo de criação.

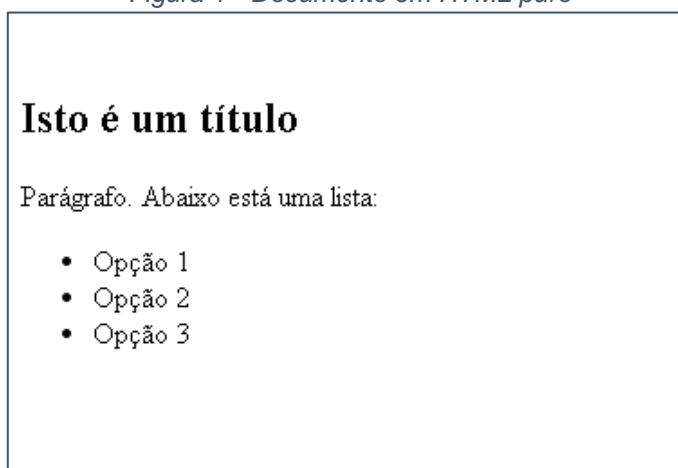
3.1 Ferramentas

Seguem descrições das ferramentas utilizadas para desenvolvimento do projeto.

3.1.1 HTML

Linguagem de marcação padrão no desenvolvimento web (W3C, 2017), o HTML é a base do *website*, descrevendo todo o conteúdo que cada página terá. Suas marcações permitem separar seções para cada tipo de conteúdo que terá a página, tais como determinar o que faz parte da lista que compõe um menu, separar as regiões onde ficarão cada área de conteúdo do *website*, diferenciar os títulos dos demais textos, entre outros.

Marcações no HTML são feitas através de *tags*. Essas *tags* são importantes para fazer a diferenciação do tipo de conteúdo que se está editando. Por exemplo, uma *tag* <p> indica o início de um parágrafo, sendo terminado pela *tag* </p>. Existem *tags* para indicar títulos, imagens, quadros, tabelas, botões e diversos outros mecanismos encontrados em páginas web. Desta forma o conteúdo fica organizado e indicado para que, ao montar a página web, todo o conteúdo seja disposto tal qual foi projetado, como pode ser visto na Figura 1.

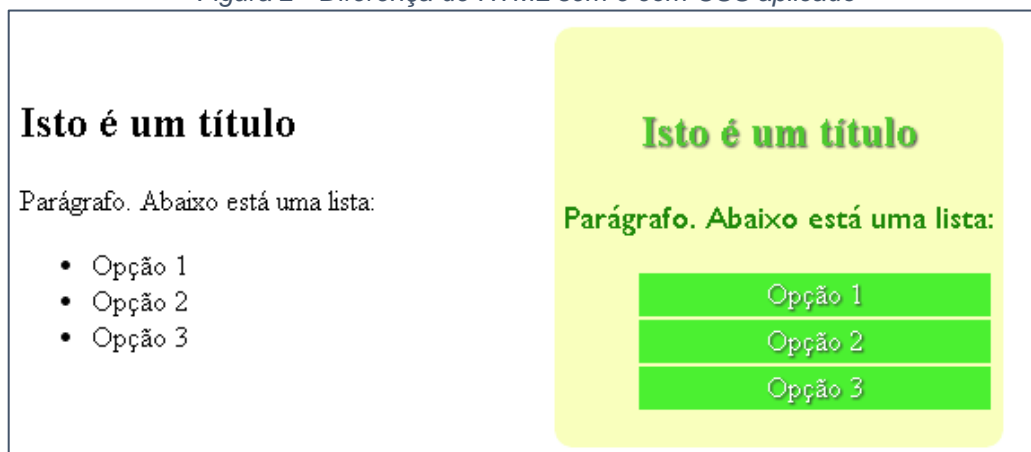
Figura 1 - Documento em HTML puro

Fonte: Elaborado pelo autor

3.1.2 CSS

A folha de estilos em cascata (CSS) é responsável por estilizar a página, atribuindo cores e formas para torná-la mais apresentável, com o carregamento mais rápido e dando ao programador o controle do layout (MARTINS; SILVA; FLÔRES, 2017).

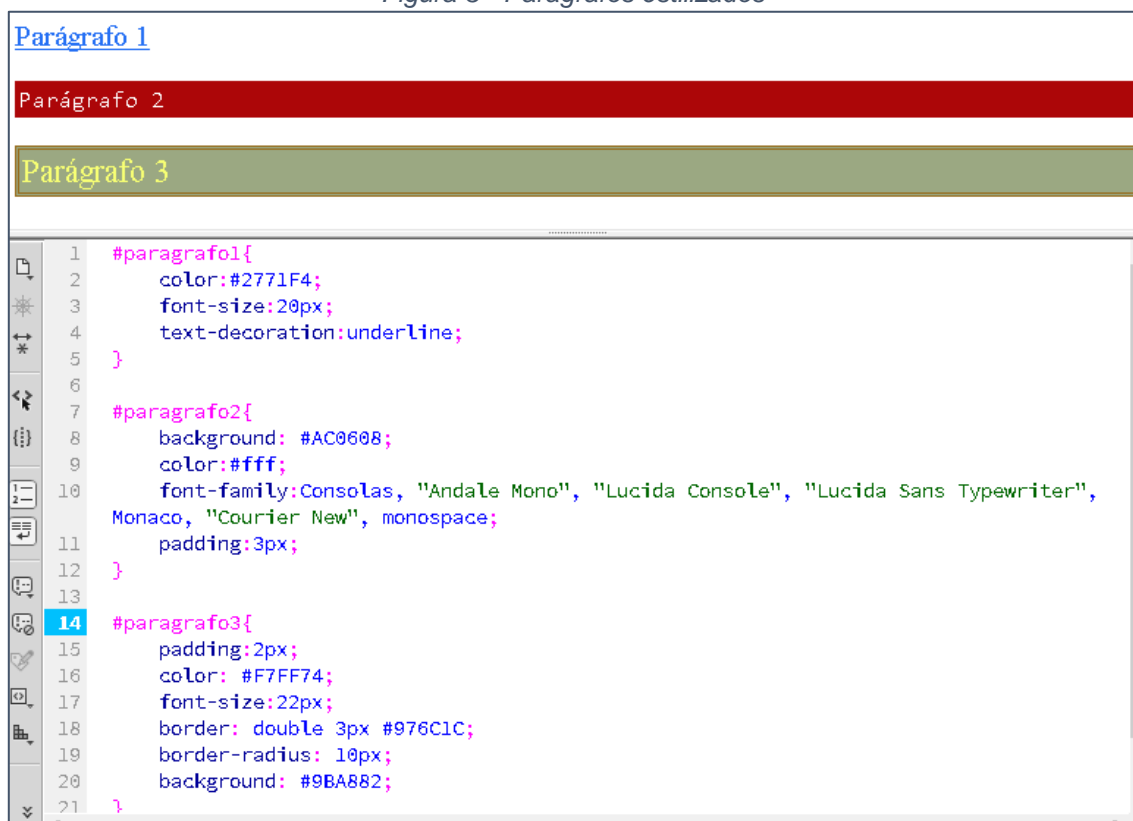
Com o CSS as páginas web possuem mais personalidade e conseguem passar uma melhor impressão a respeito de seu conteúdo, já que na forma crua do HTML a página é muito limitada no quesito de como dispor conteúdo. Isto fica mais claro vendo, na Figura 2, que a mesma página possui um acabamento diferente se estilizada com CSS.

Figura 2 - Diferença de HTML sem e com CSS aplicado

Fonte: Elaborado pelo autor

O CSS utiliza as *tags* do HTML para fazer a estilização, concedendo atributos designados pelo programador para uma *tag* específica da página web. São utilizados nomes únicos nas *tags* do HTML sendo eles do tipo *id*, para um único elemento, ou *class*, para um conjunto de elementos que terá a mesma formatação e, através destes nomes, o CSS é capaz de editar cada item de forma única, independente de possuírem a mesma *tag* ou não. Pode-se também editar todos os elementos da página com a mesma *tag* HTML, sem o uso de identificadores, como, por exemplo, mudar o tamanho de todos os parágrafos <p> da página. No exemplo exibido na Figura 3 têm-se o mesmo tipo de texto, utilizando a *tag* <p> indicativa de parágrafo, porém cada parágrafo é estilizado de maneira diferente.

Figura 3 - Parágrafos estilizados



Fonte: Elaborado pelo autor

3.1.3 Javascript

Sendo capaz de criar janelas, adicionar efeitos visuais ou apenas fazer cálculos em uma página sem precisar criar outro arquivo, o *Javascript* é uma

linguagem de programação orientada a objetos muito utilizada para o desenvolvimento de aplicações web. Com ela é possível a criação de *scripts* que são incluídos no código HTML da página para executar pequenos programas que podem ser úteis a ela. Por exemplo, se um *website* propõe mostrar uma imagem baseada no dia em que o usuário está o acessando, o *Javascript* soluciona esse problema.

Javascript é uma linguagem interpretada, assim como o Perl, e suporta variáveis *string*, *number* e *boolean*, além dos tipos *array*, *date* e expressões regulares. Com ela também é possível definir funções, criar formas geométricas, entre muitas outras funções. (FLANAGAN, 2006).

3.1.4 *Bootstrap*

O *Bootstrap* é um *framework* que, unindo CSS e *Javascript*, cria páginas web responsivas com muito mais rapidez e facilidade. Sendo *mobile first*, ou seja, prioriza o desenvolvimento web para ser visualmente adequado aos dispositivos móveis, o *Bootstrap* possui um sistema de divisão da tela em colunas, de forma que o conteúdo disposto na página não se deforma e sim é reposicionado de acordo com o número de colunas que é mostrado por vez no monitor do dispositivo que abre o *website*.

Há uma classe de *div* no *Bootstrap* chamada *container*, que como o nome sugere armazena conteúdo dentro dessa *div*. Estes recipientes de conteúdo são dispostos em *rows*, linhas dentro das colunas. Dessa forma nota-se que o *framework* faz da página web uma grade em que cada célula pode comportar um conteúdo diferente e, se necessário por conta da resolução, é inteiramente reposicionado, sem perder sua estrutura. Na Figura 4 têm-se exemplos de páginas construídas com o *Bootstrap*.

Figura 4 - Exemplos feitos com Bootstrap



Fonte: Bootstrap, disponível em: <themes.getbootstrap.com>

3.1.5 jQuery

Esta biblioteca de *Javascript* é também uma ferramenta para facilitar o desenvolvimento de *websites*. Ela é rápida, pequena e simplifica mecanismos como animações, gerenciamento de eventos, manipulação de documentos HTML, entre outros (THE JQUERY FOUNDATION, 2017).

A principal proposta desta biblioteca é transformar grandes trechos de código *Javascript* em poucas linhas, tornando o programa mais simples e fazendo com que tais trechos sejam compactados em métodos, podendo ser chamados quantas vezes forem necessários bastando uma linha de código (W3SCHOOLS, 2017). Além de proporcionar tais facilidades a biblioteca *jQuery* é utilizada pelo *framework Bootstrap*.

3.1.6 CreateJS

CreateJS é uma suíte de bibliotecas e ferramentas que utiliza HTML5 e *Javascript* para criar conteúdo interativo na web (GSKINNER, 2017). Esta suíte é composta de 4 bibliotecas, cada uma direcionada para áreas específicas que podem ser editadas numa página web.

A biblioteca *TweenJS* é voltada para criação de animações em HTML 5 e utiliza propriedades do *Javascript*. A biblioteca *SoundJS* é voltada para

edição de áudio. *PreloadJS* é uma biblioteca para facilitar a administração de carregamento de dados e arquivos.

A *EaseJS*, utilizada neste trabalho, é utilizada para facilitar o uso da função Canvas do HTML5, que permite a criação e manipulação de imagens e formas em uma página sem a necessidade de *plug-ins*. Além disso, ela está integrada ao ambiente *Animate* da suíte Adobe, também utilizada neste projeto (ADOBE SYSTEMS INCORPORATED, 2017).

3.1.7 PHP

A linguagem de *script* PHP é *open source* (de código aberto) e é "especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML" (THE PHP GROUP, 2017). A grande vantagem desta linguagem é o fato de poder ser utilizada no meio de um documento HTML, sendo demarcada inicialmente com `<?php` e, no final, é demarcado por `?>`.

Uma utilidade do PHP é a facilidade de comunicação entre os clientes e o servidor da aplicação web. Com poucas linhas é possível fazer a conexão com o banco de dados para validar um sistema de log in, para verificar a existência de uma sessão, entre outras funções. Isso torna a criação de um *website* muito mais rápida.

3.1.8 SQL

Abreviação de *Structured Query Language*, linguagem de consulta estruturada, o SQL é uma linguagem voltada para criação e manipulação de bancos de dados. Muitos programas de bancos de dados utilizam tal linguagem como MySQL, SQL Server, MS Access, Oracle, Sybase, Informix e Postgres. (W3SCHOOLS, 2017)

3.1.8.1 MySQL

É o banco de dados *open source* mais popular do mundo, sendo utilizado por empresas como Facebook, Google, Adobe, Alcatel Lucent e Zappos (ORACLE CORPORATION, 2017). Com seu custo zero e sua alta confiabilidade, é plausível que muitas empresas o utilizem.

MySQL é um banco de dados do tipo relacional, ou seja, "é uma coleção de dados com relacionamentos predefinidos entre si" (AMAZON WEB SERVICES, 2017). A organização destes dados se dá por um conjunto de tabelas, sendo que cada coluna dela representa um tipo de dado e cada linha contém um valor armazenado para este tipo. Cada linha deve ser marcada com uma chave principal, um item com valor único na tabela (AMAZON WEB SERVICES, 2017).

Na web são feitas comumente muitas leituras de dados e poucas escritas. Esse banco de dados consegue agilizar essas leituras, tornando o acesso muito mais rápido pelos usuários (SUEHRING, 2002). O ambiente UniserverZ utilizado neste trabalho e descrito no item 4.3.2 também utiliza como banco de dados padrão o MySQL.

3.1.9 Apache

Servidor web de código aberto, Apache é o mais popular do mundo, sendo desenvolvido até hoje por uma grande comunidade de colaboradores (APACHE SOFTWARE FOUNDATION, 2017). Sendo um servidor, ele cria uma conexão, gerencia os dados de uma aplicação e os clientes que a acessam e permite a comunicação entre clientes através da internet.

O Apache também pode ser utilizado como servidor local (*localhost*), que é a função usada nesse trabalho. Ele unido ao banco de dados torna possível a criação de um sistema onde o usuário possa fazer log in no *website* e navegar até encerrar sua sessão. O UniserverZ é o ambiente que propicia tal

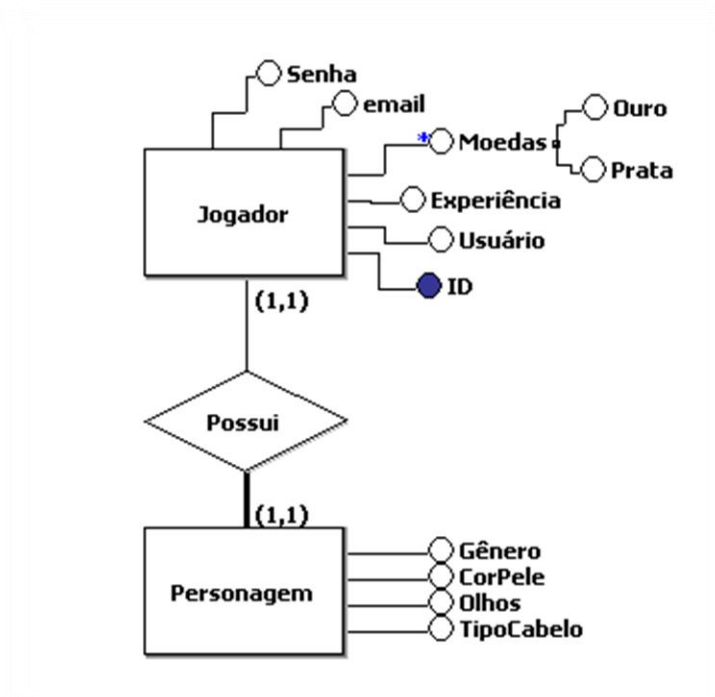
comunicação, tendo integrados em sua interface o Apache e o MySQL (TRIPLE O, LLC, 2014).

3.2 Métodos

Todo o desenvolvimento do projeto foi baseado na revisão bibliográfica e no estudo das ferramentas citadas anteriormente. Para tal foram usados diferentes ambientes de desenvolvimento, cada um com um propósito e serão identificados no decorrer da monografia.

De início efetuou-se o planejamento do *website*, de forma que o layout deva apresentar na tela todo o conteúdo proposto nos objetivos de forma clara e amigável, além de exibir todos os dados de cada jogador. Para o banco de dados, onde estarão os dados citados, foi desenvolvido um pequeno MER (Modelo Entidade Relacionamento) com a ferramenta *BrModelo* (CÂNDIDO, 2007), como pode ser visto na Figura 5.

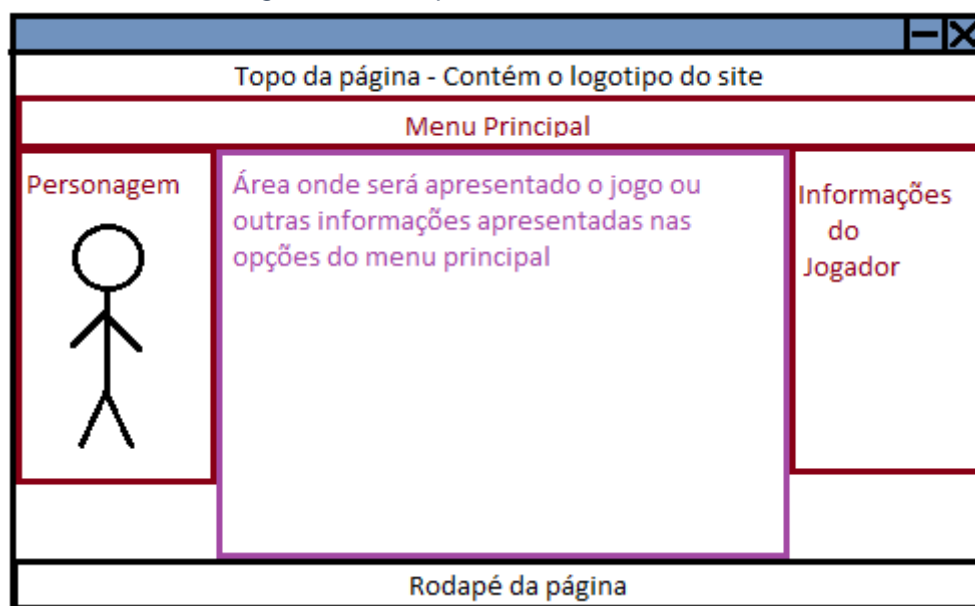
Figura 5 - Modelo Entidade Relacionamento



Fonte: Elaborado pelo autor

Utilizando como base o MER proposto foi desenvolvido um protótipo de uma página padrão de conteúdo do *website*, utilizando a ferramenta *Pencil* (EVOLUS, 2017), mostrado na Figura 6. Seguindo esta base os conteúdos principais a serem dispostos na tela, como o conteúdo específico da página, *avatar* do jogador e suas informações específicas como moedas e experiência, têm cada um seu devido lugar designado, estabelecendo um padrão e facilitando a navegação do usuário.

Figura 6 - Protótipo de interface com usuário



Fonte: Elaborado pelo autor

Seguindo com a arquitetura do website, há uma página inicial onde pode-se criar um novo cadastro ou iniciar o log in com um cadastro previamente efetuado. Ao iniciar a sessão estarão disponíveis mais duas páginas com conteúdo interativo, a página inicial em que informações breves sobre o website devem aparecer e a página de jogos, intitulados missões. Em ambas as páginas é possível visualizar o menu principal, o *avatar* do jogador e suas informações de jogo pessoais, como moedas e experiência, seguindo o protótipo proposto.

4 DESENVOLVIMENTO

Dada a metodologia e tendo um pré-planejamento da aplicação, deu-se início ao desenvolvimento do projeto. Foi escolhida como plataforma para iniciar o desenvolvimento do *website* o programa da suíte Adobe, *Dreamweaver* (ADOBE SYSTEMS INCORPORATED, 2017). Devido ao pré-conhecimento desta ferramenta e de recursos que possui para tornar mais simples a integração dos arquivos de diferentes extensões que serão utilizados, o *Dreamweaver* foi escolhido dentre outras, além do fato de possuir o *Bootstrap*, também utilizado neste projeto, integrado em seu programa.

4.1 Front End

Tendo início pelo desenvolvimento do *front end*, o projeto se baseia no esboço mostrado no item 3.2 para criar um layout base para as páginas navegáveis disponíveis no menu. Este menu contém as páginas em que o usuário já cadastrado possa encontrar as funções disponibilizadas no *website*, compatíveis com os objetivos específicos propostos.

A página base foi desenvolvida totalmente dentro do ambiente do *Dreamweaver* e utilizou o *framework Bootstrap* para melhorar a exibição do conteúdo tornando a página responsiva, podendo ser visualizada corretamente em dispositivos de diferentes resoluções. A Figura 7 apresenta os componentes da página: menu principal, quadro esquerdo com espaço para o *avatar* do jogador, quadro central para conteúdo específico da página, quadro direito com informações do usuário e o rodapé da página. Com exceção do conteúdo específico contido no interior do quadro central, toda a estrutura é mantida em toda e qualquer página visitada pelo usuário enquanto mantida a sessão.

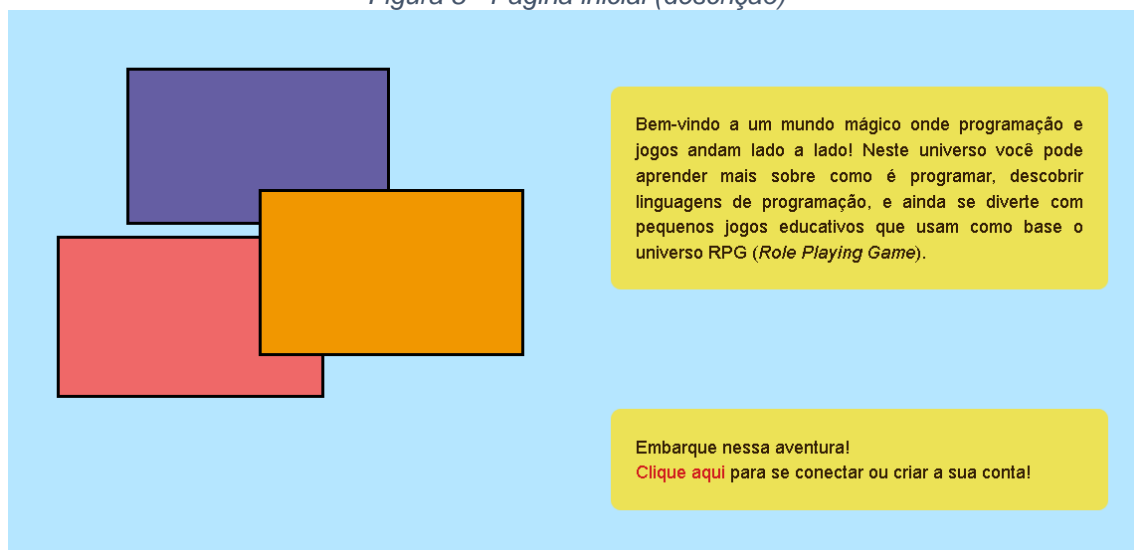
Figura 7 - Layout básico das páginas de conteúdo



Fonte: Elaborado pelo autor

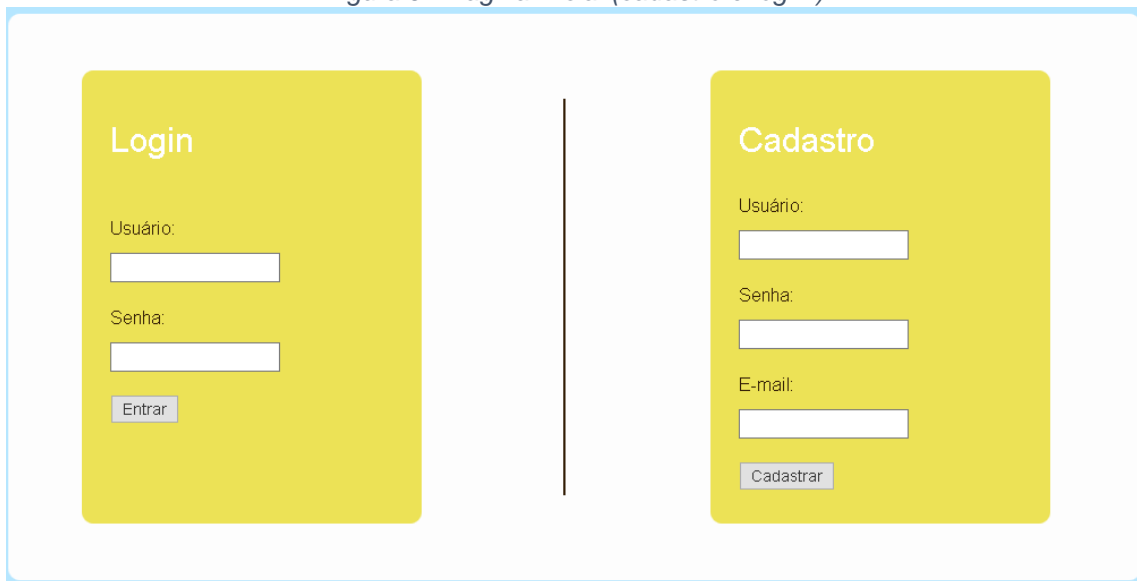
A seguir, para a página de cadastro e log in, que também é a página inicial do *website*, foi feita uma estilização diferente de modo que não há acesso ao menu pelo usuário que não está conectado, tornando obrigatório o início de uma sessão para tal. Nesta página há um breve resumo do que se trata o conteúdo do *website* e há um quadro indicando as áreas correspondentes para log in e cadastro de usuário, como visto nas Figuras 8 e 9. Estas páginas também foram desenvolvidas inteiramente no ambiente do *Dreamweaver*.

Figura 8 - Página inicial (descrição)



Fonte: Elaborado pelo autor

Figura 9 - Página inicial (cadastro e log in)



The image shows a web interface with two yellow rectangular panels side-by-side, separated by a thin vertical line. The left panel is titled 'Login' and contains two input fields labeled 'Usuário:' and 'Senha:', followed by a button labeled 'Entrar'. The right panel is titled 'Cadastro' and contains three input fields labeled 'Usuário:', 'Senha:', and 'E-mail:', followed by a button labeled 'Cadastrar'.

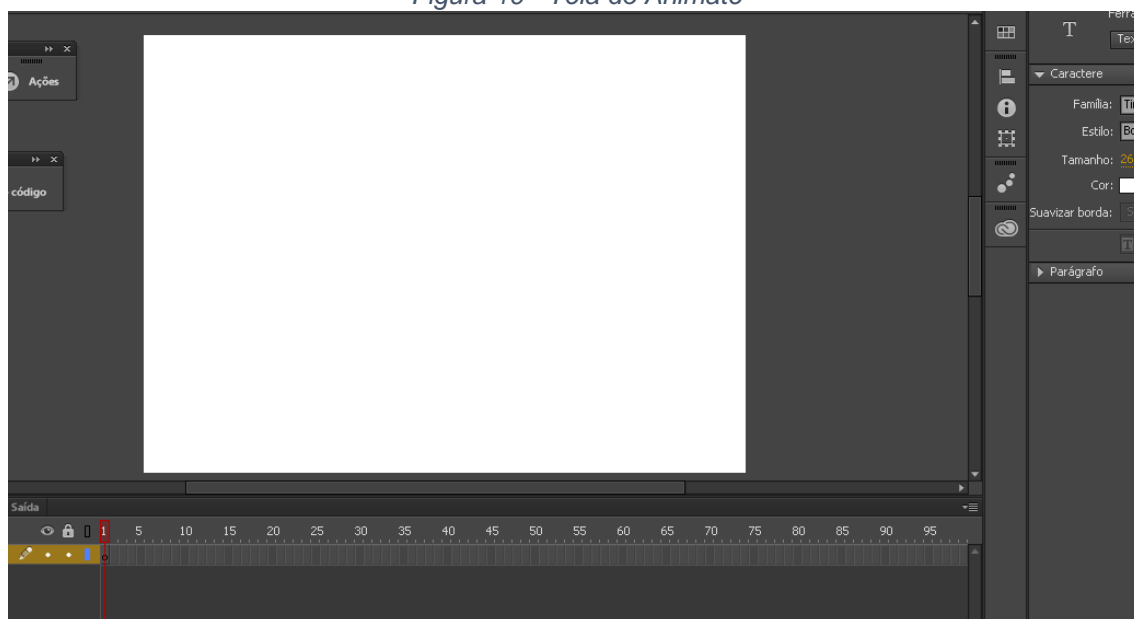
Fonte: Elaborado pelo autor

4.2 Animação do jogo

Com o layout das páginas já estabelecido, iniciou-se o desenvolvimento da animação em HTML5 que viria a se tornar o jogo inicial. Este jogo tem como objetivo introduzir o usuário no universo da computação, ensinando conceitos básicos como a definição de algoritmo.

Todo o processo da animação foi desenvolvido no ambiente *Animate*, da suíte Adobe (ADOBE SYSTEMS INCORPORATED, 2017). Nessa plataforma é possível posicionar as imagens e criar o ambiente do jogo de maneira muito simples, permitindo manter o foco maior em desenvolver os *scripts* por trás da animação. A interface do ambiente, ilustrada na Figura 10, é composta por um palco, em que se pode visualizar como ficará a imagem da animação na resolução escolhida, uma linha do tempo dividida em quadros a serem animados como o desenvolvedor preferir e ferramentas de edição de vários elementos que podem ser inseridos no palco como formas geométricas e textos.

Figura 10 - Tela do Animate



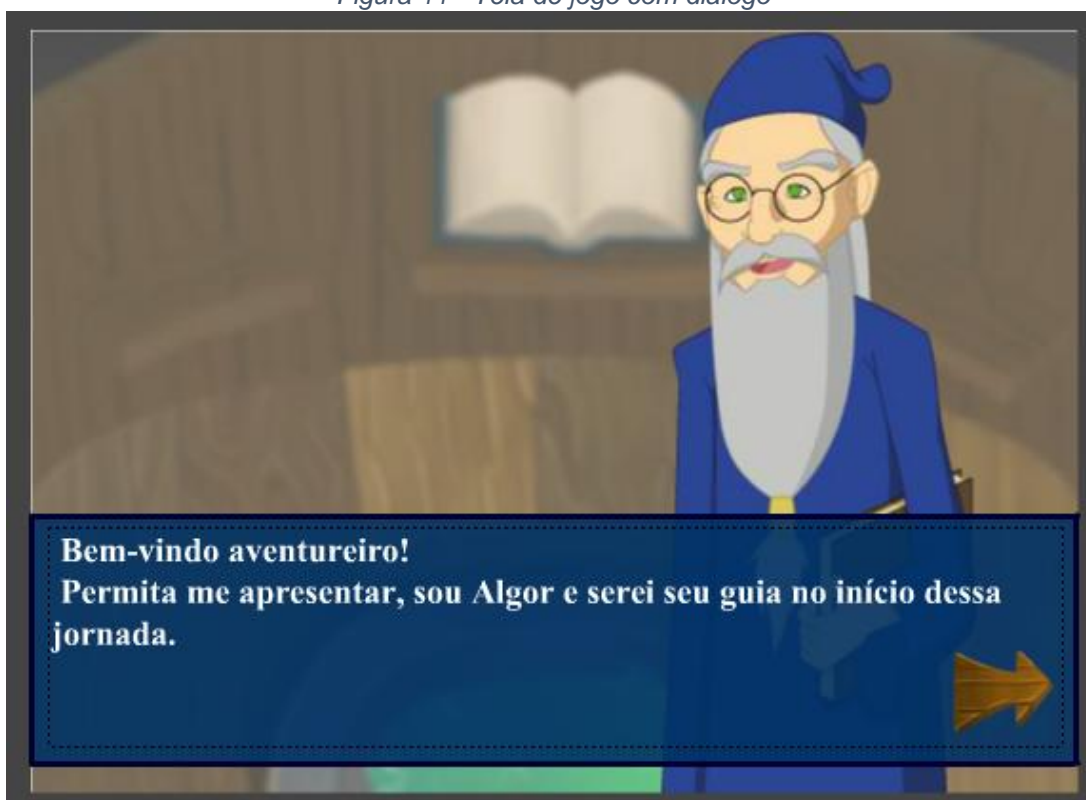
Fonte: Elaborado pelo autor

Dando início à animação do jogo, em cada quadro em que há texto foi inserido um botão para realizar a troca dos conteúdos de texto, movendo para o quadro seguinte. O texto consiste em uma história apresentada ao jogador para ambientá-lo no mundo onde se passa o jogo. Este diálogo com o usuário faz parte da imersão do jogador.

Uma das telas contendo esta conversação está exibida na Figura 11, em que o jogador é apresentado a um NPC (*Non-Player Character*), um personagem que faz parte da história do jogo, mas que não é manipulado por outro usuário. Os botões e outros elementos que são apresentados na animação foram programados com *scripts* da biblioteca *EaseJS*, citada e descrita no item 3.1.6.

Seguindo a animação, após o texto introdutório, é apresentado um pequeno jogo em que a pessoa deve selecionar poções diferentes na ordem em que são apresentados em uma receita, observada na Figura 12. Se errar, o jogador será obrigado a recomeçar a sequência até que a complete totalmente na ordem apresentada. Este minijogo tem como objetivo explicar o conceito de algoritmo, uma sequência de ações que devem seguir certa ordem de ocorrência para gerar o resultado esperado, tal qual é necessário em uma receita.

Figura 11 - Tela de jogo com diálogo



Fonte: Elaborado pelo autor

Figura 12 - Tela de jogo com receita

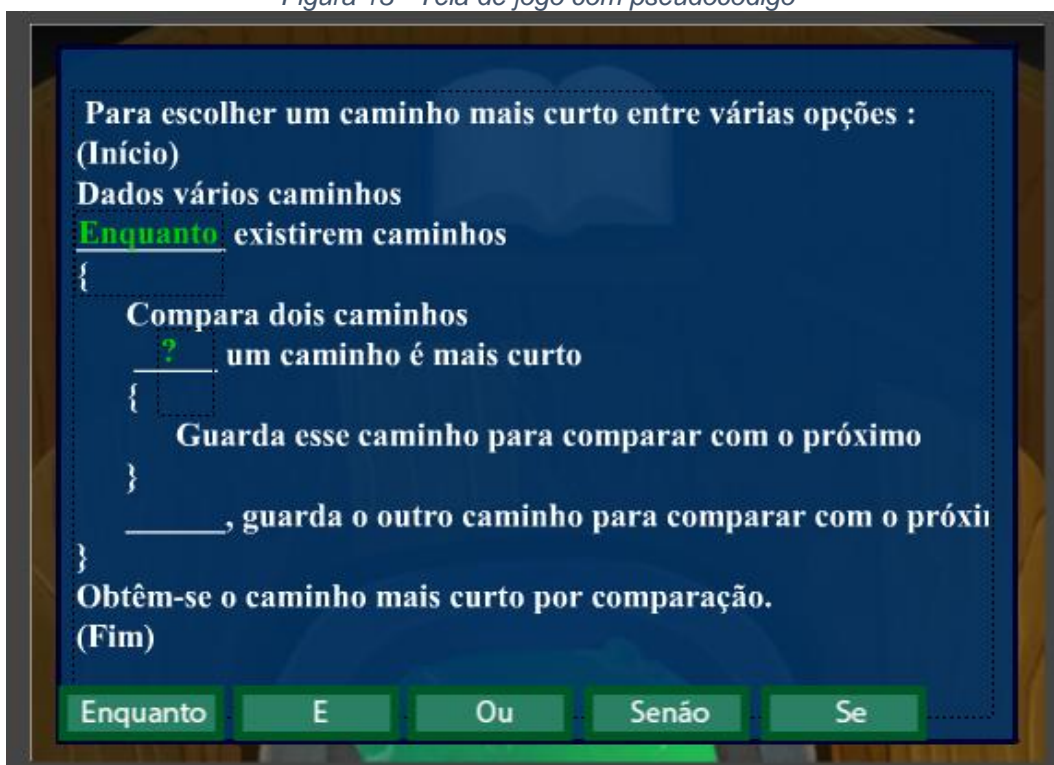


Fonte: Elaborado pelo autor

Ao terminar a receita corretamente e visualizar a explicação do conceito de algoritmo o jogador será apresentado ao conceito de pseudocódigo, um código escrito não em linguagem de programação, mas em português, que simboliza como um programa é escrito. Ao visualizar um código escrito em português o jogador terá uma noção mais clara de como ele deverá escrever seus próprios programas quando aprender, de fato, alguma linguagem de programação.

Dado o conceito segue-se outro minijogo, dessa vez utilizando pseudocódigos. A Figura 13 mostra um dos três textos apresentados para o jogador, que deve escolher, entre as opções dadas, a que preenche de maneira correta os espaços indicados pelo símbolo de interrogação, fazendo com que o resultado da proposta do enunciado seja obtido. Com isso o jogador desenvolverá um raciocínio lógico que facilitará o aprendizado de qualquer linguagem de programação.

Figura 13 - Tela de jogo com pseudocódigo



Fonte: Elaborado pelo autor

Completando mais esta etapa o jogador está pronto para prosseguir em seus estudos, tendo agora em sua base de conhecimento uma ideia de como

se deve programar para obter os resultados esperados. Chega-se, então, ao fim da animação do jogo tutorial.

4.3 Back End

Seguindo o processo de desenvolvimento, foram realizados todos os processos para tornar possível a criação de um usuário com informações únicas, armazenar tais informações e exibi-las cada vez que o mesmo usuário retornar ao *website*.

4.3.1 Sistema de cadastro e log in

Para que cada jogador tenha seu progresso registrado é necessário identifica-lo e guardar a informação de que ele é um usuário do *website*. Para isso foi criado um sistema de cadastro de usuários e de log in de usuários. Foi definido que basta que o usuário forneça um nome único para identificação, uma senha secreta para que somente ele tenha acesso a suas informações e um e-mail para futuras implementações de recuperação de conta, caso o usuário esqueça sua senha.

Para tornar o log in possível é necessário criar uma sessão para o usuário, e isto foi feito utilizando a linguagem PHP. Cria-se uma sessão e definem-se as variáveis de sessão para o usuário que está requisitando conexão. Também foi feito em PHP a validação do log in, analisando se os campos estão preenchidos e se o usuário é válido. Para o cadastro é verificado se o log in a ser registrado já existe ou se pode ser criado. Segue, como apresentado na Figura 14, um trecho do código da página que realiza a verificação de dados para realizar o log in do usuário. Para o cadastro foi feito algo similar: ao invés de verificar se o log in e a senha constam no banco e formam um par válido é verificado se há um log in com o nome já registrado (sendo o campo log in com valor único para cada usuário) e realizado o registro

dos dados no banco de dados caso os campos estejam preenchidos corretamente.

Figura 14 - Trecho de código PHP da página de log in

```
<?php
session_start();
$login = $_POST['login'];
$entrar = $_POST['entrar'];
$senha = $_POST['senha'];
$connect = new mysqli("localhost","admin","password","tccjogo");
if (isset($entrar)) {
    $verifica = mysqli_query($connect,"select * from usuarios where login = '$login'
and senha = '$senha'")
    or die("Erro ao selecionar login");
    if (mysqli_num_rows($verifica)<=0){
        echo"<script language='javascript' type='text/javascript'>alert('Login e/ou
senha incorretos');window.location.href='/#logincadastro';</script>";
        die();
    }else{
```

Fonte: Elaborado pelo autor

4.3.2 Banco de dados

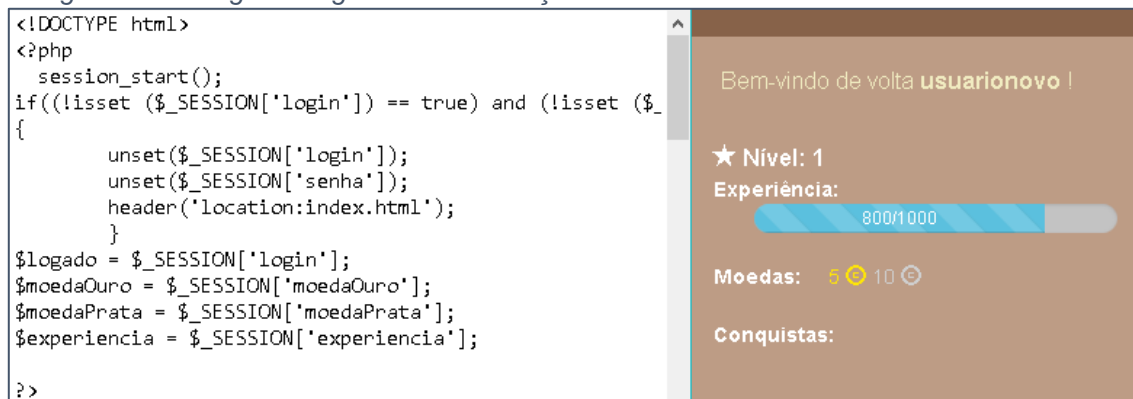
Para gravar os dados necessários, foi criado um banco de dados SQL, baseado no MER proposto no capítulo 3.2. Utilizando o servidor UniserverZ para criar uma rede local foi possível estabelecer uma conexão entre o banco criado e o aplicativo.

Este banco de dados possui uma tabela única, devido ao número de colunas ser pequeno, inviabilizando a criação de várias tabelas com duas ou três colunas cada no máximo. Esta tabela possui índices para armazenar as informações de log in, usuário, senha, e-mail, um identificador único para cada usuário e as informações referentes aos dados de cada jogador, que são o número de moedas de ouro e prata, a quantia de experiência e as informações do *avatar* do jogador.

Os dados armazenados são gravados pelo código do *website* em variáveis de sessão, de forma que cada usuário seja capaz apenas de visualizar seus próprios dados. Estas informações, por sua vez, podem ser vistas nas páginas navegáveis, sendo os dados dos personagens carregados na área designada à esquerda do layout e outros dados como experiência e moedas à direita, como apresentados na Figura 15, exemplificando a mostra de

dados e o código que controla os dados. Todas estas etapas foram feitas com auxílio da linguagem PHP e implementadas em cada página desenvolvida.

Figura 15 - Código e imagem com informações de um usuário extraído do banco de dados



Fonte: Elaborado pelo autor

Esta etapa finaliza todos os procedimentos realizados para construção do *website* proposto, incluindo jogo educativo, interface amigável e criação de sistema de usuário para cada jogador ter um progresso único.

5 CONCLUSÃO

Com o programa finalizado e o *website* testado e funcional, chegou-se a um resultado satisfatório, englobando os objetivos específicos propostos. O objetivo principal do projeto seria melhor atendido caso o *website* fosse colocado na rede aberta para que vários usuários pudessem usufruir dos jogos e retornar um *feedback* para melhorá-lo.

Jogos com a finalidade de ensinar programação são ainda pouco difundidos no Brasil, e além dos desenvolvedores que os criam divulgarem seus trabalhos, cabe aos educadores apresentarem aos seus alunos esse tipo de ferramenta para o aprendizado. E isto vale não só para a computação como para outras áreas do conhecimento que possuem jogos digitais educativos que auxiliariam muitos estudantes, porém acabam não atingindo um grande público pela falta de incentivo e divulgação.

Por fim, com o avanço crescente da tecnologia espera-se que em breve o conhecimento da programação seja, além de mais acessível, parte do ensino básico das futuras gerações, e que este projeto, caso seja continuado, somado aos jogos digitais educativos já existentes, possam fazer a diferença na educação destes futuros jovens.

5.1 Trabalhos futuros

Como continuação para este projeto, propõe-se as seguintes melhorias:

- Ampliação do conteúdo dos jogos, visando ensinar a programar em diferentes linguagens com vários níveis de aprendizado para que o jogador consiga progredir do básico de uma linguagem até o estágio avançado.
- Criação de uma página de personalização pós-cadastro, tornando a imersão mais realista de modo que o jogador terá mais opções para seu *avatar* e assim poder criá-lo como imaginar.

- Criação de um sistema de premiações por cumprir certas tarefas nos jogos, aumentando o anseio pela progressão nos estudos por parte do jogador.
- Melhorias no layout para torna-lo mais bonito e atrativo com elementos de design.
- Otimização do website para dispositivos com diferentes resoluções, criando multi-versões ou até mesmo um aplicativo para dispositivos móveis.

REFERÊNCIAS

ADOBE SYSTEMS INCORPORATED. **Adobe Creative Cloud**, Adobe Systems Inc., 2017. Disponível em: <<http://www.adobe.com/br/creativecloud.html>>. Acesso em: 23 nov. 2017.

AMAZON WEB SERVICES. **O que é banco de dados relacional?** 2017. Disponível em: <<https://aws.amazon.com/pt/relational-database/>>. Acesso em: 21 nov. 2017.

APACHE SOFTWARE FOUNDATION. **Whats is the ASF?** 2017. Disponível em: <<https://www.apache.org/foundation/>>. Acesso em: 21 nov. 2017.

CÂNDIDO, C. H. **BrModelo 2.0**. 2007. Disponível em: <<http://www.sis4.com/br/Modelo/>>. Acesso em: 22 nov. 2017.

CECIN, F. R. **FreeMMG: uma arquitetura cliente-servidor e par-a-par de suporte a jogos maciçamente distribuídos**. Instituto de informática da Universidade Federal do Rio Grande do Sul, Porto Alegre, mar. 2005.

CHAPMAN, C. **15 free games that will help you learn how to code**. Business Insider, 2017. Disponível em: <<http://www.businessinsider.com/15-free-games-that-will-help-you-learn-how-to-code-2017-4>>. Acesso em: 20 nov. 2017.

CODE COMBAT. San Francisco, 2017. Disponível em: <<https://br.codecombat.com/home>>. Acesso em: 23 nov. 2017.

CODE MONKEY STUDIOS INC. **Code Monkey**. Brooklyn, 2017. Disponível em: <<https://www.playcodemonkey.com/>>. Acesso em: 23 nov. 2017.

CODECADEMY. Nova Iorque, 2017. Disponível em: <<https://www.codecademy.com/pt>>. Acesso em: 19 nov. 2017.

CONTI, F. **História: primeiros jogos digitais**. Universidade Federal do Pará, 2015. Disponível em: <<http://www.ufpa.br/dicas/net1/int-h-jo.htm>>. Acesso em: 21 nov. 2017.

COSTA, H. A. X.; SANTOS, R. P. dos. **Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática**. Universidade Federal de Lavras Departamento de Ciência da Computação, Lavras. 24 jan. 2006.

CUNHA, V. V. da; FALKEMBACH, G.; KLEMMANN, M.; RAPKIEWICZ, C. E.; ROSA, N. dos S.; SEIXAS, L. **Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais**. Novas Tecnologias na Educação v.4 nº 2, CINTED-UFRGS, dez. 2006.

DEVMEDIA. Rio de Janeiro, 2017. Disponível em: <<http://www.devmedia.com.br/>>. Acesso em: 19 nov. 2017.

EVOLUS. Pencil Project. 2017. Disponível em: <<https://pencil.evolus.vn/>>. Acesso em: 22 nov. 2017.

FALCÃO, T.; RIBEIRO, J. C. **Mundos virtuais e identidade social: processos de formação e mediação através da lógica do jogo**. LOGOS 30 Tecnologias de Comunicação e Subjetividade. Ano 16, 1º semestre 2009.

FERRANDIN, M.; STEPHANI, S. L. **Ferramenta para o ensino de Programação via Internet**. Programa de Incentivo à Pesquisa da UNERJ, Jaraguá do Sul, 2005.

FERREIRA, W. P. **Tipos de MMOGs**. Universidade Federal de Campina Grande, set. 2006. Disponível em: <<http://www.dsc.ufcg.edu.br/~pet/jornal/setembro2006/informatica.htm>>. Acesso em: 21 nov. 2017.

FLANAGAN, D. **Javascript The Definitive Guide**. O'Reilly 5ª ed., Sebastopol, dez. 2008.

GSKINNER. CreateJS. 2017. Disponível em: <<https://createjs.com/>>. Acesso em: 20 nov. 2017.

MARQUES, D. L.; COSTA, L. F. S.; SILVA, M. A. de A.; REBOUÇAS, A. D. D. S. **Atraindo Alunos do Ensino Médio para a Computação: Uma Experiência Prática de Introdução a Programação utilizando Jogos e Python**. Aracaju: Anais do XXII SBIE - XVII WIE, 2011.

MARTINS, J. P.; SILVA, J. M. C. da; FLÔRES, O. C. **A linguagem de estilo CSS: um exemplo de plano de aula integrando as disciplinas de língua portuguesa e introdução à linguagem de programação.** XIV EVIDOSOL e XI CILTEC Online, jun. 2017. Disponível em: <<http://evidosol.textolivre.org/site/>>. Acesso em: 22 nov. 2017.

MEDEIROS, T. J.; SILVA, T.R. da; ARANHA, E. H. da S. **Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura.** RENOTE v.11, nº3, Rio Grande do Sul, CINTED-UFRGS, 2013.

ORACLE CORPORATION. **MySQL.** 2017. Disponível em: <<https://www.mysql.com/>>. Acesso em: 20 nov. 2017.

PÁDUA, V. C. de. **Ambiente de suporte a jogos WEB voltado para a área de ensino a distância.** Centro de Informática da Universidade Federal de Pernambuco, Recife, maio 2008.

ROSA, M. **A Construção de Identidades online por meio do Role Playing Game: relações com o ensino e aprendizagem de matemática em um curso à distância.** Universidade Estadual Paulista Instituto de Geociências e Ciências Exatas, Rio Claro, 2008.

SAVI, R.; WANGENHEIM, C. G. von; ULBRICHT V.; VANZIN, T. **Proposta de um Modelo de Avaliação de Jogos Educacionais.** Novas Tecnologias na Educação v.8 nº 3, CINTED-UFRGS, dez. 2010.

SILVA, G. S.; JUSTINO L. D. **Novas práticas na educação: o uso de jogos digitais para fortalecer a aprendizagem.** III Encontro nacional sobre hipertexto, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 29 a 31 out. 2009.

SUEHRING, S. **MySQL Bible.** Wiley Publishing Inc. Nova Iorque, 2002. Disponível em: <<http://justpain.com/eBooks/Databases/MySQL/MySQL%20Bible.pdf>>. Acesso em: 20 nov. 2017

TAKATALO, J.; HÄKKINEN, J.; KAISTINEN, J.; NYMAN, G. **Presence, Involvement and Flow in Digital Games.** Springer, Regina Bernhaupt Editor Evaluating User Experience in Games capítulo 3, 2010. Disponível em: <<http://www.springer.com/la/book/9781848829626> >. Acesso em: 23 nov. 2017.

THE JQUERY FOUNDATION. **What is jQuery?** 2017. Disponível em: <<https://jquery.com/>>. Acesso em: 21 nov. 2017.

THE PHP GROUP. **O que é PHP?** 2017. Disponível em: <https://secure.php.net/manual/pt_BR/intro-whatism.php>. Acesso em: 20 nov. 2017.

TRIPLE O, LLC. **Uniform Server**. 2014. Disponível em: <<http://www.uniformserver.com/>>. Acesso em: 21 nov. 2017.

W3C. **What is HTML?** The World Wide Web Consortium, 2017. Disponível em: <<https://www.w3.org/html/>>. Acesso em: 23 nov. 2017.

W3SCHOOLS. **jQuery Tutorial**. 2017. Disponível em: <<https://www.w3schools.com/jquery/default.asp>>. Acesso em: 21 nov. 2017.

WINCKLER, M.; PIMENTA, M. S. **Avaliação de usabilidade de sites web**. Instituto de Informática - UFRGS, Porto Alegre, 2002.

BIBLIOGRAFIA COMPLEMENTAR

ALMEIDA, G.; GIRAFFA, L. M. M.; MARCZAK, S. S. **O ensino de algoritmos e programação mediado por um ambiente na Web**. Pontifícia Universidade Católica do Rio Grande do Sul - Faculdade de Informática, Porto Alegre, 2003.

CRESPO, S.; EVARISTO, J. **Aprendendo a Programar Programando numa linguagem algorítmica executável (ILA)**. Formato digital, 2ª ed. Maceió, 2010. Disponível em: <<http://professor.ic.ufal.br/jaime/livros/Livro%20ILA%20Edicao%202.pdf>>. Acesso em: 20 nov. 2017.

CYBER CREATIONS INC. **MMORPG**. 2017. Disponível em: <<https://www.mmorpg.com/>>. Acesso em: 23 nov. 2017.

FEITOSA, A. **Organização da informação na web: das tags à web semântica**. Thesaurus 1 32 p., Brasília, 2006.

NIXON, R. **Learning PHP, MySQL & Javascript**. O'Reilly, Sebastopol, jul. 2009.

SAVI, R.; ULBRICHT V. R. **Jogos digitais educacionais: benefícios e desafios**. Novas Tecnologias na Educação v.6 nº 2, CINTED-UFRGS, dez. 2008.

VEER, E. V. **Javascript for dummies**. Wiley Publishing Inc. 4ª ed. Indianapolis, 2005.

APÊNDICE A - ROTEIRO DA ANIMAÇÃO "TUTORIAL"

Parte 1: Apresentação

Apresentação do personagem Algor, o guia dos iniciantes que vive na grande biblioteca do reino, que apresenta o jogador como parte do universo em que está e afirma que ele é um aventureiro prestes a iniciar uma jornada neste mundo em que os problemas e dificuldades são resolvidos com lógica e programação.

(Diálogo extraído da animação)

- Bem-vindo aventureiro! Permita me apresentar, sou Algor e serei seu guia no início dessa jornada.

- Aqui na grande biblioteca você aprenderá o básico para seguir sua aventura pelo mundo. Vamos começar com exercícios muito simples.

Parte 2: Jogo das poções

Após se apresentar, Algor explica as regras do jogo das poções, que tem o intuito de ensinar o conceito de algoritmo. Ao final do jogo este conceito é explicado mais detalhadamente.

(Diálogo extraído da animação)

- Você é capaz de memorizar uma sequência de instruções e depois segui-la na ordem correta? Parece fácil não é mesmo, veremos como se sai no seu primeiro teste!

Aqui vão as regras do Jogo das Poções:

- O objetivo é seguir a receita mostrada na sequência correta*
- Clique nas poções para selecioná-las*
- Se errar qualquer item da receita terá de começar de novo*
- Preparado? Então vamos começar!*

(Próximo quadro mostra a tela da receita a ser seguida)

...

(Telas de resposta possíveis)

Ao errar:

- Essa não! Você errou a receita, vamos ter de recomeçar...

Ao acertar (prossegue o diálogo após esta tela)

- Parabéns jovem aprendiz! Você completou a receita na ordem correta!

- É um teste muito simples não é mesmo? Mas acredite se quiser, é a base para tudo que aprenderá daqui em diante.

- Para resolver um problema de programação é necessário organizar as ideias em ordem e seguir fazendo uma coisa de cada vez, passo a passo.

- Esse processo de construir ideias em ordem é chamado de Algoritmo, a base de toda linguagem de programação.

- Para ficar mais claro vamos a mais um teste! Dessa vez vamos ter uma ideia melhor de como se estrutura um código.

Parte 3: Jogo de pseudocódigos

Nesta parte, considerando que a ideia de algoritmo foi entendida, o jogador tem de utilizar o que aprendeu para entender e completar um pseudocódigo. Serão apresentados três textos em que, para preencher as lacunas, o jogador deve pensar logicamente e seguir a ideia de um algoritmo, pensar em como deve-se prosseguir para obter o resultado esperado.

(Diálogo extraído da animação)

- Vou apresentar o que chamamos de pseudocódigo, um texto que se parece com um programa, porém sem usar uma linguagem de programação, somente com frases em português.

- O objetivo aqui será completar os espaços nesse pseudo código com as palavras que melhor resolvem os problemas que vou apresentar para você, jovem aprendiz.

- Vamos começar! Lembre-se, pense no algoritmo, na ordem em que tudo deve ser feito.

(A seguir são apresentados os textos e os botões de opções)

Texto 1:

Para subtrair dois números e ter resultado positivo deve-se:

(Início)

_____ um número A é menor que um número B

{

Resultado é = B - A

}

{

Resultado é = A - B

}

(Fim)

Opções dadas para preencher os espaços: Enquanto, E, Ou, Senão e Se

Texto 2:

Para escolher um caminho mais curto entre várias opções :

(Início)

Dados vários caminhos

_____ existirem caminhos

{

Compara dois caminhos

_____ um caminho é mais curto

{

Guarda esse caminho para comparar com o próximo

}

_____, guarda o outro caminho para comparar com o próximo

}

Obtêm-se o caminho mais curto por comparação.

(Fim)

Opções dadas: mesmas do item anterior

Texto 3:

Para dizer se um número está entre 10 e 100:

(Início)

_____ número é maior que 10 _____ número é menor que 100

{

O número é um valor entre 10 e 100.

}

{

O número não está no intervalo de 10 a 100

}

(Fim)

Opções dadas: mesmas dos itens anteriores

Parte 4: Final

Concluindo o jogo, têm-se uma breve explicação por parte do Algor sobre como estes conceitos são importantes para se aprender a programar além de uma conversação enfatizando que o jogador é um explorador deste mundo fictício em que vive e está prestes a iniciar sua jornada.

(Diálogo extraído da animação)

- Parabéns jovem aprendiz! Você agora sabe entender um algoritmo, e com isso aprenderá programação muito mais facilmente.

- Nessa nova jornada rumo ao desconhecido muitos mistérios aparecerão para você desvendar, use meus ensinamentos para lhe ajudar a resolvê-los!

- Bom, já ensinei tudo o que precisava para iniciar! Comece já a procurar novas missões e divirta-se nessa aventura! Até breve!

Fim da animação