

Resilia Educação

RESILIA

Documentação de Software
Projeto de entrega - módulo 1

Isaac Lages da Costa

Facilitadores: Jarom Aidukaitis | Anna Carolina

Março - 2022

Agradecimentos

Gostaria de agradecer de antemão todos os colaboradores, participantes do grupo e facilitadores pelo apoio prestado neste processo de desenvolvimento. Agradeço também à instituição Resilia por todo apoio dado por meio dos vários profissionais que integram a grade de ensino e apoio ao estudante.

Resumo

Este processo de documentação de software, tem como objetivo informar ao leitor todos os processos realizados entre a criação e a entrega do projeto a fim de facilitar o entendimento de tudo o que foi feito no decorrer do desenvolvimento.

Sobre o projeto

O projeto Enigmatrix, nasceu através da proposta de trabalho para encerramento de módulo, cujo o objetivo é testar e avaliar a absorção do conteúdo ministrado no módulo 1 do curso de desenvolvimento FullStacks da instituição Resilia.

Foi proposta pela instituição que cada grupo escolhesse um tema para a elaboração do projeto e que o mesmo deveria ser desenvolvido por todos em conjunto, porém, cada um com seu projeto próprio, devendo conter em todos os projetos, o mesmo tema, com no mínimo três personagens, três fases e com a história elaborada em grupo fosse seguido como um padrão para a criação deste projeto.

História

Horus, Anúbis e Bastet são divindades intergalácticas.

Há milhares de eras se instalaram no Egito e contribuíram para a construção e evolução de toda uma civilização e cultura. Atualmente, a destruição promovida pelos humanos tem desagradado as divindades e as faz querer voltar para sua origem. O caminho para o seu retorno está dentro da pirâmide de Gizé. Tudo que precisam fazer é passar pelos enigmas propostos pela esfinge.

Três etapas são necessárias para alcançar o objetivo e eles terão três chances apenas. Cada erro irá aproximar a cada um deles a permanecer no planeta Terra.

Desenvolvimento

Com base na história acima, que foi desenvolvida pela integrante Carolina de Souza Silva e adaptada pelos demais integrantes no decorrer do desenvolvimento da base do projeto, foi decidido que todo o enredo seguiria em uma direção de enigmas com suas respostas para cada fase podendo ser de forma escrita com palavras únicas ou por meio de escolhas numéricas.









Sobre o código

Utilizamos neste projeto a base inicial que foi lecionada pela instituição no módulo 1 com as linguagens HTML5, CSS3, JavaScript. Não era permitida a utilização de nenhum recurso cujo não tenha sido lecionada no período do módulo.

















Utilizamos na base do HTML, a estrutura semântica de todas as páginas foram escolhidas com base nas tags div, a, main, header, footer, ul, li para realizar a estruturação do conteúdo e sendo utilizados classes e Ids em todas as tags para que dessa forma pudesse facilitar a estilização por meio do CSS e interação com o usuário por meio do JavaScript.

Foi utilizado como forma de organização de código a separação dos arquivos em pastas, cujo no







diretório principal ficou armazenado as páginas HTML principais (index, game over, vitória) e outros diretórios para organização dos personagens, imagens, arquivos css, e javascript conforme a ilustração abaixo:

 Imagens
 anubis
 bastet
 css
 horus
 game_over.html
 index.html
 vitoria.html

No diretório imagens, está armazenado todas as imagens utilizadas no projeto, sem distinção de onde esta imagem está sendo utilizada.









 anubis.png	 bg1.jpg
 anubisNome.png	 favicon.jpg
 anubis_nome.jpg	 game_over_BG.png
 background-desktop.jpg	 gb-1.png
 bastet.png	 gb-2.png
 bastetNome.png	 horus.png
 bg-1.png	 horusNome.png
 bg-fases.jpg	 vitoria.jpg

No diretório css, estão armazenados a estilização de todas as páginas do projeto. As três fases pertencentes aos personagens, foi utilizado uma formatação padrão para todas, sendo assim não há um arquivo separado por personagem ou fase, diferente das páginas index, vitória, game over cujo cada uma possui sua estilização própria.

 fases.css
 game_over.css
 index-desktop.css
 reset.css
 vitoria-mobile.css
 vitoria.css

A estilização feita através do CSS3, utilizamos as classes e Ids atribuídas no HTML para chamar os vários pontos que precisavam ser estilizados de forma prática, utilizando mídias em diferentes tamanhos para se adaptar a responsividade.

Nos diretórios relacionados aos personagens, os arquivos estão divididos entre index, fase 2 e fase 3 e cada diretório possui uma pasta JS onde foi armazenado os arquivos javascript para a interação de cada fase.

 js	 ..
 fase2.html	 fase1.js
 fase3.html	 fase2.js
 index.html	 fase3.js

A interação com o usuário é realizada por um loop interativo cujo solicita ao jogador uma resposta em palavra única na qual é validada por uma função que utiliza if / else para levar o jogador para a próxima fase ou para as páginas de vitória e derrota de acordo com a página que o mesmo está.

De acordo com o código abaixo, o jogador terá 3 tentativas de acerto ao jogar, a cada erro o código passará pelo loop deduzindo da variável vida o total de 1 tentativa. A resposta é chamada pelo prompt cada vez que o jogador entra em uma fase clicando no botão responder e este mesmo prompt é chamado todas as vezes que o jogador erra, fazendo com que essa vida seja deduzida e apresentando uma mensagem de que a resposta está errada e informando a quantidade de erros que o usuário teve até ali.

Foi utilizado também dentro do laço de repetição a condicional **OU ||** para dar a opção ao usuário em escrever a respostas com as diferentes formas possíveis entre maiúsculo, minúsculo ou com determinada acentuação possível na resposta.

Após as 3 tentativas de acerto ou após cada resposta correta, o código passará pelo `window.location.href` que levará o mesmo para a página seguinte indiferente de qual seja.

```
1  let vida = 3
2
3  function pertunta() {
4      var resposta = prompt("Digite a sua resposta!");
5      for (var vida = 0; vida < 2; vida++) {
6          if (resposta == "morte" || resposta == "MORTE") {
7              window.location.href = "../fase2.html";
8          } else {
9              vida - 1
10             alert(`Errado! Não se preocupe, tente novamente. Este é o seu ${vida + 1}º erro`);
11
12             resposta = prompt("Digite a sua resposta!");
13         }
14     }
15     if (resposta == "morte" || resposta == "MORTE") {
16         window.location.href = "../fase2.html";
17     } else {
18         window.location.href = "../game_over.html";
19     }
20 }
```