**UNIVERSIDADE DE PERNAMBUCO – UPE**

***CAMPUS* GARANHUNS**

**LICENCIATURA EM COMPUTAÇÃO**

**ALISSON DE LIMA ALBUQUERQUE**

**ANDRILEIDE DE SOUZA SERPA**

**ELYZABETT KAROLYNA DE LIMA**

**RIELSON FEITOSA GONÇALVES**

**THIAGO FERREIRA ALMEIDA**

**RELATÓRIO DO PROJETO DA DISCIPLINA PROGRAMAÇÃO III**

**GARANHUNS**

**2019**

**SUMÁRIO**

1. **INTRODUÇÃO**
2. **DIFICULDADES**
3. **CONCLUSÃO**

**GARANHUNS**

**2019**

**INTRODUÇÃO**

Nesse projeto, o objetivo era fazer um site sobre Hogwarts e para isso tivemos que estudar bastante como programar em HTML e CSS, também pesquisar mais sobre o mundo de Hogwarts. Foi usado vários métodos de aprendizagem e nenhum participante esperava que fosse fácil, cada um sentiu dificuldade em áreas diferentes, mas todos, sem exceção, tiveram que pesquisar mais a fundo sobre como deixar a página responsiva. Mas depois de muito trabalho, conseguimos finalizá-lo.

**DIFICULDADES**

No início do projeto Thiago procurou inspiração na internet em sites bem construídos e olhou os layouts dos sites para definir qual se encaixava melhor na proposta, olhou também a paleta de cores de sites que combinavam com o tema do nosso site, olhou o código fonte de algumas páginas para se guiar em como seria a estruturação do código HTML. Então depois de pegar algumas referências, ele começou a fazer rascunhos do projeto no power point, criando várias versões de testes para poder visualizar melhor o resultado final, e depois de planejado como seria mais ou menos o layout começou a codificar.

A primeira dificuldade que ele teve foi para posicionar cada elemento dentro da página pois ele estava programando só com HTML, e só havia um pequeno pedaço em CSS, que estava dentro da própria folha do HTML, depois descobriu uma forma que conseguia posicionar os elementos do jeito que ele queria e resolveu o problema naquele momento, porém depois que ele começou a estudar como tornar a página responsiva, percebeu que não daria mais para utilizar aquele método, daí começou a reestruturar a página inteira quase do zero já visando a página responsiva.

Assim, começou a utilizar mais o CSS, com isso, ele criou uma folha dedicada apenas para o CSS. Utilizou media queries para a responsividade porque foi a primeira forma que ele encontrou, depois que terminou de fazer a página, encontrou outras formas de tornar a página responsiva. Porém ele não alterou nada no código pois já estava funcionando bem. Assim, ele conseguiu finalmente criar a página inicial completa e totalmente responsiva.

Já Rielson começou a estudar HTML pelo site “w3schools.com” e por vídeo aulas. Para ele, aprender as tags não foi muito difícil, a parte mais complicada foi usar o CSS para estilizar a página. Ele começou estudando o básico, como a sintaxe, a nomenclatura (seletor, propriedade, valor, declaração). Após isto partiu para o funcionamento do código, por exemplo, ver o que cada propriedade faz e saber os valores que ela pode receber. Nesta parte também ele viu a criação de classes e IDS.

Quando começou a usar o CSS, usou a forma interna (quando se define os estilos dentro da tag <style> que por sua vez tem que estar dentro do <head>), depois viu que era possível criar um arquivo externo somente para o CSS e depois importar ele para o código HTML através da tag <link>. Descobriu também que esta é a forma mais correta de se usar o CSS. Na sequência, deu início a construção do menu de navegação, usando uma lista não ordenada. Para deixar o menu na horizontal utilizou a propriedade display, com o valor flex, pesquisou no site w3schools.com a propriedade display e ao começar a ler percebeu que existia termos como inline-level, block-level, container flexível, que ele não conhecia. Então ele pegou o livro Fundamentos de HTML5 e CSS3 de Maurício Samy Silva e lá encontrou que elementos inline ocupam somente o espaço que ele necessita (não causa quebra de linha) e elementos nível de bloco causam quebra de linha, e ocupam toda a largura da tela.

Também encontrou a explicação para container flexível no site, lá explicava que um container flexível é um elemento que é declarado com a propriedade display e valor flex, também viu que o container flexível pode usar algumas propriedades como justify-content, align-items, flex-direction e isso o ajudou bastante na construção do menu. Mas ele achou muito difícil e demorado entender esta parte do display. Deu uma olhada na propriedade float e clear, mas decidiu utilizar containers flexíveis. A parte de responsividade foi difícil de entender, principalmente a meta tag viewport, por que ele teve que entender que existem três tipos de pixels quando estamos falando de web para smartphones:

• O pixel físico: que é a quantidade de pixels real do smartphone.

• O device pixel: que é o número de pixels que o dispositivo reporta como sendo seu tamanho (não é o tamanho real).

• CSS pixel: que é o pixel usado em HTML/CSS representando um tamanho na viewport.

Ele obtive esta explicação sobre tipos de pixel no site blog.caelum.com.br. Com isso, ele notou que a meta tag viewport , usando como valor “width=device-width” especifica que o tamanho da página será igual ao tamanho da viewport do dispositivo. Também foi usado para deixar o site responsivo, o @media, que é uma regra CSS que executa determinados códigos CSS se uma condição for verdadeira.

Elizabett começou sua página de forma livre, não usou nenhum estilo de grade, as imagens ficaram soltas e os textos centralizados, depois de refazer umas duas vezes, ela organizou em etapas, assim, ela colou primeiro as imagens e depois organizou os textos. Antes dela começar a fazer sua página, ela fez umas pesquisas e escolheu começar usando o HTML e CSS juntos, colocando os dois na mesma página, então teve que separar todo o código. Nesse processo de separação, algumas funções pararam de funcionar, daí ela refez, começando agora colocando primeiro o fundo, escolhendo uma imagem escura para combinar com o tema, e foi colocando as imagens de acordo com o conteúdo dos textos.

Para enfatizar os animais foi colocado apenas a imagem dos principais, para não sobrecarregar a página. A maior dificuldade dela não foi com os códigos, mas sim com a organização do conteúdo da página, com a organização de contêineres, e melhor posicionamento das imagens, para uma melhor disposição dos textos e ela também teve mais dificuldade em deixar a página responsiva. Ela atribuiu alguns efeitos nas imagens usando o CSS, sendo assim, quando passarmos o mouse por cima da imagem, ela se move pra cima deixando amostra o texto que tem por baixo, outro efeito usado foi o zoom, que a imagem se aproxima ao passar o mouse.

Eu(Andrileide) e Alisson, separamos nossas dificuldades da seguinte forma: A página do site em questão teve alguns pontos de dificuldade a serem tratados, destacamos a ordenação dos elementos com o uso do CSS e a responsividade da página.

**- Ordenação dos elementos com css:** essa parte teve suas dificuldades pela forma como as folhas de estilo se comportam, variando de atribuições, mudando o sentido de orientação e lógica da ordenação. Os problemas foram resolvidos com estudo, teste e compreensão dos comportamentos dos elementos e seus atributos, dentre eles os mais aplicados e testados consequentemente foram os elementos de margin, padding e position, responsáveis pela distância entre as divs e os elementos em geral, pelo preenchimento das bordas dos elementos em relação a parte interior das margins e por fim pela disposição dos elementos isolados ou ordenados em containers em relação a tela, sendo listados esses atributos respectivamente.

**- Responsividade da página:** A responsividade da tela foi separada em dois grandes grupos: as telas com resoluções de até 720 pixels de largura e as de resolução mínima de 720 pixels de largura adiante, as duas respectivamente em ordem para dispositivos menores como celulares, e-readers, tablets de tamanho menor e dispositivos semelhantes e a segunda disposição para telas de dispositivos maiores como tablets em landscape e desktops, para isso foi usada a classe @media querie para definir os dois blocos de disposição de telas e configurar os elementos dispostos no documento html de modo que se encaixassem na tela como foram imaginados.

Vale destacar a criação e disposição do menu de navegação que dependeu do uso de um id para manipulação dos elementos e de uma tag especifica chamada nav com uma lista com elementos, sendo estilizada com CSS para ocupar todo o espaço horizontal do topo da tela para referenciar os links e obter uma melhor experiência com a página em relação aos seus usuários, destacando também que a tag nav tem um significado mais especifico para os buscadores do que simplesmente um significado semântico.

Lendo e debatendo sobre a criação desse site com os outros integrantes do grupo, percebemos que no começo, todos tinham a mesma dificuldade, por não estarem acostumados a programar em HTML e saber mexer com CSS. Deixamos como mais importante, a função de deixar o site bem programado e com um visual bom de se ver. Mas também sabíamos que precisaríamos que o conteúdo fosse compreensível, então usei já a vantagem de conhecer sobre o mundo de Hogwarts e elaborei alguns textos para que fosse implementado nas páginas do site.

**CONCLUSÃO**

Concluímos juntos que houve muitas dificuldades na montagem desse site, mas cada pessoa se responsabilizou por uma página e mesmo que não conseguisse finalizar, o grupo trabalhou em equipe e o integrante que sabia mais sobre HTML, explicou e ajudou na criação de outras páginas. Foi uma tarefa bastante trabalhosa para todos, cada integrante ficou com uma função, mas todos acabaram ajudando assim que desse. Foi explicado, nesse relatório, em detalhes como foi o procedimento de inicialização e finalização das páginas do site.