

Curso Reprograme-se, IFES

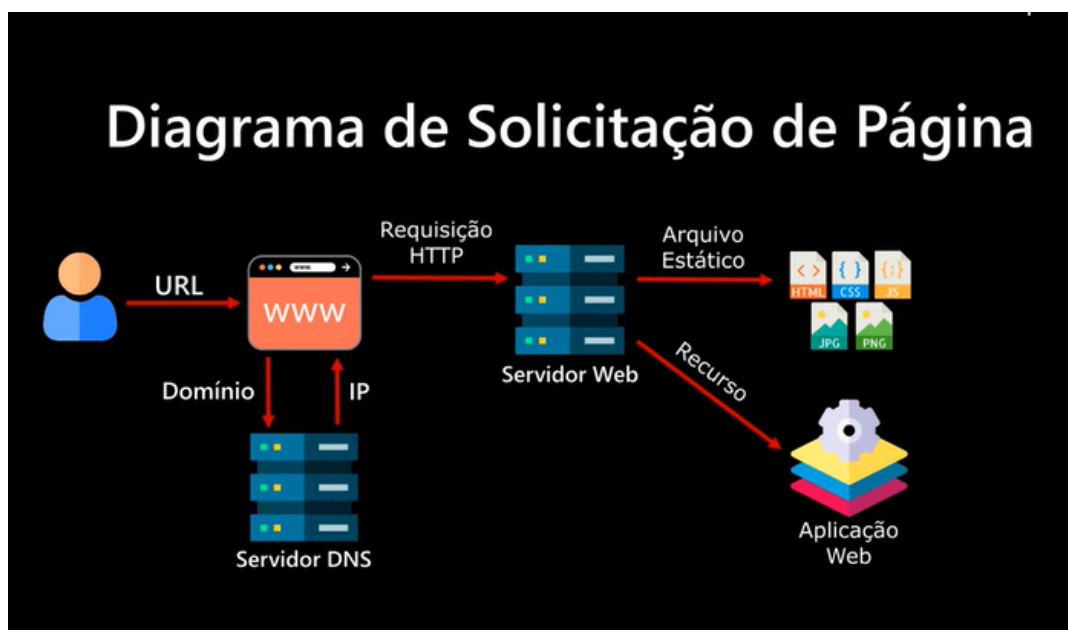
Aula Inaugural

- O curso será ministrado **online**, mas se for necessário existe **apoio presencial** em cada um dos polos
- A matrícula será liberada a partir de amanhã (27.09.2022) por email, tendo acesso a plataforma do curso

Aula 01: Princípios Básicos de Funcionamento de um Site

- Uma página é composta por um ou mais **elementos de hipertexto**, que contém textos, imagens ou elementos de multimídia.
- Uma página pode ter ligação com várias outras páginas de hipertexto através dos **hiperlinks**, dando vários caminhos de visualização do conteúdo ao usuário.
- Os **códigos** são os responsáveis por determinar e executar esses elementos em uma página;
 - Os códigos são escritos em **linguagens de programação**, que no caso das páginas web, usamos: HTML, CSS e JavaScript.
- O **HTML** (Linguagem de Marcação de Hipertexto) é responsável por definir quais elementos serão dispostos em uma página;
- O **CSS** (Folhas de Estilo em Cascata) é a linguagem de definição de estilos, determinando como os elementos serão dispostos na página;
- O **JavaScript** é uma linguagem de programação que irá executar as tarefas e eventos em uma página, tornando-a mais interativa;
- Juntas, estas três linguagens formam o **código-fonte** de uma página, que pode ser visualizado na web;
 - Esse código-fonte é distribuído em **arquivos**, normalmente divididos entre as três linguagens mencionadas, através das **extensões**:
.html, .css .js
Essa distribuição é feita para organizar o código-fonte e facilitar a manutenção do site.
- Esses arquivos podem ser **arquivos prontos** ou processados pelo programa **aplicação web**. Os arquivos prontos são aqueles baixados pelo servidor de hospedagem do site, já o programa aplicação web gera os arquivos do código-fonte toda vez que lhe é solicitado pelo usuário;
- Quando o arquivo é pronto, chamamos a página de **estática**, e quando é solicitado ao aplicação web, são páginas **dinâmicas**. Mas é muito comum termos sites com ambos tipos de páginas.
- O **número IP** (Protocolo da Internet) é uma sequência de números necessários para acessar um servidor web, identificando um endereço na internet.

- Já o **domínio** (como google.com.br) são nomes associados a esse número IP, para facilitar a navegação dos usuários na rede.
- É através do **servidor DNS** (Domain Name System) que é possível identificar o número de IP vinculado ao nome do Domínio, através de uma tabela que é atualizada com frequência.
- Já o **HTTP** (Protocolo de Transferência de Hipertexto) é o que rege e processa as solicitações de conteúdo das páginas de hipertexto ao servidor web, em síntese: carrega o conteúdo das páginas;
 - Esse conteúdo é armazenado em arquivos das três extensões mencionadas acima ou em informações encontradas em bancos de dados;
 - As consultas do HTTP aos bancos de dados são através dos **parâmetros**, que se encontram ao final de um **URL** (Uniform Resource Locator, um endereço eletrônico que permite que seu site seja encontrado na rede), usando um valor (V=) para indicar onde estão estes dados.]



Aula 02: Criação de Sites com HTML5 e CSS3

- Os **documentos HTML** possuem uma **estrutura básica**, contendo **elementos**.
- Existem **elementos container**, que guardam ainda mais elementos dentro de si - ele é delimitado por uma tag de abertura e outra de fechamento. Como por exemplo: html, head, body, title, h1 e p.
- Temos também os **elementos simples**, que não guardam outros elementos e possuem apenas a tag de abertura. Como: DOCTYPE e meta.
- Já uma **tag** corresponde a uma ou mais palavras (separadas por espaço) envolvidas por símbolos de maior e menor < >, com a primeira palavra correspondendo ao tipo de elemento HTML dessa tag.

- Nas tags de abertura, podemos ter **atributos** - que são como variáveis; como por exemplo o html em DOCTYPE na linha 1 (que define o tipo do documento). Mas muitos atributos também são acompanhados de um valor, como o lang="pt-br" em html, na linha 2 (que define o idioma do conteúdo).

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="UTF-8">
5.     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6.     <title>Título do Site</title>
7.   </head>
8.   <body>
9.     <h1>Título da Página</h1>
10.    <p>Id culpa velit deserunt ullamco veniam. Occaecat aute voluptate
11.      sint ut magna. Veniam consequat pariatur dolore eu sit ea.</p>
12.  </body>
13.</html>
```

Obs: O elemento DOCTYPE existe apenas para indicar ao navegador para indicar o tipo de documento, sempre no início do código.

- Em um código, estabelecemos uma **hierarquia** entre os elementos de pais, filhos e netos. O pai seria por exemplo o elemento DOCTYPE, seu filho o html e seus netos o head e body.
- Um documento HTML é composto por apenas um elemento html, que por sua vez só tem um elemento do tipo head e body. O elemento head tem alguns elementos de meta informações e um elemento do tipo title. O elementos body é o que contém todos elementos correspondentes ao conteúdo da página.
- Um **documento CSS** contém uma série de **seletores**, que por sua vez é composto por um **identificador** e um **bloco de propriedades** - que é delimitado por abertura e fechamento de chaves. Alguns exemplos de seletores: body, h1 e p.
- O identificador do seletor deve ser único no documento, podendo ser criado para atingir um ou vários elementos da página HTML.

```

1. body {
2.     font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
3.     padding: 50px;
4.     background-color: lightgray;
5. }
6. h1 {
7.     color: darkgray;
8.     text-align: center;
9.     border: solid 1px black;
10.}
11.p {
12.    text-align: justify;
13.    font-size: 16px;
14.}

```

- Para relacionar os associar um documento CSS a um documento HTML, basta usar um elemento html do tipo link dentro do elemento head da página, e definir o valor do atributo rel como stylesheet e o do atributo href como o nome do arquivo CSS que deseja vincular ao documento HTML. Também é comum vincular vários documentos CSS a um único documento HTML, e vice-versa.

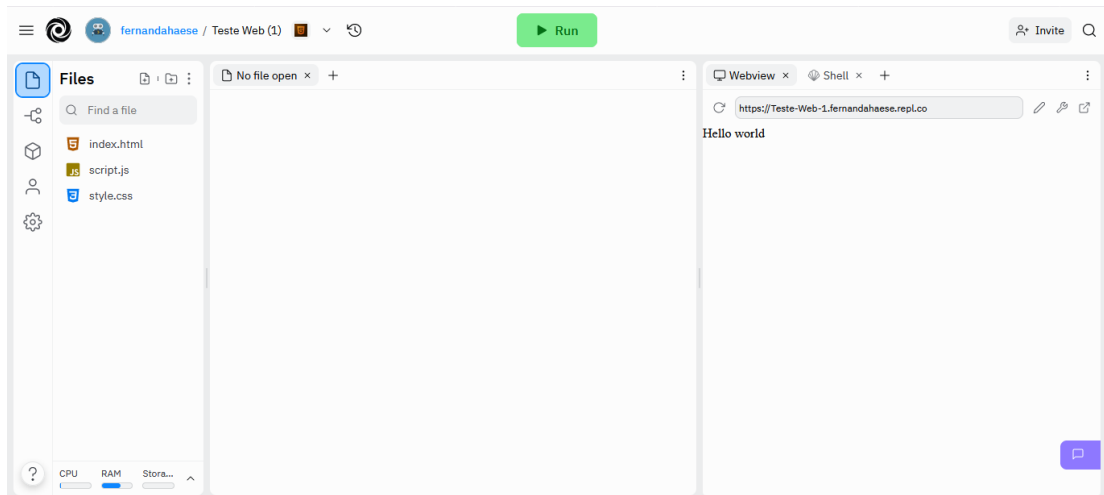
```

1. <head>
2.     ...
3.     <title>Título do Site</title>
4.     <link rel="stylesheet" href="estilos.css">
5. </head>

```

Aula 03: Editores de Código Repl.it e VSCODE

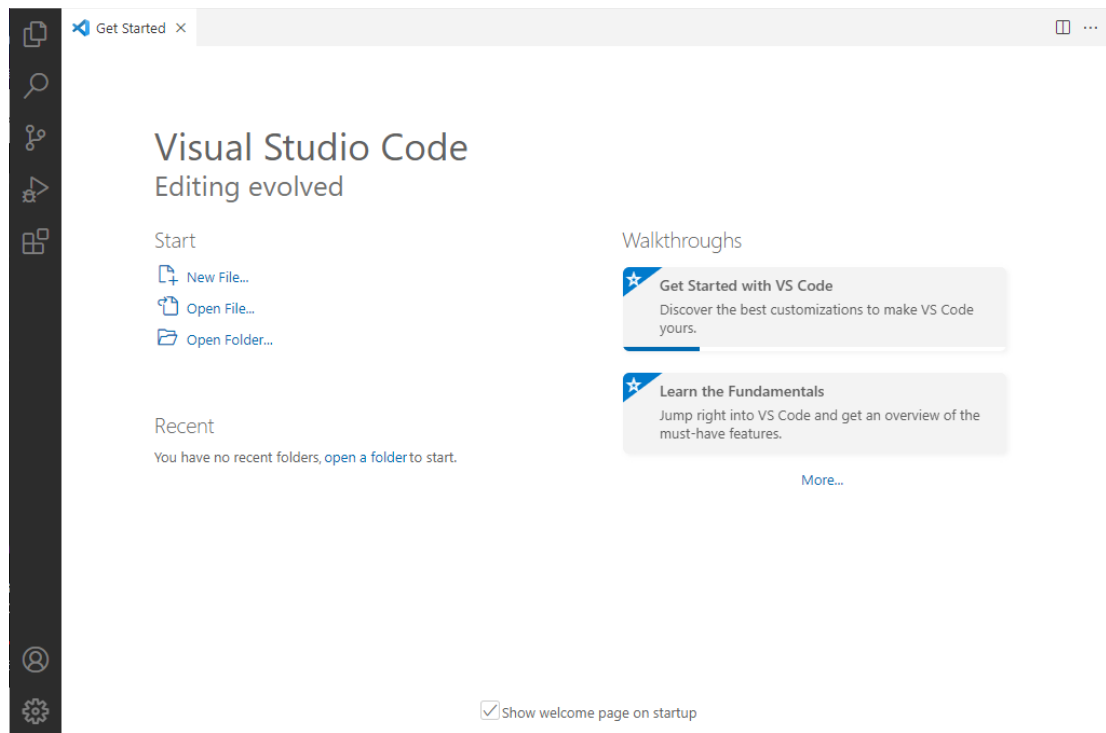
- O Replit é um editor de código que pode ser utilizado online, sem precisar de instalação do programa. Ele suporta mais de 50 linguagens de programação, colaboração em tempo real, compartilhamento de códigos e colaboração com o github.
- Para criar um novo projeto no Replit, vá para My Repls na página inicial e clique em + Create, depois informe o tipo de projeto que irá criar e defina um nome para ele. Por fim, você pode escolher se o seu projeto será público ou privado, mas a opção privada só está disponível para planos pagos.
- Na tela de editor de código do projeto, vemos:



- Na esquerda, o **painel de arquivos**: onde ficam armazenados todos os arquivos do seu projeto. Você pode renomear, abrir em uma nova aba, copiar seu link, duplicar, baixar ou deletar esses arquivos.
Dica: É uma boa prática separar o conteúdo do seu projeto de maneira organizada em vários arquivos adequados.
- No topo, o **botão Run** executa o código escrito na janela do meio, sendo mostrado na tela de visualização à direita. Essa **tela de visualização** tem duas abas: **webview** (mostra a visualização da página web) e **shell** (terminal de linha de comando que dá acesso a máquina virtual que contém seu projeto)
- Na **extremidade da esquerda, temos alguns botões**:
 - O primeiro diz respeito aos arquivos;
 - O segundo é o version control - que permite que você faça associação com o repositório do github, para fazer controle de versão do seu código-fonte;
 - O terceiro é o que te dá acesso ao instalador de pacotes que podem ser usados no seu projeto;
 - No quarto botão podemos habilitar um recurso de autenticação de usuário - para que outros usuários tenham que passar por uma tela de login com uma conta replit para entrar no seu código;
 - No último botão temos acesso a algumas configurações do editor de código: definir o layout das janelas, o tamanho da fonte, o tamanho do espaço de indentação (o recuo que o editor dá na hierarquia de elementos de um código - o mais comum são 2 ou 4 espaços), teclas de atalho do editor (emacs ou vim) , wrapping (ativar soft wrapping para quebra automática de linhas longas), coding inteligente (para ativar ou desativar o mecanismo de auxílio de escrita de código) ou habilitar a acessibilidade ao terminal de saída (para deficientes visuais).
- Na extremidade inferior esquerda no painel de arquivos, temos três barras que indicam o progresso de: consumo do processador, de

memória RAM e de armazenamento, relacionados ao computador virtual que roda esses projetos no Replit.

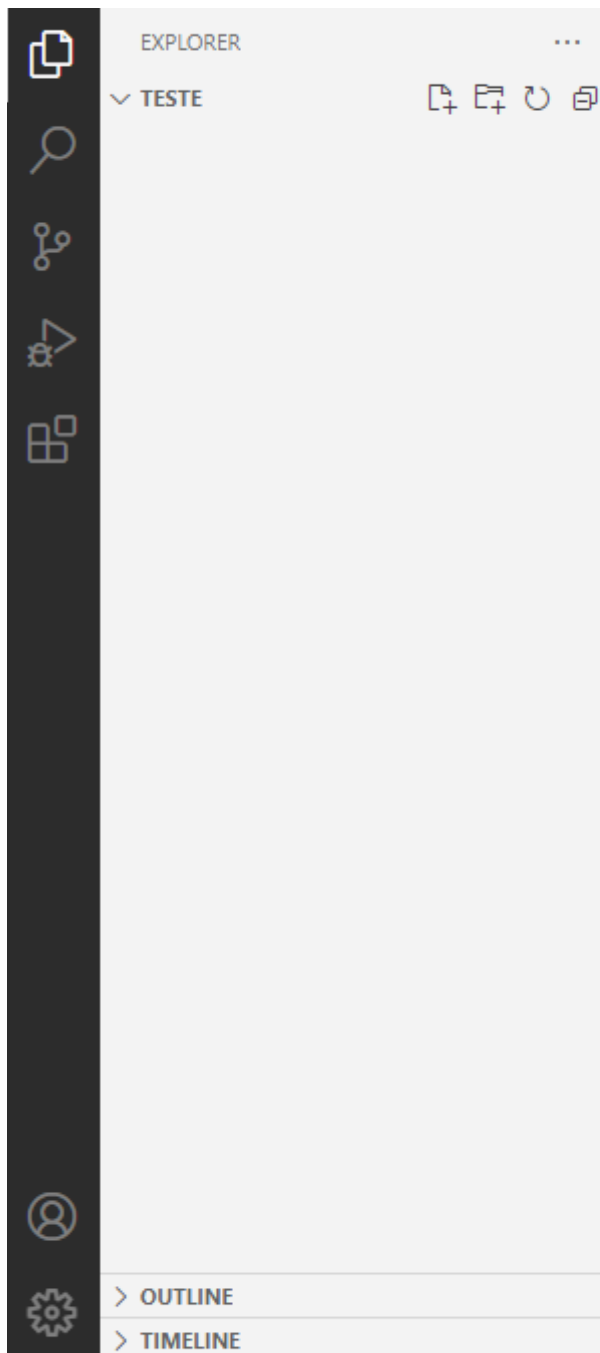
- Também temos disponíveis vários modelos de projetos que podem servir como ponto de partida para o seu projeto, em diversos modelos e linguagens de programação. Se encontram todos em **Templates**, no menu.
- Ainda no **menu**, temos outras disponibilidades como: **My Repls** (onde estão seus projetos criados), **My Cycles** (para habilitar aplicações pagas), **Community** (Contato com a comunidade de usuários do Replit), **Learn** (sessão de aprendizagem), **Teams** (para você gerenciar projetos de alunos, amigos e outras formas de trabalho colaborativo) e **Curriculum** (seleções de projetos nas mais projetos em diversas linguagem, como trilhas de aprendizagem para programação).
- Se você tem uma conta de professor, você também pode gerar uma turma e definir atividades para seus alunos, com várias opções interessantes de interação.
- o **VSCODE** ou Visual Studio Code é um dos editores de código mais populares atualmente (2022). É bastante consagrado pela comunidade de programações, leve e de código aberto, com todas funcionalidades para codificar e visualizar seus arquivos em diversas linguagens de programação.
- Para baixar e rodar o VSCODE, é necessário ter os requisitos de hardware recomendados: processador de 1.6GHz, 1GB de RAM e 500 MB de armazenamento. No caso de ambiente Windows: pelo menos um processador de 2 núcleos com mais de 2GHz cada, 4GB de RAM e 1GB de armazenamento
- Temos a seguir a **página inicial** ou tela de boas-vindas do programa:



Nela, podemos criar ou abrir um arquivo, pasta ou clonar um repositório git. Também existem ao lado direito alguns recursos de aprendizagem.

- Na extremidade esquerda, temos cinco botões: ao clicar no primeiro deles podemos abrir pastas e arquivos ao selecionar **open folder**.
Obs.: Ao abrir, será feita uma checagem de segurança da sua pasta, apenas confirme.
- O **segundo com o símbolo da lupa**, para buscar e substituir textos dentro dos seus arquivos de código. Você também pode definir quais arquivos pesquisar e quais não pesquisar.
- No terceiro botão, podemos usar os principais recursos de controle de versão baseada em Git.
- No quarto botão, temos recursos de depuração e execução de programas, algo mais utilizado em linguagens de programação mais tradicionais como java, python e C#.
- No **quinto e último**, é a janela de gerenciamento de extensões, que podem auxiliar no processo de criação e visualização de códigos. Por exemplo: A extensão HTML CSS Support, que adiciona o recurso de completamento de código automático para as classes de estilo CSS em um documento HTML; A extensão Live Server, que adiciona um servidor web ao seu vscode e permite que você abra suas páginas HTML como se estivessem publicadas realmente - importante para a testagem no processo de criação. Por fim temos uma extensão que melhora a forma que você vê os arquivos do projeto no explorador de arquivos: a extensão

- No canto inferior esquerdo, temos um ícone de usuário, em que podemos ativar a sincronização de configurações, como atalhos de teclado, extensões e outros.
 - Por fim, temos abaixo com o ícone de engrenagem, acesso a uma série de configurações: `command palette` dá acesso a barra de comandos, `settings` mostra a janela de configurações principais do vscode, `extensions` para acessar o gerenciador de extensões, `keyboard shortcuts` para configurar atalhos de teclado, `migrate keyboard shortcuts from` para espelhar os atalhos de teclado igual a outro programa, `configure user snippets` para adicionar trechos de código nomeado para usar enquanto codifica - no caso de trechos que são repetido com frequência em seu código, `color theme` para temas de cores da interface do vscode, `file icon theme` para escolher o tema de ícones do explorador de arquivos, `product icon theme` para mudar os ícones da interface do vscode, `turn of settings sync` sobre as configurações de sincronização, `manage workplace trust` para gerenciar as pastas de trabalho confiáveis e por fim `check for updates` para checar novas atualizações.
- Ao gerar a nova pasta, aparecerá um `explorador de arquivos`, e ao passar o mouse sobre ele, surgem os quatro botões do topo:

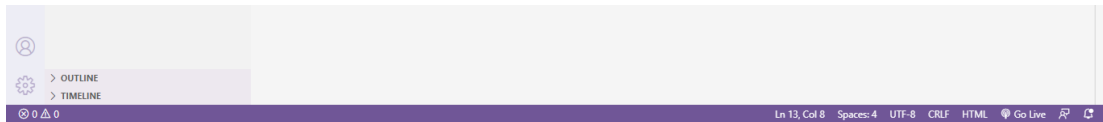


- Ao clicar no primeiro botão no topo, criamos um novo arquivo. Ao nomeá-lo de index.html, styles.css ou outros, será gerada uma nova aba no editor para trabalhar no código do arquivo.
- Ao clicar com o botão direito do mouse em cima do arquivo, surgem mais algumas opções do que fazer com o arquivo. Temos: open to side para dois editores lado a lado, open with para abrir com uma aplicação externa, reveal in file explorer para revelar onde o arquivo está no explorador de arquivos do sistema operacional, open in integrated terminal pro terminal ser aberto na pasta que está aberto o arquivo, select for compare para ser comparado com outros arquivos, open timeline para ver o histórico de atualizações do arquivo, cut e copy

para copiar ou cortar, `copy path` e `copy relative path` para copiar o caminho direto ou relativo do arquivo, e por fim `rename` e `delete` para renomear ou deletar.

→ No `segundo botão no topo`, podemos criar uma pasta dentro da pasta de trabalho, por exemplo uma pasta `img` para armazenar imagens utilizadas em um site.

- Abaixo temos a `barra de status`, em que os ícones no canto esquerdo indicam a quantidade de erros encontrados no projeto, e no outro canto algumas informações sobre o código e documento atual: sua linha atual, tamanho do espaço de indentação, tipo de codificação usada, caractere usado como quebra de linha, a linguagem que está sendo usada, opção `go live` - usada quando você instala a extensão live server, para visualizar o resultado de uma página html no navegador (com atualização simultânea das alterações feitas, indicado abrir ao lado do vscode para checar seu progresso), ícone de feedback ao vscode e sino de notificações.



- Alguns `atalhos do vscode` para acelerar o processo de codificação:
 - Para aumentar e diminuir o zoom na aba do editor de código, pressione no teclado: `ctrl +` ou `ctrl -`
 - Pressione no teclado `Ctrl+] / [` para adicionar ou remover um espaço de indentação
 - Clique no teclado `Shift+Alt+F` para formatar o documento
 - Comando `Ctrl+N` para abrir um novo arquivo
 - Pressione `Ctrl+B` para mostrar ou ocultar a barra lateral
 - Comando `Ctrl+`` para mostrar o terminal
 - `Ctrl+S` para salvar o documento. Você também pode ativar o `autosave` no menu.
 - Usar `abreviações`, por exemplo: uma exclamação `!` no editor de código html. Existem várias, busque no google para saber mais: `emmet abbreviation vscode`