

Objetivos:

- Introdução ao software necessário para a disciplina. Live Server. Browser console debug.

Programa:

- Tratando-se de uma unidade curricular de “Programação”, e sendo o primeiro laboratório, é da máxima importância que o aluno configure o seu ambiente de desenvolvimento. Para além dessa tarefa, a correta utilização das ferramentas de *debugging* existentes no *browser* é também considerada essencial para um *developer* dos dias de hoje.

Implementação:

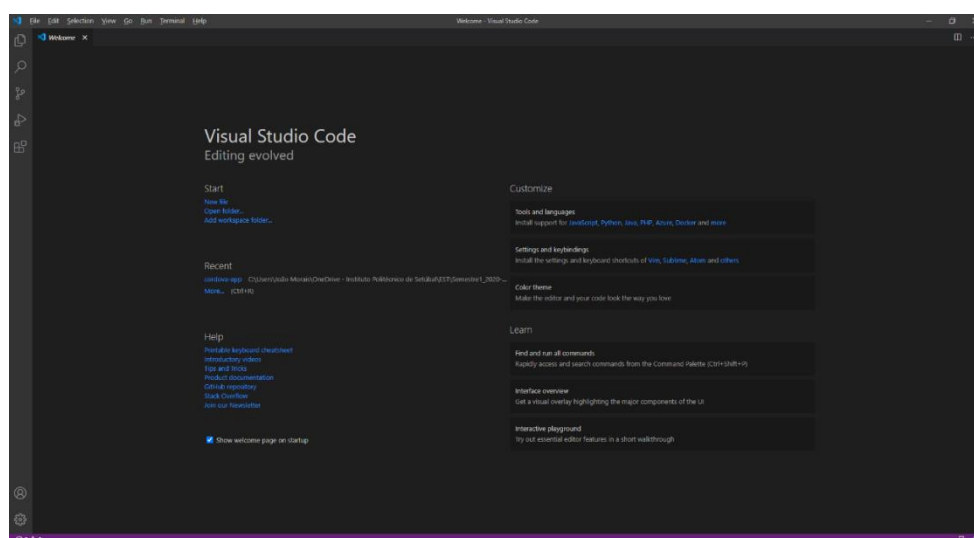
O Editor de Desenvolvimento

Para manipular os nossos ficheiros de HTML, CSS, JavaScript, etc., iremos utilizar o Visual Studio Code (VSCode). <http://code.visualstudio.com/docs/editor/whyvscode>.

Instalador e Primeira execução

<http://code.visualstudio.com/Download>

Após completar a instalação, poderá executar a aplicação:








Nesta janela:

- “Menu Principal”, no topo;

- “Barra de Visualização de Ferramentas”, à esquerda;
- “Zona de Edição de Documentos”, ocupando a quase totalidade da janela (e que na figura está preenchida pela janela de “Welcome”, que poderá ser fechada);
- “Barra de Estado”, na base da janela;

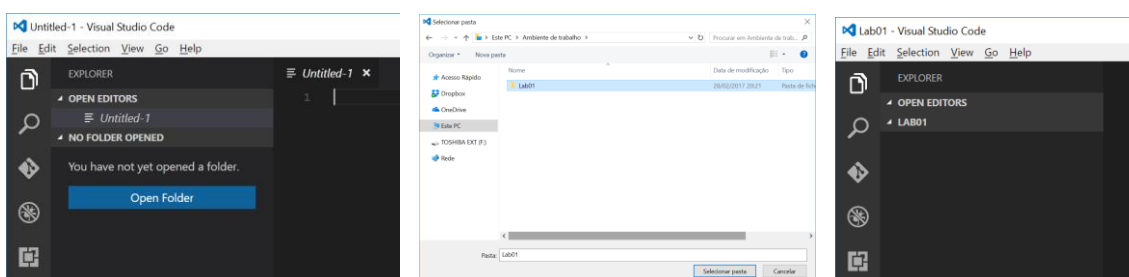
De salientar que na “Barra de Visualização de Ferramentas” podemos encontrar:


- Explorador de Ficheiros (Ctrl+Shift+E): 
- Pesquisa nos Documentos (Ctrl+Shift+F): 
- Gestão de Versões git (Ctrl+Shift+G): 
- Debugging (Ctrl+Shift+D): 
- Extensões (Ctrl+Shift+X): 

Configuração

Aplicações *web* são compostas por vários ficheiros (HTML, CSS, JavaScript, código de servidor, etc.). É conveniente que estes ficheiros se encontrem organizados. Estas pastas deverão ser utilizadas como “espaço de trabalho” (*workspace*), facilitando a sua gestão. Para abrir o *workspace* no VSCode, use o botão “Open Folder” do “Explorador de Ficheiros”.

Crie uma pasta chamada “Lab00” e selecione a mesma usando a funcionalidade “Selecionar pasta” (de acordo com as seguintes imagens).



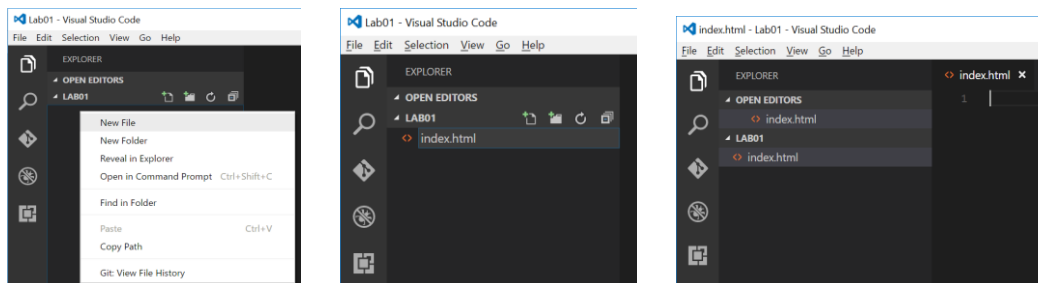
Será dentro desta pasta Lab00 (*workspace*) que se deverão criar ficheiros e outras subpastas. Tal é possível usando o menu de contexto ou os botões disponibilizados na barra , tal como pode ser observado pelas imagens seguintes.

Desenvolvimento Web

2022/2023 – LEI, LIG

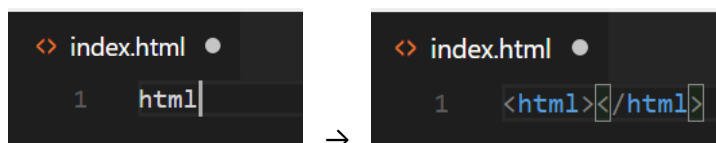
Trabalho 0 de Laboratório

0



Edição de Texto

O VSCode permite a utilização de abreviaturas *Emmet* (<http://docs.emmet.io/>) para a maioria das linguagens (HTML, CSS, XML, etc.). Exemplo, num ficheiro HTML basta escrever o nome da *tag* e usar a tecla Tab para se obter a notação HTML:



O sistema *Emmet* tem uma sintaxe própria bastante poderosa (um resumo das suas potencialidades pode ser obtido em <http://docs.emmet.io/cheat-sheet/>).

Na “Zona de Edição de Documentos” é possível editar até 3 documentos em simultâneo. A edição de um ficheiro numa secção nova é possível seleccionando-o no “Explorador de Ficheiros” através do rato e seguidamente com as teclas **Ctrl+I**.

Servidor Web

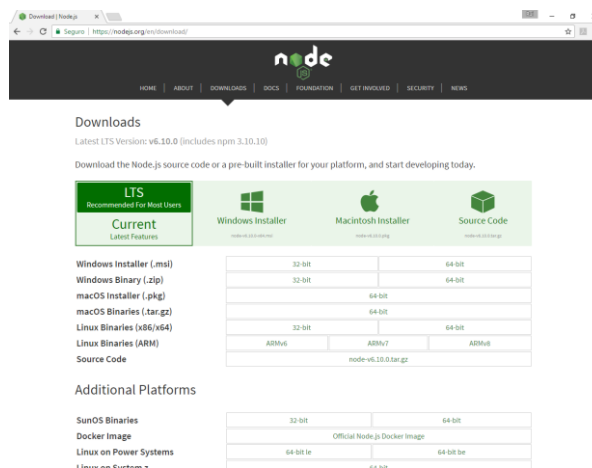
Para que um site fique disponível aos utilizadores através do browser, além do conjunto de ficheiros HTML, teremos de possuir software de servidor *web*. Este tipo de software é caracterizado por estar à escuta de pedidos dos vários clientes (browsers), e retornar os recursos solicitados através da *web* (**Esta temática será detalhada em aulas futuras**).

Seguidamente será instalado um servidor *web* simples (apenas disponibiliza ficheiros estáticos), como uma extensão do VSCode.

Em aulas futuras iremos desenvolver um servidor web simples utilizando o VSCode e Node.js , onde vamos poder analisar e perceber o respetivo código.

Node.js

Comece por instalar o ambiente Node.js. Este é disponibilizado em: <https://nodejs.org/en/download>



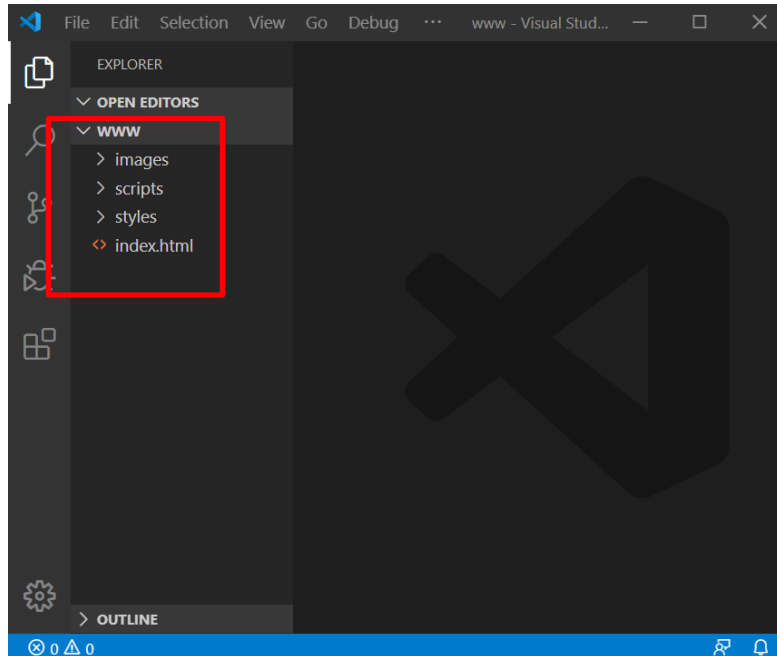
Tendo o Node.js instalado na máquina, podemos começar a desenvolver aplicações usando JavaScript. Assim, teremos ficheiros com código que será executado no servidor e ficheiros que serão enviados para o cliente (*Browser*) para lá serem executados e visualizados. Convém criar uma separação entre eles, para que o objetivo de cada um fique claro.

Criação do Ambiente de Trabalho

Os ficheiros (HTML, CSS, JavaScript, imagens, etc.) serão todos colocados dentro de uma pasta chamada *www* (seguindo a nomenclatura do servidor *web* Apache). Fora dessa pasta, iremos colocar no futuro os ficheiros da aplicação do servidor (e.g., *app.js*, *router.js*, ..., etc.), neste caso, como vamos utilizar a extensão *Live Server* não é necessário colocar nenhum ficheiro na na “raiz” do *workspace*.

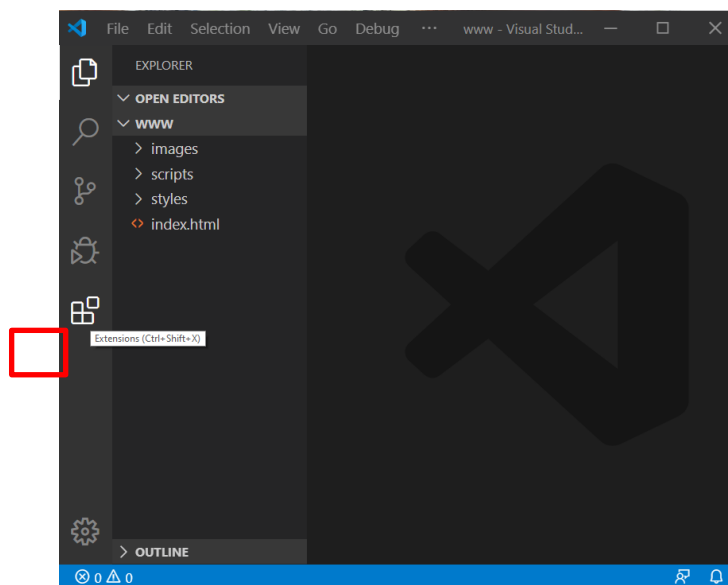
Assim, deve-se criar uma pasta *www* e mover o ficheiro *index.html* para o interior. Nela podem ser criadas as diferentes subpastas (*images*, *styles* e *scripts*) que irão armazenar os diferentes tipos de ficheiros.

No editor VSCode aparecerá de acordo com a imagem seguinte:



Instalação do Live Server

Para instalar extensões no VSCode é necessário aceder ao gestor das extensões. Para isso basta clicarem no ícone assinalado na figura abaixo:



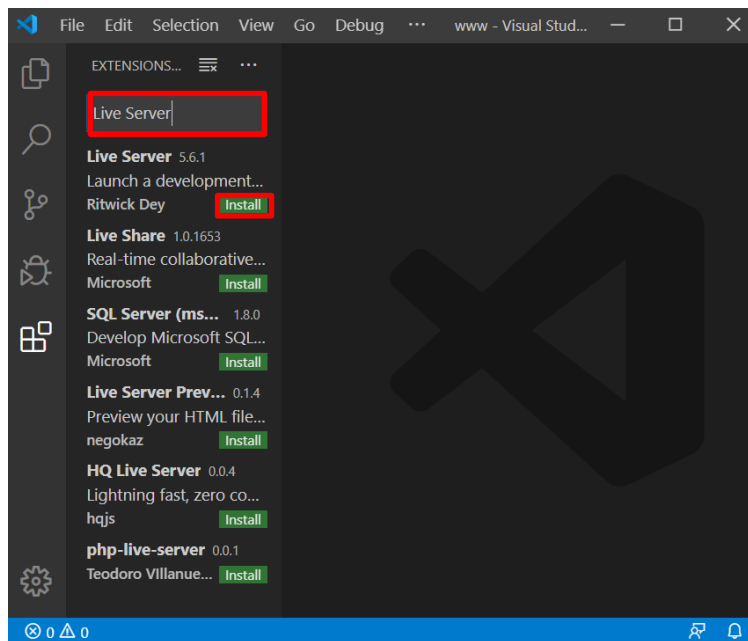
Desenvolvimento Web

2022/2023 – LEI, LIG

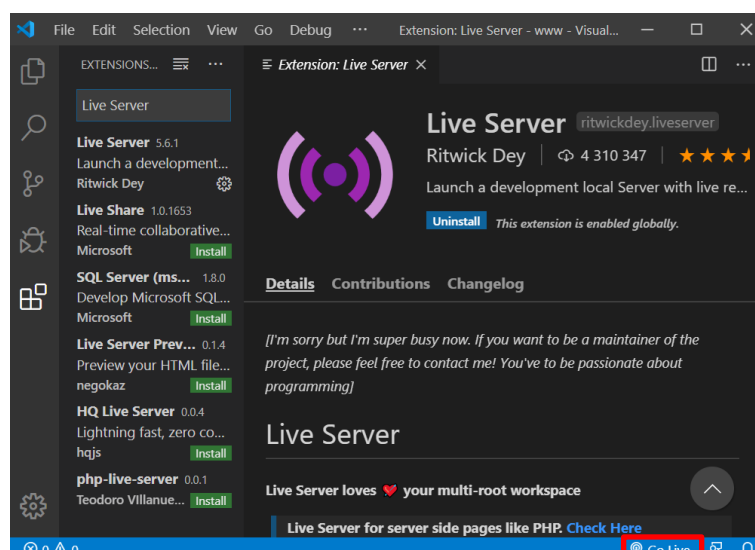
Trabalho 0 de Laboratório

0

Após clicarem é necessário escrever na caixa de texto, Live Server, depois escolher a opção “install” para instalar o Live Server 5.6.1 como se pode verificar na figura abaixo:



Após a instalação com sucesso, deverá de visualizar o ícone do Live Server no VSCode como se pode verificar na figura abaixo:



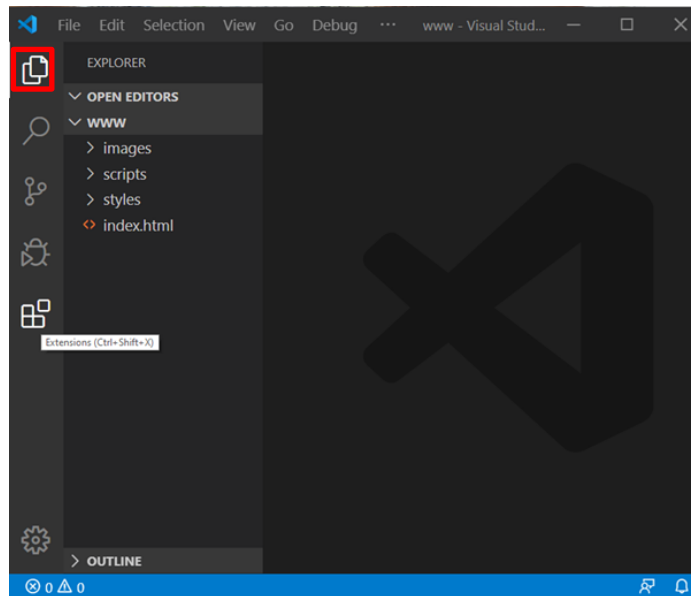
Desenvolvimento Web

2022/2023 – LEI, LIG

Trabalho 0 de Laboratório

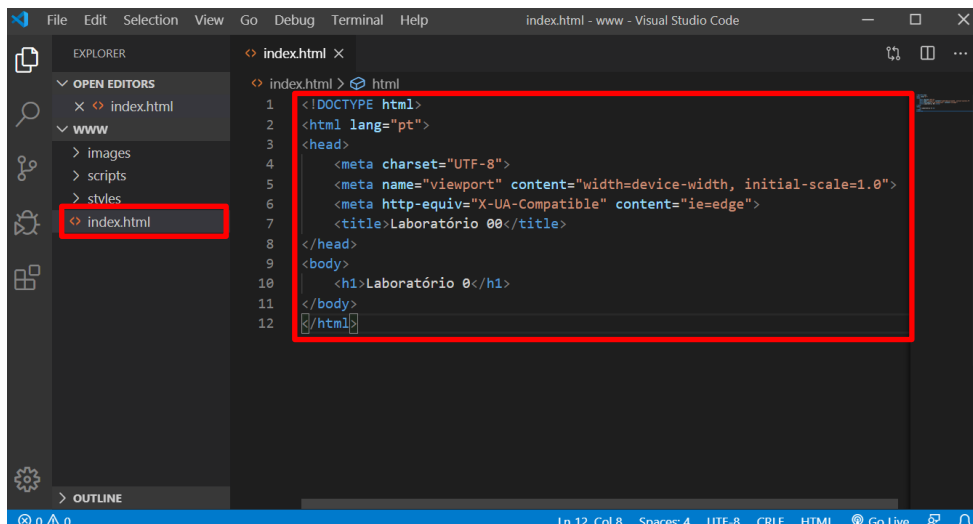
0

Antes de começarmos a utilizar o Live Server, vamos regressar à vista das pastas do nosso projeto, para isso basta clicar no respetivo ícone no VSCode, como se pode verificar na figura abaixo:



Utilização do Live Server

Uma vez que estamos na vista das pastas do nosso projeto, vamos clicar no ficheiro index.html e introduzir o seguinte código HTML que podem consultar na figura abaixo.



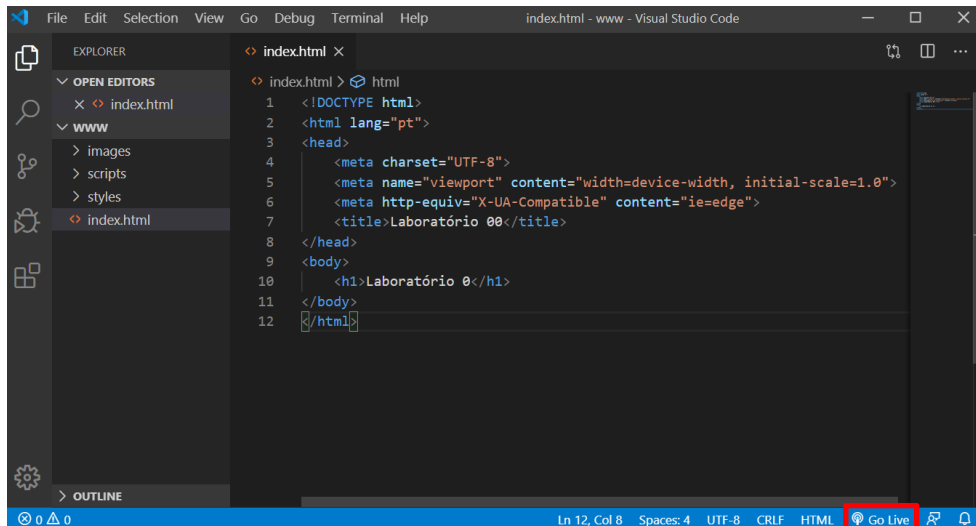
Desenvolvimento Web

2022/2023 – LEI, LIG

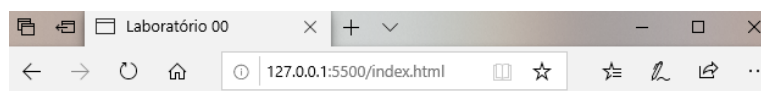
Trabalho 0 de Laboratório

0

Após gravar as alterações no index.html, e para executar o mesmo deverá de clicar no ícone do live-server, como se pode verificar na figura abaixo:



Logo após clicar no Live Server, deverá de aparecer a seguinte página no seu “browser”, como se pode verificar na figura abaixo.



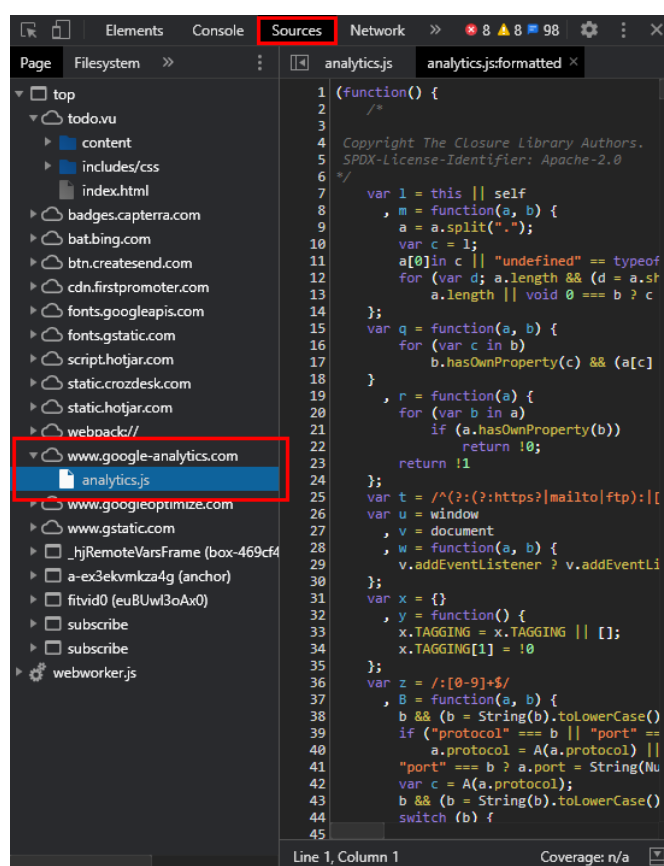
Laboratório 0

Qualquer alteração efetuada no ficheiro index.html, após a sua gravação, o Live Server faz um refrescamento e mostra a informação, baseado nas últimas alterações efetuadas.

A pasta Lab00 será a nossa base de trabalho para a maioria dos laboratórios seguintes.

Debug no browser

Nos *browsers* mais recentes existe a possibilidade de fazermos debug ao código javascript de uma determinada página. Recorrendo para este exemplo ao Google Chrome, podemos encontrar esta funcionalidade no separador Sources (Fontes) como demonstrado na figura abaixo. Uma vez escolhido esse separador é possível seleccionar qual o ficheiro .js sobre o qual queremos efectuar o debug.



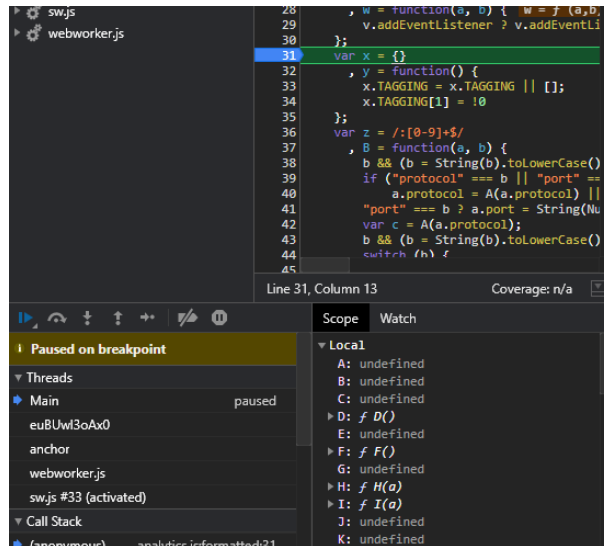
Depois de seleccionado o ficheiro, é também possível escolher a linha onde deve parar o processamento da página e verificar os valores pretendidos, como se pode ver no exemplo seguinte:

Desenvolvimento Web

2022/2023 – LEI, LIG

Trabalho 0 de Laboratório

0



Para um debug mais informativo, é sempre possível utilizar a consola presente no browser (separador Console) para que através do comando `console.log()`, aplicado no código, seja possível definir uma mensagem escrita.

