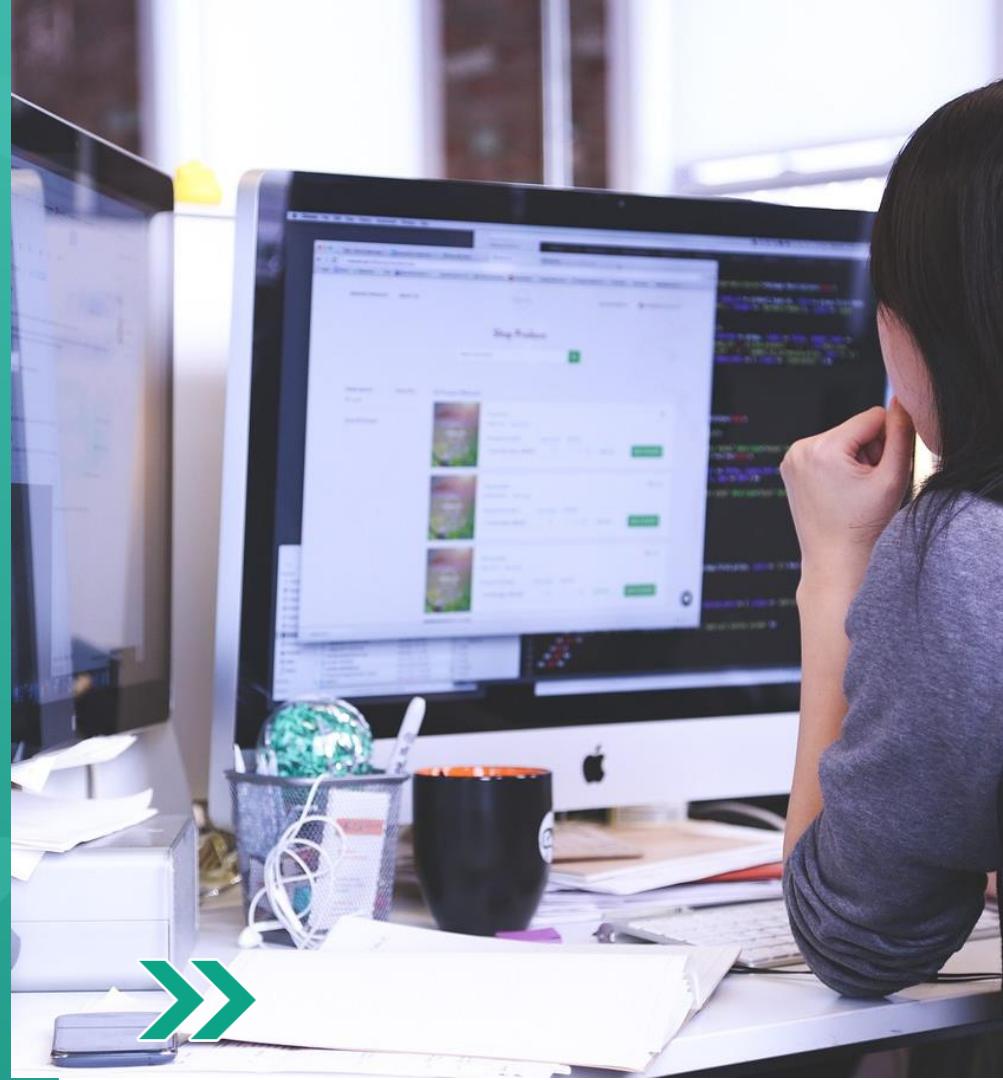


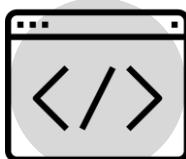
# Profissão: Engenheiro Front-End



# BOAS PRÁTICAS



# Introdução ao SASS



Confira boas práticas da  
comunidade de Front-End por  
assunto relacionado às aulas.

● **Crie tarefas**

● **Execute tarefas  
em série**

● **Execute tarefas  
em paralelo**

● **Use Gulp para  
compilar SASS**

● **Minifique JS  
com o Gulp**



# Crie tarefas

Acompanhe exemplos de aplicação do Gulp no dia a dia.



## Minificação de arquivos:

O Gulp pode ser usado para minificar arquivos CSS, JavaScript e até mesmo HTML, removendo espaços em branco, comentários e reduzindo o tamanho dos arquivos, o que melhora o desempenho do carregamento do site.



## Concatenação de arquivos:

O Gulp pode ser usado para combinar vários arquivos CSS ou JavaScript em um único arquivo, reduzindo o número de solicitações feitas ao servidor e melhorando o desempenho do site.



## Compilação de pré-processadores CSS:

Se você está usando pré-processadores CSS, como Sass ou Less, o Gulp pode ser configurado para compilar esses arquivos em CSS puro, automatizando o processo e permitindo que você escreva seu código em uma linguagem mais poderosa.

# Crie tarefas

Acompanhe exemplos de aplicação do Gulp no dia a dia.



## Otimização de imagens:

Com o Gulp, você pode automatizar a compressão e otimização de imagens, reduzindo o tamanho dos arquivos de imagem sem perda significativa de qualidade. Isso ajuda a melhorar o tempo de carregamento das páginas.



## Recarregamento automático do navegador:

Durante o desenvolvimento, o Gulp pode ser configurado para monitorar as alterações nos arquivos e recarregar automaticamente o navegador quando uma alteração for detectada, agilizando o processo de visualização das alterações feitas no código.



## Transpilação de código:

Se você está escrevendo código em uma linguagem que precisa ser transpilada, como TypeScript ou ES6+ para JavaScript, o Gulp pode ser usado para automatizar o processo de transpilação, garantindo que o código seja compatível com os navegadores mais antigos.

# Crie tarefas



## ES Modules:

Com os ES Modules, você pode usar as palavras-chave import e export para trabalhar com módulos. A palavra-chave import é usada para importar funcionalidades de outros módulos, enquanto a palavra-chave export é usada para exportar funcionalidades de um módulo para que possam ser importadas por outros módulos.

# Execute tarefas em série

Acompanhe dicas sobre a execução de tarefas em série com o Gulp.



## Defina a tarefa padrão:

Ao criar um fluxo de tarefas em série, você pode definir uma tarefa padrão que será executada quando você digitar apenas gulp no terminal, sem nenhum argumento adicional.



## Trabalhe com callbacks ou promises:

Se você tiver tarefas assíncronas, pode utilizar callbacks ou retornar Promises para indicar quando cada tarefa é concluída. Se estiver usando Promises, você pode utilizar a função `gulp.series()` normalmente, pois o Gulp espera até que todas as Promises sejam resolvidas antes de seguir para a próxima tarefa.



## Trate erros e exceções:

Certifique-se de tratar erros e exceções corretamente em cada tarefa, para evitar que o fluxo seja interrompido caso ocorra um erro. Você pode usar try-catch em tarefas síncronas ou encadeamento de Promises para capturar e lidar com erros em tarefas assíncronas.

# Execute tarefas em série

Acompanhe dicas sobre a execução de tarefas em série com o Gulp.



## Utilize módulos do Gulp:

Se você tiver muitas tarefas e quiser organizar melhor o código, pode usar módulos do Gulp. Cada módulo pode exportar uma função que será uma tarefa. Em seguida, você pode usar a função `gulp.series()` para executar várias tarefas em série.

# Execute tarefas em paralelo

Acompanhe dicas sobre a execução de tarefas em paralelo com o Gulp.



## **Aproveite o poder do paralelismo:**

Ao executar tarefas em paralelo, lembre-se de que o Gulp aproveitará o poder do paralelismo para executar as tarefas simultaneamente. Isso é especialmente útil quando você tem tarefas que podem ser executadas independentemente umas das outras, como compilação de arquivos CSS e JavaScript, por exemplo.



## **Controle de ordem:**

Apesar de serem executadas em paralelo, é importante lembrar que nem sempre a ordem de execução será garantida. Se a ordem de execução for importante, considere usar as tarefas em série ou configurar dependências explícitas entre as tarefas para garantir a ordem correta.



## **Evite conflitos:**

Certifique-se de que as tarefas em paralelo não criem conflitos em arquivos de saída ou em outros recursos compartilhados. Por exemplo, se várias tarefas tentarem gravar em um mesmo arquivo, podem ocorrer problemas de sobreposição.

# Execute tarefas em paralelo

Acompanhe dicas sobre a execução de tarefas em paralelo com o Gulp.



## Aproveite o máximo do hardware:

O paralelismo do Gulp é uma ótima forma de aproveitar o máximo do *hardware* do seu computador, executando várias tarefas ao mesmo tempo e melhorando a velocidade do processo de *build*.

# Use Gulp para compilar SASS

Acompanhe dicas sobre a  
compilação de SASS com o Gulp.



## Notificação de erros:

É útil adicionar uma notificação de erro quando a compilação do Sass falha. Para isso, você pode usar o pacote gulp-notify. Ele exibirá mensagens de erro no terminal e pode ser integrado facilmente ao Gulp.



## Organização de pastas:

Você pode organizar melhor seus arquivos CSS compilados e minificados, colocando-os em pastas específicas, como dist/css/compiled para os arquivos compilados e dist/css/minified para os arquivos minificados.



## Autoprefixer:

Para garantir a compatibilidade com diferentes navegadores, você pode adicionar o Autoprefixer aos seus arquivos CSS.

# Minifique JS com o Gulp

Acompanhe dicas sobre a  
minificação JS com o Gulp.



## Concatenar arquivos:

Antes de minificar os arquivos JavaScript, você pode considerar a opção de concatená-los em um único arquivo. A concatenação pode ajudar a reduzir o número de requisições HTTP e melhorar o desempenho do carregamento da página.



## Excluir arquivos não minificáveis:

Nem todos os arquivos JS precisam ser minificados. Alguns podem ser bibliotecas de terceiros que já estão minificadas ou arquivos de desenvolvimento. Nesses casos, você pode excluir esses arquivos do processo de minificação, para evitar erros e manter o código original. Sourcemaps: Quando você minifica seus arquivos JavaScript, o código resultante pode ficar difícil de depurar. Os sourcemaps são úteis para mapear o código minificado de volta para o código original, facilitando a depuração no navegador.

# Minifique JS com o Gulp

Acompanhe dicas sobre a  
minificação JS com o Gulp.



## Modo de produção:

Se você deseja executar tarefas específicas apenas no modo de produção, você pode utilizar variáveis de ambiente ou um pacote como o gulp-if para condicionar a execução das tarefas.



## Utilize pacotes alternativos:

Além do gulp-uglify, você pode considerar outros pacotes de minificação, como o terser ou babel-minify, que podem oferecer opções adicionais de otimização e melhor compatibilidade com versões mais recentes do JavaScript.

# Bons estudos!

