

Programação em JavaScript

ServiceNow - Deloitte

Rodrigo Costa

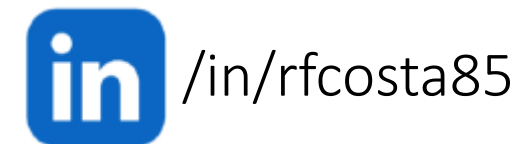
Apresentações

Sobre mim

Engenheiro Informático com sólida experiência em desenvolvimento web, especializado em Front-end, com destaque para Angular e ecossistemas JavaScript/TypeScript. Ao longo da minha carreira, tenho desenhado e implementado interfaces dinâmicas e escaláveis, integrando-as com APIs e serviços empresariais, sempre com foco na qualidade do código, boas práticas e experiência do utilizador.



Rodrigo Costa



rodrigo.costa@innominatum.pt

Apresentação dos formandos

- Seu nome;
- Motivação em participar do curso;

Apresentação do módulo

Dotar os formandos dos conhecimentos fundamentais do desenvolvimento web, permitindo-lhes compreender e aplicar os princípios essenciais de versionamento de código, estruturação de repositórios, e os fundamentos de HTML5 e CSS3, com foco em semântica, acessibilidade, layout e responsividade.

O que vamos aprender

- Conceitos essenciais de versionamento de código e de repositório em git e GitHub;
- Fundamentos de HTML e CSS.

Estrutura do módulo

- Sessão 1 – Ambiente, Ferramentas e Introdução ao git e Github;
- Sessão 2 – HTML5 Semântico e Acessível;
- Sessão 3 – Fundamentos de CSS3 e Flexbox;
- Sessão 4 – CSS Grid e Layout Final.

Cronograma – Módulo fundamentos web

Quinta – Feira	Sexta – Feira	Segunda – Feira	Terça – Feira
22/01/2025	23/01/2025	26/01/2025	27/01/2025
09:00 – 13:00 14:00 – 17:00	09:00 – 13:00 14:00 – 17:00	09:00 – 13:00 14:00 – 17:00	09:00 – 13:00 14:00 – 17:00

Fundamentos Web

Módulo 1

Ambiente, Ferramentas e Introdução ao Git e GitHub

Sessão 1

Objetivo geral

Dotar os formandos dos conhecimentos fundamentais do git e GitHub, permitindo-lhes compreender a sintaxe básica do git, a importância do versionamento e do repositório de código .

Objetivos específicos

Ao final da sessão, os formandos deverão ser capazes de:

- Executar comandos básicos no terminal para navegação, criação e edição simples de ficheiros e pastas;
- Instalar e configurar o git num ambiente local de desenvolvimento;
- Aplicar o fluxo básico de versionamento com git (init, add, commit) utilizando boas práticas de commit (Conventional Commit).

Ambiente de Desenvolvimento (VSCode)

O que é um ambiente de desenvolvimento?

Um ambiente de desenvolvimento é um conjunto de:

- Ferramentas;
- Configurações;
- Programas;
- Extensões.

VSCode

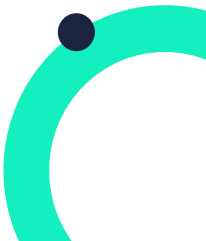
- Extensões a serem instaladas:
 - Live Server -> servidor local para ver páginas web;
 - ESLint -> ajuda a manter o código limpo;
 - Prettier -> formatação automática de código.

Terminal Integrado



O que é terminal?

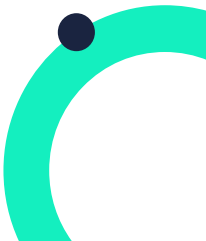
- O terminal é uma interface de texto para comunicar diretamente com o sistema operativo.
- Permite executar comandos para:
 - Navegar entre pastas;
 - Criar e modificar ficheiros;
 - Executar programas e ferramentas.





Terminal integrado no VSCode

- O VS Code possui um terminal embutido que permite:
 - Escrever código;
 - Executar comandos;
 - Utilizar o git;
- Vantagens:
 - Menos alternância entre aplicações;
 - Contexto sempre na pasta do projeto;
 - Melhor produtividade;



Abrir o terminal no VS Code

- Existem várias formas:
 - Menu;
 - Terminal -> Novo terminal;
 - Atalho de teclado:
 - Windows / Linux: `Ctrl + j`;
 - macOS: `Cmd + `` (crase).

Comandos essenciais para navegação e gestão de ficheiros

Onde estou? (conceito fundamental de diretório)

- Conceito fundamental:
 - No terminal, estamos sempre dentro de uma pasta;
 - Essa pasta chama-se diretório atual;
 - Todos os comandos atuam a partir desse local;
 - Antes de fazer qualquer coisa, é importante saber onde estou no sistema de ficheiros.

Pwd – Mostrar o diretório atual

- Comando: pwd
 - Mostra o caminho completo do diretório atual;
 - Funciona em:
 - Linux;
 - MacOS;
 - Git bash / WSL no Windows.

Ls / dir – Listar ficheiros e pastas

- Listar conteúdo do diretório

Sistema	Comando
Linux / MacOS	ls
Windows (CMD / PowerShell)	dir
Windows (Git Bash)	ls

cd – Mudar o diretório

- Comando: cd (change directory)
 - Permite entrar em outra pasta:
 - cd nome-da-pasta.
 - Permite voltar atrás para outra pasta:
 - cd ..
 - Permite ir para a pasta pessoal (home):
 - cd ~
- Funciona em:
 - Linux;
 - macOS;
 - Git Bash / WSL;
 - PowerShell.

Referências especiais importantes

Símbolo	Significado
.	Diretório atual
..	Diretório pai
~	Pasta pessoal (home)

Comandos para limpar o terminal

Sistema	Comando
Linux / MacOS	clear
Windows (CMD / PowerShell)	cls
VS Code (Git Bash)	clear

Caminho relativo x caminho absoluto

- Caminho relativo:
 - Depende de onde estás no momento:
 - `cd projetos/fundamentos-web`
- Caminho absoluto:
 - Começa na raiz do sistema:
 - `cd /home/user/projetos/fundamentos-web`
 - No Windows:
 - `cd C:\Users\Rodrigo\projetos`

O que é uma pasta (diretório)

- Uma pasta (ou diretório) serve para:
 - Organizar ficheiros;
 - Estruturar projetos;
 - Separar responsabilidades.

Criar pastas no terminal

- Comando: mkdir
 - mkdir significa make directory;
 - Serve para criar novas pastas;
- Mkdir nome-da-pasta;
- Funciona em todos os sistemas operativos.

Criar pastas no terminal

- Criar várias pastas de uma vez:
 - `mkdir sessao-01 sessao-02 sessao-03;`
- Criar pastas dentro de outras pastas:
 - `mkdir projetos/fundamentos-web;`
- Criar estrutura completa (-p):
 - `mkdir -p projetos/fundamentos-web/sessao-01.`

O que é um ficheiro?

- Um ficheiro é onde guardamos informação:
 - código;
 - texto;
 - configurações.

Criar ficheiros no terminal (Unix / Git Bash)

- Comando: touch
 - Cria um ficheiro vazio;
 - Não escreve conteúdo;
 - Funciona em:
 - Linux;
 - macOS;
 - Git Bash / WSL no Windows.
- Comando: touch
 - touch index.html
 - new-item index.html
 - No CMD e PowerShell;
 - Vários ficheiros de uma vez:
 - touch index.html style.css script.js
(sem suporte no cmd);

Edição simples de ficheiros com nano

- O que é o nano?
 - Editor de texto no terminal ideal para:
 - Pequenas edições;
 - Editar ficheiros de configuração;
- Restrição:
 - Não está disponível no CMD / PowerShell.

Comandos básicos do nano

- nano index.html -> Se o ficheiro existir, abre para edição, se não existir, cria o ficheiro e abre ele;
- O símbolo ^ significa Ctrl:
 - ^O -> Ctrl + O.
- Edição feita toda pelo teclado:
 - Utilize as setas para navegar;
 - Backspace / Delete para apagar;
- Guardar o ficheiro:
 - Ctrl + O;
 - Confirma o nome do ficheiro (Enter).
 - Ctrl + x para sair do ficheiro.

Git - Versionamento

O que é versionamento de código e por que é essencial

- Versionamento é o processo de:
 - Guardar versões do código;
 - Acompanhar alterações;
 - Recuperar estados anteriores;
- Funciona como um histórico do projeto.

Git entra em cena

- git:
 - É uma ferramenta de **versionamento de código**;
 - Funciona via **terminal**;
 - Acompanha todas as alterações feitas nos ficheiros;

Git trabalha com pastas e ficheiros

- Git não substitui o terminal;
- Git trabalha sobre a estrutura que já criamos;
- Ele observa:
 - Ficheiros;
 - Pasta;
 - Alterações.

O que o git faz na prática?

- Guarda versões do projeto:
- Regista quem mudou o que e quando;
- Permite desenvolvimento de novas features sem medo de perder a versão estável do sistema;
- Funciona como um backup inteligente.

GitHub (remoto e colaboração)

Git ≠ GitHub

- Atenção:
 - Git não é GitHub;
 - São diferentes, porém complementares;
- GitHub:
 - Plataforma online que:
 - Hospeda repositórios git;
 - Permite colaboração;
 - Facilita partilha de código.
 - Funciona como repositório remoto;
- GitHub utiliza o git por baixo.

O que é um repositório?

- É uma pasta monitorizada pelo git que contém:
 - Ficheiros do projeto;
 - Histórico de alterações.
- Podemos dizer que um projeto é um repositório



Git init

- Comando git init:
 - Inicializa um repositório git;
 - Cria uma pasta oculta .git;
- Quando digitamos git init dentro da pasta que queremos versionar, permitimos que o git comece a observar toda e qualquer alteração dentro desta pasta.



Git status e add

- git status:
 - Mostra o **estado atual do repositório**;
 - Informa:
 - Quais ficheiros foram alterados;
 - Quais ainda não estão versionados;
 - O que está pronto (ou não) para commit.
- git add:
 - Indica ao git que gostaria de guardar as alterações atuais do ficheiro;
 - Possui vertentes de utilização:
 - git add . -> Adiciona todas as alterações da pasta atual;
 - git add <nome-do-ficheiro> -> Adiciona as alterações somente deste ficheiro.

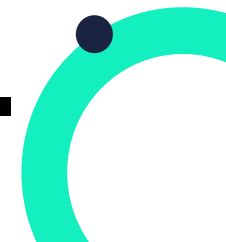
Git commit

- Comando git commit:
 - Um commit é uma versão guardada do projeto e contém:
 - Estado dos ficheiros;
 - Mensagem descritiva;
 - Data e autor.
- Comando básico:
 - git commit -m “mensagem”
 - git commit -m “feat: adicionar estrutura inicial do projeto”.
- Extra: Conventional Commit – [Link](#) – leitura importante.



Git log

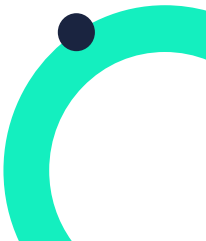
- Comando git log:
 - Mostra o histórico de commits e apresenta:
 - Identificador do commit (hash);
 - autor;
 - Data;
 - Mensagem.
- Comando básico:
 - git log --oneline;
 - git log -p.





Branch

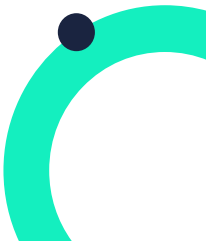
- Uma branch é uma linha paralela de desenvolvimento que permite:
 - Testar ideias;
 - Desenvolver funcionalidades;
 - Evitar conflitos e erros graves;
 - Proteger a versão principal.





Branch principal

- Branch main:
 - É a branch principal do projeto;
 - Contém o código mais estável;
 - Normalmente é a versão principal;
 - Antigamente chamava-se master.



Comandos essenciais – branch

- git branch: Mostra as branches existentes;
- git checkout -b <nome-da-branch>: Cria uma nova branch;
- git checkout <nome-da-branch>: Muda para outra branch;

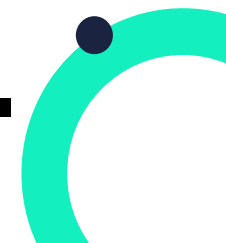
Repositório local vs remoto, git push e pull

- Local -> repositório no seu computador;
- Remoto -> repositório no GitHub;
- git push -> envia commits locais para o repositório remoto;
 - git push origin main -> Aqui tornamos o código disponível online, ou seja no repositório remoto.
- git pull -> Traz alterações do GitHub para o computador
 - git pull origin main -> Aqui atualizamos o nosso repositório local, com o repositório remoto.



Síntese

- Ferramentas
 - VSCode, terminal integrado, git e github;
- Fundamentos:
 - Terminal:
 - Comandos básicos de criação de ficheiros e pastas e navegação;
 - Git;
 - GitHub;
 - Repositório.



Conclusão

Programação em JavaScript

ServiceNow - Deloitte

Rodrigo Costa