Domínio do VS Code — Guia de Engenharia

1. Visão Geral

O VS Code é um editor de código leve porém poderoso, com suporte a linguagens variadas, controle de versão, debugging, extensões e integração remota. Visual Studio Code
Como engenheiro responsável, você deve saber configurar, personalizar e utilizar-o como ambiente central de desenvolvimento para o projeto (front-end, back-end, integração contínua) da plataforma.

2. Instalação & Configuração Inicial

- Instalar versão apropriada para o sistema operacional (Windows/macOS/Linux). Visual Studio Code
- Verificar componentes adicionais (por exemplo, extensões para .NET, Node, TypeScript) conforme stack do projeto.
- Configurar **Settings Sync** se for trabalhar em múltiplas máquinas. Visual Studio Code
- Personalizar tema, layout, atalhos para produtividade (bom para padronizar equipe).

3. Interface & Navegação

- Conhecer a **bar lateral** (Explorer/Source Control/Extensions).
- Entender **Multi-root Workspaces** se o projeto envolver múltiplos repositórios. Visual Studio Code
- Usar painel de terminal integrado para builds, testes e deploys.
- Saber usar comandos rápidos (Ctrl+P / Cmd+P) para navegação de arquivos, símbolos, comandos.

4. Edição de Código

- Habilitar **IntelliSense** (autocompletar) para linguagens usadas (ex: JavaScript/TypeScript, C#, Node). Visual Studio Code
- Usar **Refactoring** e navegação (Go to Definition, Find References) para manter código limpo.
- Configurar **Snippets** personalizados e atalhos para templates frequentes. Visual Studio Code
- Aplicar formatação automática e linting (via extensões) para consistência de código.

5. Controle de Versão (Git)

- Utilizar integração Git embutida. Visual Studio Code
- Executar operações como commit, branch, merge, checkout diretamente no VS Code.
- Configurar pré-commit hooks ou integração com CI/CD para padronizar o fluxo.
- Bons logs de commit, convenções de branch (ex: feature/*, hotfix/*) e review de código.

6. Debugging & Testes

- Dominar o painel de Debug (breakpoints, watch, call stack). Visual Studio Code
- Configurar ambientes específicos (ex: Node, .NET) com launch.json.
- Integrar testes unitários e de integração no VS Code e visualizar cobertura.
- Automatizar tarefas (Tasks) para build/test/deploy. Visual Studio Code

7. Extensões & Marketplace

- Selecionar extensões essenciais: por exemplo, ESLint, Prettier, C#, Docker, Live Share.
- Garantir runtime security das extensões (área de segurança do VS Code). Visual Studio Code
- Padronizar arquivo .vscode/extensions.json para equipe sincronizar extensões recomendadas.

8. Remote Development & Containers

- Habilitar **Dev Containers** ou Remote SSH/WSL se usar ambiente isolado ou cloud. Visual Studio Code
- Garantir que o ambiente de execução seja idêntico em todos os dispositivos da equipe (evita “funciona na minha máquina”).
- Conhecer a configuração devcontainer.json para definir dependências.

9. Configurações & Perfis

- Personalizar settings.json para definições compartilhadas (ex: tab size, indentação, formatter).

- Criar **Profiles** para diferentes contextos (ex: Front-end, Back-end). Visual Studio Code
- Ativar **Workspace Trust** para segurança quando abrir projetos externos. Visual Studio Code

10. Performance & Manutenção

- Monitorar Telemetry e performance do editor (evitar lentidão com extensões pesadas). Visual Studio Code
- Realizar atualizações regulares e gerir versões "Insiders" se testarem funcionalidades novas.
- Backup das configurações e extensões compartilhadas em equipe.

---

11. Estratégia de Adoção no Projeto RAWN PRO

- Definir template de projeto no VS Code com diretórios padrão (ex: src/, api/, docs/).
- Criar e versionar configuração comum (.vscode folder) contendo launch.json, settings.json, extensions.json.
- Treinar equipe com padrão de uso: edição em VS Code, testes integrados, commit via VS Code.
- Automatizar tarefas de build/test/deploy acionadas por Tasks no VS Code.
- Integrar com GitHub e Vercel para deploy contínuo, aproveitando terminal integrado e extensão GitHub.
- Monitorar qualidade de código com linting, formatação e companhia de extensões no VS Code.

---

12. Checklist de Domínio para o Engenheiro

- Instalação do VS Code na máquina de desenvolvimento e servidor.
- Configuração de sincronização de settings.
- Criação de Workspace padrão e multi-root se necessário.
- Instalação e configuração de extensões recomendadas.
- Template de tasks para build, test, deploy.
- Integração Git com práticas de commit/branch.

- Setup de dev container ou Remote SSH para ambiente isolado.
- Configuração de debugging para front-end e back-end.
- Documentação interna de uso do VS Code para equipe.
- Monitoramento de performance e atualização periódica.