

## 05 - Formatting

2025-10-03 22:22 pm

---

- Formatação demonstração cuidado com o código. - Regras devem ser consistentes. - Objetivo é comunicação, não rigidez. **\*\*FORMATAÇÃO VERTICAL\*\***

- Qual o tamanho ideal de um arquivo?
  - De 200 a 500 linhas parece bom, mas depende.
- Arquivos menores são mais fáceis de entender.
- Pensar código como um jornal
  - Não ter nenhum artigo muito longo cheio de detalhes.
  - Separar em seções (arquivos)
- Linhas em branco ajudam formatação e separação entre conceitos.
- Blocos visuais de código devem indicar unidades lógicas no código.
- Afinidade conceitual:
  - Conceitos próximos entre si devem estar verticalmente próximos entre si.
  - Não devem existir em outros arquivos se são próximos.
  - Evitar que leitor salte muito entre pedaços de código.
- Variáveis:
  - Variável deve ser declarada o mais perto de seu uso possível.
    - No topo de uma função (que deve ser curta por sua vez)
    - Variável de loop deve ser declarada adjacente ao loop
  - Atributos de instâncias devem existir no topo da classe.
    - Algumas convenções colocam na parte mais embaixo.
    - Importante é ser padronizado.
- Funções dependentes:
  - Funções que chamam uma à outra devem existir próximas.
  - Em sequência, se possível. Do alto nível ao baixo nível.

### FORMATAÇÃO HORIZONTAL

- Linhas curtas são melhores.
- Há vários limites sugeridos: 80, 100, 120.
  - Mais de 120 é excessivo.
- Densidade:
  - Espaço para dissociar componentes e não-espço para indicar coesão
  - Atribuição de variável deve ter espaço em branco
  - Funções não devem ter espaço entre primeiro parêntese e argumento.
  - Operações matemáticas também deve ter espaço para clareza.
- Alinhamento:
  - Alguns padrões usam alinhamento em colunas (com vários espaços em branco para isso) de termos coesos:

```

public FitNesseExpediter(Socket s,
                           FitNesseContext context) throws Exception
{
    this.context = context;
    socket = s;
    input = s.getInputStream();
    output = s.getOutputStream();
    requestParsingTimeLimit = 10000;
}

```

- 
- Mas é melhor não fazer isso.
  - Melhor deixar desalinhado mesmo (i.e. um só espaço em branco separando os elementos na linha).
- Se for uma lista longa de elementos que peçam alinhamento, o problema é o tamanho da lista, não o desalinhamento.
- Indentação:
  - Serve para explicitar hierarquia de scopes.
    - e.g. o que está fora e o que está dentro de um if/else.
  - É tentador violar a indentação para funções/loops curtos e deixar numa linha só.
    - Isso é uma má ideia.
    - É melhor ter indentação mesmo assim.

## REGRAS DO TIME

- Formatação envolve preferências
- Estilo deve ser acordado e ser único para o time.