

# Referência Rápida Conventional Commits

Formato de mensagem de commit, tipos, escopos, breaking changes

## Formato

### Estrutura da Mensagem de Commit

```
<type>[optional scope]: <description>
[optional body]
[optional footer(s)]
```

### Explicação das Partes

<b>type</b>	Categoria da alteração (feat, fix, etc.)
<b>scope</b>	Parte do código afetada (opcional)
<b>description</b>	Resumo curto no modo imperativo
<b>body</b>	Explicação detalhada (opcional, após linha em branco)
<b>footer</b>	Metadados como BREAKING CHANGE ou refs de issues

### Regras

<b>Imperative mood</b>	"add feature" não "added feature"
<b>Lowercase type</b>	feat: não Feat:
<b>No period</b>	Descrição não termina com ""
<b>Blank line</b>	Separar body/footer da descrição

### Tipos

#### Tipos Principais (relevantes para SemVer)

<b>feat</b>	Nova funcionalidade (dispara bump de versão MINOR)
<b>fix</b>	Correção de bug (dispara bump de versão PATCH)

#### Tipos Estendidos (Convenção Comum)

<b>build</b>	Sistema de build ou dependências externas
<b>chore</b>	Tarefas de manutenção (sem alterações em src/test)
<b>ci</b>	Configuração e scripts de CI
<b>docs</b>	Alterações somente em documentação
<b>perf</b>	Melhoria de performance
<b>refactor</b>	Alteração de código que não corrige nem adiciona
<b>revert</b>	Reverte um commit anterior
<b>style</b>	Formatação, espaços em branco (não estilo CSS)
<b>test</b>	Adicionar ou corrigir testes

### Escopo

#### Uso de Escopo

```
feat(auth): add OAuth2 login flow
fix(parser): handle empty input gracefully
docs(readme): update installation steps
refactor(api): extract validation middleware
```

#### Diretrizes de Escopo

<b>Module name</b>	feat(auth); fix(parser):
<b>Layer name</b>	feat(api); fix(db):
<b>Feature area</b>	feat(search); fix(checkout):
<b>Dependency</b>	build(deps); chore(npm):
<b>Omit if broad</b>	Use sem escopo para alterações transversais

### Breaking Changes

#### Marcando Breaking Changes

```
feat!: remove deprecated login endpoint
feat(api)!: change response format to JSON:API
fix!: drop Node 14 support
```

#### Breaking Change no Footer

```
feat(api): change user endpoint response
BREAKING CHANGE: response now returns array
instead of object. Update client parsing.
```

### Regras

<b>! after type/scope</b>	Marcador abreviado de breaking change
<b>BREAKING CHANGE:</b>	Token no footer (sempre em maiúsculas)
<b>BREAKING-CHANGE:</b>	Também válido (forma com hífen)
<b>SemVer impact</b>	Dispara bump de versão MAJOR
<b>Any type</b>	Breaking changes podem aplicar a qualquer tipo

### Exemplos

#### Commits Simples

```
feat: add email notifications
fix: prevent race condition in checkout
docs: correct typo in contributing guide
style: format with prettier
refactor: simplify error handling logic
```

#### Com Escopo

```
feat(blog): add comment threading
fix(auth): refresh token before expiry
test(api): add missing edge case coverage
ci(github): add Node 20 to test matrix
```

#### Com Body e Footer

```
fix(parser): handle nested quotes correctly

Previously, nested quotes caused the parser
to enter an infinite loop. This adds a depth
counter to prevent unbounded recursion.

Closes #234
```

### Footer

#### Tokens de Footer

<b>BREAKING CHANGE:</b>	Descreve breaking change na API
<b>Closes #123</b>	Fechar issue automaticamente ao merge
<b>Fixes #456</b>	Fechar issue automaticamente (referência de fix)
<b>Refs #789</b>	Referenciar issue sem fechar
<b>Reviewed-by: name</b>	Atribuição de revisor
<b>Co-authored-by: name</b>	Atribuição de co-autor

#### Múltiplos Footers

```
feat(api)!: redesign authentication flow

Migrate from session-based to JWT auth.

BREAKING CHANGE: /auth endpoints changed
Closes #101
Refs #98, #99
```

### Ferramentas

#### Lint de Commits

```
npm install -D @commitlint/cli \
  @commitlint/config-conventional
echo "module.exports = { extends: \
  ['@commitlint/config-conventional'] }" \
  > commitlint.config.js
```

#### Ferramentas Populares

<b>commitlint</b>	Lint de mensagens de commit conforme convenção
<b>husky</b>	Git hooks (executar commitlint no commit)
<b>commitizen (cz)</b>	Construtor interativo de mensagens de commit
<b>standard-version</b>	Changelog automático + bump de versão
<b>semantic-release</b>	Pipeline de release totalmente automatizado
<b>release-please</b>	Ferramenta de automação de release do Google

### Configuração do Commitizen

```
npm install -D commitizen \
  cz-conventional-changelog
npx commitizen init cz-conventional-changelog
# Use: npx cz (or git cz with alias)
```

### Padrões Comuns

#### Mapeamento de Bump de Versão

<b>fix:</b>	PATCH (1.0.0 → 1.0.1)
<b>feat:</b>	MINOR (1.0.0 → 1.1.0)
<b>BREAKING CHANGE</b>	MAJOR (1.0.0 → 2.0.0)
<b>docs:, style:, etc.</b>	Sem bump de versão

#### Agrupamento de Changelog

```
## [1.2.0] - 2026-03-27
### Features
- add email notifications (abc1234)
### Bug Fixes
- prevent race condition (#123) (def5678)
```

#### Formato de Revert

```
revert: feat(blog): add comment threading
This reverts commit abc1234def5678.
```