

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

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

Regras

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

Tipos

Tipos Principais (relevantes para SemVer)

feat	Nova funcionalidade (dispara bump de versão MINOR)
fix	Correção de bug (dispara bump de versão PATCH)

Tipos Estendidos (Convenção Comum)

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

Module name	feat(auth); fix(parser):
Layer name	feat(api); fix(db):
Feature area	feat(search); fix(checkout):
Dependency	build(deps); chore(npm):
Omit if broad	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

! after type/scope	Marcador abreviado de breaking change
BREAKING CHANGE:	Token no footer (sempre em maiúsculas)
BREAKING-CHANGE:	Também válido (forma com hífen)
SemVer impact	Dispara bump de versão MAJOR
Any type	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

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

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

fix:	PATCH (1.0.0 → 1.0.1)
feat:	MINOR (1.0.0 → 1.1.0)
BREAKING CHANGE	MAJOR (1.0.0 → 2.0.0)
docs:, style:, etc.	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.
```