

Commits Semânticos

1. **build:** Alterações que afetam o sistema de construção ou dependências externas (escopos de exemplo: gulp, broccoli, npm),
2. **ci:** Changes to our CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, SauceLabs);
3. **docs:** referem-se a inclusão ou alteração somente de arquivos de documentação;
4. **feat:** Tratam adições de novas funcionalidades ou de quaisquer outras novas implantações ao código;
5. **fix:** Essencialmente definem o tratamento de correções de bugs;
6. **perf:** Uma alteração de código que melhora o desempenho;
7. **refactor:** Tipo utilizado em quaisquer mudanças que sejam executados no código, porém não alterem a funcionalidade final da tarefa impactada;
8. **style:** Alterações referentes a formatações na apresentação do código que não afetam o significado do código, como por exemplo: espaço em branco, formatação, ponto e vírgula ausente etc.);
9. **test:** Adicionando testes ausentes ou corrigindo testes existentes nos processos de testes automatizados (TDD);
10. **chore:** Atualização de tarefas que não ocasionam alteração no código de produção, mas mudanças de ferramentas, mudanças de configuração e bibliotecas que realmente não entram em produção;
11. **env:** basicamente utilizado na descrição de modificações ou adições em arquivos de configuração em processos e métodos de integração contínua (CI), como parâmetros em arquivos de configuração de containers.

Como usar:

git commit -m "feat(header): Adicionar funcionalidade de autenticação de usuário"