

CCF 322 - Engenharia de Software II

Padrões de Commit GitHub

“Reinados da Web”

<Versão 1.0>

Esses serão os padrões de commit que deverão ser seguidos por todos os membros da equipe. Esse conjunto de regras têm como objetivo padronizar e facilitar o entendimento dos commits feitos no GitHub. É necessário a compreensão e a utilização das regras abaixo.

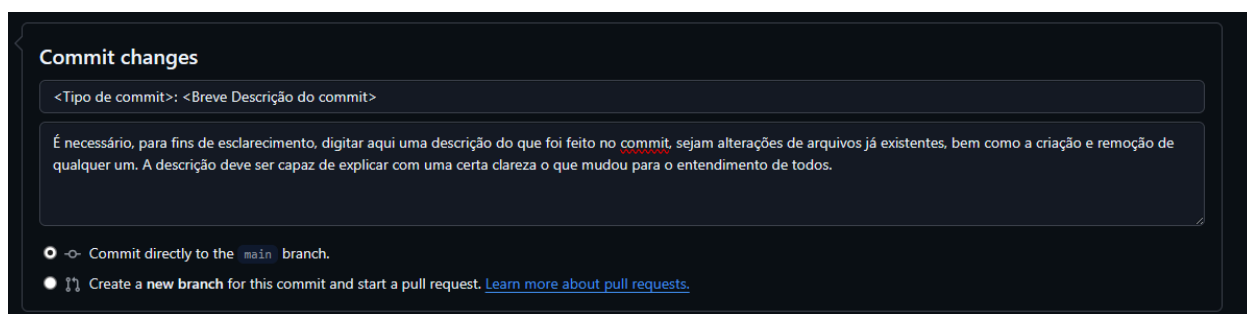
Sumário

1. Formato Geral do Commit	3
2. Commit do tipo feat	3
3. Commit do tipo fix	4
4. Commit do tipo docs	4
5. Commit do tipo test	5
6. Commit do tipo build	5
7. Commit do tipo remove	5
8. Commit do tipo refactor	6
9. Commit do tipo style	7
10. Referências e leituras recomendadas	8

1. Formato Geral do Commit:

O commit deve sempre seguir o seguinte padrão.

- Título do commit: **<Tipo de commit>: <Breve Descrição do commit>**
- Descrição do commit: **É necessário, para fins de esclarecimento, digitar aqui uma descrição do que foi feito no commit, sejam alterações de arquivos já existentes, bem como a criação e remoção de qualquer um. A descrição deve ser capaz de explicar com uma certa clareza o que mudou para o entendimento de todos.**

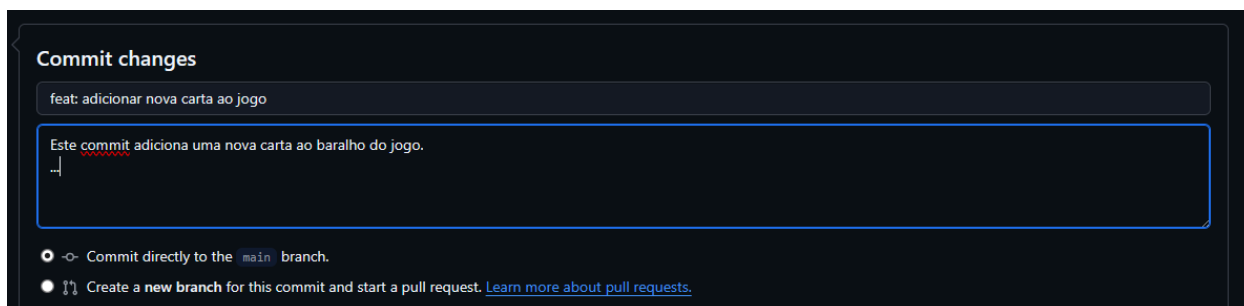


The screenshot shows the 'Commit changes' section in a dark-themed GitHub interface. At the top, there is a text input field with the placeholder text '<Tipo de commit>: <Breve Descrição do commit>'. Below this is a larger text area containing the instruction: 'É necessário, para fins de esclarecimento, digitar aqui uma descrição do que foi feito no commit, sejam alterações de arquivos já existentes, bem como a criação e remoção de qualquer um. A descrição deve ser capaz de explicar com uma certa clareza o que mudou para o entendimento de todos.' At the bottom, there are two radio buttons: the first is selected and labeled 'Commit directly to the main branch.', and the second is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.'

Figura 1. Estrutura padrão de commits no Github

2. Commit do tipo feat:

Commits do tipo **feat** indicam que o trecho de código commitado está incluindo um novo recurso. Utilizado sempre que há mudanças significativas no código com adição de funcionalidades novas.



The screenshot shows the 'Commit changes' section in a dark-themed GitHub interface. The text input field at the top contains the text 'feat: adicionar nova carta ao jogo'. Below it, the larger text area contains the description 'Este commit adiciona uma nova carta ao baralho do jogo.' followed by a cursor. At the bottom, the same two radio buttons are visible: 'Commit directly to the main branch.' (selected) and 'Create a new branch for this commit and start a pull request. Learn more about pull requests.'

Figura 2. Exemplo de commit do tipo **feat** feito corretamente

3. Commit do tipo fix:

Commits do tipo **fix** indicam que o trecho de código commitado está solucionando um problema. Utilizado sempre que há soluções para problemas de código anteriores.

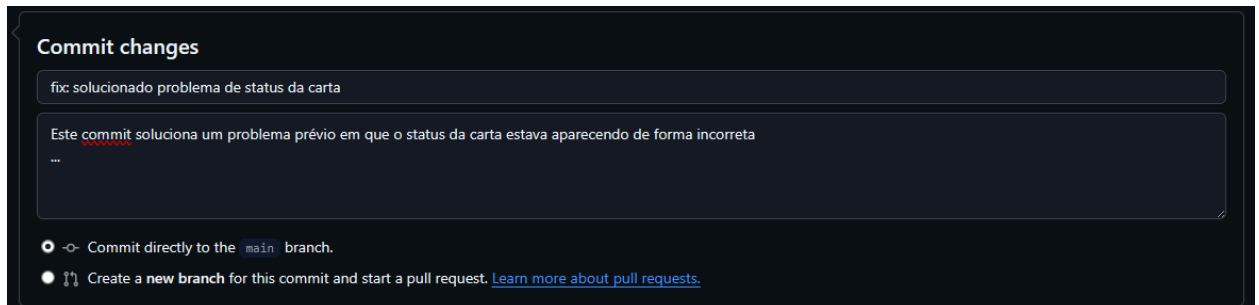


Figura 3. Exemplo de commit do tipo **fix** feito corretamente

4. Commit do tipo docs:

Commits do tipo **docs** indicam que houveram quaisquer mudanças em uma documentação, podendo ser mudanças em documentos já existentes, bem como a criação deles. Devido ao seu uso mais geral para documentos, é visto ainda mais a necessidade de especificar com clareza na descrição do commit o propósito dele, e possíveis alterações feitas.

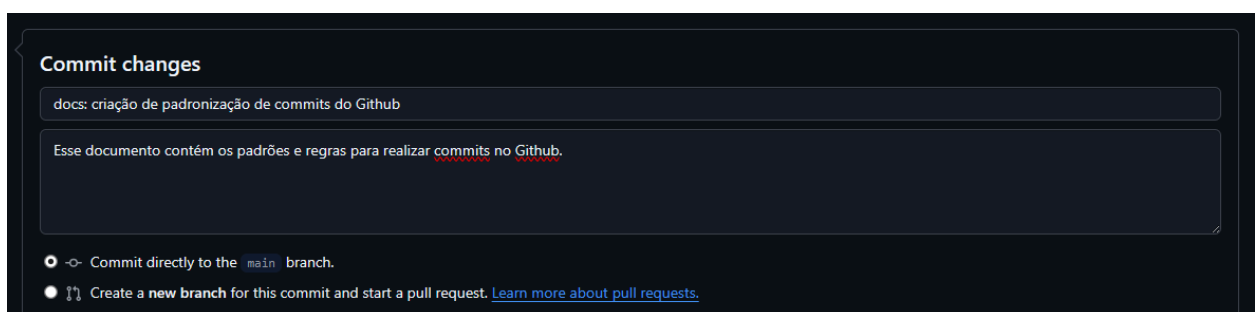
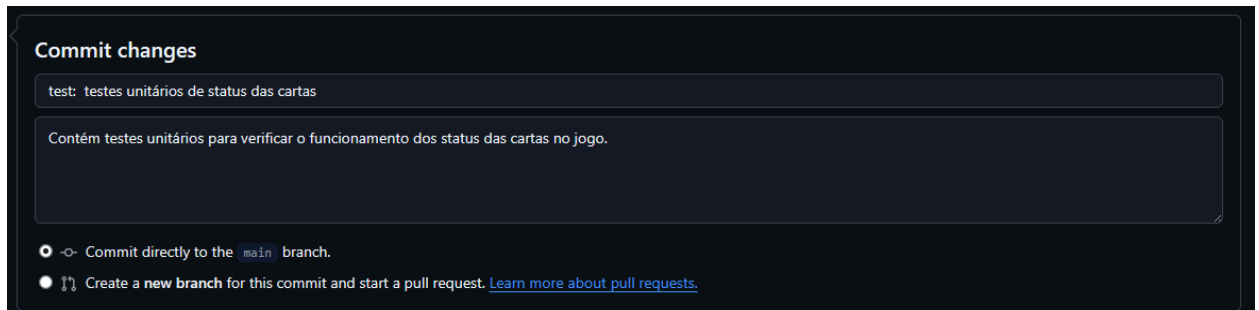


Figura 4. Exemplo de commit do tipo **docs** feito corretamente

5. Commit do tipo test:

Commits do tipo **test** indicam que houveram quaisquer mudanças em relação a testes unitários, podendo ser mudanças em testes unitários já existentes, bem como a criação deles.

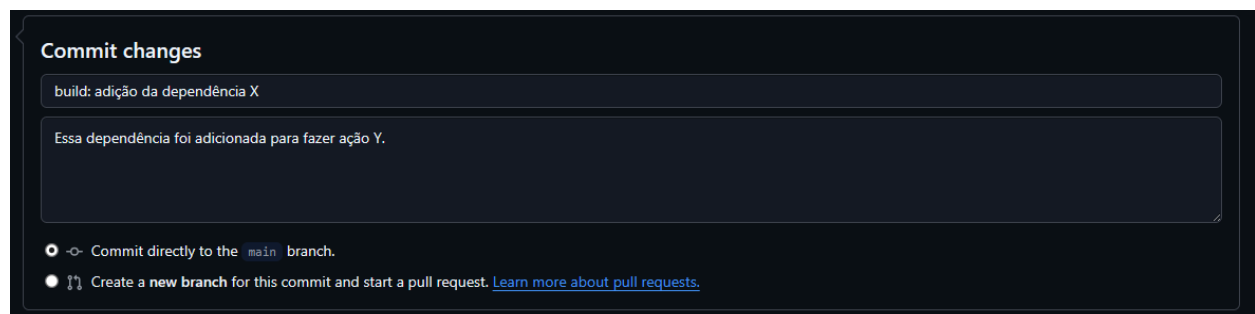


The screenshot shows a GitHub 'Commit changes' dialog. The commit type is set to 'test'. The commit message is 'test: testes unitários de status das cartas'. The description is 'Contém testes unitários para verificar o funcionamento dos status das cartas no jogo.' At the bottom, there are two radio buttons: the first is selected and labeled 'Commit directly to the main branch.', and the second is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.'

Figura 5. Exemplo de commit do tipo **test** feito corretamente

6. Commit do tipo build:

Commits do tipo **build** indicam que houveram quaisquer mudanças em relação a arquivos relacionados com o tipo build e suas dependências, podendo ser mudanças em arquivos já existentes, bem como a criação deles.



The screenshot shows a GitHub 'Commit changes' dialog. The commit type is set to 'build'. The commit message is 'build: adição da dependência X'. The description is 'Essa dependência foi adicionada para fazer ação Y.' At the bottom, there are two radio buttons: the first is selected and labeled 'Commit directly to the main branch.', and the second is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.'

Figura 6. Exemplo de commit do tipo **build** feito corretamente

7. Commit do tipo remove:

Commits do tipo **remove** indicam que houveram alguma remoção de arquivos existentes no diretório. Devido ao seu uso mais geral, é visto ainda mais a necessidade

de especificar com clareza na descrição do commit o propósito dele, e possíveis alterações feitas.

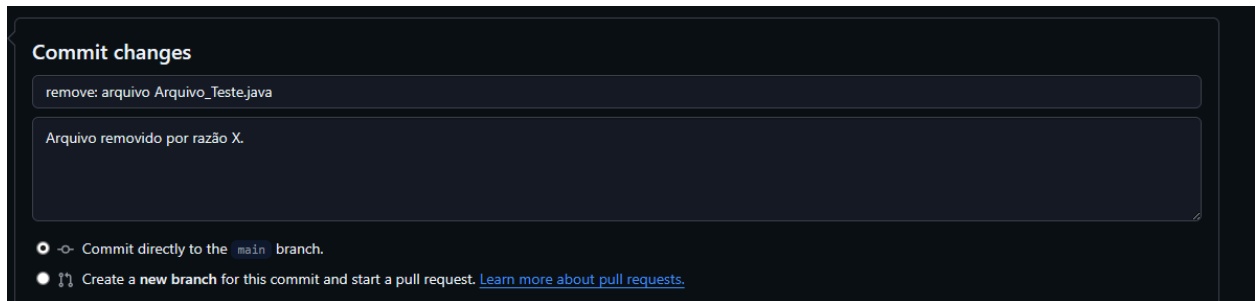


Figura 7. Exemplo de commit do tipo **remove** feito corretamente

8. Commit do tipo refactor:

Commits do tipo **refactor** indicam mudanças no código que melhoram sua estrutura, legibilidade ou desempenho, sem alterar o comportamento externo do sistema. Devem ser utilizados quando não há nenhum recurso novo, somente a alteração de recursos já existentes no código para alguma possível melhora de interpretação e usabilidade.

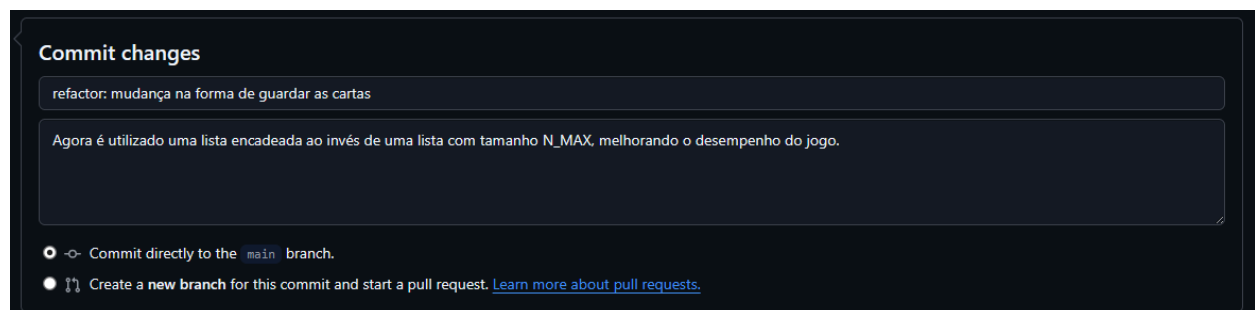


Figura 8. Exemplo de commit do tipo **refactor** feito corretamente

9. Commit do tipo style:

Commits do tipo **style** indicam alterações que afetam apenas o estilo do código e não alteram sua lógica ou comportamento. Devem ser utilizados sempre que há uma

mudança de nome de variáveis (sem, no entanto, alterar suas funcionalidades), além de ajustes de formatação em geral.

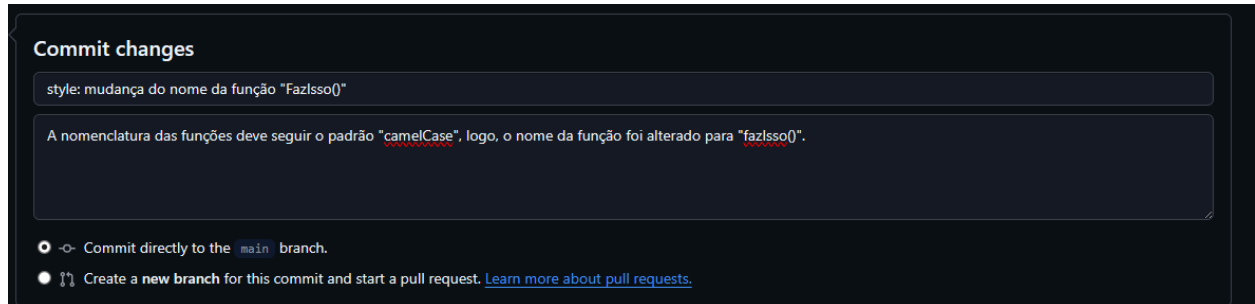


Figura 9. Exemplo de commit do tipo **style** feito corretamente

10. Referências e leituras recomendadas:

- <https://github.com/iuricode/padroes-de-commits>