|  |  |
| --- | --- |
| **RS11 GitHub – Grupo 3** | Pág. 1 de 3  Versão 1  Data: 22/04/2024 |

|  |
| --- |
| O que é? |

O GitHub é uma plataforma de hospedagem de código e colaboração para programadores. Permite que os programadores trabalhem em conjunto em projetos, acompanhem revisões de código, façam a gestão de versões, hospedem e compartilhem os seus códigos.

|  |
| --- |
| O que permite fazer? |

Algumas das principais características do GitHub incluem:

* Repositórios: Os repositórios são onde os projetos são armazenados.
* Controlo de Versão: O GitHub utiliza o sistema de controlo de versão Git, que permite que os programadores controlem e registem as alterações feitas no código ao longo do tempo.
* Colaboração: Os programadores podem colaborar em projetos trabalhando em *branches* separados e integrando as suas alterações quando estiverem prontas. Também podem rever, resolver conflitos de código, fazer *merge* e comentar o código uns dos outros.
* Segurança: O GitHub oferece recursos de segurança para proteger os repositórios e análises de dependências.
* Integrações: A plataforma tem integrações com várias ferramentas e serviços populares de desenvolvimento de software, como IDEs (por exemplo *Visual Studio*), sistemas de integração contínua, ferramentas de gestão de projetos e muito mais.

GitHub é uma plataforma essencial para programadores que desejam colaborar em projetos de software, partilhar o seu trabalho e contribuir para a comunidade de código. Permite que vários utilizadores trabalhem de forma independente, num mesmo projeto e “ramificar” o código por cada um dos elementos da equipa e mais tarde reuni-lo, numa versão final e completa.

|  |  |
| --- | --- |
| **GitHub** | Pág. 2 de 3  Versão 1  Data: 22/04/2024 |

|  |
| --- |
| Como funciona? |

* Adicionar um repositório implica ir aos repositórios, e no canto superior direito clique em Novo repositório. Escreva o nome, escolha se o seu repositório será Público ou Privado, adicione um ficheiro README (recomendado), adicione *git ignore* apropriado, e crie o repositório. O documento *git ignore* nesta fase é importante porque especifica quais os arquivos que o Git deve ignorar, o que é fundamental para garantir que apenas os arquivos relevantes para o código fonte do projeto sejam versionados. O ficheiro README é importante num projeto GitHub porque fornece uma introdução ao projeto, instruções de utilização, exemplos, diretrizes para contribuição, informação e recursos adicionais. Também é possível criar o repositório a partir do *Visual Studio*.
* Para ficar com uma cópia do repositório do GitHub localmente é importante fazer o comando *git clone* na diretoria que queremos o projeto a partir da linha de comandos. Depois adicionamos o projeto dentro da diretoria e fazemos o nosso primeiro *commit* para que o nosso projeto fique no Github. Outra forma, será introduzir os files no repositório do Github, fazer o primeiro *commit* e ir buscar graficamente no Visual Studio.
* Por *default*, o seu repositório tem um *branch* chamado *main* e pode criar *branches* adicionais (novo *branch*). O *branching* é útil quando deseja adicionar novas funcionalidades a um projeto sem alterar a fonte principal de código. O trabalho feito em *branches* diferentes não aparecerá no *branch* principal até ser *merged*, o que facilita o trabalho em equipa em *features* diferentes.
* No GitHub, as alterações guardadas são chamadas de *commits*. Cada *commit* tem uma mensagem de *commit* associada, que é uma descrição que explica por que foi feita uma determinada alteração (*add*, *commit* e *push* no *Visual Studio* com o *Git changes* para ser integrado no GitHub). As mensagens de *commit* capturam o histórico das suas alterações para que outros contribuidores possam entender o que foi feito e porquê. Para que os outros

|  |  |
| --- | --- |
| **GitHub** | Pág. 3 de 3  Versão 1  Data: 22/04/2024 |

contribuintes no mesmo código tenham as alterações que estão no GitHub, é essencial fazer o *pull* no *Visual Studio* com o *Git changes* para ficar com as alterações localmente e continuar

o projeto. Importante também é estar atento ao *branch* escolhido para trabalhar para não alterar código indevido (opção no *Visual Studio*).

* As *pull requests* são o cerne da colaboração no GitHub. Quando inicia uma *pull request* (clicar em *'New pull request'*) está a propor alterações ao *branch* para onde se direciona o *pull request* e a solicitar que alguém faça a revisão e/ou comente ou proponha alterações com propostas de melhoria. Esta característica de revisão é útil na colaboração em projetos e em equipas. Depois da revisão concluída e aceite, é necessário a aprovação do *merge*, confirmada no botão ‘*Merge pull request*’. Depois disso, as alterações no teu *branch* serão incorporadas no *branch* principal (*main*). Após tudo finalizado, poderá ser relevante apagar o *branch* da *feature* para manter um repositório limpo e organizado.
* Além disto, profissionais mais experientes tendem a usar a linha de comandos para realizar estas operações. Alguns dos comandos importantes mais usados são: *git status, git clone, git add, git –version, git reset, git checkout, git push, git pull, git commit, git merge*, etc.

|  |  |
| --- | --- |
| Elaborado por: | Alexandra Mendes, Paulo Melo, Sílvia Almeida, Thiago Rocha |