

Controle de Versão

Controle das Tarefas a serem realizadas para a Disciplina

NOME DO ESTUDANTE 1

NOME DO ESTUDANTE 2

DISCIPLINA

Tecnologias para Desenvolvimento Web

1 Contextualização

O controle de versão visa possibilitar o trabalho colaborativo no qual os artefatos construídos sejam conhecidos e acompanhados pelos diferentes desenvolvedores de um projeto. Essa é uma prática muito comum, pois assim diferentes pessoas podem trabalhar em diferentes frentes do mesmo projeto e depois consolidar essas partes em uma versão entregável de um produto, de forma fácil e acessível.

Para promover esse acompanhamento, nos apoiaremos em uma ferramenta em nuvem muito conhecida, o GitHub, que é uma plataforma de hospedagem de código-fonte, com controle de versão. Nele manteremos nosso repositório de artefatos, que poderão ter diferentes versões, com respectivo histórico de alterações, caso seja necessário recuperar trabalhos anteriores.

2 Enunciado do Trabalho

Para este trabalho, a equipe organizará um **repositório na ferramenta GitHub** (<https://github.com/>), onde irá organizar todo o código fonte das tarefas solicitadas para os Resultados de Aprendizagem 1 e 2 – ou RA1 e RA2.

IMPORTANTE: Cada membro da equipe deve contribuir com atualizações **regulares**, ao longo das **semanas previstas na disciplina** para desenvolvimento dos **tutoriais e trabalhos** de implementação de aplicações Java Web, uma vez que **todos os componentes da Equipe devem executar todas as tarefas**.

2.1 Passos para a Execução

Com base no material de apoio **TDW_Tema3_ControleVersão.pdf**, siga as orientações para o trabalho a ser avaliado.

2.1.1 Crie sua conta no GitHub

Acesse <http://github.com> e solicite a criação da conta na opção “Sign up”.

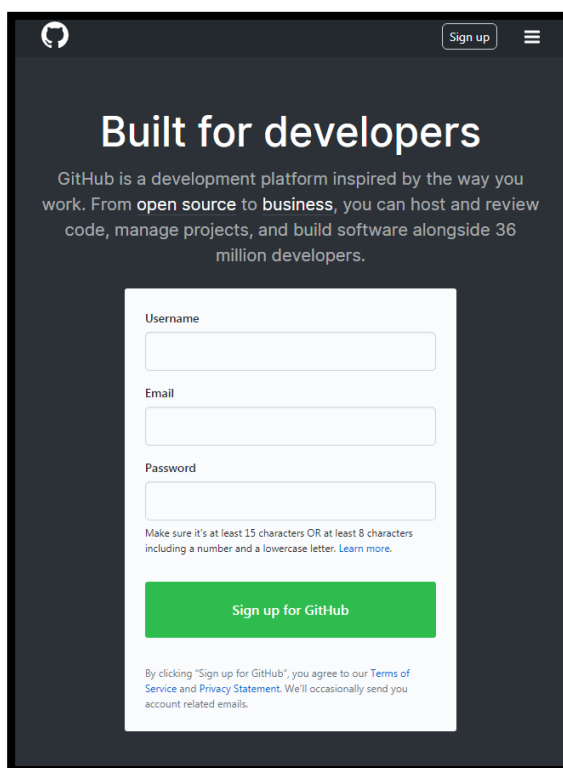
The image shows the GitHub sign-up page. At the top, there's a navigation bar with the GitHub logo on the left, a 'Sign up' button in the center, and a hamburger menu icon on the right. Below the navigation bar, the main heading is 'Built for developers'. Underneath this, a paragraph describes GitHub as a development platform inspired by the way you work, mentioning 'open source' and 'business'. Below the text, there's a sign-up form with three input fields: 'Username', 'Email', and 'Password'. The 'Password' field has a note below it: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.' Below the form is a large green button labeled 'Sign up for GitHub'. At the bottom, there's a small disclaimer: 'By clicking "Sign up for GitHub", you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account related emails.'

Figura 1: Criação de conta no repositório GitHub.

2.1.2 Crie um repo (repositório) para a Equipe

Para colocar seu projeto no GitHub, você precisará criar um repositório para ele. Para isso, acesse no canto superior direito de qualquer página, clique em **+** e, em seguida, clique em **New repositor**, como na Figura 2.

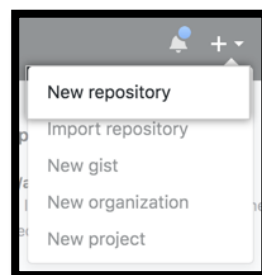


Figura 2. Criando um novo repositório no GitHub

Digite um nome curto e fácil de lembrar para o seu repositório. Por exemplo, "TDW-Equipe-Pedro&Lucas ", como na Figura 3.

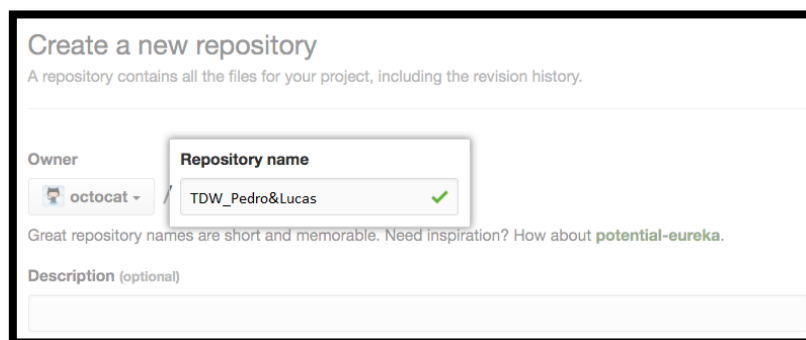


Figura 3. Nomeando o Repositório.

Opcionalmente, adicione uma descrição do seu repositório. Por exemplo, "Meu primeiro repositório no GitHub". Após, escolha tornar o repositório **público** ou **privado**. Repositórios **públicos** são visíveis por todos, enquanto os **privados** são acessíveis apenas para você e para as pessoas com quem você os compartilha. Por fim, clique em **Create repository**.

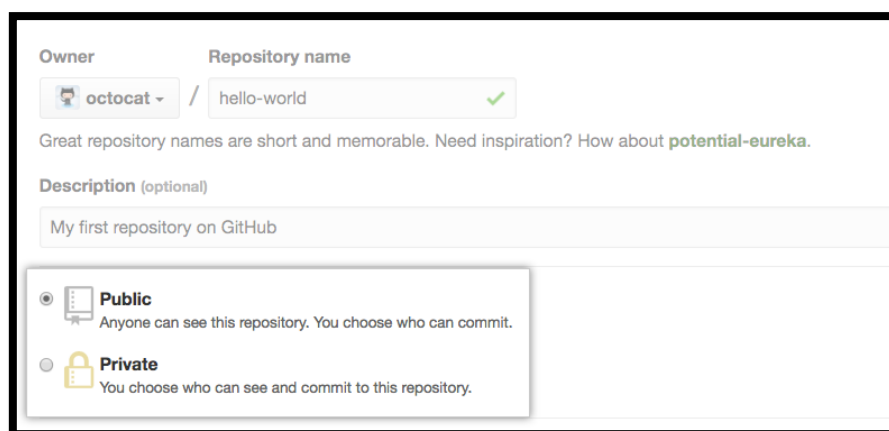


Figura 4: Escolha entre um repositório público ou privado.

2.1.3 Compartilhando o Repositório

Você pode convidar usuários para se tornarem colaboradores em seu repositório pessoal. Peça o nome de usuário da pessoa que você está convidando como colaborador. Se eles ainda não tiverem um nome de usuário, eles deverão se inscrever no GitHub.

No GitHub, navegue até a página principal do repositório. Sob o nome do seu repositório, clique em **Settings** (configurações).

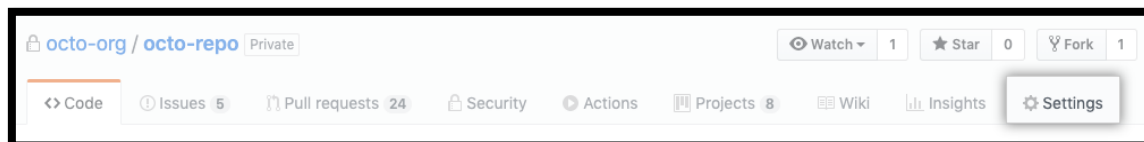


Figura 5. Configurações do repositório.

No painel à esquerda, clique em **Collaborators**.

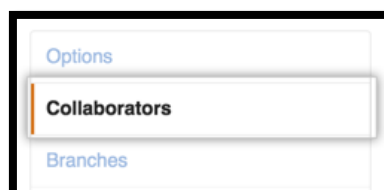


Figura 6. Colaboradores do repositório.

Digite o nome do colaborador e clique em **Add collaborator**. O usuário receberá um email convidando-os para o repositório. Assim que ele aceitar o convite, ele terá acesso de colaborador ao seu repositório.

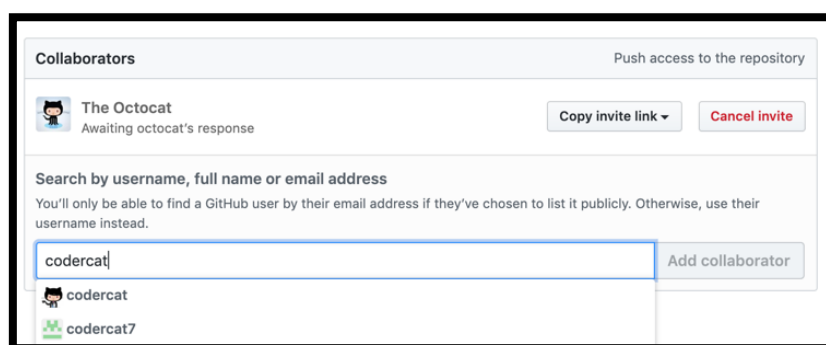


Figura 7. Escolhendo os colaboradores do repositório.

2.1.4 Escolha a ferramenta Git para controle do diretório / pasta local

GitHub Desktop – siga os passos de utilização apresentados na referência:

GitHub para Windows em <https://herbertf.wordpress.com/2015/06/10/utilizando-o-github-no-windows/>

Git BASH – siga os passos de utilização apresentados na referência:

Git - Guia prático: http://rogerdudler.github.io/git-guide/index.pt_BR.html

2.1.5 Preparação

Dependendo da ferramenta Git para controle da pasta local, Git BASH ou GitHub Desktop, realize os passos indicados nos tutoriais, de forma a executar a sequência de ações listadas abaixo:

- 1) **Criação da pasta local de trabalho** – que será sincronizada com o repositório da equipe no GitHub. Como cada integrante da equipe precisa desenvolver seu próprio exercícios, na pasta de trabalho sincronizada com o repositório do GitHub, crie uma subpasta para cada integrante da equipe. Com isso, cada um consegue desenvolver seus próprios exercícios, com total visibilidade sobre o trabalho dos demais componentes da equipe.
- 2) **Clone do repositório** – clonar o repositório do Github na pasta local de trabalho.
- 3) **Pasta de trabalho monitorada** – quando adicionamos subpastas ou arquivos, ou alteramos arquivos dentro da pasta local de trabalho, esses arquivos e subpastas são monitorados pelo controle de versão.
- 4) **Enviando as alterações** – faça **commit** dos arquivos novos, removidos e alterados, indicando um título e uma mensagem que descreva o motivo das modificações que estão sendo salvas; essa ação é importante para os demais membros da equipe acompanhar o trabalho dos demais.
- 5) **Workflow** – o fluxo de trabalho utilizando o versionamento GIT é
 - a. Faz a cópia do repositório em uma pasta local de trabalho (uma única vez);
 - b. Atualiza o diretório versionando (pull);
 - c. Trabalha no diretório versionado;
 - d. Realiza o commit;
 - e. Sincroniza com o repositório (push).

2.1.6 Orientações Gerais

O professor tutor irá avaliar este trabalho de acordo com o solicitado no presente documento, no item “**Critérios de Correção – Controle de Versão**”. Assim, combine com seus colegas para constantemente acrescentarem e atualizarem os códigos fonte dos trabalhos individuais, verificando se todos contribuíram de forma regular. Veja na Figura 8 como acompanhar as contribuições.

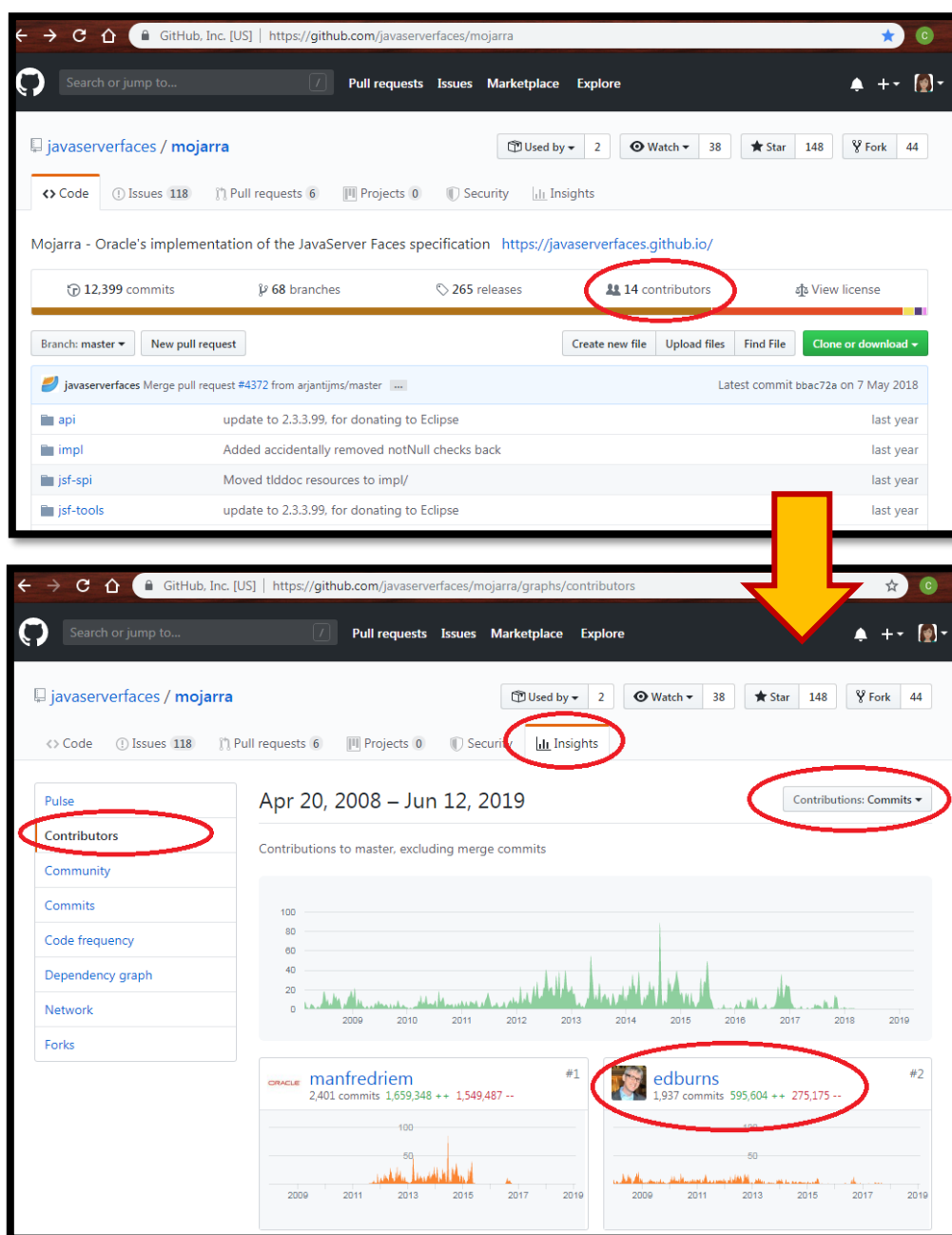


Figura 8: Exemplo de controle no GitHub, com as contribuições individuais da equipe.

3 Critérios de Correção – Controle de Versão

O presente trabalho é avaliativo e será corrigido conforme detalhado a seguir.

Equipe	Nome Estudante 1:	
	Nome Estudante 2:	
Critério	Instâncias	Pontuação
O Repositório do GitHub é compartilhado pela Equipe	Repositório do GitHub está compartilhado por pelo menos 2 (dois) estudantes da Equipe	1,0
	Repositório do GitHub está atribuído a apenas 1 (um) estudante, ou seja, não permite o trabalho colaborativo.	0,0
O Repositório do GitHub é compartilhado com o Tutor	O Repositório do GitHub da equipe está compartilhado com o Tutor	1,0
	O Repositório do GitHub da equipe não está compartilhado com o Tutor	0,0
Contribuição na confecção dos artefatos, por todos os membros da Equipe	Todos os componentes da Equipe contribuíram com a confecção dos artefatos (códigos-fonte e demais documentos) do Projeto, atualizando o GitHub, de forma regular (ver Figura 8 deste documento)	4,0
	Todos os componentes da Equipe contribuíram com a confecção dos artefatos (códigos-fonte e demais documentos) do Projeto, atualizando o GitHub, de forma pontual	3,0
	Alguns componentes da Equipe não contribuíram com a confecção dos artefatos (códigos-fonte e demais documentos) do Projeto, atualizando o GitHub	1,0
	GitHub do Projeto da Equipe não apresenta atualizações / inserções.	0,0
Organização da Estrutura do Repositório e Ferramenta de Controle de Versões	Todos os artefatos do Projeto encontram-se organizados em pastas e subpastas por tipo e/ou assunto, em um repositório GitHub com controle de versão <ul style="list-style-type: none"> • Códigos-fonte • Recursos do sistema (imagens, estilos, ...) • Documentos do Projeto 	4,0
	Pelo menos metade dos artefatos do Projeto encontram-se organizados em pastas e subpastas por tipo e/ou assunto, em um repositório GitHub com controle de versão	3,0
	Menos da metade dos artefatos do Projeto encontram-se organizados em pastas e subpastas por tipo e/ou assunto, em um repositório GitHub com controle de versão	1,0
	Os artefatos do Projeto não se encontram organizados em pastas e subpastas por tipo e/ou assunto, em um repositório GitHub com controle de versão	0,0
Pontuação Máxima		10,0

REFERÊNCIAS

Site oficial Git Bash: <https://git-scm.com/>

Git Essencial - Obtendo um Repositório Git: <https://git-scm.com/book/pt-br/v1/Git-Essencial-Obtendo-um-Reposit%C3%B3rio-Git>

Ramificação (Branching) no Git - Básico de Branch e Merge: <https://git-scm.com/book/pt-br/v1/Ramifica%C3%A7%C3%A3o-Branching-no-Git-B%C3%A1sico-de-Branch-e-Merge>

Livro do GIT, em português: <https://git-scm.com/book/pt-br/v1/Primeiros-passos-Sobre-Controle-de-Vers%C3%A3o>

Dicionário Git e GitHub: <https://gist.github.com/victorsenam/8580499>

Git - Guia prático: http://rogerdudler.github.io/git-guide/index.pt_BR.html

GitHub para Windows: <https://herbertf.wordpress.com/2015/06/10/utilizando-o-github-no-windows/>