GitHub

- Plataforma de hospedagem de código-fonte com controle de versão usando o Git que será implementada no trabalho;

→ Cadastro (ou acesso caso já possuam conta)
https://github.com/

Git

→ Instalação (Windows)
https://git-for-windows.github.io/

- Aceitar tudo e adicionar o Git Bash na área de trabalho para ficar mais fácil de manipular. Durante a instalação há a opção de adicionar o git bash ao path deixando ele disponível no terminal do Windows. Essa opção vai do agrado de cada um.

→ Instalação (Linux <3)
$ sudo apt-get update
$ sudo apt-get install git

→ Configurando Git
$ git config --global user.name "Seu_Nome"
$ git config --global user.email "seu_email@email.com"

- Para quem estiver curioso, a pasta .git (que estará oculta) possui o arquivo config que terá os dados do git e github tratando da clonagem que será explicada mais à frente. (ls -la no bash ou terminal Linux - enquanto estiver dentro da pasta - para ver a existência da pasta e ctrl h (em linux) ou acessar o Painel de Controle, clicar em Aparência e Personalização, clique em Opções de Pasta, clicar em Visualizar, clicar em Exibir Arquivos, Pastas e Drives Ocultos, em Configurações Avançadas, e clicar em OK (em Windows) para ter a visualização da pasta oculta);

Integrando o GitHub no Git (Windows e Linux)

- O projeto terá um repositório com uma pasta para cada grupo que conterá seus códigos e que serão monitorados pelos gerentes;
- O repositório estará na conta da Aline: https://github.com/NinaCris16/
- Cada grupo fará uma "garfada" no código principal. Para isso, acessar o repositório do trabalho e clicar em "Fork". Basicamente essa ação pegará o repositório completo do trabalho e copiará para sua conta git, lá você poderá editar esses arquivos e depois devolver ao projeto* com as edições, que devem estar corretas para fazerem parte do repositório.
- * Para melhor controle **apenas os líderes de cada grupo** poderão mandar as modificações dos arquivos;

→ Obtendo os códigos para modificação

- Ir no repositório do trabalho do seu fork, ou seja, da cópia criada na sua conta e clicar em "clone or download";
- Aparecerá um link que deverá ser copiado e colado no bash ou terminal Linux da seguinte forma:

  $ **git clone link_copiado**

- Agora, vocês possuem o repositório em suas máquinas, basta procurá-los para editá-los. Eu sugiro movê-las para a área de trabalho;

→ LÍDERES ENTRAM EM AÇÃO

- É hora de juntar o que foi feito durante a semana! Junte o que seu grupo fez e acrescente a pasta que clonaram. É hora de mexer no bash ou terminal Linux:

  $ **cd <local_da_pasta>**

// exemplo: cd desktop

  $ **cd <nome_da_pasta>**

// agora você está dentro da pasta. Para saber o que existe nela: ls -la

  $ **git add <nome do arquivo>**

// adiciona os arquivos em seu repositório local

  $ **git commit -m "modificações"**

// faz commit de seus arquivos para o repositório local

  $ **git push origin master**

// envia o commit mais atual para o GitHub

Após esses comandos, seguir para o GitHub e conferir se no Fork do trabalho na conta de vocês possuem as modificações mandadas.
Ao estarem na pasta onde se encontram as modificações, procurar o “pull request” que se encontra abaixo dos botões “Upload files, find file” e clicá-lo. Então, basta clicar no create pull request que suas modificações serão enviadas para o repositório principal do trabalho que se encontra na conta da Aline.

→ Atualização do Fork com o projeto principal

É importante sempre conferir as atualizações de código no repositório principal, ou seja, sempre atualizar seu Fork. Para isso os seguintes passos deverão ser feitos:

$ **cd <local_da_pasta>**
// exemplo: cd desktop
$ **cd <nome_da_pasta>**
// agora você está dentro da pasta. Para saber o que existe nela: ls -la
$ **git remote add upstream <repositorio_principal>**
// adicionando um remote com o nome de upstream
$ **git fetch upstream**
// obtendo arquivos do repositório principal (atualização)
$ **git rebase upstream/master**
// agora tudo que existe no repositório principal está na sua máquina

Tal processo atualiza os arquivos da sua máquina, agora você precisa enviá-los para o seu GitHub. Para isso, siga os passos que se encontram em “LÍDERES EM AÇÃO”.

→ Resolução de Conflitos quanto ao “git push”

O Git possui um sistema que não permite que você mande sua edição caso seu arquivo esteja “desatualizado”. Imagine que o Nícolas tenha iniciado uma modificação no master e tenha dado um git push, ou seja, agora o master possui um novo atributo. Ao mesmo tempo a Aline também estava mexendo em uma modificação e decide enviá-las dando um git push. A Aline receberá uma mensagem de erro pois antes de enviar suas modificações ela precisa importar as modificações do Nícolas pois se esse sistema não existisse toda hora alguém iria sobrescrever o código do de outra pessoa. Para isso, Aline precisa dar um git pull e depois de importada as modificações ela poderá tranquilamente dar seu git push.

$ **git pull --rebase origin master**
// atualiza seu repositório em relação ao github

Tutorial Git

https://www.codeschool.com/courses/try-git

Sites de Auxílio - Como usar Git

http://rogerdudler.github.io/git-guide/index.pt_BR.html
https://blog.da2k.com.br/2015/02/04/git-e-github-do-clone-ao-pull-request/

// Este tutorial ensina a mexer com git no Bitbucket um repositório como o GitHub. Os conceitos que estão aqui, para quem possui curiosidade de aprender mais sobre Git, são muito interessantes.

https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud

Sites de Auxílio - Como resolver conflitos

https://www.atlassian.com/git/tutorials/comparing-workflows