

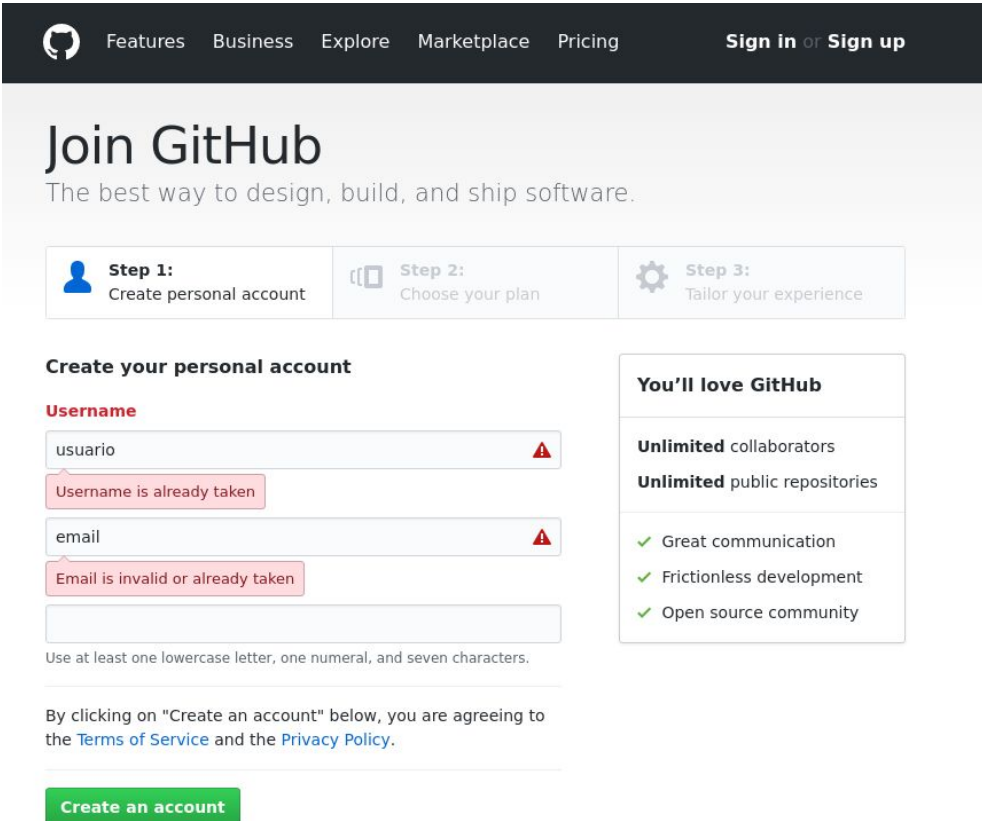


1. Objetivo

O objetivo deste tutorial é introduzir os conceitos básicos para trabalharmos com a plataforma GitHub. Esses conceitos básicos englobam: criação de conta, criação de repositório e exemplo de inserção de um arquivo no repositório.

2. Criação da Conta no GitHub

Ao acessar o website <https://github.com>, clique na opção “Sign up” e preencha as informações como ilustrado na figura abaixo:



The screenshot displays the GitHub sign-up interface. At the top, there are navigation links: Features, Business, Explore, Marketplace, Pricing, and Sign in or Sign up. The main heading is "Join GitHub" with the tagline "The best way to design, build, and ship software." Below this, a progress bar shows three steps: Step 1: Create personal account (active), Step 2: Choose your plan, and Step 3: Tailor your experience. The "Create your personal account" section includes a "Username" field with the value "usuario" and a red error message "Username is already taken", and an "email" field with the value "email" and a red error message "Email is invalid or already taken". A green "Create an account" button is located at the bottom of the form.

Figura 1: Página inicial para criação da conta GitHub.

Esse processo de inscrição requer 3 passos: no primeiro passo as informações básicas são preenchidas, no segundo passo (figura abaixo), é pedido que se escolha o plano, vamos usar o plano gratuito. No terceiro passo é feito um questionário, para customizar a experiência do usuário.

Welcome to GitHub

You've taken your first step into a larger world, @peixotogabriel.

The screenshot shows the GitHub account creation interface. At the top, there are three progress steps: 'Completed' (Set up a personal account), 'Step 2: Choose your plan' (highlighted), and 'Step 3: Tailor your experience'. Below this, the 'Choose your personal plan' section offers two options: 'Unlimited public repositories for free.' (selected) and 'Unlimited private repositories for \$7/month.'. A note states: 'Don't worry, you can cancel or upgrade at any time.' Below the plan selection, there are two checkboxes: 'Help me set up an organization next' (with a sub-note about organizations) and 'Send me updates on GitHub news, offers, and events' (with a sub-note about email preferences). A green 'Continue' button is at the bottom.

Figura 2: segundo passo na criação da conta do GitHub.

Não se esqueçam de confirmar os e-mails após a criação das contas!!!

3. Criação do Repositório

Após criada a conta, vamos criar o repositório que será usado na avaliação das listas de exercício. A convenção de nomenclatura se encontra abaixo:

- Nome do repositório: **pe_q32017_RA**
- Cada lista ficará em uma pasta, **ex: lista01**
- Cada questão da lista será respondida em um arquivo com o nome igual ao número da questão.c, **ex: 01.c**

Na ferramenta GitHub, acesse sua conta e clique na opção “+” ao lado do seu retrato, dentro dessa janela selecione “*new repository*”, como ilustrado pelas Figuras 3 e 4, logo a seguir. Na Figura 5 temos um exemplo de preenchimento da janela de criação de repositório, o repositório deve ser público e ter o nome de acordo com a convenção apresentada anteriormente.

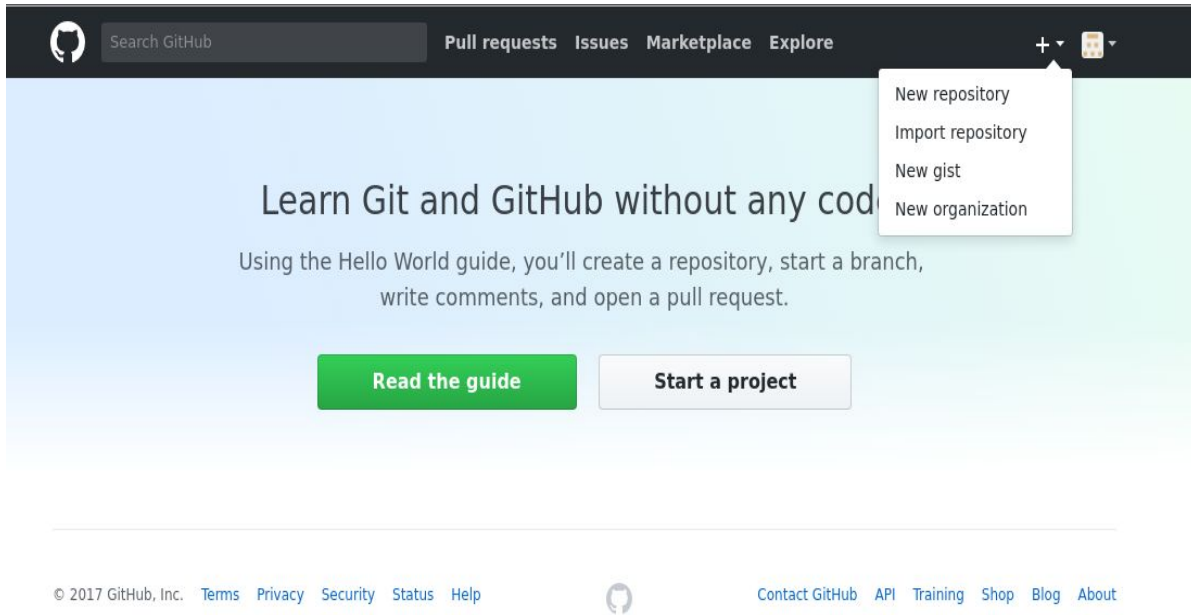


Figura 3: Criação de repositório

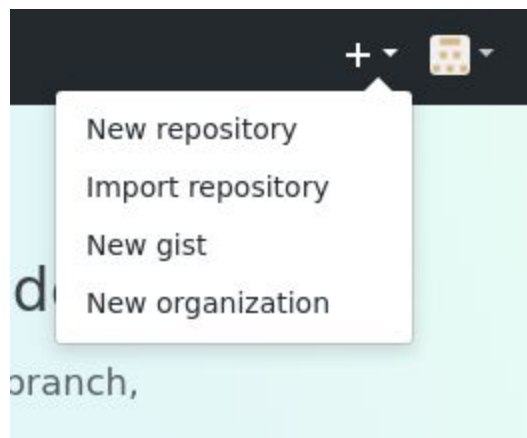


Figura 4: criação de repositório com a opção ***new repository***.

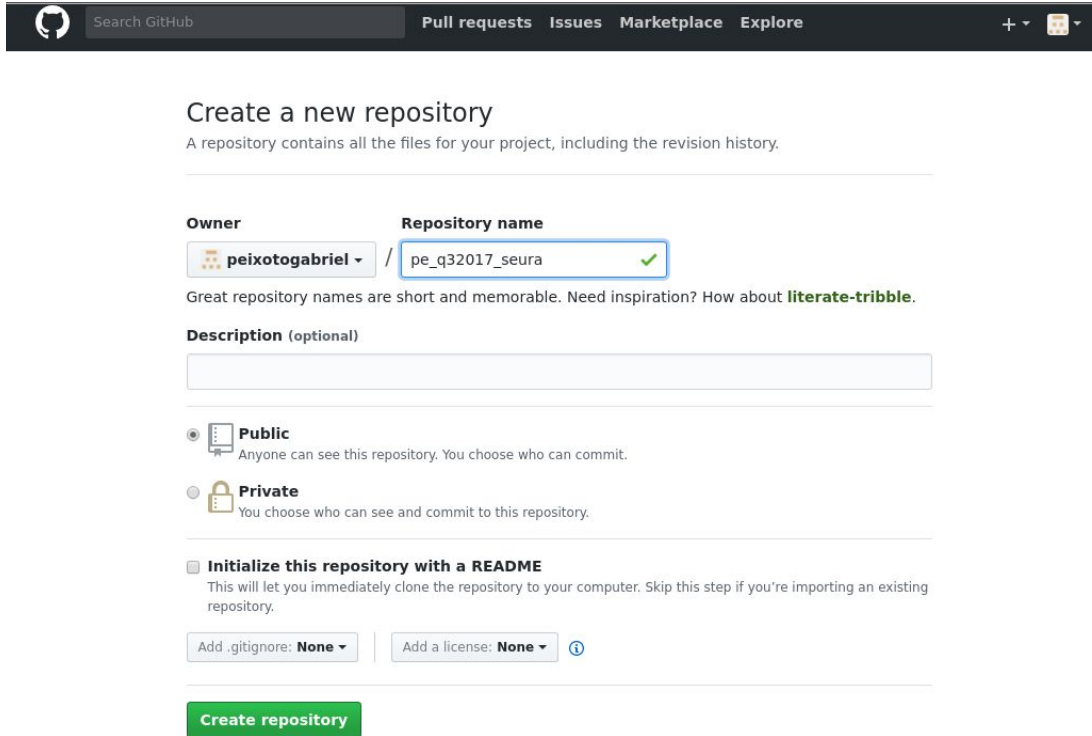


Figura 5: Exemplo de preenchimento da tela de criação de repositório

Após criarem o repositório, vocês serão apresentados a seguinte tela:

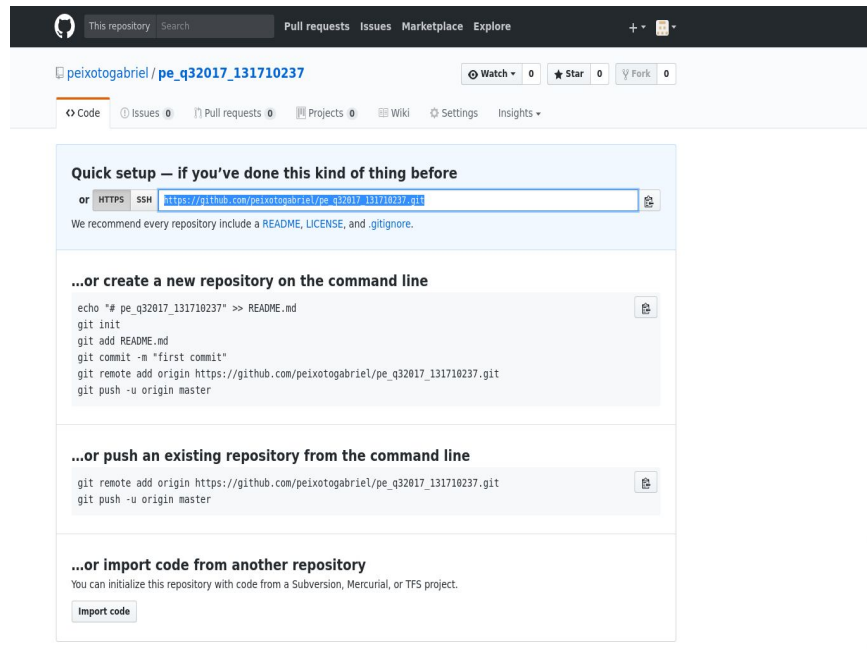
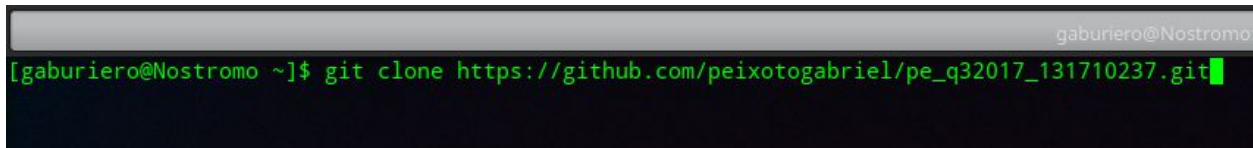


Figura 6: Tela apresentada ao usuário após a criação do repositório

Na Figura 6 temos a tela que é apresentada após a criação do repositório, agora podemos começar a resolução das listas. Copie o endereço de seu repositório que estão no retângulo azul, no formato: ***https://github.com/usuario/repositorio.git***

4. Exemplo de fluxo de desenvolvimento

Com o link copiado, abra um terminal. Com o terminal aberto entre com o comando da Figura 7 abaixo:



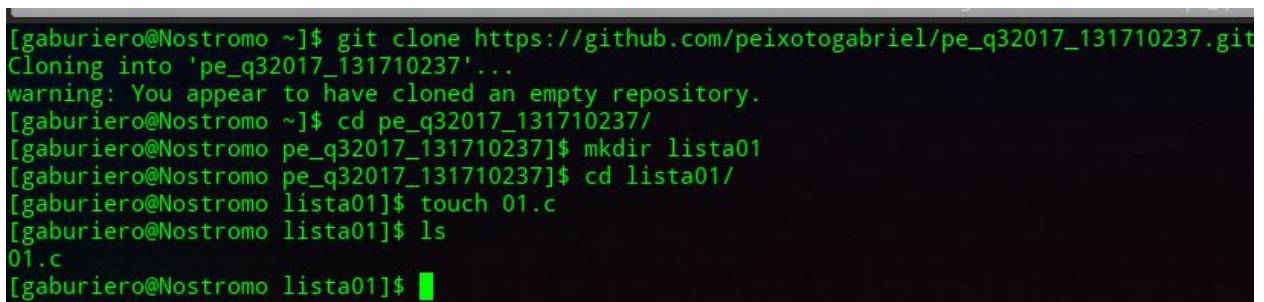
```
gaburiero@Nostromo
[gaburiero@Nostromo ~]$ git clone https://github.com/peixotogabriel/pe_q32017_131710237.git
```

Figura 7: git clone

O comando é no formato: ***git clone https://github.com/usuario/repositorio.git***. Esse comando vai copiar o repositório do servidor (*remote*) para sua máquina no diretório local (*local*). Com isto feito é possível começar a resolver as questões.

Em seguida vamos revisar alguns comandos básicos de linux para: criação de diretórios, entrar e sair de diretórios e criar arquivos vazios. No trecho de código da Figura 8 (Abaixo), temos exemplos dos comandos básicos que iremos utilizar no terminal para solucionar as listas:

1. ***cd pe_q32017_RA*** : entra no diretório do repositório;
2. ***mkdir lista01*** : cria um diretório com o nome lista01;
3. ***cd lista01*** : entra no diretório da lista01;
4. ***touch 01.c*** : cria um arquivo vazio com o nome 01.c;
5. ***ls*** : Lista todos os arquivos e diretórios que se encontram no diretório corrente.



```
[gaburiero@Nostromo ~]$ git clone https://github.com/peixotogabriel/pe_q32017_131710237.git
Cloning into 'pe_q32017_131710237'...
warning: You appear to have cloned an empty repository.
[gaburiero@Nostromo ~]$ cd pe_q32017_131710237/
[gaburiero@Nostromo pe_q32017_131710237]$ mkdir lista01
[gaburiero@Nostromo pe_q32017_131710237]$ cd lista01/
[gaburiero@Nostromo lista01]$ touch 01.c
[gaburiero@Nostromo lista01]$ ls
01.c
[gaburiero@Nostromo lista01]$
```

Figura 8: Trecho de código para iniciar a resolução da lista 1

Após executados os comandos acima, temos um arquivo inicial e vazio para trabalharmos. Abra este arquivo no editor de texto de sua escolha, neste exemplo iremos usar o vim. O editor vim pode ser usado diretamente do terminal, entretanto não entraremos em detalhes do seu uso, pois está fora do escopo deste documento. Na Figura 9 temos uma solução para a questão da lista.

```
1 #include <stdio.h>
2
3 int main () {
4
5     int valor, resto;
6     printf("insira um numero: ");
7     scanf("%d", &valor);
8
9     resto=valor%2;
10
11     if (resto == 0) {
12         printf("0 numero inserido e par\n");
13     } else {
14         printf("o numero inserio e impar\n");
15     }
16
17     return 0;
18
19 }
```

Figura 9: uma solução para a questão 01 da lista01.

Após fazer o código da questão 01, vamos salvá-lo. Com o código salvo vamos compilá-lo e testá-lo. Não se deve colocar códigos incompletos ou defeituosos no repositório, é preferível esperar até que o problema seja resolvido e colocar o código fonte completo no repositório remoto (servidor). A Figura 10 exemplifica o fluxo de desenvolvimento git com os comandos básicos.

1. Git add 01.c : Adiciona um arquivo para ser inserido no sistema de controle de versão.
2. Git commit -m "adicionado exercicio 01" : Fala para o controle de versão guardar as mudanças adicionadas com a mensagem inserida.
3. Git push origin master : Envia as mudanças locais para o remoto, ou seja, do seu computador para o servidor.

```
[gaburiero@Nostromo lista01]$ ls
01 01.c
[gaburiero@Nostromo lista01]$ git add 01.c
[gaburiero@Nostromo lista01]$ git commit -m "adicionado exercicio 01"
[master (root-commit) 0456d93] adicionado exercicio 01
 1 file changed, 19 insertions(+)
 create mode 100644 lista01/01.c
[gaburiero@Nostromo lista01]$ git push origin master
Username for 'https://github.com': peixotogabriel
Password for 'https://peixotogabriel@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 416 bytes | 416.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/peixotogabriel/pe_q32017_131710237.git
 * [new branch]      master -> master
[gaburiero@Nostromo lista01]$ █
```

Figura 10: Exemplo do fluxo de desenvolvimento Git

Erros e dicas

Como provavelmente para muitos vai ser a primeira vez que estarão usando a ferramenta git, pode ser que ocorra um erro na hora de executar o comando git push:

```
> git init
✓ Running command - done!
> git add -A
✓ Running command - done!
> git commit -m "Initial commit" --no-gpg-sign
✗ Running command - failed!
[ERROR] An error occurred while running git commit -m "Initial commit"
--no-gpg-sign (exit code 128):

*** Please tell me who you are.

Run

git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got
'jorge@equipo-jorge.(none)')

C:\Users\jorge\Desktop\Important Folders\ionic>
```

Figura 11: Erro de identidade Git, retirado de <https://stackoverflow.com/questions/45664901/can-they-help-me-with-this-error-of-ionic-error-an-error-occurred-while-runnin>

Esse problema ocorre porque o git instalado em sua máquina não sabem quem é você, ele precisa saber um e-mail e nome de usuário para colocar como autor do commit. Para solucionar este problema use os comandos abaixo:

```
git config --global user.name "usuariogithub"  
git config --global user.email emailregistradonogithub@email.com
```

Para instalar o github no ubuntu é só executar o seguinte comando:

```
sudo apt-get install git
```

Cada commit cria um nó na árvore de commits, essa árvore “conta a história” do repositório. É possível visualizar essa árvore por aplicativos de interface gráfica (consulte os slides para mais informações). A ferramenta da Figura 12 é o gitk, uma ferramenta bem simples para visualizar os commits.

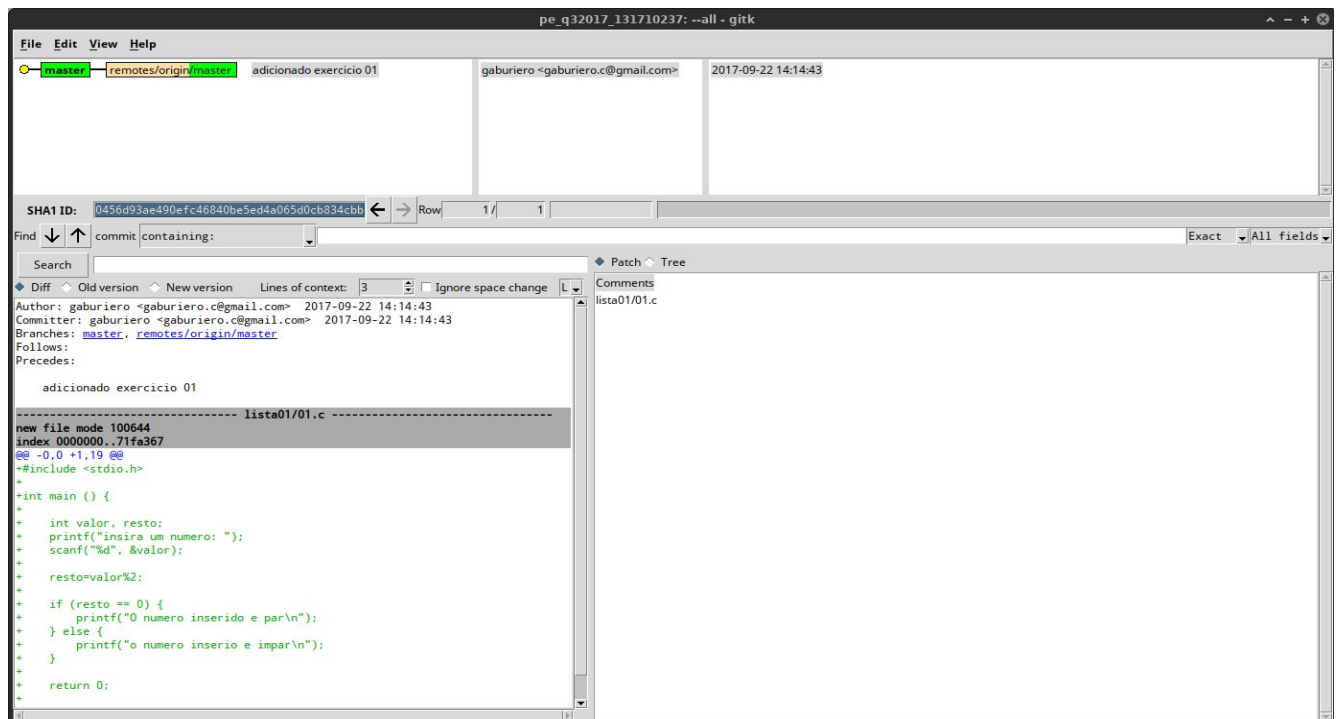


Figura 12: exemplo de ferramenta para visualizar as mudanças no repositório graficamente (gitk).

Contato e Dúvidas

Cada aluno deverá enviar um e-mail para: gabriel.carvalho@ufabc.edu.br

Com o título: pe_q32017 github

Neste e-mail deve conter as informações de: Nome, RA e se possível link para o repositório.