



Sistemas de Colaboração em Projetos (GitHub)

Objetivos:

- Sistemas de colaboração em projetos.
- Repositórios Git.

4.1 Introdução

Quando diversas entidades colaboram num projeto comum é necessário que existam meios de coordenar as atividades. Tal não se faz através de plataformas genéricas como o *Facebook* ou o *Google+*. Por exemplo, é necessário saber se o projeto está atrasado ou não, quais são as próximas tarefas e quem está responsável por elas, ou quem realizou uma determinada tarefa. Também é necessário que existam meios de comunicação rápida entre todos, tal como uma *mailing-list*. Também pode existir necessidade de ter uma base de informação partilhada.

Esta é a razão pela qual utilizamos a plataforma <http://elearning.ua.pt>, pois possui ferramentas que facilitam o desenrolar das aulas e a colaboração entre docentes e alunos.

Quando se fala de projetos com carácter de desenvolvimento de aplicações (programação), o processo de colaboração necessita de meios ainda mais evoluídos. Isto porque este tipo de projetos têm uma complexidade acrescida ao nível da escrita do código, verificação das funcionalidades, identificação das alterações e gestão de problemas (vg. *bugs*¹).

De igual forma, quando uma aplicação é desenvolvida por vários programadores, não se espera que trabalhe um de cada vez, ficando os outros à espera. Neste guião iremos abordar a gestão de projetos através da plataforma **GitHub**.

¹Ver http://en.wikipedia.org/wiki/Software_bug

4.2 A plataforma GitHub

A plataforma **GitHub** foi desenvolvido por Chris Wanstrath, J. Hyett, Tom Preston-Werner e Scott Chacon e começou em fevereiro de 2008. A empresa, GitHub, Inc., existe desde 2007 e está localizada em São Francisco. Em junho de 2018 foi comprada pela Microsoft [1].

A plataforma **GitHub**, que pode ser encontrada em <https://github.com>, é um sistema de gestão de projetos disponível para qualquer utilizador da Universidade de Aveiro.

Foi implementado usando a *framework* Ruby on Rails e inclui diversas ferramentas para facilitar a gestão de projetos de *software*, nomeadamente a criação de repositórios Git.

A plataforma pode ser utilizada para a gestão de qualquer tipo de projeto, mas é particularmente útil para projetos de desenvolvimento de aplicações, que mais podem beneficiar do sistema de controlo de versões de software coordenado com as ferramentas de gestão de tarefas e de funcionalidades diversas.

Os alunos podem usar a plataforma **GitHub** para gerir projetos das suas disciplinas, os professores utilizam-na para criar e partilhar conteúdos das aulas e gerir projetos de avaliação dos alunos e todos a podem utilizar para divulgar trabalhos que realizem durante o seu contacto com a *UA*.

A Figura 4.1 mostra a página inicial da plataforma **GitHub**.

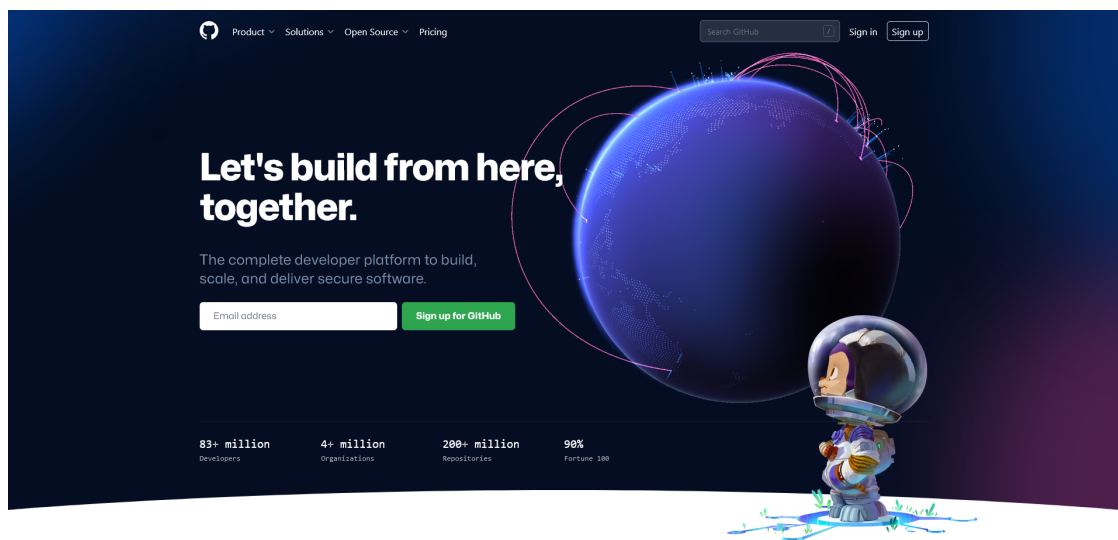


Figura 4.1: Página de entrada da plataforma **GitHub**.

Esta página permite fazer: o registo de novos utilizadores (**Sign up**); ou o acesso a utilizadores já registados (**Sign in**).

Todos os membros da *UA* podem possuir uma conta na plataforma **GitHub**, bastando para isso identificarem-se perante o sistema usando as credenciais de utilizador universal da *UA*. No caso dos alunos e professores do Departamento de Eletrónica, Telecomunicações e Informática é relevante a existência da organização DETI-UA onde estão colocados todos os seus projetos/repositórios.

Após o acesso bem sucedido e caso seja o seu primeiro acesso ao **GitHub** (ver Figura 4.2) ele é convidado a identificar-se preenchendo o ficheiro README. Para esse efeito basta selecionar o botão **Continue** e preencher alguns dados pessoais.

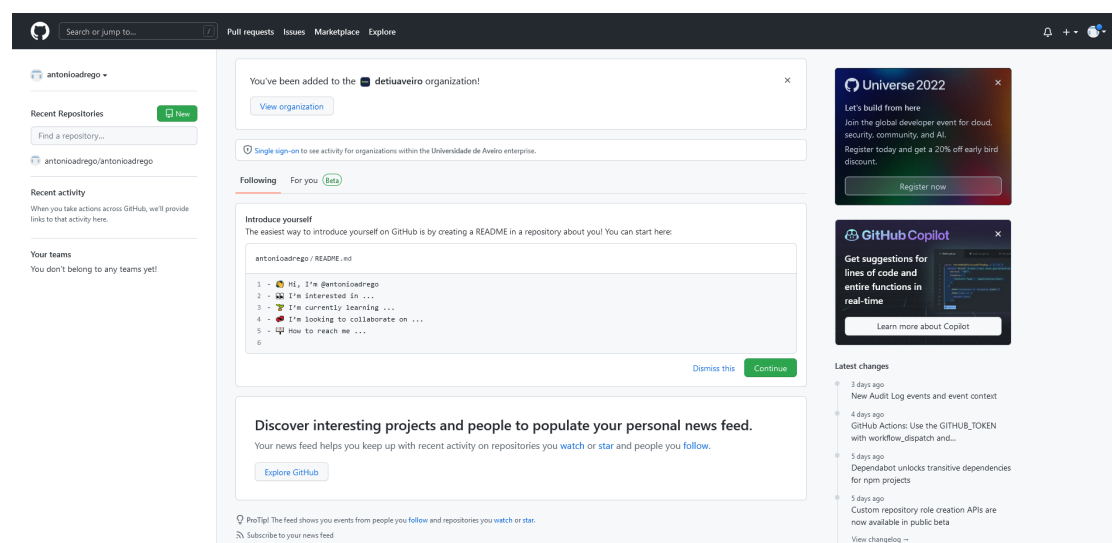


Figura 4.2: Página de entrada do utilizador do **GitHub** no seu primeiro acesso.

Exercício 4.1

Aceda ao **GitHub** no endereço <https://github.com> e registe-se como utilizador. De seguida faça *login*.

Nos acessos seguintes a página de entrada muda de aspeto tal como se apresenta na Figura 4.3. E logo no início da página permite aceder à organização **detiaveiro** clicando no botão **View organization**.

Exercício 4.2

Faça *logout* do **GitHub** e volte a fazer *login*.

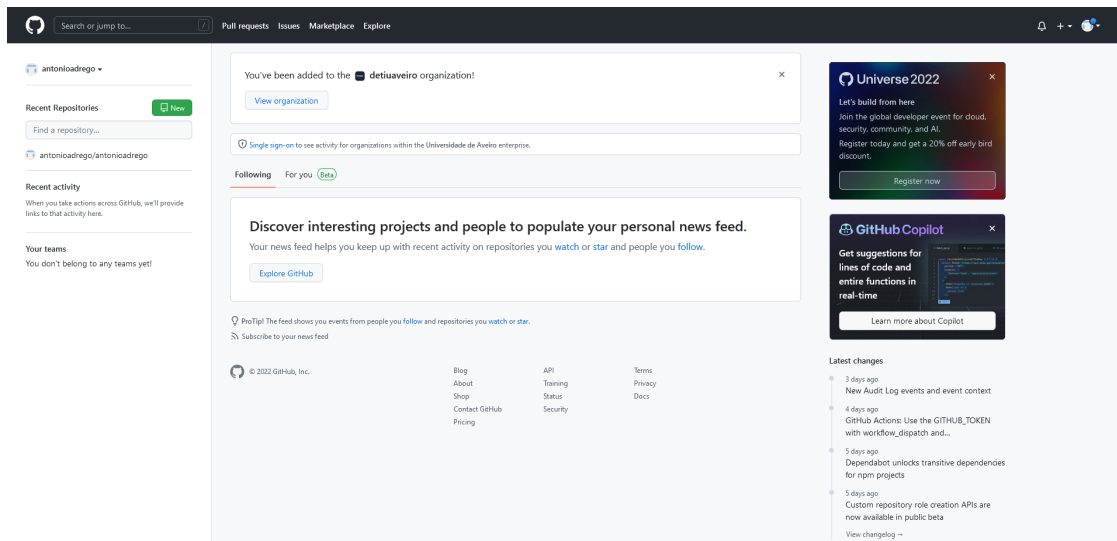


Figura 4.3: Página de entrada do utilizador do **GitHub** depois do seu primeiro acesso.

Após a devida autenticação com sucesso o utilizador acede à página de entrada do departamento (ver Figura 4.4) onde são apresentados alguns dos repositórios mais populares, sendo também possível ver e/ou pesquisar os repositórios já existentes.

A partir deste momento o utilizador pode criar projetos e/ou repositórios e gerir o seu funcionamento.

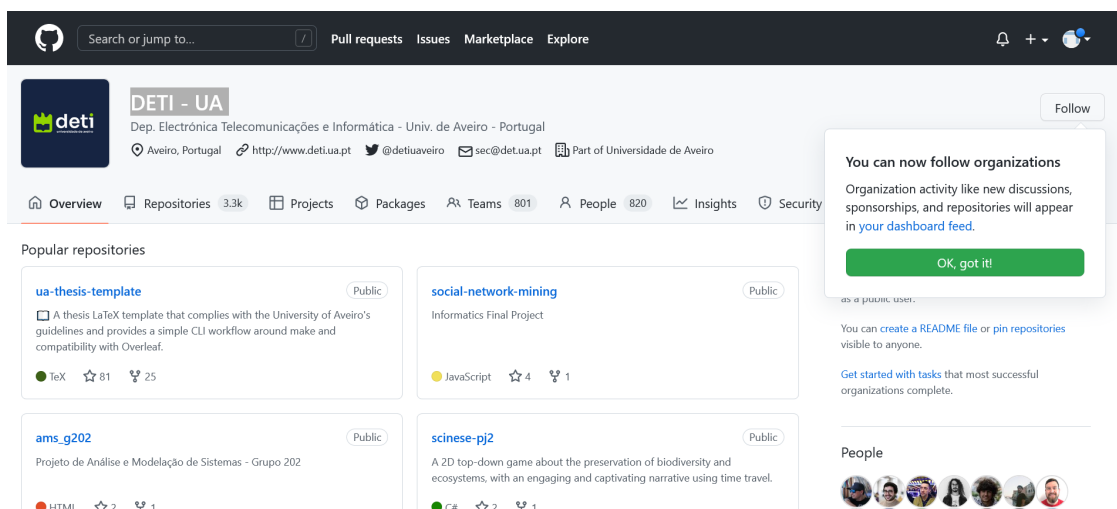


Figura 4.4: Página de entrada da organização **DETI - UA**.

4.3 Repositórios

Como o próprio nome **GitHub** dá a entender esta plataforma é orientada à criação e gestão de repositórios Git. Vamos agora ver como criar um repositório Git.

Como se pode ver na Figura 4.4 no canto superior direito existe um menu de operações do tipo *drop-down* que permite aceder a dados sobre o utilizador da plataforma, entre outros: ao seu perfil; aos seus repositórios; aos seus projetos; e às organizações de que é membro. Neste menu também podemos fazer *logout*.

Repare que alguns destes dados também estão acessíveis através dos botões que se encontram na parte inferior do cabeçalho superior da página debaixo do símbolo da *UA*. Mas no caso do botão de repositórios (segundo botão do lado esquerdo) acedemos a todos os repositórios e não apenas aos nossos. Para esse efeito temos que seleccionar no menu de operações a opção *Your repositories* e todos os repositórios do utilizador caso ele já tenha criados alguns serão apresentados tal como se mostra na Figura 4.5.

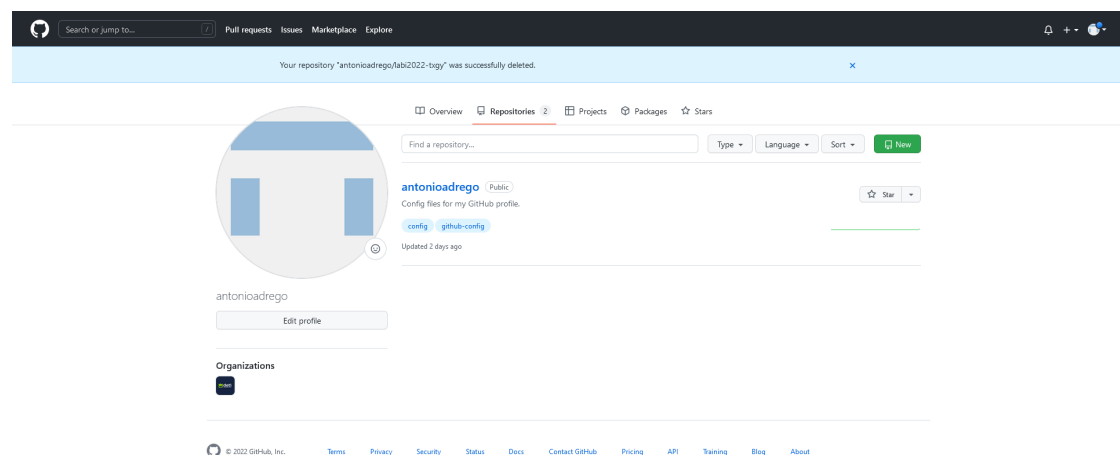


Figura 4.5: Página dos repositórios do utilizador.

Repare que existe um repositório, que é criado por defeito, com o mesmo nome do utilizador e que contém os dados pessoais que foram inseridos no primeiro acesso. No lado esquerdo desta página existe o botão verde **New** que permite criar repositórios. Vamos criar um repositório novo com o nome **infor2022-txgy**.

A Figura 4.6 apresenta a página de criação de um novo repositório designado por `infor2022-txgy`. Este novo repositório pertence ao utilizador indicado `antonioadrego`, tem uma descrição "**Repositório de teste para treinar a utilização dos comandos GIT**", ele é privado e também foi indicado que se pretende colocar no repositório um ficheiro `README`. Este ficheiro além de servir para descrever a utilidade do repositório, permite criar um repositório não vazio.

Repository template
Start your repository with a template repository's contents.
No template

Owner *
antonioadrego / Repository name *
infor2022-txgy

Great repository names are short and memorable. Need inspiration? How about [redesigned-chainsaw](#)?

Description (optional)
Repositório de teste para treinar os comandos GIT

☐ Public
Anyone on the internet can see this repository. You choose who can commit.

☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.
☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add gitignore
Choose which files not to track from a list of templates. [Learn more.](#)
gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)
License: None

This will set `main` as the default branch. Change the default name in your [settings](#).

☐ You are creating a private repository in your personal account.

Create repository

Figura 4.6: Criação de um novo repositório.

Para concretizar a criação do repositório é só clicar no botão **Create repository** e acedemos à página que se apresenta na Figura 4.7. Nesta página temos acesso ao ficheiro `README` que já está preenchido com o comentário que foi colocado na página anterior e que pode ser editado.

Um repositório é sempre criado por um utilizador mas normalmente vai ser utilizado por um grupo de programadores. Por isso há a necessidade de adicionar (convidar) outros utilizadores que podem usar o repositório. Para esse efeito usando o botão menu de operações e seleccionando a opção **Settings** podemos aceder a uma página de configuração do sistema.

Nesta página o utilizador selecciona no menu do lado esquerdo a opção **Repositories**, ver a lista dos seus repositórios e para cada um deles aceder à lista de **collaborators**. Nesta lista é indicado o número de colaboradores e quando ela é seleccionada é apresentado ao utilizador a página que se mostra na Figura 4.8 onde é possível convidar outros utilizadores da plataforma **GitHub** a terem acesso ao repositório.

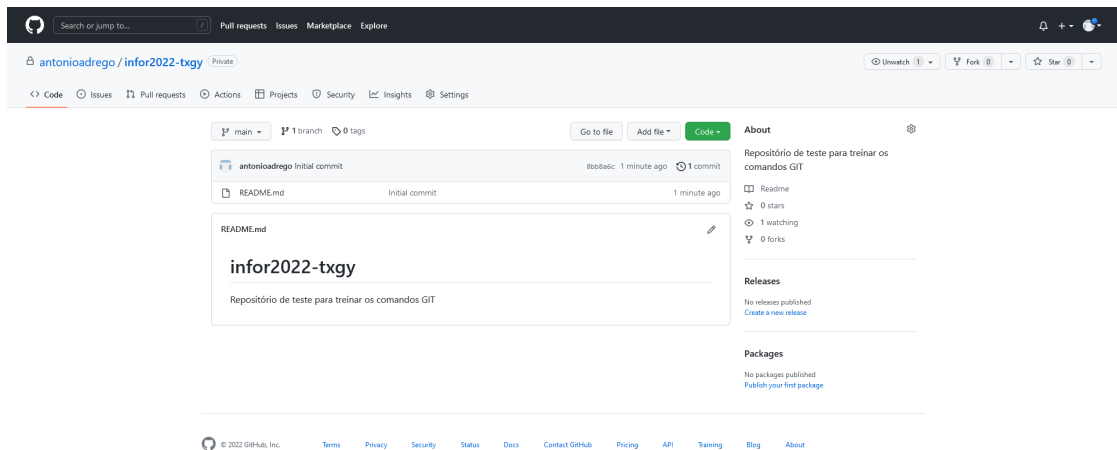


Figura 4.7: Página do novo repositório.

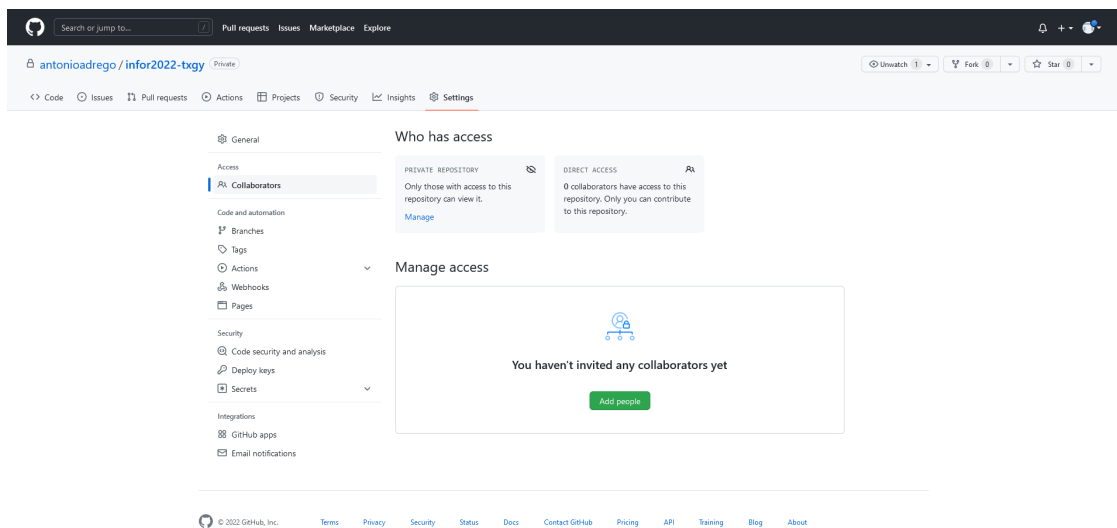


Figura 4.8: Adição de colaboradores do repositório.

Exercício 4.3

Crie um repositório com o identificador **infor2023-alunoN** em que N representa o seu número de aluno na disciplina. Use o mesmo N que está associado ao *login* de aluno que está no ficheiro **Logins no servidor deti-labi.ua.pt**.

Convide pelo menos um colega a ser colaborador do repositório criado.

4.4 Acesso remoto aos repositórios

Um repositório Git criado numa qualquer plataforma de trabalho colaborativo, como é o caso do **GitHub**, serve sempre como repositório central remoto para projetos de *software*. Pelo que os elementos do grupo de trabalho têm um repositório local no seu computador pessoal para trabalharem individualmente nas tarefas que lhe foram distribuídas. Depois podem atualizar o seu repositório local a partir do repositório remoto ou vice-versa.

Assim o primeiro passo consiste em criar o repositório central remoto, eventualmente com o modelo que vai servir como código base, e depois cada elemento do grupo deve cloná-lo para o seu computador pessoal e obter o repositório local onde deve trabalhar. O acesso ao repositório central tem de ser feito obviamente de forma segura de maneira a que nenhum elemento exterior ao grupo e não devidamente autorizado possa aceder-lhe.

4.4.1 Acesso por SSH

Para aceder a um repositório **GitHub** por Secure Shell (SSH) é preciso usar uma senha do tipo chave SSH protegida.

Para esse efeito é preciso proceder da seguinte forma. Através do botão menu de operações e selecionando a opção **Settings** acedemos à página de configuração do sistema e seleciona-se no menu do lado esquerdo a opção **SSH and GPG keys**.

Então é apresentada a página que se mostra na Figura 4.9 onde se pode inserir uma senha do tipo `ssh-rsa` na caixa de texto Key.

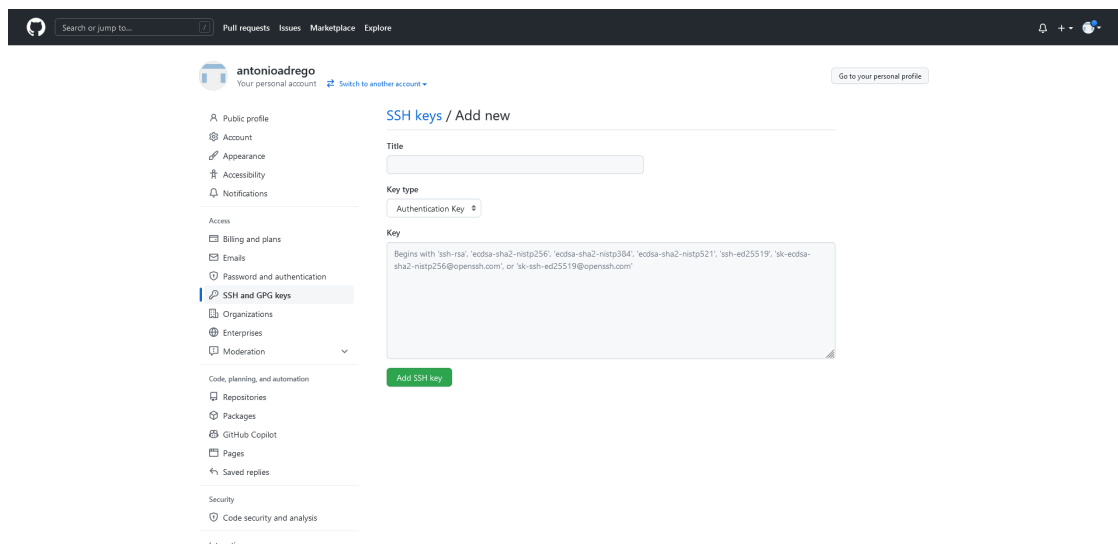


Figura 4.9: Página da chave de acesso SSH.

A Figura 4.10 mostra a introdução dos dados e depois de se clicar no botão Add SSH Key obtemos a página que se apresenta na Figura 4.11.

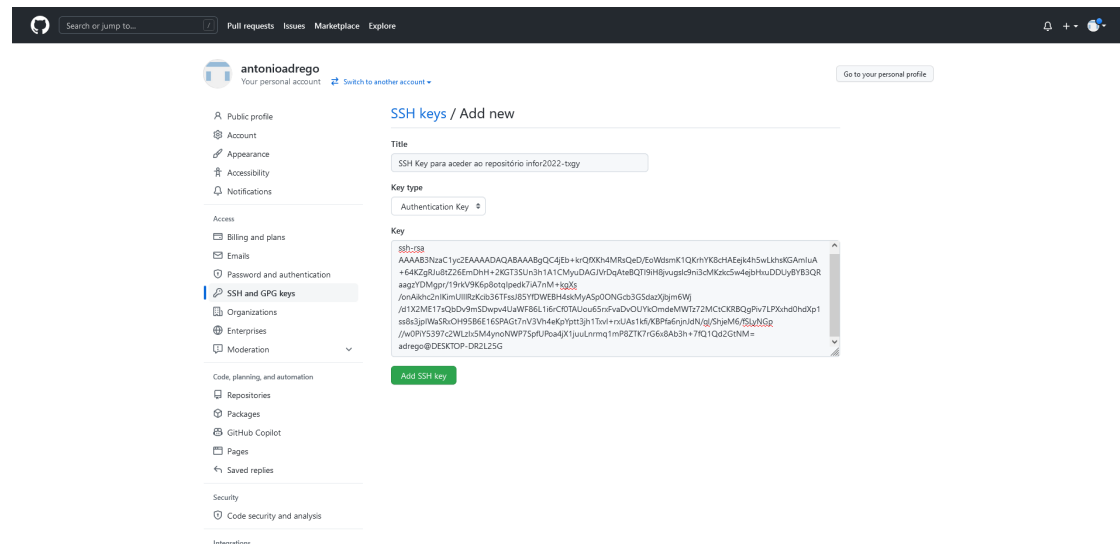


Figura 4.10: Página da chave de acesso SSH preenchida.

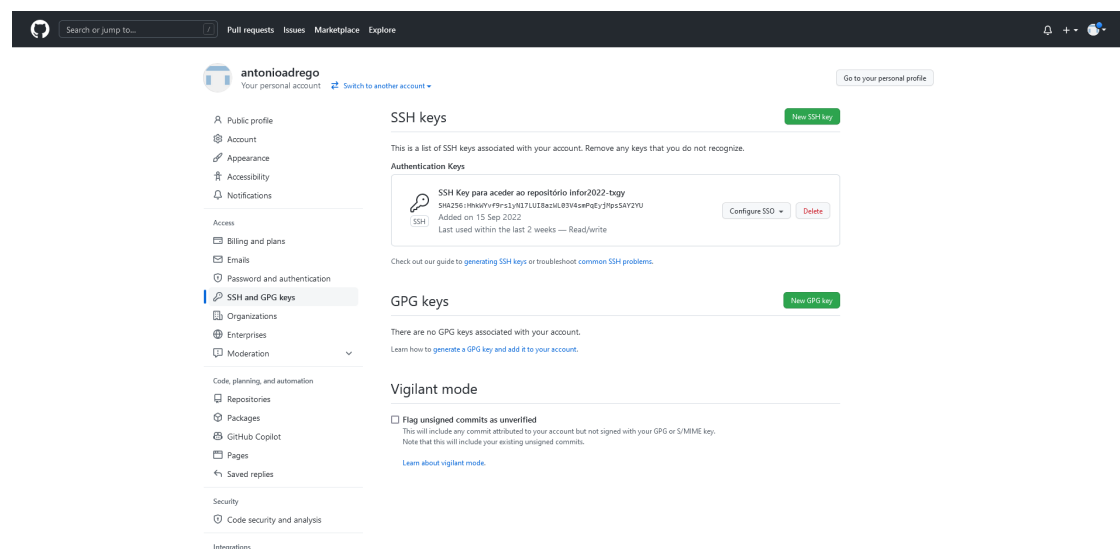


Figura 4.11: Página da chave de acesso SSH para validação.

Nota muito importante: para validar esta senha o utilizador deve clicar no botão Configure SSO e autenticar-se com as credenciais de utilizador universal da UA.

Para saber ou relembrar o que é uma senha do tipo ssh-rsa deve consultar o guião Tema 2 - Sistema Operativo de Linux, secção Acesso a servidores remotos, sub-secção Autenticação por chaves.

Se programou o acesso ao servidor `deti-labi` com autenticação por chaves pode usar a senha pública `ssh-rsa` (designada por `id_rsa.pub`) que se encontra no diretório `~/.ssh`.

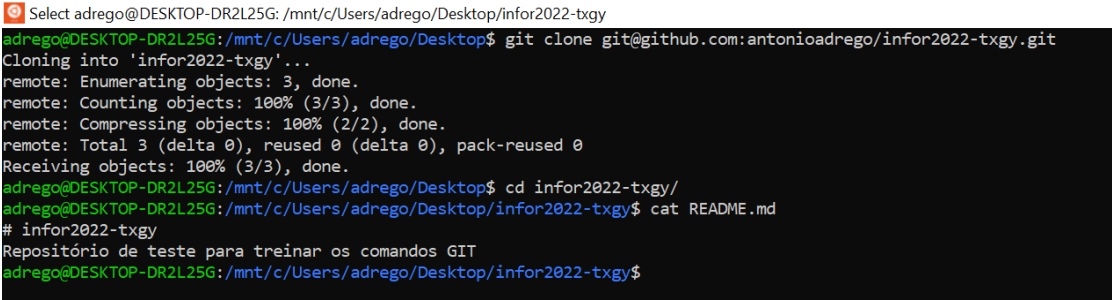
Como se pode ver na Figura 4.7 na página do repositório existe o botão **Code** que permite obter os endereços necessários para fazer a clonagem do repositório. Se escolhermos a opção **SSH** o comando deve usar o seguinte endereço de computador remoto:

```
git@github.com:antonioadrego/infor2022-txgy.git
```

Para clonar um repositório Git é preciso usar o comando `git clone`. Primeiro o utilizador deve colocar-se no diretório aonde pretende colocar o diretório *clone*. Depois executa o comando usando o endereço acima referido como se apresenta na Figura 4.12.

Nota muito importante: neste tipo de acesso não precisa de fazer nada porque o computador valida a operação no **GitHub** usando a senha pública `ssh-rsa`.

Para aceder ao repositório executa-se o comando `cd infor2022-txgy` e se executar o comando `cat README.md` visualizamos no monitor o comentário que foi colocado quando o repositório foi criado. A clonagem do repositório foi feita com sucesso, o que se pode inferir também, pela ausência de mensagens de erro na operação `git clone`.



```
Select adrego@DESKTOP-DR2L25G: /mnt/c/Users/adrego/Desktop/infor2022-txgy
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop$ git clone git@github.com:antonioadrego/infor2022-txgy.git
Cloning into 'infor2022-txgy'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop$ cd infor2022-txgy/
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop/infor2022-txgy$ cat README.md
# infor2022-txgy
Repositório de teste para treinar os comandos GIT
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop/infor2022-txgy$
```

Figura 4.12: Clonagem do repositório `infor2022-txgy` com SSH.

Exercício 4.4

Programa o acesso ao seu repositório com uma senha pública `ssh-rsa` e clone-o para o seu computador pessoal.

O acesso por **SSH** é recomendado porque a autenticação é mais simples e rápida. No entanto tem uma limitação à sua utilização. A senha pública `ssh-rsa` só pode ser usada pelo computador que a gerou. Caso um utilizador tenha mais do que um computador ele deve usá-la no computador que mais frequentemente utiliza para aceder aos seus repositórios na plataforma **GitHub**.

4.4.2 Acesso por HTTP

Para aceder a um repositório **GitHub** por HyperText Transfer Protocol (HTTP)[2] é preciso usar um *token* de segurança. Logo, primeiro é preciso criar o *token* procedendo da seguinte forma. Através do botão menu de operações e selecionando a opção **Settings** acedemos à página de configuração do sistema. Nesta página o utilizador seleciona no menu do lado esquerdo a última opção **Developer settings** e na página que lhe é exibida seleciona a opção do menu do lado esquerdo identificada por uma chave e designada por **Personal access tokens** obtendo a página que se apresenta na Figura 4.13.

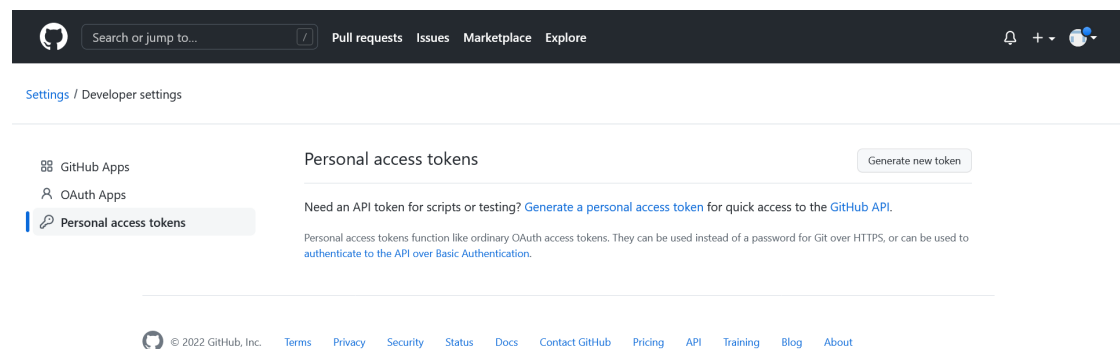


Figura 4.13: Página inicial de requisição do *token* de segurança.

Agora nesta página é preciso clicar no botão do lado direito **Generate new token** e será apresentada uma página cuja parte superior se apresenta na Figura 4.14. Existe um campo para escrever um comentário explicativo da utilidade do *token* de segurança, o período de duração do mesmo e seleciona-se o bloco inicial **repo**.

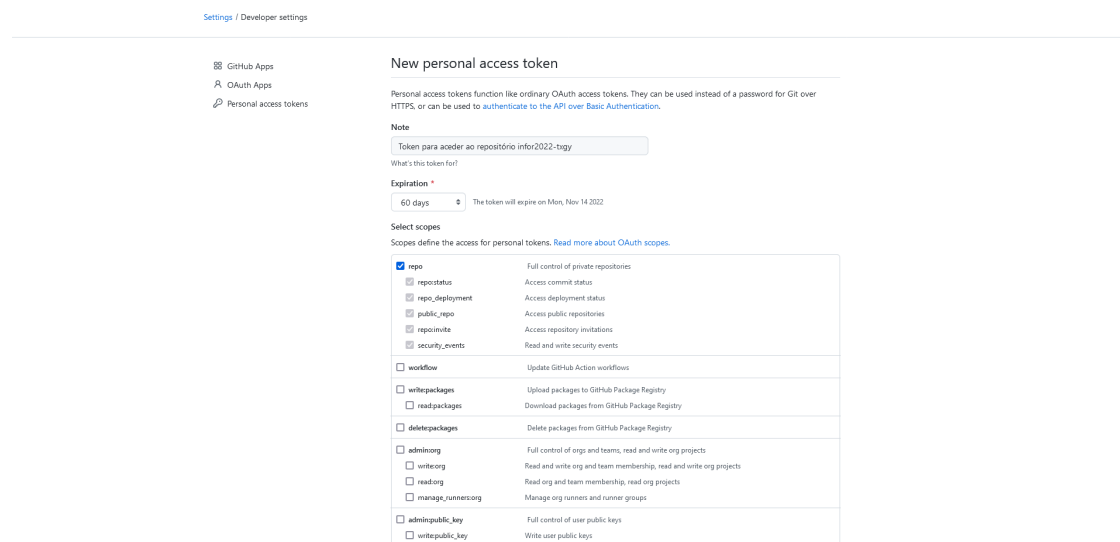


Figura 4.14: Parte superior da página de requisição do *token* de segurança.

Na parte inferior da página que se apresenta na Figura 4.15 pode-se eventualmente indicar que o token também serve para remover o repositório e depois clica-se no botão verde **Generate token** para gerar o *token* de segurança.

<input type="checkbox"/> readrepo_hook	Read repository hooks
<input type="checkbox"/> adminorg_hook	Full control of organization hooks
<input type="checkbox"/> git	Create gists
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update ALL user data
<input type="checkbox"/> readuser	Read ALL user profile data
<input type="checkbox"/> useremail	Access user email addresses (read-only)
<input type="checkbox"/> userfollow	Follow and unfollow users
<input checked="" type="checkbox"/> delete_repo	Delete repositories
<input type="checkbox"/> writediscussion	Read and write team discussions
<input type="checkbox"/> readdiscussion	Read team discussions
<input type="checkbox"/> adminenterprise	Full control of enterprises
<input type="checkbox"/> manage_runnersenterprise	Manage enterprise runners and runner groups
<input type="checkbox"/> manage_billingenterprise	Read and write enterprise billing data
<input type="checkbox"/> readenterprise	Read enterprise profile data
<input type="checkbox"/> project	Full control of projects
<input type="checkbox"/> readproject	Read access of projects
<input type="checkbox"/> admin_gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write_gpg_key	Write public user GPG keys
<input type="checkbox"/> read_gpg_key	Read public user GPG keys
<input type="checkbox"/> admin_ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write_ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read_ssh_signing_key	Read public user SSH signing keys

Generate token Cancel

Figura 4.15: Parte inferior da página de requisição do *token* de segurança.

Então será apresentado o *token* de segurança gerado pelo **GitHub** numa nova página, tal como se mostra na Figura 4.16 e que o utilizador deve memorizar ou guardar de forma segura pois vai precisar de o usar sempre que precisar de aceder ao repositório.

Settings / Developer settings

Personal access tokens

Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_5DrRSf9gRIUbWu8iba54kzZbtBS1St0eBC75 [Copy](#) Configure SSO Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Figura 4.16: Página de requisição do *token* de segurança para validação.

Neste caso foi gerado o *token* **ghp_5DrRSf9gRIUbWu8iba54kzZbtBS1St0eBC75**.

Notas muito importantes: para validar este *token* o utilizador deve clicar no botão **Configure SSO** e autenticar-se com as credenciais de utilizador universal da *UA*. Deve também guardá-lo num ficheiro de texto para poder utilizá-lo sempre que necessário.

Tal como já foi referido anteriormente, na página do repositório (ver Figura 4.7) existe o botão **Code** que permite obter os endereços necessários para fazer a clonagem do repositório. Se escolhermos a opção **HTTP** o comando deve usar o seguinte endereço URL:

`https://github.com/antonioadrego/infor2022-txgy.git`

Também já foi referido anteriormente, que para clonar um repositório Git é preciso usar o comando **git clone**. Primeiro o utilizador deve colocar-se no diretório aonde pretende colocar o diretório *clone*. Depois executa o comando usando o endereço acima referido como se apresenta na Figura 4.17.



```
adrego@DESKTOP-DR2L25G: /mnt/c/Users/adrego/Desktop/infor2022-txgy
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop$ git clone https://github.com/antonioadrego/infor2022-txgy.git
Cloning into 'infor2022-txgy'...
Username for 'https://github.com': antonioadrego
Password for 'https://antonioadrego@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 651 bytes | 10.00 KiB/s, done.
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop$ cd infor2022-txgy/
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop/infor2022-txgy$ ls -l
total 0
-rwxrwxrwx 1 adrego adrego 68 Sep 14 17:43 README.md
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop/infor2022-txgy$ cat README.md
# infor2022-txgy
Repositório de teste para treinar os comandos GIT
adrego@DESKTOP-DR2L25G:/mnt/c/Users/adrego/Desktop/infor2022-txgy$
```

Figura 4.17: Clonagem do repositório `infor2022-txgy` com HTTP.

Para validar a operação no **GitHub** tem de inserir o seu identificador de utilizador seguido do *token* de segurança.

Nota muito importante: quando se introduz o *token*, por exemplo usando o *Copy/Paste*, nenhum carácter é mostrado no monitor e deve de seguida clicar na tecla *Enter/Return*.

Para aceder ao repositório executa-se o comando **cd infor2022-txgy** e se executar o comando **cat README.md** visualizamos no monitor o comentário que foi colocado quando o repositório foi criado. A clonagem do repositório foi feita com sucesso, o que se pode inferir também, pela ausência de mensagens de erro na operação **git clone**.

O acesso por **HTTP** é recomendado para utilizadores que tenham e/ou que utilizem mais do que um computador para aceder à plataforma **GitHub**, porque é possível partilhar o *token* de segurança.

Exercício 4.5

Programe o acesso ao seu repositório com um *token* de segurança e clone-o para o seu computador pessoal.

Tenha em atenção que deve clonar o repositório para outro local no seu computador pessoal.

4.5 Notas finais

Quando se clona um repositório da plataforma **GitHub** no computador pessoal, o Git cria um *alias* para designar o repositório remoto evitando assim a utilização do URL (ver comandos **git clone** na Figura 4.12 e na Figura 4.17).

Normalmente esse *alias* é designado por **origin**. É essencialmente uma abreviação para o repositório remoto. Assim a sincronização dos repositórios locais com o repositório remoto (**GitHub**) utilizando os comandos **git push** e **git pull** fica simplificada.

Apesar de ser possível aceder aos repositórios da plataforma **GitHub** através de operações **upload** e **download** tal não deve ser feito. Os repositórios Git (locais e central) devem ser atualizados/sincronizados através dos comandos **git push** e **git pull**, de maneira a manter um efetivo controlo das sucessivas versões dos ficheiros armazenados.

4.6 Para aprofundar

Exercício 4.6

Explore os outros módulos da plataforma **GitHub** e identifique qual a sua utilidade para a gestão de projetos.

Glossário

HTTP HyperText Transfer Protocol

SSH Secure Shell

Referências

- [1] *GitHub*, <https://pt.wikipedia.org/wiki/GitHub>.
- [2] R. Fielding, J. Gettys, J. Mogul et al., *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616 (Draft Standard), Updated by RFCs 2817, 5785, 6266, Internet Engineering Task Force, jun. de 1999.