

# Utilizando Git com o Eclipse

Prof. Rafael Guterres Jeffman  
rafael.jeffman@gmail.com

# Objetivos

- Utilizar o Git em projetos Eclipse para as disciplinas do SENAC.
- Criar um repositório Git na máquina local.
- Configuração básica do Git.
- Utilizar o Eclipse em conjunto com o repositório criado.
- Sincronizar o repositório local com um repositório remoto.

# Ferramentas Utilizadas

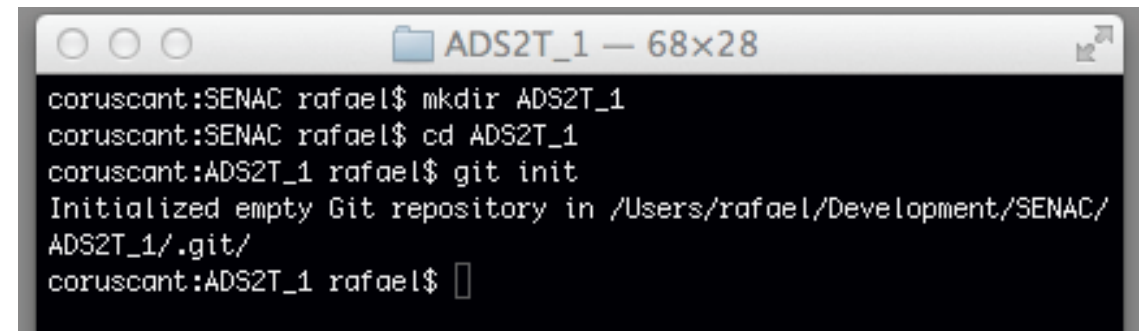
- Eclipse: para gerenciamento e edição do código fonte.
- Git: como sistema de controle de versão.
- Github: como hospedagem do repositório remoto.
- *Sugestão*: utilize o Git Bash ou Git GUI como ferramenta no SENAC. Fora do SENAC, utilize a ferramenta disponível ou selecione uma que lhe agrade no site <http://github.com>

# Configuração do Git

- Configurando o Proxy de conexão.
  - `git config --global http.proxy 192.168.0.1:8080`
- Retirando a conexão do Proxy.
  - `git config --global --unset http.proxy`
- Configurando o nome do usuário
  - `git config --global user.name "Rafael Jeffman"`
- Configurando o email do usuário
  - `git config --global user.email "rafael.jeffman@gmail.com"`

# Criação do Repositório

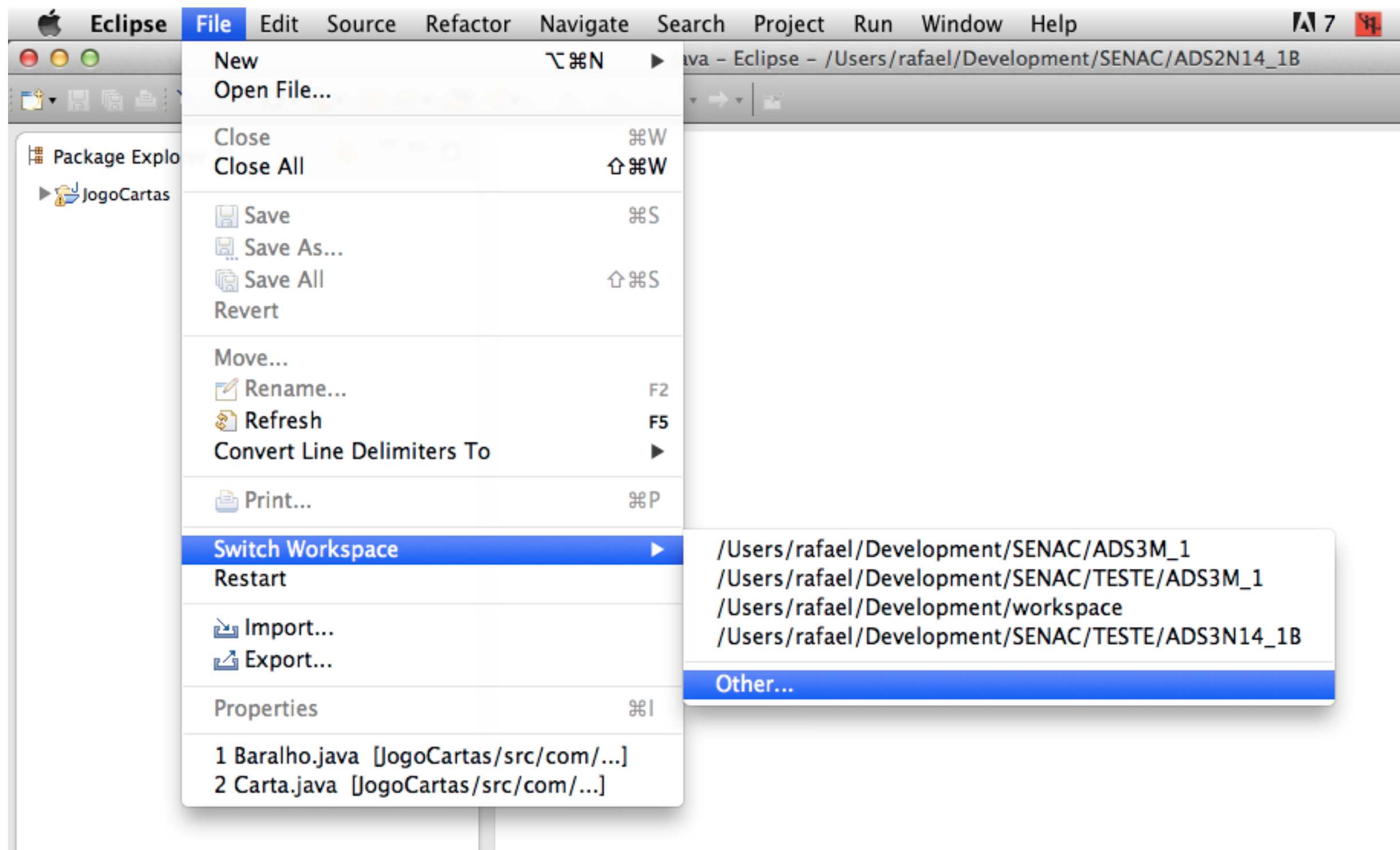
- Um repositório Git reside em um diretório do sistema de arquivos local.
- Para criar um repositório, você deve selecionar um diretório e inicializa o repositório com o comando 'git init'.

A screenshot of a terminal window titled "ADS2T\_1 — 68x28". The terminal shows the following commands and output:

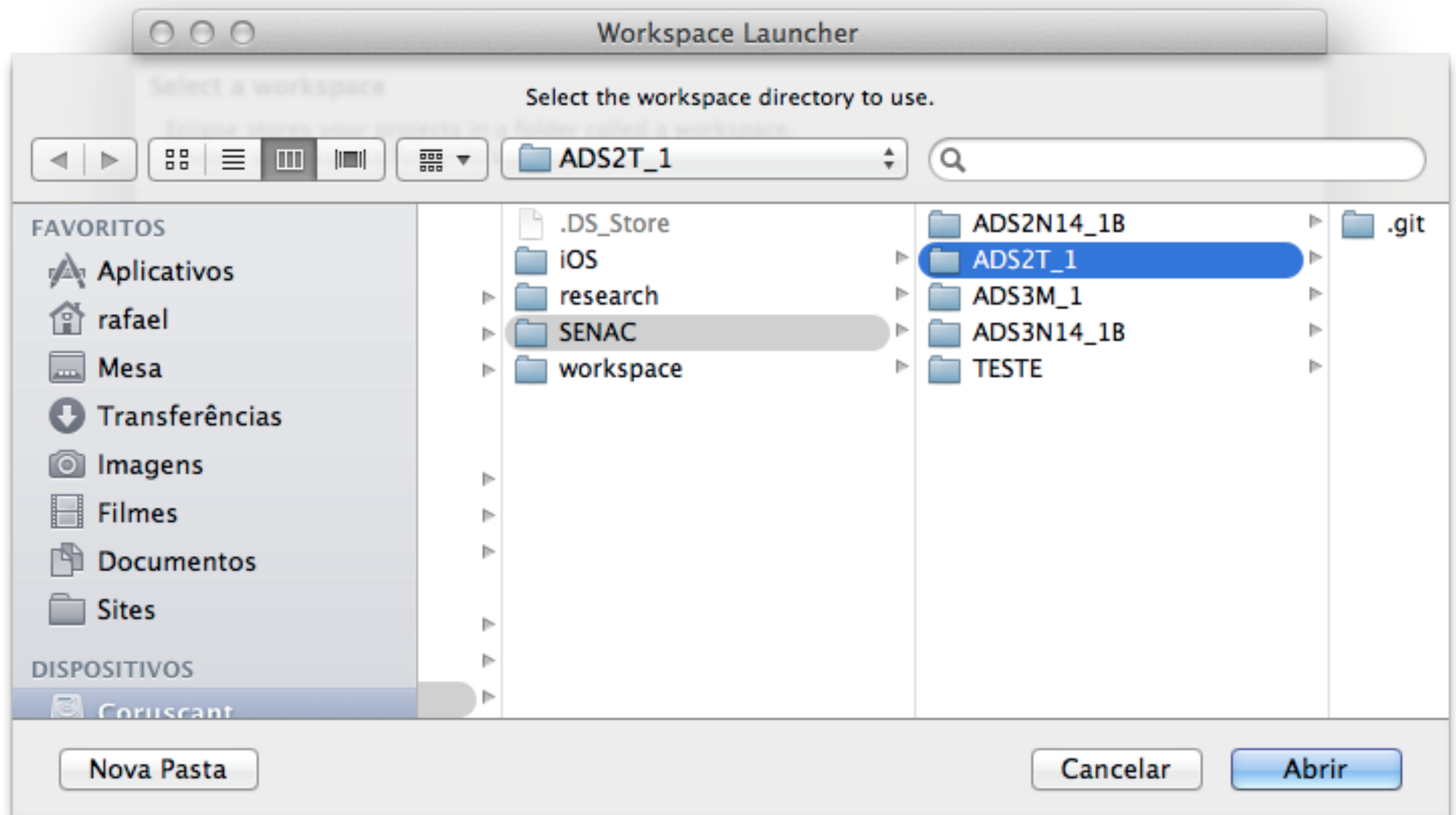
```
coruscant:SENAC rafael$ mkdir ADS2T_1
coruscant:SENAC rafael$ cd ADS2T_1
coruscant:ADS2T_1 rafael$ git init
Initialized empty Git repository in /Users/rafael/Development/SENAC/ADS2T_1/.git/
coruscant:ADS2T_1 rafael$
```

# Configuração do Eclipse

- Para evitar cópias manuais de arquivos, o ideal é configurar o Eclipse para utilizar o repositório como “Workspace”.
- Com isso, todos os projetos criados no Eclipse já estarão disponíveis no repositório, bastando utilizar o Git para controlar as suas modificações.

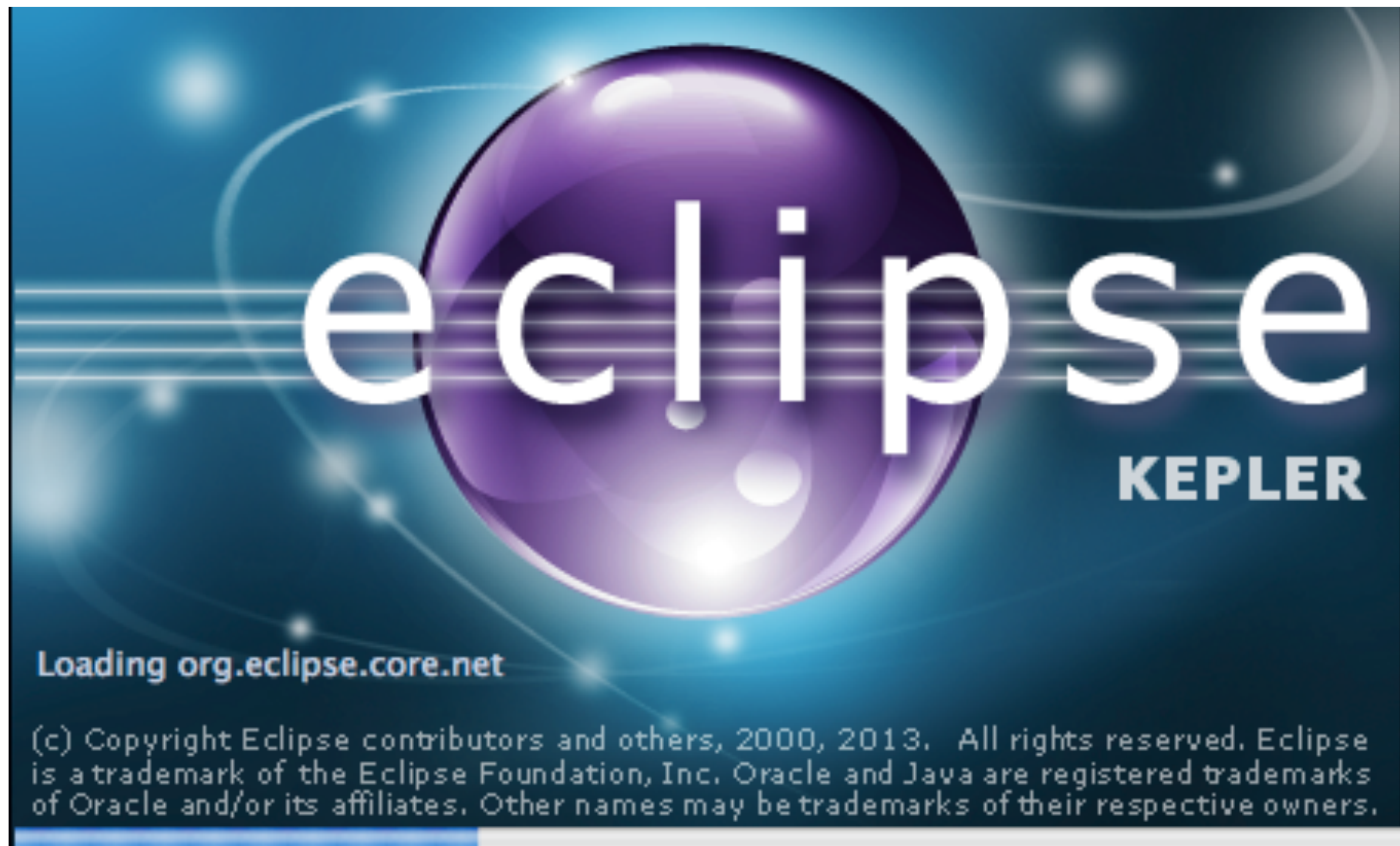


Selecione a opção de trocar de Workspace.

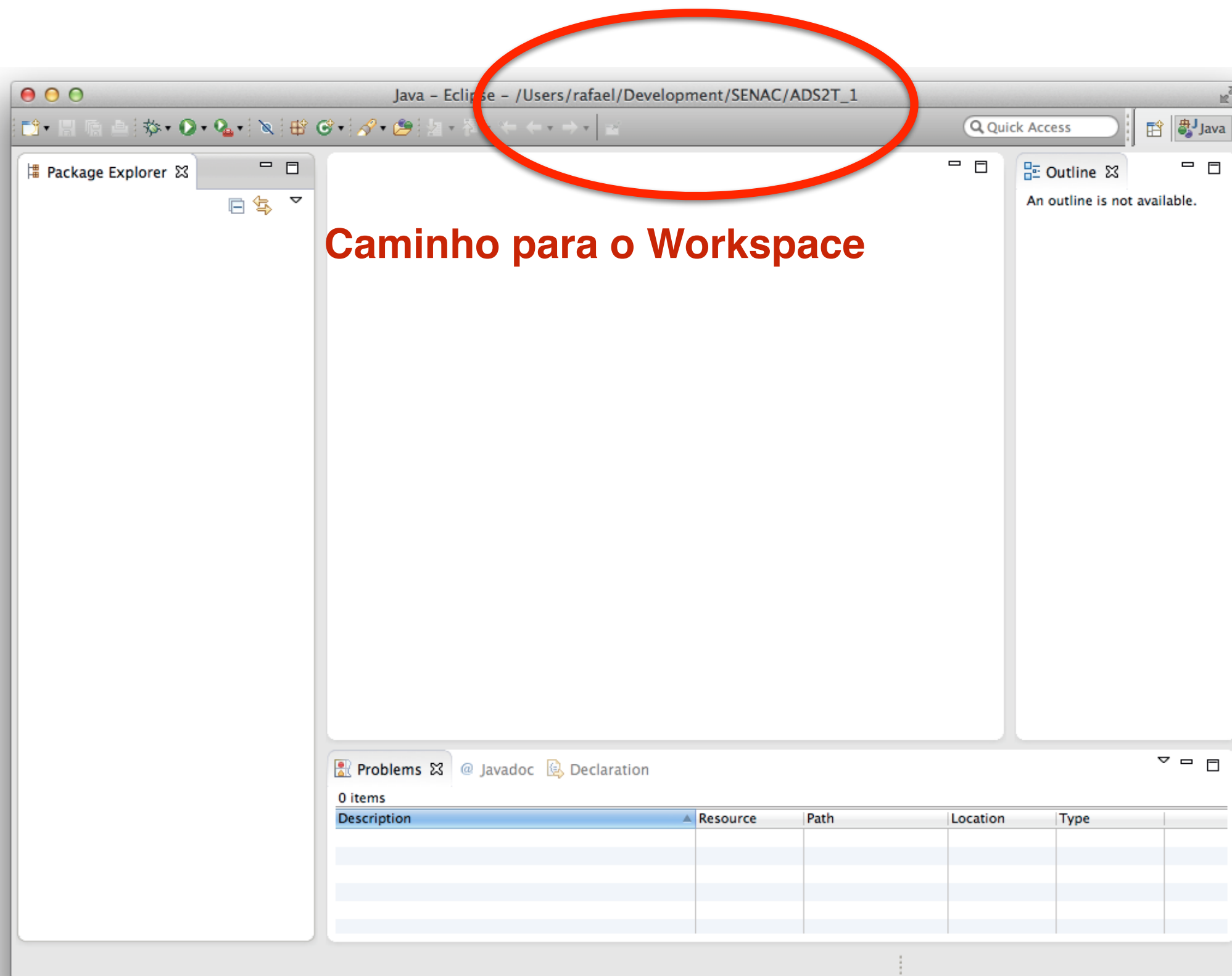


Selecione o diretório onde foi criado o repositório Git.

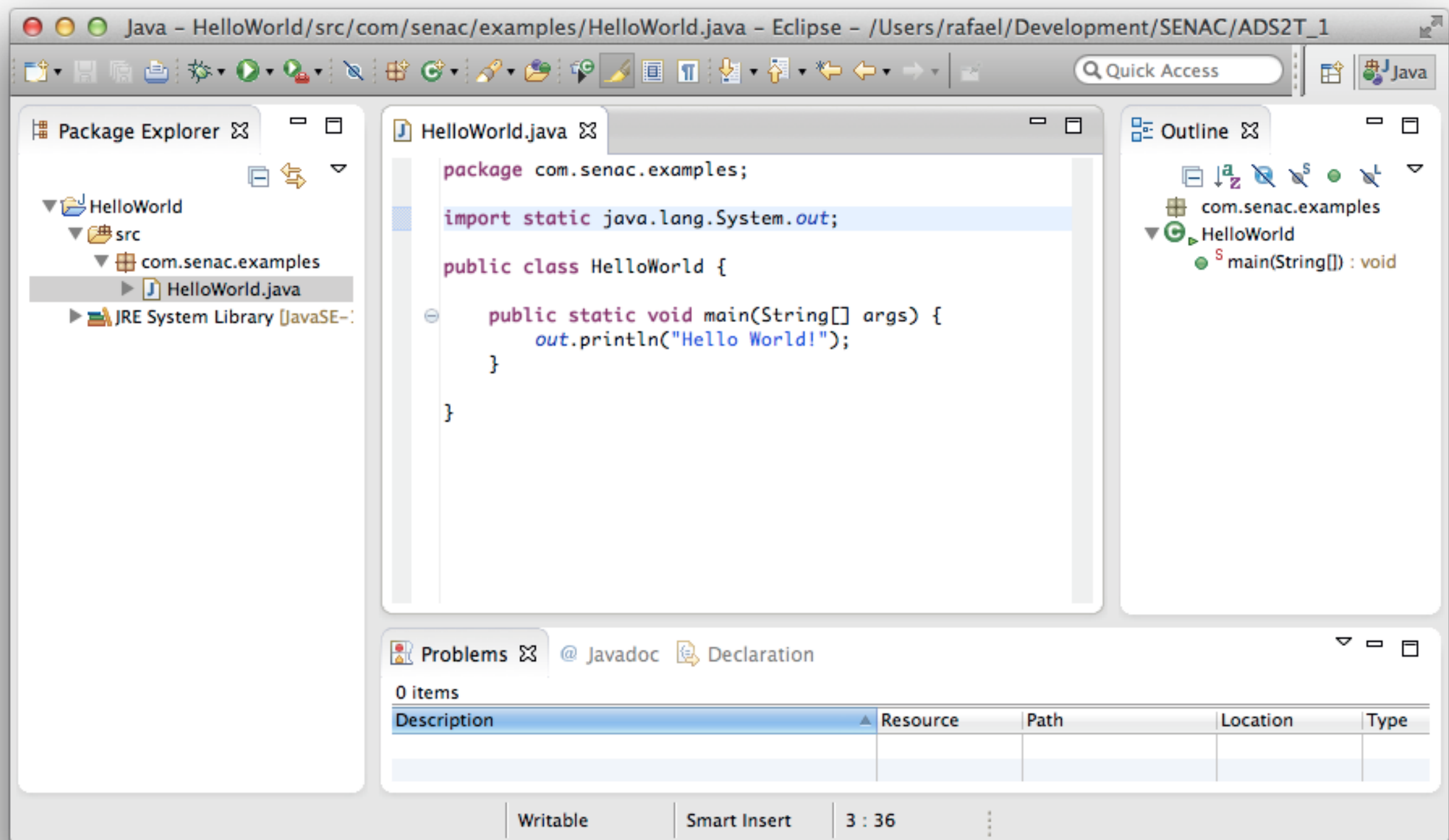




Ao trocar o workspace o Eclipse é reinicializado.



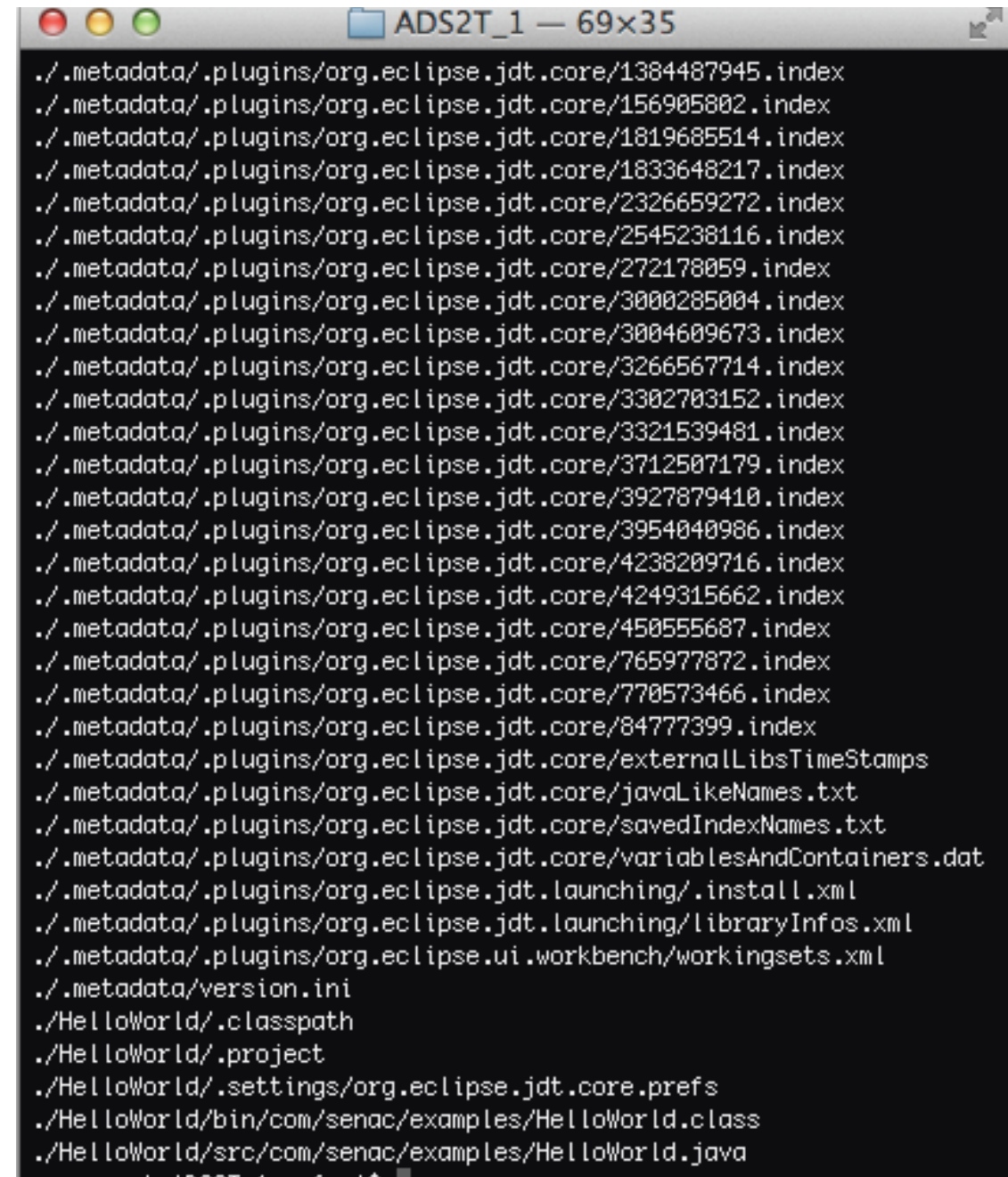
Ao reiniciar, o novo Workspace estará vazio.



Crie um novo projeto e uma nova classe.

# Arquivos

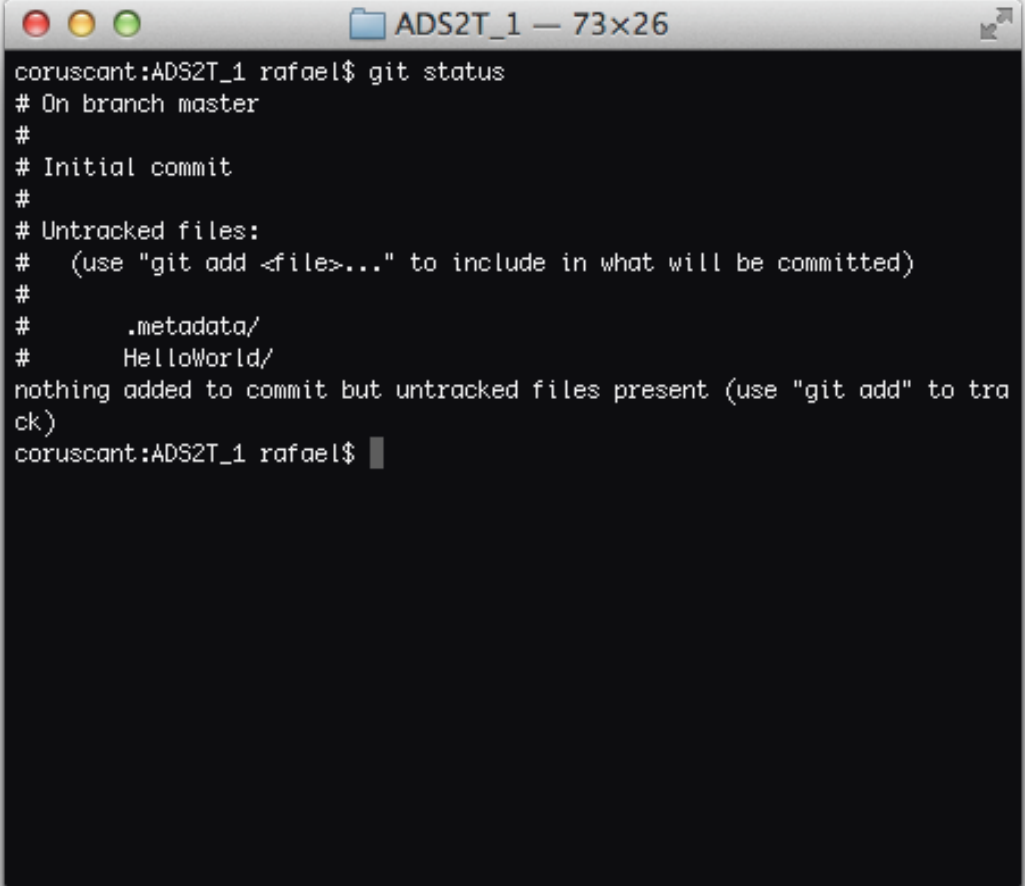
- Ao criar o novo workspace, o projeto e a nova classe, diversos arquivos foram adicionados ao diretório do repositório.
- Nem todos esses arquivos devem ser adicionados ao repositório.



```
ADS2T_1 — 69x35
./.metadata/.plugins/org.eclipse.jdt.core/1384487945.index
./.metadata/.plugins/org.eclipse.jdt.core/156905802.index
./.metadata/.plugins/org.eclipse.jdt.core/1819685514.index
./.metadata/.plugins/org.eclipse.jdt.core/1833648217.index
./.metadata/.plugins/org.eclipse.jdt.core/2326659272.index
./.metadata/.plugins/org.eclipse.jdt.core/2545238116.index
./.metadata/.plugins/org.eclipse.jdt.core/272178059.index
./.metadata/.plugins/org.eclipse.jdt.core/3000285004.index
./.metadata/.plugins/org.eclipse.jdt.core/3004609673.index
./.metadata/.plugins/org.eclipse.jdt.core/3266567714.index
./.metadata/.plugins/org.eclipse.jdt.core/3302703152.index
./.metadata/.plugins/org.eclipse.jdt.core/3321539481.index
./.metadata/.plugins/org.eclipse.jdt.core/3712507179.index
./.metadata/.plugins/org.eclipse.jdt.core/3927879410.index
./.metadata/.plugins/org.eclipse.jdt.core/3954040986.index
./.metadata/.plugins/org.eclipse.jdt.core/4238209716.index
./.metadata/.plugins/org.eclipse.jdt.core/4249315662.index
./.metadata/.plugins/org.eclipse.jdt.core/450555687.index
./.metadata/.plugins/org.eclipse.jdt.core/765977872.index
./.metadata/.plugins/org.eclipse.jdt.core/770573466.index
./.metadata/.plugins/org.eclipse.jdt.core/84777399.index
./.metadata/.plugins/org.eclipse.jdt.core/externalLibsTimeStamps
./.metadata/.plugins/org.eclipse.jdt.core/javaLikeNames.txt
./.metadata/.plugins/org.eclipse.jdt.core/savedIndexNames.txt
./.metadata/.plugins/org.eclipse.jdt.core/variablesAndContainers.dat
./.metadata/.plugins/org.eclipse.jdt.launching/.install.xml
./.metadata/.plugins/org.eclipse.jdt.launching/libraryInfos.xml
./.metadata/.plugins/org.eclipse.ui.workbench/workingsets.xml
./.metadata/version.ini
./HelloWorld/.classpath
./HelloWorld/.project
./HelloWorld/.settings/org.eclipse.jdt.core.prefs
./HelloWorld/bin/com/senac/examples/HelloWorld.class
./HelloWorld/src/com/senac/examples/HelloWorld.java
```

# Estado do Repositório

- Para verificar o estado do repositório utilizasse o comando 'git status'.
- O 'status' mostra o estado dos arquivos com relação ao repositório.
- O 'git' mostrará os arquivos modificados, novos, excluídos ou que não foram adicionados ao repositório.
- Nesse momento, não existe nenhum arquivo adicionado ao repositório.

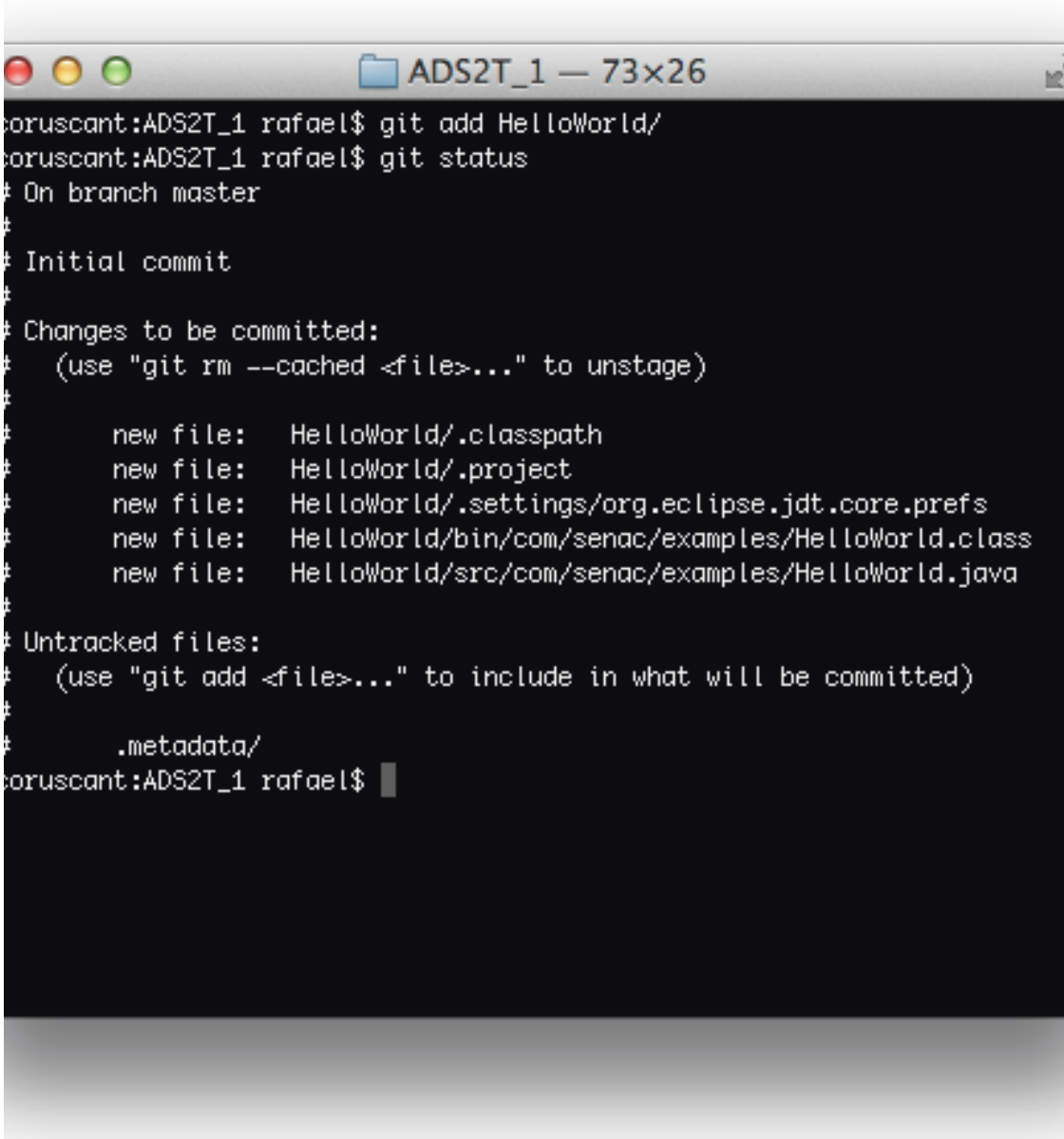
A terminal window titled 'ADS2T\_1 — 73x26' showing the output of the 'git status' command. The output indicates an initial commit on the master branch with untracked files in the .metadata/ and HelloWorld/ directories.

```
coruscant:ADS2T_1 rafael$ git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .metadata/
#       HelloWorld/
nothing added to commit but untracked files present (use "git add" to track)
coruscant:ADS2T_1 rafael$
```



# Adicionando apenas os arquivos corretos.

- Ao adicionar o diretório do projeto ao Git, todos os arquivos existentes são adicionados recursivamente.
- A adição dos arquivos desta forma, insere no repositório arquivos não desejados com os arquivos compilados e as preferências do programador.
- Para solucionar este problema deve ser utilizado um arquivo de configuração '.gitignore'.

A screenshot of a terminal window titled 'ADS2T\_1 — 73x26'. The terminal shows the execution of 'git add HelloWorld/' and 'git status'. The output indicates an initial commit on the master branch with five new files staged for commit: HelloWorld/.classpath, HelloWorld/.project, HelloWorld/.settings/org.eclipse.jdt.core.prefs, HelloWorld/bin/com/senac/examples/HelloWorld.class, and HelloWorld/src/com/senac/examples/HelloWorld.java. It also lists untracked files including .metadata/.

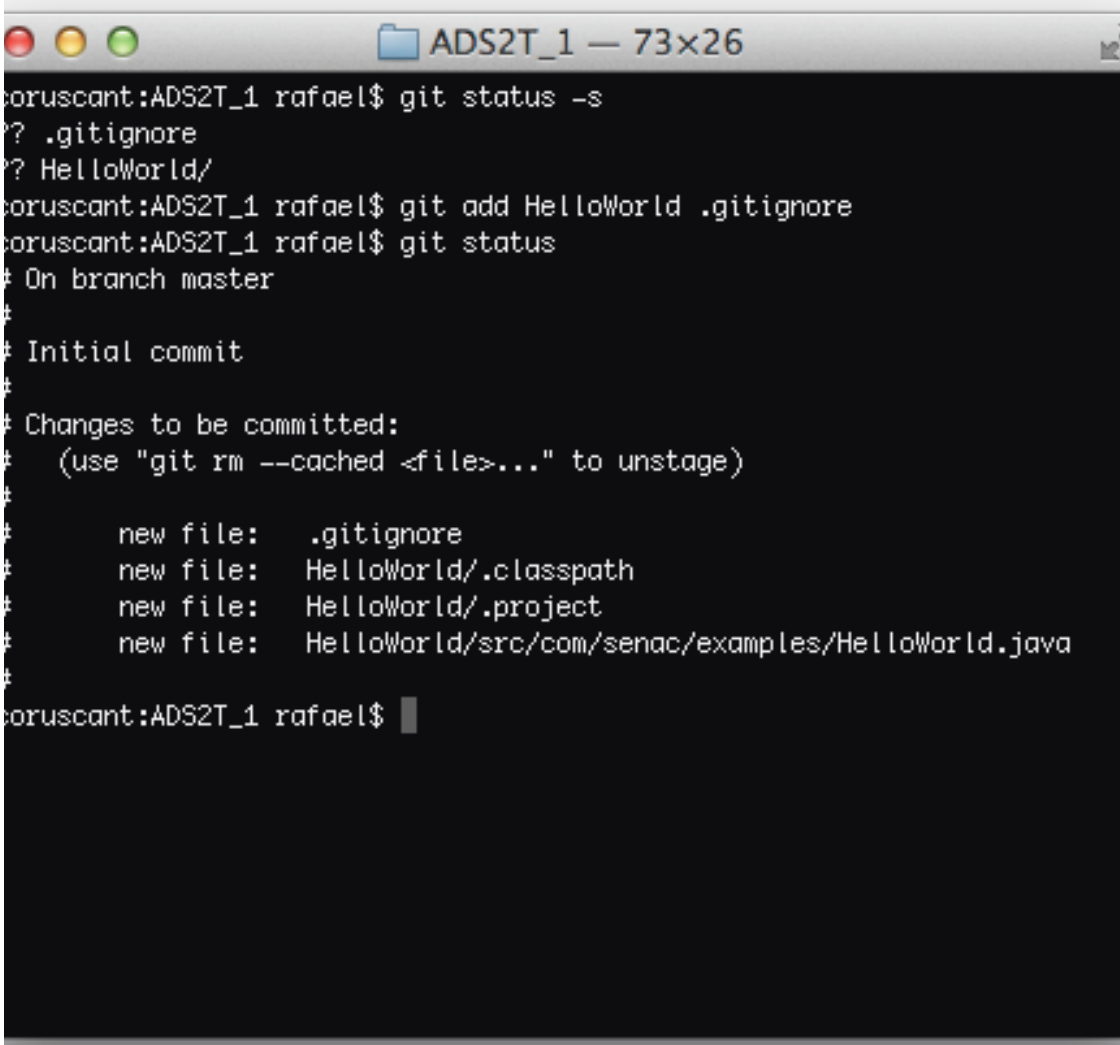
```
coruscant:ADS2T_1 rafael$ git add HelloWorld/
coruscant:ADS2T_1 rafael$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   HelloWorld/.classpath
#       new file:   HelloWorld/.project
#       new file:   HelloWorld/.settings/org.eclipse.jdt.core.prefs
#       new file:   HelloWorld/bin/com/senac/examples/HelloWorld.class
#       new file:   HelloWorld/src/com/senac/examples/HelloWorld.java
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .metadata/
coruscant:ADS2T_1 rafael$
```

# .gitignore

- Configura que tipos de arquivos o git deve ignorar quando executar um “add”.
- O arquivo “.gitignore” deve ser adicionado ao repositório como qualquer outro arquivo.
- Para funcionar corretamente, deve estar presente na raiz do repositório.
- *NOTA:* Obtenha um arquivo .gitignore do repositório de códigos de exemplo da disciplina.

# Utilizando o .gitignore

- Ao utilizar a configuração correta, apenas os arquivos necessários são inseridos no repositório.
- Desta forma, fica mais simples desenvolver em estações diferentes ou com diversos desenvolvedores, e diminui o tamanho do repositório, sem a adição de arquivos desnecessários.
- Caso você tenha adicionado arquivos que não queira no repositório, desfça a “adição” com o comando “git reset”.



```
coruscant:ADS2T_1 rafael$ git status -s
?? .gitignore
?? HelloWorld/
coruscant:ADS2T_1 rafael$ git add HelloWorld .gitignore
coruscant:ADS2T_1 rafael$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   .gitignore
#       new file:   HelloWorld/.classpath
#       new file:   HelloWorld/.project
#       new file:   HelloWorld/src/com/senac/examples/HelloWorld.java
coruscant:ADS2T_1 rafael$
```



# Gravando as alterações

- Para gravar permanentemente as alterações no repositório utilize o comando “git commit”.
- SEMPRE adicione uma mensagem ao “commit” de forma que seja possível recuperar o histórico do trabalho executado.
- Após o commit as alterações serão registradas no repositório local.



```
ADS2T_1 — 73x26
ant:ADS2T_1 rafael$ git commit -m "Adicionando o projeto HelloWorld no repositório."
[ (root-commit) 33ec32b] Adicionando o projeto HelloWorld no repositório
2 files changed, 64 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 HelloWorld/.classpath
 create mode 100644 HelloWorld/.project
 create mode 100644 HelloWorld/src/com/senac/examples/HelloWorld.java
ant:ADS2T_1 rafael$
```

# Github

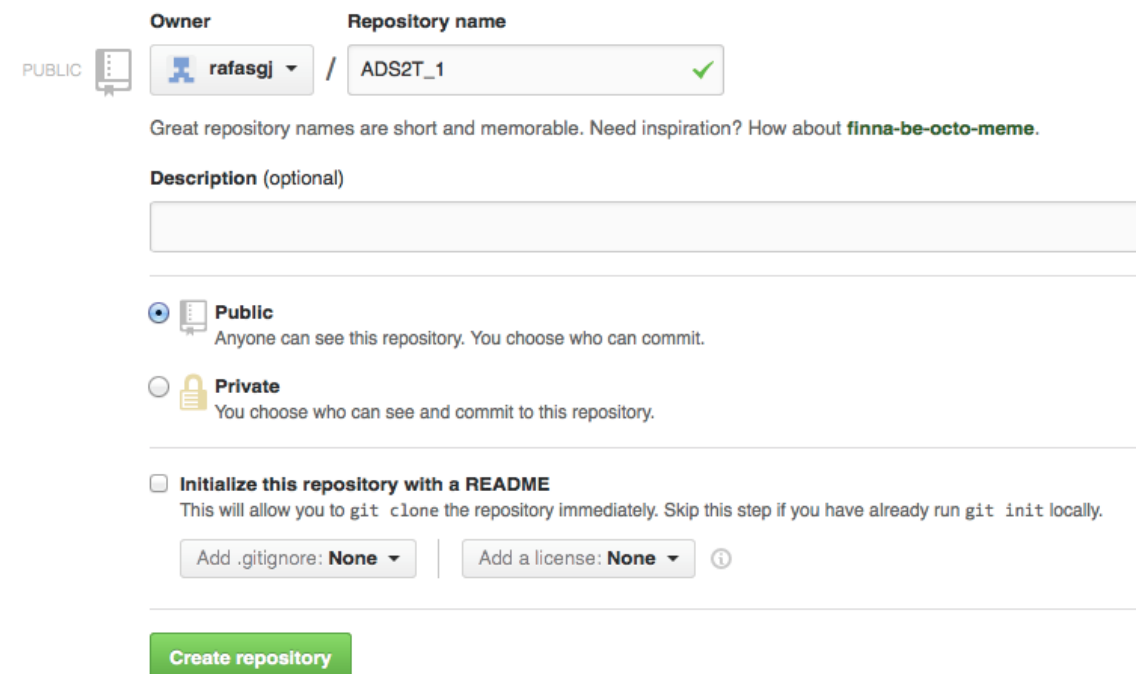
- Github é um serviço de hospedagem web para projetos de software, disponível a partir do endereço

<http://github.com>

- O serviço utiliza o Git como sistema de controle de versão.
- Possui opções para repositórios públicos (gratuitos) e privados (pagos).
- Possui um sistema de rede social para acompanhamento dos projetos e dos desenvolvedores.

# Criando um Repositório

- Após cadastrar-se no [github.com](https://github.com), você já pode criar um repositório.
- Ao clicar no botão “New Repository”, você deve configurar o seu novo repositório.
- Você deve obrigatoriamente dar um nome ao seu repositório.
- Opcionalmente, você pode dar uma descrição.
- Repositórios privados devem ser pagos.
- Adicionar um “README” ao projeto é uma boa opção, mas não é necessário quando se cria o repositório localmente ANTES de criá-lo no Github.
- Para a disciplina atual, não utilize um gitignore



The screenshot shows the GitHub 'Create new repository' interface. At the top, there's a 'PUBLIC' button and a dropdown menu for the owner, currently set to 'rafasgj'. Next to it is a text input for the 'Repository name' with 'ADS2T\_1' entered and a green checkmark. Below this, a message says 'Great repository names are short and memorable. Need inspiration? How about finna-be-octo-meme.' There's a 'Description (optional)' text area. Under 'Visibility', the 'Public' option is selected with a radio button, and the 'Private' option is unselected. Below that, there's a checkbox for 'Initialize this repository with a README' which is unchecked. At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by a green 'Create repository' button.

# Sincronizando Repositórios

- Primeiro, verifique se a configuração do proxy está correta

```
git config --global --get http.proxy
```

- Adicione uma nova conexão remota ao seu repositório local, apontando para o repositório remoto

```
git remote add origin <repositório>
```

- Verifique a conexão remota

```
git remote show origin
```

- Verifique o status do repositório local

```
git status -s
```

- Envie as alterações para o repositório remoto

```
git push origin master
```

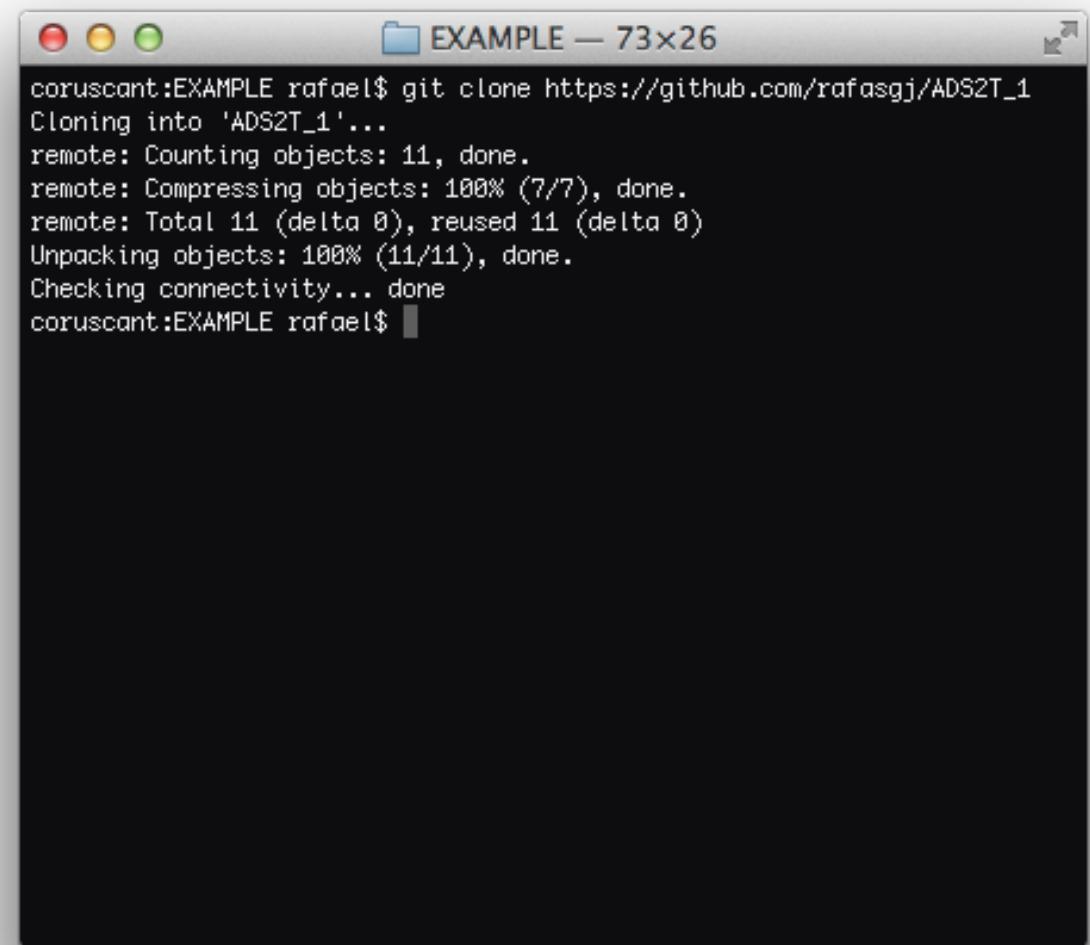


```
coruscant:ADS2T_1 rafael$ git config --global --get http.proxy
coruscant:ADS2T_1 rafael$ git remote add origin https://github.com/rafasgj/ADS2T_1
coruscant:ADS2T_1 rafael$ git remote show origin
* remote origin
  Fetch URL: https://github.com/rafasgj/ADS2T_1
  Push URL: https://github.com/rafasgj/ADS2T_1
  HEAD branch: (unknown)
coruscant:ADS2T_1 rafael$ git status -s
coruscant:ADS2T_1 rafael$ git push origin master
Counting objects: 11, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (11/11), 1.29 KiB | 0 bytes/s, done.
Total 11 (delta 0), reused 0 (delta 0)
To https://github.com/rafasgj/ADS2T_1
 * [new branch]      master -> master
coruscant:ADS2T_1 rafael$
```

# Baixando um Repositório

- De vez em quando, a fonte original do repositório não é a máquina local, mas sim o repositório remoto.
- Nestes casos, não é necessária a criação local do repositório, basta que seja efetuada a cópia do repositório remoto.
- Para fazer a cópia do repositório, utilize o comando 'git clone'

git clone <repositório>

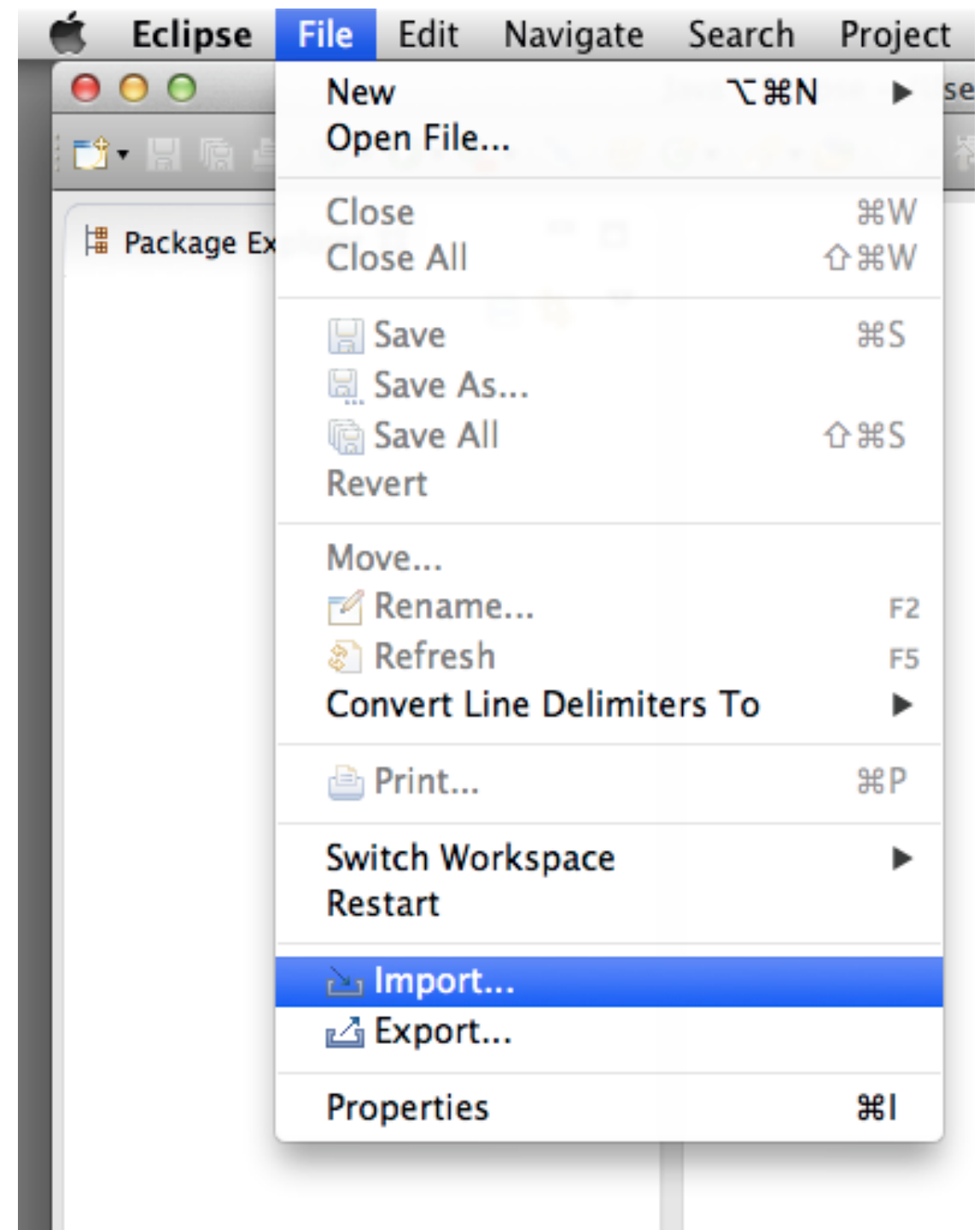
A screenshot of a terminal window titled 'EXAMPLE — 73x26'. The window shows the execution of the 'git clone' command to clone a repository from GitHub. The output text is as follows:

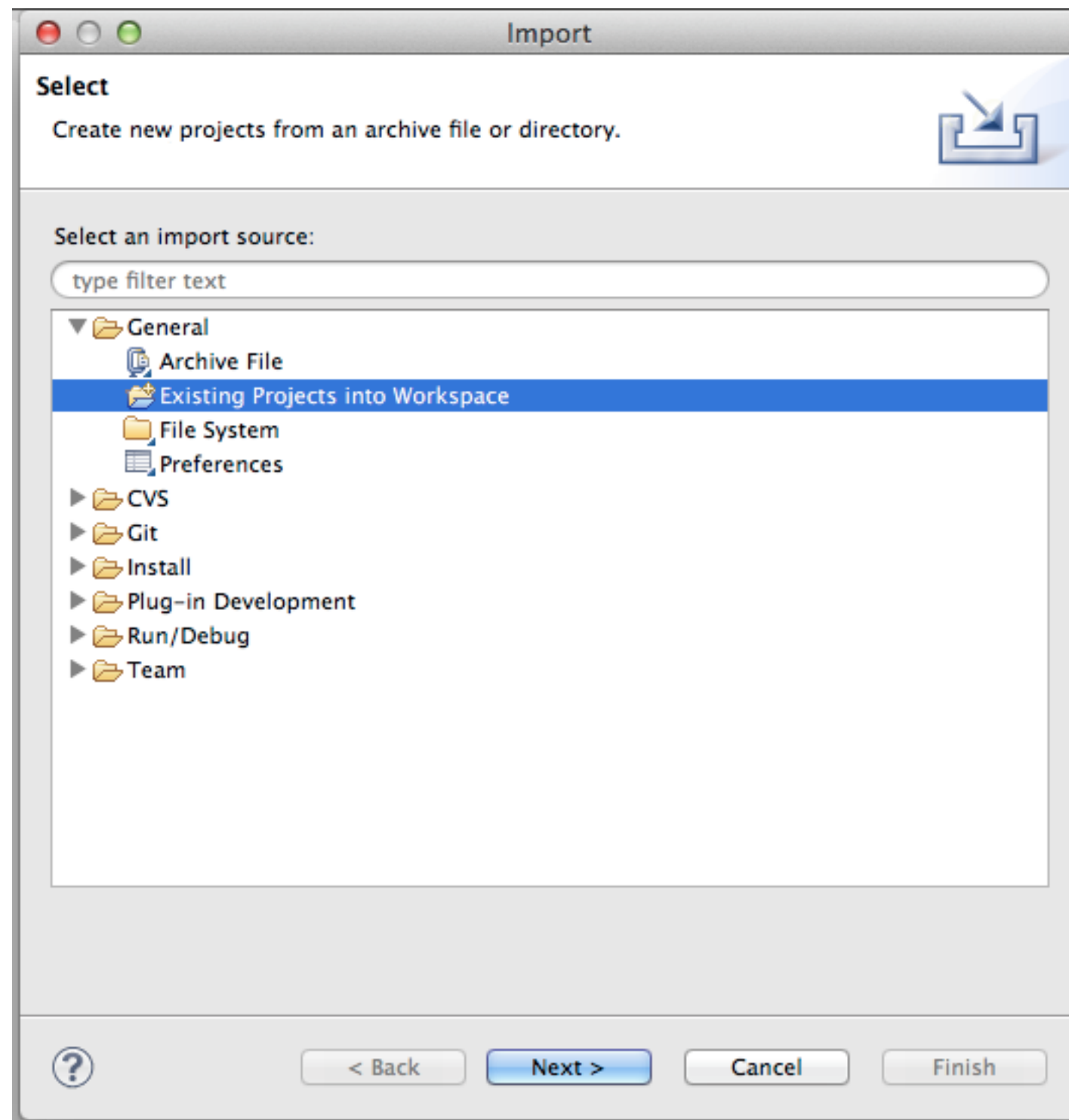
```
coruscant:EXAMPLE rafael$ git clone https://github.com/rafasgj/ADS2T_1
Cloning into 'ADS2T_1'...
remote: Counting objects: 11, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 11 (delta 0), reused 11 (delta 0)
Unpacking objects: 100% (11/11), done.
Checking connectivity... done
coruscant:EXAMPLE rafael$
```

# Importando um Projeto no Workspace do Eclipse

- Quando você baixa um repositório, mesmo que este contenha os arquivos de um projeto, este projeto não será automaticamente configurado no Eclipse.
- Para isso, você deve importar o projeto no eclipse antes de trabalhar.

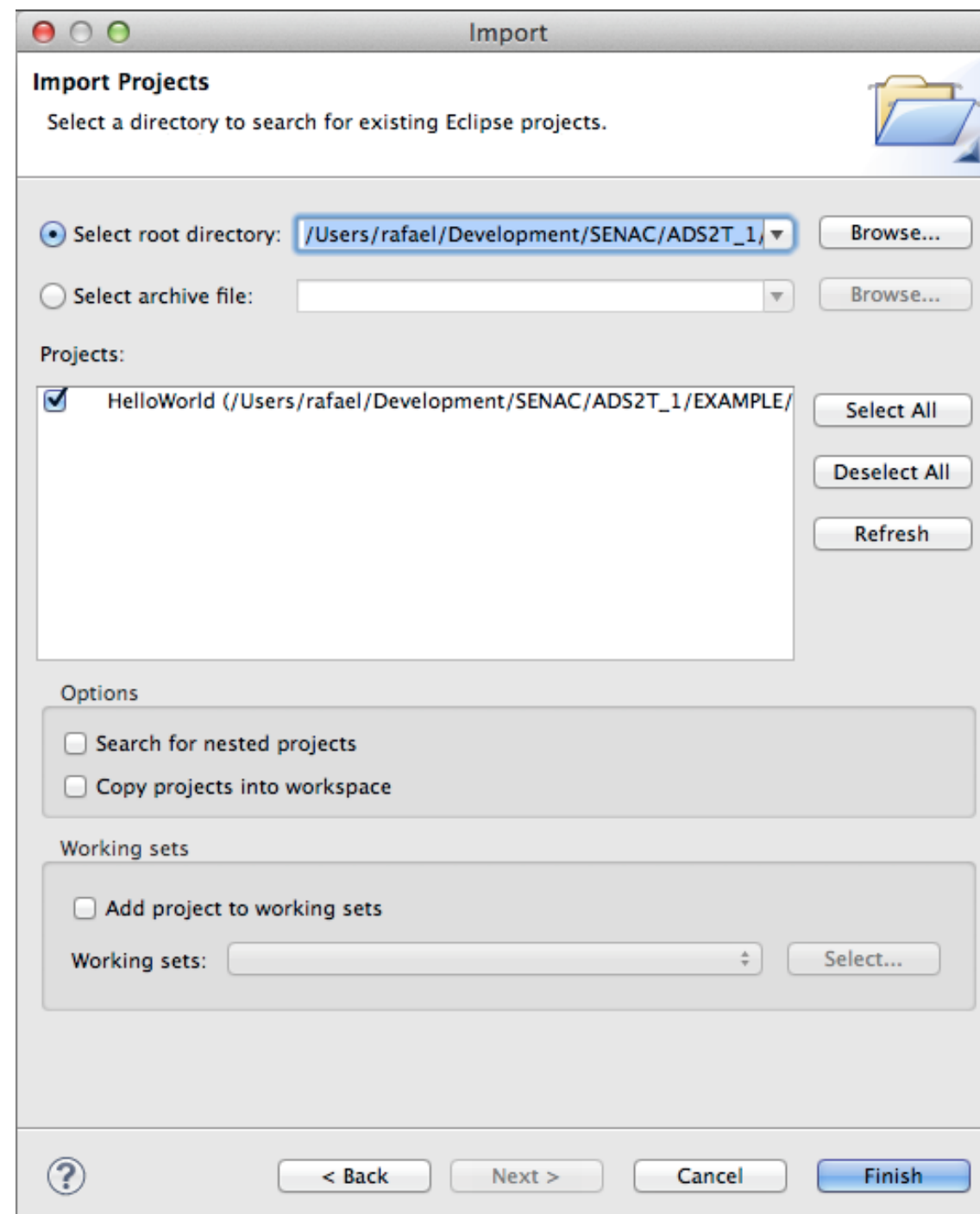
Selecione “Import”



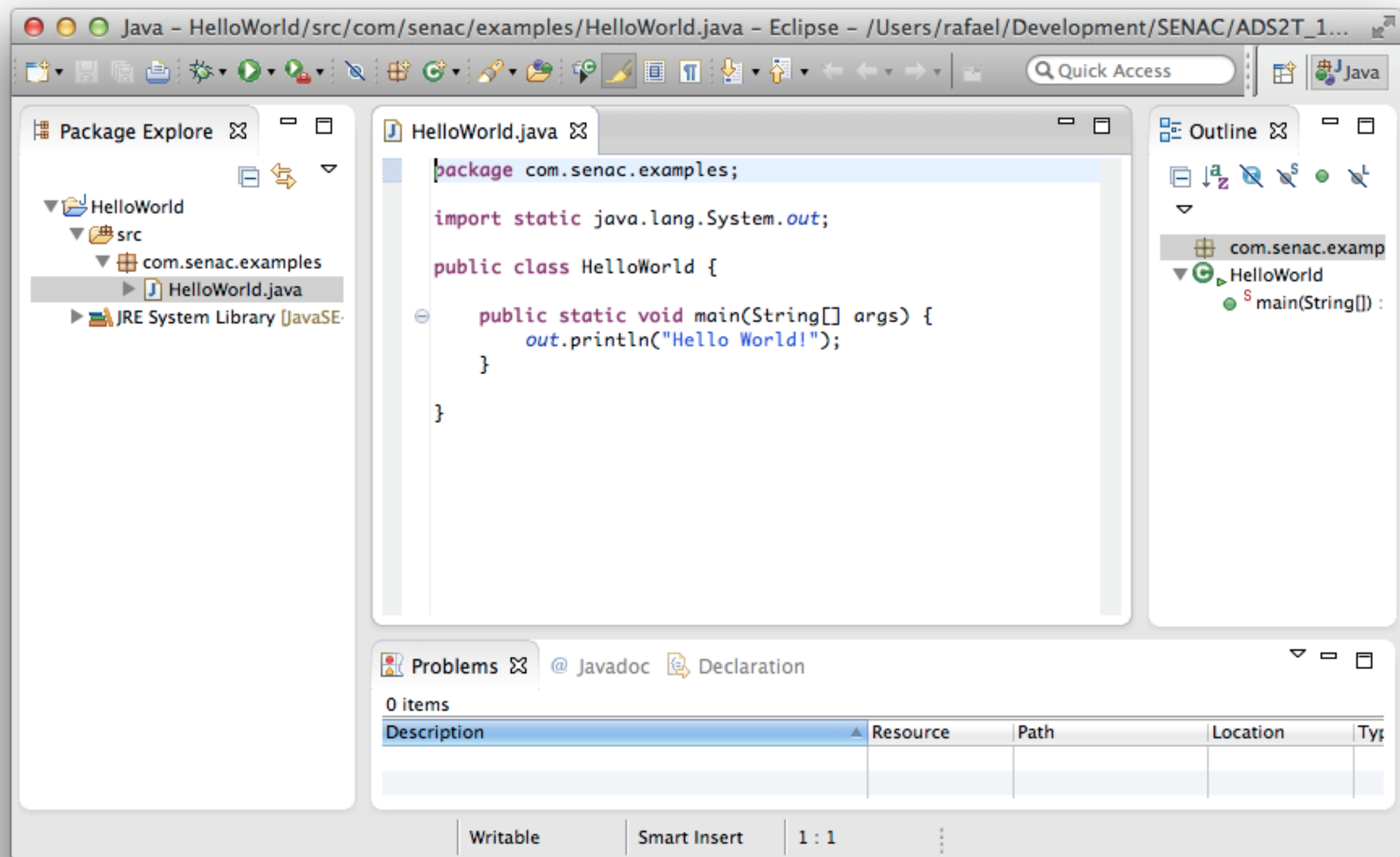


Selecione importar projetos existentes para o *Workspace*.





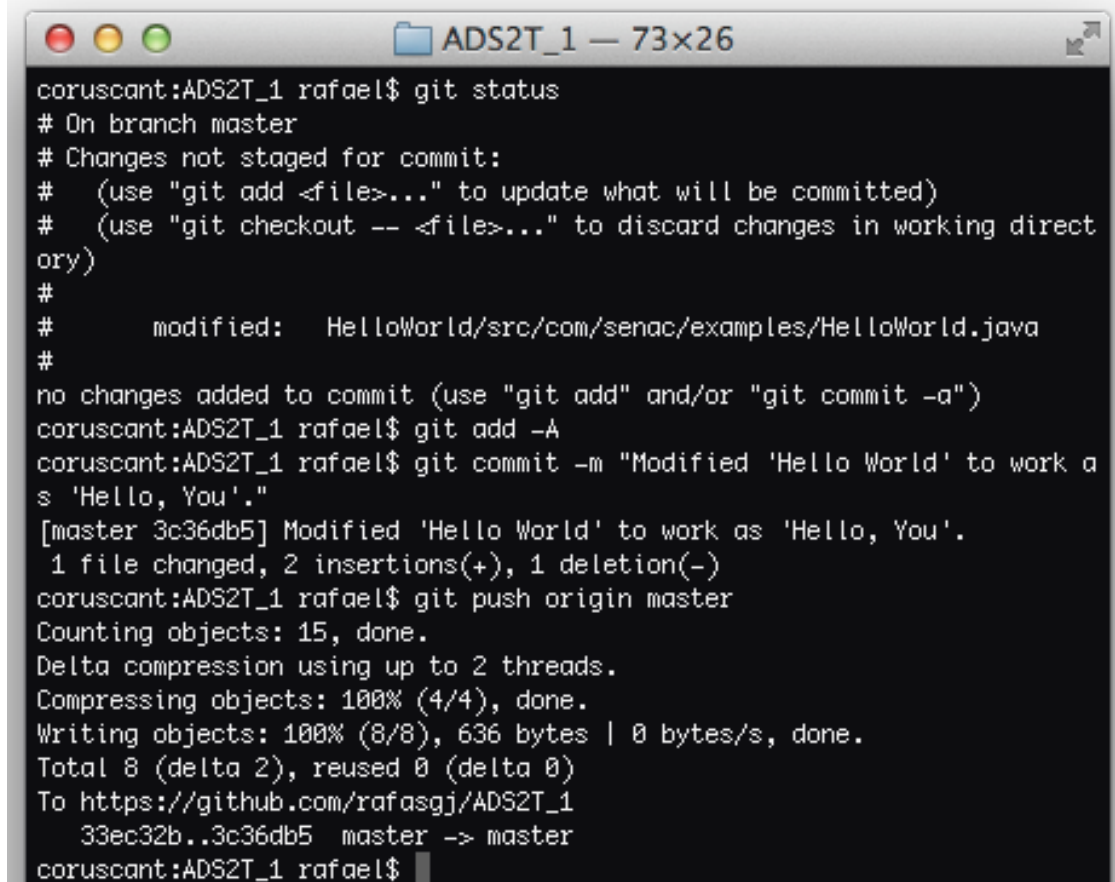
Selecionando o diretório do repositório, os projetos disponíveis serão exibidos.



Ao final, seu projeto estará disponível para trabalhar.

# Gravando as alterações de um projeto.

- Uma vez que você alterou o código fonte de um projeto que já foi armazenado no repositório, estas alterações também devem ser armazenadas no repositório.
- Para isso, é necessário utilizar os comandos 'add' e 'commit' novamente.
- Lembre-se, 'add' não é para adicionar “arquivos”, mas sim, “modificações”.
- Utilize “git push” para sincronizar novamente com o repositório remoto.



```
coruscant:ADS2T_1 rafael$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   HelloWorld/src/com/senac/examples/HelloWorld.java
#
no changes added to commit (use "git add" and/or "git commit -a")
coruscant:ADS2T_1 rafael$ git add -A
coruscant:ADS2T_1 rafael$ git commit -m "Modified 'Hello World' to work as 'Hello, You'."
[master 3c36db5] Modified 'Hello World' to work as 'Hello, You'.
 1 file changed, 2 insertions(+), 1 deletion(-)
coruscant:ADS2T_1 rafael$ git push origin master
Counting objects: 15, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 636 bytes | 0 bytes/s, done.
Total 8 (delta 2), reused 0 (delta 0)
To https://github.com/rafasgj/ADS2T_1
 33ec32b..3c36db5  master -> master
coruscant:ADS2T_1 rafael$
```

# Link Úteis

- <http://gitref.org/basic/>
- <http://git-scm.com/book/pt-br/Primeiros-passos-Noções-Básicas-de-Git>
- [http://rogerdudler.github.io/git-guide/index.pt\\_BR.html](http://rogerdudler.github.io/git-guide/index.pt_BR.html)
- <http://codexico.com.br/blog/linux/tutorial-simples-como-usar-o-git-e-o-github/>
- <http://br.kernelnewbies.org/node/58>
- <http://try.github.io/levels/1/challenges/1>