Android Studio and GitHub

Vídeo de suporte PT: https://www.youtube.com/watch?v=j8KDLvi5MDs
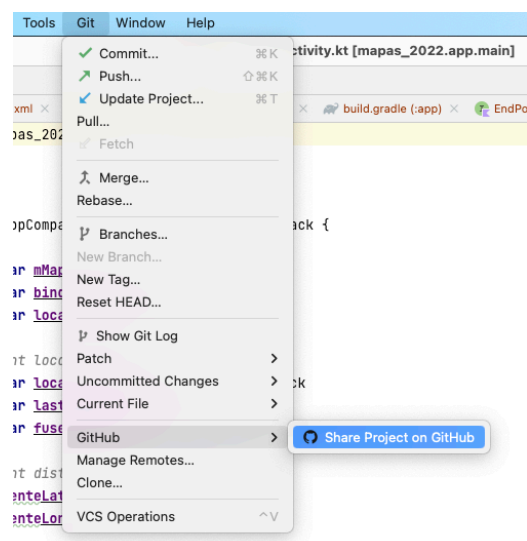
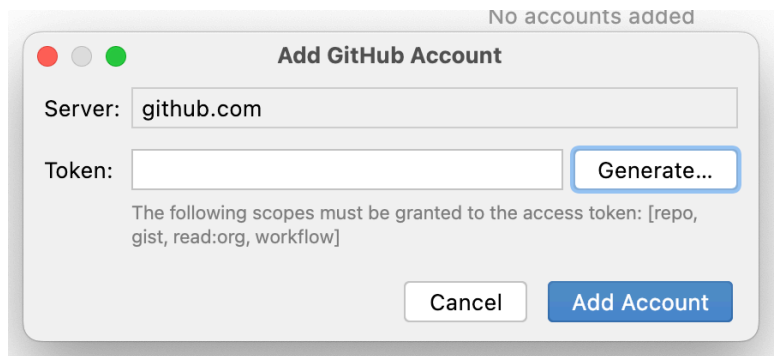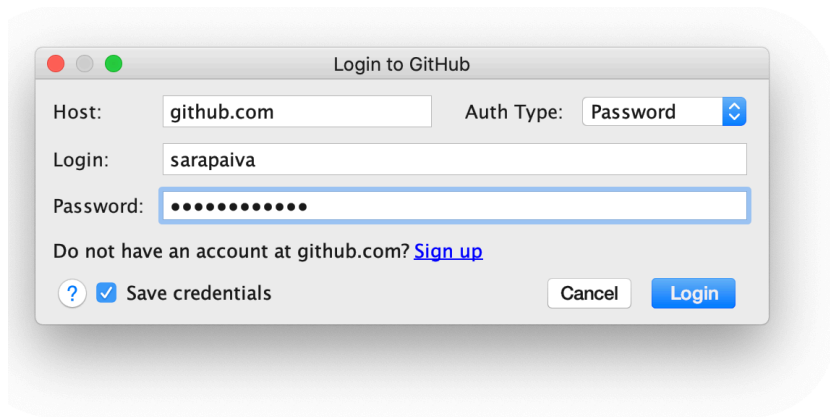Start by going to VCS and choose "Enable Version Control Integration"
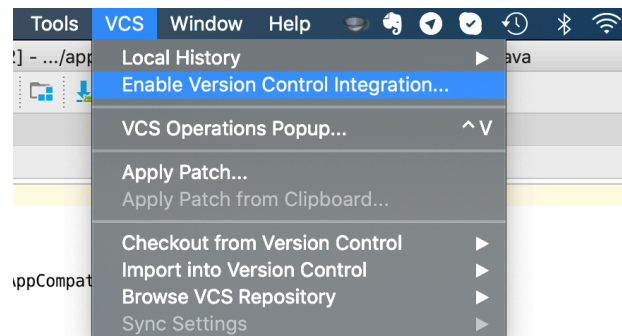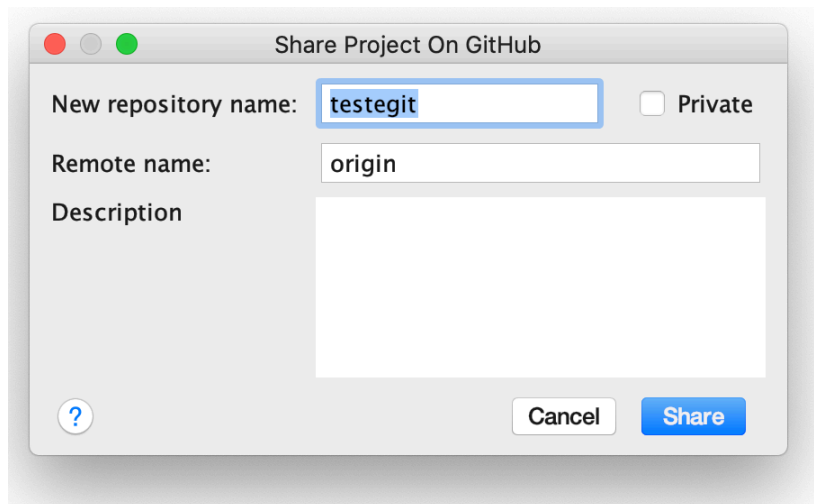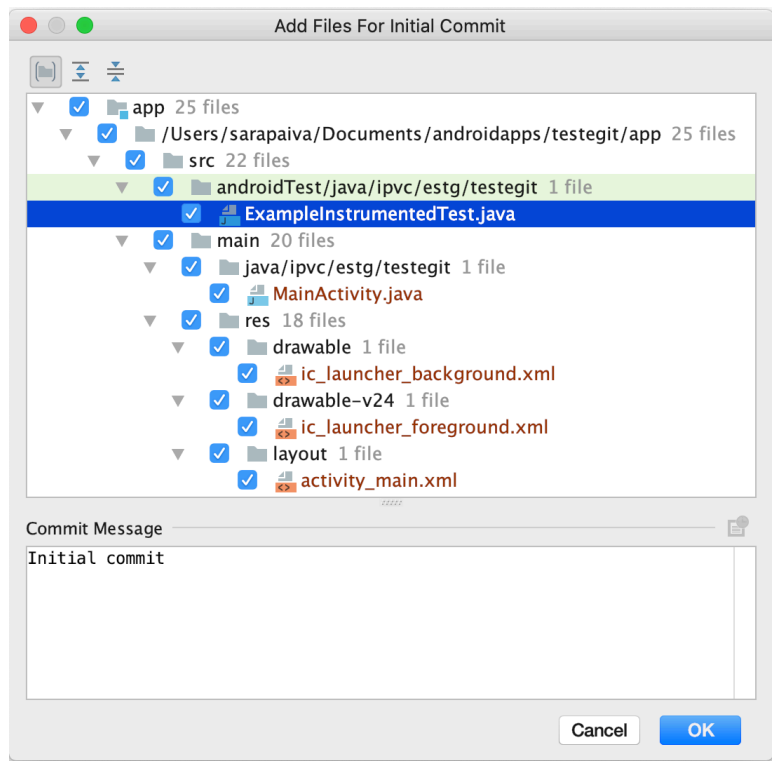

Choose Git
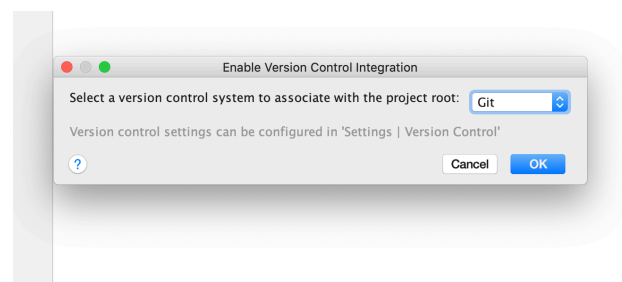
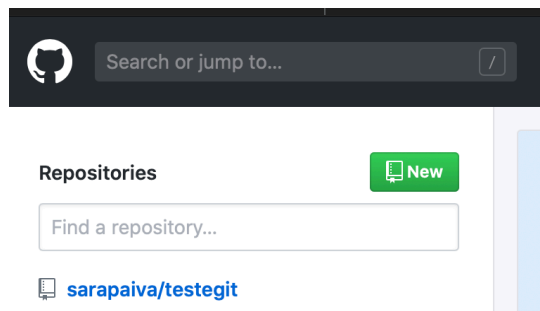Now, share Project on GitHub

Enter your GitHub credentials or Token.

Login to GitHub

Host: github.com   Auth Type: Password
Login: sarapaiva
Password: ••••••••••••

Do not have an account at github.com? Sign up

✓ Save credentials   Cancel   Login

No accounts added

Add GitHub Account

Server: github.com

Token: Generate...

The following scopes must be granted to the access token: [repo, gist, read:org, workflow]

Cancel   Add Account

Provide the name of the repository to be created in github remotely

Share Project On GitHub

New repository name: testegit   ☐ Private

Remote name: origin

Description

?   Cancel   Share

Add initial files that will be sent to the repository and enter a commit message.

Check in the GitHub website the repository is already created:

Click it and see a recent commit has been done and all android files have been uploaded.

Change something in a file and commit

See the changes made, compare to the previous version and insert a message describing the changes made.

Commit and push so changes are sent to the remote server.

Confirm on the website:

Create 3 or more commits, making some changes to the file and entering a commit message each time.

Create new branch through android studio. Go to Menu VCS=>Git
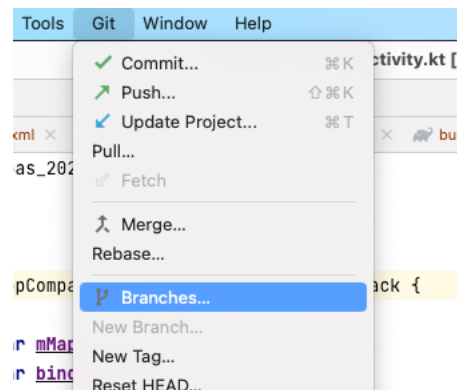
Choose New Branch
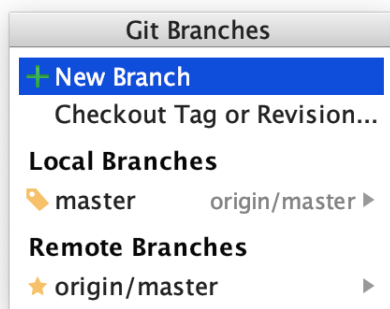
Give the branch a name
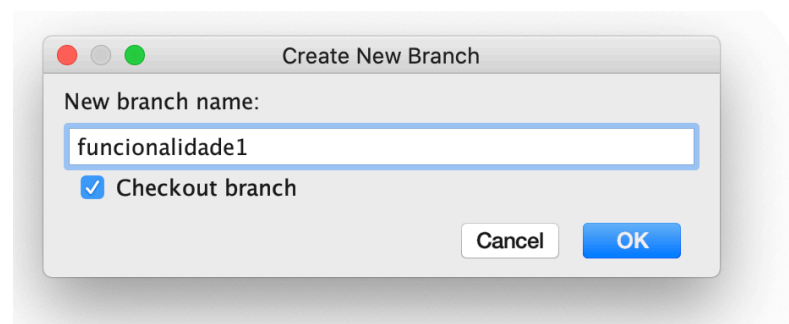
On the bottom right side you can see the branch you are in:

18:1   LF⇕   UTF-8⇕   Git: funcionalidade1 ⇕   Context: <no

Add some new line specific of this funcionalidade1

```
activity_main.xml ×    Ⓒ MainActivity.java ×

1        package ipvc.estg.testegit;
2
3        import android.support.v7.app.AppCompatActivity;
4        import android.os.Bundle;
5
6        public class MainActivity extends AppCompatActivity {
7
8            @Override
9            protected void onCreate(Bundle savedInstanceState) {
10               super.onCreate(savedInstanceState);
11               setContentView(R.layout.activity_main);
12
13               //change something
14
15               // something from branch funcionalidade1
16           }
17       }
18
```
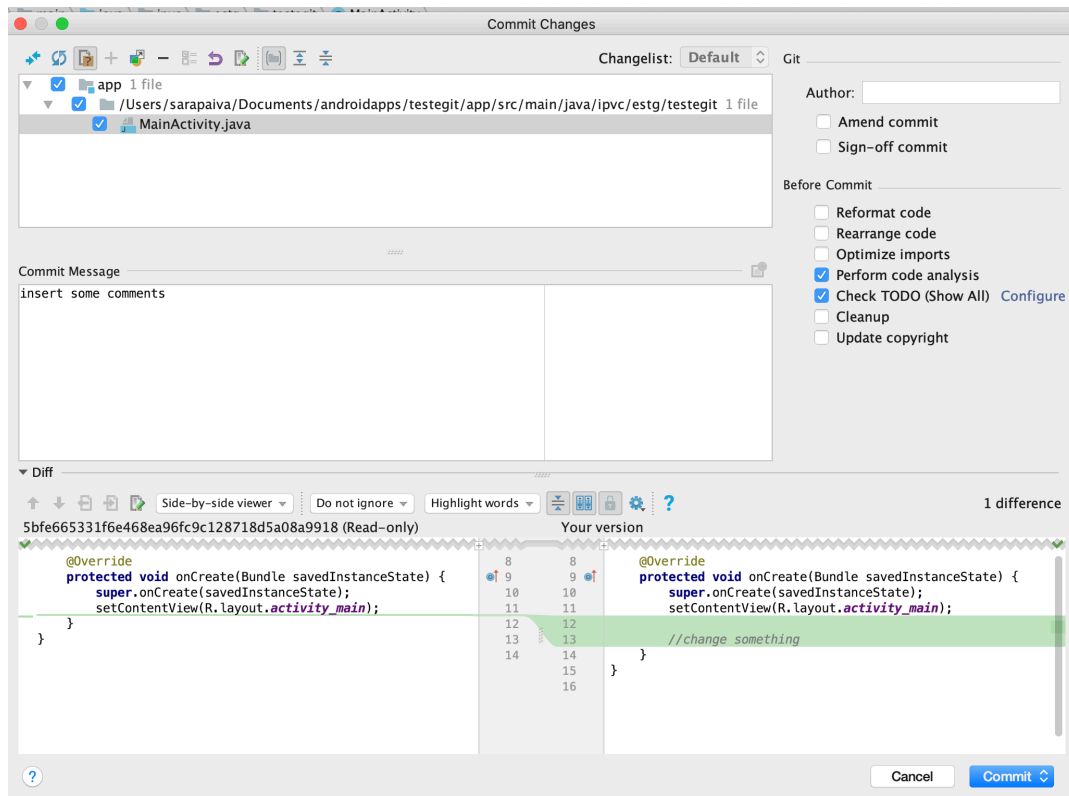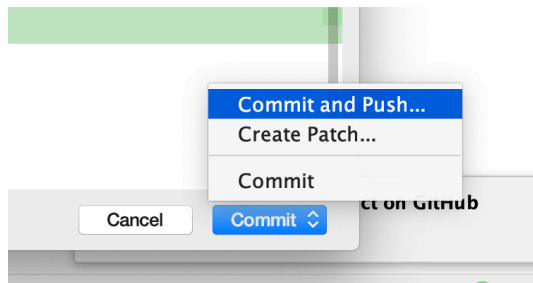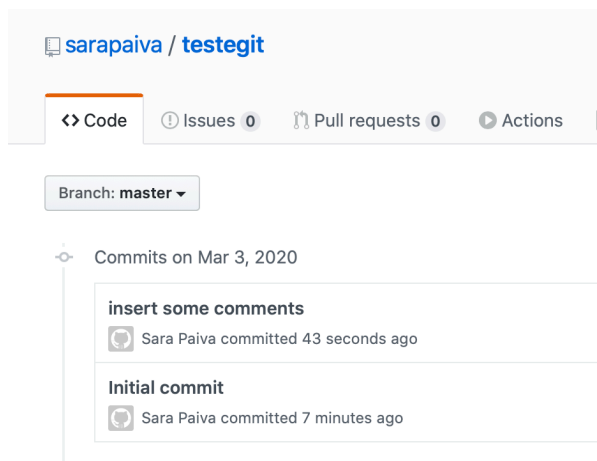
Commit.
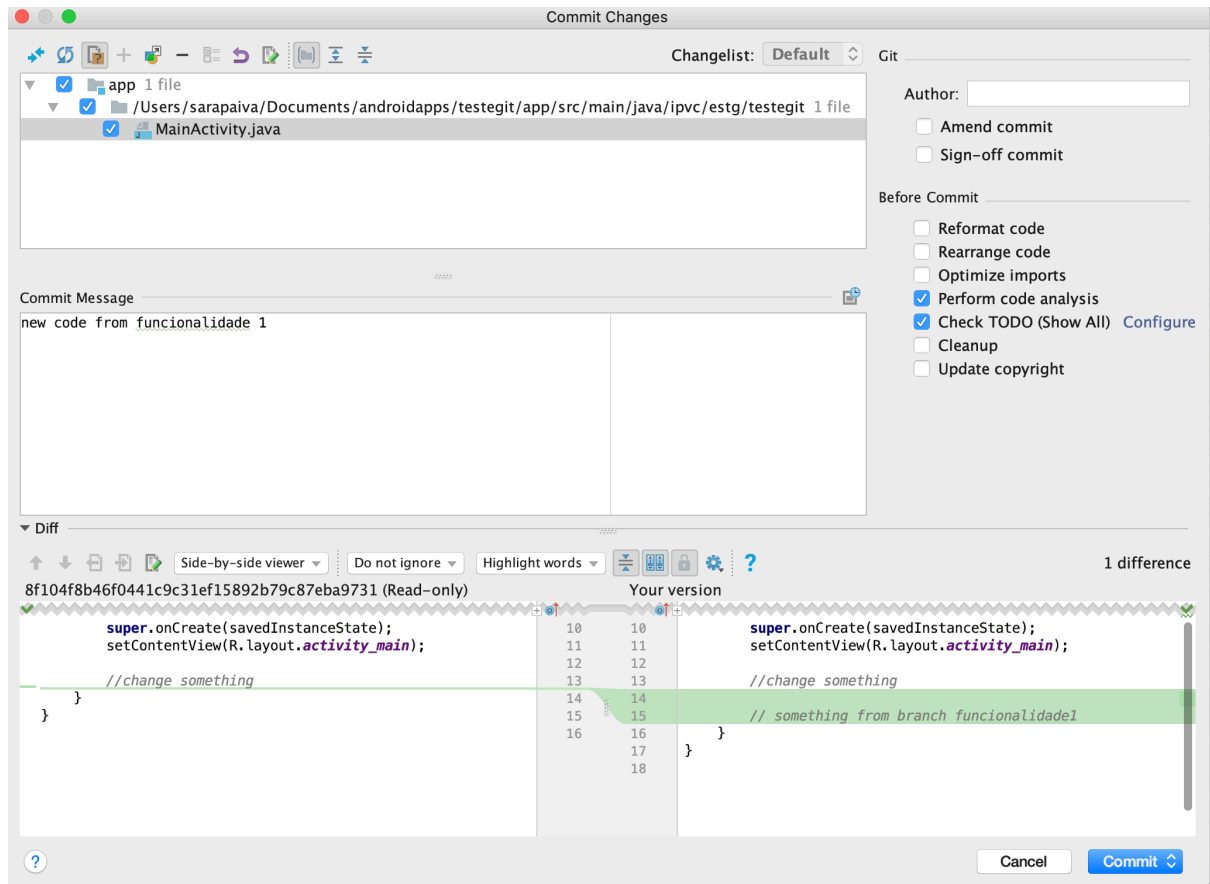
Let´s suppose we want to go to master branch so we can make a quick correction.

On the bottom, click, choose master and then checkout.

Automatically, we get the code a version that does not have what we are developing in funcionalidade1 branch.
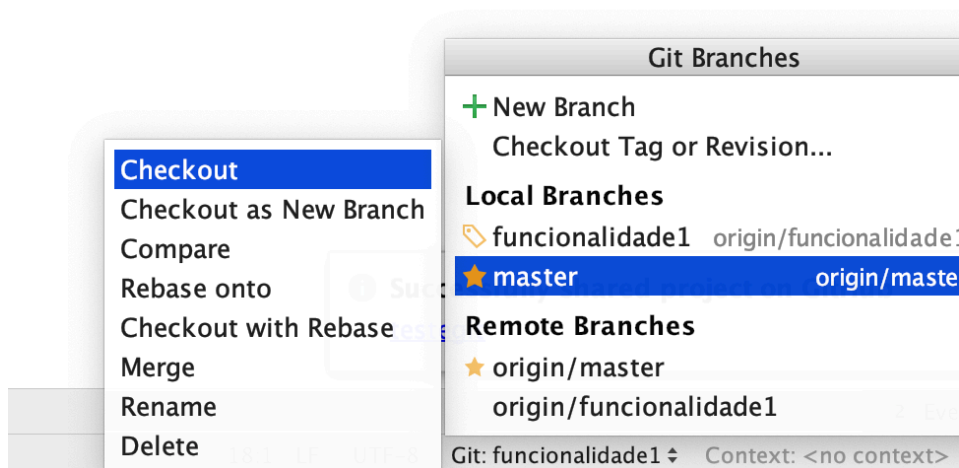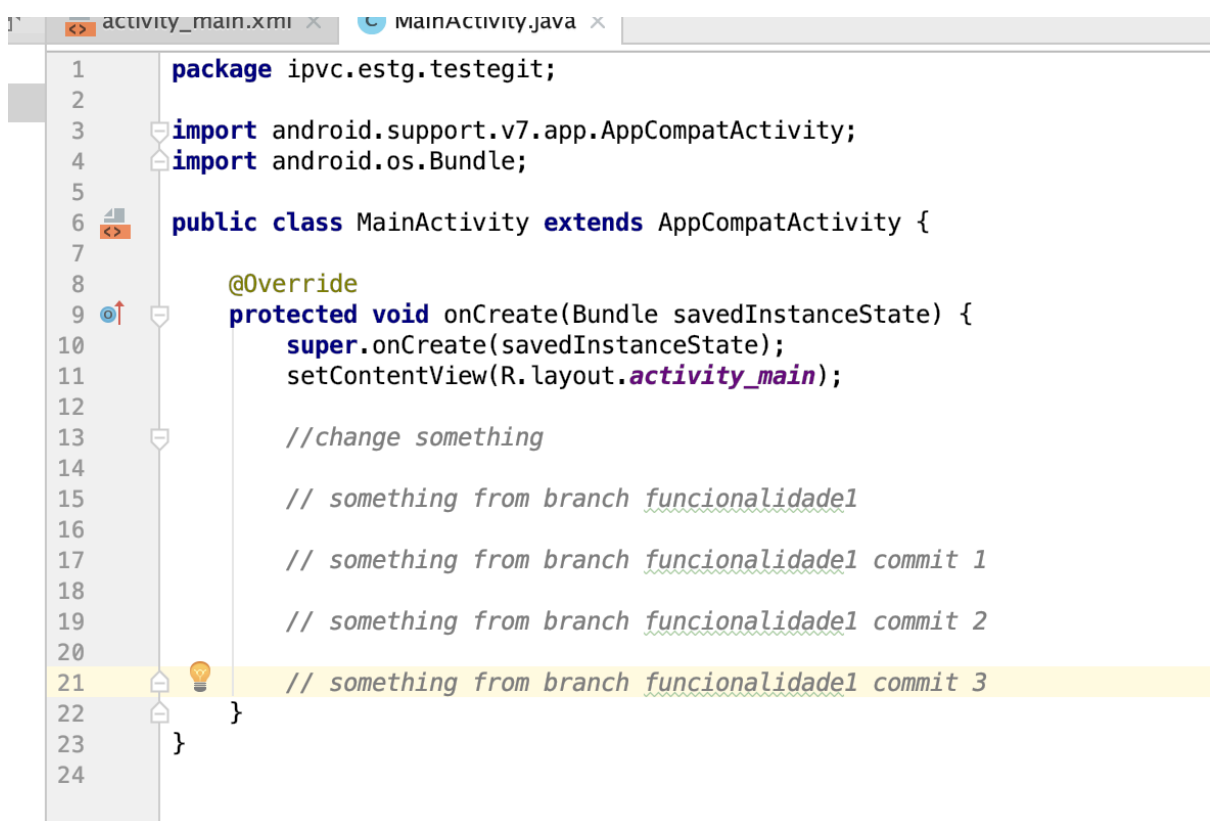
```
   activity_main.xml ×    C  MainActivity.java ×

1        package ipvc.estg.testegit;
2
3        import android.support.v7.app.AppCompatActivity;
4        import android.os.Bundle;
5
6        public class MainActivity extends AppCompatActivity {
7
8            @Override
9            protected void onCreate(Bundle savedInstanceState) {
10               super.onCreate(savedInstanceState);
11               setContentView(R.layout.activity_main);
12
13               //change something
14           }
15       }
16
```

Let´s go back to funcionalidade1 branch and make 3 new commits and push to the server.

```
   activity_main.xml ×    C  MainActivity.java ×

1        package ipvc.estg.testegit;
2
3        import android.support.v7.app.AppCompatActivity;
4        import android.os.Bundle;
5
6        public class MainActivity extends AppCompatActivity {
7
8            @Override
9            protected void onCreate(Bundle savedInstanceState) {
10               super.onCreate(savedInstanceState);
11               setContentView(R.layout.activity_main);
12
13               //change something
14
15               // something from branch funcionalidade1
16
17               // something from branch funcionalidade1 commit 1
18
19               // something from branch funcionalidade1 commit 2
20
21               // something from branch funcionalidade1 commit 3
22           }
23       }
24
```

Let´s go the web site and choose Insights and then Network (if nothing shows change your repository to be public instead of private).
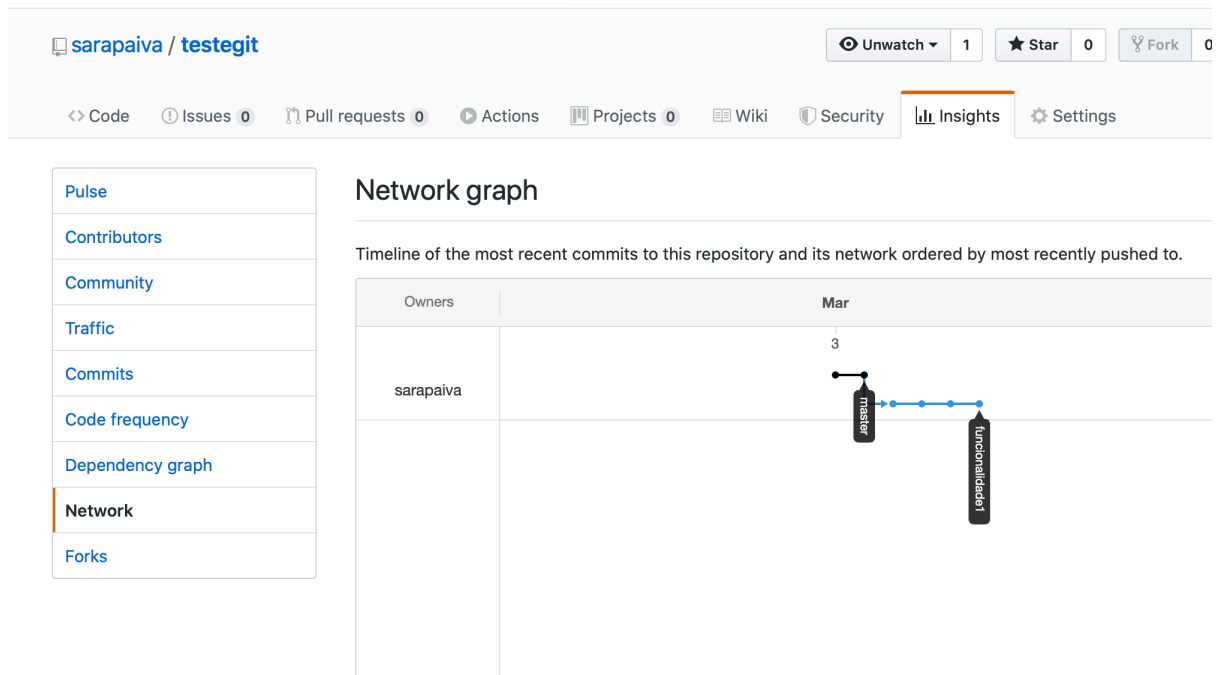
Here you can see what is happening. A master branch and another branch (funcionalidade1) with the several commits we have done.

Through GitHub in the website, access the code of the new branch (make sure you change the branch) and add some new lines to the MainActivity file, to simulate another person working on the same file.
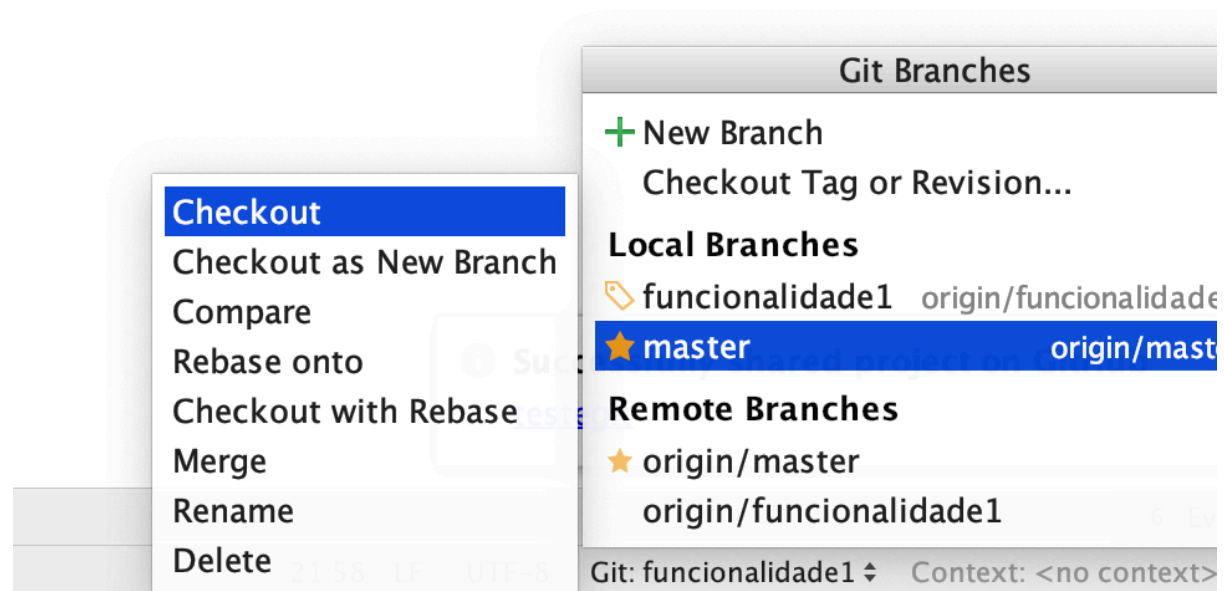
Now go to Android Studio an choose Pull (blue arrow down).
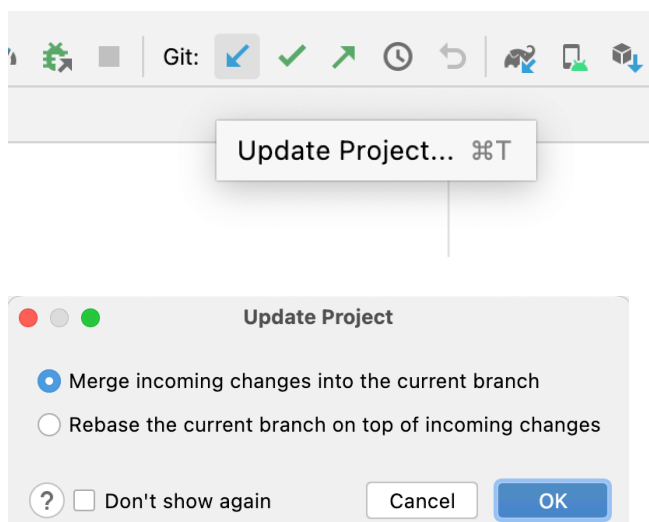
You are now working on the latest version that your colleagues sent to the repository.

Let us now assume we ended and tested everything related to funcionalidade1 and want to merge.
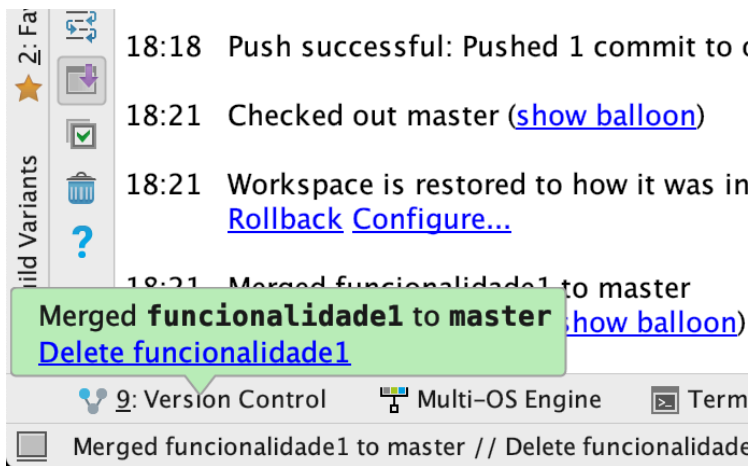
Let´s go to android studio and checkout the branch we want to merge to, in this case the master.

Again, let´s go to the bottom, click funcionalidade1 and then merge (it will merge with the current one which is the master).
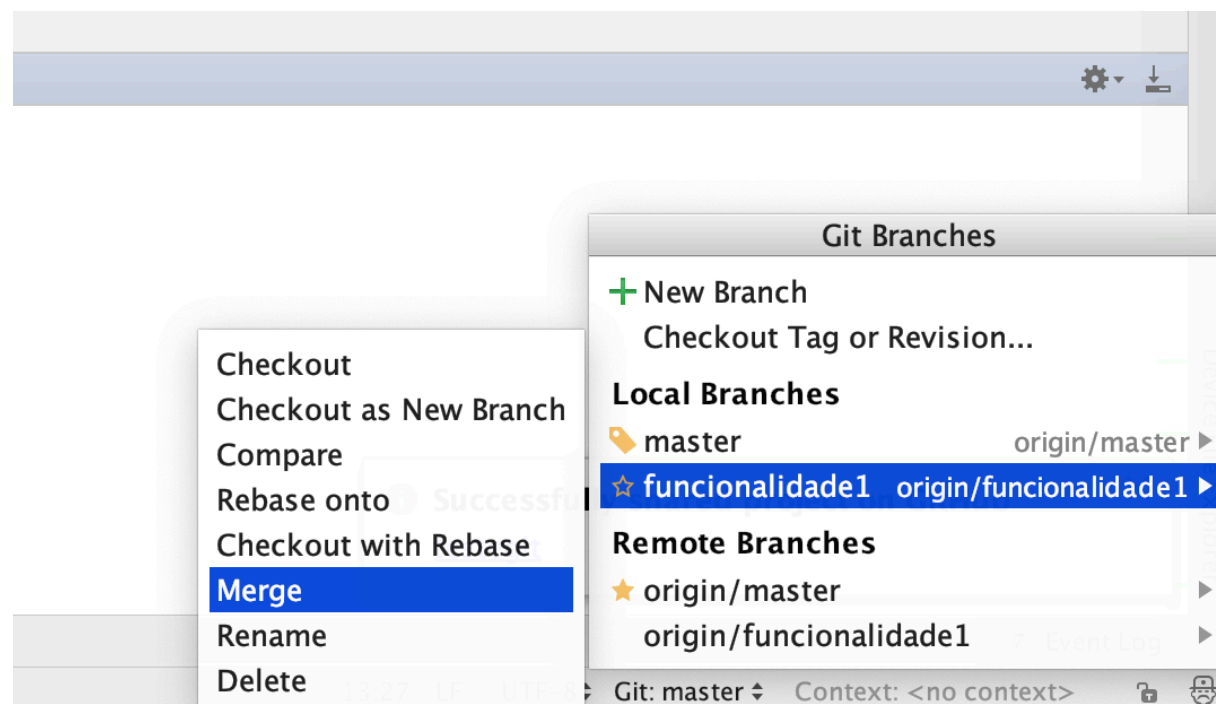
After merge, you have the option in the left side to delete the branch recently merged.

Push to remote all changes. When you merge you need to do this.

And check the website network graph again where you can see the branch is merged with the master: