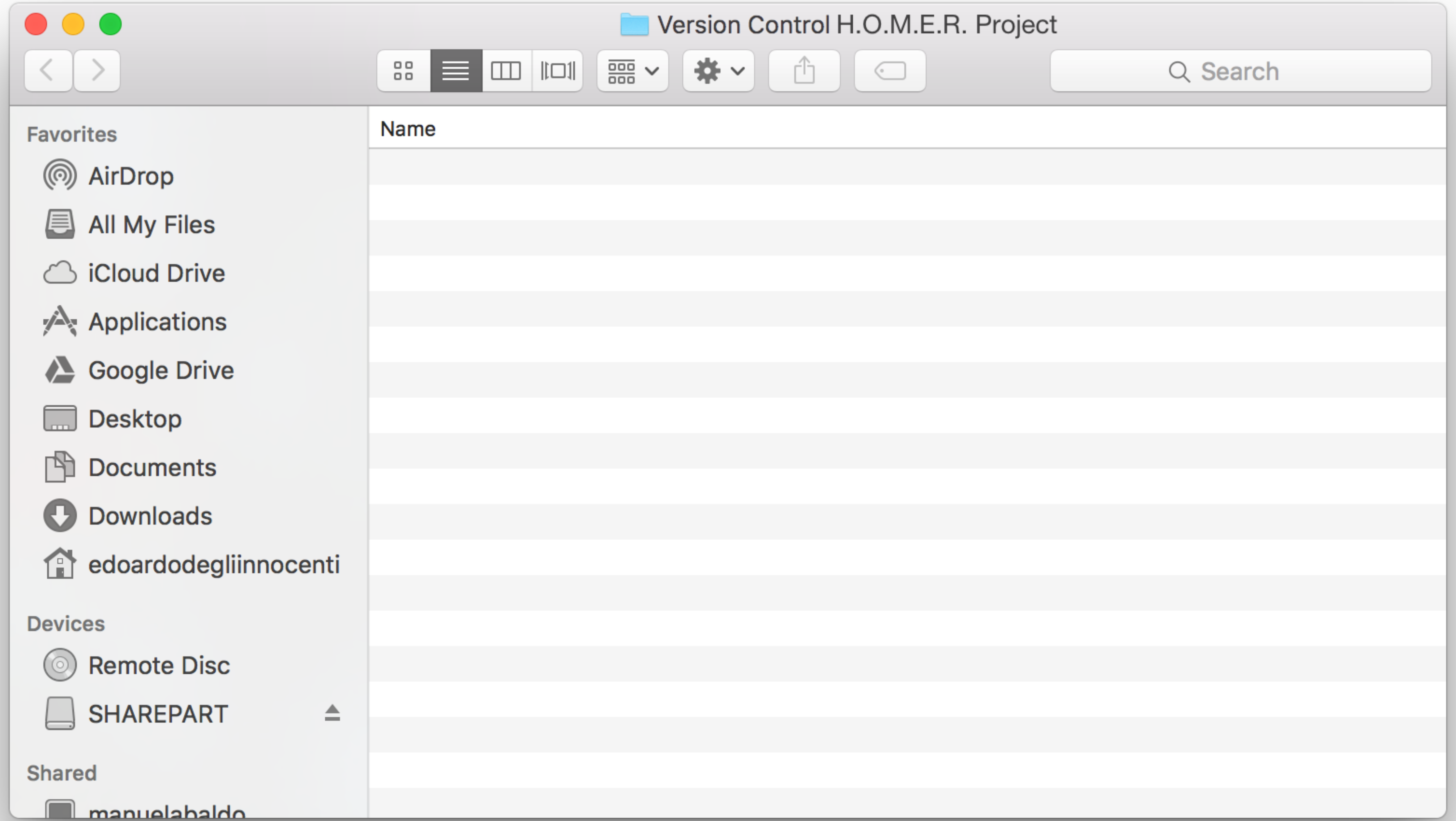
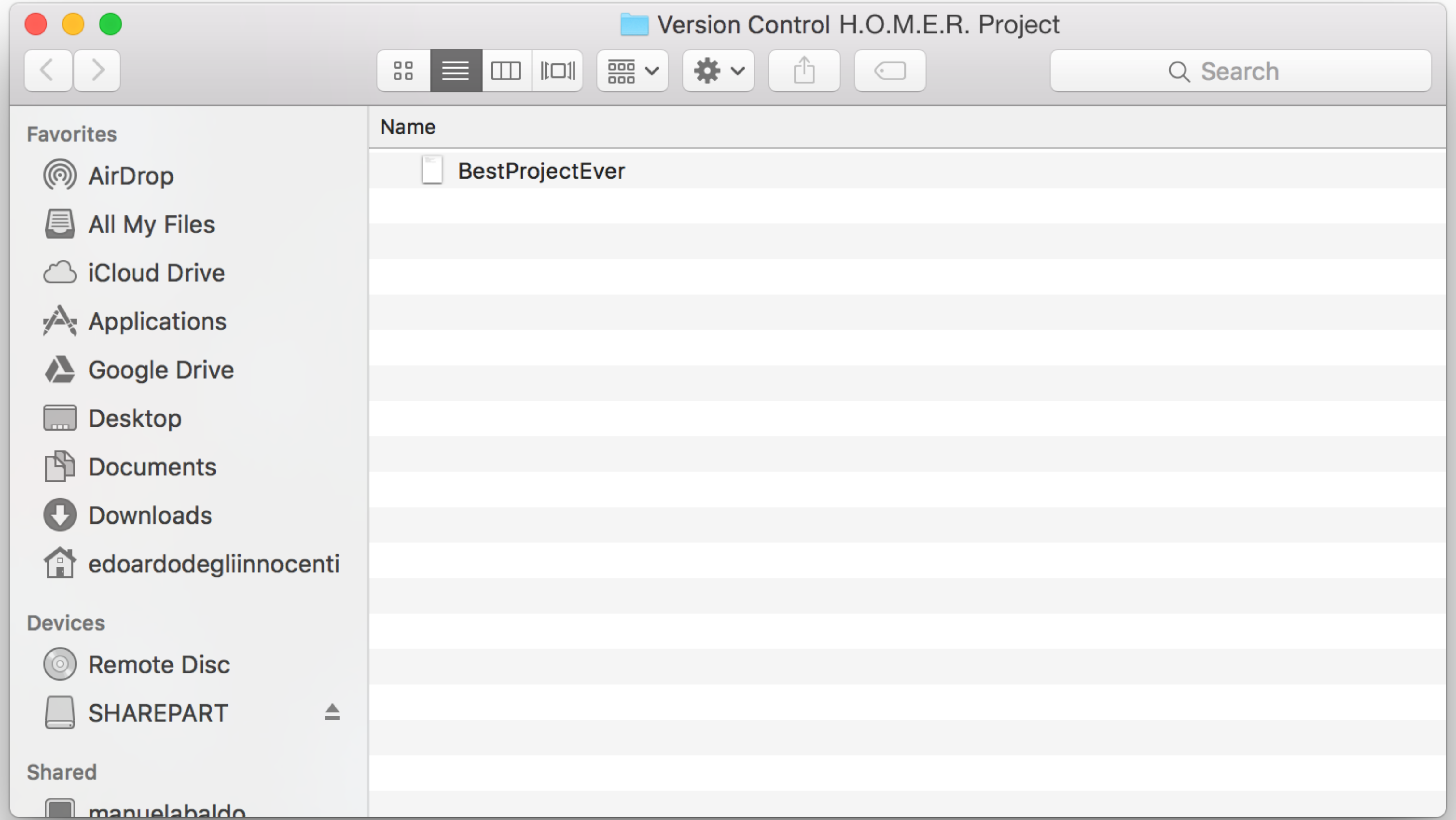
A BASIC UNDERSTANDING OF

VERSION CONTROL

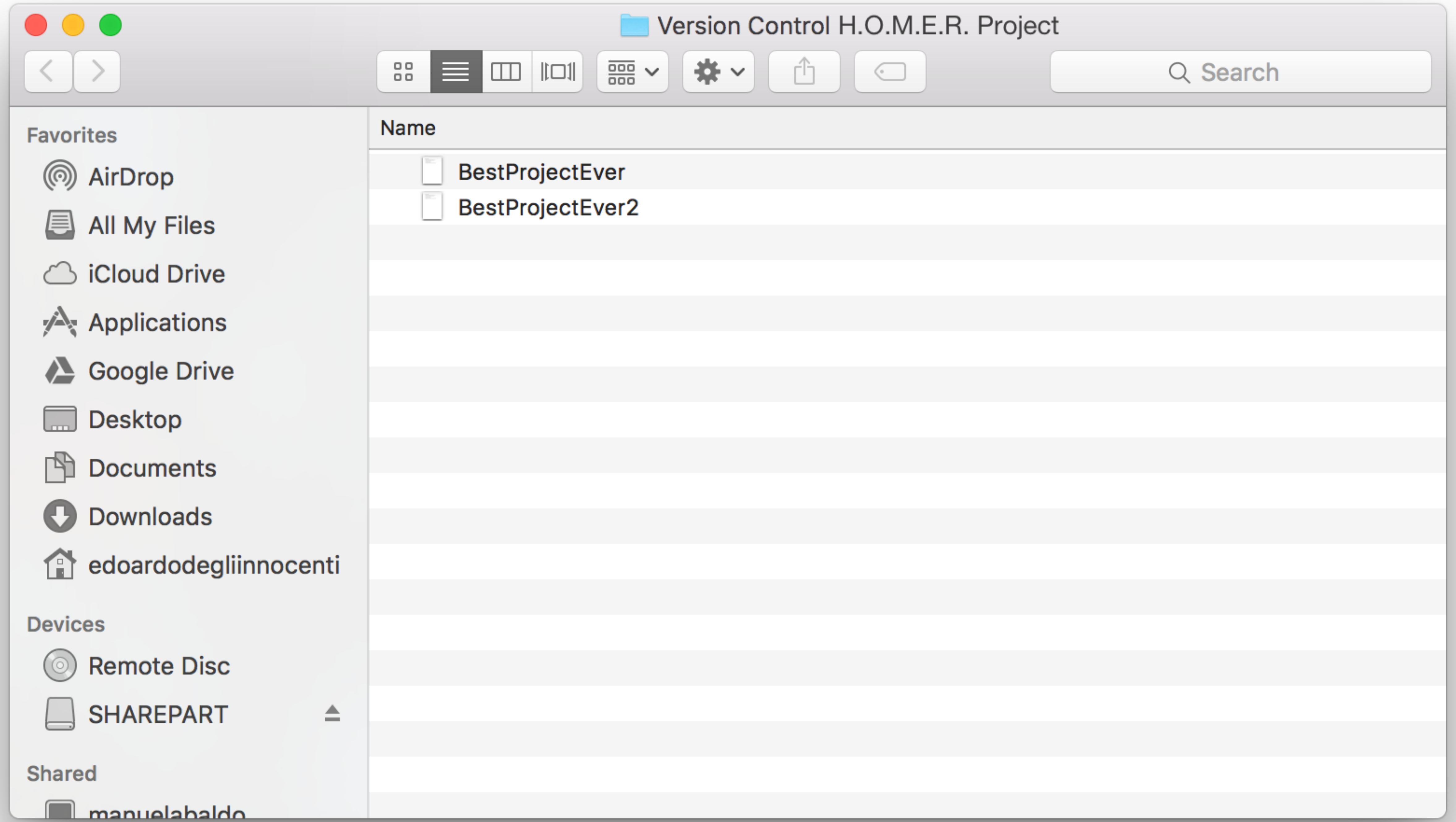
DID YOU EVER DO THIS?



DID YOU EVER DO THIS?

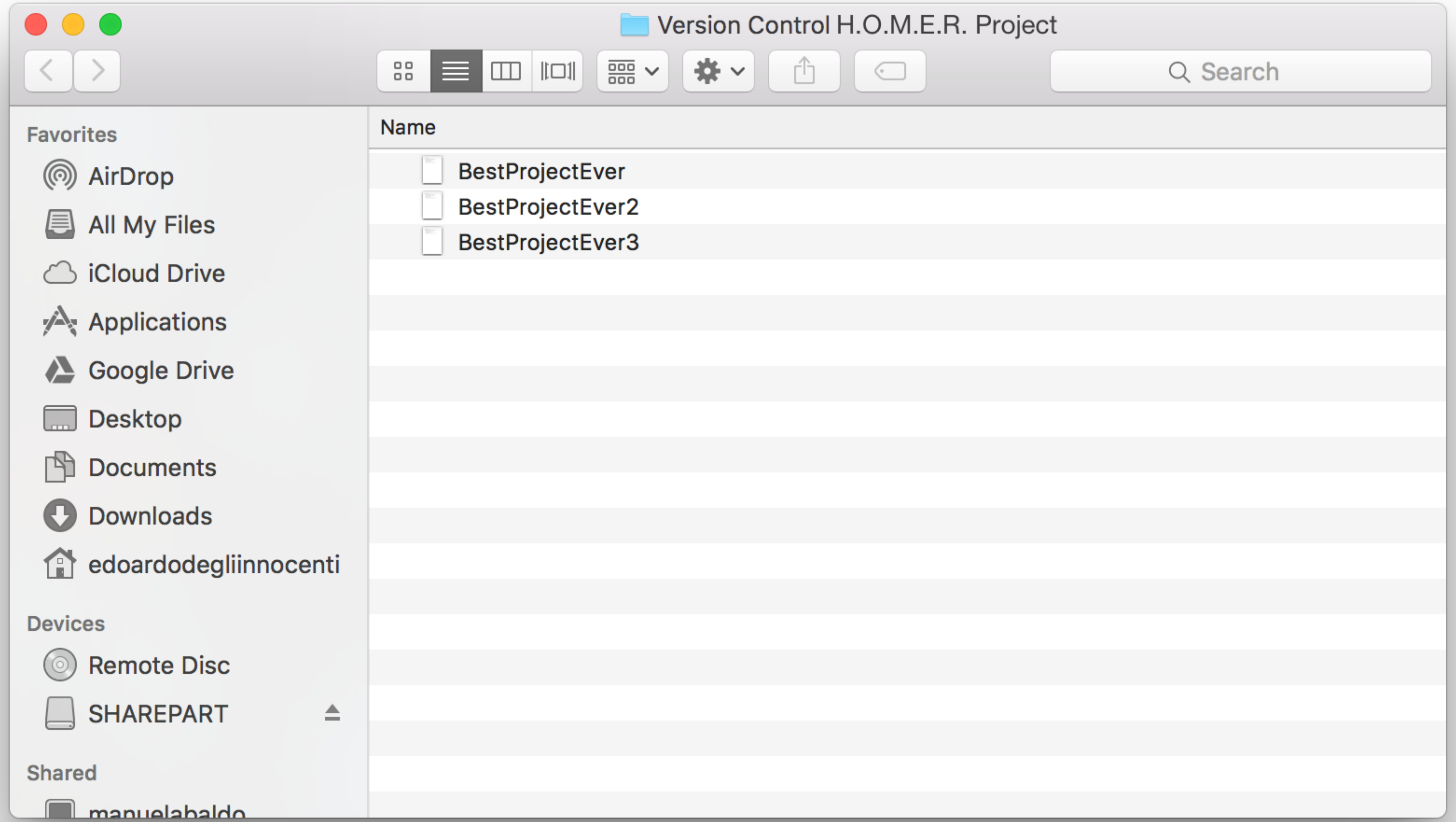


DID YOU EVER DO THIS?



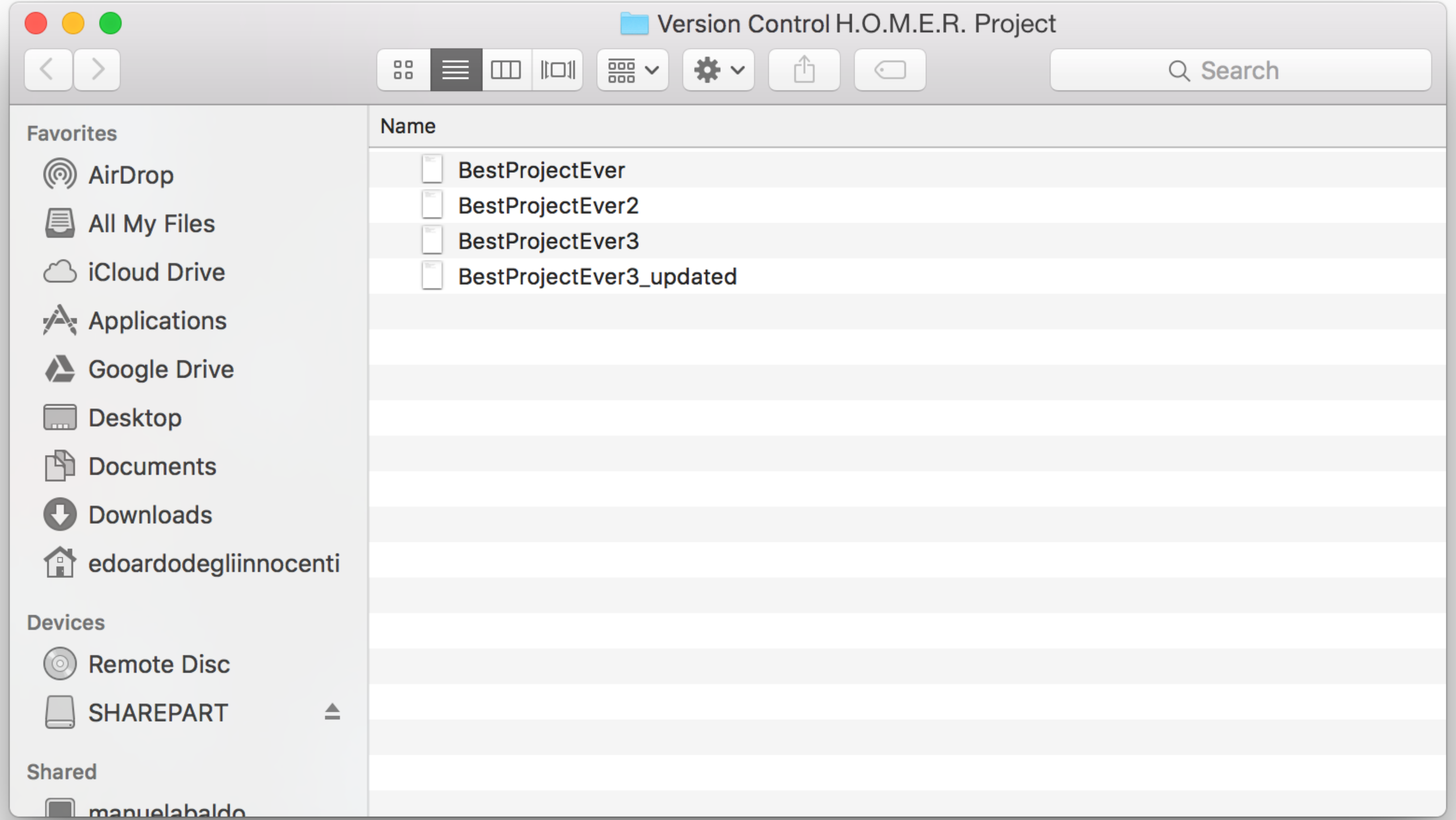
BestProjectEver
BestProjectEver2

DID YOU EVER DO THIS?



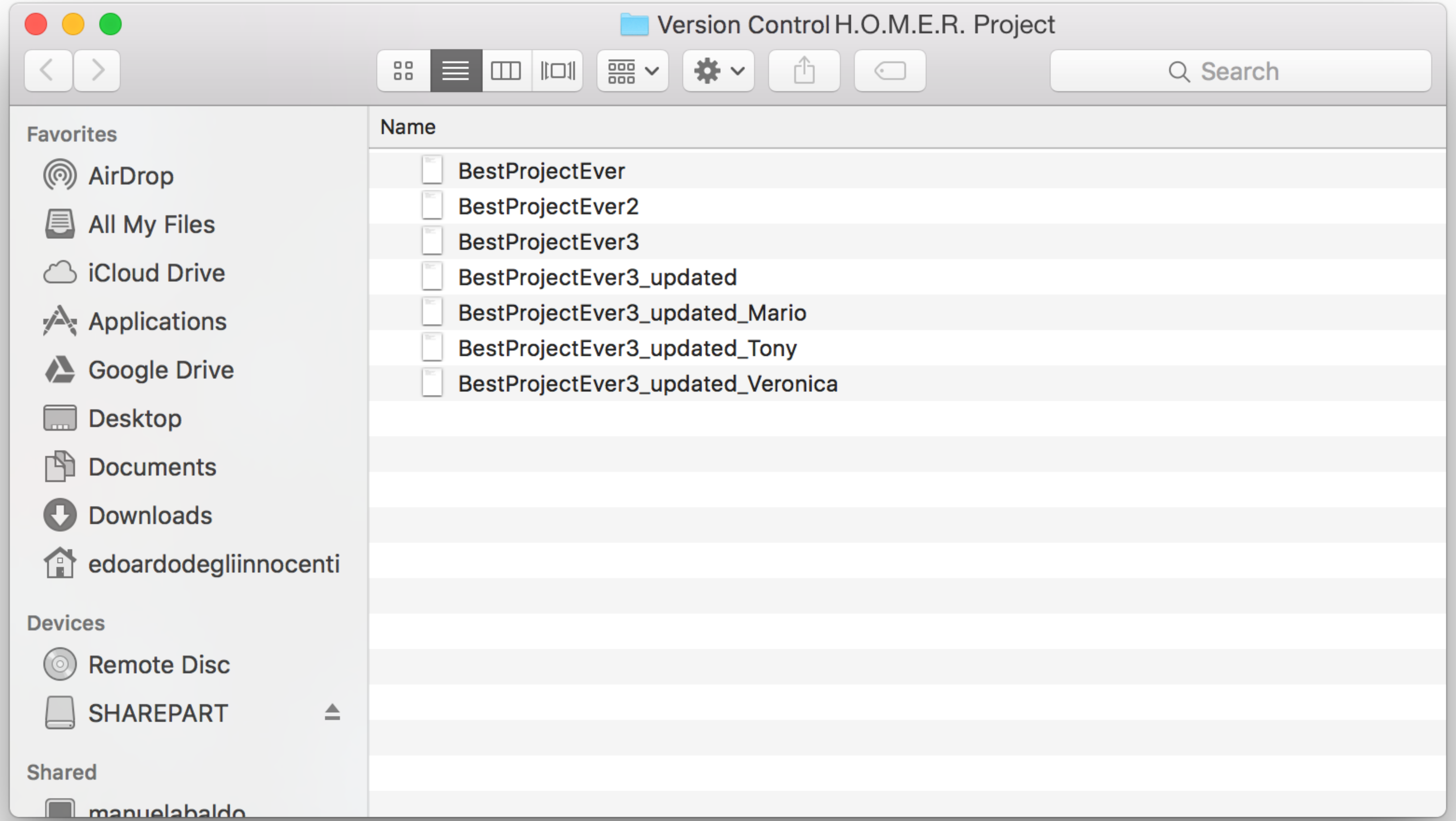
Name
BestProjectEver
BestProjectEver2
BestProjectEver3

DID YOU EVER DO THIS?



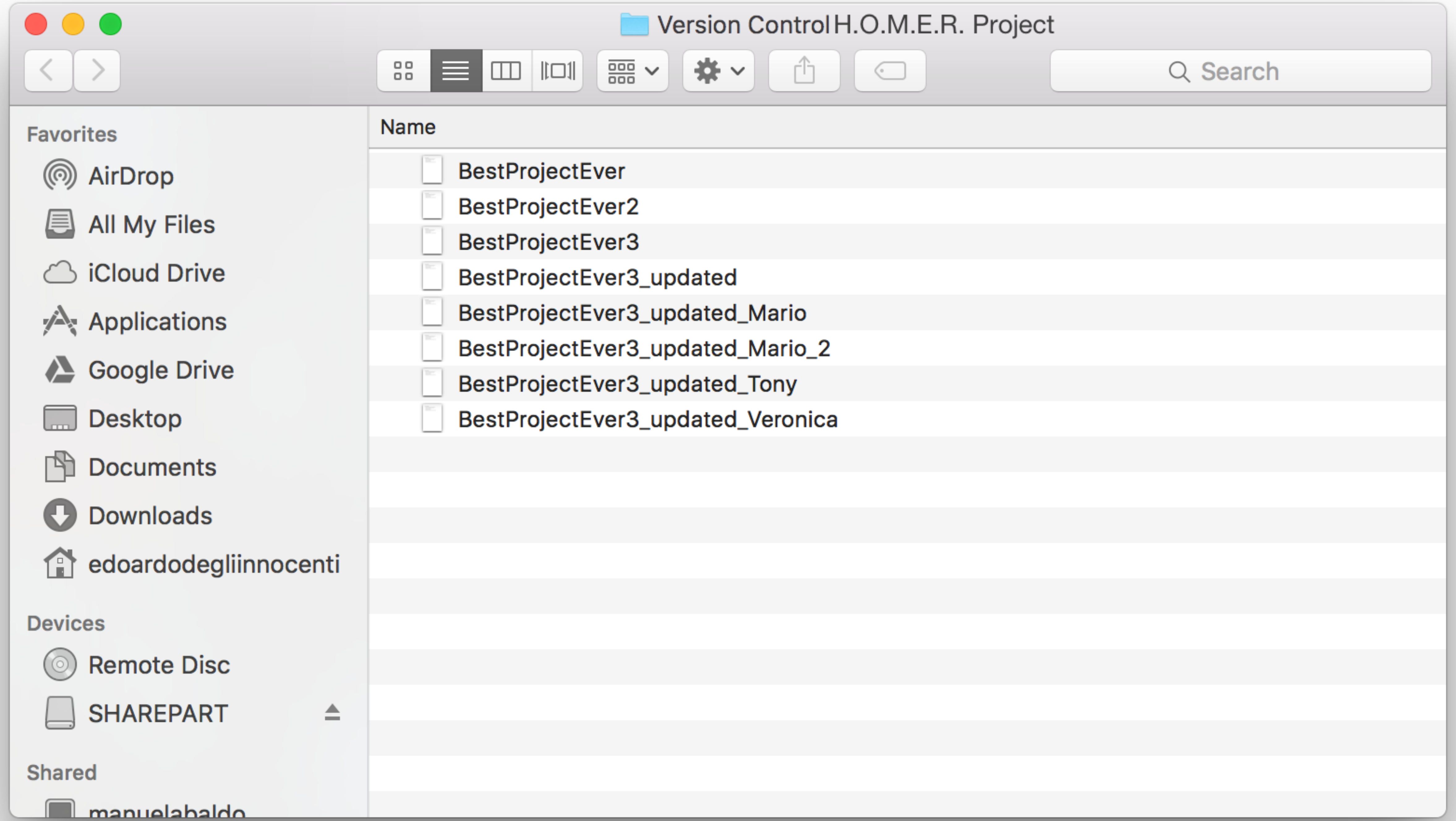
BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated

DID YOU EVER DO THIS?



Name
BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated
BestProjectEver3_updated_Mario
BestProjectEver3_updated_Tony
BestProjectEver3_updated_Veronica

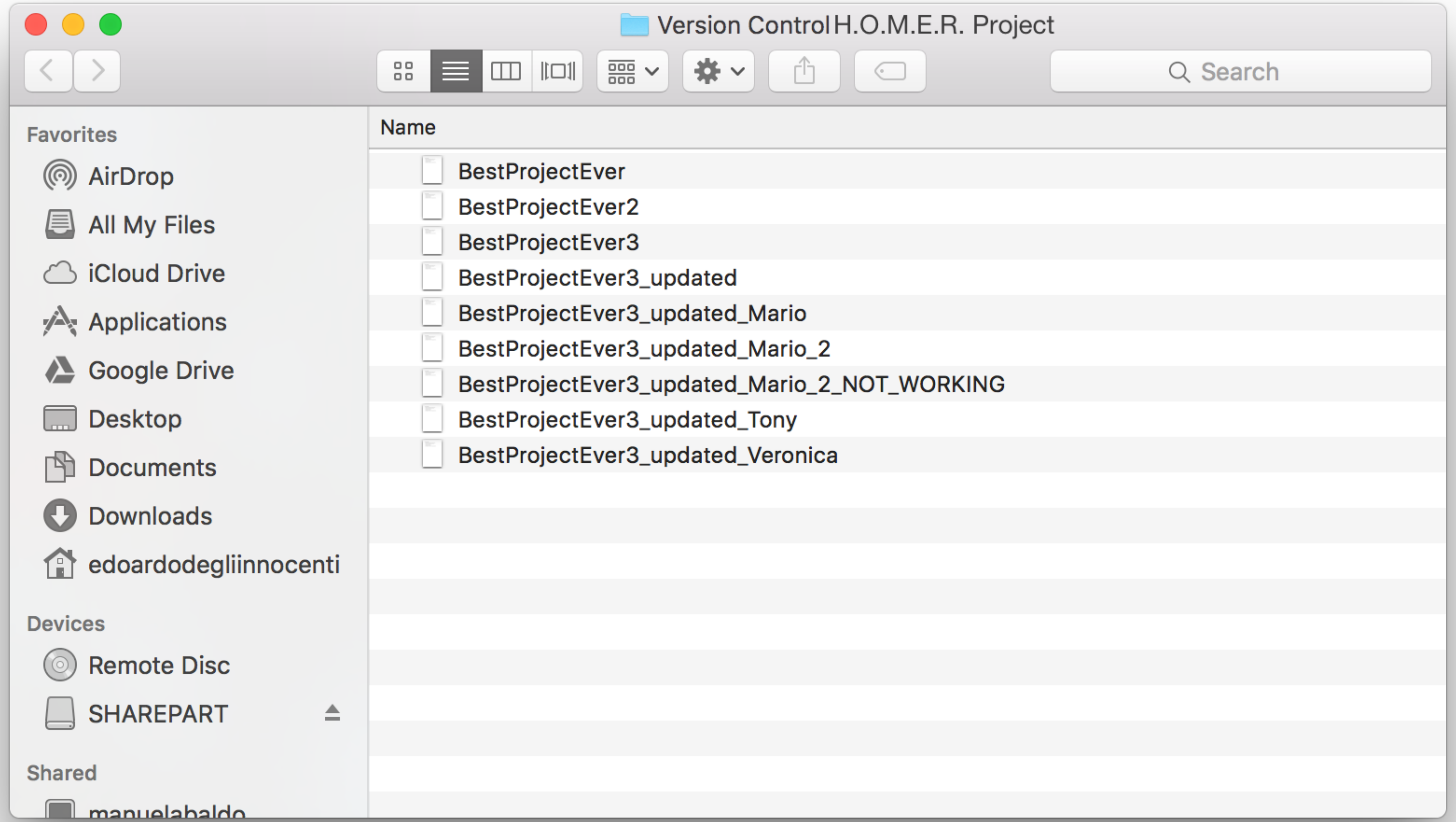
DID YOU EVER DO THIS?



Name

BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated
BestProjectEver3_updated_Mario
BestProjectEver3_updated_Mario_2
BestProjectEver3_updated_Tony
BestProjectEver3_updated_Veronica

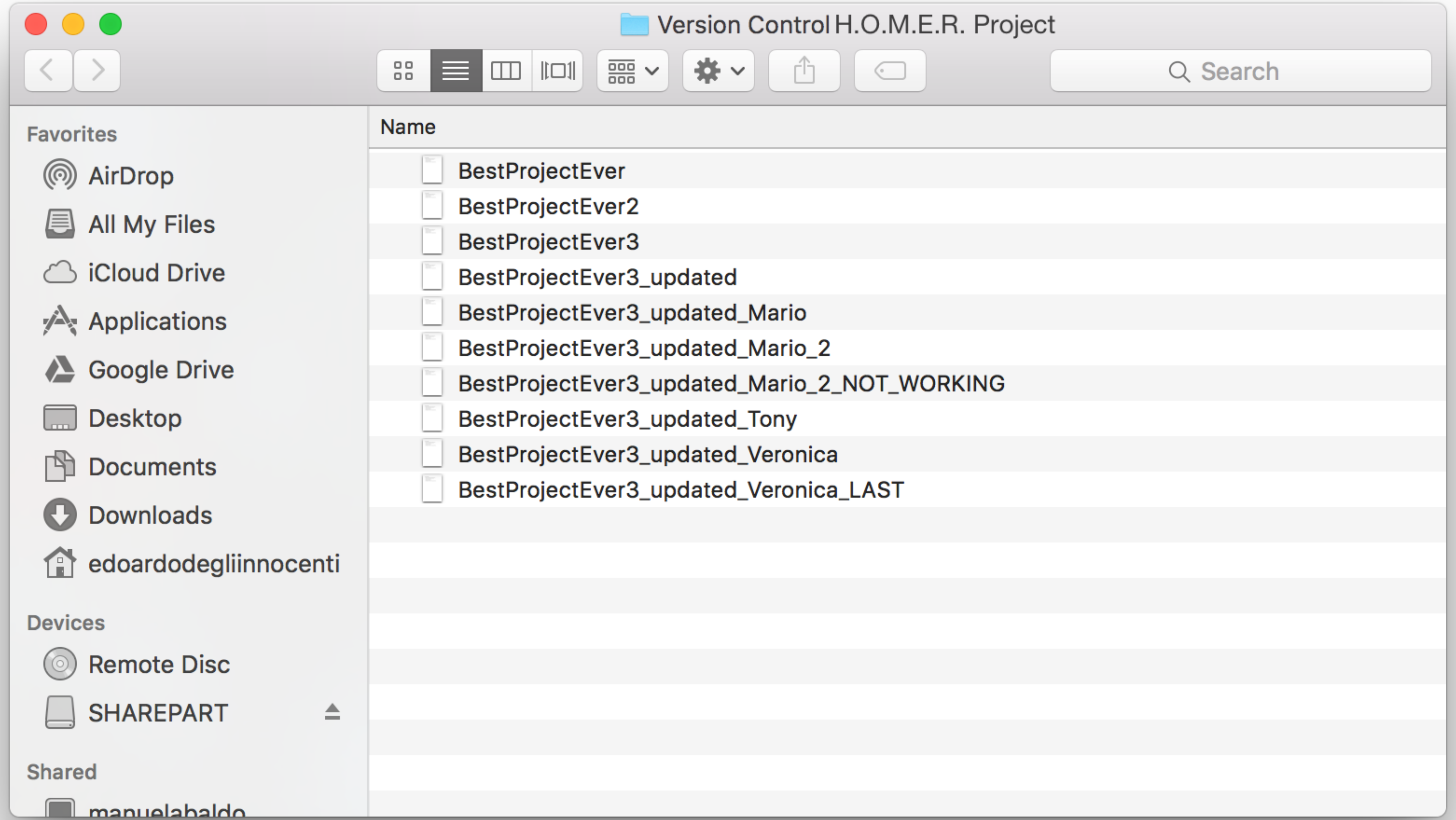
DID YOU EVER DO THIS?



Name

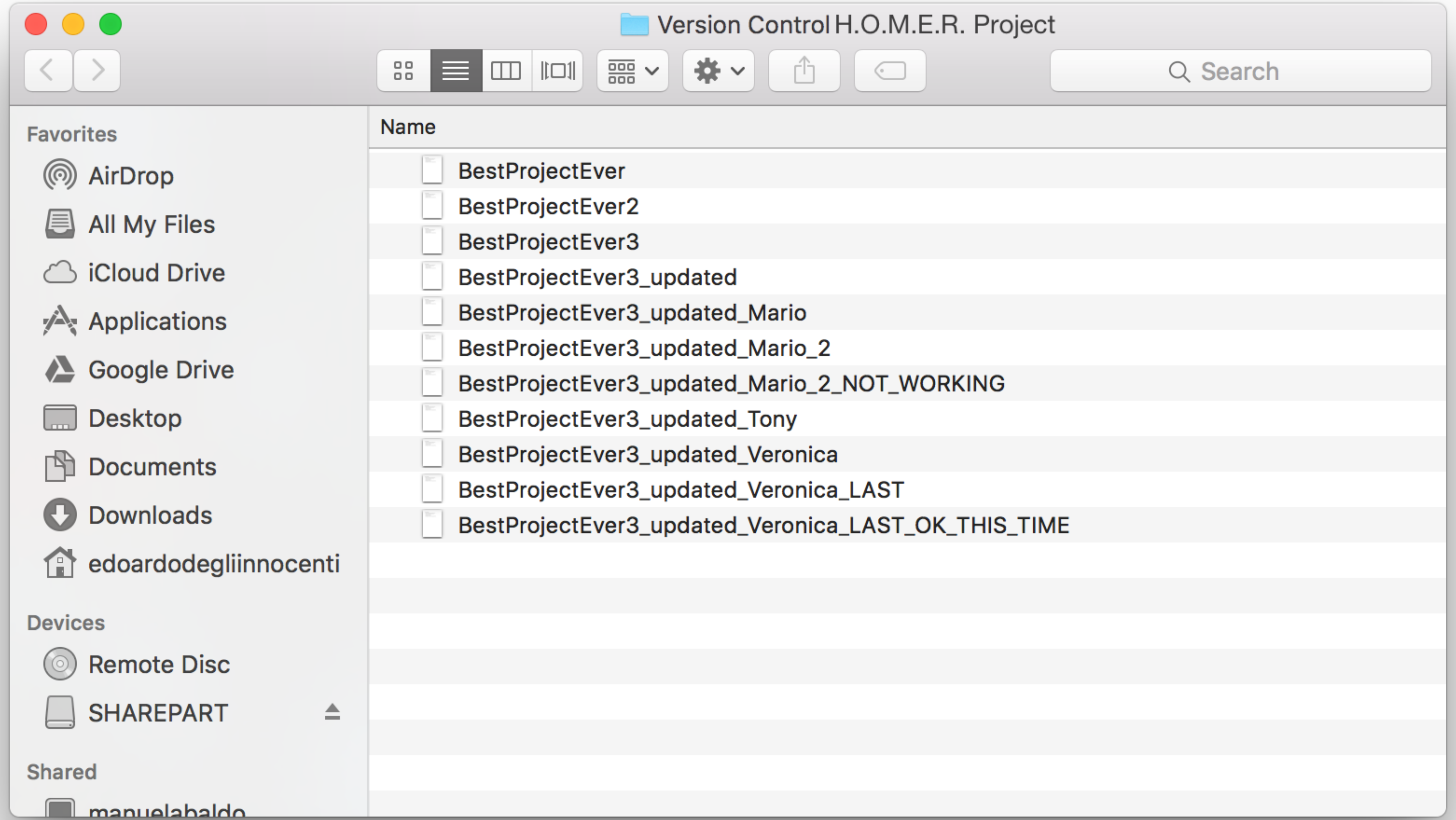
BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated
BestProjectEver3_updated_Mario
BestProjectEver3_updated_Mario_2
BestProjectEver3_updated_Mario_2_NOT_WORKING
BestProjectEver3_updated_Tony
BestProjectEver3_updated_Veronica

DID YOU EVER DO THIS?



Name
BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated
BestProjectEver3_updated_Mario
BestProjectEver3_updated_Mario_2
BestProjectEver3_updated_Mario_2_NOT_WORKING
BestProjectEver3_updated_Tony
BestProjectEver3_updated_Veronica
BestProjectEver3_updated_Veronica_LAST

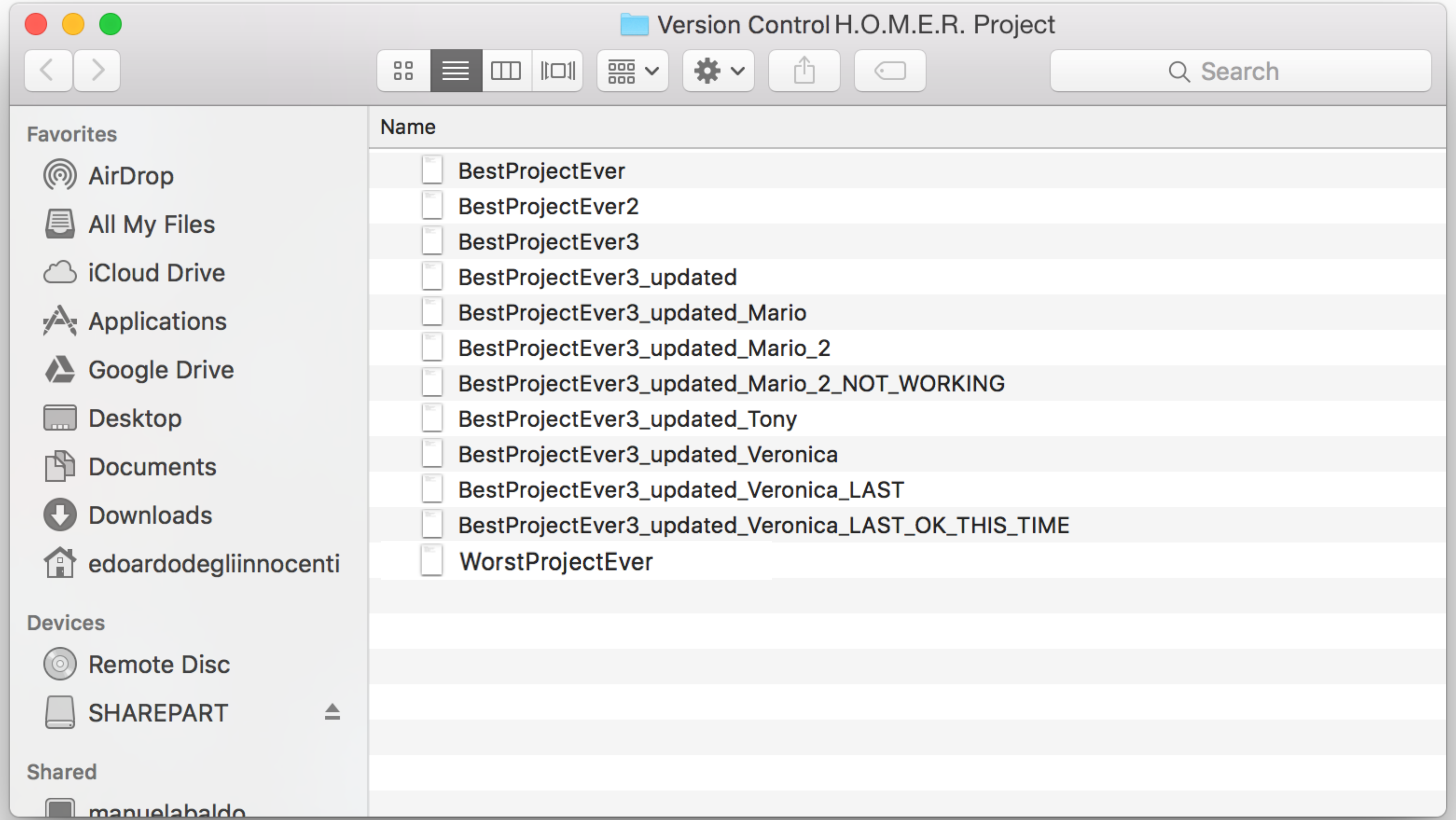
DID YOU EVER DO THIS?



Name

BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated
BestProjectEver3_updated_Mario
BestProjectEver3_updated_Mario_2
BestProjectEver3_updated_Mario_2_NOT_WORKING
BestProjectEver3_updated_Tony
BestProjectEver3_updated_Veronica
BestProjectEver3_updated_Veronica_LAST
BestProjectEver3_updated_Veronica_LAST_OK_THIS_TIME

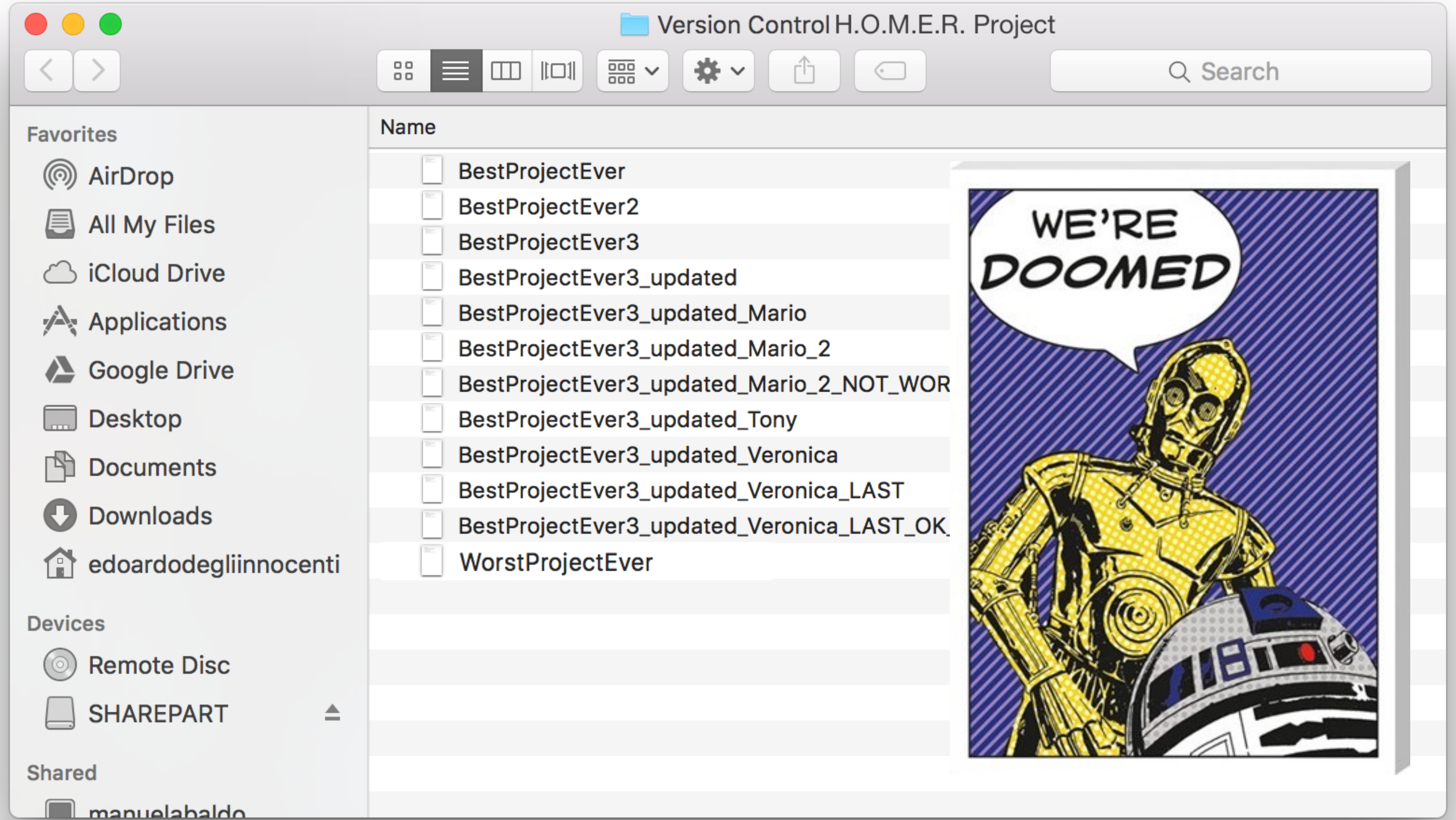
DID YOU EVER DO THIS?



Name

BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated
BestProjectEver3_updated_Mario
BestProjectEver3_updated_Mario_2
BestProjectEver3_updated_Mario_2_NOT_WORKING
BestProjectEver3_updated_Tony
BestProjectEver3_updated_Veronica
BestProjectEver3_updated_Veronica_LAST
BestProjectEver3_updated_Veronica_LAST_OK_THIS_TIME
WorstProjectEver

DID YOU EVER DO THIS?



BestProjectEver
BestProjectEver2
BestProjectEver3
BestProjectEver3_updated
BestProjectEver3_updated_Mario
BestProjectEver3_updated_Mario_2
BestProjectEver3_updated_Mario_2_NOT_WOR
BestProjectEver3_updated_Tony
BestProjectEver3_updated_Veronica
BestProjectEver3_updated_Veronica_LAST
BestProjectEver3_updated_Veronica_LAST_OK_
WorstProjectEver

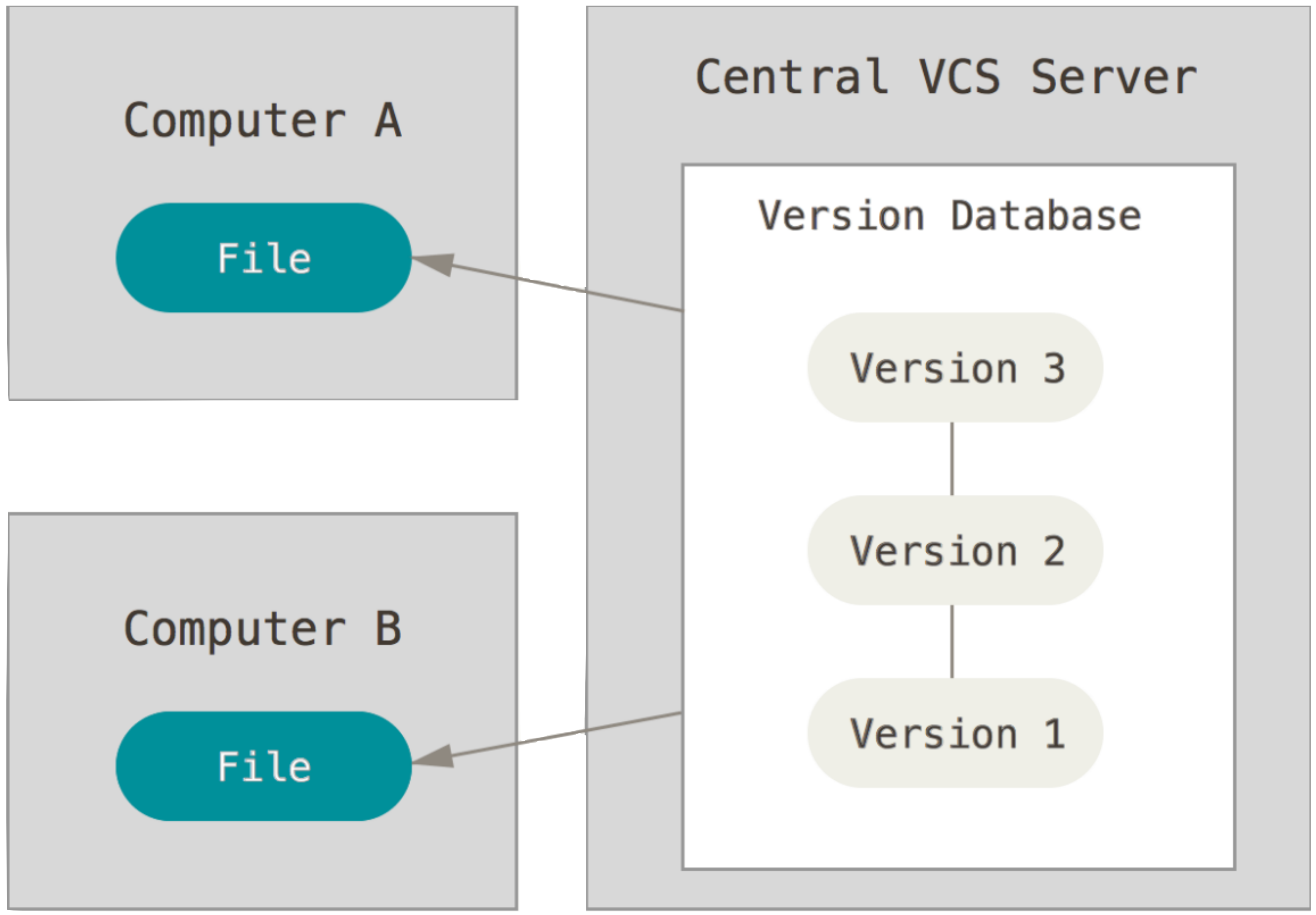
WHAT IS A VERSION CONTROL SYSTEM?

CVCS vs DVCS

CVCS - CENTRALISED VERSION CONTROL SYSTEM

- There is ONE SERVER hosting all project's files
- Developers have to connect to the server in order to "commit" changes
- PRO: Developers no longer have to keep many copies of files on their computers as mentioned before, because the version control tool manages all the different versions of the same project.
- CON: If you lack of connectivity you can't work on the project.
- examples: SVN, CVS or Perforce

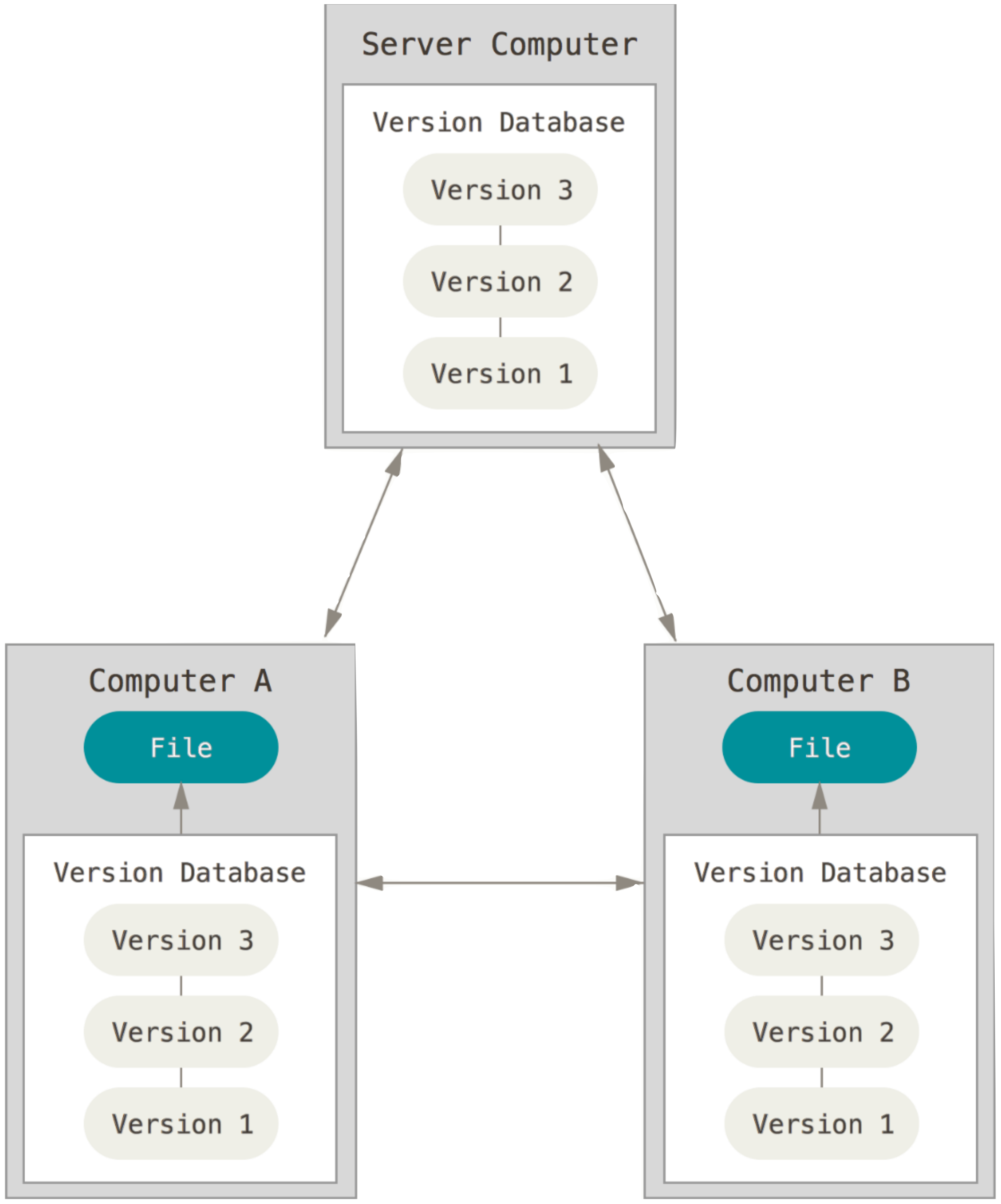
CVCS – CENTRALISED VERSION CONTROL SYSTEM



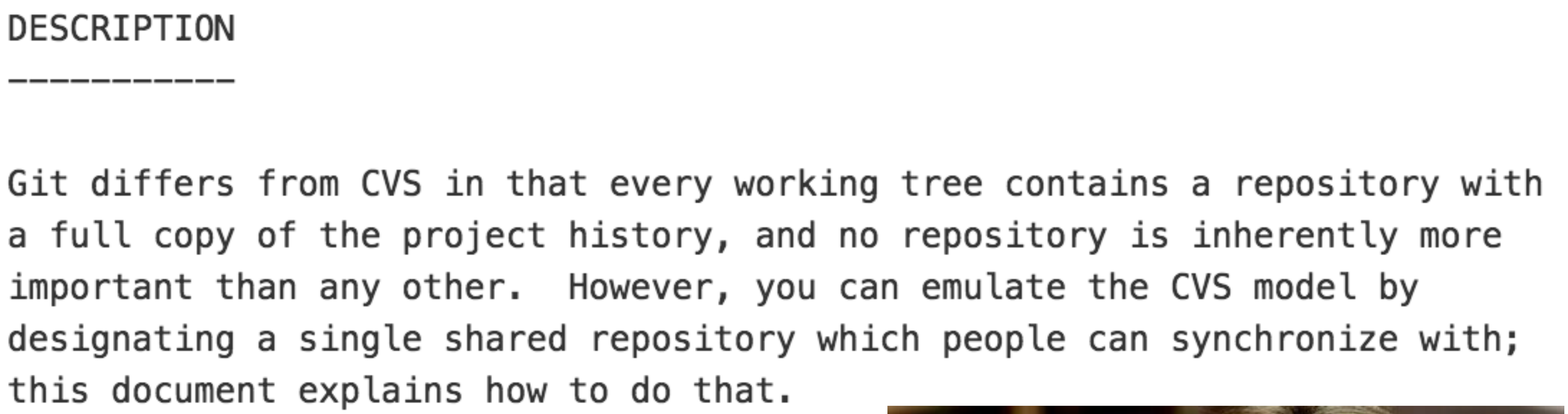
DVCS - DISTRIBUTED VERSION CONTROL SYSTEM

- Every developer working on the project has HIS OWN COPY of all the files and versions from the beginning.
- Every developer has the FULL HISTORY of the project on their own computer
- You can still define a repository as “central”
- You can share some changes with another developer before pushing them to everyone
- examples: Git or Mercurial

DVCS - DISTRIBUTED VERSION CONTROL SYSTEM



DVCS - DISTRIBUTED VERSION CONTROL SYSTEM



```
DESCRIPTION
-----------

Git differs from CVS in that every working tree contains a repository with
a full copy of the project history, and no repository is inherently more
important than any other.  However, you can emulate the CVS model by
designating a single shared repository which people can synchronize with;
this document explains how to do that.
```

Images taken from: https://github.com/git/git/blob/master/Documentation/gitcvs-migration.txt

WHAT IS A VERSION CONTROL SYSTEM?

TERMINOLOGY

TERMINOLOGY

- BRANCH: When a set of files is duplicated and can be modified independently from the origin.
- TRUNK / MAINLINE / MASTER: Represents the main line of development.
- WORKING COPY: The local copy of files of a repository. Is the copy on which you are working on.

TERMINOLOGY

- CLONE: Creating a new repository copying all files and revisions from another repository.
- PULL / FETCH: Downloading revisions from a repository to your own repository, usually done before pushing in order to check if other developers made some changes.
- PUSH: Uploading your changes to a repository.

TERMINOLOGY

- MERGE: is a fundamental operation that reconciles some changes made to a set of files, for example when two people on two different computers modified the same set of file at the same time.
- COMMIT: Called also CHECK IN, means that a set of files modified on your working copy is merged into your own repository.
- CONFLICT: When different parties make some changes to the same set of files, sometimes the the system is unable to reconcile the changes.

WHAT IS A VERSION CONTROL SYSTEM?

GIT

MEANING

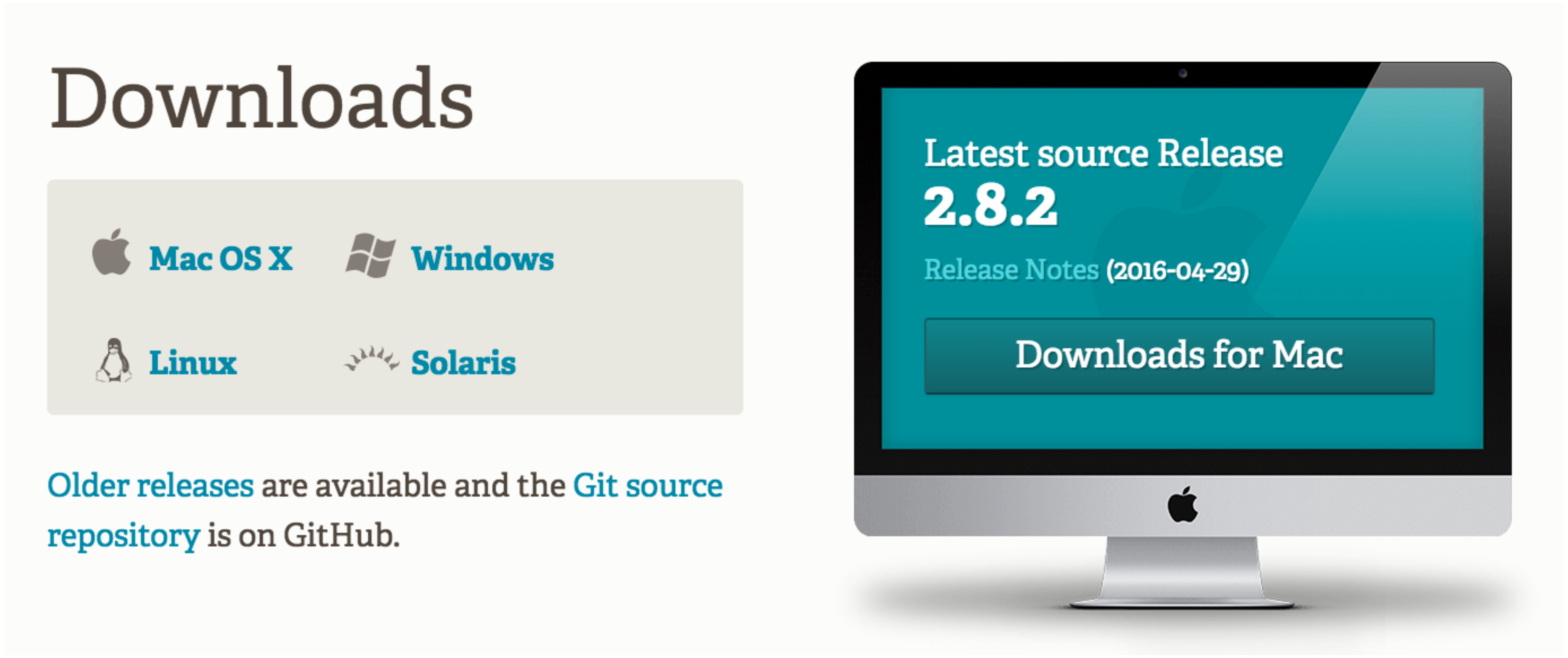
The name "git" was given by Linus Torvalds when he wrote the very first version. He described the tool as "the stupid content tracker" and the name as (depending on your mood):

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
- "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- "goddamn idiotic truckload of sh*t": when it breaks

Taken from: https://github.com/git/git/blob/master/README.md

DOWNLOADS

https://git-scm.com/downloads



Downloads

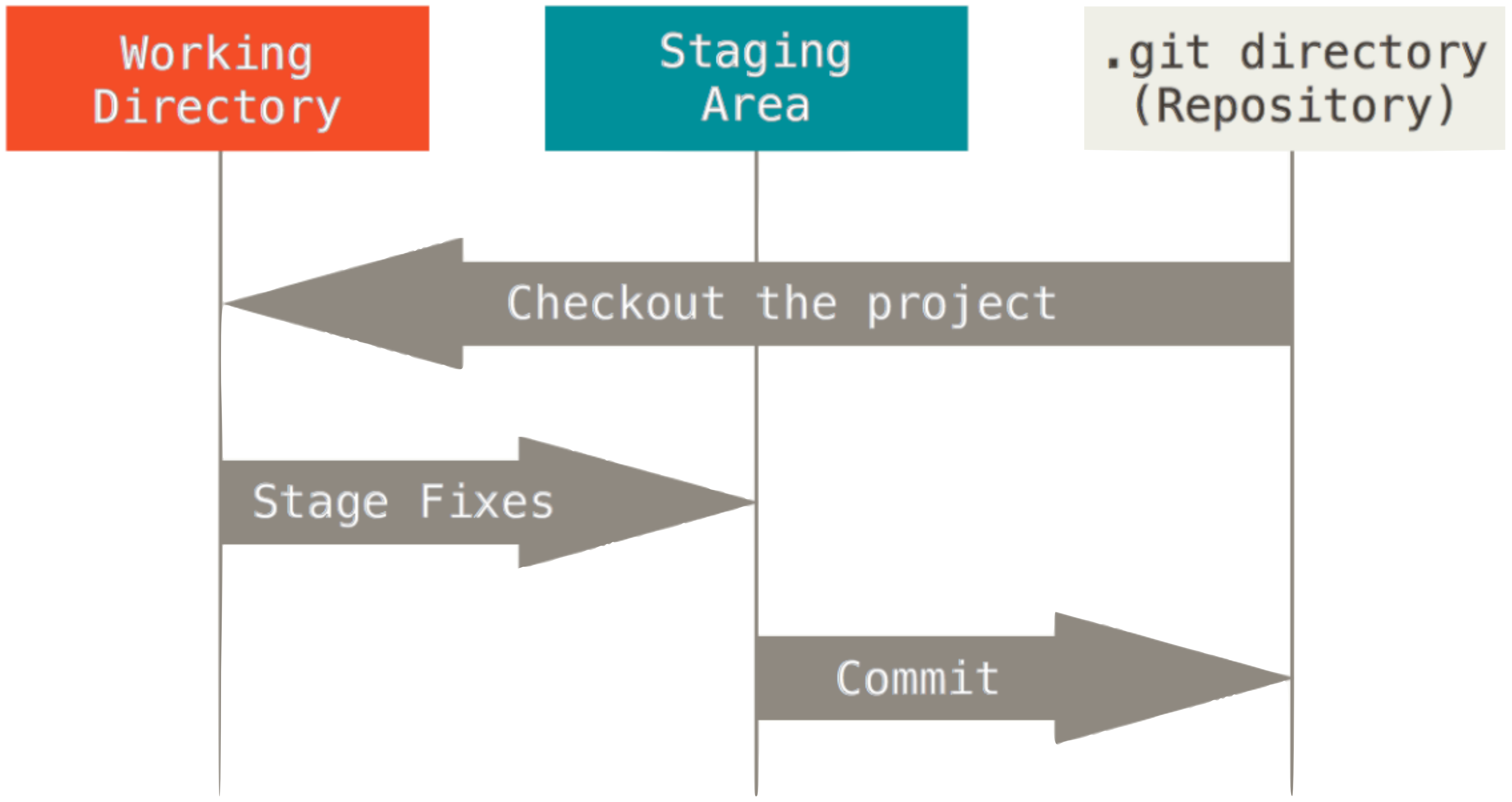
Mac OS X Windows

Linux Solaris

Older releases are available and the Git source repository is on GitHub.

Images taken from: https://git-scm.com

THE THREE STATES



GENERAL SETUP

```
$ git --version


$ git config --global user.name “Edoardo Degli Innocenti”

$ git config --global user.email “degliinn@dei.unipd.it”

$ git config --list


$ git help <verb>

$ git <verb> --help
```

LOCAL PROJECT - SETUP

If you have a local project to track

```
$ git init
```

```
$ git status
```

You can add files to the STAGING AREA with the command

```
$ git add <file name>
```

You can add ALL files to the STAGING AREA with the command

```
$ git add -A
```

LOCAL PROJECT - COMMIT

When you make a commit, remember to write a detailed message about what you are committing.

```
$ git commit -m "My first commit"
```

To see a list of commits

```
$ git log
```

REMOTE REPOSITORY - SETUP

If you have a remote project to track

```
$ git clone <url> <path where to clone>
```

All previous command seen before work correctly.

```
$ git status

$ git add -A

$ git commit -m “Message about the commit”
```

To see informations about changes in a set of files

```
$ git diff
```

REMOTE REPOSITORY - PULL / PUSH

In order to see informations about all branches in a repository

```
$ git branch -a
```

If you want to upload your commits, before remember to do a PULL request

```
$ git pull origin master
```

```
$ git push origin master
```

To create a new branch to work on

```
$ git branch <branch name>
```

To switch to a different branch

```
$ git checkout <branch name>
```

REMOTE REPOSITORY - MERGE

In order to merge the files modified in a different branch to the master one

```
$ git checkout <master branch name>

$ git pull <master branch name>

$ git merge <branch to merge name>

$ git push <master branch name>
```

If you want to delete (locally and remotely) the merged branch

```
$ git branch -d <branch name to delete>

$ git push origin --delete <branch name to delete>
```

REMOTE REPOSITORY – UNDOING THINGS

If you have just done a commit and you forgot to add some files

```
$ git commit –m “initial commit”

$ git add <forgotten file name>

$ git commit --amend
```

If you want to revert your file back to the last committed version

```
$ git checkout –– <file name to revert>
```

EXERCISE

Create a folder in order to store your working copy and go there with the terminal

```
$ git clone https://labesp16@bitbucket.org/labesp16/labesp16.git

$ cd labesp16


$ git status

$ git log
```

EXERCISE

Go to the repository folder and add a list of your choice

```
$ git status

$ git add -A

$ git commit -m “<describe your list>”


$ git pull origin master

$ git push origin master
```

PASSWORD: lab-esp16

EXERCISE

Create a new branch

```
$ git branch <new branch name>

$ git checkout <new branch name>

$ git status

$ git log
```

Push some files, and invite others to modify your files.

Try to manage some conflicts.

MORE INFOS

Read the book about Git at

https://git-scm.com/book/en/v2