ANNEX A · GIT REMOTES AND COLLABORATION

Name: _____ **Group:** _____ **Date:** _____

Our forge: _____ My user: _____ My partner: _____

In the commands, write the forge's address instead of `FORGE` and the user name instead of `USER`.

PART A **Give the album a second home — alone**

1. Show your public key and copy the whole line. If the file does not exist: `ssh-keygen -t ed25519`.

```
cat ~/.ssh/id_ed25519.pub
```

On the forge: Settings → SSH keys → add key → paste. Then test:

```
ssh -T git@FORGE
```

DONE WHEN the forge answers with your user name.

2. On the forge create a new repository: name `tramuntana-docs`, **private**, **empty** (no README, no licence, no ignore file).

3. Connect and publish:

```
cd ~/tramuntana-docs
git remote add origin git@FORGE:USER/tramuntana-docs.git
git remote -v
git push -u origin main
git status
```

DONE WHEN the forge shows your files and your commits in the browser, and `git status` says `up to date with 'origin/main'`.

PART B **A guest in your repository — pairs: owner and guest**

4. Owner: on the forge, add your partner **and the teacher** as collaborators with write permission.
Guest: clone the owner's repository into a separate folder and look at its history:

```
mkdir ~/guest && cd ~/guest
git clone git@FORGE:OWNER/tramuntana-docs.git
cd tramuntana-docs
git log --oneline
```

5. Guest: add the line `Reviewed by: your name` at the end of `README.md`, then:

```
git add README.md
git commit -m "Add reviewer to the README"
git push
```

Owner: run `git pull` and `git log`. Word Git prints for this simple update: _____

DONE WHEN the owner's `git log` shows a commit with the guest's name.

PART C **A branch and a pull request — owner writes, guest reviews**

6. Owner:

```
git switch -c add-network-page
```

Create `docs/03-network.md`: title `# Network`, the address of the server, the network it is in, the ports it listens on. Then:

```
git add docs/03-network.md
git commit -m "Add network page"
git push -u origin add-network-page
```

7. Owner: on the forge, open a pull request from `add-network-page` into `main`.
Guest: read the changes, write **one comment about a specific line**, then merge it.
Owner:

```
git switch main
git pull
git branch -d add-network-page
```

DONE WHEN `ls docs` on `main` shows `03-network.md`, and the pull request appears as merged with a comment.

PART D — A conflict on purpose

8. Both, each in your own clone of the owner's repository: change the sentence under the title of `README.md` — each in a different way — and commit. **Owner:** `git push`.

9. Guest:

```
git push
git config --global pull.rebase false
git pull
git status
```

Word after the `!` when the push is refused: _____ File in conflict: _____

10. Guest, with the owner looking: open `README.md`. Copy here the three marker lines you see:

_____

_____

Agree on the final sentence, delete the three marker lines, save. Then:

```
git add README.md
git commit -m "Merge main: agree on the first line"
git push
git log --oneline --graph
```

Owner: `git pull`.
DONE WHEN both clones show the same sentence, `grep "<<<<" README.md` prints nothing, and the graph shows two lines joining.

EXIT — Exit ticket — alone

1. `origin` is…

- ☐ a) the first commit
- ☐ b) the main branch
- ☐ c) the usual short name of the remote
- ☐ d) the owner of the repository

2. Your push is rejected with "fetch first". What do you do?

- ☐ a) `git pull`, then push again
- ☐ b) delete the remote repository
- ☐ c) `git init` again
- ☐ d) wait a minute and retry

3. A pull request is…

- ☐ a) a Git command that downloads commits
- ☐ b) a copy of the repository
- ☐ c) another name for `git pull`
- ☐ d) a proposal to merge a branch, with review

4. In a conflict, the lines between `<<<<<<< HEAD` and `=======` are…

- ☐ a) the version of the other person
- ☐ b) your version
- ☐ c) the merged result
- ☐ d) a Git error message

5. You edited a file, saved it, and ran `git push`. The forge shows no change because…

- ☐ a) the forge is slow
- ☐ b) the key is wrong
- ☐ c) you did not commit
- ☐ d) the branch is private