ANNEX A · PRACTICAL 7.2 — YOUR FIRST REPOSITORY

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

Work in `deb-lab`. Type the commands; do not paste them. Hand in `practical-7.2-surname.md` on Classroom today. **Keep the folder `~/tramuntana-docs` until 9 June: it becomes your assessed work.**

1 Install and introduce yourself

```
sudo apt update && sudo apt install -y git
git config --global user.name "Your Name"
git config --global user.email "your school e-mail"
git config --global init.defaultBranch main
git config --global --list
```

DONE WHEN the last command prints three lines with your name, your e-mail and `main`.

2 Create the repository

```
mkdir ~/tramuntana-docs
cd ~/tramuntana-docs
git init
ls -a
git status
```

Hidden folder that appeared: _____ Branch you are on: _____

3 First commit

Create `README.md` with a title (`# Tramuntana office server`) and one sentence saying what this repository is. Then:

```
git status
git add README.md
git status
git commit -m "Add README with project summary"
git log
```

Git's word for the file **before** `git add`: _____ Start of your commit id: _____

DONE WHEN `git log` shows one commit with your name.

4 Change, look, commit

Add a line to `README.md` naming the server (`ubu-srv`) and its operating system. Then:

```
git status --short
git diff
git add README.md
git diff
git diff --staged
git commit -m "Describe the server in the README"
```

Letter before the file in `git status --short`: _____ First symbol of your new line in `git diff`: _____

After `git add`, what does plain `git diff` print? _____ (you explain why in the report)

5 A folder and a second file

Run `mkdir docs` and create `docs/01-overview.md`: title `# Overview` and two sentences (what the server is for, who uses it). Then `git status`, `git add docs/01-overview.md` and commit with your own message.

DONE WHEN `git log --oneline` shows three commits.

6 Keep things out

```
mkdir build
touch build/test.pdf
echo "password: 1234" > passwords.txt
git status
```

Now create the file `.gitignore` with two lines: `build/` and `passwords.txt`. Then run the commands on the right.

```
git status
git add .
git status --short
git commit -m "Ignore build and secrets"
```

DONE WHEN `passwords.txt` and `build/` are not in the second `git status`, and not in the commit.

7 Break it and get it back

```
echo "oops" > README.md
git diff --stat
git restore README.md
cat README.md
```

DONE WHEN `README.md` is back; `git status` says `working tree clean`.

8 Read the album

```
git log --oneline
git show --stat HEAD
```

Number of commits: _____

Files changed by the last one: _____

9 Real content: one idea, one commit

Complete `docs/01-overview.md` with the data of **your** server: what it is for · who uses it (the three departments) · its name and address (`hostnamectl`, `ip -br address` on `ubu-srv`) · virtual CPUs, memory, disk · operating system and version. Commit after each idea.

DONE WHEN at least 6 commits in total, each message says what changed.

REPORT What to hand in — marking out of 10

`practical-7.2-surname.md` with: the output of `git log --oneline`, `git status` and `cat .gitignore`; and three answers: **(1)** Why did `git diff` print nothing after `git add`? **(2)** What would `git add .` have done with `passwords.txt` without step 6? **(3)** Working directory, staging area, repository: explain each in one line, in your words.

Marking: set-up 1 · 6 clear commits 3 · ignore file works 2 · overview with real data 2 · answers 2.

EXIT Exit ticket — alone

1. A commit is…

- ☐ a) the last saved version of one file
- ☐ b) a snapshot of the project, with a message
- ☐ c) a copy of the folder on a website
- ☐ d) a list of files to ignore

2. Which command chooses what goes into the next commit?

- ☐ a) `git status`
- ☐ b) `git log`
- ☐ c) `git init`
- ☐ d) `git add`

3. You changed a file and ran no Git command. `git diff` shows…

- ☐ a) the lines you changed
- ☐ b) nothing
- ☐ c) the last commit
- ☐ d) all the files of the project

4. A password was committed by mistake. What makes it safe again?

- ☐ a) deleting the file and committing
- ☐ b) adding it to `.gitignore`
- ☐ c) changing the password
- ☐ d) `git restore`

5. `git restore notes.md` …

- ☐ a) deletes the file from the history
- ☐ b) discards uncommitted changes to it
- ☐ c) undoes the last commit
- ☐ d) uploads the file