

Tarea principal (7 puntos)

1. Realiza la instalación de git en linux

Comando para instalar git:

```
sudo apt update && sudo apt install git -y
```

2. Explica cómo obtener Git a través del propio Git para futuras actualizaciones, de manera que descargaría automáticamente el código fuente desde su repositorio.

```
xubu@xubu-VirtualBox:/usr/src/git$ # Instalar dependencias necesarias
sudo apt install make autoconf libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libssl-dev -y

# Clonar el código fuente de Git
cd /usr/src
sudo git clone https://github.com/git/git.git

# Entrar en el directorio clonado
cd git

# Compilar e instalar
sudo make prefix=/usr/local all
sudo make prefix=/usr/local install
```

```
done && \
for p in git-add git-am git-annotate git-apply git-archive git-bisect git-blame git-branch git-bugreport git-bundle git-cat-file git-check-attr git-check-ignore git-check-mailmap git-check-ref-format git-checkout--worker git-checkout-index git-checkout git-clean git-clone git-column git-commit-graph git-commit-tree git-commit git-config git-count-objects git-credential-cache-daemon git-credential-cache git-credential-store git-describe git-diagnose git-diff-files git-diff-index git-diff-tree git-diff git-difftool git-fast-export git-fast-import git-fetch-pack git-fetch git-fmt-merge-msg git-for-each-ref git-for-each-repo git-fsck git-fsmonitor git-gc git-get-tar-commit-id git-grep git-hash-object git-help git-hook git-index-pack git-init-db git-interpret-trailers git-log git-ls-files git-ls-remote git-ls-tree git-mailinfo git-mailsplit git-merge-base git-merge-file git-merge-index git-merge-ours git-merge-recursive git-merge-tree git-merge git-mktag git-mktree git-multi-pack-index git-mv git-name-rev git-notes git-pack-objects git-pack-redundant git-pack-refs git-patch-id git-prune-packed git-prune git-pull git-push git-range-diff git-read-tree git-rebase git-receive-pack git-reflog git-refs git-remote-ext git-remote-fd git-remote git-repack git-replace git-replay git-rerere git-reset git-rev-list git-rev-parse git-revert git-rm git-send-pack git-shortlog git-show-branch git-show-index git-show-ref git-sparse-checkout git-stash git-stripspace git-submodule-helper git-symbolic-ref git-tag git-unpack-file git-unpack-objects git-update-index git-update-ref git-update-server-info git-upload-archive git-upload-pack git-var git-verify-commit git-verify-pack git-verify-tag git-worktree git-write-tree git-cherry git-cherry-pick git-format-patch git-fsck-objects git-init git-maintenance git-merge-subtree git-restore git-show git-stage git-status git-switch git-version git-whatchanged; do \
    rm -f "$execdir/$p" && \
    if test -z ""; \
    then \
        test -n "" && \
        ln -s "$destdir_from_execdir_SQ/bin/git" "$execdir/$p" || \
        { test -z "" && \
            ln "$execdir/git" "$execdir/$p" 2>/dev/null || \
            ln -s "git" "$execdir/$p" 2>/dev/null || \
            cp "$execdir/git" "$execdir/$p" || exit; }; \
        fi \
    done && \
remote_curl_aliases="git-remote-https git-remote-ftp git-remote-https" && \
for p in $remote_curl_aliases; do \
    rm -f "$execdir/$p" && \
    test -n "" && \
    ln -s "git-remote-http" "$execdir/$p" || \
    { test -z "" && \
        ln "$execdir/git-remote-http" "$execdir/$p" 2>/dev/null || \
        ln -s "git-remote-http" "$execdir/$p" 2>/dev/null || \
        cp "$execdir/git-remote-http" "$execdir/$p" || exit; }; \
    done
```

Resultado del fin de la ejecución, pues ya podríamos con el siguiente comando actualizar git de manera automática.

Utilizamos los siguientes comandos, una vez haya acabado:

```
xubu@xubu-VirtualBox:/usr/src/git$ cd /usr/src/git
sudo git pull
sudo make prefix=/usr/local all
sudo make prefix=/usr/local install
Already up to date.
```

Y nos actualiza.

3. Comprueba la versión que se ha instalado de Git.

```
xubu@xubu-VirtualBox:/usr/src/git$ git --version
git version 2.48.1.268.q9520f7d998
```

4. Establecer el nombre de usuario y dirección de correo electrónico en la configuración de Git.

```
git version 2.48.1.268.q9520f7d998
xubu@xubu-VirtualBox:/usr/src/git$ git config --global user.name "Miisae"
git config --global user.email "hugofernandezpinho@gmail.com"
```

Verificamos que se haya configurado correctamente

```
xubu@xubu-VirtualBox:/usr/src/git$ git config --global --list
user.name=Miisae
user.email=hugofernandezpinho@gmail.com
```

5. Cambiar el editor de texto que trae por defecto Git al editor emacs.

```
xubu@xubu-VirtualBox:/usr/src/git$ git config --global core.editor "emacs"
```

6. Dentro de la carpeta /var/cache/git/ crear una carpeta para un nuevo proyecto denominado tarea_DAW06 e iniciar un repositorio con el nuevo proyecto

Creamos un directorio, vamos al directorio y iniciamos git dentro de el.

```
xubu@xubu-VirtualBox: /usr/src/git$ sudo mkdir -p /var/cache/git/tarea_DAW06
cd /var/cache/git/tarea_DAW06
sudo git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /var/cache/git/tarea_DAW06/.git/
```

7. Haz alguno de los siguientes ejercicios indicados en (como crear un fichero y hacer commit del cambio):

```
xubu@xubu-VirtualBox: /var/cache/git/tarea_DAW06$ sudo su
root@xubu-VirtualBox: /var/cache/git/tarea_DAW06# echo "Este es mi archivo README en tarea_DAW06" > README.md
root@xubu-VirtualBox: /var/cache/git/tarea_DAW06# git add README.md
root@xubu-VirtualBox: /var/cache/git/tarea_DAW06# git commit -m "Primer commit: Agregado README.md"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

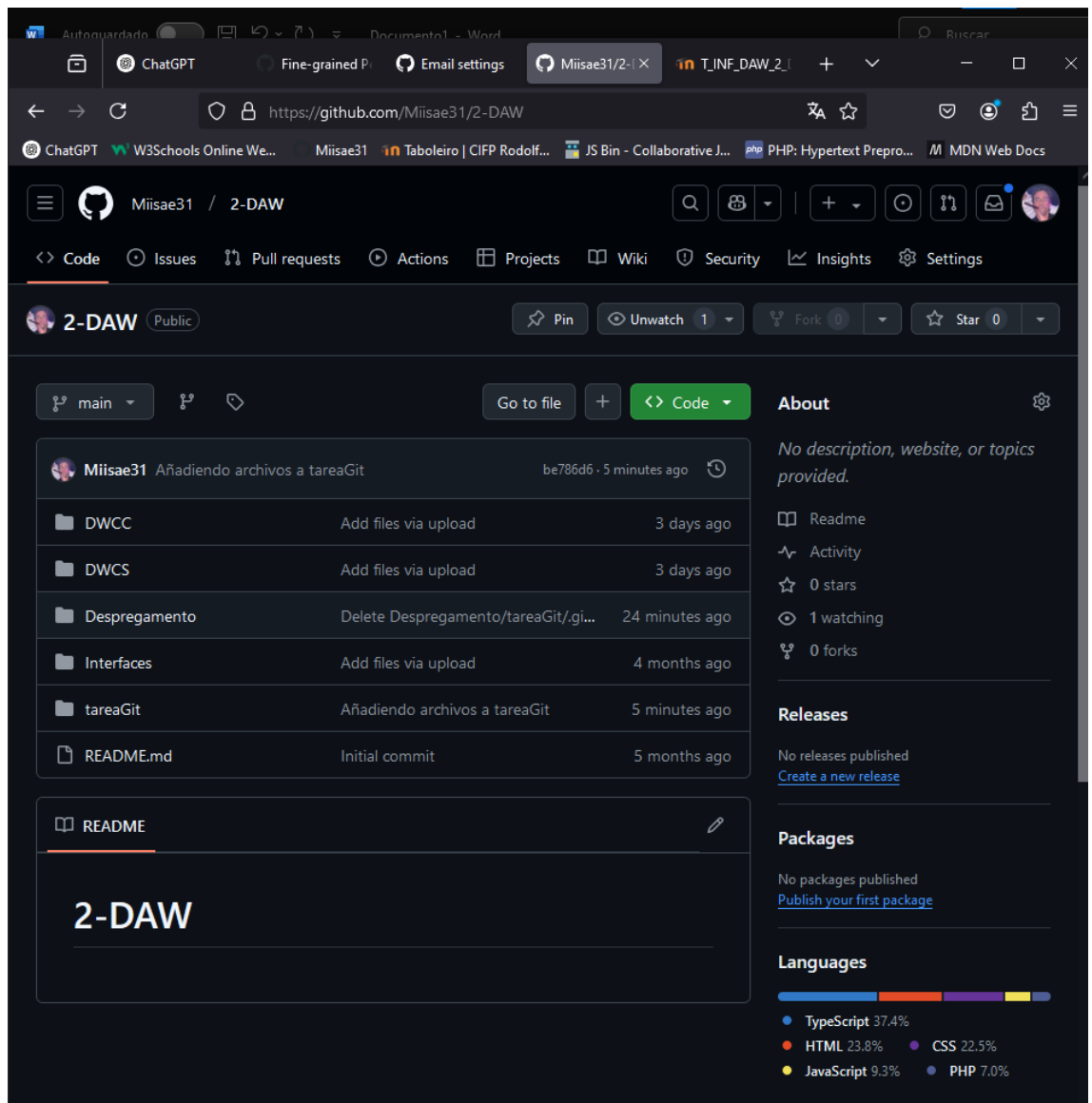
fatal: unable to auto-detect email address (got 'root@xubu-VirtualBox.(none)')
root@xubu-VirtualBox: /var/cache/git/tarea_DAW06# git config --global user.email "hugofernandezpinho@gmail.com"
root@xubu-VirtualBox: /var/cache/git/tarea_DAW06# git config --global user.name "Miisae31"
root@xubu-VirtualBox: /var/cache/git/tarea_DAW06# git commit -m "Primer commit: Agregado README.md"
[master (root-commit) bdiff122] Primer commit: Agregado README.md
1 file changed, 1 insertion(+)
 create mode 100644 README.md
root@xubu-VirtualBox: /var/cache/git/tarea_DAW06#
```

Creamos un archivo, le colocamos cualquier cosa dentro, indicamos el correo de mi cuenta de github y mi nombre de usuario y hacemos un commit.

8. Sube el resultado a tu repositorio de github

```
xubu@xubu-VirtualBox:/var/cache/git/Despregamento$ git clone https://github.com/Miisae31/2-DAW.git
Cloning into '2-DAW'...
remote: Enumerating objects: 926, done.
remote: Counting objects: 100% (627/627), done.
remote: Compressing objects: 100% (589/589), done.
remote: Total 926 (delta 211), reused 0 (delta 0), pack-reused 299 (from 1)
Receiving objects: 100% (926/926), 52.74 MiB | 6.01 MiB/s, feito.
Resolving deltas: 100% (253/253), feito.
xubu@xubu-VirtualBox:/var/cache/git/Despregamento$ cd 2-DAW
xubu@xubu-VirtualBox:/var/cache/git/Despregamento/2-DAW$ mkdir -p tareaGit
xubu@xubu-VirtualBox:/var/cache/git/Despregamento/2-DAW$ echo "Archivo de prueba" > tareaGit/prueba.txt
xubu@xubu-VirtualBox:/var/cache/git/Despregamento/2-DAW$ ls tareaGit/
prueba.txt
xubu@xubu-VirtualBox:/var/cache/git/Despregamento/2-DAW$ git init
git branch -M main
git remote add origin https://github.com/Miisae31/2-DAW.git
Reinitialized existing Git repository in /var/cache/git/Despregamento/2-DAW/.git/
error: remote origin already exists.
xubu@xubu-VirtualBox:/var/cache/git/Despregamento/2-DAW$ git add .
git commit -m "Añadiendo archivos a tareaGit"
[main be786d6] Añadiendo archivos a tareaGit
1 file changed, 1 insertion(+)
create mode 100644 tareaGit/prueba.txt
xubu@xubu-VirtualBox:/var/cache/git/Despregamento/2-DAW$ git push -u origin main
Username for 'https://github.com': Miisae31
Password for 'https://Miisae31@github.com':
Enumerating objects: 5, feito.
Counting objects: 100% (5/5), feito.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), feito.
Writing objects: 100% (4/4), 365 bytes | 365.00 KiB/s, feito.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Miisae31/2-DAW.git
42ef4c5..be786d6 main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
xubu@xubu-VirtualBox:/var/cache/git/Despregamento/2-DAW$
```

Aquí lo que hacemos es iniciar un container dentro de nuestra carpeta, después de eso hacemos un git clone para clonar nuestro repositorio al repositorio local, después de eso lo que tenemos que hacer es crear el archivo y mediante un push lo subimos el archivo y el resultado final seria este.



Y ahí vemos nuestra tarea subida