

# **PAC3 Control de Versions i Documentació**

Alfredo Rafael Vicente Boix

15 de desembre de 2015

# Índex

1	Introducció	3
2	Control de Versions	4
3	Documentació	8
4	Conclusions	14

# 1 Introducció

Aquest document presenta les 3 parts de la pràctica realitzada de l'assignatura Desenvolupament de Programari de la Universitat Oberta de Catalunya. El document està format per dues parts on s'explica el procés i els passos a seguir per resoldre els diferents problemes.

Al primer problema s'ha utilitzat la ferramenta git per a crear el repositori i github<sup>1</sup> per a compartirlo.

El segon problema s'han utilitzat les etiquetes doxygen i el programa doxygen per a poder formatar la documentació amb html.

---

<sup>1</sup><https://github.com/alviboi/PAC3-UOC/>


## 2 Control de Versions


En aquesta part hem decidit crear un Control de versions amb git i github, és epr això que el primer que hem fet és donar-nos d'alta a github i crear un repositori amb el nom de PAC3-UOC. D'aquesta manera el resultat de la nostre dipòsit serà compartit.

### Create a new repository

A repository contains all the files for your project, including the revision history.

---

**Owner**  
 **alviboi** ▼


**Repository name**  
/ PAC3-UOC 


Great repository names are short and memorable. Need inspiration? How about **jocular-octo-spork**.

**Description (optional)**

Pràctica 3 de Desenvolupament de Pogramari de la UOC

---

☒  **Public**  
Anyone can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

---

☒ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▼

Add a license: **None** ▼ ⓘ

---

**Create repository**

Donada la naturalesa de github el dipòsit pot ser públic o privat, en aquest cas hem decidit fer-lo públic, però podriem haver gestionat el projecte per a que fos provat i estigués compartit únicament amb el professor.

Evidentment per a treballar al nostre ordinador, el primer que hem de fer és instal·lar git al nostre ordinador:

*sudo apt-get install git*

Després passem el nostre dipòsit al nostre ordinador amb el següent comandament:

```
alviboi@MEU ~ $ git clone https://github.com/alviboi/PAC3-U0C.git
Cloning into 'PAC3-U0C'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
```

De manera que creem una còpia del dipòsit al nostre ordinador. De moment el dipòsit no té més que un arxiu README que github et crea automàticament. Una vegada creada la còpia al nostre ordinador per a afegir arxius, en aquest cas els arxius `filtres.c` i `filtres.h` que es troben al directori `libsrc`, per tant hem de tenir en compte que en primer lloc hem de afegir-los al nostre repositori amb el commandament:

*git add ./libsrc*

Això no significa que els arxius estiguen al nostre depòsit haurem d'afegir-los, amb un comentari. Pit resultar redundant, però d'aquesta manera pots afegir diversos arxius i posteriorment guardar-los tots de cop al dipòsit amb el commandament:

*git commit -m "Afegeint documents"*

Podem veure l'exemple a la següent imatge:

```
alviboi@MEU ~/PAC3 $ git add ./libsrc
alviboi@MEU ~/PAC3 $ git commit -m "Afegeint documents"
[master (root-commit) 8fc9c87] Afegeint documents
 2 files changed, 93 insertions(+)
 create mode 100644 libsrc/filtres.c
 create mode 100644 libsrc/filtres.h
```

Però, certament, si el que necessitem és que els arxius que hem modificat siguin accessibles per a tot el món que necessite treballar en el projecte, hem de pujar-los al nostre repositori haurem de fer el següent:

*git push*

Així tots els arxius que tenim al nostre dipòsit local passaran a estar accessibles per a tot el món que desitge accedir a ells:

```

alviboi@MEU ~/PAC3-UOC $ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': alviboi
Password for 'https://alviboi@github.com':
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 991 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/alviboi/PAC3-UOC.git
5ec06ac..8df51d7  master -> master

```

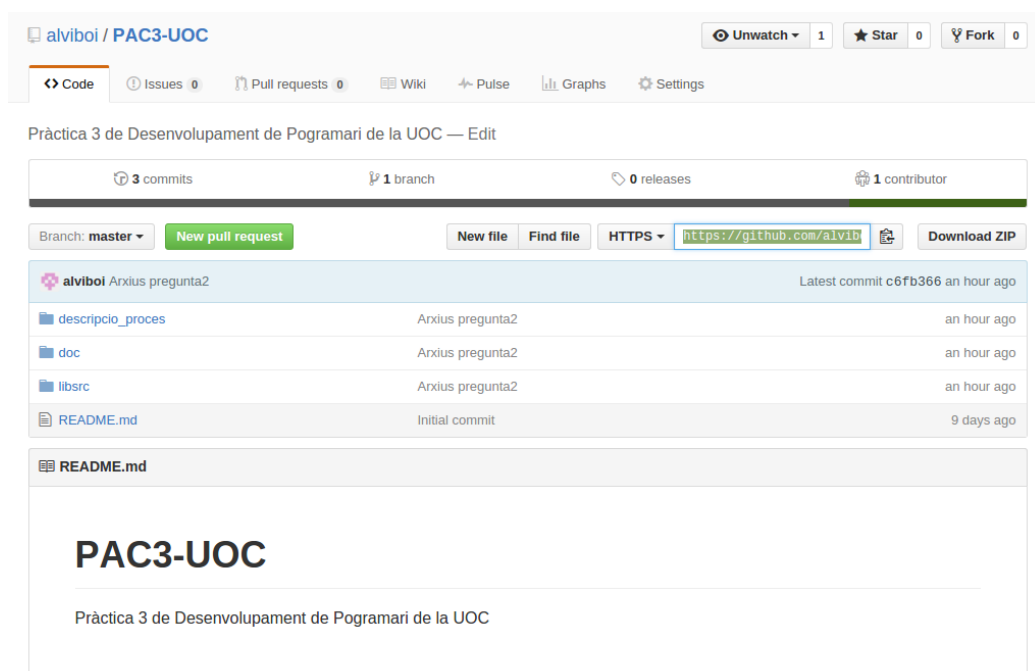
Com poden veure en aquest cas si que ens demanarà el nostre nom d'usuari i la nostra constrasenya, principalment per a poder portar un control de qui modifica cada arxiu. Així, podrem aconseguir una còpia al nostre ordinador local sempre amb el comandament:

*git clone https://github.com/alviboi/PAC3-UOC.git*

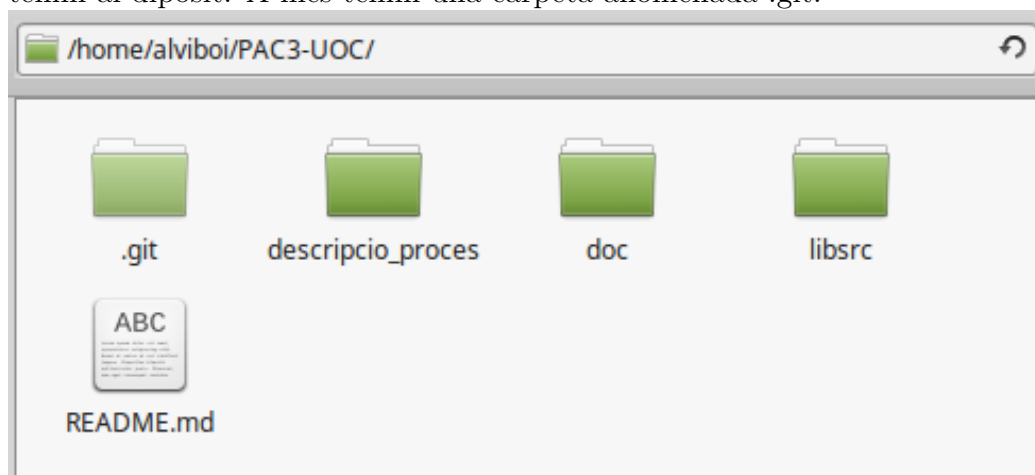
De manera que sempre tindrem accessible el nostre projecte des de qualsevol ordinador. Podem veure com quedaria la pàgina del nostre projecte a github<sup>2</sup>.

---

<sup>2</sup>Podem veure que els arxius que tenim són de les diferents preguntes que hem anat tractant, no només d'aquesta



Al nostre ordinador podem veure com tenim una còpia de tots els arxius que tenim al dipòsit. A més tenim una carpeta anomenada `.git`.



### 3 Documentació

Per a la part de documentació hem utilitzat etiquetes doxygen en els comentaris que hem fet a cadascun dels arxius. Com a exemple mostrem l'encapçalament que hem fet per a l'arxiu `filtres.c`:

```
/**
@file filtres.c
@brief Codi font de diferents funcions que serveixen per filtrar un text
donat com a paràmetre d'entrada en format .txt.
Les funcions són:
- void fs_head( int fd ) Filtre que mostra les tres primeres línies de l'arxiu i
ho mostra per pantalla. Les línies estan separades pel caràcter salt de línia.
- void fs_wc( int fd ) Filtre que conta el nombre de caràcters, paraules i línies
de l'arxiu d'entrada i ho mostra per pantalla.
- void fs_nl( int fd ) Filtre que numera les línies de l'entrada i ho trau per
pantalla.
- void fs_cut( int fd, int col ) Filtre que mostra la paraula a la posició col de
cada línia, la primera paraula de cada línia està a la posició 1, no 0.
@author Alfredo Rafael Vicente Boix i Eduardo César Galobardes
@date 14/12/2015
*/
```

Com podem veure el que hem fer és utilitzar les etiquetes doxygen que ens permeten especificar què es fa en cada punt, així per exemple:

<i>@file</i>	Ens permet especificar quin fitxer estem tractant
<i>@brief</i>	Ens permet fer una breu descripció de l'arxiu
<i>@author</i>	Ací indiquem l'autor i/o autors de l'arxiu
<i>@date</i>	I finalment la data en que s'ha creat el document

De manera semblant en cadascuna de les funcions fem una xicoteta explicació de quin és el seu propòsit tot especificant.



```

/**
@brief Funció que imprimix per pantalla el número donat en el paràmetre
d'entrada

Amb aquesta funció fem la reserva de memòria necessària per a posar en
memòria els enters passats en el paràmetre d'entrada que s'alliberarà poste-
riorment. Després es crea una cadena de tipus "1. " que es passa per a ser
escrita per pantalla. Si la funció presenta cap error es deixa d'executar i no
s'allibera la memòria. Si es desitja modificar aquest punt es pot utilitzar la
següent modificació:

\verbatim
if(write( 1, nstr, strlen( nstr ) ) == -1){free (nstr); exit;}
\endverbatim

@param num enter proporcionat per a ser escrit per pantalla.

@returns void la funció no torna cap valor

*/

```

Així les etiquetes que hem utilitzat per a documentar cadascuna de les funcions han sigut les seüents:

<i>@file</i>	Ens permet fer una breu descripció de la funció.
<i>/verbatim</i>	Ens permet incrustar codi per especificar alguna cosa.
<i>@param</i>	Ací expliquem els paràmetres d'entrada que rep la funció.
<i>@returns</i>	Ací els paràmetres de sortida, que en aquest cas és void.

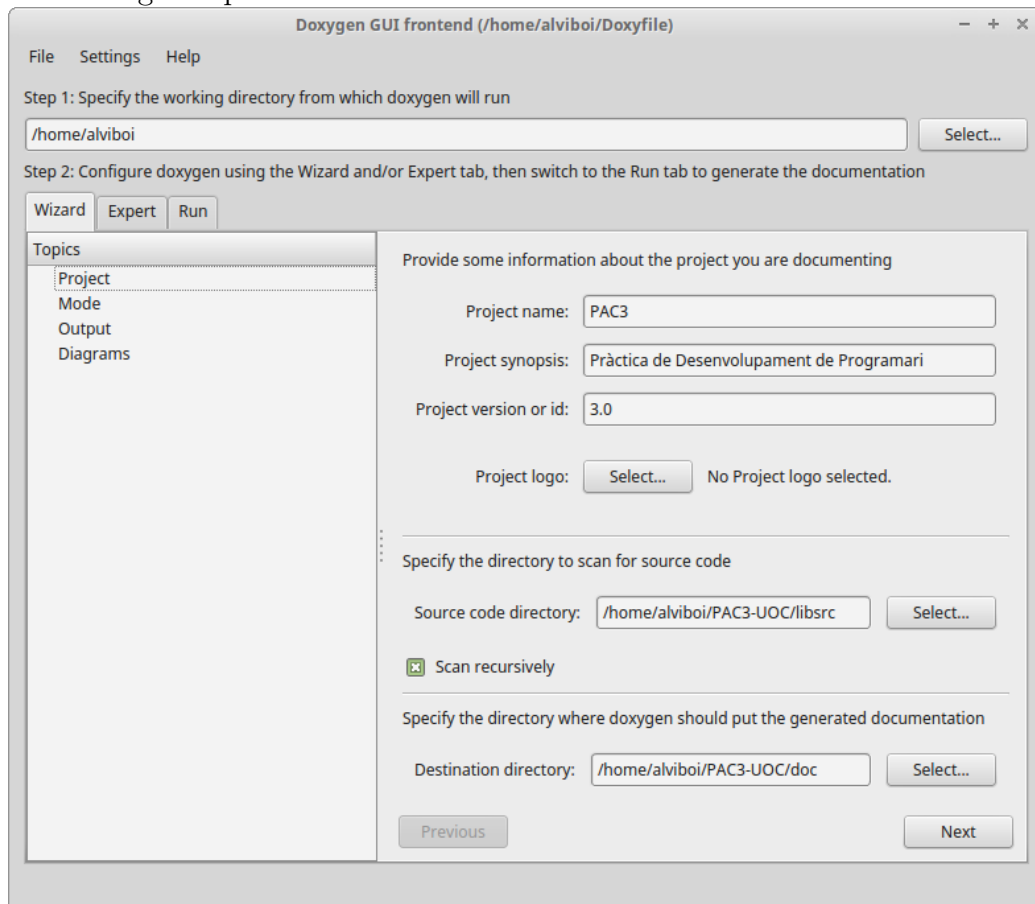
Com podem veure hi ha una apart de text que no està precedida per cap etiqueta, aquesta part és on anem a explicar detalladament quina és la finalitat de cada funció.

Una vegada tenim detallades totes les funcions i ambdós arxius, per a poder realitzar tota la documentació en format html, fer els enllaços a les altres operacions utilitzarem *doxywizard*, na ferramenta que, a través d'un tutorial i especificant-li uns paràmetres ens permet generar tota a informació necessària. Fins ara l'única cosa que hem fet és afegir etiquetes d'oxygen per a fer les explicacions, per a poder generar tota la informació, en primer lloc,

haurem d'instal·lar els programes que anem a utilitzar, tant doxywizard com doxygen, per això executem:

```
sudo apt-get install doxygen doxygen-gui
```

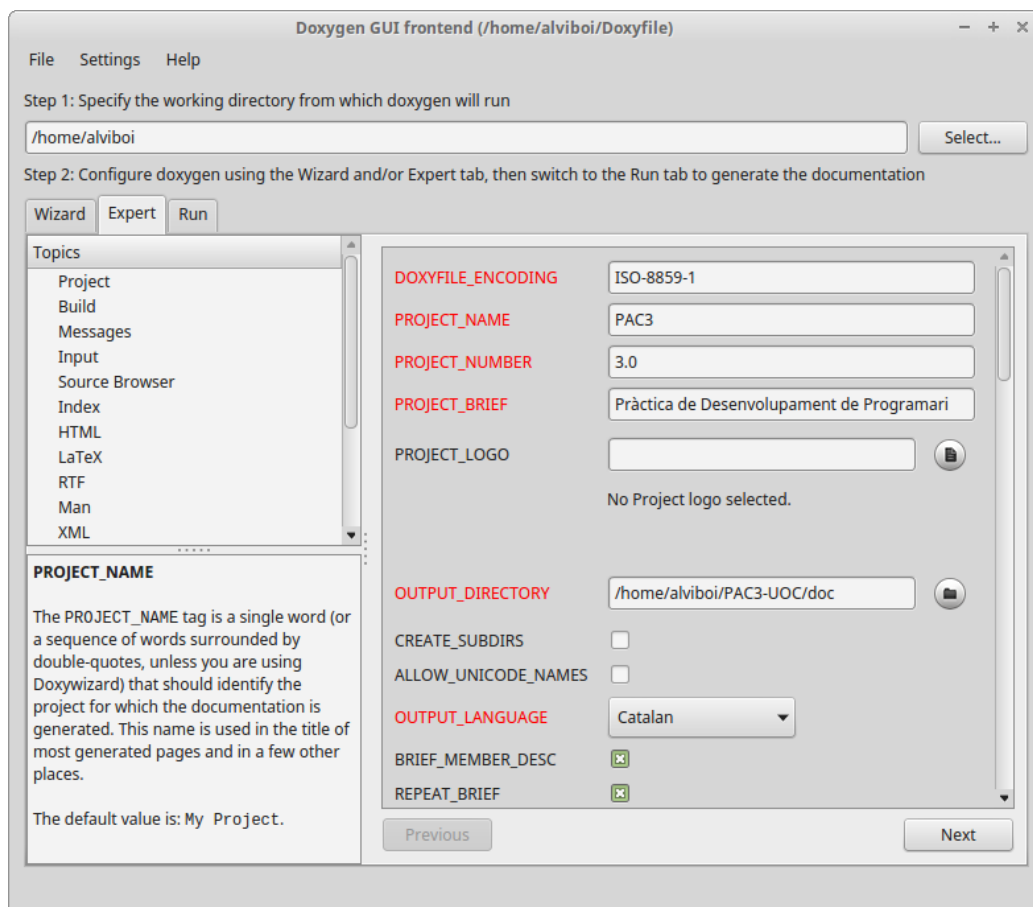
Per a poder fer ús del programa, en primer lloc hem d'instal·lar i ens trobem amb la següent pantalla:



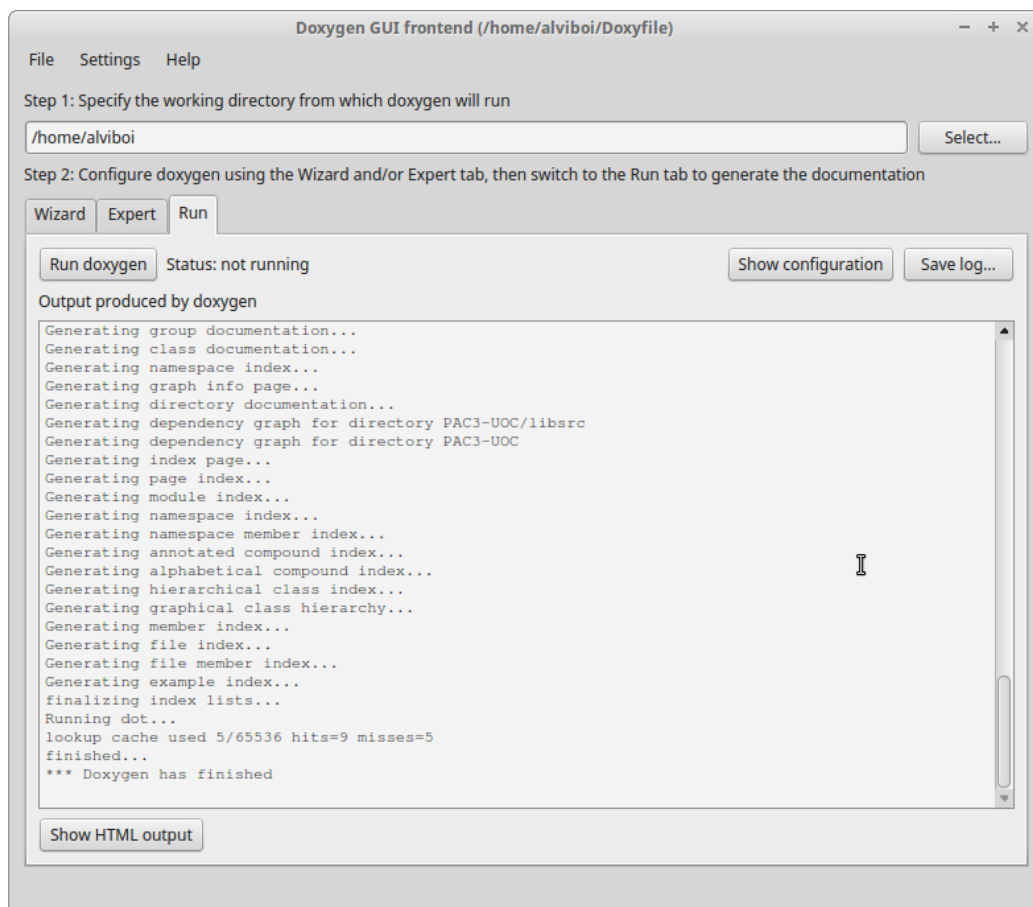
Podem veure com ja incloem el nom del projecte, el directorio on anem a traure el codi i les etiquetes doxygen, que és el qui es troba al nostre dipòsit<sup>3</sup>. A més especifiquem on volem que extraiga la documentació en format html. A banda d'aquestes coses, hem hagut de modificar uns paràmetres a la pestanya Expert de doxywizard:

---

<sup>3</sup>dintre de la carpeta libsrc



Hem hagut de modificar el paràmetre DOXYFILE\_ENCODING i el paràmetre OUTPUT\_LANGUAGE per a que ens puga reconèixer els accents del català i per a especificar en quina llengua està escrita la documentació. Una vegada canviats els paràmetres ja només ens queda generar la documentació:



Hem de recordar que tot i que la documentació ha sigut generada dins de la carpeta, dins del nostre dipòsit, fins que no fem un git add i després un git commit... els arxius no es trobaran al dipòsit. Per a poder accedir a la documentació creada anem al directori de treball i podem veure l'arxiu index.html, podem veure que el programa ha creat automàticament tots els enllaços i explicacions, formatant la documentació per a que quedi d'una manera molt més visual i agradable:

## PAC3 3.0

Pràctica de Desenvolupament de Programari

Pàgina principal	<b>Filtres</b>
Llista dels Filtres	Membres de Filtres
PAC3-UOC	lib-9rc

### Referència del Fitxer `filtres.c`

Codi font de diferents funcions que serveixen per filtrar un text donat com a paràmetre d'entrada en format `.txt`. [Més...](#)

```
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include "filtres.h"
```

Inclou el graf de dependències per a `filtres.c`:



[Veure el codi d'aquest fitxer.](#)

### Funcions

```
void printstr (int num)
    Funció que imprimeix per pantalla el número donat en el paràmetre d'entrada. Més...

void fs_head (int fd)
    Funció que imprimeix per pantalla les tres primeres línies. Més...

void fs_wc (int fd)
    Funció que conta el nombre de caràcters, paraules i línies. Més...

void fs_nl (int fd)
    Filtra que numera les línies de l'entrada i ho trau per pantalla. Més...

void fs_cut (int fd, int col)
```

## 4 Conclusions

Aquesta pràctica ha resultat molt didàctica, ja que hem tractat dos punts fonamentals sobre la creació de programari i que sovint oblidem, que són el control de versions i la documentació que deu acompanyar a qualsevol projecte. En el nostre cas ens hem decantat per github, encara que puga resultar l'opció més senzilla, ja que de cara al projecte final de màster és el repositori amb el que el meu tutor de l'empresa li agrada treballar, encara que han alguns punts com al creació de branches o fer un merge per a fer proves abans d'incorporar-ho a la rama principal. De tota manera hem creat també un repositori amb cvsd per a poder veure com es treballaria amb ell, ja que subversion l'hem tractat amb altres assignatures:

```
alviboi@MEU /etc/xinetd.d $ cd /var/lib/cvsd/
alviboi@MEU /var/lib/cvsd $ sudo mkdir PAC3
alviboi@MEU /var/lib/cvsd $ sudo chown cvsd:cvsd PAC3
alviboi@MEU /var/lib/cvsd $ sudo cvsd-buildroot /var/lib/cvsd
creating directory structure under /var/lib/cvsd... done.
installing binaries... cvs.
looking for non-linked system libraries... done.
installing libraries... done.
creating /var/lib/cvsd/dev devices... already there.
adding users to /var/lib/cvsd/etc/passwd....
fixing ownership... done.
chrooted system created in /var/lib/cvsd
if your cvs binary changes (new version) you should rerun cvsd-buildroot
alviboi@MEU /var/lib/cvsd $ sudo cvs -d /var/lib/cvsd/PAC3/ init
alviboi@MEU /var/lib/cvsd $ sudo cvsd-passwd /var/lib/cvsd/PAC3/ alviboi
/usr/sbin/cvsd-passwd: adding user 'alviboi' to '/var/lib/cvsd/PAC3/CVSR00T/passwd'
Enter new password:
Retype new password:
alviboi@MEU /var/lib/cvsd $ sudo cvsd-passwd /var/lib/cvsd/PAC3/ professor
/usr/sbin/cvsd-passwd: adding user 'professor' to '/var/lib/cvsd/PAC3/CVSR00T/passwd'
Enter new password:
Retype new password:
alviboi@MEU /var/lib/cvsd $ cvs -d :pserver:alviboi@localhost:/PAC3 login
Logging in to :pserver:alviboi@localhost:2401/PAC3
CVS password:
alviboi@MEU /var/lib/cvsd $
```

La part de documentació utilitzant doxygen, que és una ferramenta que no coneixia, m'ha semblat molt adequada encara que s'han hagut d'instal·lar alguns paquets addicionals per a que funcionara correctament. Hem de dir que les ferramentes utilitzades són ferramentes bastant testades i utilitzades i tenen una forta presència en el món de la programació. Per això no hem

trobat grans problemes a l'hora de treballar amb elles, i anat tot bastant rodat. Probablement les majors dificultat les hem trobat generant el document  $\text{\LaTeX}$  que no coneixia. En realitat, al començar a redactar el document, resultava un tant enfarragós, però hem utilitzat el TeXstudio, que tot i que costa acostumar-te un poc al seu funcionament, una vegada agafes un poc de soltura, resulta una ferramenta mot útil i quasi imprescindible per a redactar els documents de caràcter oficial.