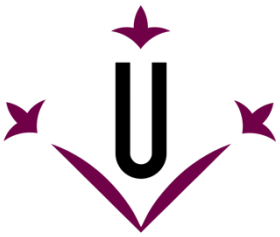


Universitat de Lleida

Escola Politècnica Superior

Grau en Enginyeria Informàtica

Xarxes

# Pràctica 1

## Programació d'aplicacions de xarxa

Serrano Ortega, Aniol

GPraLab 3

# Índex



## Índex de il·lustracions

# 1. Introducció

En aquesta pràctica s'ha programat un protocol de comunicació de xarxes amb *sockets*. Per a implementar aquest protocol s'ha usat l'estructura client-servidor, on el client s'ha implementat amb C i el servidor en *python* 3. Tant el client com el servidor consten de dues fases, la fase de registre i la fase de manteniment de comunicació.

El desenvolupament d'aquesta pràctica ajuda a conèixer les llibreries d'aplicacions de xarxes i permet entendre com funciona la implementació d'un protocol en concret. L'objectiu d'aquesta pràctica és aconseguir que el client es registri i mantingui una comunicació periòdica amb el servidor. D'altra banda, el servidor ha de processar aquestes peticions i acceptar-les o rebutjar-les en funció de diferents paràmetres.

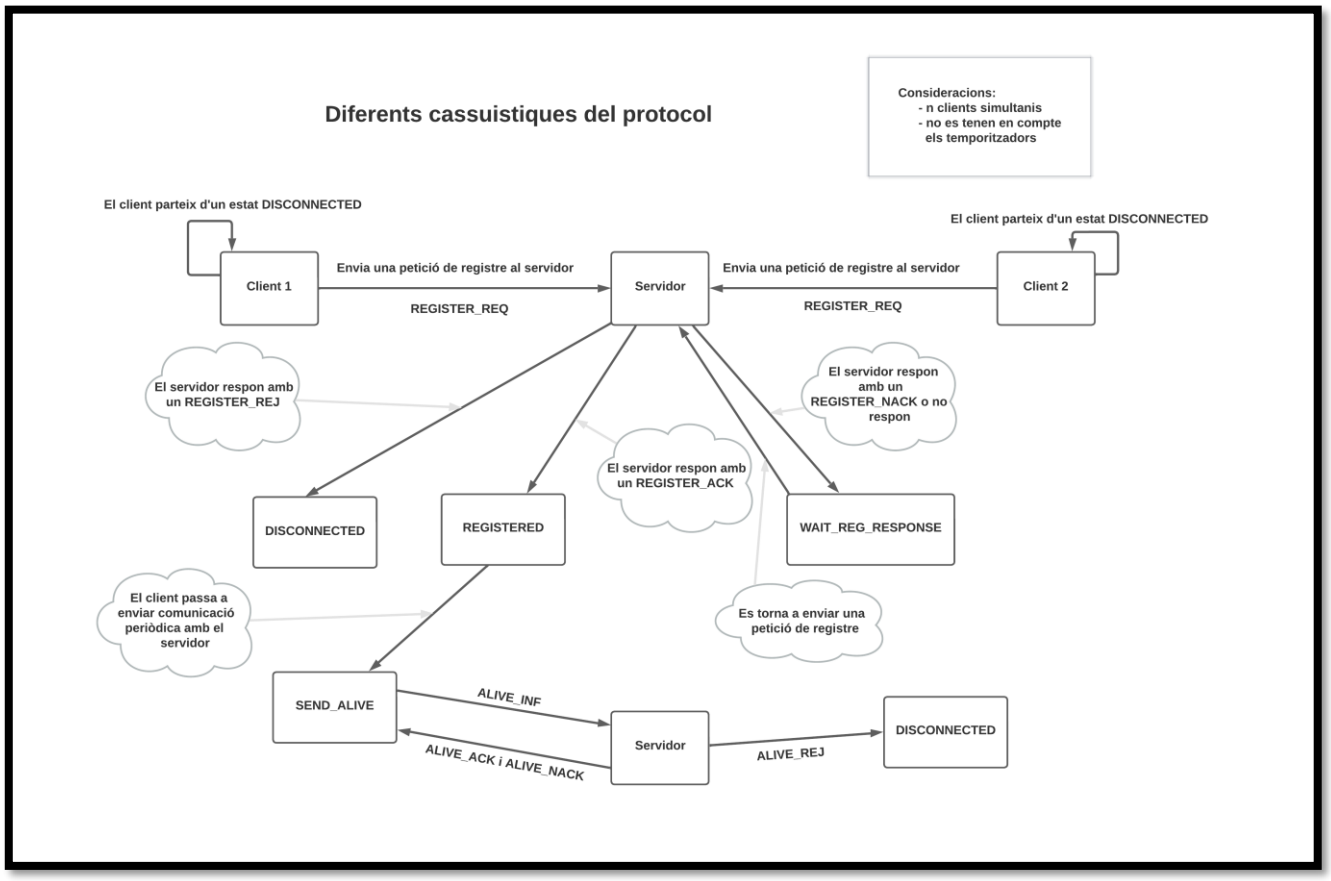
# 2. Desenvolupament

Per a la implementació d'aquest protocol només s'han fet les dues primeres fases, és a dir, la fase de registre i la fase de comunicació periòdica en UDP. Tot i això, els codis tant del client com del servidor són escalables per tal d'escalar-se en un futur i implementar les altres fases en TCP.

## 2.1. Implementació del client

Pel que respecta al client, aquest s'ha estructurat d'una forma lògica en el que segons succeeixen els esdeveniments es van trucant a les funcions que es requereixen. Les diferents casuístiques s'han dissenyat utilitzant el disseny descendent amb funcions per a poder separar cada cas en una funció més petita i manejable.

Durant l'execució del programa el client passa per una sèrie d'estats que van canviant en funció de la resposta del servidor. Per entendre les diferents fases per les quals passa el client en aquest procés, s'ha dissenyat el següent diagrama de blocs dels diferents estats del client segons la resposta del servidor.



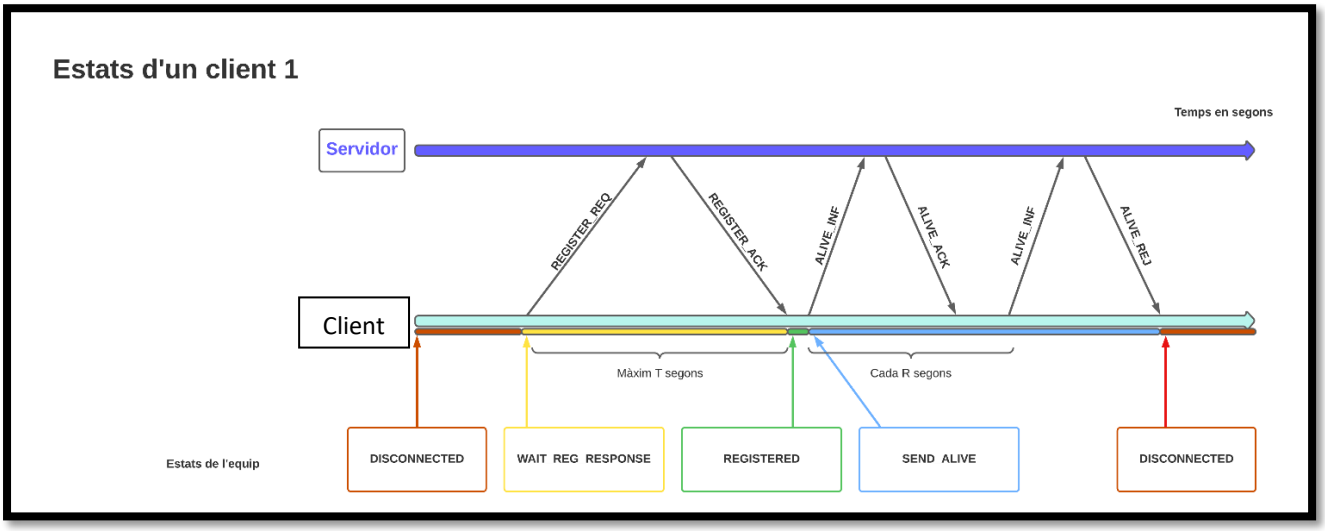

*Il·lustració 1: Diagrama de blocs dels diferents estats del client*

D'altra banda, per a mantenir la comunicació periòdica en el client, s'ha tractat en una funció anomenada *void send_alives()* on periòdicament s'envia un paquet UDP [ALIVE_INF] al servidor utilitzant la funció *sendto()*. Aquest paquet conté l'identificador MAC, l'adreça IP i el nombre aleatori de l'equip.

El temps d'enviament entre cada paquet es controla mitjançant una variable R en segons. Aquesta variable representa el temps entre cada enviament i es defineix al principi del programa. Per a assegurar-se que el servidor ha rebut correctament el paquet, es comprova si efectivament s'ha rebut resposta del servidor en un temps determinat. En cas que no s'hagi rebut res, el client s'espera un ALIVE_ACK i es continua amb el procés d'enviament

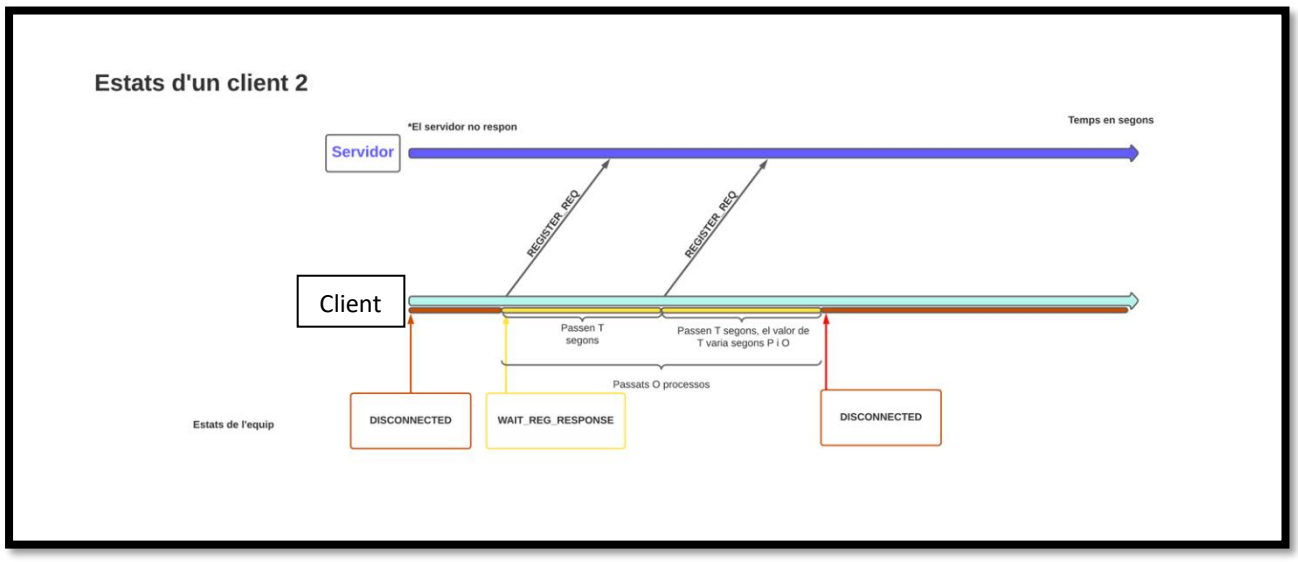
Si transcorre el nombre determinat per la variable U_2 segons sense haver rebut cap resposta del servidor, es considera que hi ha hagut una pèrdua de la comunicació i es reinicia el procés de registre. Aquest nou procés de registre s'efectua mitjançant una petició de registre REGISTER_REQ al servidor.

Per a veure esquemàticament el protocol de registre i manteniment de comunicacions entre el client i el servidor, s'han dissenyat aquests dos diagrames que contemplen en general la majoria dels casos:




*Il·lustració 2: Diagrama d'una comunicació del client (1)*

En aquest primer diagrama es pot veure el cas entre una comunicació normal entre el client i el servidor des del punt de vista del client. Primerament, s'efectua la fase de registre i seguidament una comunicació periòdica. Finalment, el client passa a l'estat



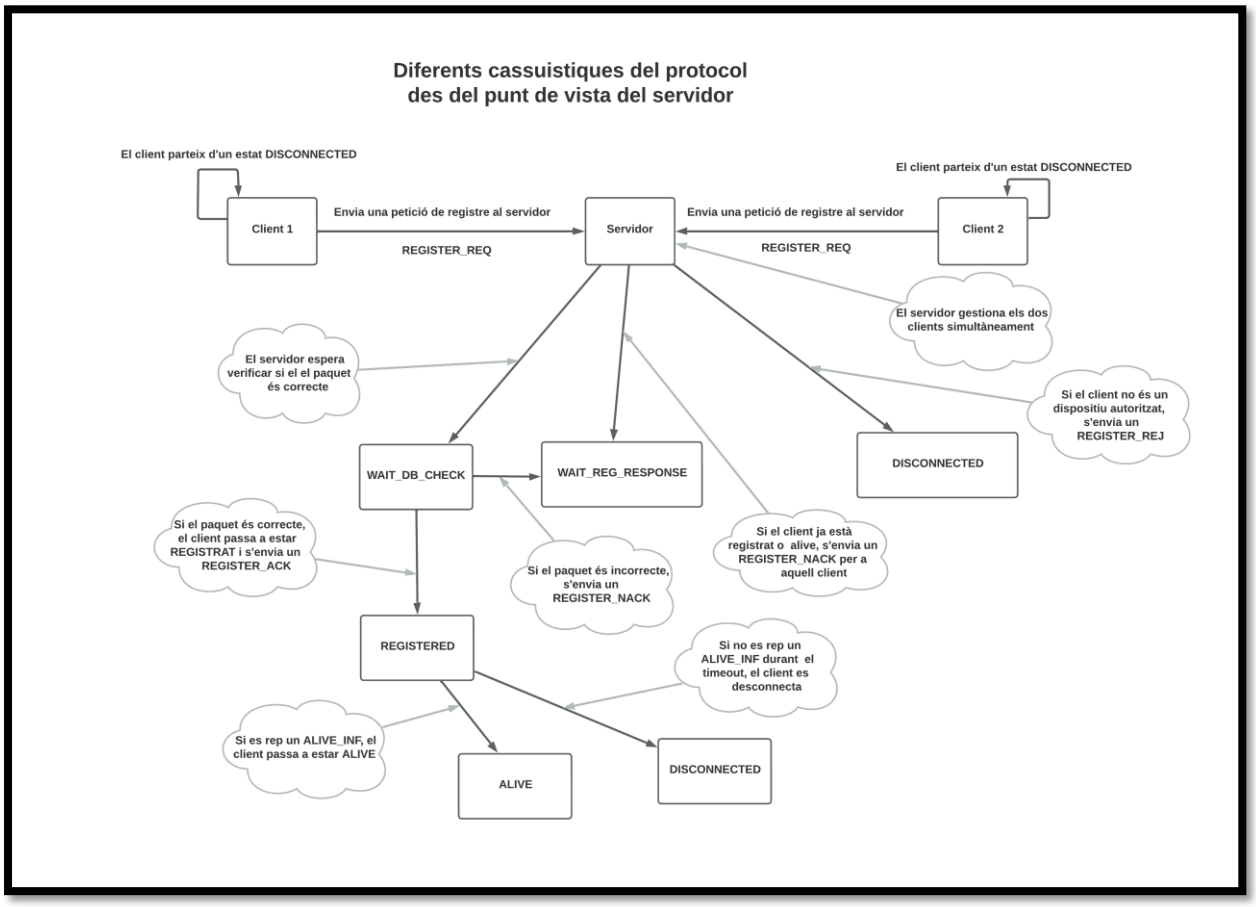

*Il·lustració 3: Diagrama d'una comunicació del client (2)*

En aquest segon diagrama es pot observar una comunicació en el cas en el que el servidor no està operatiu. En aquest cas el client no es pot registrar i passats certs paràmetres el client es desconnecta perquè suposa que el servidor no esta operatiu.

## 2.2. Implementació del servidor

Pel que fa el servidor, aquest s'ha estructurat d'una forma en la que les peticions es tracten per separat per cada equip. Les funcions per tractar les peticions de cada equip es fa en un fil, en canvi el programa principal queda a l'espera de noves comandes per terminal. També hi ha un tercer fil per a l'espera de *timeouts* del client en l'estat ALIVE.

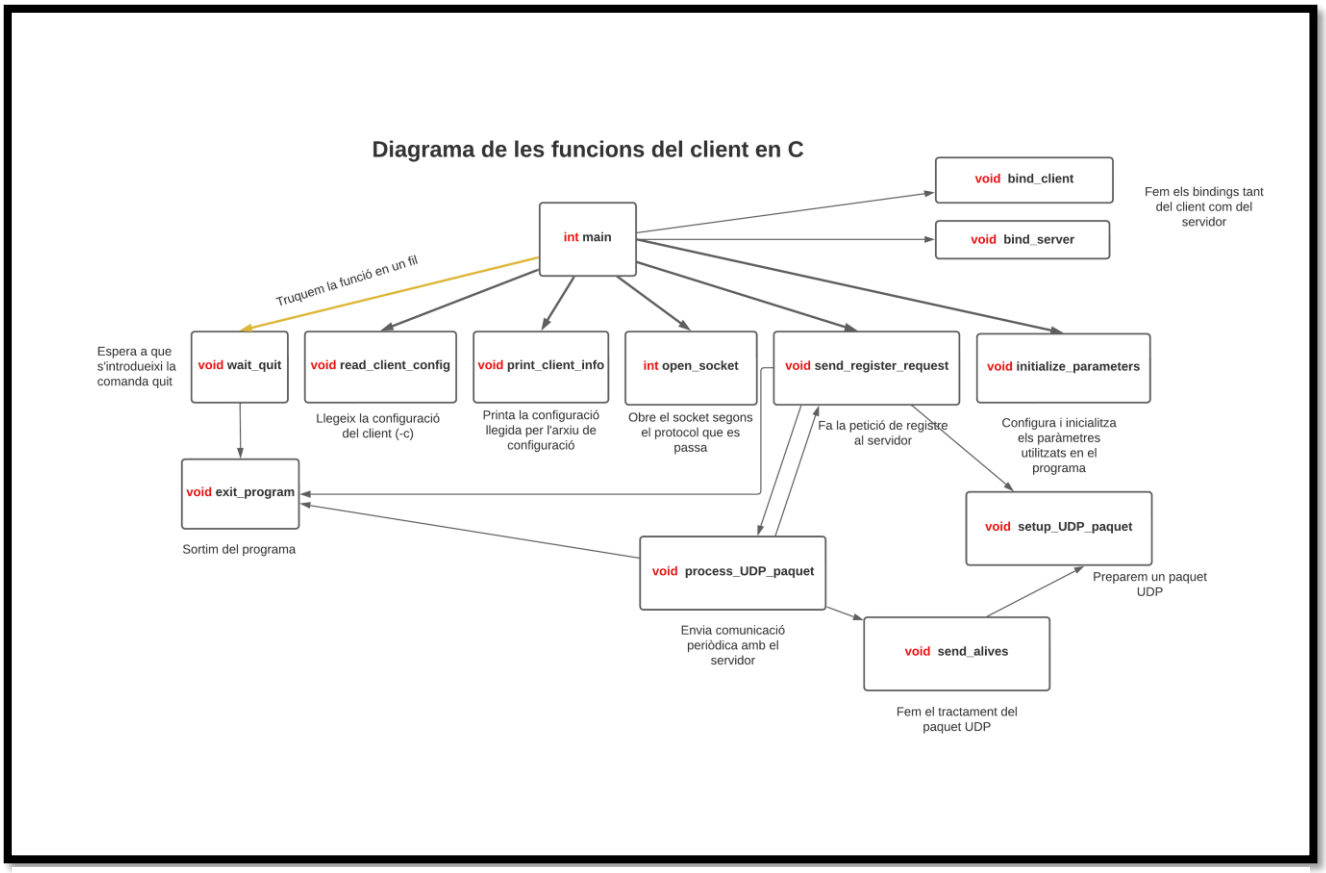
Durant l'execució del programa el servidor modifica els estats de cada client per separat en funció dels esdeveniments que succeeixen. Per entendre els diferents estats que passa cada client, s'ha dissenyat el següent diagrama de blocs.



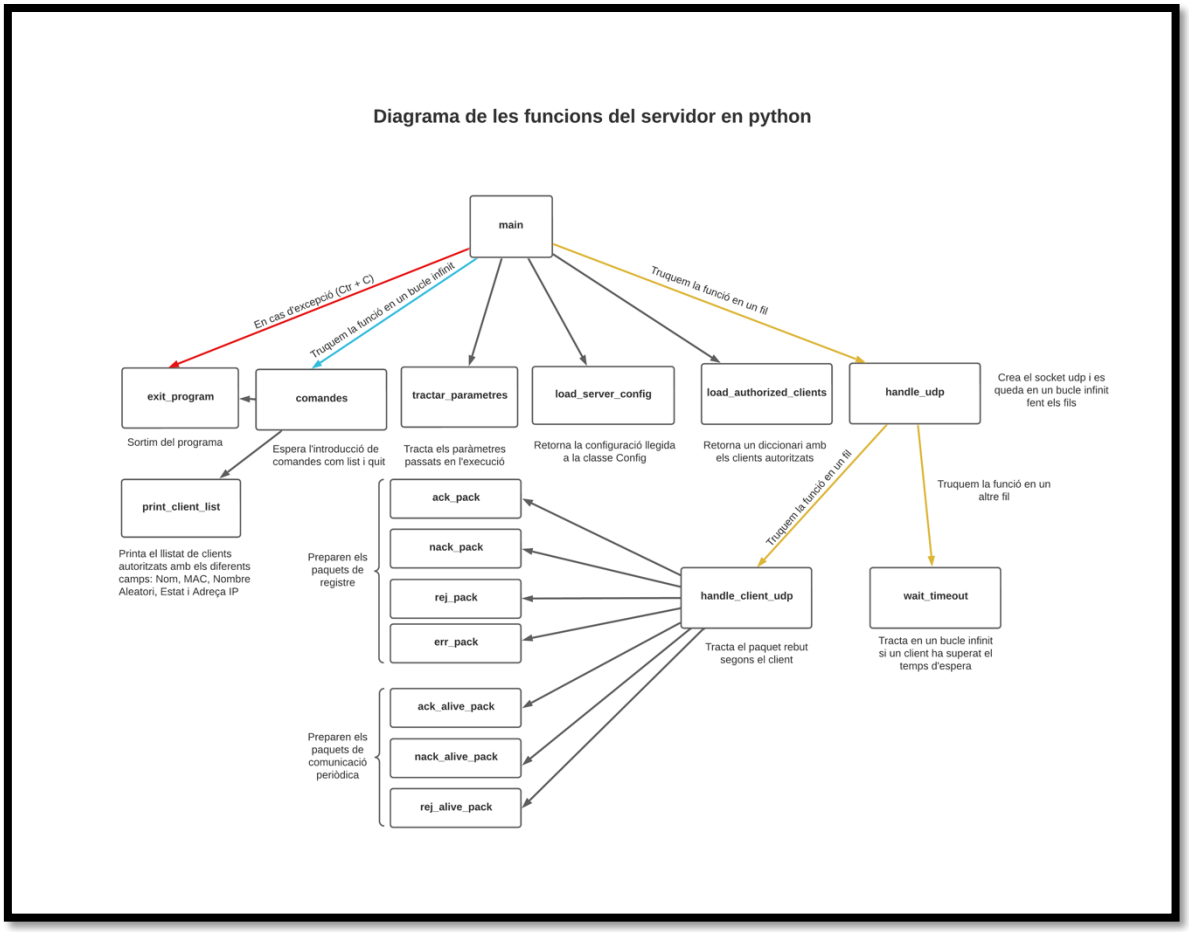

*Il·lustració 4: Diagrama de blocs dels diferents estats de cada client en el servidor*

## 2.3. Diagrames de les funcions del client i servidor

Per a seguir adequadament el codi, és adient seguir el següent diagrama de les diferents funcions tant del client com del servidor. Amb aquest diagrama es pot veure la lògica de les funcions dels programes i entendre com aquests estan estructurats.




*Il·lustració 5: Diagrama de les funcions del client en C*




*Il·lustració 6: Diagrama de les funcions del servidor en python*

## 3. Conclusions

La realització d'aquesta pràctica ha comportat certs reptes que han hagut de ser superats, però el fet de resoldre aquest problemes ha millorat la comprensió de la programació d'una aplicació de xarxa i els aspectes teòrics que aquesta comporta. S'ha hagut de investigar diverses llibreries i certs procediments els que un estudiant de 2n no està molt familiaritzat.

D'altra banda, aquesta pràctica posa a prova la resiliència del autor sobre problemes complexos. Per resoldre aquests problemes és necessari dividir aquests problemes complexos en altres que no ho són tant, i en aquesta pràctica definitivament s'han complert aquests objectius.

Pel que respecta als resultats obtinguts, s'esperava una implementació més complerta, malgrat això els aspectes que s'han treballat s'han implementat amb coherència i dedicació. Finalment, val a dir que s'han pogut complir els objectius prèviament establerts i l'aprenentatge en aquesta pràctica ha estat ampli i divers.