M05UF2.- Optimització de Programari. Pràctica

[1.- Objectius 1](#_gjdgxs)

[2.-](#_30j0zll) Proves de Caixa Negra. JUnit 1

[3.-](#_1fob9te) Documentació amb Javadoc 4

[4.-](#_3znysh7) Control de Versions amb Git 6

[5.-](#_tyjcwt) Domjudge 8

[6.-](#_1t3h5sf) PROBLEMA PEL DOMJUDGE 8

# 1.- Objectius

Els objectius d’aquesta pràctica són:

* Conèixer i practicar les eines que permeten fer proves unitàries d’un programa o mètode (funció), així com entendre la seva importància.
* Conèixer i practicar les eines que permeten fer documentació d’un programa
* Conèixer i practicar les eines que permeten gestionar versions dels nostres programes, així com entendre la seva importància
* Continuar millorant com a programadors, desenvolupant un programa que servirà per poder aplicar tots els punts anteriors.

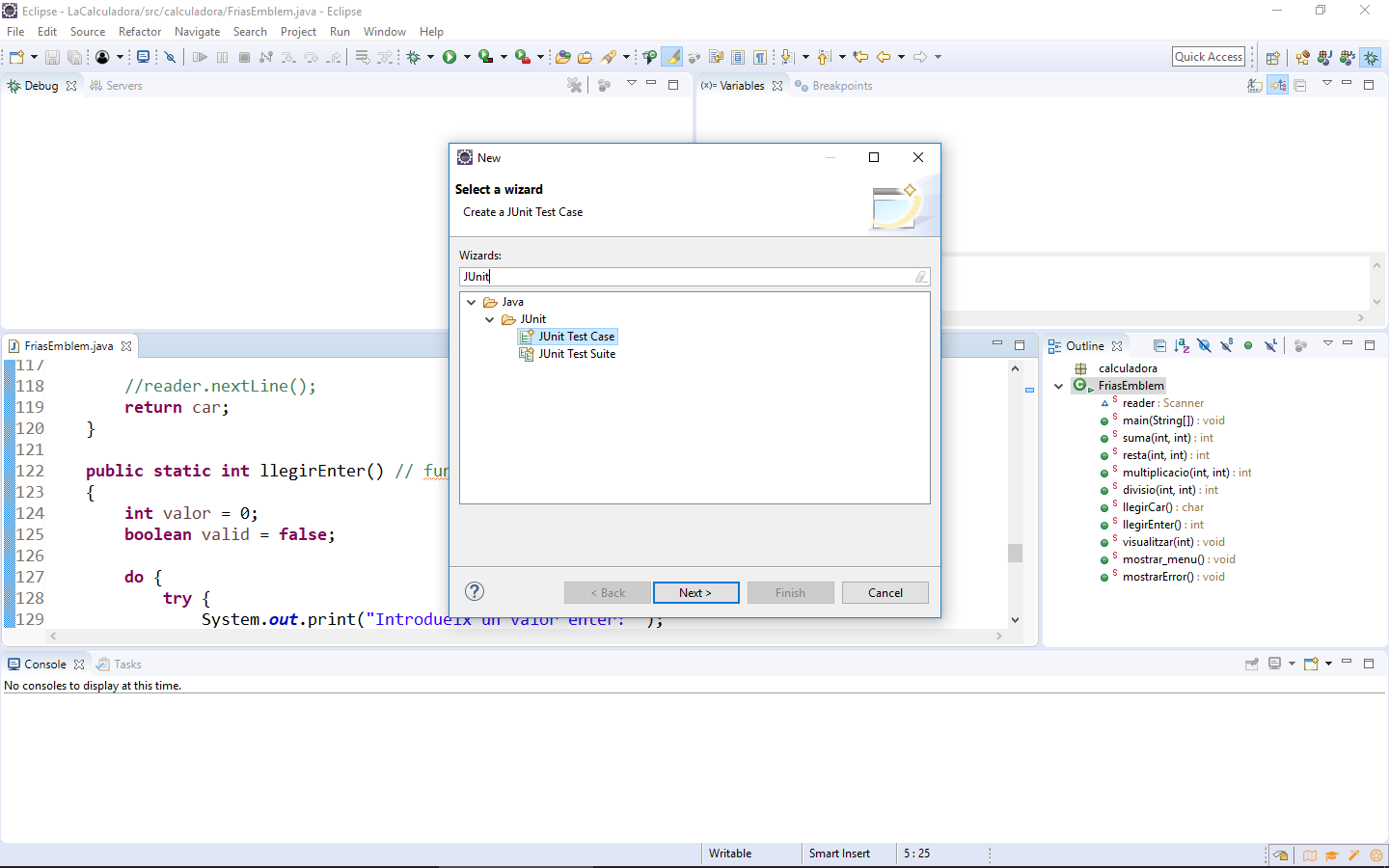
# 2.- Proves de Caixa Negra. JUnit

JUnit es una de les llibreries externes de Java més utilitzades, i serveix per generar i gestionar casos de prova dels nostres programes i funcions.

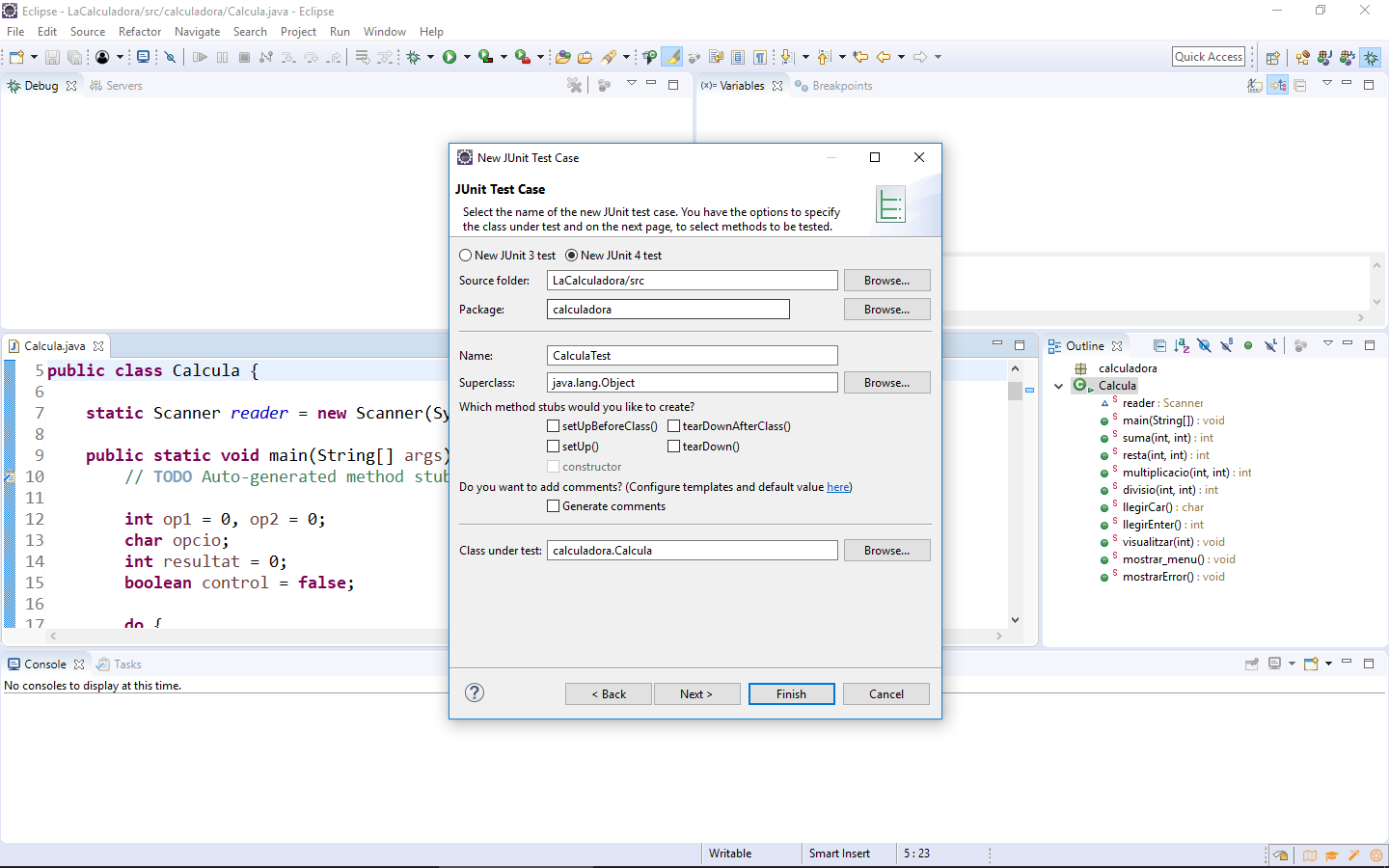
JUnit és un entorn de treball que permet executar les classes Java de manera controlada, de forma que es pot avaluar si el funcionament de cadascun dels mètodes de la classe es comporta com s'espera. La seva funció és, donat un valor d'entrada, s'analitza el valor de retorn que s'espera, si la classe acompleix amb l'especificació, aleshores JUnit indicarà que el mètode de la classe ha passat la prova; en cas que el valor esperat sigui diferent, JUnit indicarà una fallada en el mètode corresponent.  
  
Haurieu de tenir JUnit integrat a Eclipse. En cas contrari o nova instal.lació, seguiu les següents instruccions.

<https://www.tutorialspoint.com/junit/junit_plug_with_eclipse.htm>

Creeu amb Eclipse un nou cas de JUnit. Per fer-ho, aneu a New, com si volguessiu crear una nova classe, pero escolliu JUnit Test Case

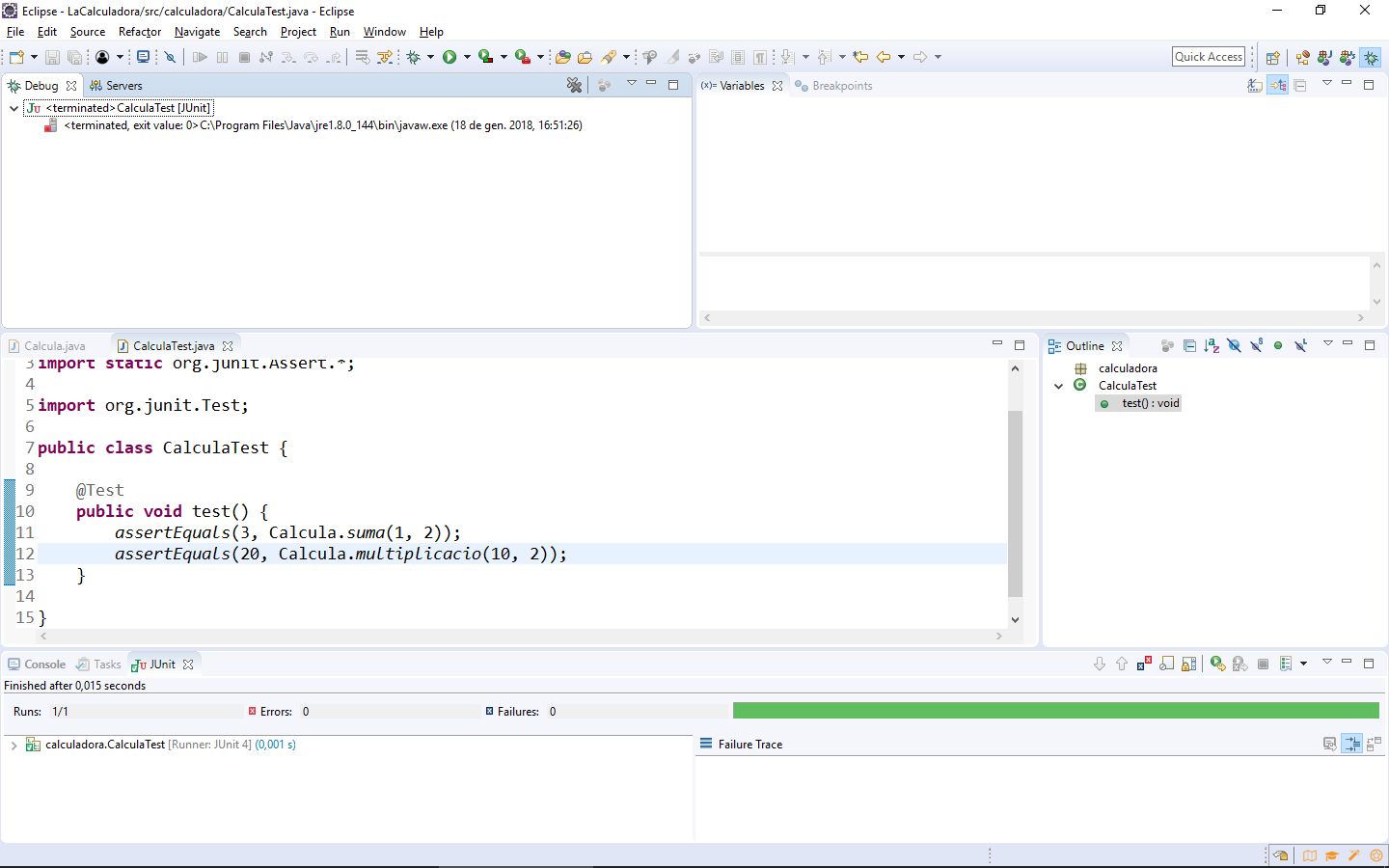


Assegureu-vos de crear el Test de forma que estigui a la carpeta de la classe que voleu testejar i que “Class under test” sigui aquesta.



Executa el Test. Comprova que fallarà per no estar implementat.

Canvia el fail per un parell de casos de prova senzills mitjançant assertEquals. Comprova que en aquest cas funcionarà.



Crea una classe Test que faci suficients jocs de prova per assegurar-te que la teva solució funciona perfectament.

Teniu un tutorial complet sobre JUnit aquí: <https://www.tutorialspoint.com/junit/index.htm>

Teniu una activitat específica de jUnit.

## **El meu cas de prova :**

## 

# 3.- Documentació amb Javadoc

Javadoc es una utilitat d’Oracle per a la generació de documentació d’APIs en format HTML a partir de codi Java. Javadoc es l’estàndard de la indústria per a documentar classes de Java.

Es poden trobar bons exemples de javadoc en la Wikipedia: <https://en.wikipedia.org/wiki/Javadoc>

**Copia aquest codi en una classe i genera el Javadoc per veure com queda.**

/\*\*  
 \* @author Marc Albareda <dm.albareda@ies-sabadell.cat> (nom i correu)  
 \* @version 1.0 (versió actual del programa)  
 \* @since 1.0 (La primera versió del programa  
 \*/

public class JavadocTest {  
 public static void main (String[] args){  
 int res = sumames(4,5);  
 System.out.println(res);  
 }

/\*\*  
 \* Suma els dos nombres i afegeix 1 (descripció curta) (1)  
 \* <p>  
 \* Suma els dos nombres proporcionats. Després de sumar-los (2)  
 \* afegeix 1 a la resposta. (descripció llarga, opcional)  
 \* <p>  
 \* Si voleu posar més explicació les podeu anar posant  
 \* Com veieu els tags html funcionen en javadoc  
 \*  
 \* @param int a Primer nombre a sumar(variable Descripcio) (3)  
 \* @param int b Segon nombre a sumar  
 \* @return Retorna la suma de les dues entrades més 1.  
 \*/

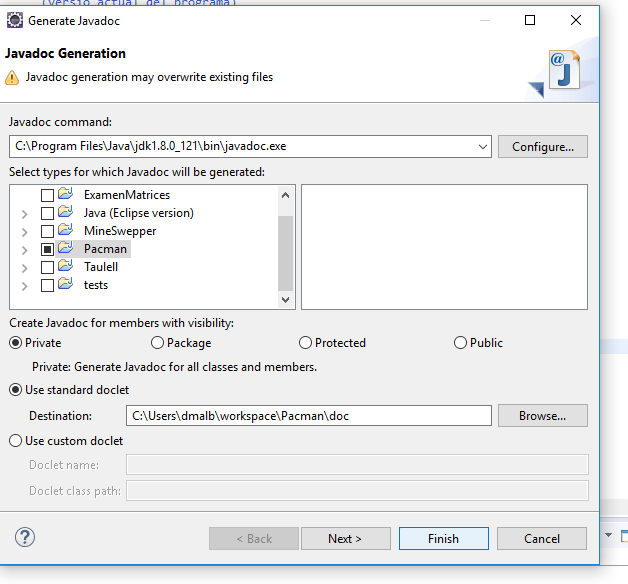
private static int sumames(int a, int b) {

int res = a +b;  
 res++;  
 return res;

}

}

En Eclipse el Javadoc es genera desde Project->Generate Javadoc



En Javadoc command haureu de buscar el javadoc.exe al vostre Windows Això es el que se sol conèixer com el path al programa. En Linux (Hola M1) hauríeu de posar la comanda de terminal per a accedir. Genereu el projecte que vulgueu fer, i les classes que voleu fer (si les voleu fer totes poseu private).

**El javadoc crearà un HTML de la vostra classe. Genereu el Javadoc i adjunteu el fitxer HTML.**

**Feu el Javadoc per el vostre projecte o projecte guiat Feu tant el principal com per a tots els mètodes. Genereu el HTML i adjunteu-lo.**

**El javadoc esta adjuntat en la carpeta entregada.**

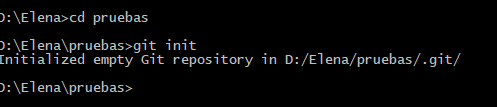
# 4.- Control de Versions amb Git

Farem el control de versions amb Git. Per a fer-ho, necessitarem instal·lar un plugin d’Eclipse anomenat EGit.

IMPORTANT: Abans d’instal·lar el plugin d’Eclipse, has d’assegurar-te que tens el git instal·lat al teu ordinador.

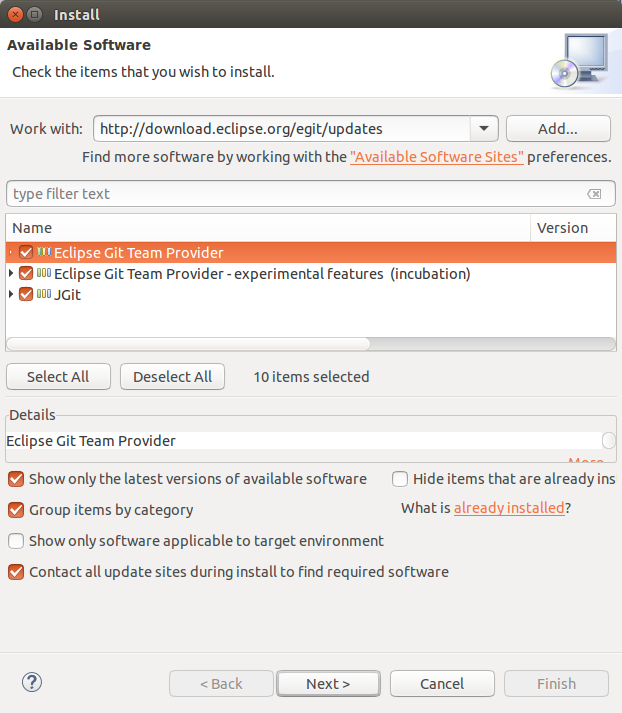
<https://git-scm.com/download/win>

i crea un repositori situant-te a la carpeta a on vulguis crear-lo (en terminal de comandes) i fent servir la comanda “git init”

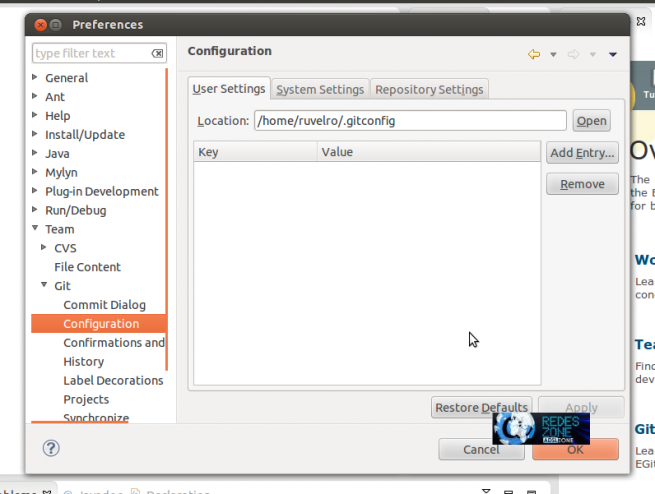


Els Plugins d’eclipse son components addicionals que se li poden afegir a l’eclipse per a afegir noves funcionalitats. Tots els Plugins funcionen amb un sistema de repositoris (se li proporciona una web a l’eclipse des de la que es baixa l’última versió del programa). Per fer això aneu a l’eclipse a Help->Install New Software i feu servir aquest repositori

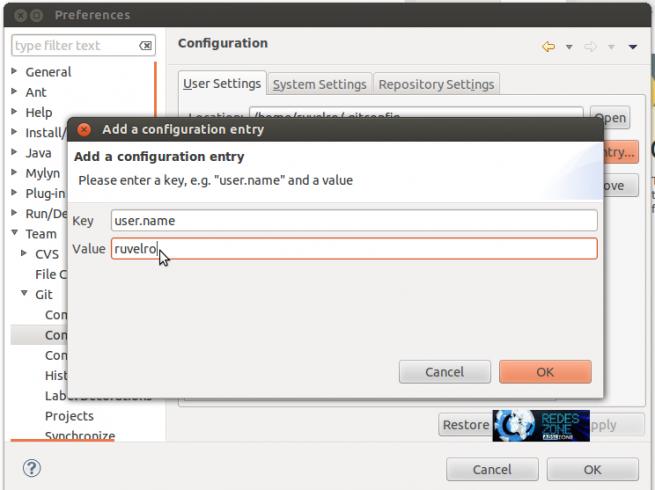
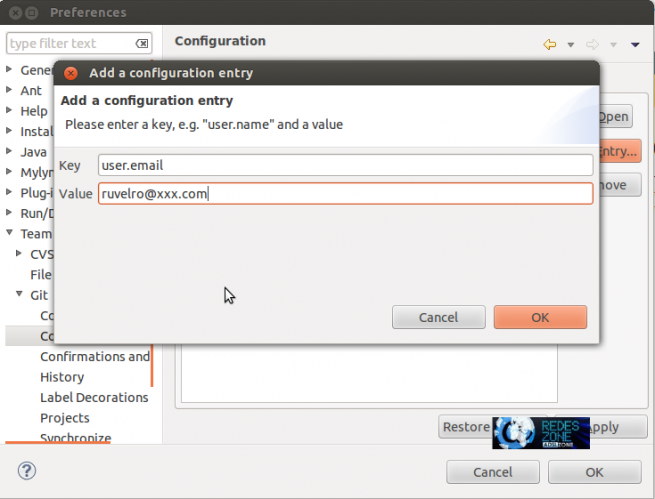
http://download.eclipse.org/egit/updates



Un cop reiniciat Eclipse hem de connectar-nos a la nostra compta de Git per a començar a utilitzar-lo. Es fa en el menú Window, **Preferences > Team > Git > Configuration**

****

Caldrà clickar a Add Entry i afegir els valors user.name i user.mail per identificar-nos amb el git



**Heu de mantenir el vostre projecte amb git, controlant le seves versions. Ho hauràs de fer amb les comandes necessàries.**

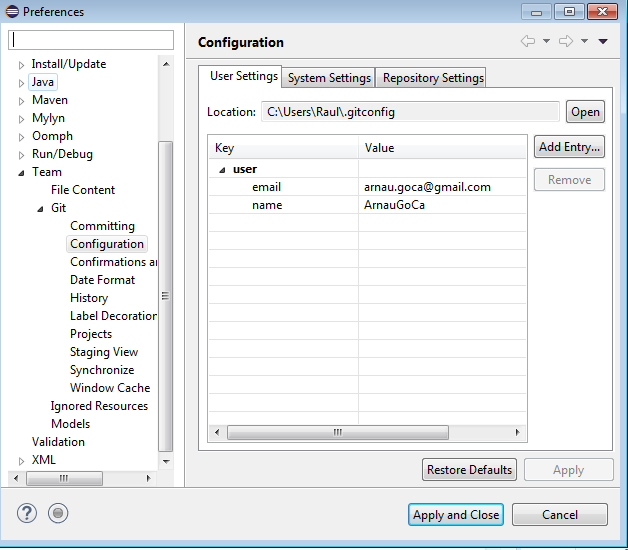
**Extra: Desplegament Web amb GitHub**

Llegeix-te aquest manual d’integració d’Eclipse amb GitHub (abans t’hauràs de crear una compta a GitHub)

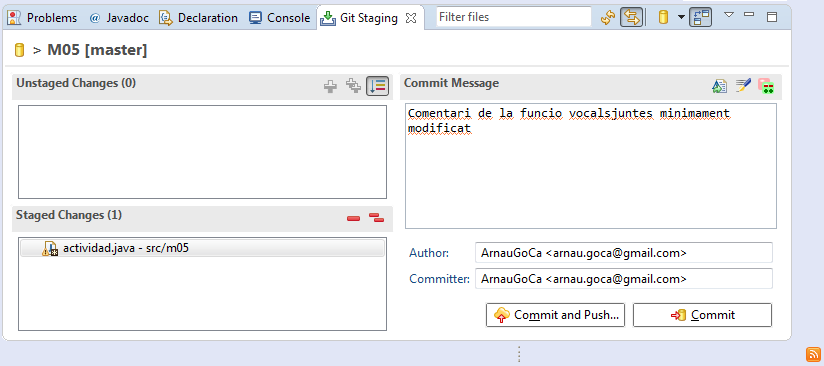
[**https://www.redeszone.net/2013/04/19/como-utilizar-github-para-sincronizar-procesos-de-eclipse-con-otros-usuarios%EF%BB%BF%EF%BB%BF/**](https://www.redeszone.net/2013/04/19/como-utilizar-github-para-sincronizar-procesos-de-eclipse-con-otros-usuarios%EF%BB%BF%EF%BB%BF/)

**Pots pujar el teu projecte al teu GitHub per a que tot el mon pugui gaudir d’ell.**

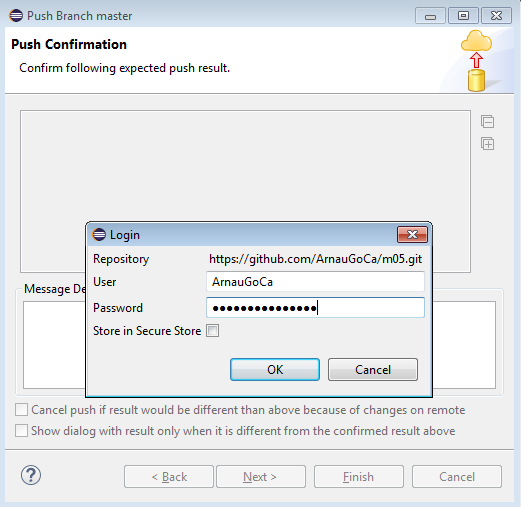
**Afegeixo el meu compte a l’eclipse:**



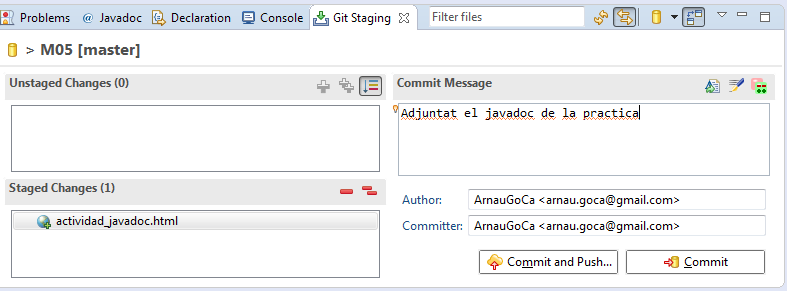
**Fer un commit del projecte amb tot actualitzat al repositori creat amb la nostra compta a GitHub:**



**El pugem al nostre repositori del Git amb la direcció de la nostra compta i conformem amb la nostra compta i contrasenya:**



**Per últim he afegit el javadoc de la practica al Git també:**



# 5.- Domjudge

Agafa la màquina virtual del programari per poder disposar de *domjudge* (192.168.0.151 i directori DAM\_M05UF2/domjudge)

nano /etc/network/interfaces

allow-hotplug enp0s3

iface enp0s3 inet static

address 192.168.1.77

netmask 255.255.255.0

gateway 192.168.1.1

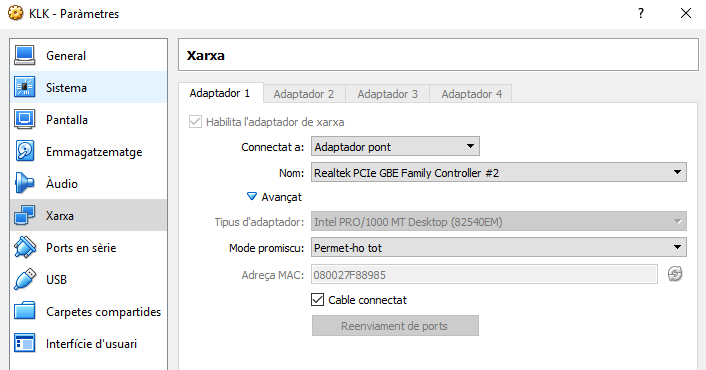
Arrenca-la amb *VirtualBox*, i inicia sessió amb usuari *admin,* password *domjudge*

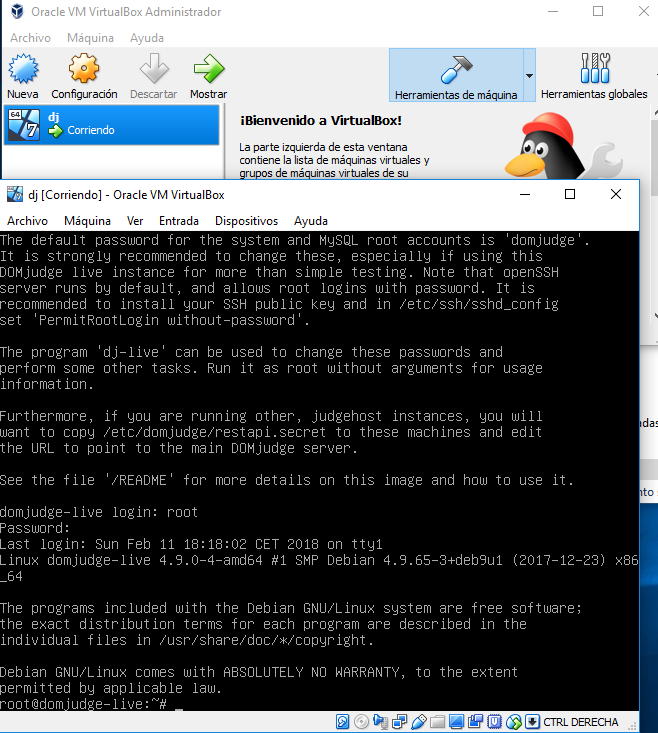
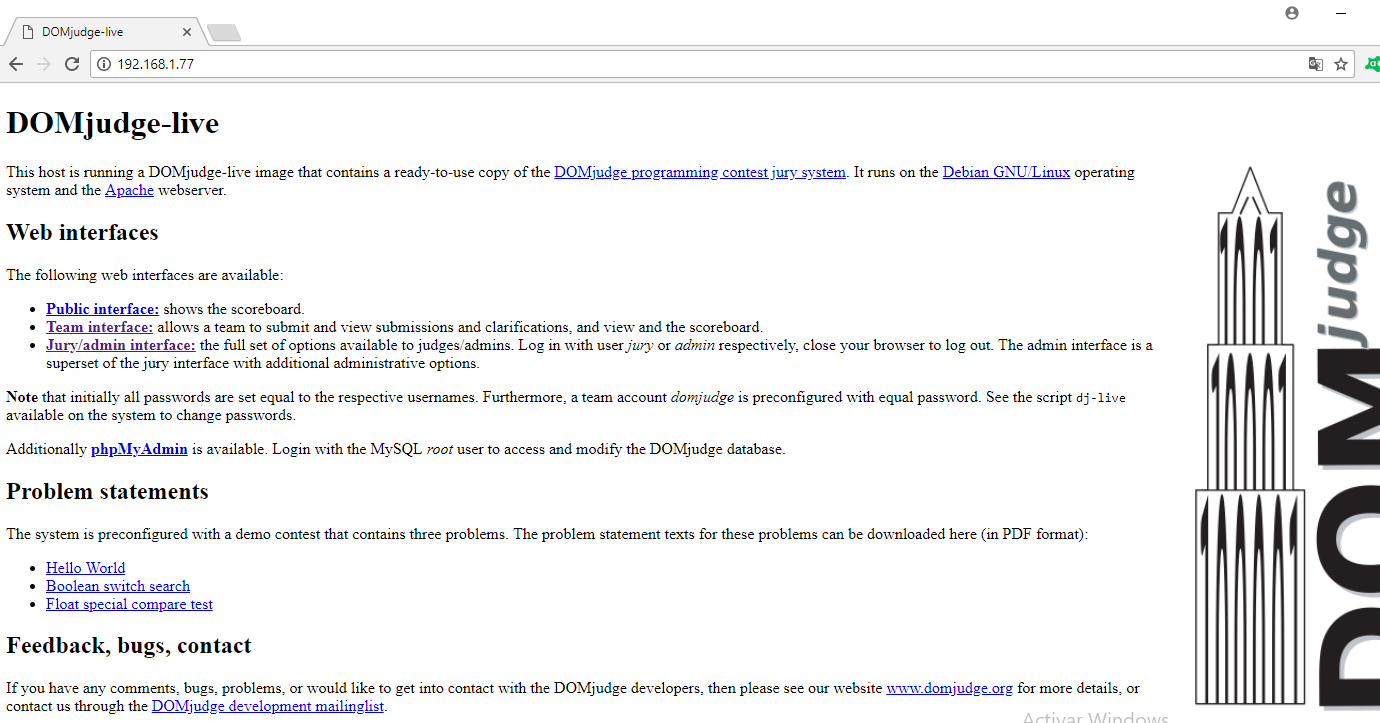
Si tot va be, pots obrir el navegador accedint a 192.168.1.77

Cas contrari, pots editar el fitxer /etc/network/interfaces i posar

o la address i gateway que necessitis (ex. 10.1.83.40 i 10.1.83.1)

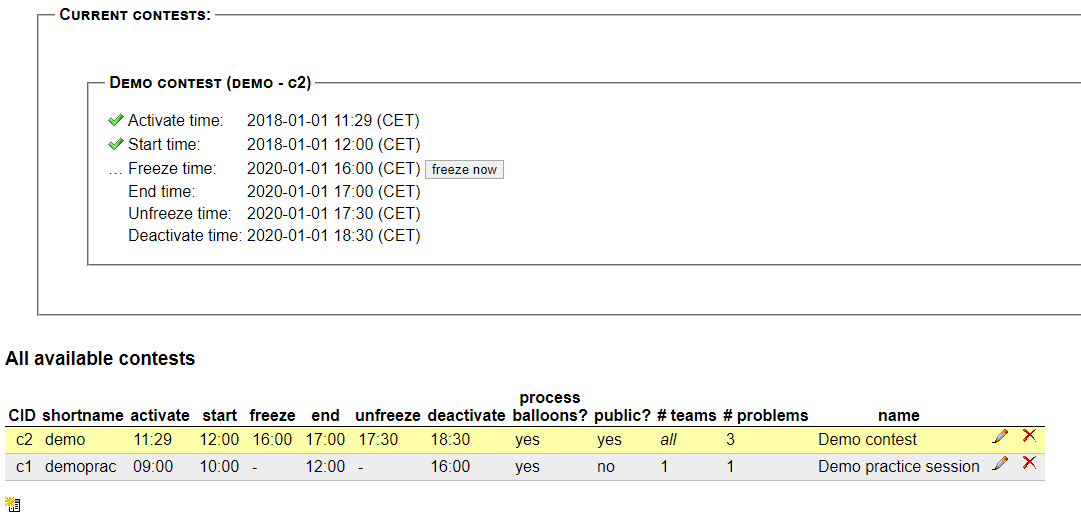
A la configuració de la màquina virtual enrecordat de posar *Adaptador Pont* i *Mode Promiscu*

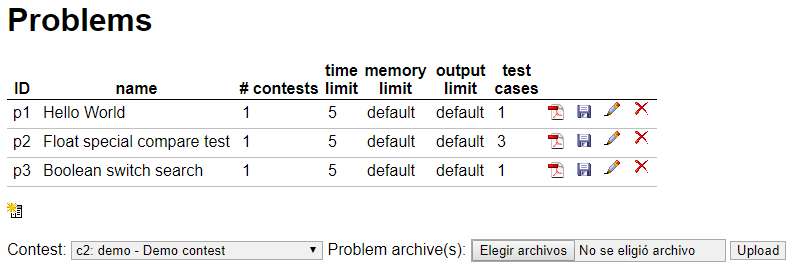




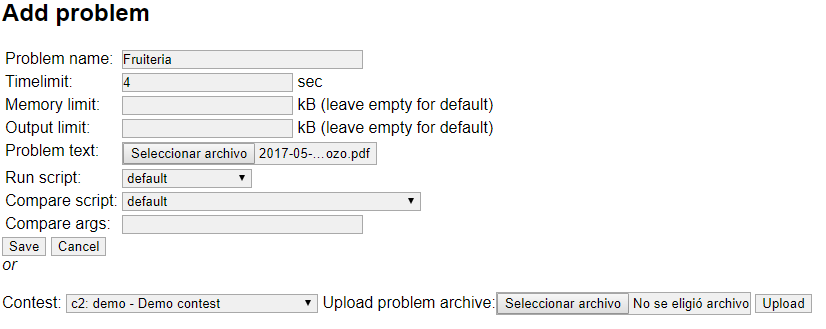
Tria jury-admin interfacte, amb usuari *admin* password *admin* i ha de sortir aquesta finestra:



*Contest →* concurs, és un conjunt de problemes que es posen en una competició. Comprova que tens actiu el *constest c2 (demo)*

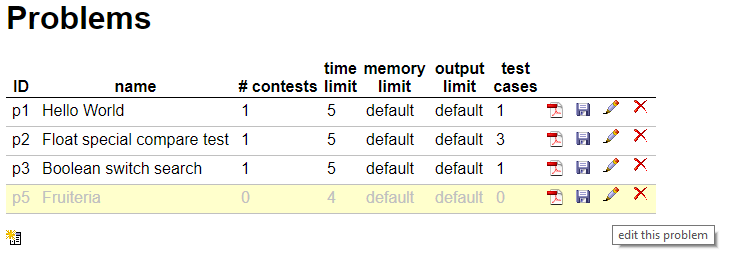
*Problems →* problemes, són els problemes que els concursants han de resoldre. Incialment hi ha tres problemes preparats.

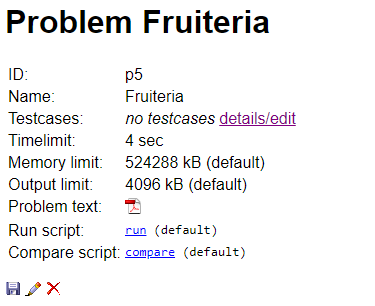
Tu hauràs de fer un i pujar-ho a un contest.

Si li dones a nou problema, sortirà una pantalla on podrem configurar un nou problema.

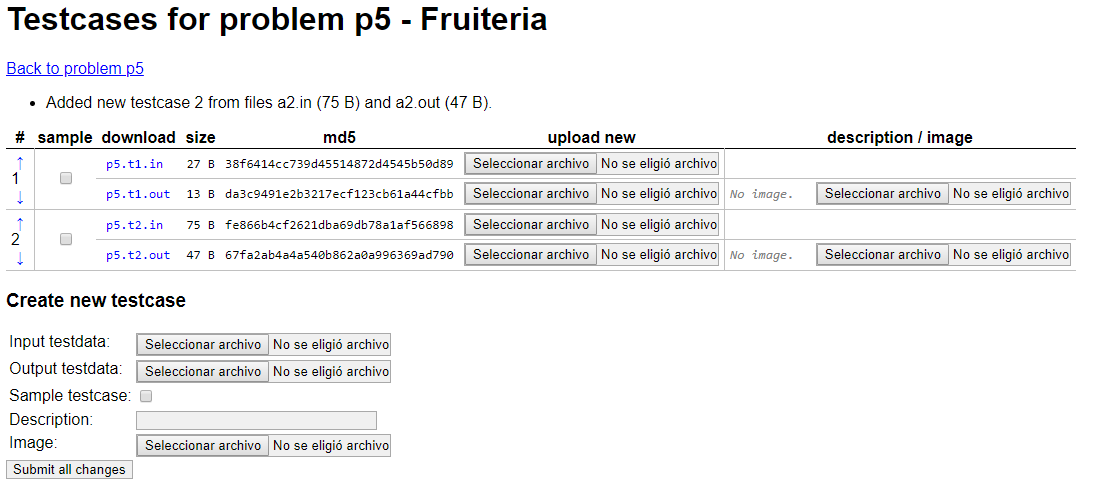
Per exemple, pugem *La Fruiteria del Pozo*.

Caldrà posar un *timelimit* (posa qualsevol valor positiu). Pots pujar també un enunciat *(problem text)*

Un cop salvat el problema, es poden afegir els casos de prova, seleccionant de nou el problema i editant-lo

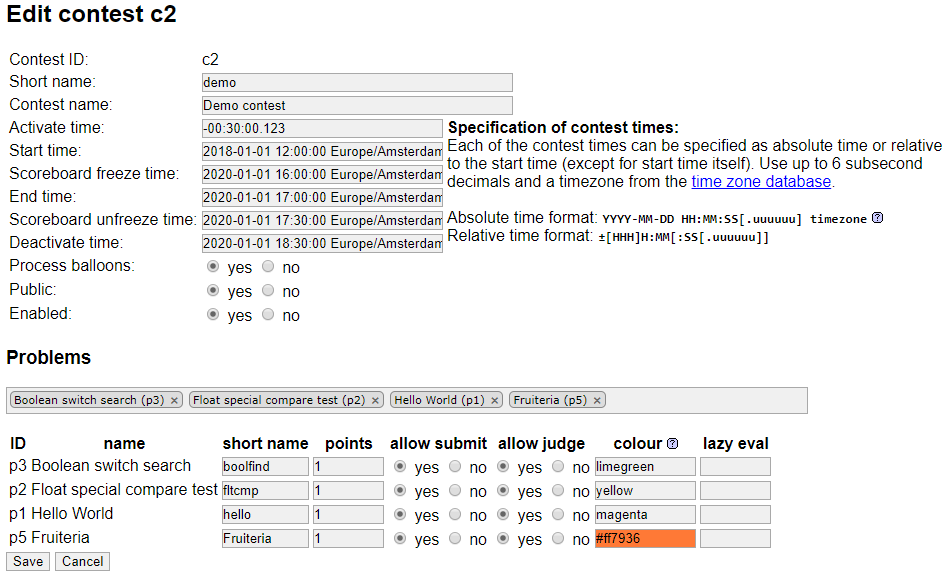


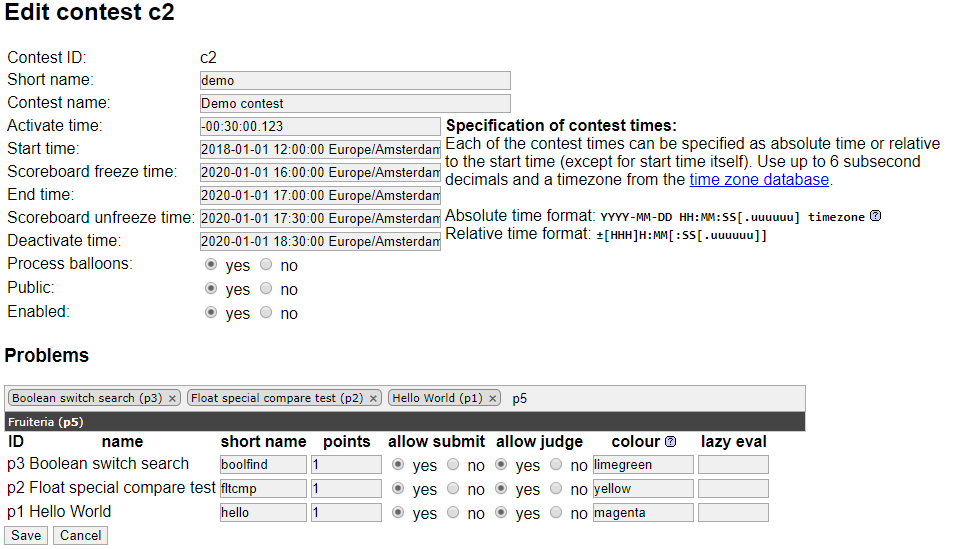
# 

En aquest cas hem posat dos blocs de casos de proves. Normalment es diuen 

*\*.in* per les entrades

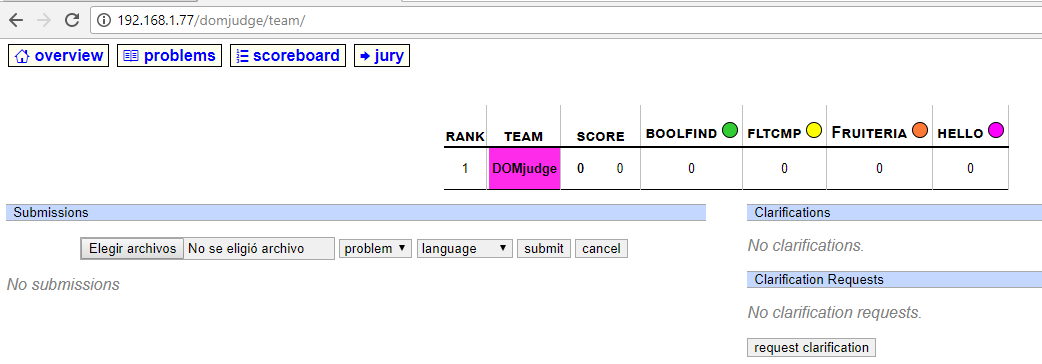
*\*.out* per les sortides

Finalment, editarem el *contest* actiu (en aquest cas **c2**) i afegirem el problema indicant el seu nom



Al final ha de quedar el problema enregistrat dins el *contest*. Cas contrari el problema no sortirà al concurs.

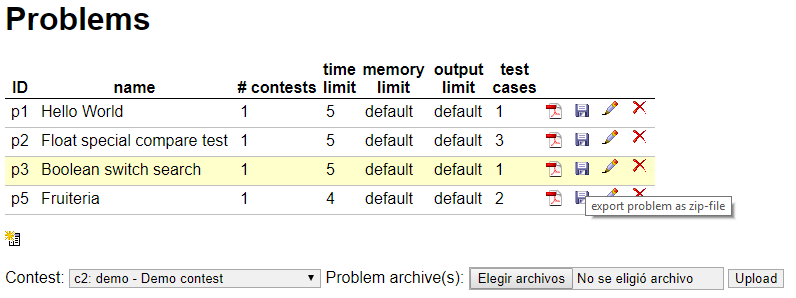
Canviant la vista a ***team*** s’haurà de veure el nostre problema, i si te un enunciat afegit, clickant a sobre s’obrirà.



La resta, ja és conegut. Podeu pujar propostes de solució en diferents llenguatges i el jutge validarà les solucions.

Si us dona qualsevol tipus d’error, podeu veure l’apartat *submissions* del jutge.

Un problema es pot exportar en un fitxer zip, mitjançant l’opció corresponent



# 6.- PROBLEMA PEL DOMJUDGE

La feina que es demana és:

1. Pensa un problema com els que has treballat de domjudge.

* La seva temàtica serà lliure, però t’animem a que tingui a veure amb coses relatives a l’institut, o a la teva realitat propera, simplement perquè sigui més divertit.
* Caldrà escriure l’enunciat en un document en català sense faltes d’ortografia i un estil amigable, intenta que tingui una mica de gràcia. Deixa volar la imaginació. En cap cas s’acceptarà cap enunciat que sigui ofensiu per ningú.
* El problema s’ha de poder resoldre en termes de domjudge, tal com has conegut a classe dins del crèdit M03UF1, per exemple.

1. Implementa la solució correcta en llenguatge Java.
2. Genera la documentació corresponent amb JavaDoc, tal com s’indica al punt 3
3. Genera amb jUnit proves pel teu problema.
4. Dissenya els casos de prova, tant l’entrada com la sortida, el més correctes i exhaustius possible.
5. Puja-ho al domjudge tal com s’ha explicat al punt 5 (enunciat i cas de prova). El nom del problema serà el mateix que el teu nom d’usuari.
6. Puja a GitHub perquè tothom pugui veure què has fet i compartir-ho.