

CFGS Desenvolupament d'Aplicacions Web
Mòdul 8 – Desplegament d'Aplicacions Web
UF4 – Sistemes de control de versions i documentació
EAC4
(Curs 2021–22 / 1r semestre)

Presentació i resultats d'aprenentatge

Aquest exercici d'avaluació continuada (EAC) es correspon amb els continguts treballats a la unitat 4 **Control de versions i documentació**.

Els **resultats d'aprenentatge** que es plantegen són:

- RA1. Elabora la documentació de l'aplicació web avaluant i seleccionant eines de generació de documentació i control de versions.

Criteris d'avaluació

La puntuació màxima assignada a cada activitat s'indica a l'enunciat.

Els criteris que es tindran en compte per avaluar el treball de l'alumnat són els següents:

- CA1.1 Identifica diferents eines de generació de documentació.
- CA1.2 Documenta els components de programari utilitzant els generadors específics de les plataformes.
- CA1.3 Utilitza diferents formats per a la documentació.
- CA1.4 Utilitza eines col·laboratives per a l'elaboració i el manteniment de la documentació.
- CA1.5 Instal·la, configura i utilitza un sistema de control de versions.
- CA1.6 Garanteix l'accessibilitat i la seguretat de la documentació emmagatzemada pel sistema de control de versions.
- CA1.7 Documenta la instal·lació, la configuració i l'ús del sistema de control de versions utilitzat.


Forma i data de lliurament

Per respondre les qüestions heu d'utilitzar aquest mateix fitxer. Elimineu els apartats *Presentació i resultats d'aprenentatge*, *Criteris d'avaluació* i *Forma i data de lliurament*.

Un cop finalitzat l'exercici d'avaluació continuada heu d'enviar el document des de l'apartat **M08 Lliurament EAC4** de l'aula, dins del termini establert. Tingueu en compte que el sistema no permetrà fer lliuraments després de la data i hora indicades.

Lliureu el fitxer en **format PDF**. El nom del fitxer serà el següent: **DAW_M08B0_EAC4_Cognom1_Inicial del cognom2.pdf**. Els cognoms s'escriuran sense accents. Per exemple, l'estudiant *Joan García Santos* posaria el següent nom al seu fitxer de l'EAC3: **DAW_M08B0_EAC4_Garcia_S.pdf**.

Substituïu **Nom i cognoms** de la capçalera per les vostres dades personals.

	Codi: I71	Exercici d'avaluació contínua 4	Pàgina 1 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

El termini de lliurament finalitza a les 23:55 h del dia 14/12/2021. La proposta de solució de l'EAC4 es publicarà el dia 21/12/2021 i les qualificacions el dia 21/12/2021.


Creació de videotutorials

Per la creació dels videotutorials, a banda de la imatge, haureu d'utilitzar la vostra veu per descriure les passes que esteu seguint. A més, seguint les indicacions de la nova normativa de l'IOC, **heu d'aparèixer en el vídeo mentre feu l'explicació**. Se us ha de poder identificar visualment. És imprescindible per validar l'autoria.



Per realitzar el vídeo, podeu utilitzar les eines de gravació/captura de vídeo que vulgueu, tanmateix us recomanem dues possibles solucions.

- Per una banda, podeu utilitzar l'aplicació web gratuïta, [Loom](#), atès que permet la gravació de vídeos capturant la imatge i el so de l'ordinador i de la webcam, i alhora emmagatzemar-los al núvol.
- Per l'altra, també us recomano l'eina [OBS](#), amb aquesta podreu gravar la pantalla així com a vosaltres mateixos/es a través d'una webcam o mòbil. En aquest cas, haureu de pujar el vídeo a alguna plataforma tipus Youtube (en privat però donant-me accés), o qualsevol altra eina per veure-ho per Internet (exemple: Mega, DailyMotion, Drive...).

	Codi: 171	Exercici d'avaluació contínua 4	Pàgina 2 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

Enunciat

És necessari documentar tots els passos que es demanen dins l'EAC. Aquesta documentació s'haurà de dur a terme tant amb captures de pantalles com amb les explicacions corresponents.

La pràctica està dividida en cinc parts:

- **Part 1 - Documentació d'aplicacions.** - 2 punts
- **Part 2 - Serveis web d'hostatge de projectes utilitzant GitLab** - 1 punt
- **Part 3 - Treballant amb Netbeans i amb branques. Resolució bàsica de conflictes.** - 2,5 punts
- **Part 4 - Fonaments bàsics de funcionament del sistema de control de versions Git. Ordres per gestionar el SCV Git.** - 2 punts
- **Part 5 - Resolució de conflictes entre branques del sistema de control de versions Git** – 2,5 punts

En cadascuna de les parts trobareu la puntuació de cada subapartat.


Introducció: Metodologies de documentació, Git i GitLab

Per desenvolupar un projecte informàtic cal generar molts tipus diferents de documentació a cada fase del projecte. Els elements que s'han de documentar depenen de la metodologia utilitzada per planificar el projecte. En aquest EAC treballarem diferents metodologies per documentar documents **JavaDoc** i **JSDoc**.

Git va ser creat per **Linus Torvalds** l'any **2005** per al desenvolupament del nucli de **Linux**. Es tracta d'un programari de control de versions que utilitza un sistema distribuït i, per consegüent, cada usuari que clona un repositori obté una còpia completa dels fitxers i l'historial de canvis.

Tot i que **Git** és un sistema distribuït, pot utilitzar-se una còpia del repositori en un servidor de manera que tots els usuaris pugin i baixin els canvis d'aquest repositori per facilitar la sincronització.

GitLab és una plataforma per a la col·laboració en la programació de codi obert que utilitza el sistema de control de versions Git.

	Codi: 171	Exercici d'avaluació contínua 4	Pàgina 3 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

Part 1 – Documentació d'aplicacions

1- Documentació de codi amb Javadoc – 1 punt

L'objectiu d'aquesta activitat és aprendre a documentar codi en format Javadoc.

1. Documenteu la següent classe de Java utilitzant el format Javadoc. Mostra els resultats utilitzant el codi que tens disponible següidament:

```
public class Persona {  
  
    private String nom;  
    private int edat;  
    private String DNI;  
  
    public Persona (String nom, int edat, String DNI) {  
  
        this.nom = nom;  
        this.edat = edat;  
        this.DNI = DNI;  
  
    }  
  
    public String getNom() { return this.nom; }  
  
    public int getEdat() { return this.edat; }  
  
    public String getDNI() { return this.DNI; }  
  
    public String getPersona() { return "En/Na " + this.nom + " amb DNI " +  
        this.DNI + " té " + this.edat + " anys" + ; }  
  
}
```

2- Documentació de codi amb JSDoc – 1 punt

L'objectiu d'aquesta activitat és aprendre a documentar codi en format **JSDoc**.

2. Documenteu el següent codi de **JavaScript (ES2015/ES16)** en format **JSDoc**:

```
class Casa {  
  
  constructor(nHabitacions, nPisos, tipusCalefacció){  
  
    this.nHabitacions = nHabitacions;  
    this.nPisos = nPisos;  
    this.tipusCalefacció = tipusCalefacció;  
  
  };  
  
  inquilinsMaxims(){  
  
    return this.nHabitacions * 2;  
  
  };  
};  
  
let c = new Casa(2,1, gas);  
console.log(c.inquilinsMaxims());
```

Part 2 - Serveis web d'hostatge de projectes utilitzant GitLab

Pràctica

Haureu de crear un compte a **GitLab**. Es pretén treballar amb diferents casuístiques i problemes típics dels sistemes de control de versions.

Heu de realitzar un document amb captures de pantalla on mostreu el procés que heu seguit i els diferents jocs de proves.

Recursos

- Treball bàsic de **Git**:

<http://git-scm.com/book/en/Getting-Started-Git-Basics#The-Three-States>

<http://git-scm.com/book/en/Git-Basics-Recording-Changes-to-the-Repository>

<http://git-scm.com/book/en/Git-Basics-Working-with-Remotes#Pushing-to-Your-Remotes>

- Compartir projectes partint de GitLab:

<https://docs.gitlab.com/ee/user/project/members/>

3- Introducció a GitLab – 1 punt


- Escriuiu les diferències principals entre GitLab i Github. També les principals característiques que comparteixen.
- Creeu un nou compte a **GitLab** - <https://about.gitlab.com/>
- Creeu un nou repositori en blanc de nom **eacGitLabIOC**, de nivell d'accés **Públic**.
- Des de Netbeans creeu un nou projecte de tipus Java Application i de nom **prjava01**. Si treballes amb **NetBeans 12.0 IDE**, el projecte serà de tipus Java Application i de categoria Java with Gradle.

Creeu el següent codi:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package prjava01;

import java.io.*;

public class prjava01 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException {
        File f = new File("fitxer.html");
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(f))) {
            bw.write("<html>");bw.newLine();
            bw.write("  <head>");bw.newLine();
```

	Codi: I71	Exercici d'avaluació contínua 4	Pàgina 6 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

```
        bw.write("    <title>");bw.newLine();
        bw.write("        Nova p&agrave;gina web");bw.newLine();
        bw.write("    </title>");bw.newLine();
        bw.write(" </head>");bw.newLine();
        bw.write(" <body>");bw.newLine();
        bw.write("    Nova p&agrave;gina web");bw.newLine();
        bw.write(" </body>");bw.newLine();
        bw.write("</html>");bw.newLine();
        bw.close();
    } }
```

NOTA: Si treballes amb **NetBeans IDE 12** en comptes de public class **prjava01** hauràs d'utilitzar public class Main.

Ara compileu el codi, comproveu que no hi ha errors (si n'hi ha solucioneu-los) i després executeu l'aplicació. Comproveu que a la vostra carpeta del projecte s'ha generat un fitxer HTML de nom **fitxer00.html**.

7. Des de **Netbeans** aprofitant les eines del menú "**Team**" creeu i inicialitzeu un dipòsit local de tipus **Git**. Comproveu que s'ha creat una carpeta de nom **.git** dins de la vostra carpeta del projecte **prjava01**.
8. Feu un **Add** i després un **Commit** del projecte. A la secció "**Commit Message**" escriviu "**Versió 0.1 del projecte eacGitLabIOC**".
9. Sincronitzeu el dipòsit local amb el remot situat al servidor de **GitLab**. Utilitzeu la informació de l'adreça URL que trobareu a la secció "**Quick setup**" del dipòsit remot. Haureu d'utilitzar el nom d'usuari complet i la contrasenya d'accés a **GitLab**.
10. Actualitzeu el dipòsit **prjava01** a **GitLab** i comproveu que el dipòsit local ha estat pujat al dipòsit remot. Comproveu el contingut de la versió pujada al dipòsit remot.
11. Afegiu el següent codi com a primera línia del cos (body) de la pàgina web:

```
        bw.write(" <h1>DAW2 m08uf4eac4</h1>");bw.newLine();
```

12. Feu un **Add** i després un **Commit** del projecte. A la secció "**Commit Message**" escriviu "**Versió 0.2 del projecte eacGitLabIOC**". Torneu a fer un "**push**" per pujar els canvis locals al dipòsit remot dins del **GitLab**. Comproveu que els canvis s'han pujat.
13. Visualitzeu al repositori de GitLab:
 - a) El codi de la versió 0.2 del projecte.
 - b) Els canvis respecte de la versió 0.1 del projecte.

Part 3 - Treballant amb Netbeans i amb branques.

NetBeans és un entorn de desenvolupament integrat lliure, fet principalment per al llenguatge de programació Java. Existeixen a més un nombre important de mòduls per estendre'l. **NetBeans** és un producte lliure i gratuït sense restriccions d'ús.

Pràctica

Treballareu amb el programari **Netbeans**. Tractareu les diferents casuístiques i problemes típics dels sistemes de control de versió utilitzant **Netbeans** com a recolzament.

Heu de realitzar un document amb captures de pantalla on mostreu el procés que heu seguit per instal·lar el servidor i els diferents jocs de proves.

Recursos

- Treball amb Netbeans i Git: <https://netbeans.org/kb/docs/ide/git.html#add>
- Resolució de conflictes amb Git: http://githowto.com/resolving_conflicts

4 - Configuració Netbeans – 0,5 punts

14. Des de **Netbeans** importeu el projecte de tipus **Java Application** i de nom **prjava02** que trobareu comprimit en forma **.zip**.

Compileu el codi, comproveu que no hi ha errors i que podeu executar l'aplicació.

15. Heu de crear un nou dipòsit en blanc a GitLab de nom **prjava02** i de nivell d'accés Públic.

16. Des de **Netbeans** inicialitzeu un dipòsit local de tipus Git. Feu un **Add** de tot el projecte i després un Commit amb el comentari "**Versió 1 del projecte prjava02**" i afegint el teu correu a **Author** i **Committer**. Després, sincronitzeu el dipòsit local amb el remot situat al servidor de **GitLab**. Comproveu que el dipòsit local ha estat pujat al dipòsit remot. Tanqueu el projecte **prjava02** i surtiu de **Netbeans**.

5 – Treball per crear i fusionar branques – 2 punts

17. Des de dins de la carpeta del projecte **prjava02** del vostre ordinador, heu de crear una branca de nom **branca00**. Entra dins de la **branca00**. Trobareu informació de com fer-ho a la documentació. Comproveu des de l'interpret d'ordres l'estat del projecte i a quina branca et trobes.


18. Modifiqueu el fitxer **Prjava02.java**. Afegiu la següent línia de codi:

```
System.out.println("Creació d'una branca del projecte prjava02");
```

Des de dins de la carpeta del projecte **prjava02**, comproveu l'estat del projecte. Comproveu quins fitxers s'afegiran al dipòsit git en el següent commit.

19. Des de dins de la carpeta del projecte **prjava02**, feu un "add" i després un "commit" de **branca00**. Escriviu el comentari "Treballant amb ordres Git i branques: Versió 1.1 del projecte prjava02". Comproveu quins fitxers s'han afegit al dipòsit git.

20. Mostreu l'historial de la **branca00**.

	Codi: 171	Exercici d'avaluació contínua 4	Pàgina 8 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

21. Torneu a la branca **master** del projecte. Comproveu l'estat del projecte. Mostreu l'historial de la branca **master**. Quina diferència hi ha amb l'historial de la **branca00**?
22. Actualitzeu el contingut del fitxer **Prjava02.java**. Podeu veure l'última línia afegida? Per què?
23. Torneu a la **branca00**. Actualitzeu el contingut del fitxer **Prjava02.java**. Podeu veure l'última línia afegida? Per què?
24. Modifiqueu el fitxer **Prjava02.java**. Afegeiu la següent línia de codi al programa:

```
System.out.println("Afegint més codi a la branca00 del projecte prjava02");
```

Comproveu l'estat del projecte. Comproveu quins fitxers s'afegiran al dipòsit git.

25. Feu un **Add** i un **commit**. Escriviu el comentari "Treballant amb ordres Git i branques: Versió 1.2 del projecte prjava02". Comproveu ara l'estat i l'historial de la branca **branca00**.
26. Torneu a la branca **master**. Comproveu el contingut del fitxer **Prjava02.java**. **Fusioneu (merge)** la branca **branca00** amb la branca **master**. Trobareu informació a la documentació. Comproveu que passa amb el contingut del fitxer **Prjava02.java**.
27. Comproveu ara l'estat i l'historial del projecte a la branca **master**.
28. Feu un **push** del dipòsit local al dipòsit remot. Des de **GitLab**, actualitzeu la pàgina web del projecte, i comproveu que els canvis han estat realitzats correctament. Comproveu que només s'ha actualitzat la branca **master** únicament. Per què?
29. Des de dins de la carpeta del projecte **prjava02** heu de crear una branca de nom **branca01**. Entreu a la branca **branca01**. Modifiqueu el fitxer **Prjava02.java**. Afegeiu la següent línia de codi al programa:

```
System.out.println("Afegint codi a la branca01 del projecte prjava02");
```

Feu un **Add** i un **commit** local del fitxer del projecte. Escriviu el comentari "Treballant amb ordres Git i branques: Versió 1.3 del projecte prjava02".

30. Des de la **branca01**, Feu un **push** del dipòsit local al dipòsit remot. Des de **GitLab**, actualitzeu la pàgina web del projecte, i comproveu que els canvis han estat realitzats correctament. Comproveu que ara s'ha generat la **branca01** a **GitLab**.
31. Torneu a la **branca00**. Des de la **branca00** feu una **fusió** amb la **branca01** i després feu un **push** del dipòsit local al dipòsit remot. Des de **GitLab**, actualitzeu la pàgina web del projecte, i comproveu que els canvis han estat realitzats correctament. Comproveu l'estat de la **branca00** a **GitLab**.

Part 4 - Fonaments bàsics de funcionament del sistema de control de versions Git. Ordres per gestionar el SCV Git.

Pràctica

Treballareu directament amb el programari **GIT**. Tractareu les diferents casuístiques i problemes típics dels sistemes de control de versions.

Heu de realitzar un document amb captures de pantalla on mostreu el procés que heu seguit per instal·lar el servidor i els diferents jocs de proves.

Recursos

- 1- Conceptes bàsics de Git: Snapshots. Integritat. Operacions locals. Les 3 àrees de residència dels fitxers.
<http://git-scm.com/book/en/v2> → 1.3 What is Git?
<http://git-scm.com/book/es/v2> → 1.3 Fundamentos de Git
- 2- El cicle de vida dels fitxers: Enregistrant canvis en el dipòsit.
<http://git-scm.com/book/en/v2> → 2.2 Recording changes to the Repository
<http://git-scm.com/book/es/v2> → 2.2 Guardando cambios en el Repositorio
- 3- Configuració inicial de Git
<http://git-scm.com/book/en/v2> → 1.6 First-Time Git Setup
<http://git-scm.com/book/es/v2> → 1.6 Configurando Git por primera vez
- 4- Ordres bàsiques de Git
<http://git-scm.com/book/en/v2> → 2 Git Basics (2.1 a 2.5 per aquesta pràctica)
<http://git-scm.com/book/es/v2> → 2 Fundamentos de Git (2.1 a 2.5 per aquesta pràctica)


6 - Treballant amb ordres per gestionar el SCV Git – 2 punts

32. Creeu un fitxer de configuració mínima de **GIT** pel vostre usuari amb la informació que trobareu als enllaços. Feu que a la configuració surti com a nom d'usuari, el nom d'usuari **GitLab** que va crear a la part anterior. Executeu l'ordre per comprovar la nova configuració. Comproveu que s'ha creat el fitxer **.git/config**.
33. Implementeu dins del teu directori personal un nou directori de nom **EACgitM08**. Dins d'aquest directori, creeu un altre directori de nom **mitjana**. Dins del directori implementarem el fitxer **mitjana.c**

```
#include <stdio.h>

int main() {
    float suma=0;
    float mitjana;
    float nou_valor;
    int numero_repeticions=0;
    char consulta='s';

    while(consulta=='s') {
        printf("Dona la següent dada: ");
        scanf("%f",&nou_valor);
        suma=suma+nou_valor;
    }
}
```

	Codi: I71	Exercici d'avaluació contínua 4	Pàgina 10 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

```

        numero_repeticions=numero_repeticions+1;
        printf("Vols continuar (s/n): ");
        scanf(" %c",&consulta);
    }
    mitjana=suma/numero_repeticions;
    printf("La mitjana resultant és: %.2f\n",mitjana);

    return 0;
}

```

34. Creeu i inicialitzeu un nou dipòsit local GIT pels fitxers del projecte mitjana. Comproveu que s'ha creat el directori **.git** dins del directori mitjana. Comprova l'estat del dipòsit. Indiqueu quin és l'estat del fitxer **mitjana.c** i expliqueu el motiu pel qual es troba en aquest estat.
35. Afegiu el fitxer a l'àrea de preparació (**staged area**). Comproveu ara l'estat del dipòsit. Indiqueu quin és l'estat del fitxer **mitjana.c**.
36. Modifiqueu **mitjana.c**. Canveu **scanf(" %c",&consulta);** per **scanf("%s",&consulta);**. Comproveu ara l'estat del dipòsit. Indica quin és l'estat del fitxer **mitjana.c**. D'acord amb l'estat actual, en el moment de fer **una consignació**, quina versió de l'arxiu s'emmagatzemarà en el dipòsit **GIT**?
37. **Consigneu** els fitxers que es troben a l'àrea de preparació (**staged area**). En el moment de consignar afegiu el comentari "**Primer commit del projecte mitjana**". Trobeu el valor del **checksum (o suma de comprovació)** que s'ha creat per aquest **consignació**. Comproveu la llista de fitxers afegits.
38. Comproveu ara l'estat del dipòsit. Indiqueu quin és l'estat del fitxer **mitjana.c**.
39. Afegiu **mitjana.c** a l'àrea de preparació i feu una nova consignació. En el moment de consignar afegiu el comentari "**Segon commit del projecte mitjana**". Trobeu el nou valor de la nova suma de comprovació. Comproveu novament l'estat del dipòsit. Indiqueu el motiu del resultat mostrat per pantalla.
40. Compileu el programa **mitjana.c**. Comproveu l'estat del dipòsit **GIT**. Quin és l'estat del programa **mitjana.bin**? Per què?
41. Expliqueu quin és el propòsit del fitxer **.gitignore**. Heu de crear un fitxer de nom **.gitignore** dins del directori mitjana amb el següent contingut:


```

*.sio
*~
.gitignore
mitjana.bin

```

A continuació compileu el programa **mitjana.c** amb els següents paràmetres: **gcc mitjana.c -save-temps -o mitjana.bin**. Comproveu els arxius que hi ha ara al vostre directori. Comproveu l'estat del dipòsit. Expliqueu per què no hi ha res per afegir i consignar.

42. Afegiu el fitxer **mitjana.bin** a l'àrea de preparació. Ho podeu fer? Per què? Comproveu l'estat del dipòsit.
43. Modifica el fitxer **mitjana.c**. Canvieu **float suma=0;** per **float suma=0.0;**. Comproveu ara l'estat del dipòsit i del fitxer **mitjana.c**. Afegiu el fitxer **mitjana.c** a l'àrea de preparació (que pot tenir altres

	Codi: 171	Exercici d'avaluació contínua 4	Pàgina 11 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

noms: <http://www.gitguys.com/topics/whats-the-deal-with-the-git-index>). Comproveu ara l'estat del dipòsit i del fitxer **mitjana.c**.

44. Torneu a modificar el fitxer **mitjana.c**. Canvieu **float mitjana;** per **float mitjana=0.0;**. Comproveu l'estat del dipòsit.
45. Presenteu per pantalla un historial de totes les consignacions realitzades. Comproveu el valor del **checksum** de tots els **commits**.
46. Feu un **add** del fitxer **mitjana.c** i després un **commit**. En el moment de consignar afegiu el comentari "**Tercer commit del programa mitjana**". Comproveu l'estat del projecte. Comproveu també l'historial. Comproveu a quina branca us trobeu en el projecte.
47. Heu de crear un projecte **públic** de nom **mitjana** (o utilitzar el creat en l'apartat anterior) al vostre dipòsit personal de **GitLab**. Veureu que en el moment de la creació del nou dipòsit us donen informació de com utilitzar-lo. Llegiu la quarta recomanació (Push an existing Git repository) i feu un **push** del vostre projecte local al dipòsit remot. Per fer-ho, primer, haureu d'afegir el dipòsit remot a la llista de dipòsits remots amb els quals treballem fent un **remote add** i després fareu el **push**. Un cop hagueu acabat, comproveu que heu pujat el projecte al vostre dipòsit remot.
- Expliqueu què és **origin**. --> Ajut: <http://stackoverflow.com/questions/9529497/what-is-origin-in-git>.
48. Creeu una nova carpeta de nom **clone_mitjana**. I clona el projecte (Ajut: <https://git-scm.com/docs/git-clone>) dins d'aquesta carpeta. Comproveu que s'ha creat la carpeta del projecte i que dins d'aquesta carpeta hi ha un nou dipòsit local. Comproveu l'estat del dipòsit clonat. Comproveu l'historial.
49. Modifiqueu el fitxer **mitjana.c** que es troba al projecte clonat. Canvieu **float nou_valor;** per **float nou_valor=0.0;**. Feu un **add** i un **commit**. En el moment de consignar afegiu el comentari "**Quart commit del programa mitjana**". Pugeu el canvi (feu un **push**) al dipòsit remot. Tingueu en compte que com heu clonat el dipòsit no cal afegir el dipòsit remot a llista de manera que només heu de fer **push**. Comproveu des de **GitLab** l'estat del dipòsit remot.
50. Torneu al directori original. Baixeu (**pull**) del projecte que està al dipòsit remot al dipòsit local. Ara obriu el fitxer **mitjana.c** i comproveu que s'ha afegit el nou codi.
51. Modifiqueu el fitxer **mitjana.c** que es troba al directori original. Canvieu a la línia 8 **char consulta='s';** per **consulta='y';**. Afegiu el fitxer **mitjana.c** a l'àrea de preparació. Modifiqueu a la línia 10 **while(consulta=='s')** per **while(consulta=='y')**. Comproveu les següents diferències:
- a) La diferència entre la informació que hi ha al directori de treball i a l'últim **commit**.
- b) La diferència entre la informació que hi ha a l'àrea de prep directori de treball i a l'últim **commit**.
52. Esborreu el fitxer **mitjana.c** de l'últim **add** realitzat. Comproveu l'estat del projecte.
53. Feu que el fitxer **mitjana.c** que es troba al directori de treball torni a utilitzar el codi que hi havia a l'últim **commit** realitzat. Comproveu l'estat del projecte. Comproveu el contingut de **mitjana.c**.
54. Revertiu el fitxer **mitjana.c** que es troba a l'àrea de treball a la versió que tenia al 2n **commit**. Comproveu el contingut de **mitjana.c**.

55. Comproveu l'historial de **commits** realitzats fins aquest punt. Ara esborreu completament l'últim commit realitzat a la branca **master**.

56. Visualitzeu ara l'historial de **commits** realitzats i comproveu que l'últim **commit** ha estat esborrat.

Part 5 - Resolució de conflicte entre branques del sistema de control de versions Git

Pràctica

Tractareu les diferents casuístiques i problemes típics dels sistemes de control de versió dins d'un servidor GIT.

Heu de realitzar un document amb captures de pantalla on mostreu el procés que heu seguit per instal·lar el servidor i els diferents jocs de proves.

Recursos

- Treball amb Branques: <http://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>
- Resolució de conflictes típics: <https://help.GitLab.com/articles/resolving-a-merge-conflict-from-the-command-line/>
- HEAD: https://www.youtube.com/watch?feature=player_detailpage&v=ZDR433b0HJY#t=2791s

7 - Preparació – 0,5 punts

57. Heu de crear una carpeta de nom **mitjana00** dins del vostre directori personal. Descarregueu el fitxer **mitjana00.tar.gz**.

Extreure els fitxers **mitjana00.c** i **README** que trobareu dins de **mitjana00.tar.gz**. Després esborreu el fitxer que heu descarregat de manera que només tingueu els dos fitxers extrets dins de la carpeta **mitjana00**.

58. Heu de crear i inicialitzar un dipòsit de tipus **git** dins de la carpeta **mitjana00**. Feu un **add** de tot el projecte i un **commit** amb el comentari "**Versió 1 del projecte mitjana00**".

59. Heu de crear un nou dipòsit a **GitLab** de nom **mitjana00** i de nivell d'accés **Públic**. Heu de deixar la resta d'opcions per defecte. Pugeu el commit del projecte local al nou dipòsit de **GitLab**. Comproveu que el dipòsit local ha estat pujat al dipòsit remot.


8 - Resolució de conflictes – 1 punt

IMPORTANT: per resoldre i justificar tot aquest subapart 8 (*preguntes de la 59-68*) haureu de lliurar un videotutorial, seguint les indicacions que es demanen a l'apartat *Creació de videotutorial* que hi ha la capçalera de l'EAC.

Si ho necessiteu, per completar aquest videotutorial, podeu afegir captures de pantalla i explicacions dins dels diferents punts d'aquest apartat.

60. Heu de crear una branca de nom **br0**. Canvieu-vos a la branca **br0** i obriu el fitxer **mitjana00.c**. Afegiu una nova línia de codi:

```
printf("Operació finalitzada amb èxit\n");
```

	Codi: 171	Exercici d'avaluació contínua 4	Pàgina 14 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

61.Des de **br0**, feu un **add** de tot el projecte i un **commit** amb el comentari “**Conflicte 1 entre br0 i master**”.

62.Torneu a la branca **master**. Afegiu una nova linea de codi:

```
printf("La mitjana ha estat trobada amb èxit\n");
```

63.Des de **master**, feu un **add** de tot el projecte i un **commit** amb el comentari “**Conflicte 1 entre master i br0**”.

64.Des de **master**, **fusioneu** la branca **master** amb la branca **br0**. Comproveu el missatge donat pel sistema. Comproveu si s'han fusionat les dues branques. Comproveu l'estat del sistema i comproveu quin missatge dona el sistema.

65.Comproveu que Git ha afegit les següents línies:

```
<<<<<<< HEAD
printf("La mitjana ha estat trobada amb èxit\n");
=====
printf("Operació finalitzada amb èxit\n");
>>>>>>> br0
```

66.Decidiu solucionar el conflicte, mantenint les dues línies de codi. Solucioneu el conflicte editant **mitjana00.c**, mantenint les dues línies de codi i esborrant <<<<<<< HEAD, ===== i >>>>>>> br0. Torneu a comprovar l'estat del sistema. Des del punt de vista de git el conflicte està solucionat?

67.Feu un **add**. Torneu a comprovar l'estat del projecte. Des del punt de vista de git el conflicte està solucionat?

68.Feu un **commit** local. Escriviu el comentari “**Conflicte entre master i br0 solucionat**”. Torneu a comprovar l'estat del projecte. Des del punt de vista de git el conflicte està solucionat?

69.Torneu a **br0**. Fusioneu **br0** amb **master**. Comproveu l'estat del projecte i confirmeu que no hi ha cap problema. Comproveu que hi ha el mateix codi a la **br0** i a **master**.


9 - Resolució de conflictes II – 1 punt

70.Des de **br0** heu de crear un fitxer de nom **README** i escriviu dins del fitxer “**Projecte mitjana00: Versió 2**”. Tanqueu el fitxer **README**. Comproveu l'estat del projecte. Feu un **add** i un **commit** per consignar els nous canvis. Pel commit escriviu el comentari “**Afegint el fitxer README al projecte mitjana00.c**”.

71.Torneu a la branca **master**. Comproveu que el nou fitxer no existeix. **Fusioneu** les dues branques. Comproveu que el fitxer **README** ha estat afegit a la branca **master**. Comproveu l'estat del projecte i l'històric de canvis.

72.Des de la branca **master**, afegiu a **README** una nova línia amb el contingut: “**Segona versió de README**”. Feu un **add** i un **commit** per consignar els nous canvis. Pel commit escriviu el comentari “**Afegint contingut a README**”.

73.Torneu a la **br0**. Elimineu el fitxer **README**. Ara feu un **add** i **commit** amb una única ordre.

	Codi: 171	Exercici d'avaluació contínua 4	Pàgina 15 de 16
	Versió: 02	DAW_M08B0_EAC4_2122S1	Lliurament: 14/12/2021

74.Torneu a la branca **master**. **Fusioneu** la branca **master** amb la **br0**. Comproveu si s'han fusionat les dues branques. Comproveu l'estat del sistema i quin missatge dona el sistema.

75.Feu un **push** de les dues branques del teu dipòsit local al dipòsit remot de **GitLab**.