

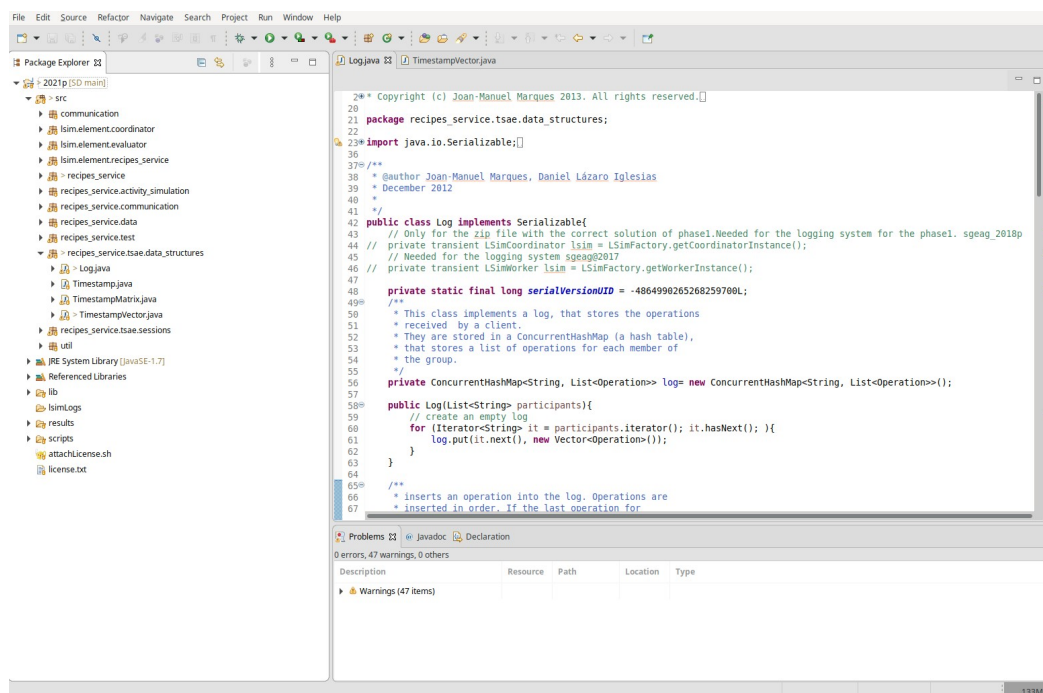
## PRA1. Phase 1. Informe

### Kepa Sarasola Bengoetxea. Grupo 202\_75\_644\_02

## 1. Introduction

### 1.1 Estructura de la solución

En esta primera fase, se nos ha pedido la implementación de varios métodos para poder así añadir operaciones al log. Una vez cargado el proyecto en el IDE Eclipse, me he centrado en entender el funcionamiento de la aplicación centrándome en la carpeta `tsae.data_structures`, donde se encuentran las clases `Log` y `TimestampVector` a las que alude el enunciado de la práctica.



### 1.2 Decisiones tomadas.

#### 1.2.1 Métodos desarrollados para realizar esta práctica

Aunque en los ficheros `.java` descargados junto con el enunciado hay varios métodos no implementados, solamente he desarrollado aquellos que he creído necesarios para cumplir con los requisitos de la práctica, que recordemos era la implementación de las opciones de añadir operaciones al log.

- **Clase Log**

La clase `Log` registra las operaciones recibidas por el cliente e implementa un log, la información sobre las operaciones recibidas se guardan en un `ConcurrentHashMap`. Esta clase implementa algunos métodos que no están desarrollados todavía, para desarrollar esta primera fase, “Phase1”, he desarrollado los métodos `add(Operation op)`, que se encarga de añadir un elemento al *hash table* y `equals()` que evaluar si el objeto recibido es igual al que estamos tratando en la clase.

- **Clase TimestampVector**

La clase TimestampVector se encarga de gestionar un resumen de marcas de tiempo para cada host y se encarga de guardar en el *summary* la marca de tiempo (*timestamp*) de la última operación recibida desde este host.

Para realizar esta primera fase he implementado los métodos *updateTimestamp()* que añade la marca de tiempo recibida en el objeto al vector de marcas de tiempo y *equals(Object obj)* que se encarga de evaluar si dos objetos de *timestampVector* son iguales

### 1.2.2 Acceso simultáneo a los datos

Como bien se ha hecho notar en el enunciado, puede existir un acceso simultáneo a los datos, es decir dos instancias de objetos recibidos desde diferentes *threads* pueden mezclarse, produciendo confusión y resultados erróneos.

Ante esta situación en la que puede darse *Multi-thread*, es necesario asegurarse de que mientras un hilo esta accediendo a un recurso dado, solamente el *thread* implicado pueda realizar cambios en los recursos comprometidos.

Para evitar estas interferencias y posibles errores, he usado la palabra clave *synchronized* en la declaración del método. Java ofrece esta palabra clave para poder sincronizar tareas mediante el uso de bloques sincronizados, asegurando así que todos los bloques sincronizados sobre el mismo objeto sólo pueden tener un hilo ejecutándose dentro de ellos a la vez.

Todos los demás hilos que intentan entrar en el bloque sincronizado se bloquean, hasta que el hilo que está usando los recursos del bloque sale de este liberando los recursos.

## 2. Implementation

### 2.1 Partes añadidas al código

Como ya he comentado en el punto anterior, los métodos que añadido en el código son las siguientes:

- **Clase Log**
  - **Método add(Operation op)**

Este método es usado para añadir una nueva operación al log. El funcionamiento del método está sujeto a una restricción que hay que tener en cuenta y es que si la última operación que hay en el log no es anterior a la operación que queremos añadir, esta no se añade al log. Por lo tanto antes de añadir nada al log hay que evaluar las marcas de tiempo.

El método realiza diversas comprobaciones para ver que la operación contiene un objeto que no es null, una vez comprobado esto, comprueba que se cumpla la restricción indicada.

El método es precedido por la palabra clave *synchronized* tal y como hemos indicado anteriormente con el objeto de proteger la integridad de los recursos que utiliza el método.

Imagen con la implementación del método:

```
public synchronized boolean add(Operation op){
    //TODO Phase 1
    //Añadir la operación al log, en caso de que el timeStamp sea posterior al último registrado en este

    Timestamp lastTime; //se usa para registrar el timestamp de la última operación
    String hostId = op.getTimestamp().getHostid(); //obtenemos el id del host que ha enviado la operación
    List<Operation> ops = log.get(hostId); //Creamos una lista con los logs que atañen al host que envía la operación op

    //chequeamos si existe alguna operación previa en la lista de operaciones
    if (ops == null || ops.isEmpty()) {
        //No hay ninguna operación en el log
        lastTime = null;
    }else {
        lastTime = ops.get(ops.size() -1).getTimestamp();
        //Obtenemos el timestamp de la última operación
        //Sabemos que la última operación es la que ocupa la posición, tamaño de lista -1,
        //ya que la primera ocupa la posición 0
    }
    //Una vez obtenido el timeStamp de la última operación la compararemos con el timestamp de la operación actual
    //para comprobar que no sea anterior
    long diff = op.getTimestamp().compare(lastTime);

    if ((lastTime == null && diff == 0) || (lastTime != null && diff==1)) {
        //si el timeStamp de la operación op es posterior al de la última anotación en el log, la añadimos
        log.get(hostId).add(op);
        return true;
    }else {
        //en el caso de que el timeStamp de la operación op no cumpla con la condición,
        //no la añadimos al log
        return false;
    }
}
}
```

#### ○ Método equals (Object obj)

El objetivo de este método es la de observar si el objeto que recibe la función como parámetro *obj* referencia al mismo objeto *log* que es la instancia de la clase Log.

El método *equals* realiza las comprobaciones pertinentes para evitar trabajar con objetos que sean null y comprueba que el objeto recibido por parámetro sea una instancia de Log, una vez realizadas estas comprobaciones, compara los dos objetos utilizando la clase genérica de Java `java.util.AbstractMap.equals(Object obj)`.

Imagen con la implementación del método:

```
public boolean equals(Object obj) {
    //TODO Phase 1
    //comprobamos que el objeto que hemos recibido no es null y que es una instancia de la clase Log
    if ((obj == null) || !(obj instanceof Log)) {
        //Si no es una instancia de la clase Log, devolvemos false
        return false;
    }
    Log temp = (Log) obj;
    //Tras chequear que tanto el objeto recibido como el de la clase no son null,
    //realizamos la comprobación de que ambos objetos sean iguales o no mediante la función equals
    if ((this.log == null) || (temp.log == null)) {
        return false;
    }else {
        return this.log.equals((temp).log);
    }
}
}
```

- **Clase TimestampVector**

- **Método updateTimestamp(Timestamp timestamp)**

Este método añade al vector de marcas de tiempo, el *timestamp* que recibe como parámetro junto con el id del host, siempre que este no sea null.

Imagen con la implementación del método:

```
public void updateTimestamp(Timestamp timestamp){
    //TODO Phase 1
    //Añadimos la marca de tiempo recibida al vector de marcas de tiempo, si esta no es null
    if (timestamp != null) {
        this.timestampVector.put(timestamp.getHostid(),timestamp);
    }
}
```

- **Método equals (TimestampVector param)**

El objetivo de este método es la de observar si el parámetro del objeto que recibe la función como parámetro *TimestampVector* referencia al mismo objeto *timestampVector* que una instancia de la clase *TimestampVector*.

Este método realiza varias comprobaciones sobre el objeto que recibe, comprobando que no sea null.

Realiza un primer check para ver si los dos objetos son iguales y finalmente utiliza la clase genérica de Java `java.util.AbstractMap.equals(Object obj)` para realizar la comparación.

```
public boolean equals(TimestampVector param){
    //TODO Phase 1
    //Comprobamos si ambos vectores de marcas de tiempo son iguales
    if (this.timestampVector == (param.timestampVector)) {
        return true;
    }else if (this.timestampVector == null || param.timestampVector == null){
        //comprueba que ninguna de las dos marcas de tiempo sean null, en caso de que alguna lo sea devuelve null
        return false;
    }else {
        //devuelve el valor de la función equals
        return this.timestampVector.equals(param.timestampVector);
    }
}
```

## 2.2 Limitaciones de la solución aportada

Como ya he comentado en el primer punto, la solución aportada solamente trata el problema planteado en el enunciado de la práctica “Phase 1”, es decir, solamente implementa la solución necesaria para poder realizar operaciones de agregación de operaciones al log.

Con respecto a la realización de un proyecto para la implementación del protocolo TSAE, la solución aportada tiene bastantes deficiencias, tanto en cuanto deja bastantes métodos sin implementar:

- En la clase Log: *purgeLog(TimestampMatrix ack)*, *listNewer(TimestampVector sum)*
- En la clase TimestampVector: *updateMax(TimestampVector tsVector)*, *getLast(String node)*, *mergeMin(TimestampVector tsVector)*, *clone()*

### 3. Test

#### 3.1 Test en entorno local

##### 3.1.1 Test de funcionamiento del proyecto descargado, sin ninguna implementación

Antes de nada he realizado un test de funcionamiento una vez descargado el código y sin haber realizado ninguna modificación del código, para ver cuales son los resultados iniciales. Los test se han realizado mediante los dos scripts que contiene la carpeta original:

##### Test1 – recipes\_service.Server

- **Instrucción para el test:** java -cp ../bin:../lib/\* recipes\_service.Server –phase1
- **Descripción:** el test ejecuta el programa java y pone en marcha un servidor el cual permite realizar ciertas operaciones como añadir recetas, obtener el listado de recetas, ver el contenido de los logs y el contenido de los summary.
- **Resultado:** se puede observar el resultado de varias interacciones con la aplicación en la imagen del terminal.

```

Log:

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
1
Enter the title of the recipe to add
recipe1 title
Enter the recipe to add
recipe1

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
3
15-03-2021 18:19:42:268 Phase1_TSAE [TRACE] : Updating the TimestampVectorInserting with the timestamp: localhost:9000: 0
3
Recipes:
(recipe1 title=[recipe1 title, recipe1, localhost:9000])

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
4
Log:

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
5
Summary:
localhost:9000: -1000

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit

```

- **Conclusión:** la aplicación realiza las operaciones que ofrece el menú de manera correcta, pero se observa que ni el Log ni el Summary tienen ningún tipo de contenido.

**Test2 – start.sh**

- **Instrucción para el test:** `./start.sh 20004 --phase1`
- **Descripción:** el test ejecuta el programa java y trata de realizar las operaciones TSAE, al final compara si los ficheros log y summary de los diferentes instancias de servidores son iguales
- **Resultado:** se puede observar el resultado de varias interacciones con la aplicación en la imagen del terminal.

```

kepa@Belkoain: ~/Dokumentuak/Keparenak/UOC/2021p_SD_practi... x  kepa@Belkoain: ~/Dokumentuak/Keparenak/UOC/2021p_SD_practi... x

Your summary: user1:  -1000
user2:  -1000
user0:  -1000
user3:  -1000
user4:  -1000

Log: AddOperation [recipe=[mrtsccko, Content--mrtsccko, user1], timestamp=user1: 0]
AddOperation [recipe=[fbybqmkf, Content--fbybqmkf, user1], timestamp=user1: 1]
AddOperation [recipe=[qycxcapi, Content--qycxcapi, user1], timestamp=user1: 2]
AddOperation [recipe=[gsktxxsh, Content--gsktxxsh, user1], timestamp=user1: 3]
AddOperation [recipe=[hlvdoryn, Content--hlvdoryn, user1], timestamp=user1: 4]
AddOperation [recipe=[hauelsfc, Content--hauelsfc, user1], timestamp=user1: 5]
AddOperation [recipe=[kpetxjq, Content--kpetxjq, user1], timestamp=user1: 6]
AddOperation [recipe=[yjcacuhb, Content--yjcacuhb, user1], timestamp=user1: 7]
AddOperation [recipe=[vgrxismv, Content--vgrxismv, user2], timestamp=user2: 0]
AddOperation [recipe=[igutjmas, Content--igutjmas, user2], timestamp=user2: 1]
AddOperation [recipe=[enwxuoaq, Content--enwxuoaq, user2], timestamp=user2: 2]
AddOperation [recipe=[bycjuiap, Content--bycjuiap, user2], timestamp=user2: 3]
AddOperation [recipe=[clibbrbl, Content--clibbrbl, user2], timestamp=user2: 4]
AddOperation [recipe=[rlretsf, Content--rlretsf, user2], timestamp=user2: 5]
AddOperation [recipe=[vqkrfrdd, Content--vqkrfrdd, user2], timestamp=user2: 6]
AddOperation [recipe=[srhvdadm, Content--srhvdadm, user2], timestamp=user2: 7]
AddOperation [recipe=[skotncrs, Content--skotncrs, user2], timestamp=user2: 8]
AddOperation [recipe=[tlugowex, Content--tlugowex, user2], timestamp=user2: 9]
AddOperation [recipe=[fdfyucma, Content--fdfyucma, user2], timestamp=user2: 10]
AddOperation [recipe=[ftyqrryp, Content--ftyqrryp, user0], timestamp=user0: 0]
AddOperation [recipe=[wfvfcioe, Content--wfvfcioe, user0], timestamp=user0: 1]
AddOperation [recipe=[ayxlnxfj, Content--ayxlnxfj, user0], timestamp=user0: 2]
AddOperation [recipe=[jeqkicvn, Content--jeqkicvn, user0], timestamp=user0: 3]
AddOperation [recipe=[gjtvehkq, Content--gjtvehkq, user0], timestamp=user0: 4]
AddOperation [recipe=[hnsfjsrm, Content--hnsfjsrm, user0], timestamp=user0: 5]
AddOperation [recipe=[tustxmqr, Content--tustxmqr, user0], timestamp=user0: 6]
AddOperation [recipe=[naenayif, Content--naenayif, user3], timestamp=user3: 0]
AddOperation [recipe=[paqlrlar, Content--paqlrlar, user3], timestamp=user3: 1]
AddOperation [recipe=[brqcmawf, Content--brqcmawf, user3], timestamp=user3: 2]
AddOperation [recipe=[vribeukc, Content--vribeukc, user3], timestamp=user3: 3]
AddOperation [recipe=[lupjovh, Content--lupjovh, user3], timestamp=user3: 4]
AddOperation [recipe=[qbmyxmco, Content--qbmyxmco, user3], timestamp=user3: 5]
AddOperation [recipe=[jvuacyfw, Content--jvuacyfw, user3], timestamp=user3: 6]
AddOperation [recipe=[booxoevo, Content--booxoevo, user3], timestamp=user3: 7]
AddOperation [recipe=[cefxgrfy, Content--cefxgrfy, user3], timestamp=user3: 8]
AddOperation [recipe=[nikvppdg, Content--nikvppdg, user3], timestamp=user3: 9]
AddOperation [recipe=[phmfqbsu, Content--phmfqbsu, user3], timestamp=user3: 10]
AddOperation [recipe=[fewjqrh, Content--fewjqrh, user3], timestamp=user3: 11]
AddOperation [recipe=[mtxhcsxk, Content--mtxhcsxk, user4], timestamp=user4: 0]
AddOperation [recipe=[guimxcvn, Content--guimxcvn, user4], timestamp=user4: 1]
AddOperation [recipe=[jdpmsrot, Content--jdpmsrot, user4], timestamp=user4: 2]
AddOperation [recipe=[prrrbyyq, Content--prrrbyyq, user4], timestamp=user4: 3]
AddOperation [recipe=[nhmswiev, Content--nhmswiev, user4], timestamp=user4: 4]
AddOperation [recipe=[dtuaotah, Content--dtuaotah, user4], timestamp=user4: 5]
AddOperation [recipe=[jfyqttss, Content--jfyqttss, user4], timestamp=user4: 6]
AddOperation [recipe=[xakmtotn, Content--xakmtotn, user4], timestamp=user4: 7]
AddOperation [recipe=[ooqdugv, Content--ooqdugv, user4], timestamp=user4: 8]
AddOperation [recipe=[lvckthrq, Content--lvckthrq, user4], timestamp=user4: 9]
AddOperation [recipe=[pytljyec, Content--pytljyec, user4], timestamp=user4: 10]
AddOperation [recipe=[roppuwew, Content--roppuwew, user4], timestamp=user4: 11]

Your Log:
Summaries are NOT equal and Logs are NOT equal

```

- **Conclusión:** la aplicación realiza las operaciones pero no logra terminar el protocolo TSAE de forma correcta, el mensaje al final de la ejecución “Summaries are NOT equal and Logs are NOT equal” indica que el protocolo TSAE no esta correctamente implementado.



### 3.1.2 Test de funcionamiento, una vez realizadas las implementaciones

Una vez realizadas las implementaciones de los métodos requeridos en las clases Log y TimestampVector, se han vuelto a realizar los mismos test:

#### Test1 – recipes\_service.Server

- **Instrucción para el test:** `java -cp ../bin:../lib/* recipes_service.Server --phase1`
- **Descripción:** el test ejecuta el programa java y pone en marcha un servidor el cual permite realizar ciertas operaciones como añadir recetas, obtener el listado de recetas, ver el contenido de los logs y el contenido de los summary.
- **Resultado:** se puede observar el resultado de varias interacciones con la aplicación en la imagen del terminal.

```
kepa@Belkoain:~/Dokumentuak/Keparenak/UOC/SD/PRA1/2021p_SD_practical_assignment/2021p_SD--students/scripts$ (main) java -cp ../bin:../lib/* recipes_service.Server --phase1

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
1
Enter the title of the recipe to add
recipe1 title
Enter the recipe to add
recipe1

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
3
Recipes:
{recipe1 title=[recipe1 title, recipe1, localhost:9000]}

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
4
Log:
AddOperation [recipe=[recipe1 title, recipe1, localhost:9000], timestamp=localhost:9000: 0]

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
5
Summary:
localhost:9000: 0

Server localhost:9000

Select a command:
1: Add a recipe
3: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
```

- **Conclusión:** la aplicación realiza las operaciones que ofrece el menú de manera correcta, además se observa que implementa el método add de manera correcta, ya que añade acciones al log.

**Test2 – start.sh**

- **Instrucción para el test:** ./start.sh 20004 --phase1
- **Descripción:** el test ejecuta el programa java y trata de realizar las operaciones TSAE, al final compara si los ficheros log y summary de los diferentes servidores son iguales
- **Resultado:** se puede observar el resultado de varias interacciones con la aplicación en la imagen del terminal.

```

Fixstrategia Editatu Ikusi Terminala Fixak Laguntza
user2: 10
user0: 6
user3: 11
user4: 11

15-03-2021 18:57:18:651 Phase_1 [INFO] : Log:
AddOperation [recipe=[mrtsccko, Content--mrtsccko, user1], timestamp=user1: 0]
AddOperation [recipe=[fbybqmkf, Content--fbybqmkf, user1], timestamp=user1: 1]
AddOperation [recipe=[qycxccpi, Content--qycxccpi, user1], timestamp=user1: 2]
AddOperation [recipe=[gsktxxsh, Content--gsktxxsh, user1], timestamp=user1: 3]
AddOperation [recipe=[hlvdoryn, Content--hlvdoryn, user1], timestamp=user1: 4]
AddOperation [recipe=[hauelsfc, Content--hauelsfc, user1], timestamp=user1: 5]
AddOperation [recipe=[kpetxjqj, Content--kpetxjqj, user1], timestamp=user1: 6]
AddOperation [recipe=[yjcauehb, Content--yjcauehb, user1], timestamp=user1: 7]
AddOperation [recipe=[vgrxismv, Content--vgrxismv, user2], timestamp=user2: 0]
AddOperation [recipe=[igutjmas, Content--igutjmas, user2], timestamp=user2: 1]
AddOperation [recipe=[enwxuoaq, Content--enwxuoaq, user2], timestamp=user2: 2]
AddOperation [recipe=[bycjuiap, Content--bycjuiap, user2], timestamp=user2: 3]
AddOperation [recipe=[clibbrbl, Content--clibbrbl, user2], timestamp=user2: 4]
AddOperation [recipe=[rlretsfx, Content--rlretsfx, user2], timestamp=user2: 5]
AddOperation [recipe=[vqkrfrdd, Content--vqkrfrdd, user2], timestamp=user2: 6]
AddOperation [recipe=[srhvddam, Content--srhvddam, user2], timestamp=user2: 7]
AddOperation [recipe=[skotncrs, Content--skotncrs, user2], timestamp=user2: 8]
AddOperation [recipe=[tluqowex, Content--tluqowex, user2], timestamp=user2: 9]
AddOperation [recipe=[fdfyucma, Content--fdfyucma, user2], timestamp=user2: 10]
AddOperation [recipe=[ftyqgryp, Content--ftyqgryp, user0], timestamp=user0: 0]
AddOperation [recipe=[wfvvcioe, Content--wfvvcioe, user0], timestamp=user0: 1]
AddOperation [recipe=[ayxlnxfj, Content--ayxlnxfj, user0], timestamp=user0: 2]
AddOperation [recipe=[jeqkicvn, Content--jeqkicvn, user0], timestamp=user0: 3]
AddOperation [recipe=[gjtvehkq, Content--gjtvehkq, user0], timestamp=user0: 4]
AddOperation [recipe=[hnsfjsrm, Content--hnsfjsrm, user0], timestamp=user0: 5]
AddOperation [recipe=[tustxmqr, Content--tustxmqr, user0], timestamp=user0: 6]
AddOperation [recipe=[naenayif, Content--naenayif, user3], timestamp=user3: 0]
AddOperation [recipe=[paqhlrar, Content--paqhlrar, user3], timestamp=user3: 1]
AddOperation [recipe=[brqcmawf, Content--brqcmawf, user3], timestamp=user3: 2]
AddOperation [recipe=[vribeukc, Content--vribeukc, user3], timestamp=user3: 3]
AddOperation [recipe=[lupjjovh, Content--lupjjovh, user3], timestamp=user3: 4]
AddOperation [recipe=[qbmyxmoe, Content--qbmyxmoe, user3], timestamp=user3: 5]
AddOperation [recipe=[jvuacyfw, Content--jvuacyfw, user3], timestamp=user3: 6]
AddOperation [recipe=[booxoevo, Content--booxoevo, user3], timestamp=user3: 7]
AddOperation [recipe=[cefxgrfy, Content--cefxgrfy, user3], timestamp=user3: 8]
AddOperation [recipe=[nikvppdg, Content--nikvppdg, user3], timestamp=user3: 9]
AddOperation [recipe=[phmfqbsu, Content--phmfqbsu, user3], timestamp=user3: 10]
AddOperation [recipe=[fewjqkrh, Content--fewjqkrh, user3], timestamp=user3: 11]
AddOperation [recipe=[mtxhcsxk, Content--mtxhcsxk, user4], timestamp=user4: 0]
AddOperation [recipe=[guimxcvn, Content--guimxcvn, user4], timestamp=user4: 1]
AddOperation [recipe=[jdpmsrot, Content--jdpmsrot, user4], timestamp=user4: 2]
AddOperation [recipe=[prrrbyyq, Content--prrrbyyq, user4], timestamp=user4: 3]
AddOperation [recipe=[nhmswiev, Content--nhmswiev, user4], timestamp=user4: 4]
AddOperation [recipe=[dtuaotah, Content--dtuaotah, user4], timestamp=user4: 5]
AddOperation [recipe=[jfyqtts, Content--jfyqtts, user4], timestamp=user4: 6]
AddOperation [recipe=[xakmtotn, Content--xakmtotn, user4], timestamp=user4: 7]
AddOperation [recipe=[ooqduvgv, Content--ooqduvgv, user4], timestamp=user4: 8]
AddOperation [recipe=[lvckthrq, Content--lvckthrq, user4], timestamp=user4: 9]
AddOperation [recipe=[pytljyec, Content--pytljyec, user4], timestamp=user4: 10]
AddOperation [recipe=[roppuwew, Content--roppuwew, user4], timestamp=user4: 11]

15-03-2021 18:57:18:651 Phase_1 [INFO] : Summary:
user1: 7
user2: 10
user0: 6
user3: 11
user4: 11

Result is correct: Log and Summary are correct

```

- **Conclusión:** la aplicación realiza las operaciones y logra realizar el protocolo TSAE de forma correcta, el mensaje al final de la ejecución “Result is correct: Log and Summary are correct” indica que el protocolo TSAE esta correctamente implementado.



### 3.2 Test en DSLab

Una vez realizadas las pruebas en el entorno local, se han realizado los test en la herramienta DSLab (<https://sd.uoc.edu/dslab/>). Para realizar el test se ha seguido la serie de instrucciones que indica el enunciado:

1. Crear un proyecto y cargar los ficheros que contienen las clases Log y TimestampVector en el proyecto:

Mostrar PRA1 - Fase1

Profile
Files

User uploaded files

Path	Name	Logs added	Date	Size
src/recipes_service/tsae/data_structures/	<a href="#">TimestampVector.java</a>	No	13:48:32 14-03-2021	4.3 KB
src/recipes_service/tsae/data_structures/	<a href="#">Log.java</a>	No	13:48:32 14-03-2021	5.7 KB

Log files generated by build

Name	Date	Size
<a href="#">log_compile.txt</a>	13:48:42 14-03-2021	1.7 KB
<a href="#">log_common_part.txt</a>	13:48:42 14-03-2021	567 bytes
<a href="#">log_build_project.txt</a>	13:48:42 14-03-2021	2.2 KB

Submit

2. Crear un experimento que ejecute el proyecto

#### Mostrar Submission

Submission with id "35" is running. The duration of the execution will depend on the parameters of the execution. In case of an error, this execution will last a maximum of 6 minutes.

Id 35

User [Kepa Sarasola Bengoetxea \(ksarasola@uoc.edu\)](#)

Assignment Lab session 1

Task Fase 1

Project [PRA1 - Fase1](#)

Submission Date 13:48:52 14-03-2021

State FINISHED

Result Date 13:49:08 14-03-2021

Success ✓

Result Summary Correct: Phase1 Summaries and Logs are equal.

Result [View result details](#)

Logs [View logs](#)

## 3. Verificar los resultados

Correct: Phasel Summaries and Logs are equal.

List of users: [user0, user1, user2, user3, user4]

=====

List of operations:

```
AddOperation [recipe=[yjeokmyg, Content--yjeokmyg, user4], timestamp=user4: 0]
AddOperation [recipe=[ockembbt, Content--ockembbt, user4], timestamp=user4: 1]
AddOperation [recipe=[fiasnglq, Content--fiasnglq, user2], timestamp=user2: 0]
AddOperation [recipe=[lbyahqfr, Content--lbyahqfr, user4], timestamp=user4: 2]
AddOperation [recipe=[wfcyhtkl, Content--wfcyhtkl, user0], timestamp=user0: 0]
AddOperation [recipe=[jsispbxx, Content--jsispbxx, user3], timestamp=user3: 0]
AddOperation [recipe=[kvijarhm, Content--kvijarhm, user0], timestamp=user0: 1]
AddOperation [recipe=[hxdftedy, Content--hxdftedy, user1], timestamp=user1: 0]
AddOperation [recipe=[qqfyfiie, Content--qqfyfiie, user2], timestamp=user2: 1]
AddOperation [recipe=[xmrtxrm, Content--xmrtxrm, user3], timestamp=user3: 1]
AddOperation [recipe=[resdprjk, Content--resdprjk, user3], timestamp=user3: 2]
AddOperation [recipe=[dtacpgmk, Content--dtacpgmk, user1], timestamp=user1: 1]
AddOperation [recipe=[wmktyusa, Content--wmktyusa, user0], timestamp=user0: 2]
AddOperation [recipe=[rklqtdny, Content--rklqtdny, user2], timestamp=user2: 2]
AddOperation [recipe=[sdtqgbvo, Content--sdtqgbvo, user2], timestamp=user2: 3]
AddOperation [recipe=[gtmeyahe, Content--gtmeyahe, user2], timestamp=user2: 4]
AddOperation [recipe=[sfh^niqq, Content--sfh^niqq, user1], timestamp=user1: 2]
AddOperation [recipe=[miehtcdw, Content--miehtcdw, user4], timestamp=user4: 3]
AddOperation [recipe=[bxkihemx, Content--bxkihemx, user2], timestamp=user2: 5]
AddOperation [recipe=[wtrbkjgr, Content--wtrbkjgr, user2], timestamp=user2: 6]
AddOperation [recipe=[leskrdqi, Content--leskrdqi, user1], timestamp=user1: 3]
AddOperation [recipe=[hhrhvkcl, Content--hhrhvkcl, user1], timestamp=user1: 4]
AddOperation [recipe=[kbljrefr, Content--kbljrefr, user4], timestamp=user4: 4]
AddOperation [recipe=[fbjfgijh, Content--fbjfgijh, user4], timestamp=user4: 5]
AddOperation [recipe=[nddiuexe, Content--nddiuexe, user2], timestamp=user2: 7]
AddOperation [recipe=[unxubvvh, Content--unxubvvh, user2], timestamp=user2: 8]
AddOperation [recipe=[moricrux, Content--moricrux, user4], timestamp=user4: 6]
AddOperation [recipe=[biudgtvp, Content--biudgtvp, user3], timestamp=user3: 3]
AddOperation [recipe=[sacffloe, Content--sacffloe, user2], timestamp=user2: 9]
AddOperation [recipe=[mrfxaumy, Content--mrfxaumy, user2], timestamp=user2: 10]
AddOperation [recipe=[jowacuee, Content--jowacuee, user0], timestamp=user0: 3]
AddOperation [recipe=[hcmiejvn, Content--hcmiejvn, user3], timestamp=user3: 4]
```

En esta segunda captura se ve el resultado final del test en DSLab:

The screenshot shows a web browser window with the URL `sd.uoc.edu/dslab_sd/submission/show_result/35`. The page displays a list of 40 'AddOperation' entries, each with a recipe, content, user, and timestamp. Below the list is a 'Summary' section showing counts for users 0 through 4. A 'Previous' button is visible at the bottom right.

```

AddOperation [recipe=cvqesxlj, Content--cvqesxlj, user2], timestamp=user2: 12]
AddOperation [recipe=ajvpalre, Content--ajvpalre, user2], timestamp=user2: 13]
AddOperation [recipe=gxuevpbf, Content--gxuevpbf, user2], timestamp=user2: 14]
AddOperation [recipe=ulodwufe, Content--ulodwufe, user2], timestamp=user2: 15]
AddOperation [recipe=qdcbceke, Content--qdcbceke, user2], timestamp=user2: 16]
AddOperation [recipe=wfcyhtkl, Content--wfcyhtkl, user0], timestamp=user0: 0]
AddOperation [recipe=kvijarhm, Content--kvijarhm, user0], timestamp=user0: 1]
AddOperation [recipe=wmktyusa, Content--wmktyusa, user0], timestamp=user0: 2]
AddOperation [recipe=jowacuee, Content--jowacuee, user0], timestamp=user0: 3]
AddOperation [recipe=ybdnvqgg, Content--ybdnvqgg, user0], timestamp=user0: 4]
AddOperation [recipe=dwmiipfy, Content--dwmiipfy, user0], timestamp=user0: 5]
AddOperation [recipe=aweddbmq, Content--aweddbmq, user0], timestamp=user0: 6]
AddOperation [recipe=qkdnynths, Content--qkdnynths, user0], timestamp=user0: 7]
AddOperation [recipe=xpdpdhao, Content--xpdpdhao, user0], timestamp=user0: 8]
AddOperation [recipe=xbixsbif, Content--xbixsbif, user0], timestamp=user0: 9]
AddOperation [recipe=jsispbxx, Content--jsispbxx, user3], timestamp=user3: 0]
AddOperation [recipe=xmrtxrm, Content--xmrtxrm, user3], timestamp=user3: 1]
AddOperation [recipe=resdprjk, Content--resdprjk, user3], timestamp=user3: 2]
AddOperation [recipe=biudgtvp, Content--biudgtvp, user3], timestamp=user3: 3]
AddOperation [recipe=hcmiejvn, Content--hcmiejvn, user3], timestamp=user3: 4]
AddOperation [recipe=uehpjdab, Content--uehpjdab, user3], timestamp=user3: 5]
AddOperation [recipe=cthcjaow, Content--cthcjaow, user3], timestamp=user3: 6]
AddOperation [recipe=dwedsump, Content--dwedsump, user3], timestamp=user3: 7]
AddOperation [recipe=blxwyarm, Content--blxwyarm, user3], timestamp=user3: 8]
AddOperation [recipe=yjeokmyg, Content--yjeokmyg, user4], timestamp=user4: 0]
AddOperation [recipe=ockembbt, Content--ockembbt, user4], timestamp=user4: 1]
AddOperation [recipe=lbyahqfr, Content--lbyahqfr, user4], timestamp=user4: 2]
AddOperation [recipe=miehtcdw, Content--miehtcdw, user4], timestamp=user4: 3]
AddOperation [recipe=kbljrefr, Content--kbljrefr, user4], timestamp=user4: 4]
AddOperation [recipe=fbjfgijh, Content--fbjfgijh, user4], timestamp=user4: 5]
AddOperation [recipe=moricrux, Content--moricrux, user4], timestamp=user4: 6]
AddOperation [recipe=cvngkgdo, Content--cvngkgdo, user4], timestamp=user4: 7]

=====

Summary:

user1: 5
user2: 16
user0: 9
user3: 8
user4: 7
  
```

Previous

Universitat Oberta de Catalunya - 2020

## 4. Conclusiones

### 4.1 Dificultades

En mi caso la mayor dificultad ha sido la de entender el enunciado y situarme en el proyecto Java, ya que al principio no entendí muy bien cual era el alcance de esta primera fase “Phase1”. Me costo entender que no era necesaria la implementación de todos los métodos vacíos, es decir que no había que implementar todos aquellos métodos que no tenían código implementado.

En cuanto a la resolución del problema planteado por el enunciado, el método más complejo de implementar ha sido *add(Operation opp)*, ya que había que tener en cuenta la restricción de la marca de tiempo, para lo cual había que implementar una solución para obtener la información de tiempo del último mensaje del log.