

Fase 1:

Test 1: A continuación, veremos el resultado `java -cp ../bin:../lib/* recipes_service.Server --phase1` adjunto las pruebas que se han hecho en local donde creamos unos recibos y después comprobamos el log si se ha actualizado, todo esto pasa gracias a los cambios que hemos modificado los métodos de las clases `Log.java` y `TimeStampVector.java`:

Empezamos con la orden `add()` para añadir un nuevo recibo:

```
Ubuntu
root@MacBookPro:/mnt/c/users/musta/desktop/project/scripts# java -cp ../bin:../lib/* recipes_service.Server --phase1
Server localhost:9000
Select a command:
1: Add a recipe
2: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
1
Enter the title of the recipe to add
RecipeOne
Enter the recipe to add
fruits and vegetables, oil
Server localhost:9000
Select a command:
1: Add a recipe
2: Show the list of recipes
4: Show the Log
5: Show the summary
0: Exit
10-11-2025 17:01:45:381 Phase1_TS4E [TRACE] : Log.add: Added first operation (seq 0) from localhost:9000
10-11-2025 17:01:45:385 Phase1_TS4E [TRACE] : Updating the TimestampVectorInserting with the timestamp: localhost:9000: 0
1
Enter the title of the recipe to add
RecipeTwo
Enter the recipe to add
wine, coke,
Server 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
10-11-2025 17:02:29:341 Phase1_TS4E [TRACE] : Log.add: Added sequential operation from localhost:9000: localhost:9000: 1
10-11-2025 17:02:29:341 Phase1_TS4E [TRACE] : Updating the TimestampVectorInserting with the timestamp: localhost:9000: 1
4
Log:
AddOperation [recipe=[RecipeOne, fruits and vegetables, oil, localhost:9000], timestamp=localhost:9000: 0]
AddOperation [recipe=[RecipeTwo, Wine, Coke,, localhost:9000], timestamp=localhost:9000: 1]

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:
[RecipeOne=[RecipeOne, fruits and vegetables, oil, localhost:9000], RecipeTwo=[RecipeTwo, Wine, Coke,, localhost:9000]]
```

Test 2: El Segundo test se trata de una lista predefinida de un conjunto de usuarios y sus operaciones y la compara con la lista anteriormente calculada ./start.sh 20004 --phase1 el 20004 es el puerto de escucha que tenemos habilitado

```
recipes_service.test.Phase1TestServer
COMPARING summary and log ...

10-11-2025 17:06:49:244 Phase_1 [INFO] : List of users: [user0, user1, user2, user3, user4]
10-11-2025 17:06:49:250 Phase_1 [INFO] : List of operations:
[AddOperation [recipe=[ftyqqrp, Content--ftyqqrp, user0], timestamp=user0: 0], AddOperation [recipe=[vrxismw, Content--vrxismw, user2], timestamp=user2: 0], AddOperation [recipe=[mrtsccko, Content--mrtsccko, user1], timestamp=user1: 0], AddOperation [recipe=[mtxhcsxk, Content--mtxhcsxk, user4], timestamp=user4: 0], AddOperation [recipe=[tigtjm3as, Content--tigtjm3as, user2], timestamp=user2: 1], AddOperation [recipe=[enwxuoaq, Content--enwxuoaq, user2], timestamp=user2: 2], 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=[wfvfcloe, Content--wfvfcloe, user4], timestamp=user0: 1], AddOperation [recipe=[jdpmrsot, Content--jdpmrsot, user4], timestamp=user4], AddOperation [recipe=[bycjtup, Content--bycjtup, user2], timestamp=user2: 3], AddOperation [recipe=[prrbyyq, Content--prrbyyq, user4], timestamp=user4: 3], AddOperation [recipe=[ayxlnxfj, Content--ayxlnxfj, user0], timestamp=user0: 2], AddOperation [recipe=[qycxcpi, Content--qycxcpi, user1], timestamp=user1: 2], AddOperation [recipe=[jeqkicvn, Content--jeqkicvn, user0], timestamp=user0: 3], AddOperation [recipe=[gjtvrekq, Content--gjtvrekq, user1], timestamp=user1: 3], AddOperation [recipe=[hnsfjsrm, Content--hnsfjsrm, user0], timestamp=user0: 4], AddOperation [recipe=[rletsfx, Content--rletsfx, user2], timestamp=user2: 5], AddOperation [recipe=[rletsfx, Content--rletsfx, user4], timestamp=user4: 5], AddOperation [recipe=[dtaotah, Content--dtaotah, user4], timestamp=user4: 5], AddOperation [recipe=[vqkrfrdd, Content--vqkrfrdd, user2], timestamp=user2: 6], AddOperation [recipe=[hldvrdyn, Content--hldvrdyn, user1], timestamp=user1: 4], AddOperation [recipe=[srhvddan, Content--srhvddan, user2], timestamp=user2: 7], AddOperation [recipe=[jfytts, Content--jfytts, user1], timestamp=user1: 6], AddOperation [recipe=[tustxmqr, Content--tustxmqr, user0], timestamp=user0: 6], AddOperation [recipe=[vribekc, Content--vribekc, user3], timestamp=user3: 3], AddOperation [recipe=[xakmtotn, Content--xakmtotn, user4], timestamp=user4: 7], AddOperation [recipe=[skotn, Content--skotn, user2], timestamp=user2: 8], AddOperation [recipe=[tluquowex, Content--tluquowex, user2], timestamp=user2: 9], AddOperation [recipe=[ooqduvgv, Content--ooqduvgv, user4], timestamp=user4: 8], AddOperation [recipe=[qbmyxmeo, Content--qbmyxmeo, user3], timestamp=user3: 5], AddOperation [recipe=[hauelsfc, Content--hauelsfc, user1], timestamp=user1: 5], AddOperation [recipe=[jvucacyfw, Content--jvucacyfw, user3], timestamp=user3: 6], AddOperation [recipe=[kpetxjqi, Content--kpetxjqi, user1], timestamp=user1: 6], AddOperation [recipe=[fdffvucma, Content--fdffvucma, user2], timestamp=user2: 10], AddOperation [recipe=[lvckthrq, Content--lvckthrq, user4], timestamp=user4: 9], AddOperation [recipe=[booxoevo, Content--booxoevo, user3], timestamp=user3: 7], AddOperation [recipe=[pytljyec, Content--pytljyec, user4], timestamp=user4: 10], AddOperation [recipe=[cefxgrfy, Content--cefxgrfy, user3], timestamp=user3: 8], AddOperation [recipe=[nikvpddg, Content--nikvpddg, user3], timestamp=user3: 9], AddOperation [recipe=[phmfgbsu, Content--phmfgbsu, user3], timestamp=user3: 10], AddOperation [recipe=[fewjqkrh, Content--fewjqkrh, user3], timestamp=user3: 11], AddOperation [recipe=[lupjjovh, Content--lupjjovh, user3], timestamp=user3: 4], AddOperation [recipe=[qbmyxmeo, Content--qbmyxmeo, user3], timestamp=user3: 5], AddOperation [recipe=[jvucacyfw, Content--jvucacyfw, user3], timestamp=user3: 6], AddOperation [recipe=[booxoevo, Content--booxoevo, user3], timestamp=user3: 7], AddOperation [recipe=[cefxgrfy, Content--cefxgrfy, user3], timestamp=user3: 8], AddOperation [recipe=[nikvpddg, Content--nikvpddg, user3], timestamp=user3: 9], AddOperation [recipe=[phmfgb-su, Content--phmfgb-su, user3], timestamp=user3: 10], AddOperation [recipe=[fewjqkrh, Content--fewjqkrh, user3], timestamp=user3: 11], AddOperation [recipe=[vribekc, Content--vribekc, user3], timestamp=user3: 3], 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=[jfytts, Content--jfytts, user4], timestamp=user4: 6], AddOperation [recipe=[dtuaotah, Content--dtuaotah, user4], timestamp=user4: 5], AddOperation [recipe=[nhswiev, Content--nhswiev, user4], timestamp=user4: 4], AddOperation [recipe=[dtuaotah, Content--dtuaotah, user4], timestamp=user4: 5], AddOperation [recipe=[jfytts, Content--jfytts, 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]

10-11-2025 17:06:49:278 Phase_1 [TRACE] : Log: add: Added first operation (seq 0) from user0
10-11-2025 17:06:49:278 Phase_1 [TRACE] : Log updated:
AddOperation [recipe=[ftyqqrp, Content--ftyqqrp, user0], timestamp=user0: 0]
10-11-2025 17:06:49:278 Phase_1 [TRACE] : Summary updated:
```

```
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=[gjtvrekq, Content--gjtvrekq, 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=[vribekc, Content--vribekc, user3], timestamp=user3: 3]
AddOperation [recipe=[lupjjovh, Content--lupjjovh, user3], timestamp=user3: 4]
AddOperation [recipe=[qbmyxmeo, Content--qbmyxmeo, user3], timestamp=user3: 5]
AddOperation [recipe=[jvucacyfw, Content--jvucacyfw, user3], timestamp=user3: 6]
AddOperation [recipe=[booxoevo, Content--booxoevo, user3], timestamp=user3: 7]
AddOperation [recipe=[cefxgrfy, Content--cefxgrfy, user3], timestamp=user3: 8]
AddOperation [recipe=[nikvpddg, Content--nikvpddg, user3], timestamp=user3: 9]
AddOperation [recipe=[phmfgb-su, Content--phmfgb-su, 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=[jdpmrsot, Content--jdpmrsot, user4], timestamp=user4: 2]
AddOperation [recipe=[prrbyyq, Content--prrbyyq, user4], timestamp=user4: 3]
AddOperation [recipe=[nhswiev, Content--nhswiev, user4], timestamp=user4: 4]
AddOperation [recipe=[dtuaotah, Content--dtuaotah, user4], timestamp=user4: 5]
AddOperation [recipe=[jfytts, Content--jfytts, 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]
```

```
10-11-2025 17:06:49:307 Phase_1 [INFO] : Summary:
user1: 7
user2: 10
user0: 6
user3: 11
user4: 11
```

```
Result is correct: Log and Summary are correct
root@MacBookPro:/mnt/c/users/musta/Desktop/project/scripts# █
```



Preguntas de reflexión: No soy un programador muy experimentado, pero entiendo lo suficiente sobre DPOO y otros lenguajes de programación que me han ayudado a completar esta práctica, empezamos.

1. ¿Qué has aprendido de este proyecto?

La verdad es que he aprendido bastante, sobre todo lo difícil que es revisar los métodos de una clase y con poca información. Si que es verdad que el enunciado decía que tuviéramos cuidado con la concurrencia, y ahora entiendo por qué, tuve que usar `synchronized` en el método `add` de `Log.java` para que no se rompa si dos cosas pasan a la vez.

Otra cosa que me costó entender, pero creo que ya lo tengo, es cómo el `TimestampVector` es un resumen del `Log`. Al principio no veía la conexión, pero ahora entiendo que cada vez que añades algo al `Log`, tienes que actualizar el `TimestampVector` para que se registre. En general es difícil pasar la teoría del libro a algo real pero creo que lo he conseguido.

2. ¿Qué parte fue la más desafiante y cómo la superaste?

Para mí, la parte más difícil fue hacer que el código compilara. Me pasé muchísimo tiempo por no decir días con errores en las clases `Log.java` y `TimestampVector.java` porque no entendía cómo funcionaba la clase `Timestamp.java`. Intentaba buscar un método como `getSeqNumber()` para conseguir el número de secuencia, pero no existía y lo peor es que Eclipse no es un IDE al que estoy acostumbrado y no entendía porque me estaba dando tantos errores aparte de los obvios.

Estuve atascado hasta que buscando en algunas prácticas de años anteriores de la uoc, volví a leer `Timestamp.java` línea por línea y indagué en otras clases relacionadas y al final me di cuenta de que no existía el `getSeqNumber()`, y que tenía que usar el método `compare()` para ver si un `timestamp` era más nuevo o viejo, y `getHostid()` para saber de qué réplica era.

Cuando superé eso, el otro desafío fue el método `add` de `Log.java`, para asegurarme de que solo añadía la operación si era exactamente la siguiente (con `compare() == 1`) y no una repetida.

3. ¿Qué concepto no comprendiste completamente y te gustaría explorar más a fondo?

No entiendo para qué sirve la clase `TimestampMatrix.java`.

El enunciado la menciona en la lista, pero para esta fase (solo operaciones `add` y sin purga) no la hemos usado para nada. Vi que el `Log.java` tiene un método `purgeLog` que la usa, así que supongo que es para eso, pero ahora mismo no tengo ni idea de cómo funciona ese mecanismo de "Acknowledgements" (ACKs) ni cómo sabe la matriz qué es lo que han visto todas las réplicas.

4. Si tuvieras que repetir el proyecto, ¿qué cambiarías o mejorarías?

Lo que haría diferente está clarísimo: me habría ahorrado muchos dolores de cabeza si hubiera leído todos los ficheros "auxiliares" primero.

Empecé a modificar el `Log.java` directamente, asumiendo que `Timestamp.java` funcionaría de alguna manera, y estaba equivocado. Si tuviera que empezar desde cero, lo primero que haría sería abrir y entender la función de las clases que mencionan `Timestamp.java` y

Operation.java antes de escribir una sola línea de código en las clases que tenía que entregar. Habría terminado mucho más rápido.