

---

This is the **published version** of the bachelor thesis:

Farrés de Bunes, Albert; Ramos Terrades, Oriol, dir. Arquitectura master-slave i multiMaster en Oracle. 2021. (958 Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/248532>

under the terms of the  license

# Arquitectura Master-Slave i Multi-Master en Oracle

Albert Farrés de Bunes

**Resum**—Aquest projecte buscava estudiar els tipus de sistemes distribuïts disponibles al sistema de gestió de bases de dades relacionals Oracle Database. Aquest potent sistema de gestió de bases de dades és el més utilitzat mundialment i permet administrar diferents tipus de sistemes distribuïts. L'objectiu principal d'aquest estudi és aplicar les arquitectures de replicació Master-slave i Multi-master al sistema Oracle Database utilitzant els mètodes que el sistema proporciona. Aquestes arquitectures són tipus d'estructures de replicació de dades en què es poden basar els sistemes distribuïts. Aquests sistemes garanteixen una major disponibilitat i seguretat a les dades, i augmenten el rendiment del sistema.

**Paraules clau**—Oracle Database, base de dades relacional, sistemes distribuïts, sistemes de replicació, DataGuard, màquines virtuals, entorn de replicació, arquitectura Multi-master, arquitectura Master-slave, Standby físic, Standby lògic.

**Abstract**—This project sought to study the different types of distributed systems available in the Oracle Database relational database management system. This powerful database management system is the most widely used worldwide and allows you to manage different types of distributed systems. The main goal of this study is to apply the Master-slave and Multi-master replication architectures to the Oracle Database system using the methods provided by the database management system (DMBS). These architectures are types of data replication structures on which distributed systems can be based. These systems ensure greater availability and security of data, and increase system performance.

**Index Terms**—Oracle Database, relational database, distributed systems, replication systems, DataGuard, virtual machines, replication environment, Multi-master architecture, Master-slave architecture, Physical Standby, Logical Standby.



## 1 INTRODUCCIÓ

AQUEST projecte té com a objectiu principal estudiar els mètodes de replicació de què disposa actualment el sistema Oracle i aplicar els seus sistemes distribuïts a un clúster de servidors.

La replicació de bases de dades és un procés que permet copiar i distribuir dades d'una base de dades a un altre. D'aquesta manera s'aconsegueix sincronitzar-les i mantenir la consistència i integritat de les seves dades. Tenir dues bases de dades sincronitzades té beneficis com que els usuaris puguin accedir a les dades des de dos servidors diferents, augmentant la disponibilitat de les dades. Un altre benefici és la capacitat de recuperació de les dades, ja que en cas que caigui un servidor, es poden recuperar les dades de l'altre base de dades, i el servei pot també seguir en funcionament.

De tots els sistemes de gestió de bases de dades disponibles actualment, s'ha utilitzat Oracle Database per les següents raons. És el motor de base de dades relacionals més utilitzat mundialment. És un software multiplataforma, es pot executar en multitud de sistemes, des d'un portàtil a un supercomputador. Es pot executar en la majoria de sistemes operatius (Linux, Mac, Windows...). Permet l'ús de particions per millorar l'eficiència i permet administrar bases de dades distribuïdes.

Els inconvenients d'un sistema tan potent són el seu nivell de dificultat per configurar-lo correctament. La seva documentació és molt extensa i suposa usar molt temps per entendre i utilitzar la seva informació. Destaco aquest punt perquè a l'inici del projecte no es tenia cap experiència amb el sistema Oracle, i hi havia una incògnita sobre les múltiples opcions de replicació que ofereix.

### OBJECTIUS

En un inici, el primer objectiu del projecte va ser estudiar els mètodes de replicació i els sistemes distribuïts d'Oracle, donada la dificultat del sistema i la multitud d'opcions de replicació de què disposa. El segon objectiu va ser desplegar aquests sistemes al clúster de servidors del departament d'informàtica de la Universitat Autònoma de Barcelona utilitzant contenidors Docker. Els tipus de sistemes distribuïts que es proposava implementar van ser l'arquitectura Master-slave i l'arquitectura Multi-master. A l'apartat "estat de l'art" es detalla el funcionament d'aquest tipus de sistemes.

Abans d'implementar els sistemes al clúster de la universitat es va proposar d'aplicar-los primer sobre un entorn controlat i assegurar el seu funcionament. Es prioritza l'arquitectura Master-slave per tenir un funcionament teòricament més senzill que l'arquitectura Multi-master. Donada la incògnita i la dificultat dels mètodes de replicació d'un sistema tan potent i complex com Oracle Database, es va decidir relegar el desplegament dels sistemes distribuïts al clúster de la universitat a objectius secundaris. Així doncs els objectius del projecte són:

- E-mail de contacte: [albert.farresd@e-campus.uab.cat](mailto:albert.farresd@e-campus.uab.cat)
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Oriol Ramos Terrades (Departament de Ciències de la Computació)
- Curs 2020/21

## OBJECTIUS PRINCIPALS

- O1. Estudiar els sistemes distribuïts d'Oracle Database i els mètodes de replicació disponibles.
- O2. Implementar l'arquitectura Master-slave sobre un entorn controlat.
- O3. Implementar l'arquitectura Multi-master sobre un entorn controlat.

## OBJECTIUS SECUNDARIS

- O4. Desplegar el sistema Multi-master sobre el clúster de servidors de la universitat.
- O5. Desplegar el sistema Multi-master sobre el clúster de servidors de la universitat.

L'avaluació dels resultats i els procediments obtinguts en el desenvolupament del treball es troba documentada al dossier del projecte.

## 2 ESTAT DE L'ART

Les arquitectures Master-slave i Multi-master són els tipus de sistemes distribuïts que es desplegaran en aquest projecte. L'arquitectura de replicació Master-slave té com a base un servidor anomenat master, una base de dades principal on es modifiquen les dades. Aquest master està connectat a un nombre a un nombre de servidors alternatius anomenats esclaus, que són còpies de la base de dades principal, i repliquen les modificacions que s'hi produeixen. Les bases de dades alternatives només s'hi pot accedir per llegir-ne les dades, no es poden modificar, ja que tots els canvis que s'hi realitzin venen des del servidor master (figura 1).

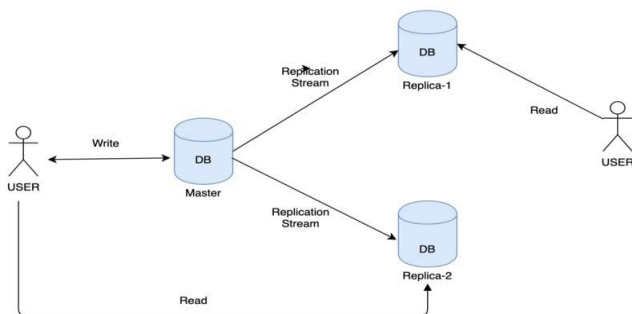


Fig. 1: Estructura Arquitectura Master-slave

Per altra banda, en l'arquitectura de replicació Multi-master tots els servidors tenen la mateixa importància al sistema. Tots els servidors posseeixen la mateixa base de dades, que es pot llegir i modificar. Els canvis realitzats a cada base de dades es propaguen a tots els altres servidors del clúster (figura 2). El fet que tots els servidors puguin propagar canvis promou els conflictes de dades, fet que augmenta la dificultat d'aquesta arquitectura vers la Master-slave.

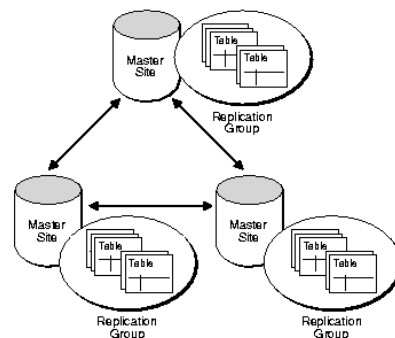


Fig. 2: Estructura Arquitectura Multi-master

Al sistema Oracle Database existeixen diferents mètodes o opcions per aconseguir replicar una arquitectura Master-Slave i Multi-master en un clúster de servidors. Depenent de la versió d'Oracle hi ha disponibles uns o altres mètodes de replicació. La versió més utilitzada actualment és la 19c. Aquesta versió és essencialment igual que la versió 12c però amb algunes modificacions poc rellevants pels objectius del projecte. En aquest cas s'ha optat per utilitzar la versió Oracle Database 12c, ja que és una de les versions més senzilles d'obtenir, i és compatible amb tecnologies de contenidors Docker i màquines virtuals Vagrant. Aquesta versió actualitza sistemes de replicació obsolets de versions anteriors i afegeix noves funcionalitats utilitzades actualment en la gestió de bases de dades relacionals [1][2]. A diferència de la versió 19c, la versió 12c es troba disponible al repositori oficial dels contenidors Docker. Per tant, s'ha decidit utilitzar aquesta versió degut a la seva millora de disponibilitat.

Un cop triada la versió Oracle Database 12c, es van estudiar les opcions de replicació que ofereix el sistema. Oracle Database ofereix múltiples opcions de replicació de dades. Amb la versió 12c, algunes d'aquestes opcions com Oracle CDC, Oracle Streams i Advanced Replication queden obsoletes per donar pas a l'opció més popular i potent Oracle GoldenGate [3]. Aquesta opció és la ideal per aplicar les arquitectures de replicació. Tot i ser difícil d'utilitzar és el mètode de replicació més eficient de què disposa Oracle. L'inconvenient és que es tracta d'un mètode amb un preu molt elevat, pensat per companyies i producció. Per aquest factor l'opció queda descartada pel projecte. L'alternativa que la versió 12c segueix donant suport és la tecnologia Dataguard, que ve inclosa a la versió descarregable gratuïtament. Aquest software no és tan popular, complex ni potent com Oracle GoldenGate, però és gratuït i segueix tenint el suport d'Oracle actualment. Per tant, és l'opció utilitzada per aquest treball.

Dataguard assegura una alta disponibilitat i protecció de les dades, i opcions de recuperació enfront de caigudes de servidors [4]. Ofereix un conjunt de serveis que administren els servidors de bases de dades alternatives que s'utilitzen per recuperar dades o recuperar el servidor principal d'una caiguda. Aquestes bases de dades alternatives s'anomenen Standby, i són còpies de la base de dades principal creades a partir d'una còpia de seguretat (backup). Aquests servidors alternatius Standby reben els canvis que es produeixen a la base de dades primària en forma de registre de canvis (redo logs), i apliquen els canvis a la seva base de dades. D'aquesta manera les bases de dades primària i Standby queden sincronitzades.

Existeixen dos tipus de Standby, el Standby físic i el Standby lògic [5][6]. El sistema de replicació Standby físic utilitza la tecnologia Redo Apply (figura 3). Aquesta tecnologia envia el registre de canvis del servidor primari al servidor Standby. Llavors aquest

servidor esclau utilitza el servei de recuperació de dades d'Oracle Database per, aplicar el registre de canvis que rep del servidor master. D'aquesta manera es repliquen els canvis guardats al registre del servidor primari al servidor Standby, i aquest servidor es manté sincronitzat i idèntic al servidor master. Els usuaris poden accedir al servidor alternatiu Standby només per realitzar operacions de lectura. No permet operacions d'escriptura perquè es troba en fase de recuperació de dades (replicant dades del servidor primari) en tot moment.

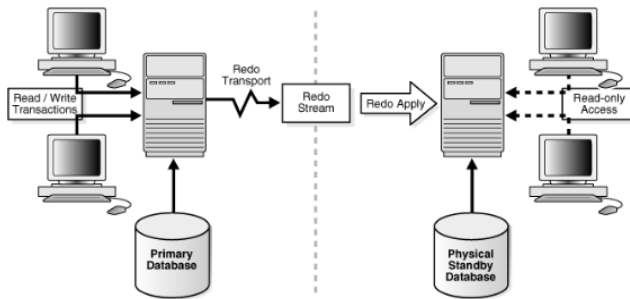


Fig. 3: Estructura Sistema de Replicació Standby Física

El segon sistema de replicació Standby lògic és una millora del sistema Standby físic, però que utilitza una tecnologia diferent (SQL Apply). Aquesta tecnologia transforma el registre de canvis rebut de la base de dades primària en comandes SQL i les aplica a la base de dades Standby lògica (figura 4). En aplicar-se els canvis com a transaccions SQL i no utilitzant un procés de recuperació (Standby físic), la base de dades secundària (slave) pot estar oberta a modificacions. D'aquesta manera es pot utilitzar la base de dades Standby per guardar altres dades a part de les rebudes pel servidor primari. Amb aquestes característiques el sistema lògic millora el funcionament del Standby físic.

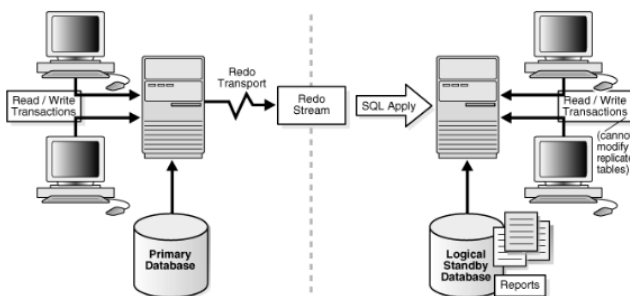


Fig. 4: Estructura Sistema de Replicació Standby Lògic

En aquest treball s'han implementat els dos tipus de sistemes de replicació Standby per assolir l'objectiu de replicar les arquitectures Master-slave i Multi-master a Oracle respectivament. Amb l'estudi realitzat fins ara es pretén utilitzar el sistema Standby físic per aplicar l'arquitectura de replicació Master-slave i el sistema Standby lògic per aplicar l'arquitectura de replicació Multi-master.

### 3 METODOLOGIA

Per determinar la metodologia de treball emprada en el desenvolupament del projecte s'ha tingut en compte l'ús de metodologies àgils, per poder gestionar el treball amb una visió global dels objectius i tasques a realitzar. Ja que inicialment no es coneixia la dificultat que proposava el projecte ni els canvis o modificacions que es durien a terme, es va optar per una metodologia que pogués adaptar-se ràpidament als canvis.

Un altre aspecte que es va tenir en compte és el caràcter individual del projecte. Per tant, es van descartar metodologies àgils de treball en equip com Scrum. Finalment es va optar per l'ús de kanban, perquè no requereix rols i és força senzilla d'utilitzar.

Aquesta metodologia permet la creació de tasques que formen el total del projecte. A mesura que avança el projecte l'estat de les tasques varia i es poden agrupar i organitzar segons ens sembli oportú. Això permet observar el flux de treball en tot moment i tenir una visió general de l'evolució del projecte.

S'ha utilitzat l'eina online gratuïta KanbanTool, que és un administrador de tasques en format kanban [7]. Aquesta eina permet categoritzar les tasques i programar-les al temps en una interfície gràfica. A mesura que es treballa en les tasques del projecte el seu estat canvia de "Pendants" a "En Desenvolupament" a "Finalitzades" (figura 5).

Durant el desenvolupament dels sistemes de replicació es van anar fent paral·lelament avaluacions i proves sobre els resultats obtinguts, per tenir un seguiment a temps real de les implementacions dels sistemes. Aquestes avaluacions i proves es troben explicades més endavant a l'apartat de resultats del projecte.

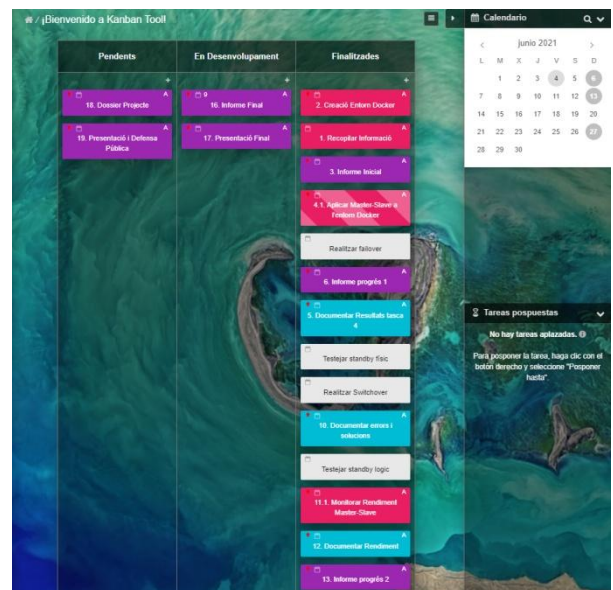


Fig. 5: Planificació Projecte amb KanbanTool

## 4 PLANIFICACIÓ

La planificació del desenvolupament del projecte ha seguit una evolució donats els canvis realitzats al projecte a mesura que avançava. Després de les primeres setmanes d'estudi del sistema Oracle Database, es van descartar els objectius d'implementar els sistemes al clúster de servidors de la universitat (Objectius 4 i 5). Es va arribar a aquesta decisió després d'estudiar els mètodes de replicació d'Oracle Database 12c i la dificultat i el temps d'implementació que comportaven (Objectiu 1).

També es va modificar la prioritat dels sistemes de replicació a implementar després de la investigació. El sistema Multi-master era molt més complexa d'implementar que el sistema Master-slave, així que es va prioritzar la implementació d'aquest últim. Llavors els objectius que derivaren d'aquests arguments quedaren en el següent ordre de prioritació:

01. Estudiar els sistemes distribuïts d'Oracle Database i els mètodes de replicació disponibles.

02. Implementar el sistema de replicació Standby Físic.

03. Implementar el sistema de replicació Standby Lògic.

La següent planificació organitzada en tasques és la planificació definitiva desenvolupada a un estat avançat del projecte. Les tasques han estat dividides en fases del projecte que representen punts importants del desenvolupament (figura 6).

Tasques	Data Inicial	Duració (Setmanes)
<b>Aplicar Arquitectures en un Entorn Controlat</b>		
1. Estudi sistemes distribuïts d'Oracle.	15/02/2021	3
2. Creació entorn màquines virtuals.	01/03/2021	2
3. Informe inicial.	08/03/2021	1
4. Aplicar sistema Standby físic.	08/03/2021	3
5. Aplicar sistema Standby lògic.	29/03/2021	2
6. Documentar resultats tasca 4.	12/04/2021	1
<b>Provar Arquitectura i Assegurar el seu Funcionament</b>		
7. Informe de progrés 1.	12/04/2021	1
8. Testejar sistemes Standby.	19/04/2021	2
9. Efectuar Failover.	03/05/2021	1
10. Efectuar Switchover.	10/05/2021	1
11. Documentar proves failover i switchover.	10/05/2021	2
12. Informe de progrés 2.	17/05/2021	1
13. Documentar errors i solucions de les proves.	24/05/2021	1
14. Document Multi-master.	31/05/2021	1
<b>Documentació Final i Presentació del Projecte</b>		
15. Article final.	31/05/2021	4
16. Presentació final.	21/06/2021	2
17. Dossier del projecte.	28/06/2021	1
18. Presentació i defensa pública.	05/07/2021	1

Fig. 6: Taula Tasques de la Planificació del Projecte

A l'apèndix (A1) es troba el diagrama de Gantt amb les tasques del projecte i els seus períodes de temps representats gràficament.

## 5 DESENVOLUPAMENT

En aquesta secció s'explica en detall el procés de desenvolupament del projecte. Inicialment s'explica tot el procediment de creació de l'entorn de treball on s'aplicaran els sistemes de replicació. En l'apartat Arquitectura Master-slave es descriu el sistema de replicació Standby físic i la seva implementació. A l'últim apartat Arquitectura Multi-master s'explica el sistema de replicació Standby lògic i també la seva implementació.

### 5.1. ENTORN DE TREBALL

Un cop estudiades les tecnologies de replicació que proporciona Oracle Database 12c, s'havia de preparar l'entorn de treball on s'aplicarien els sistemes. Aquest entorn de treball consta de dos entorns diferenciats. Aquests entorns són iguals, però un serveix per implementar el sistema Standby físic i l'altre pel sistema Standby lògic. Cada un d'aquests entorns havia d'estar format per dos servidors Oracle Database 12c, ja que és el mínim necessari per implementar les arquitectures. La idea inicial va ser crear aquests entorns mitjançant contenidors Docker [8], i això va derivar a problemes de compatibilitat entre Oracle Database i Docker. La tecnologia Data Guard que s'utilitzaria per a la replicació de les arquitectures no és compatible amb la imatge de Docker d'Oracle Database 12c. Donat aquest escenari es va optar per utilitzar màquines virtuals en comptes de contenidors.

Per crear un entorn de màquines virtuals es va proposar utilitzar l'eina Vagrant, que permet generar entorns de desenvolupament de forma senzilla, fàcils de configurar i compartir [9]. Aquesta opció portava a problemes de configuració difícils de resoldre no relacionats amb els objectius d'aquest projecte, així que també es va descartar. Aquests problemes de compatibilitat amb els contenidors i Vagrant van contribuir a descartar els objectius secundaris de desplegament dels sistemes de replicació al clúster de servidors de la universitat (objectius 4 i 5), que requerien aquests tipus de tecnologies de desplegament.

Finalment es van crear els dos entorns de dos servidors cadascun amb el sistema Oracle Database 12c utilitzant màquines virtuals configurades manualment amb VirtualBox [10]. Amb aquest programa no hi varen haver problemes d'instal·lació ni configuració. Un cop preparats els entorns el següent pas era aplicar els sistemes de replicació Standby als servidors dels entorns utilitzant la tecnologia Data Guard.

### 5.2. ARQUITECTURA MASTER-SLAVE: STANDBY FÍSIC

Un cop preparat l'entorn, que consta de dos servidors amb el sistema Oracle Database 12c, es va procedir a implementar el sistema Standby físic que replica l'arquitectura Master-slave. En aquest sistema, un dels dos servidors és la base de dades primària (master), i l'altre és la base de dades standby (slave). Entre els dos servidors existeix una comunicació on el servidor primari envia el registre de canvis al servidor Standby, i aquest els aplica a mesura que els rep a la seva base de dades mitjançant un procés de recuperació de la base de dades. D'aquesta manera el servidor Standby és una còpia exacta de la base de dades primària. Com ja s'ha explicat no es podrà modificar la base de dades alternativa mentre repliqui les dades, ja que està constantment executant el procés de recuperació de dades, però s'hi podrà accedir per fer operacions de lectura.

## IMPLEMENTACIÓ

A continuació es descriu el procés d'implementació del sistema Standby físic (figura 7):

### Configuració de la base de dades Primària

1. Preparar registre de canvis: per defecte Oracle no registra tots els canvis que es produeixen a la base de dades. Així doncs, inicialment habilitem els modes necessaris (archive mode i force logging) per guardar tots els canvis que s'efectuin al sistema un cop desplegat.
2. Definir sistema distribuït al servidor primari: es modifiquen diversos paràmetres i arxius (tnsnames) de la base de dades per definir el sistema Standby físic. En aquest pas es crea la relació entre el servidor master i el servidor esclau que disposem.

### Configurar el servidor Standby Físic

3. Preparació servidor Standby: s'inicialitzen els directoris i paràmetres necessaris (arxiu spfile) pel posterior funcionament de la base de dades del servidor.
4. Contrasenyes del sistema distribuït: s'utilitza una còpia de l'arxiu de contrasenyes del servidor primari al servidor Standby. D'aquesta manera podrem accedir amb les mateixes claus als dos servidors.
5. Definir sistema distribuït al servidor Standby: es modifiquen diversos paràmetres i arxius (tnsnames) per establir la relació cap al servidor primari.

### Configurar la base de dades al servidor Standby Físic

6. Inicialitzar base de dades secundària: S'inicialitza la base de dades al servidor Standby físic, que de moment serà buida.
7. Duplicar la base de dades del servidor primari: es duplica el contingut de la base de dades del servidor primari a la base de dades del servidor Standby físic utilitzant el gestor de recuperació d'Oracle (RMAN [11]).
8. Iniciar replicació: un cop copiada la base de dades del servidor primari i configurada la connexió entre servidors, obrir la base de dades (mode lectura) i iniciar el procés de recuperació de Data Guard que replicarà les dades del servidor primari.

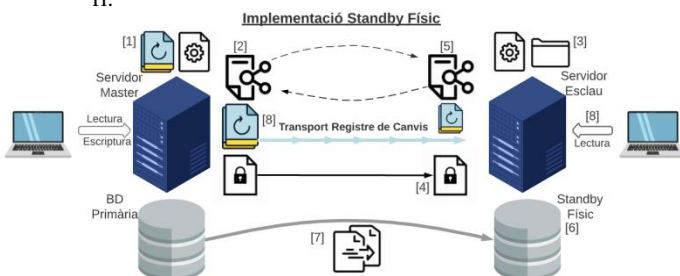


Fig. 7: Implementació Standby Físic

Durant el procés d'implementació del sistema Standby físic es va determinar que aquest sistema aplica correctament l'estructura de replicació Master-slave tal com s'havia previst. La versió detallada

d'aquesta implementació es troba documentada al següent document del dossier del projecte [11].

### 5.3. ARQUITECTURA MULTI-MASTER: STANDBY LÒGIC

El sistema Standby lògic es va implementar al segon entorn creat amb màquines virtuals. Es va implementar aquest sistema amb la intenció de replicar l'arquitectura de replicació Multi-master, ja que compartia característiques com la modificació de tots els servidors. La tecnologia d'aquest sistema (SQL Apply) és diferent de la tecnologia que utilitza el Standby físic (Redo Apply). Recordem que el Standby lògic converteix les dades rebudes del servidor primari en transaccions SQL i les aplica a la base de dades alternativa Standby lògica. A diferència del Standby físic, no es troba executant un procés de recuperació constantment. D'aquesta manera també es pot modificar la base de dades alternativa i utilitzar per a altres operacions.

### PASSOS PREVIS A LA IMPLEMENTACIÓ

Abans de començar amb la implementació del Standby lògic s'han de realitzar els següents passos previs:

1. Crear Standby Físic: el sistema Standby lògic és una modificació d'un Standby físic. Per tant, per obtenir aquest sistema lògic abans s'ha de crear un Standby físic utilitzant els passos descrits a l'apartat anterior.
2. Revisar les dades: existeixen diferents tipus de dades que el sistema Oracle no replica en un Standby lògic. Per tant, s'ha d'assegurar que no hi ha aquest tipus de dades. En cas de trobar-ne s'haurien d'eliminar o modificar perquè es puguin replicar.

### IMPLEMENTACIÓ

A continuació es descriu el procés d'implementació del Standby lògic (figura 8):

1. Habilitar conversió del registre de canvis: es construeix un diccionari al servidor primari que permet interpretar el contingut del registre de canvis. D'aquesta manera el registre de canvis que el servidor esclau rebra del master es convertirà en operacions de la base de dades.
2. Conversió de la base de dades: es converteix la base de dades Standby física en Standby lògica utilitzant instruccions d'Oracle.
3. Aplicar el procés de replicació: Un cop convertida la base de dades en Standby lògic, s'obre la base de dades i s'inicia el procés de replicació de dades.

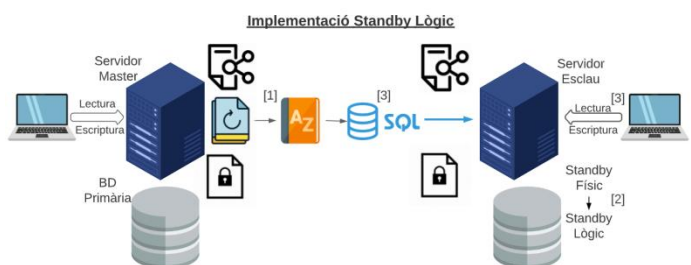


Fig. 8: Implementació Standby Lògic

Durant el procés d'implementació del Standby lògic es va determinar que aquest sistema no es pot utilitzar com una rèplica de l'arquitectura de replicació Multi-master. Durant l'estudi previ a la implementació del sistema es van confondre conceptes i funcionalitats del sistema Standby lògic de l'extensa documentació d'Oracle [6]. Les raons per les quals aquest sistema no replica l'estructura Multi-master són els següents. Els canvis que es poden realitzar als servidors Standby no es propaguen cap als altres servidors, només s'apliquen a ells mateixos. Els servidors Standby lògics només reben dades d'un servidor principal. El servidor Standby no està al mateix nivell de jerarquia que el servidor primari. Tots aquests fets contradueixen les bases de l'arquitectura Multi-master. Es recorda que en aquesta arquitectura tots els servidors estan al mateix nivell, es poden modificar, i els canvis produïts a cadascun d'ells es propaguen a tots els altres servidors del sistema.

Després d'arribar a aquestes conclusions es va seguir estudiant les alternatives disponibles per implementar l'arquitectura Multi-master. Es va determinar que era possible aplicar aquesta estructura utilitzant altres tècniques que es descriuen al document "Arquitectura Multimaster" del dossier. Aquesta alternativa però, es troba obsoleta avui dia (Advanced Replication) i substituïda per l'opció GoldenGate mencionada a l'estat de l'art, que és l'opció que s'hauria d'utilitzar per aplicar un sistema Multi-master.

## 6 RESULTATS

En aquest apartat es troben descrits els resultats obtinguts de les proves realitzades sobre els sistemes de replicació Standby Físic i Standby Lògic. Les proves funcionals recullen resultats relacionats amb l'estat dels sistemes de replicació i l'estat dels servidors un cop implementats els sistemes. També s'expliquen els resultats derivats de les operacions de canvi de rol típiques per posar a prova el funcionament del sistema Standby físic i lògic implementat. L'últim apartat exposa els resultats de proves més tècniques de rendiment i configuració dels sistemes. La implementació de les totes aquestes proves es troba detallada als documents del dossier del projecte [13] [14].

### 6.1. RESULTATS PROVES FUNCIONALS

Les següents proves estan relacionades amb l'estat de la base de dades primària i Standby, l'estat dels serveis de replicació i recuperació dels servidors i el funcionament de la replicació als sistemes distribuïts.

#### Estat de les bases de dades

Una de les proves més importants radica en l'estat dels servidors posterior a la implementació del sistema Standby. El resultat esperat depèn del tipus de Standby (físic o lògic). Mitjançant una operació sobre els servidors primari i secundari s'obté la informació necessària per determinar si l'estat de les bases de dades és el correcte. L'operació consulta el nom de la base de dades, el nom únic, l'estat d'obertura i el seu rol.

Els resultats són els esperats al servidor primari (figura 9) i al servidor secundari (figura 10). El nom de les bases de dades és idèntic perquè són el mateix sistema, però el seu nom únic difereix al ser bases de dades diferents. El mode d'obertura és el que correspon a un servidor master, llegir i escriure. Al servidor esclau o Standby l'estat és només de lectura. També es pot detectar que el servei de replicació de dades hi està actiu amb el valor "with apply". El rol dels servidors és el correcte, sent el primer el servidor primari i el segon el servidor secundari o physical standby en

aquest cas. Amb el rol es pot verificar que existeix una relació de recuperació de dades Standby física entre els dos servidors.

```
SQL> select name, db_unique_name, open_mode, database_role from v$database;
NAME          DB_UNIQUE_NAME          OPEN_MODE          DATABASE_ROLE
-----
ORADB         ORADB                    READ WRITE        PRIMARY
```

Fig. 9: Estat base de dades primària al sistema Standby Físic

```
SQL> select name, db_unique_name, open_mode, database_role from v$database;
NAME          DB_UNIQUE_NAME          OPEN_MODE          DATABASE_ROLE
-----
ORADB         ORADB_S2                READ ONLY WITH APPLY PHYSICAL STANDBY
```

Fig.10: Estat base de dades alternativa al sistema Standby Físic

En el cas del Standby lògic, els resultats també són els esperats al servidor master (figura 11) i al servidor esclau (figura 12).

```
SQL> select name, db_unique_name, open_mode, database_role from v$database;
NAME          DB_UNIQUE_NAME          OPEN_MODE          DATABASE_ROLE
-----
ORADB         ORADB                    READ WRITE        PRIMARY
```

Fig. 11: Estat base de dades primària del sistema Standby Lògic

```
SQL> select name, db_unique_name, open_mode, database_role from v$database;
NAME          DB_UNIQUE_NAME          OPEN_MODE          DATABASE_ROLE
-----
ORADB_S2     ORADB_S2                READ WRITE        LOGICAL STANDBY
```

Fig. 12: Estat base de dades alternativa al sistema Standby Lògic

El resultat és l'esperat. La primera comanda, realitzada a la base de dades primària, és igual a la base de dades del Standby físic. En canvi en el segon cas, el nom del sistema difereix del servidor primari i és igual al seu nom únic. En el Standby lògic això és correcte, ja que la base de dades secundària és molt més independent que en el Standby físic, on només replica les dades del màster. En aquest cas, com es pot observar la base de dades Standby està oberta per llegir i escriure, fet que la diferencia del Standby físic. En aquest cas la base de dades esclava còpia les dades del servidor primari, però també pot funcionar com una base de dades a part amb la seva pròpia configuració i dades, fet que la fa més flexible. L'últim paràmetre serveix per verificar que, efectivament el servidor esclau és un Standby lògic.

#### Estat Tecnologies i Serveis de Replicació

Una altra comprovació important que realitzar és l'estat de les tecnologies de replicació utilitzades pels dos entorns. Cada cop que s'inicien les bases de dades standby s'ha d'engegar manualment el servei que replicarà les dades del servidor primari. Per comprovar que el procés està en funcionament revisem la vista "v\$dataguard status", que permet veure l'estat de la tecnologia Data Guard. Els resultats són satisfactoris. Al resultat del Standby físic (figura 13) el servei de recuperació de dades (Log Apply) informa que el procés de recuperació en temps real està activat. Al resultat del Standby lògic (figura 14) el servei de control de rols avisa que la replicació Standby lògica s'ha aplicat.

```
FACILITY          SEVERITY          DEST_ID MESSAGE_NUM ERROR_CODE CAL
-----
TIMESTAMP
MESSAGE
-----
CON_ID
-----
0
Log Apply Services      Informational      0      32      0 NO
29-MAY-21
Managed Standby Recovery starting Real Time Apply
```

Fig. 13: Estat servei de replicació Standby Físic

```

FACILITY              SEVERITY      DEST_ID MESSAGE_NUM ERROR_CODE CAL
-----
TIMESTAMP
MESSAGE
-----
CON_ID
Role Management Services Control          0          17          0 YES
29-MAY-21
ALTER DATABASE START LOGICAL STANDBY APPLY
0

```

Fig. 14: Estat servei de replicació Standby Lògic

## Replicació de Dades

Aquesta prova es basa en observar si la replicació de dades entre servidors es du a terme correctament. Així doncs es van realitzar múltiples operacions d'escriptura a la base de dades primària dels sistemes de replicació.

Es van fer operacions per modificar la base de dades de diferents tipus (INSERT, CREATE, UPDATE, DELETE i ALTER). En tots els casos el servidor secundari replicava les dades, que s'hi imprimen amb precisió a la base de dades Standby. La velocitat de replicació és quasi immediata gràcies a utilitzar dues màquines virtuals al mateix dispositiu. Els resultats de la replicació són idèntics al Standby físic i lògic. Una observació interessant és que la replicació només es du a terme un cop feta la confirmació de les operacions (commit) a la base de dades primària.

Una prova addicional sobre la replicació de dades va consistir a apagar el procés de replicació de dades al servidor esclau i escriure dades a la base de dades master. Mentre el servei de replicació fos apagat no es replicava cap dada, com era esperat. En el moment de reiniciar el procés de recuperació es replicaven totes les dades que havia anat creant al servidor primari al secundari. En iniciar-se el sistema detecta les operacions pendents realitzades a la base de dades primària i les aplica correctament al Standby físic i lògic.

A l'exemple de procés de replicació Standby físic proporcionat utilitzant la taula "v\$dataguard\_status" (figura 15), els dos primers missatges són la creació del procés en segon pla del procés de recuperació (Log Apply) de la base de dades Standby. El següent missatge informa que el procés de recuperació s'està iniciant. A continuació el sistema informa que està aplicant la recuperació d'un registre de canvis (redo log). El missatge ens indica el destí de l'arxiu on es troben els canvis. Com es pot veure el fitxer ens indica el thread i la seqüència que està utilitzant. Per últim, l'últim missatge ens indica que el procés està esperant a la següent seqüència de dades, això significa que ha acabat de recuperar les dades de la seqüència del missatge anterior, i per tant les dades estan disponibles a la base de dades.

```

Log Apply Services      Control          0
Attempt to start background Managed Standby Recovery process

Log Apply Services      Control          0
MRP0: Background Managed Standby Recovery process started

FACILITY              SEVERITY      ERROR_CODE
-----
MESSAGE
-----
Log Apply Services      Informational    0
Managed Standby Recovery starting Real Time Apply

Log Apply Services      Informational    0
Media Recovery Log +FRA/ORADB_S2/ARCHIVELOG/2021_05_29/thread_1_seq_21.270.10738
48243

Log Apply Services      Warning          0
Media Recovery Waiting for thread 1 sequence 22 (in transit)

```

Fig. 15: Exemple procés de replicació al Standby Físic

## 6.2. RESULTATS OPERACIONS CANVI DE ROLS

Les següents operacions Failover, Switchover i Switchback es basen en modificar els rols dels servidors dels sistemes de replicació. El correcte funcionament d'aquestes operacions és important per determinar si els sistemes de replicació funcionen correctament i estan ben configurats.

### Failover

La primera operació realitzada és el Failover. Aquesta operació consisteix a fer caure el servidor primari de l'entorn Master-slave. A continuació, es configura el servidor esclau (standby) perquè agafi el rol de Master. D'aquesta manera, tot i haver una caiguda al servidor principal el servei de la base de dades segueix en funcionament gràcies a la base de dades de contingència. Un cop el servidor secundari ha adquirit el nou rol de master, es pot recuperar el servidor caigut i convertir-lo en un nou servidor standby (esclau) de la nova base de dades primària (figura 16). Aquest últim pas és addicional al procés Failover, però era interessant de provar.

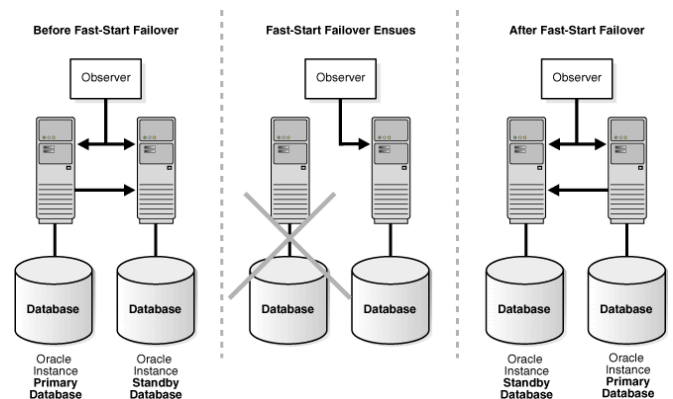


Fig. 16: Evolució dels rols dels servidors en un Failover

Els resultats de l'operació difereixen lleument entre el Standby físic i lògic. Ambdós casos el resultat és l'esperat, el servidor standby canvia de rol correctament i es converteix en el servidor primari. La diferència radica en convertir el servidor caigut en una nova base de dades secundària. En el Standby físic és força senzill canviar el rol del servidor caigut, però al Standby lògic s'ha de tornar a crear un standby lògic seguint els passos del document "Crear Standby Lògic Utilitzant Oracle Database 12c".

### Switchover

El Switchover és una operació semblant a l'anterior, però en aquest cas no és necessari provocar una caiguda del servidor primari. Simplement volem canviar els rols de les bases de dades (figura 17). Una de les diferències més importants radica en el fet que el canvi de rols del Failover només es du a terme una vegada i no hi ha marxa enrere, en canvi, en el Switchover el canvi de rols es pot produir les vegades que faci falta.



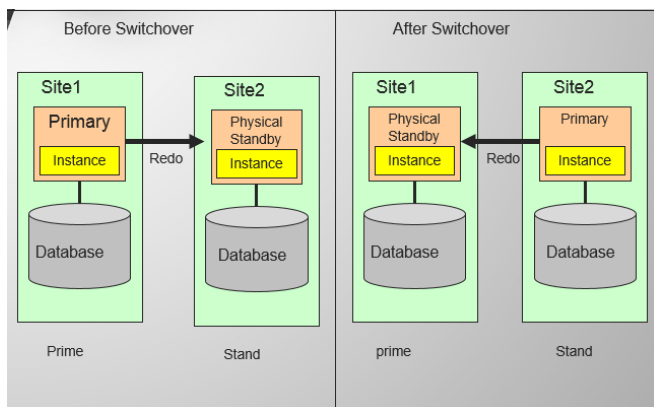


Fig. 16: Evolució dels rols dels servidors en un Switchover

En aquest cas els resultats són idèntics en els dos Standby físic i lògic. El canvi de rols es realitza correctament. El servidor primari es converteix a secundari (Standby), i el servidor esclau agafa el rol de Master. És una operació més senzilla, ja que no incorpora la caiguda ni recuperació de cap servidor.

### Switchback

El Switchback és una operació complementària al Switchover, es tracta de tornar a la situació inicial després d'haver realitzat un Switchover. Com diu el nom, és tornar enrere. Es basa en realitzar un Switchover a la inversa per tornar a la situació inicial. En ser bàsicament un Switchover el resultat és correcte igual que la prova anterior als dos entorns Standby (físic i lògic).

### 6.3. RESULTATS PROVES DE CONFIGURACIÓ I RENDIMENT

Aquestes proves tenen relació amb el rendiment del sistema de replicació Standby físic i lògic i el seu funcionament a un nivell més tècnic. Aquestes proves tenen a veure amb les possibles pèrdues de dades entre els servidors, el temps (delay) que tarden les dades en transportarse, en la búsqueda de canvis perduts (gaps) i en l'anàlisi dels processos relacionats amb la replicació dels sistemes Standby.

#### Pèrdua de Dades Entre Servidors

Per verificar el correcte funcionament del sistema Standby s'ha de verificar que no hi ha pèrdua de dades entre els servidors primari i secundari. En aquesta prova es mira la seqüència de logs dels dos servidors al seu historial de logs. Si coincideix és un bon senyal que no hi ha pèrdua de dades, ja que el servidor Standby és una còpia del servidor primari. Una altra manera d'assegurar-se és comprovant que l'última seqüència de logs generada al Standby és la mateixa que l'última seqüència detectada pel sistema, fet que significa que no hi ha hagut modificacions entre el procés de recuperació de logs a la base de dades.

El resultat del Standby físic (figura 18) ha sigut el mateix tant al servidor primari com al standby, per tant el resultat és satisfactori i es determina que no hi ha pèrdua de dades.

```
SQL> SELECT MAX(SEQUENCE#) FROM V$LOG_HISTORY;
```

```
MAX(SEQUENCE#)
-----
                22
```

```
SQL> SELECT THREAD# "Thread", SEQUENCE# "Last Sequence Generated" FROM V$ARCHIVE
D_LOG WHERE (THREAD#, FIRST_TIME) IN (SELECT THREAD#, MAX(FIRST_TIME) FROM V$AR
HIVED_LOG GROUP BY THREAD#) ORDER BY 1;
```

```
Thread Last Sequence Generated
-----
1 22
```

Fig. 18: Comprovació pèrdua de dades entre servidors al Standby Físic

El resultat del Standby lògic també és satisfactori (figura 19), els dos servidors tenen el mateix resultat, per tant no hi ha pèrdua de dades.

```
SQL> select max(sequence#) from v$log_history;
```

```
MAX(SEQUENCE#)
-----
                33
```

```
SQL> SELECT THREAD# "Thread", SEQUENCE# "Last Sequence Generated" FROM V$ARCHIVE
(SELECT THREAD#, MAX(FIRST_TIME) FROM V$ARCHIVED_LOG GROUP BY THREAD#) ORDER BY
```

```
Thread Last Sequence Generated
-----
1 33
```

Fig. 19: Comprovació pèrdua de dades entre servidors al Standby Lògic

#### Delay Entre Servidors

Un altre test important és el temps o "delay" que hi ha entre els servidors. Com que en aquest cas s'utilitzen màquines virtuals aquest temps hauria de ser pràcticament zero. Així doncs es realitza la prova als dos entorns. També és rellevant saber si un entorn és més ràpid que l'altre.

Al resultat del Standby físic (figura 20) es pot veure el temps de transport i aplicació dels logs és zero i el temps d'inicialització del procés és de 14 segons.

```
SQL> set lines 1000
```

```
select name,value from v$dataguard_stats;SQL>
```

NAME	VALUE
transport lag	
apply lag	+00 00:00:00
apply finish time	+00 00:00:00.000
estimated startup time	14

Fig. 20: Consulta sobre el delay del sistema Standby Físic

En el cas del resultat del Standby lògic (figura 21), els temps són també pràcticament zero, així que es determina que en màquines virtuals no hi ha pràcticament delay en el transport i aplicació de les dades. El temps d'inicialització és més elevat, 21 segons.

```
SQL> set lines 1000
```

```
select name,value from v$dataguard_stats;SQL>
```

NAME	VALUE
transport lag	+00 00:00:00
apply lag	+00 00:00:02
apply finish time	+00 00:00:00.387
estimated startup time	21

Fig. 21: Consulta sobre el delay del sistema Standby Lògic

## Búsqueda de Canvis Perduts

En aquesta prova es busca si hi ha canvis (redo logs) que s'hagin perdut en el transcurs del procés de transport i aplicació. Com que les altres proves relacionades amb el transport de dades han donat resultats positius, no s'hauria de trobar cap canvi perdut. Aquests arxius perduts s'anomenen gaps, i per trobar-los es mira la vista “v\$archive\_gap” que mostra tota la seva informació. Els resultats als dos entorns són positius, ja que en cap dels dos troba gaps (figura 22). En cas de trobar-ne s'hauria de seguir un procés per recuperar els arxius perduts [15].

```
SQL> select thread#, low_sequence#, high_sequence# from v$archive_gap;
no rows selected
```

Fig. 22: Consulta arxius gaps als sistemes de replicació Standby

## Processos del Servei Standby

L'última prova més tècnica consisteix en comprovar l'estat dels processos relacionats amb el control del standby. Mitjançant la vista “v\$managed\_standby” es pot veure si hi ha algun problema a algun procés en concret, i focalitzar l'error. En cas d'haver-hi algun error l'estat del procés seria “ERROR”, però també ens pot donar més informació interna del sistema. Per exemple, si es veu que les dades no es repliquen i hi ha un procés que està en estat “WAIT\_FOR\_LOG” voldrà dir que no li arriba el log per aplicar-lo al sistema. En aquest cas als dos entorns no hi ha hagut cap error ni activitat sospitosa dels processos.

Al resultat del Standby físic els processos es troben en un estat correcte (figura 23). Com a nota interessant es pot observar l'últim procés que està aplicant els canvis (redo logs) activament a la base de dades standby.

```
SQL> select process, client_process,thread#,sequence#,status from v$managed_standby;
```

PROCESS	CLIENT_P	THREAD#	SEQUENCE#	STATUS
ARCH	ARCH	1	22	CLOSING
ARCH	ARCH	1	24	CLOSING
ARCH	ARCH	0	0	CONNECTED
ARCH	ARCH	1	23	CLOSING
RFS	LGWR	1	25	IDLE
RFS	ARCH	0	0	IDLE
RFS	UNKNOWN	0	0	IDLE
MRPO	N/A	1	25	APPLYING_LOG

Fig. 23: Consulta processos Standby Físic

Als processos del Standby lògic tampoc hi ha cap activitat sospitosa, per tant els resultats han sigut els esperats (figura 24).

```
SQL> select process, client_process,thread#,sequence#,status from v$managed_standby;
```

PROCESS	CLIENT_P	THREAD#	SEQUENCE#	STATUS
RFS	ARCH	0	0	IDLE
ARCH	ARCH	1	58	CLOSING
ARCH	ARCH	0	0	CONNECTED
ARCH	ARCH	1	35	CLOSING
ARCH	ARCH	1	35	OPENING
RFS	LGWR	1	59	IDLE
RFS	UNKNOWN	0	0	IDLE

Fig. 24: Consulta processos Standby Lògic

## 7 CONCLUSIONS

Un cop avaluats els resultats obtinguts durant el desenvolupament del projecte podem determinar que s'han complert els objectius principals del projecte. Amb l'estudi del sistema Oracle i els seus mètodes de replicació de dades s'han localitzat les opcions disponibles per crear els sistemes distribuïts proposats i s'ha comprès la naturalesa del sistema Oracle Database (Objectiu 1).

Com ja s'ha explicat, l'opció de replicació d'Oracle més utilitzada avui dia pels usuaris és la tecnologia Oracle GoldenGate. Aquesta tecnologia és molt eficient, però suposa un gran cost monetari i és difícil d'administrar. Les opcions de replicació utilitzant la tecnologia DataGuard triades per desenvolupar el projecte no suposen un cost monetari i la seva implementació comporta una dificultat moderada, en comparació a les alternatives de replicació de què disposa el sistema Oracle. Utilitzar DataGuard però, va causar inconvenients relacionats amb la compatibilitat entre el sistema Oracle Database i els contenidors Docker. Per aquests inconvenients es van descartar els dos objectius secundaris del treball (Objectius 4 i 5), que requerien utilitzar la tecnologia de contenidors.

Utilitzant l'extensió Data Guard i el sistema de replicació Standby que proporciona es pot replicar l'arquitectura Master-Slave proposada. Aquest sistema disposa de dues variants, el Standby físic i el Standby lògic. El primer replica l'arquitectura Master-slave més estricta, mentre que el Standby Lògic permet la modificació de les bases de dades Standby, que es poden utilitzar per a altres objectius. Això el converteix en un sistema més flexible. S'han realitzat un conjunt de proves sobre els dos sistemes Standby que es troben descrites als resultats d'aquest article. Els resultats satisfactoris d'aquestes proves garanteixen la correcta instal·lació i funcionament dels sistemes de replicació (Objectiu 2).

Inicialment es volia utilitzar la tecnologia de DataGuard i el sistema Standby Lògic per intentar replicar l'arquitectura Multi-master. Durant el transcurs de la implementació d'aquest sistema es va concloure que l'arquitectura Multi-master no es pot aplicar amb aquesta tecnologia, a causa de les raons explicades al desenvolupament de l'arquitectura Multi-master.

Com a treball futur, s'implementaria l'arquitectura Multi-master. S'utilitzaria l'opció descrita al document “Arquitectura Multi-master” del dossier [15] (Objectiu 3). Aquesta opció però, és obsoleta avui dia i la substitueix i actualitza el software d'Oracle GoldenGate, al que no podem accedir pel seu alt cost d'adquisició.

## AGRAÏMENTS

Agrair a Oriol Ramos per l'ajuda i guia donada durant tot el seguiment del projecte, i tots els consells donats. Donar les gràcies a la meua família pel seu suport acompanyament durant tota l'etapa de desenvolupament del projecte.

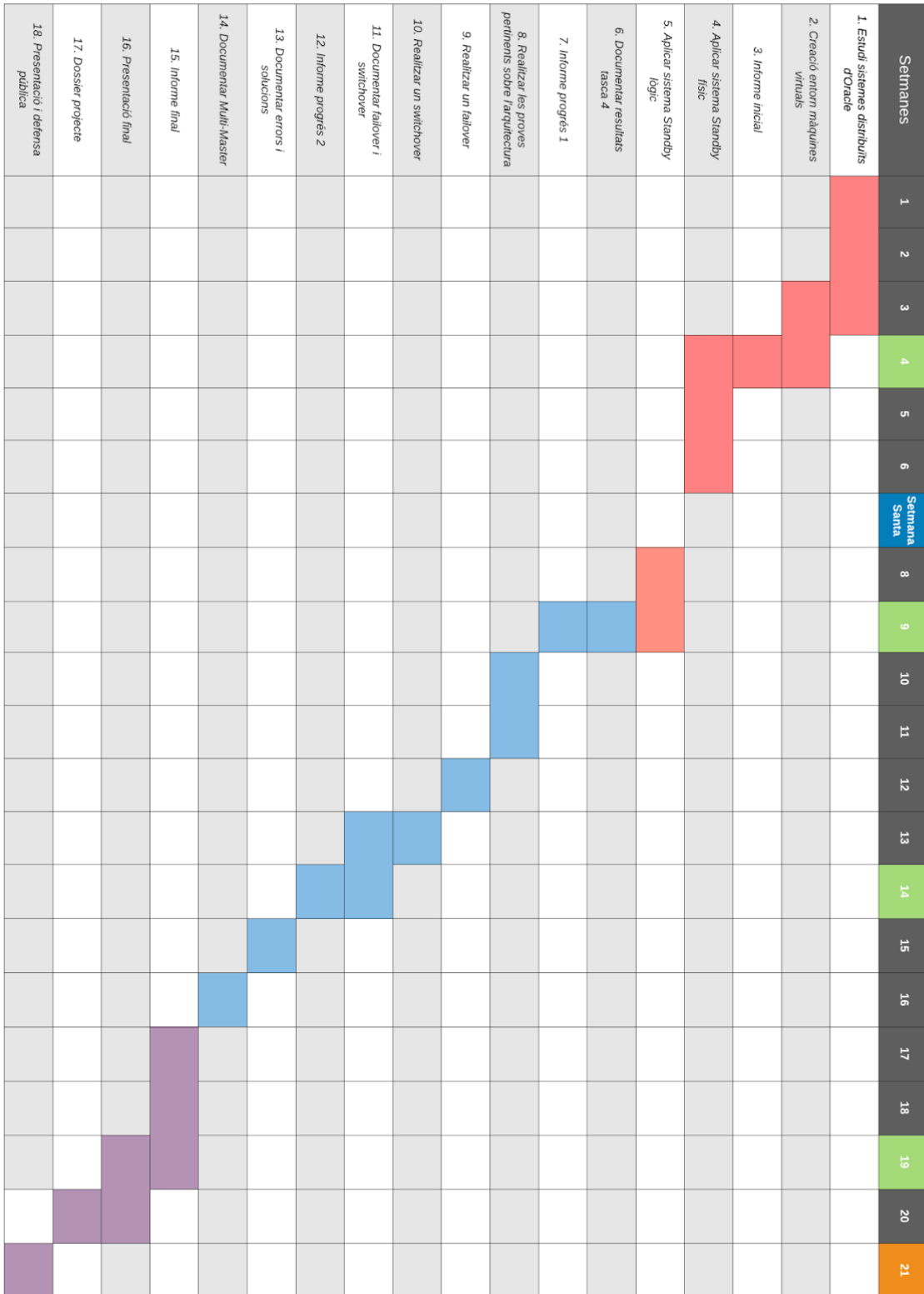
## BIBLIOGRAFIA

- [1] ORACLE FAQ. Top 12 New Features of Oracle 12c Summarized. URL: <https://www.oracle.com/node/2822> [Consulta: Març de 2021]
- [2] DAVE PARTRIDGE. Oracle 19c, Autonomous or not Autonomous? URL:

- <<https://www.mythics.com/about/blog/oracle-19c-autonomous-or-not-autonomous>> [Consulta: Març de 2021]
- [3] VIT SPINKA. Logical Replication in 12c: What are the Options Now? URL: <[https://www.doag.org/formes/pubfiles/8435938/2016-DB-Vit\\_Spinka-Logical\\_Replication\\_in\\_12cR2\\_-\\_What\\_are\\_the\\_Options\\_Now\\_-\\_Manuskript.pdf](https://www.doag.org/formes/pubfiles/8435938/2016-DB-Vit_Spinka-Logical_Replication_in_12cR2_-_What_are_the_Options_Now_-_Manuskript.pdf)> [Consulta: Març de 2021]
- [4] ORACLE HELP CENTER. Data Guard Concepts and Administration. 1 Introduction to Oracle Data Guard. URL: <<https://docs.oracle.com/database/121/SBYDB/concepts.htm#SBYDB00010>> [Consulta: Març i Abril de 2021]
- [5] ORACLE HELP CENTER. Data Guard Concepts and Administration. 3 Creating a Physical Standby Database. URL: <[https://docs.oracle.com/database/121/SBYDB/create\\_ps.htm#SBYDB00200](https://docs.oracle.com/database/121/SBYDB/create_ps.htm#SBYDB00200)> [Consulta: Abril i Maig de 2021]
- [6] ORACLE HELP CENTER. Data Guard Concepts and Administration. 4 Creating a Logical Standby Database. URL: <[https://docs.oracle.com/database/121/SBYDB/create\\_ls.htm#SBYDB00300](https://docs.oracle.com/database/121/SBYDB/create_ls.htm#SBYDB00300)> [Consulta: Abril i Maig de 2021]
- [7] KANBAN TOOL. Increase team performance with a visual project management tool. URL: <<https://kanbantool.com/>> [Consulta: Febrer de 2021]
- [8] DOCKER. Empowering App Development for Developers. URL: <<https://www.docker.com/>> [Consulta: Febrer i Març de 2021]
- [9] HASHICORP. Vagrant. Development Environments Made Easy. URL: <<https://www.vagrantup.com/>> [Consulta: Febrer i Març de 2021]
- [10] ORACLE. VirtualBox. URL: <<https://www.virtualbox.org/>> [Consulta: Març de 2021]
- [11] ORACLE HELP CENTER. Database Backup and Recovery User's Guide. 2 Getting Started with RMAN. <<https://docs.oracle.com/database/121/BRADV/rcmqick.htm#BRADV89346>> [Consulta: Març i Abril de 2021]
- [12] ALBERT FARRÉS. Arquitectura Master-Slave i MultiMaster en Oracle. Crear Standby Físic utilitzant Oracle Database 12c. [Consulta: Abril de 2021]
- [13] ALBERT FARRÉS. Arquitectura Master-Slave i MultiMaster en Oracle. Crear Standby Lògic utilitzant Oracle Database 12c. [Consulta: Abril de 2021]
- [14] ALBERT FARRÉS. Arquitectura Master-Slave i MultiMaster en Oracle. Failover, Switchover i Switch-back. [Consulta: Abril i Maig de 2021]
- [15] ALBERT FARRÉS. Arquitectura Master-Slave i MultiMaster en Oracle. Test i Rendiment del Sistema Standby. [Consulta: Maig de 2021]
- [16] ALBERT FARRÉS. Arquitectura Master-Slave i MultiMaster en Oracle. Arquitectura Multi-Master a Oracle Database 12c. [Consulta: Maig i Juny de 2021]

APÈNDIX

A1. DIAGRAMA DE GANTT DEL PROJECTE



Albert Farrés | June 17, 2021

Planificació TFG

Fig. A1: Diagrama de Gantt del Projecte