

# Terraform IaC GUI Development Tool

Oscar Martin Salas

**Resum**— La gestió dels recursos de les infraestructures de xarxa físiques convencionals, així com la seva configuració, avui dia està molt controlat i són, en la majoria dels casos, fàcilment tractables. El problema ha arribat amb l'aparició de les infraestructures Cloud degut a la gran elasticitat, escalabilitat i versatilitat que aquestes demanden. Per atacar aquest problema varen sorgir les eines Infrastructure as Code [1], comunament abreujat com a IaC (e.g., Terraform), la finalitat de les quals és, per mitjà d'*scripts*, automatitzar la gestió de sistemes distribuïts i arquitectures orientades a serveis al Cloud. A Oracle, l'equip de treball del que formo part, es disposa d'eines per generar codi IaC enfocades a tractar infraestructures a gran escala, per posteriorment fer-les córrer a l'Oracle Cloud. No obstant això, no disposen d'eines per a la gestió d'arquitectures de petita i mitjana escala. Aquesta problemàtica va donar origen a aquest projecte, el qual tracta del desenvolupament d'una aplicació web on l'usuari disposi d'una interfície on pugui dissenyar i configurar una infraestructura de xarxa de petita o mitjana escala, la qual generi codi IaC (Terraform *script*), per a posteriorment aplicar-lo a l'Oracle Cloud.

**Paraules clau**— Aplicació Web, Cloud, Helidon, Infraestructura de xarxa, Infrastructure as Code (IaC), Interfície, Java, Javascript, Oracle Jet, Script, Terraform, Web Service

**Abstract**— Nowadays, the management of physical network infrastructures and its configuration as well, are in most cases very well monitored and approachable. With Cloud infrastructures' appearance a new problem has emerged due to the elasticity, scalability and versatility they demand. To face this issue Infrastructure as Code [1], commonly abbreviated as IaC (e.g., Terraform) have been developed in order to, through scripts, automate the management of distributed systems and Cloud service oriented architectures. The Oracle's work team where I belong to, has IaC tools focused on treating large scale infrastructures, to later execute them in the Oracle Cloud. However, they don't have medium or small scale architectures' management tools. That's the issue that originated this project, the development of a web application where users through an interface can design and configure a small/medium scale network infrastructure, which will then export IaC code in Terraform format in order to later run it in the Oracle Cloud.

**Index Terms**— Web Application, Cloud, Helidon, Infrastructure as Code (IaC), Network Infrastructure Interface, Java, Javascript,, Oracle Jet, Script, Terraform, Web Service

---

◆

## 1 INTRODUCCIÓ

LES infraestructures de xarxa físiques convencionals han estat des dels seus inicis configurades i gestionades manualment de forma individualitzada en cadascun dels seus components. Al anys 90, la revolució tecnològica va originar els primers serveis basats en xarxes virtuals privades (VPN), els quals eren els primers brots del que als 2000 esdevindria el Cloud Computing [2]. Empreses com Amazon, Google, la NASA i posteriorment Microsoft van iniciar un procés de recerca en aquest camp i varen inaugurar els seus propis Data Centers [3], convertint-se així en les potències de serveis de còmput al núvol. No obstant, l'administració i configuració de cadascun dels components virtuals que conformen aquestes xarxes virtuals havien de ser tractades de forma individual d'igual forma que venia fent-se amb les físiques

convencionals. Als inicis d'aquests nous sistemes de còmput al núvol l'abast d'aquestes infraestructures no dificultava gaire la seva gestió, però els últims 10 anys el Cloud Computing s'ha consolidat en gairebé totes les grans companyies del sector amb expectatives d'augmentar exponencialment d'ara en endavant. El creixement d'aquestes arquitectures no només s'ha produït pel que fa al nombre d'empreses que disposen de Data Centers d'aquestes, sinó que les exigències en quant a versatilitat, elasticitat i sobretot escalabilitat, estan dificultant considerablement la gestió i configuració de tots els components que les conformen. És per això que va sorgir el concepte de *Infrastructure as Code*, del anglès Infraestructura com a Codi (IaC), el que correspon al procés d'automatitzar les tasques de gestió, configuració i establir dependències d'una infraestructura per mitjà d'arxius de configuració (Fig. 1) dissenyats per ser interpretats per cadascun dels seus components, evitant així la difícil tasca de tractar-los de forma individual.

- E-mail de contacte: oscar.martins@e-campus.uab.cat
- Menció realitzada: *Tecnologies de la Informació*.
- Treball tutoritzat per: Jordi Pons Aróztégui (DEIC)
- Curs 2018/19

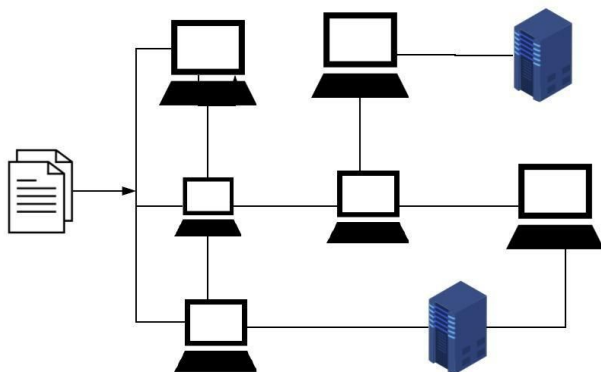


Fig. 1. Per mitjà d'arxius de configuració (*scripts*) les IaC ens permeten la gestió, configuració i administració dels seus components.

Tot i que els guanys que s'obtenen d'aplicar IaC al Cloud són molt notables no treu que s'empri a les xarxes i Data Centers físics. Com era d'esperar, l'aparició de la Infraestructura com a Codi va donar pas al sorgiment de diverses eines que donaven suport com ara Puppet (Puppet) [4], Terraform (HashiCorp) [5] i DSC (Microsoft) [6] d'entre d'altres.

Cal esmentar en aquest punt que el desenvolupament d'aquest projecte s'ha dut a terme sota el recolzament i supervisió d'ORACLE CORPORATION [7], empresa amb la qual s'està realitzant un conveni de col·laboració amb la universitat en forma de pràctiques. El sub-departament on s'ha portat a terme és el de Core dintre del departament de Consulting. En aquest departament, d'entre d'altres moltes tasques, es gestionen, configuren i administren infraestructures de xarxa per a altres empreses, tot aplicant-les a Oracle Cloud [8]. Per la realització d'aquestes comeses es disposa d'eines IaC en el cas d'infraestructures de gran escala. La motivació d'aquest projecte sorgir de la necessitat de disposar d'una eina que, per mitjà d'una interfície gràfica senzilla, es pogués dissenyar una petita o mitjana infraestructura de xarxa, que permetés una disposició espacial dels seus components i que proveís d'elements de configuració d'aquests amb la finalitat d'exportar el model a un *script* apte per Terraform, per a posteriorment fer-lo córrer a l'Oracle Cloud. Amb aquesta eina s'estalviaria una quantitat molt considerable de temps i es facilitaria el seu disseny i la capacitat de fer proves.

L'aplicació constarà de dos mòduls principals, el *Back end* al servidor i del *Front end* a la part del client. El *Front end* tindrà com a objectiu principal, tal i com s'ha esmentat prèviament, proporcionar a l'usuari d'una interfície gràfica on pugui dissenyar la infraestructura de xarxa que desitgi, per a posteriorment, quan se li requereixi, enviar el model de dades corresponent al servidor. El *Back end* s'encarregarà doncs de procesar la informació rebuda del client tot transformant-la a llenguatge *script* de Terraform. La comunicació entre ambdues parts es farà per mitjà de RESTful Web Services [9] ja que ens proveirà

d'una gran agilitat a l'hora de tractar les peticions client/servidor.

Al llarg d'aquest document es detallarà el procés de desenvolupament que s'ha portat a terme per al projecte, així com les tecnologies aplicades, detallades anteriorment.

L'organització de les seccions constarà dels següents punts:

- Introducció a les IaC i presentació del projecte.
- Objectius de la implementació de l'aplicació i funcionalitats a abordar.
- *Front End*.
- *Back End*.
- *Base de Dades*.
- Resultats i dificultats del projecte.
- Conclusió.

## 2 OBJECTIUS

Tal i com hem pogut anar veient a la secció introductoria d'aquest document l'objectiu principal d'aquest projecte consisteix en el desenvolupament d'una aplicació web on l'usuari pugui dissenyar i configurar una infraestructura de xarxa de petita o mitjana escala, la qual generi codi IaC (Terraform *script*), per a posteriorment aplicar-lo a l'Oracle Cloud.

Els 6 elements de xarxa que des d'Oracle interessen amb més rellevància són els següents:

1. VCN's
2. Subnets
3. Security lists
4. Internet gateways
5. Route tables
6. Dhpc options

Per la correcta execució del projecte, s'ha desglossat la seva implementació en els següents objectius:

- Interfície gràfica que permeti a l'usuari una disposició espacial dels diferents elements que formin la infraestructura de forma personalitzada i a mida.
- Proveir al dissenyador d'elements de configuració (formularis) per a la construcció d'un codi en el qual la funcionalitat post-deployment és el principal objectiu.
- L'aplicació ha de permetre un disseny de l'arquitectura de la xarxa de forma senzilla i intuïtiva.
- Dotar el servei de l'opció de desar el disseny per a posteriorment seguir tractant-lo, així com la seva configuració.
- Autenticació al Oracle Access Manager [10] per mitjà del protocol LDAP [11] per tal de permetre l'ús de l'eina exclusivament a usuaris d'Oracle amb permís.

Per assolir aquests objectius, s'ha proposat el següent disseny de solució:

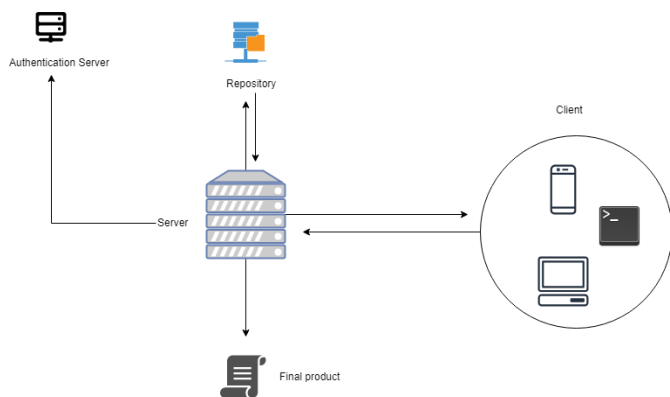


Figura 2. Disseny de solució

Pel que fa al mòdul del client es desenvoluparà amb Oracle JET [12], una tecnologia orientada a proporcionar elements de disseny gràfic web creada per Oracle, la qual es veurà més en detall més endavant.

La tecnologia que s'aplicarà per implementar els *RESTful web services* al mòdul del servidor serà Helidon [13], un software orientat a *micro web services* [14].

### 3 ESTAT DE L'ART

Els *Web Services* és una tecnologia de software dissenyada per donar suport a la interoperabilitat màquina-màquina per l'intercanvi de dades. Aquests poden ser de dos tipus principalment, SOAP [15] o REST [16], els quals van sorgir més tard, al 2004. Tot i que ambdós operen sobre HTTP els primers basen la seva comunicació en protocols de més baix nivell, treballant únicament amb missatges XML. És per això que posteriorment es van originar els *RESTful Web Services*, els quals permeten pràcticament qualsevol tipus de format en els seus missatges com HTTP, XML, JSON [17] o text pla. En aquest projecte s'ha escollit el segon tipus degut a que la informació que viatjarà entre client i servidor serà en format JSON, degut a que aquest ens ofereix una gran capacitat de modelatge de les dades.

Avui en dia en la majoria d'arquitectes de software, especialment d'aplicacions web que contemplen *Front end* i *Back end*, fan ús de *Web Services* ja que aquests operen independentment de la plataforma que es faci servir en cadascun dels mòduls de desenvolupament i proporciona una interfície de comunicació única per a tota l'aplicació. A més la seva facilitat d'ús, la qual es basa en l'accés de cada servei per mitjà d'una URI establerta esdevé una qualitat de considerable importància a l'hora d'escollir el protocol de comunicació d'una aplicació. apliquen els de tipus REST, principalment per la versatilitat que aquests ofereixen.

Pel que fa a la tecnologia emprada al client, Oracle Jet és un framework *Open Source* [18] poc globalitzat però que

es troba en auge, degut al gran ventall d'opcions i elements d'interfície d'usuari que ofereix.

D'altra banda, l'Helidon (*Open Source* també) és una eina molt utilitzada per desenvolupadors avui en dia per treballar amb *Web Services*. Un dels principals motius del seu èxit es deu a que es tracta d'un framework que treballa sobre Java, llenguatge de programació dominant en el mercat pel que fa la implementació de servidors web i d'altres aplicacions.

Aquestes tecnologies han sigut proposades per l'equip d'Oracle que ha estat supervisant aquest projecte de cara a fer ús de software de la propia empresa.

## 4 DESENVOLUPAMENT

### 4.1 Arquitectura

Tal i com s'ha especificat amb anterioritat en el disseny de la solució d'aquest document, l'aplicació constarà de dos mòduls: la part del client i la part del servidor, la comunicació entre els quals es realitzarà per mitjà de *Web Services* allotjats al servidor.

Les tecnologies que s'utilitzaran per implementar cadascun d'aquests mòduls es troben esmentades a la Figura 3, la utilitat de les quals es veu descrita a continuació.

La part del client s'encarregarà de tot el que és el disseny de la interfície i d'enviar les dades que el servidor recollirà per processar-les i realitzar les operacions que detallarem més endavant. La tecnologia que utilitzarem per aquest mòdul serà una aplicació web Javascript/HTML5, per mitjà del framework Oracle Jet. El pas de dades entre servidor i client es farà en format JSON, i faran la comunicació a través de *RESTful micro web services*. Per fer ús de l'eina els usuaris hauran d'estar registrats a un servidor d'autenticació LDAP, en concret OAM (Oracle Access Manager) per tal d'autenticar-se i accedir al servei.

El mòdul del servidor tindrà les funcions de validar les dades que l'usuari introdueix, respecte les configuracions que realitzi, de forma asíncrona, per assegurar la funcionalitat global de l'aplicació. A més, el servidor serà l'encarregat de tractar les dades que li envii el client i transformar la informació al *script* definitiu desitjat. La tecnologia utilitzada per a implementar la part del servidor serà Helidon, un conjunt de llibreries Java, desenvolupades per Oracle, que s'executen en un nucli web ràpid. En aquest projecte aquest framework ens proveirà de *micro web services*. Aquests últims han sigut escollits a la implementació ja que ens subministren un codi molt mantenible i testeuable i un desplegament del servei modular i molt independent, d'entre d'altres característiques. Pel que fa a la gestió de les dades, l'usuari podrà desar la configuració del disseny a una base de dades Oracle MySQL [19], per mitjà de la mateixa

codificació dels elements, de forma que aquest disseny sigui portable i alhora l'usuari sigui capaç de reprendre el disseny posteriorment sense que aquest hagi de ser arxivat en format *script*.

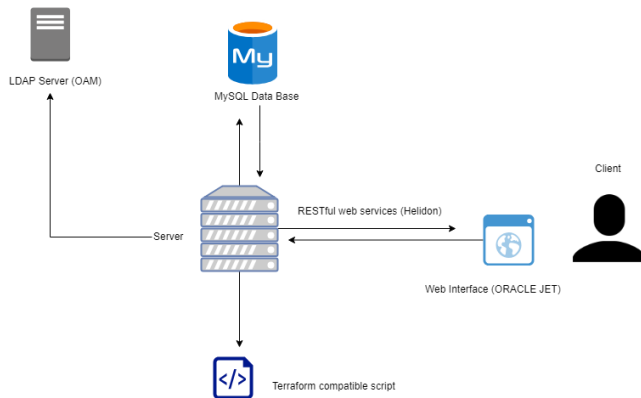


Figura 3. Arquitectura del projecte, juntament amb els tipus de components i les tecnologies emprades en cadascun d'ells

## 4.2 Metodologia

Pel seguiment del projecte i la correcta execució del mateix, s'ha fet ús d'una metodologia Agile (Scrum) [20]. Aquesta decisió va ser presa ja que hi ha certs punts d'inflexió en el projecte, els quals marcaven el ritme d'execució del mateix així com replantejar-se o requerir modificacions en quant a la tecnologia a utilitzar, l'eina o inclús els requisits. Per la implantació d'aquesta metodologia s'ha utilitzat l'eina Taiga [21], una aplicació web que proporciona suport per a la gestió de tasques i seguiment de projectes, tot seguint la metodologia Scrum.

L'aplicació d'aquesta metodologia s'ha dut a terme orientat als principals objectius definits a l'inici d'aquest document, reflexats al *product backlog* (conjunt de requisits globals del projecte), format per les següents tasques, ordenades de menor a major prioritat en funció de la relació de dependència del projecte i per tant en ordre d'execució:

### Front end

- Mostra dels components a la interfície gràfica
- *Drag and drop*
- Lincatge dels nodes (visual)
- Lògica corresponent a les relacions entre els diferents components
- Proveir formularis de configuració i atribució de dades als elements de la xarxa
- Recepció i descàrrega del *script*

### Back end

- Implementació d'un *web service*
- Recepció de dades del client
- Processament del model de dades rebut a format *script* de Terraform

- Retorn del contingut del *script*
- Autenticació LDAP
- Desament de models de disseny a la base de dades
- Obtenció de models de la base de dades

### Altres

- Disseny de la base de dades
- Tests/proves

Seguint el model de treball d'Scrum i després d'haver definit els requisits s'han anat realitzant sprints (cicles) cada dues setmanes per a disposar d'un seguiment continu del progrés del projecte.

Degut a la gran dificultat de quadrar dates i gran ocupació dels demés membres, s'han anat realitzant, amb menys freqüència, reunions (*sprint review*) amb el tutor i la supervisora del projecte, s'han dut a terme les tasques d'avaluació i seguiment dels avenços del darrer període, així com la definició de les noves comeses (*sprint backlog*) assignades per al següent. La realització d'aquestes eren de gran utilitat per tal d'avaluar el progrés del projecte, analitzar el codi de l'aplicació i proposar solucions i millores davant els punts d'inflexió i/o dificultats. El seguiment més exhaust del projecte ha estat portat a terme per la meua persona.

## 4.3 Front end

En aquesta secció s'exposaran detalladament les diverses implementacions que s'han produït al mòdul client, així com la seva estructura, l'organització de les dades i l'eina que s'ha fet servir pel seu desenvolupament.

L'objectiu final del *front end*, recordem, consisteix en dotar l'aplicació d'una interfície gràfica que faciliti a l'usuari dissenyar una infraestructura de xarxa on, de forma intuïtiva, pugui disposar espacialment els diferents components que la composin. A més, l'usuari ha de poder configurar i gestionar els atributs dels elements de xarxa afegits a la infraestructura per tal que aquestes dades es vegin reflectides en l'*script* final.

Pel que respecte al mètode de disseny de la interfície es farà per mitjà de *drag and drop* ("arrossega i deixa anar") tot permetent lincar els diferents elements de la xarxa entre sí amb opcions d'eliminació de nodes i enllaços, distribució espacial i edició dels atributs dels components que la formen.

Tot i no tractar-se d'una funcionalitat visual, el *front end* ha de mantenir un seguit d'estructures de dades que permetin establir una lògica pel que fa a les relacions entre els diferents elements que conformen la infraestructura, el qual es veurà amb més detall al final d'aquesta secció. Aquesta lògica dotarà a l'aplicació una primera organització de les dades, essencial per obtenir un arxiu de configuració resultant funcionalment correcte un cop executat a l'Oracle Cloud.

Per satisfer aquestes necessitats, s'ha fet ús d'Oracle Jet, que, tal i com s'ha esmentat prèviament, consisteix en un software lliure d'Oracle, l'objectiu principal del qual recau en oferir un gran catàleg de possibilitats gràfiques, ja implementades, per ser instaurades en la interfície d'usuari d'una aplicació web. Per la implementació tant de tots els elements visuals com dels lògics, s'ha consultat contínuament el *Cookbook* [22] ("llibre de receptes") que l'eina ens ofereix. Aquest, ha sigut utilitzat i pres com a exemple tant en el procés de formació, com al llarg del transcurs del projecte a mesura que s'anaven afegint elements a la interfície i/o funcionalitats.

L'Oracle Jet té una particularitat en quant al seu funcionament respecte a les eines tradicionals de desenvolupament d'interfície d'usuari i és que es basa en una arquitectura modular (Figura 4) [23]. Aquests mòduls estan dissenyats per a correspondre a cadascuna de les possibles vistes/finestres que es desitja disposar i basen el seu funcionament en una arquitectura composta per un *back end*, implementada en JavaScript i d'un *front end*, codificat en HTML. La relació entre ambdós arxius es veu reflectida en un document que exerceix de controlador de la interfície d'usuari anomenat *appControler.js*.

L'organització modular està suportada per RequireJS [24], un fitxer JavaScript que s'encarrega de la càrrega de mòduls, fet que simplifica la gestió de les referències de les llibreries i que està dissenyat per millorar la velocitat i la qualitat del codi de l'aplicació. RequireJS implementa l'API de definició de mòduls asíncrons (AMD) que proporciona un mecanisme per carregar de forma asíncrona un mòdul i les seves dependències.

El funcionament de la modulació consisteix en vincular els elements visuals utilitzats a l'arxiu HTML amb una variable "*observable*" ubicada al document JavaScript. El vincle d'aquestes dades es realitza per mitjà de KnockoutJS [25].

KnockoutJS o sovintament abreujat KO, consisteix en un framework de codi obert que dona suport a l'Oracle Jet, dissenyat per simplificar la connexió de dades JavaScript amb els components de la interfície d'usuari, permetent als desenvolupadors evitar les trucades de jQuery DOM de baix nivell per actualitzar els elements de la interfície d'usuari quan les dades canvien o, en altres paraules, actualitzar el DOM, si les dades d'un dels components de la interfície es veuen modificades, en nom del desenvolupador. Això proveeix a l'aplicació d'un gran ventall de possibilitats i facilitats a l'hora de manipular el contingut del *front end* a través del seu component vinculat de l'arxiu *JavaScript* associat, tal i com es mostra a la Figura 5.

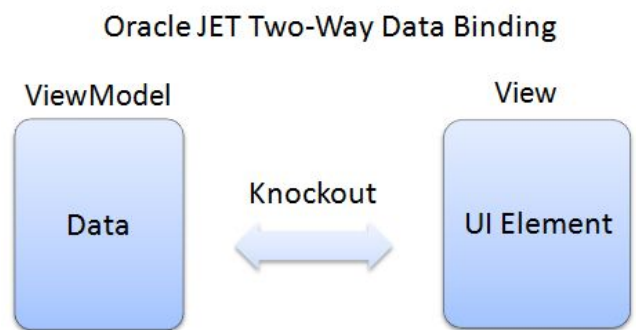


Figura 5. Oracle JET Data Binding. KnockoutJS exerceix la funció de vinculació entre els elements visuals de la vista amb la seva estructura de dades corresponent al model.

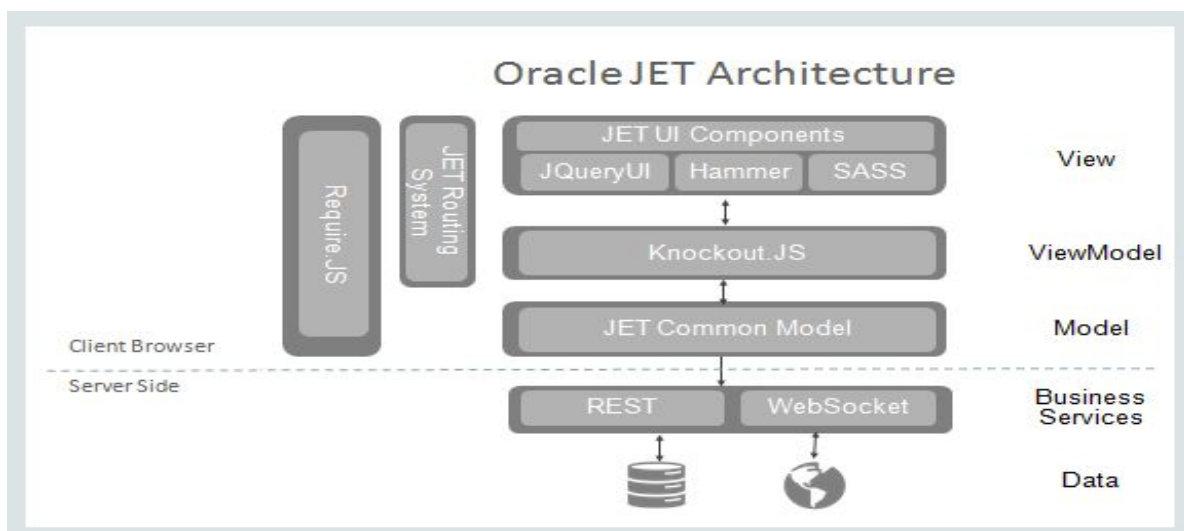


Figura 4. The Oracle JET Architecture

### 4.3.1 Autenticació Oracle Access Manager

L'Oracle Access Manager (OAM) és una aplicació implementada en un servidor d'Oracle, la qual proveeix un punt d'entrada comú per realitzar la validació i autenticació de totes les aplicacions de la companyia, així com d'usuaris, processos i d'altres entitats que ho requereixin tot implementant un servidor de directoris.

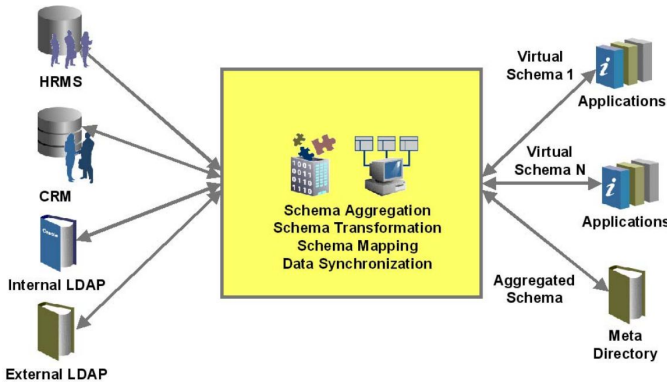


Figura 6. Arquitectura funcionament OAM [26]

L'OAM conté un mòdul OIM (Oracle Identity Manager) el qual dona suport a l'autenticació d'usuaris del marc d'empleats, no només a nivell nacional, sinó internacional amb un gran nombre de possibilitats pel que fa a la creació de grups de treball i administració dels treballadors d'Oracle, el qual oferirà a l'administrador de la Terraform IaC GUI Development Tool d'un alt nivell de decisió pel que respecta a qui es vol que pugui fer ús de l'eina.

Totes les dades del servidor OAM, incloses les d'autenticació d'usuari es troben a una base de dades inclosa en l'arquitectura, a la qual s'accedeix a través del protocol LDAP (*Lightweight Directory Access Protocol*), un protocol software que permet localitzar organitzacions, individus, arxius i/o dispositius ja sigui a Internet o a una xarxa privada com és el cas de la implementació emprada per aquest projecte. Aquest protocol servirà el mòdul d'autenticació d'autenticació, com a element de seguretat i gestió de permisos.

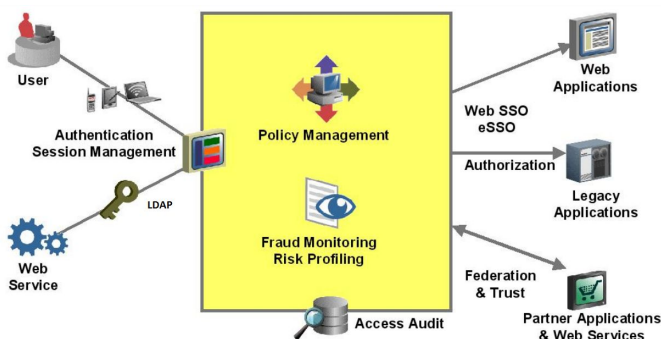


Figura 7. Gestió d'identitat de l'OIM [26]

### 4.3.2 Seguretat *Front End*

L'autenticació a l'Oracle Access Manager s'ha dut a terme al servidor de la nostra aplicació per termes d'usabilitat els quals es troben exposats a la secció 4.4 (Base de dades), per tant, s'ha implementat en Java. No obstant això les dades es veuen recollides de l'usuari per l'Oracle Jet, qui posteriorment les envia al *Back end*, qui les rep per realitzar l'autenticació.

La comunicació client - servidor d'aquesta funcionalitat està establerta, a l'igual que la resta de comunicacions, per un *web service* dedicat exclusivament a proveir a l'usuari de permisos per fer ús de la interfície.

En aquests missatges hi ha contingut confidencial, en particular, les credencials privades de cada usuari i per tant, de l'empresa. És per això que s'ha aplicat una capa de seguretat per establir una comunicació i autenticació segura, tot i que per accedir a l'OAM es precisa trobar-se a la xarxa interna d'Oracle.

Per tal d'evitar atacs del tipus *man in the middle* s'ha establert un canal TLS 1.2. [27], un descendent del protocol SSL que utilitza certificats x.509 [28] per establir un sistema criptogràfic asimètric basat en xifrar les dades de punt a punt. Això proveirà l'eina de confidencialitat i integritat en els missatges enviats entre client i servidor, ja siguin relacionats amb el *login* o no.

A més, al servidor de la base de dades s'aplicarà encriptació *MySQL Data at Rest Encryption* [29], el que farà que les dades emmagatzemades estiguin encriptades i només siguin accessibles quan un usuari s'ha autenticat i realitza consultes contra la base de dades.

De la mateixa manera que la base de dades, la màquina que allotjarà l'aplicació tindrà el disc dur encriptat pel que el codi serà inaccessible sense les credencials que Oracle li assigni al dispositiu.

### 4.3 Back end

Al llarg d'aquesta secció s'exposaran les característiques, consideracions del mòdul *Back end* de l'aplicació Terraform IaC GUI Development Tool, equivalent al servidor de la mateixa, així com una introducció a la tecnologia que s'ha aplicat pel seu desenvolupament, com s'ha utilitzat i quina aplicació se li ha donat per aquest projecte.

Tal i com s'ha argumentat a la secció 4.1 d'aquest document, per la implementació del servidor del sistema s'ha fet ús del framework Helidon. Aquesta tecnologia d'Oracle basada en la creació de *web services*, ofereix dues versions funcionalment equivalents, però amb mètodes i sintaxis diferents. La primera, Helidon MP, la qual conté APIs molt similars a les de *Java Enterprise* [30] basada en anotacions i la segona, Helidon SE, fonamentada en la transparència de tot el que es realitza utilitzant únicament llibreries d'Helidon. Aquesta característica ha sigut

determinant a l'hora d'escollir aquesta opció per aquest projecte, degut a l'enteniment del codi i assimilació per al procés de formació.

La implementació dels *web services* funciona mitjançant directoris on, a partir del directori arrel, es pot gestionar l'organització en funció del que es desitgi. En aquest projecte s'ha definit un subdirector, és a dir, un *web service*, per a cada funcionalitat que requeria d'interacció client - servidor. Aquestes es veuen definides a la Figura 8.

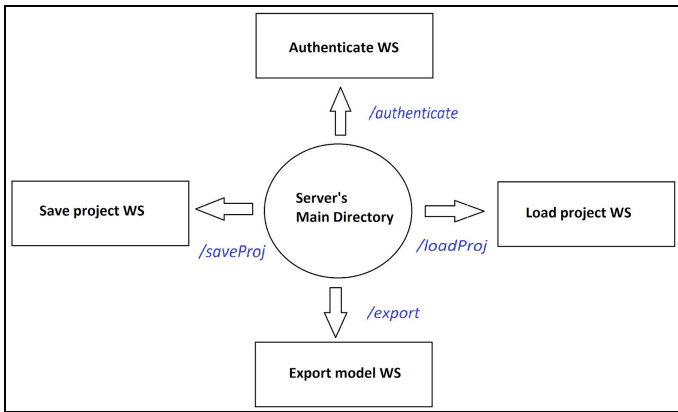


Figura 8. Organització dels directoris (*web services*) del *Back end*.

*Web Services Terraform IaC GUI Development Tool:*

- **Authenticate:** Es crida a l'iniciar l'aplicació per tal de validar que l'usuari que està fent ús de l'eina és empleat d'Oracle i a més té permisos per utilitzar-la. El seu funcionament es veu reflectit a la Figura 9.

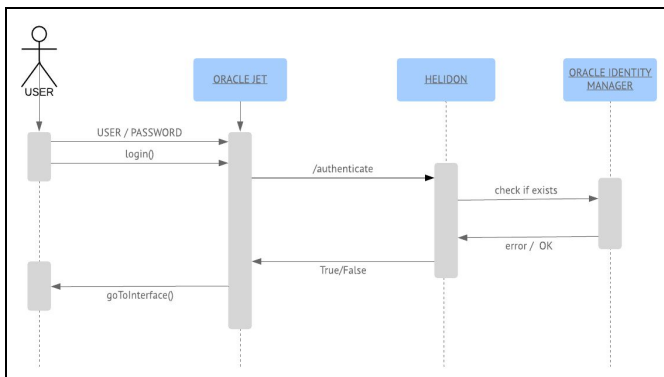


Figura 9. Diagrama de seqüència de l'autenticació a l'Oracle Identity Manager.

- **Export:** Aquesta funció esdevé l'objectiu principal de l'aplicació. Un cop dissenyat el model d'interfície de xarxa destijat amb les dades corresponents, l'usuari, per mitjà d'un botó "Export" ubicat a la interfície, cridarà al *web service* per rebre les dades de l'*script* per posteriorment crear un arxiu Terraform amb aquest contingut.

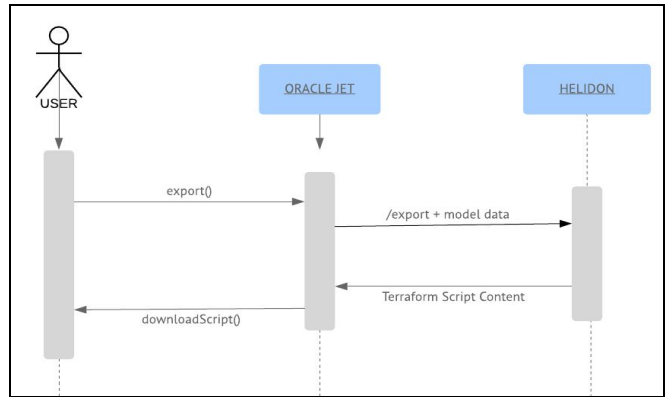


Figura 10. Diagrama de seqüència de la funcionalitat d'exportar el model a un *script* compatible amb Terraform.

- **Save project:** Si l'usuari desitja desar el disseny del model d'infraestructura per posteriorment reprendre'l, se li dona l'opció de desar-lo a la base de dades detallada a la secció 4.4. La seqüència d'accions d'aquesta funcionalitat es veu contemplada a la Figura 11.

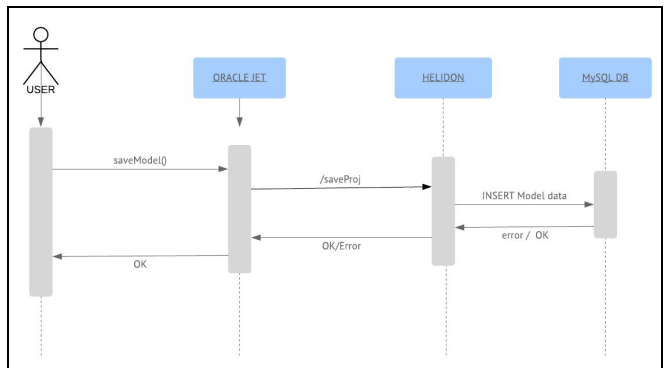


Figura 11. Diagrama de seqüència del desament del model d'infraestructura a la base de dades.

- **Load project:** De la mateixa forma que la interfície permet l'opció de desar un model d'infraestructura, l'aplicació pot carregar un model ja existent que pertanyi a l'usuari que està fent ús de l'eina, tal i com es mostra a la Figura 12.

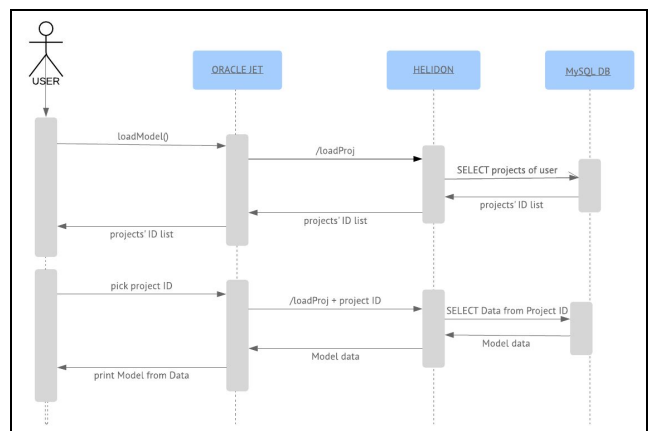


Figura 12. Diagrama de seqüència de la càrrega del model d'infraestructura de xarxa tot recuperant les dades de la BD MySQL.

#### 4.4 BASE DE DADES

A l'hora de plantejar l'arquitectura per aquest projecte es va decidir, tal i com s'ha anat argumentant al llarg del document, proporcionar un servidor de base de dades de tipus MySQL per oferir a l'usuari l'opció de desar el model d'infraestructura de xarxa en curs, per posteriorment reprendre'l i prosseguir el seu disseny sense haver de fer-se de nou.

Per la realització del model de base de dades que dona suport a l'aplicació es va tenir molt en compte el model de dades que es tenia present en el servidor, el qual es va dissenyar per donar suport a la funció d'exportar, juntament amb els requisits i consideracions dels supervisors del projecte a Oracle.

El model de base de dades dissenyat, contempla el nom d'usuari de qui està fent ús de l'eina en l'instant del desament, el qual es desa quan fa *Log in* i d'un identificador (ID) de cara a possibilitar que un usuari pugui desar més d'un projecte.

El diagrama entitat - relació de la base de dades es pot trobar a l'apèndix 1. En aquest es pot apreciar com s'han establert les relacions lògiques que la interfície d'usuari permet per cadascun dels elements contemplats.

### 5 RESULTATS

A continuació, en aquesta secció es mostren algunes proves de la funcionalitat de l'aplicació per a dos models d'infraestructura diferents.

Cal esmentar, que aquest projecte no es troba finalitzat en la seva totalitat, sinó que es continuarà fins la conclusió de l'estada de pràctiques a l'empresa. Per tant patirà canvis, millores i la inclusió dels elements de xarxa que manca implementar, així com els nous que es considerin.

Tal i com es mostra a la Figura 13 la interfície en un inici disposa d'un panell de selecció d'elements (VCN, Subnet, Internet Gateway, etc) els quals l'usuari pot arrossegar al diagrama en blanc on s'aniran disposant tots els elements de la infraestructura, així com les seves configuracions.

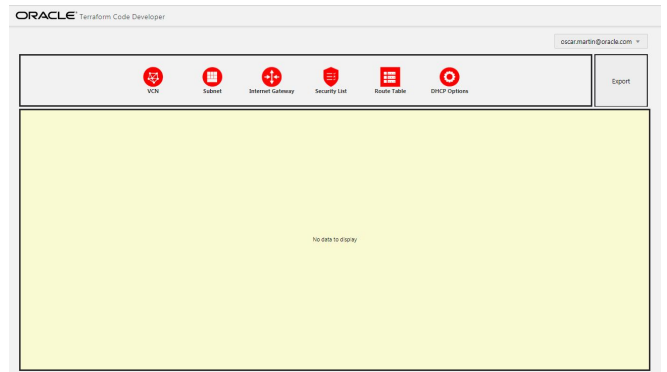


Figura 13. Vista inicial de la interfície (això canviarà)

Quan es deixa anar un element del panell dins la zona del diagrama apareix un diàleg com a la Figura 14, on s'han d'insertar els atributs de cada component necessaris per obtenir un correcte *script*. Si en comptes de deixar anar un element al diagrama es fa a sobre d'un altre, si els elements escollits es poden relacionar, aquests es lincaran visual i lògicament.

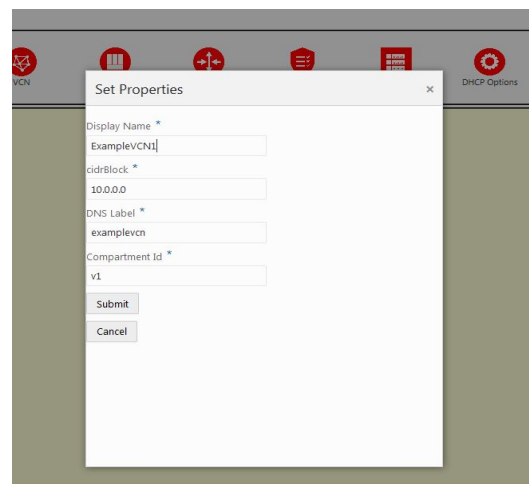


Figura 14. Atribució de propietats de la nova VCN creada un cop arrossegada a la interfície de disseny.

Els diferents components del model es poden anar ubicant pel diagrama per mitjà del *drag and drop*, on de la mateixa manera que en el cas anterior, si es disposa d'un element a sobre d'un altre queden relacionats com es pot veure a la Figura 15.

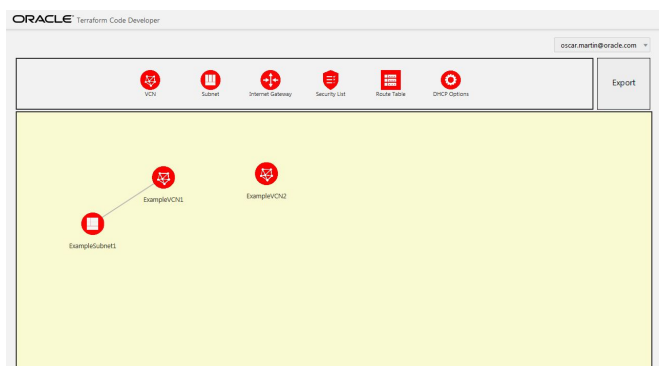


Figura 15. S'ha afegit una segona VCN a la infraestructura i s'ha atribuït una subxarxa a la primera VCN.

Un cop l'usuari ha dissenyat el model d'interfície de xarxa, pot clicar sobre el botó "Export". En aquest moment s'obre un diàleg on el client decideix on vol desar l'*script* resultant, tal i com es mostra a la Figura 16.

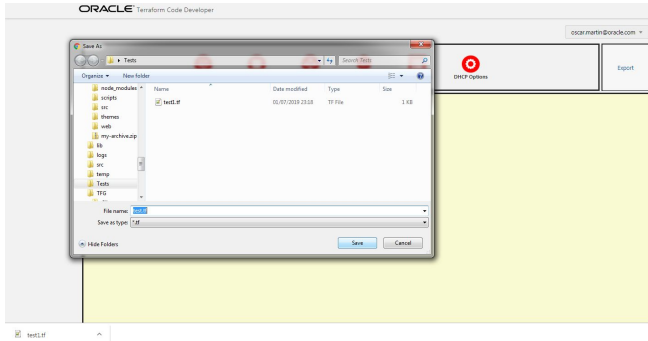


Figura 16. Descàrrega del *script* Terraform (amb extensió .tf).

A la Figura 17 s'observa l'*script* de Terraform equivalent al model d'infraestructura de xarxa dissenyat llest per fer-lo córrer a l'Oracle Cloud.

```

1 variable "tenancy_ocid" {}
2 variable "user_ocid" {}
3 variable "fingerprint" {}
4 variable "private_key_path" {}
5 variable "compartment_ocid" {}
6 variable "region" {}
7
8 provider "oci" {
9   tenancy_ocid = "${var.tenancy_ocid}"
10  user_ocid    = "${var.user_ocid}"
11  fingerprint  = "${var.fingerprint}"
12  private_key_path = "${var.private_key_path}"
13  region       = "${var.region}"
14 }
15
16
17 resource "oci_core_virtual_network" "vcn1" {
18   cidr_block      = "10.0.0.0/16"
19   compartment_id = "${var.compartment_ocid}"
20   display_name    = "ExampleVCN1"
21   dns_label       = "examplevcn1"
22 }
23
24 resource "oci_core_virtual_network" "vcn2" {
25   cidr_block      = "10.1.0.0/16"
26   compartment_id = "${var.compartment_ocid}"
27   display_name    = "ExampleVCN2"
28   dns_label       = "examplevcn2"
29 }
30
31 resource "oci_core_subnet" "subnet1" {
32   cidr_block      = "10.0.1.0/24"
33   display_name    = "ExampleSubnet1"
34   dns_label       = "examplesubnet1"
35   compartment_id = "${var.compartment_ocid}"
36   vcn_id          = "${oci_core_virtual_network.vcn1.id}"
37   security_list_ids = ["${oci_core_virtual_network.vcn1.default_security_list_id}"]
38   route_table_id  = "${oci_core_virtual_network.vcn1.default_route_table_id}"
39   dhcp_options_id = "${oci_core_virtual_network.vcn1.default_dhcp_options_id}"
40 }
41
length:1,322 lines:41 Ln:28 Col:32 Sel:0|0 Windows (CRLF) UTF-8 INS

```

Figura 17. *Script* resultant

## 6 CONCLUSIÓ

Per concloure aquest projecte podem dir que s'ha aconseguit una aplicació que ens proporcionarà automatització en el procés de gestió i configuració de

sistemes distribuïts de xarxa i arquitectures orientades a serveis en comptes de fer aquestes tasques de forma individual i manual a cadascuna de les màquines de la infraestructura. Sent doncs, una eina molt potent que ens concedirà una gran elasticitat, escalabilitat i versatilitat a l'hora de treballar a l'Oracle Cloud.

Així doncs, l'eina serà de gran utilitat i estalviarà un gran volum d'hores als empleats d'Oracle que treballen implementant aquest tipus de serveis i implantant infraestructures de xarxa al núvol.

Com a possibles millores hi trobem la verificació del format de les dades introduïdes als diàlegs dels diferents elements, la comprovació del rang de les IP's i que aquestes siguin vàlides, realitzar una interfície adaptable (responsive layout) amb l'ajuda del Oracle JET Cookbook, implementar un protocol de seguretat adicional de cara a no dependre del canal TLS, ja que en cas de fallar es perdria confidencialitat. Tal i com s'ha explicat a l'apartat de "Resultats", per la finalització d'aquest projecte, en el mes que resta, es tindran en compte les millores exposades.

Personalment, l'elaboració d'aquesta eina a Oracle ha sigut una experiència molt fructífera i enriquidora, ja que m'ha permès veure com afrontar un projecte de magnituds considerables des d'un punt inicial, passant per totes les seves fases (estudi, disseny, implementació, avaluació). A més, tot i haver sigut el principal desenvolupador el fet de respondre a l'equip supervisor, m'ha aportat tenir una mentalitat més oberta a l'hora d'escoltar i rebre suggeriments de millora o correccions.

Amb l'aplicació pràcticament finalitzada, considero que aportarà gran valor per als empleats d'Oracle, especialment al departament de Core. Quan des de l'empresa es va pensar en aquest projecte es desitjava automatitzar un gran nombre de tasques i estalviar temps en configuracions i gestions de clients amb infraestructures a l'Oracle Cloud, objectiu el qual s'ha aconseguit, pel que considero que després del resultat obtingut ha sigut un encert el desenvolupament d'aquesta eina.

## AGRAÏMENTS

En primer lloc, voldria agrair a l'empresa Oracle Corporation, no només l'oportunitat d'efectuar una estada de pràctiques sinó per donar-me l'oportunitat de realitzar aquest projecte sota la seva tutela, amb especial menció a en Joaquim Omella, per la seva ajuda i dedicació en tot el projecte. En segon lloc, al meu tutor a la universitat, en Jordi Pons, per les seves constants recomanacions, exigir-me sempre el màxim de cara a donar la millor versió pel seguiment tant en el Treball de Fi de Grau, com a l'estada de pràctiques. Per acabar, m'agradaria agrair a la meua família i amics que m'han donat suport al llarg de tot el grau, especialment a la meua mare, qui m'ha animat en tot aquest procés.

## BIBLIOGRAFIA

- [1] Sam Guckenheimer (4 d'Abril, 2017). Microsoft. What is Infrastructure as Code? Recuperat de: <https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code>
- [2] tic.Portal. ¿Qué es Cloud Computing? Recuperat de: <https://www.ticportal.es/temas/cloud-computing/que-es-cloud-computing>
- [3] Cisco. What is a Data Center. Recuperat de: <https://www.cisco.com/c/en/us/solutions/data-center-virtualization/what-is-a-data-center.html>
- [4] Puppet. Unparalleled infrastructure automation and delivery. Recuperat de: <https://puppet.com/>
- [5] HashiCorp. Write, Plan, and Create Infrastructure as Code. Recuperat de: <https://www.terraform.io/>
- [6] Microsoft. Configuraciones DSC. Recuperat de: <https://docs.microsoft.com/es-es/powershell/dsc/configurations/configurations>
- [7] Oracle. Acerca de Oracle. Recuperat de: <https://www.oracle.com/es/corporate>
- [8] Oracle Cloud (ORACLE). Why Oracle Cloud Infrastructure. Recuperat de: [https://cloud.oracle.com/en\\_US/cloud-infrastructure](https://cloud.oracle.com/en_US/cloud-infrastructure)
- [9] Oracle. What Are RESTful Web Services? Recuperat de: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqv.html>
- [10] ORACLE. Oracle Access Management. Recuperat de: <https://www.oracle.com/technetwork/middleware/id-mgmt/overview/index-090417.html>
- [11] Jose Antonio Castillo (5 de Gener, 2019). LDAP: Qué es y para que se utiliza este protocolo. Recuperat de: <https://www.profesionalreview.com/2019/01/05/ldap/>
- [12] Oracle. Welcome to Oracle JET. Recuperat de: <https://www.oracle.com/webfolder/technetwork/jet/index.html>
- [13] ORACLE. Helidon. Recuperat de: <https://helidon.io/#/>
- [14] Martin Fowler, James Lewis (25 Març, 2014). Microservices. Recuperat de: <https://martinfowler.com/articles/microservices.html>
- [15] W3Schools. Why Soap? Recuperat de: [https://www.w3schools.com/xml/xml\\_soap.asp](https://www.w3schools.com/xml/xml_soap.asp)
- [16] Code academy. What is Rest? Recuperat de: <https://www.codecademy.com/articles/what-is-rest>
- [17] YUI Team (10 Abril, 2007) JSON and Browser Security. Recuperat de: <https://yuiblog.com/blog/2007/04/10/json-and-browser-security/>
- [18] OpenSource. What is open source? Recuperat de: <https://opensource.com/resources/what-open-source>
- [19] ORACLE. World's Most Popular Open Source Database. Recuperat de: <https://www.oracle.com/MySQL/>
- [20] SCRUM.org. WHAT IS SCRUM? Recuperat de: <https://www.scrum.org/resources/what-is-scrum>
- [21] Taiga. Taiga PM Tool. Recuperat de: <https://taiga.io/>
- [22] Oracle.. JET Developer Cookbook. Recuperat de: <https://www.oracle.com/webfolder/technetwork/jet/jetCookbook.html?component=home&demo=all>
- [23] Oracle. The Oracle JET Architecture. Recuperat de: <https://docs.oracle.com/middleware/jet410/jet/developer/GUID-293CB342-196F-4FC3-AE69-D1226A025FBB.htm#JETDG113>
- [24] RequireJs. RequireJs. Recuperat de: <https://requirejs.org/>
- [25] Knockoutjs. Knockout. Recuperat de: <https://knockoutjs.com/>
- [26] Oracle. Arquitectura funcionamiento OAM. Recuperat de: <https://www.oracle.com/technetwork/es/documentation/317540-esa.pdf>
- [27] Wikipedia. Transport Layer Security. Recuperat de: [https://es.wikipedia.org/wiki/Transport\\_Layer\\_Security#TLS\\_1.2](https://es.wikipedia.org/wiki/Transport_Layer_Security#TLS_1.2)
- [28] Wikipedia. X.509. Recuperat de: <https://es.wikipedia.org/wiki/X.509>
- [29] Percona. Data at Rest. Recuperat de: <https://www.percona.com/blog/2016/04/08/mysql-data-at-rest-encryption/>
- [30] Java Platform , Enterprise Edition (Java EE) <https://www.oracle.com/technetwork/java/javaee/overview/index.html>

# APÈNDIX

## A1. Diagrama de base de dades Terraform IaC GUI Development Tool

