

Disseny i implementació d'una base de dades per la creació d'una aplicació que permet la gestió de les pràctiques d'estudiants a les empreses.

**Jorge Lara Martinez**  
Grau en Enginyeria Informàtica  
Base de dades

**Jordi Ferrer Duran**  
**Javier Jiménez Pelayo**

20 de Gener del 2023



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Disseny i implementació d'una base de dades per la creació d'una aplicació que permet la gestió de les pràctiques d'estudiants a les empreses.</i>
<b>Nom de l'autor:</b>	<i>Jorge Lara Martinez</i>
<b>Nom del consultor/a:</b>	<i>Jordi Ferrer Duran</i>
<b>Nom del PRA:</b>	<i>Javier Jiménez Pelayo</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>01/2023</i>
<b>Titulació o programa:</b>	<i>Grau d'Enginyeria Informàtica</i>
<b>Àrea del Treball Final:</b>	<i>Base de dades</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Base de dades, Pràctiques empresa, Control Estudiants Pràctiques</i>

### Resum del Treball

Aquest treball se centra en la problemàtica que tenen les universitats per fer els seguiments als estudiants mentre realitzen les pràctiques a les empreses. El CIC vol posar-hi una solució creant una aplicació on totes les universitats, estudiants i empreses tinguin accés, i que doni facilitat per fer el seguiment de tots els interessats.

L'aplicació consta de dues parts: una aplicació d'alt nivell com podria ser una pàgina web, aplicació Java..., i la gestió de dades mitjançant una base de dades relacional d'Oracle. En aquest treball es desenvolupa la base de dades.

La creació de la base de dades es fa mitjançant el mètode en cascada (waterfall), ja que els requisits estan ben detallats i no s'han de fer canvis constants. No obstant això, es divideix en tres fases on a cada fase es desenvolupa una part del projecte i es fan proves per veure'n l'avenç. La base de dades ha de tenir tres mòduls diferenciats: dades de les entitats principals, registre d'errors i mòdul d'estadístic.

En aquest treball s'han aplicat tècniques estudiades en altres matèries per poder optimitzar les dades emmagatzemades a les taules. Així i tot, s'han pogut adquirir coneixements i experiència amb la base de dades d'Oracle.

Com a resultat s'obté una base de dades que compleix els requisits donats pel client i un joc de proves per poder fer-ne l'avaluació.

**Abstract (in English, 250 words or less):**

The following work involves issues at universities when they want to perform student followups at companies practise hirings. CIC wants to solve it, creating an application where each university, students and companies, will have access and offer easy followups to everybody interested.

The application is divided in two parts: A high level one like a webpage, Java application,... and data management using a relational data base like Oracle. At this essay, the data base work is explained further.

The data base creation is performed in a waterfall methodology, given the detailed requirements and few constant changes. Even though, it is divided in three steps and each one of them is developed and tested to give a detailed followup. The data base should have three differentiated modules: Principal entities data, error logs and statistic module.

Techniques learned on other courses have been taken into account to write this essay, in order to optimize data that will be stored at the data base. Furthermore, new knowledge and experience has been acquired while working with Oracle.

As a result, we obtained a database with the data required by the customer among a test suite to perform further evaluations.

## Llista de figures

- Figura 1: Calendari general de les fases definides al treball
- Figura 2: Calendari amb detall de les activitats del treball
- Figura 3: Càlcul del cost mensual d'una base de dades al núvol d'Oracle
- Figura 4: Càlcul del cost mensual d'una base de dades al núvol d'Amazon
- Figura 5: Model ER de la base de dades relacional.
- Figura 6: SQL de taula amb integritat referencial d'eliminació en cascada
- Figura 7: Creació d'una BD amb sqldeveloper d'Oracle
- Figura 8: Creació d'una taula amb sqldeveloper d'Oracle mitjançant GUI
- Figura 9: Creació d'un arxiu de comandes SQL amb sqldeveloper d'Oracle
- Figura 10: Llista de l'arbre de taules i procediments a l'sqldeveloper
- Figura 11: Creació de procediments mitjançant GUI a l'sqldeveloper
- Figura 12: Creació de procediments mitjançant codi SQL a l'sqldeveloper
- Figura 13: Procediments d'afegir, modificar i eliminar de la Fase I
- Figura 14: Model ER del registre d'accions
- Figura 15: Model ER del mòdul estadístic
- Figura 16: Model ER del producte
- Figura 17: Sortida d'error per dada no coherent
- Figura 18: Vista d'un procediment amb funcions
- Figura 19: Creació d'una funció mitjançant el GUI de l'sqldeveloper
- Figura 20: Exemple de codi per la creació d'una funció mitjançant codi SQL de l'sqldeveloper
- Figura 21: Creació d'un disparador mitjançant el GUI de l'sqldeveloper
- Figura 22: Creació d'un índex mitjançant el GUI de l'sqldeveloper
- Figura 23: Exemple de codi per la captura d'un error mitjançant codi SQL de l'sqldeveloper
- Figura 24: Exemple de conversió data-POSIX

- Figura 25: Exemple d'implementació de tècniques DataWarehouse
- Figura 26: Exemple de taula de dades agrupades per temes
- Figura 27: Especialització de les entitats Estudiant, Professor i Treballador
- Figura 28: Exemple de funció i procediment amb sessions
- Figura 29: Exemple de millora amb tècniques DataWarehouse

# Índex

1. Introducció.....	3
1.1 Context i justificació del Treball .....	3
1.2 Objectius del Treball.....	4
1.3 Enfocament i mètode seguit.....	4
1.4 Planificació del Treball.....	5
1.5 Cost econòmic.....	8
1.6 Breu resumari de productes obtinguts .....	10
1.7 Breu descripció dels altres capítols de la memòria .....	10
2. Anàlisi de Riscos .....	13
3. Fase I .....	16
3.1 Introducció.....	16
3.2 Anàlisi de requisits .....	16
3.3 Criteris d'acceptació .....	17
3.4 Instal·lació del programari .....	17
3.5 Disseny conceptual .....	18
3.5.1 Decisions de disseny.....	19
3.6 Disseny lògic .....	21
3.6 Normalització.....	23
3.6.1 Primera Forma Normal (1FN).....	23
3.6.2 Segona Forma Normal (2FN).....	24
3.6.3 Tercera Forma Normal (3FN).....	24
3.6.4 Quarta Forma Normal(4FN) .....	24
3.6.5 Quinta Forma Normal (1FN).....	25
3.7 Disseny físic .....	25
3.7.1 PL/SQL.....	27
3.7.2 Creació de procediments .....	28
3.9 Proves i optimitzacions.....	29
4. Fase II .....	30
4.1 Introducció.....	30
4.2 Anàlisi de requisits. ....	30
4.3 Criteris d'acceptació .....	31
4.4 Disseny Conceptual.....	31
4.4.1 Entitats del registre d'accions.....	31
4.4.2 Entitats del mòdul estadístic.....	32
4.4.3 Entitats resultants fase II.....	33
4.4.4 Decisions de disseny.....	33
4.5 Disseny lògic .....	34
4.5.1 Registre d'accions .....	35
4.5.2 Mòdul estadístic .....	35
4.6 Disseny físic .....	38
4.6.1 Funcions.....	38
4.6.2 Creació de funcions.....	39
4.6.3 Creació de disparadors .....	40
4.6.4 Creació d'índex .....	41
4.6.5 Captura d'errors .....	42
4.7 Mòdul estadístic .....	42

4.7.1 Disparadors (Triggers) .....	42
4.7.2 Llistat d'estadístiques bàsiques.....	43
4.7.3 Decisions de disseny.....	44
4.8 Tècniques DataWarehouse (magatzem de dades) .....	46
4.8.1 Integrat .....	47
4.8.2 Històric .....	47
4.8.3 No Volàtil.....	47
4.8.4 Orientat a temes.....	48
4.9 Joc de dades .....	49
4.10 Proves i optimitzacions .....	50
5. Entrega Final .....	51
6. Futur .....	52
6.1 Introducció.....	52
6.1 Seguretat del Sistema .....	52
6.2 Aplicació i millora de tècniques de magatzem de dades .....	53
6.3 Altres millores .....	54
7. Conclusions.....	56
7.1 Fase I .....	56
7.2 Fase II .....	57
7.3 Entrega final .....	57
8. Glossari .....	58
9. Bibliografia.....	61
10. Annexos .....	64
10.1 Manual d'importació del joc de proves .....	64



# 1. Introducció

## 1.1 Context i justificació del Treball

Molts estudiants universitaris no tenen cap coneixement del món laboral, i els que en tenen, normalment no és un coneixement en el seu àmbit d'estudi. Per l'estudiant, això pot tornar-se un problema un cop finalitzada la seva etapa formativa, tant per la manca d'informació com la falta d'experiència enfront del funcionament d'una empresa i el lloc de treball que pot ocupar.

Per introduir-los al món laboral i permetre'ls obtenir aquesta experiència, durant la seva etapa formativa tenen la possibilitat de realitzar pràctiques a les empreses. És important la seva realització, perquè a més d'experimentar en primera persona com funcionen les empreses, poden aplicar els coneixements adquirits i aprendre'n d'altres mitjançant la pràctica. A part d'aplicar els coneixements, també els permet créixer i guanyar confiança per començar una nova etapa al món laboral un cop finalitzats els estudis.

Per efectuar aquestes pràctiques, els estudiants poden buscar una empresa pel seu compte, o bé escollir-ne alguna de les llistes que tenen les universitats a les bases de dades. Un cop triada, es té l'opció de signar un conveni de pràctiques entre l'alumne, la universitat i l'empresa (si totes les parts estan d'acord), i aleshores l'alumne passa a treballar en pràctiques per l'empresa.

La principal funció de la universitat mentre l'estudiant fa les pràctiques és assegurar-ne que tant l'estudiant com l'empresa respecten els punts acordats al conveni signat, és a dir, fa un seguiment durant el període de pràctiques perquè totes les parts respectin els punts signats: s'assegura que l'estudiant fa les hores pactades, l'empresa li encomana un tutor... També assumeix la gestió de possibles inspeccions de treball a l'empresa que requereixi el corresponent departament de Govern respecte al conveni signat.

Actualment, el seguiment de l'alumne durant la realització d'aquestes pràctiques és un procés molt difícil de realitzar. Hi ha molts aspectes a supervisar (horaris, treball a fer...) i intervenen moltes persones al procés (el responsable de RRHH de l'empresa, el tutor de la universitat, l'estudiant...) Com a conseqüència, al tutor de la universitat li resulta una missió molt problemàtica i complexa fer el seguiment de l'alumne durant el període de pràctiques.

Per aquest motiu, el Consell Interuniversitari de Catalunya (per les seves sigles CIC) vol una aplicació que estigui disponible en qualsevol moment, sigui segura i ràpida de consultar. Mitjançant aquesta aplicació, els tutors universitaris tindran més facilitat i seran més eficients per controlar les pràctiques dels alumnes i poder-ne fer el seguiment. A més, aquesta eina millorarà la comunicació entre la universitat i l'empresa.

Aquest treball se centrarà en la construcció d'una base de dades que compleixi amb els requisits de l'aplicació, emmagatzemi les dades d'una forma segura i

eficaç, i s'asseguri que les dades estiguin sempre disponibles (alta disponibilitat) en el menor temps possible (rapidesa).

## 1.2 Objectius del Treball

Els objectius d'aquest treball són:

- Creació d'una BD relacional per emmagatzemar tota la informació requerida pel client.
- La consulta a les dades ha de ser ràpida
- Permetre grans volums de dades sense perdre eficiència ni eficàcia
- Creació d'un repositori estadístic per la seva consulta
- Creació d'un registre d'accions i errors dels usuaris.

L'assoliment d'aquests objectius s'aconsegueixen mitjançant les següents tasques:

- Creació d'un calendari de treball
- Anàlisi de riscos
- Criteris d'acceptació
- Identificació, definició i anàlisi dels requisits
- Disseny conceptual
- Disseny lògic
- Disseny Físic
- Joc de dades

Un cop finalitzat el treball, el producte resultant ha de ser capaç de:

1. Emmagatzemar tota la informació necessària per poder fer el seguiment dels estudiants a les empreses
2. Permetre la consulta de les dades de forma ràpida i senzilla per part dels tutors universitaris, treballadors de les empreses i qualsevol entitat amb permís per veure les dades.
3. Permetre l'avaluació dels estudiants respecte les pràctiques realitzades a l'empresa.
4. Emmagatzematge de dades estadístiques per la seva consulta i anàlisi.
5. Creació d'un registre d'accions dutes a terme, com per exemple l'alta d'una empresa al sistema o la inscripció d'un estudiant a una oferta de treball, i enregistrar si l'acció ha estat correcta o ha donat algun error.

## 1.3 Enfocament i mètode seguit

Per assolir els objectius del treball i aconseguir un producte de qualitat se seguirà una metodologia predictiva (waterfall), que consisteix a executar cadascuna de les activitats de manera esglaonada. És el millor mètode per la creació d'una BD

on s'han definit tots els requisits des d'un començament, ja que per crear-la s'han de començar unes activitats concretes abans d'unes altres. Algunes activitats són complementàries, però no poden començar al mateix temps, pel fet que requereix informació prèvia per poder-la començar. Per exemple, no es pot començar el disseny lògic sense tindre el conceptual començat.

L'opció d'aquest mètode i no un d'àgil és a conseqüència de les necessitats ben definides pel client, cosa que permet no tenir la necessitat d'anar revisant el treball cada cert temps ni fer entregues per provar el producte i decidir si és necessari fer canvis. També s'ha de tenir en compte que encara que aquest treball té tres entregues amb data ben definida (Fase I, Fase II i Entrega Final), les revisions de cadascun d'ells té la finalitat d'avaluar l'evolució del treball i confirmar que l'entrega final del producte es realitzarà amb èxit (tots els requisits demanats), no per fer canvis als requisits que no han estat demanats pel CIC mentre dura l'execució d'aquest treball.

S'ha de fer menció respecte a la decisió de fer coincidir amb l'entrega de cada PAC un producte mínim viable per la seva avaluació, i poder verificar que s'estan seguint els requisits demanats pel CIC i evitar possibles problemes en una fase més avançada del treball. En concret, es faran les següents entregues:

- Una versió *alpha* del producte a la PAC 2, anomenada FASE I on es farà una entrega d'un joc de dades i procediments essencials pel funcionament del producte. L'objectiu és la creació d'una base i l'avaluació d'aquesta per poder verificar el compliment d'uns requisits mínims.
- Una versió *beta* del producte a la PAC 3, anomenada FASE II on es farà una entrega d'un joc de dades i procediments avançats pel funcionament del producte. L'objectiu és tenir un producte ja acabat i en fase de proves per fer l'avaluació i verificació dels requisits demanats. A més, també es vol comprovar si les actualitzacions i modificacions al producte són fàcils de realitzar
- Una última entrega on s'hauran corregit els errors o deficiències detectades a la versió beta. A més, també es lliurarà tota la documentació requerida (presentació, memòria i producte)

#### 1.4 Planificació del Treball

El treball per la realització del projecte s'ha dividit en cinc fases que corresponen amb entregues o dates clau del projecte:

- Pla de Treball: Consisteix en l'entrega de la documentació del pla de treball del projecte.
- Fase I: Consisteix en l'entrega de la documentació i una versió *alpha* del producte per comprovar-ne l'evolució. En aquesta entrega ha d'haver-hi:

- Entitats principals: Les entitats principals del producte que són:
  - Companyia i treballadors
  - Universitats, estudiants i professors
  - Ofertes de treball, contractes i el pla formatiu
  - Inspeccions de treball
- Procediments: Els procediments d'afegir, modificar i eliminar de les entitats. També es generen els procediments necessaris per al correcte flux de les dades, per exemple els passos per validar un candidat.
- Joc de dades: Per poder comprovar que les relacions entre les entitats són correctes.
- Consultes: Per fer les comprovacions de les dades i poder comprovar la integritat
- Fase II: Consisteix en l'entrega de la documentació i una versió beta del producte per comprovar-ne l'evolució. Aquesta entrega consistirà en una revisió en el disseny i les entitats principals de la Fase I per afegir el registre d'errors i accions, i el repositori estadístic requerit pel client. En aquesta entrega ha d'haver-hi:
  - Entitats: Es revisen les entitats de la Fase I i s'afegeix el registre d'errors i accions, i les entitats necessàries per fer el mòdul estadístic
  - Procediments: Es modifiquen els procediments per afegir codis d'error i guardar la informació a la BD.
  - Joc de dades: Es revisen en funció de les necessitats generades pel mòdul estadístic i per assegurar-ne la integritat de les dades.
  - Consultes: S'afegeixen les consultes requerides al mòdul estadístic amb les seves restriccions
- Entrega Final: Consisteix en l'entrega de la documentació i el producte finalitzat a més d'una presentació. En aquest punt es fa una revisió per millorar les dades i assegurar-ne l'èxit del producte.
- Tribunal D'avaluació: Exposició i resolució de dubtes referents al producte resultant

El calendari de treball a seguir serà el següent:

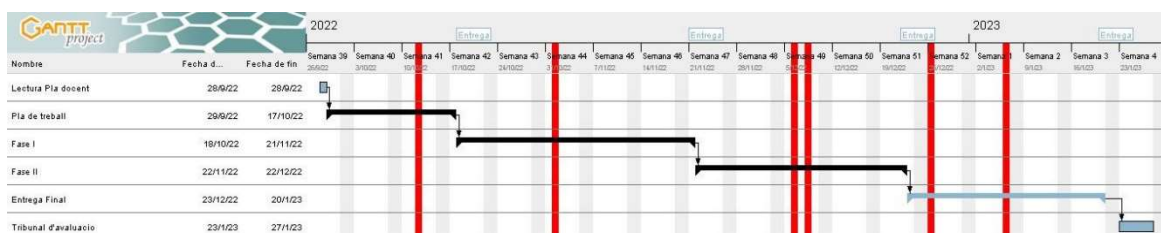


Figura 1: Calendari general de les fases definides al treball

A la següent taula es poden veure en detall les activitats de cada entrega:

<b>Nom</b>	<b>Inici</b>	<b>Fi</b>	<b>Dies</b>	<b>Total Hores</b>
<b>Lectura Pla docent</b>	<b>28/9/22</b>	<b>28/9/22</b>	<b>1</b>	<b>1</b>
<b>Pla de treball</b>	<b>29/9/22</b>	<b>17/10/22</b>	<b>12</b>	<b>24</b>
Documentació	29/9/22	17/10/22	12	12
Anàlisi de riscos	3/10/22	5/10/22	3	3
Planificació (Gantt)	6/10/22	11/10/22	4	8
Entrega	18/10/22	18/10/22	1	1
<b>Fase I</b>	<b>18/10/22</b>	<b>21/11/22</b>	<b>24</b>	<b>108</b>
Documentació	18/10/22	21/11/22	24	24
Anàlisi de Requisits	18/10/22	20/10/22	3	9
Instal·lació del programari	21/10/22	21/10/22	1	1
Disseny Conceptual	21/10/22	27/10/22	5	10
Disseny Logic	25/10/22	28/10/22	4	4
Disseny Físic	25/10/22	3/11/22	7	21
Joc de dades	27/10/22	9/11/22	9	18
Proves i Optimitzacions	8/11/22	21/11/22	10	20
Entrega	21/11/22	21/11/22	1	1
<b>Fase II</b>	<b>22/11/22</b>	<b>22/12/22</b>	<b>21</b>	<b>101</b>
Documentació	22/11/22	22/12/22	21	21
Anàlisi de Requisits	23/11/22	25/11/22	3	9
Disseny Conceptual	28/11/22	01/12/22	4	8
Disseny Lògic	30/11/22	02/12/22	3	3
Disseny Físic	02/12/22	07/12/22	3	9
Joc de dades	05/12/22	12/12/22	4	12
Proves i Optimitzacions	09/12/22	21/12/22	9	36
Lliurament	22/12/22	22/12/22	1	1
<b>Entrega Final</b>	<b>23/12/22</b>	<b>20/1/23</b>	<b>19</b>	<b>66</b>
Documentació	23/12/22	20/1/23	19	38
Optimitzacions i Millores	28/12/22	10/1/23	9	18
Presentació	10/1/23	20/1/23	9	9
Entrega	20/1/23	20/1/23	1	1
<b>Tribunal d'avaluació</b>	<b>23/1/23</b>	<b>27/1/23</b>	<b>N.D</b>	<b>N.D</b>
<b>Total</b>			<b>77</b>	<b>300</b>

A l'hora de realitzar el calendari, s'ha tingut en compte els següents dies festius que no són en caps de setmana[1][2]:

- Dimecres 12 d'octubre del 2022: Festa nacional
- Dimarts 1 de novembre del 2022: Tots Sants
- Dimarts 6 de desembre del 2022: Dia de la constitució
- Dijous 8 de desembre del 2022: La Puríssima.
- Dilluns 26 de desembre del 2022: Sant Esteve
- Divendres 6 de gener del 2023: Reis

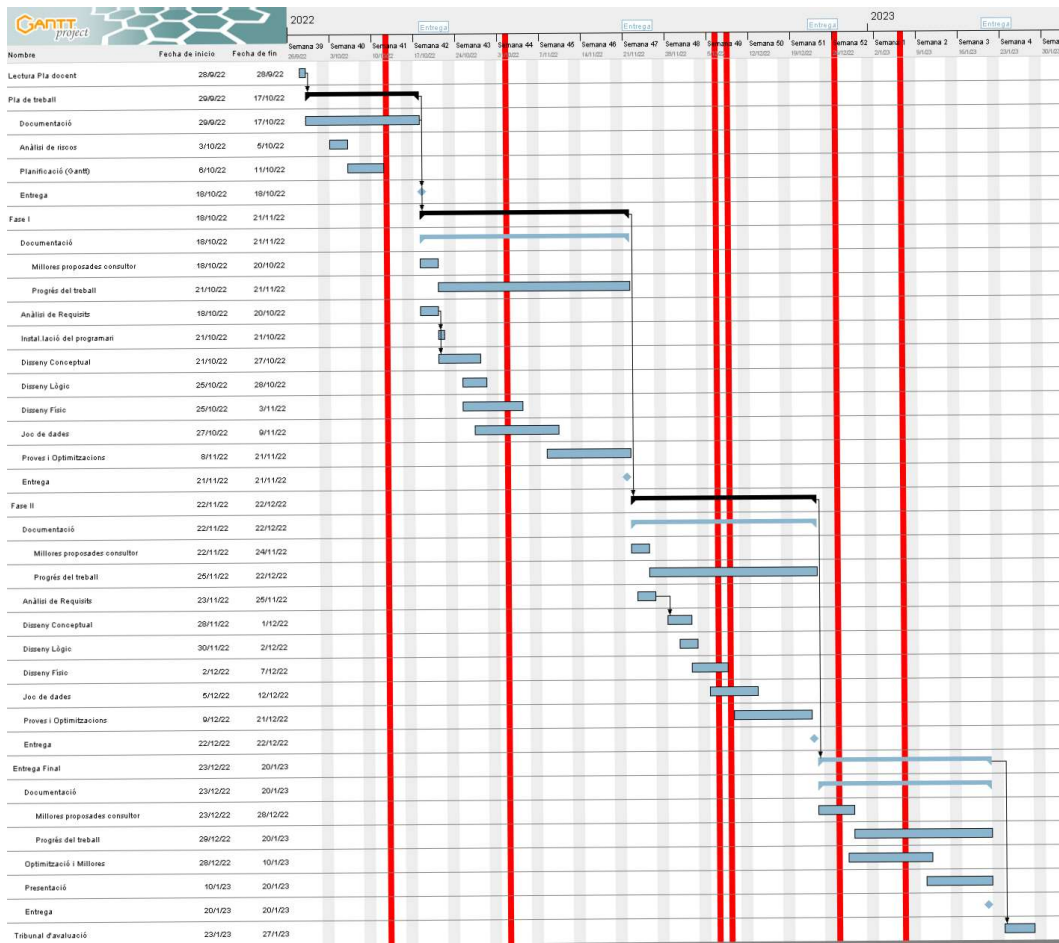


Figura 2: Calendari amb detall de les activitats del treball

## 1.5 Cost econòmic

Per dur a terme aquest projecte es necessita un servidor amb un Sistema Gestor de Base de Dades (SGBD) instal·lat. Per requisits del treball, aquest SGBD ha de ser Oracle, cosa que fa que les altres opcions com MySQL o MariaDB quedin descartades. També queden descartades pel mateix motiu qualsevol BD noSQL com MongoDB. Oracle té un cost, però per aquest treball s'ha escollit la versió gratuïta amb cost 0.

[1] <https://web.gencat.cat/ca/actualitat/reportsatges/calendari-laboral/calendari-laboral-2022/>

[2] <https://web.gencat.cat/es/actualitat/reportsatges/calendari-laboral/calendari-laboral-2023/>

Tot programari necessita un Sistema Operatiu (SO), en aquest cas es pot decidir per un sistema Windows o un sistema Linux. Si el producte s'ha de posar en producció, la millor opció seria un servidor Linux amb RedHat[3] o Debian[4]. La principal diferència entre els dos sistemes recau en el fet que RedHat és de pagament i Debian és gratuït. Com que és un treball acadèmic i l'ordinador que s'utilitzarà és un sistema Windows, s'ha optat per aquesta opció sense cost econòmic. A més, en cas que l'ordinador principal es trenqués, la migració a un altre ordinador seria més fàcil, ja que els ordinadors de reserva tenen Windows instal·lat.

Respecte al maquinari, la millor opció seria un servidor allotjat a una VM al núvol, per exemple a Amazon EC2 [5], perquè té una alta disponibilitat. Aquesta opció té un cost econòmic que per aquest projecte no és necessari, ja que es tracta d'un treball acadèmic. Per aquest motiu la instal·lació del SGBD es farà de forma local sense cost econòmic.

Finalment, hi ha un cost econòmic pel subministrament elèctric i la connexió a internet. Com en els casos anteriors, el cost entra dins el pressupost de la llar on viu l'estudiant.

En cas de posar-ho en producció, hi ha múltiples opcions: servidors allotjats en VM al núvol, serveis de SGBD sota demanda... Entre tots ells s'han seleccionat dues opcions per la facilitat de posada en marxa i el preu:

- VM gestionada per Oracle, amb una base de dades estàndard. Té alta disponibilitat i és escalable per poder augmentar les prestacions a mesura que van creixent les dades. Té un cost mensual de 297.53 €[6].

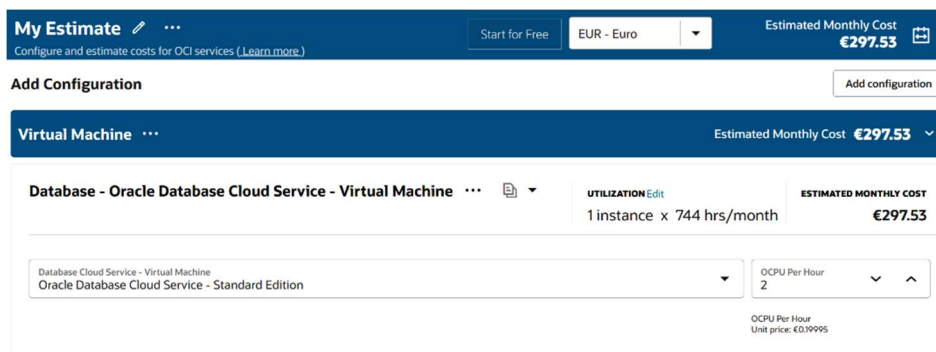


Figura 3: càlcul del cost mensual d'una base de dades al núvol d'Oracle

- Servei Amazon RDS per Oracle [7] amb base de dades estàndard. Té alta disponibilitat i és escalable. Té un cost mensual de 317.30\$ [8]. A aquest preu s'hauria d'afegir la llicència d'Oracle, i per fer-ho s'ha de contactar amb el departament de vendes.

[3] <https://www.redhat.com/es/technologies/linux-platforms/enterprise-linux>

[4] <https://www.debian.org/index.es.html>

[5] <https://aws.amazon.com/es/ec2/>

[6] <https://www.oracle.com/cloud/costestimator.html?source=:ow:o:h:po:OHPPanel1nav0625>

[7] [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Oracle.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Oracle.html)

[8] <https://calculator.aws/#/estimate>

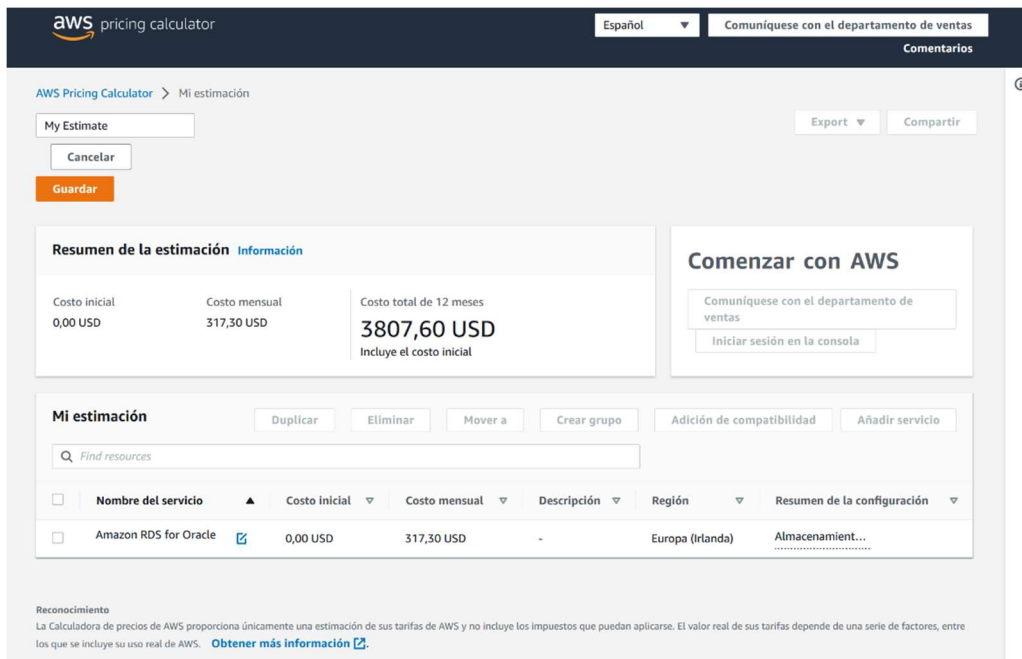


Figura 4: càlcul del cost mensual d'una base de dades al núvol d'Amazon

D'entre totes dues, la primera opció (servei d'Oracle) és la més econòmica.

## 1.6 Breu sumari de productes obtinguts

Els productes resultant un cop finalitzat el treball són:

- Memòria del treball amb tota la informació rellevant del treball: pla de treball, decisions preses, evolució del producte...
- Scripts de creació de la BD per la seva importació en qualsevol altre gestor que compleixi els requisits demanats (SGBD Oracle)
- Joc de dades per la seva importació a la BD. Permetrà fer proves al producte per la seva verificació.
- Joc de proves i avaluació per poder confirmar el funcionament segons els requisits demanats pel CIC.

Presentació del treball amb una explicació l'elaboració del producte i algunes decisions preses durant la seva creació.

## 1.7 Breu descripció dels altres capítols de la memòria

Els següents capítols del treball corresponen amb les fases de creació del producte i el seu lliurament. Cada fase es compon del cicle de disseny → implementació → proves per poder garantir l'evolució del projecte.



Els capítols són:

- Anàlisi de riscos: Identifiquem i analitzem els principals riscos que pot haver-hi per no poder finalitzar el treball a temps. Es proposen solucions als possibles problemes detectats.

- Fase I:

En aquesta fase, es vol fer entrega d'una versió *alpha* del producte amb les entitats més importants i les seves relacions. Aquesta entrega ha de ser funcional amb el compliment d'uns requeriments mínims que en garanteix el bon funcionament. Aquests requisits són:

- Requisits del client: Identifiquem els requisits demanats pel client que han de ser funcionals al sistema
- Criteris d'acceptació: Identifiquem els criteris que ha de tenir el producte per al seu èxit.
- Principals entitats i les seves relacions:
  - Companyia i treballadors
  - Universitats, estudiants i professors
  - Ofertes de treball, contractes i el pla formatiu
  - Inspeccions de treball
- Procediments: Els procediments necessaris per poder fer altes, modificacions i eliminar dades a la BD. També es crearan els procediments necessaris per gestionar el flux de dades.
- Consultes: Les consultes necessàries per poder verificar les dades entrades al sistema.
- Joc de dades: per poder verificar les entitats i la relació entre elles.

- Fase II:

En una segona fase, es vol fer entrega d'una versió *beta* del producte amb totes les funcionalitats demanades. Això inclou:

- Registre d'accions: Implementació d'un registre d'accions dels usuaris que utilitzen el producte. Ha de quedar registrat qualsevol acció i el resultat.
- Mòdul estadístic: Fer els canvis necessaris per poder registrar les dades estadístiques necessàries.

- Gran volum de dades: Millorar l'emmagatzematge de la BD per implementar tècniques de magatzem de dades (Data Warehouse) i permetre un gran volum de dades sense perdre rendiment
- Entrega Final:

A l'entrega final, es lliurarà el producte amb un joc de proves i tot el codi necessari per poder crear, modificar, eliminar i consultar tant les taules i procediments com les dades emmagatzemades.

També es farà entrega d'una presentació del producte amb una explicació del funcionament

## 2. Anàlisi de Riscos

Els principals riscos[9] identificats per aquest treball són:

<b>Risc</b>	Caiguda de la web de la UOC
<b>Descripció</b>	Caiguda de la web de la UOC on s'ha d'entregar els arxius de seguiment del treball
<b>Conseqüència</b>	No es podria entregar la documentació del treball abans de la data fixada
<b>Probabilitat</b>	Baixa
<b>Impacte</b>	Alt.
<b>Mitigació</b>	Intentar l'entrega de la documentació del treball un dia abans de la data fixada. En cas d'error o problemes en l'entrega posar-se amb contacte amb el consultor.

<b>Risc</b>	Incidència del maquinari
<b>Descripció</b>	L'ordinador, com tota màquina, pot tenir alguna avaria o trencar-se.
<b>Conseqüència</b>	No es podria continuar amb la mateixa màquina fins a la seva reparació o substitució.
<b>Probabilitat</b>	Baix
<b>Impacte</b>	Alt
<b>Mitigació</b>	- Fer còpies de seguretat al núvol amb serveis com Google Drive o Microsoft One Drive.  - Es disposa d'un ordinador portàtil en cas de necessitat.

<b>Risc</b>	Pèrdua del subministrament elèctric
<b>Descripció</b>	A conseqüència de problemes aliens a l'estudiant (com una tempesta elèctrica) es pot perdre el subministrament elèctric.
<b>Conseqüència</b>	Sense electricitat no es podria continuar amb el treball per la seva entrega dins els terminis previstos. A més, sense electricitat també cau la connexió d'internet.

[9] Llibre: RODRÍGUEZ, JOSE RAMÓN i MARINÉ JOVÉ, Pere. Planificació del projecte. UOC - PID\_00247936

<b>Probabilitat</b>	Baix
<b>Impacte</b>	Alt
<b>Mitigació</b>	Fins al restabliment del subministrament, es disposen de les següents opcions: <ul style="list-style-type: none"> <li>• Ordinador portàtil.</li> <li>• Connexió 4G mitjançant connexió mòbil.</li> </ul>

<b>Risc</b>	Pèrdua de la connexió a internet
<b>Descripció</b>	A conseqüència de problemes aliens a l'estudiant es pot perdre la connexió a internet.
<b>Conseqüència</b>	Sense connexió a internet es pot continuar amb el projecte, però no es pot entregar la documentació en els terminis previstos
<b>Probabilitat</b>	Baix
<b>Impacte</b>	Baix
<b>Mitigació</b>	Fins al restabliment del servei, es pot utilitzar la connexió mòbil per fer les entregues.

<b>Risc</b>	Malaltia de l'estudiant
<b>Descripció</b>	El treball s'ha de fer a les estacions de la tardor i l'hivern, cosa que fa que sigui molt probable que l'estudiant agafi una malaltia.
<b>Conseqüència</b>	Podria causar un endarreriment en el projecte i l'entrega de la documentació requerida.
<b>Probabilitat</b>	Mitjana
<b>Impacte</b>	Mitja
<b>Mitigació</b>	Fer hores els caps de setmana o els dies festius, en cas que ho requereixi el calendari, per poder fer les entregues a la data fixada.

<b>Risc</b>	Ingrés clínic
<b>Descripció</b>	L'estudiant té dona i filla, i qualsevol pot requerir un ingrés clínic d'urgència.
<b>Conseqüència</b>	Podria causar un endarreriment en el projecte i l'entrega de la documentació requerida en els terminis previstos.
<b>Probabilitat</b>	Baix
<b>Impacte</b>	Mitja
<b>Mitigació</b>	Fer hores els caps de setmana o els dies festius, en cas que ho requereixi el calendari, per poder fer les entregues a la data fixada. Comunicar-ho al consultor en cas que l'ingrés sigui més greu.

## 3. Fase I

### 3.1 Introducció

En aquesta primera fase del treball es crearà el producte amb les principals entitats amb les seves relacions i unes funcionalitats mínimes.

La finalitat és aconseguir una versió mínima del producte per poder fer una primera avaluació i comprovar si l'estructura i les relacions entre les entitats són les apropiades per les necessitats requerides del producte o es necessiten correccions.

En aquesta fase també s'identificaran les necessitats del client respecte al funcionament del producte, i quins són els objectius (criteris d'acceptació) mínims per acceptar-lo.

Un altre aspecte d'aquesta primera fase és la necessitat la instal·lació del programari i la creació de l'entorn de treball, és a dir, la instal·lació del SGBD i la creació de la base de dades per treballar les taules, els procediments, les dades i les sentències SQL per assolir el propòsit d'aquesta fase.

### 3.2 Anàlisi de requisits

S'han pogut recollir els següents requisits[9] del projecte:

Codi	Descripció
<b>R001</b>	S'ha de poder identificar els diferents treballadors de l'empresa que prendran part en el procés. És important la seva funció dins l'empresa
<b>R002</b>	Si l'empresa no ha estat acceptada pel CIC, no pot publicar llocs de treball per pràctiques
<b>R003</b>	Un treballador serà el responsable de l'estudiant en pràctiques
<b>R004</b>	Perquè un alumne sigui seleccionat per una empresa per fer les pràctiques, ha de passar dues validacions: <ul style="list-style-type: none"><li>• Les universitats han de validar les inscripcions</li><li>• Les empreses han de seleccionar l'estudiant que ha estat validat per la universitat</li></ul>
<b>R005</b>	El treballador de l'empresa responsable de l'estudiant que ocupa un lloc de treball en pràctiques, és l'única persona que pot seleccionar l'estudiant
<b>R006</b>	S'han d'emmagatzemar totes les preguntes i respostes que es fan a l'entrevista de l'estudiant que opta al lloc de treball en pràctiques

[9] Llibre: RODRÍGUEZ, JOSE RAMÓN i MARINÉ JOVÉ, Pere. Planificació del projecte. UOC - PID\_00247936

<b>R007</b>	Un estudiant que ocupa un lloc de treball en pràctiques, ha de rebre una valoració final d'apte o no apte un cop finalitzat el contracte
<b>R008</b>	Un contracte en pràctiques té una durada d'entre 6 mesos i 1 any aproximadament
<b>R009</b>	El treballador responsable de l'estudiant a l'empresa ha de: <ul style="list-style-type: none"> <li>• Preparar un programa formatiu per l'estudiant</li> <li>• Presentar informes parcials del desenvolupament de l'estudiant</li> <li>• Presentar un informe final amb comentari objectiu i amb una valoració d'apte o no apte</li> </ul>
<b>R010</b>	L'empresa pot rebre inspeccions de treball per part del Govern, i han de quedar registrades al sistema.

### 3.3 Criteris d'acceptació

S'han de complir aquests criteris[9] per poder avaluar els resultats del treball

<b>Codi</b>	<b>Descripció</b>
<b>C001</b>	El SGBD es Oracle
<b>C002</b>	El disseny de la BD ha de permetre l'emmagatzematge de grans quantitats de dades
<b>C003</b>	Existeixen com a mínim tres funcions per cadascuna de les entitats de la base de dades, que són: <ul style="list-style-type: none"> <li>• Afegir</li> <li>• Modificar</li> <li>• Eliminar</li> </ul>
<b>C004</b>	Totes les dades han de quedar enregistrades per poder fer consultes

### 3.4 Instal·lació del programari

Per fer funcionar el programari, es necessita:

- Un servidor o un ordinador personal (PC)
- Un Sistema Operatiu (SO)
- Un gestor de base de dades (SGDB)
- Un client per la connexió a la base de dades

[9] Llibre: RODRÍGUEZ, JOSE RAMÓN i MARINÉ JOVÉ, Pere. Planificació del projecte. UOC - PID\_00247936

Tot programari té uns requisits mínims per poder funcionar. Per requisits del client, el SGDB ha de ser Oracle amb els següents requisits mínims:

- Un SO Windows[10][12] o Linux[11][12]
- 8.5 Gb d'espai lliure[12].
- 2Gb de RAM[12]

El servidor, en aquest cas, serà un ordinador personal (PC) que compleix amb els requisits del programari. Compta amb 8 Gb de RAM i més de 10 Gb de disc lliure. Per fer aquest treball no es necessita un servidor perquè tant les connexions de clients com la càrrega de treball seran mínimes.

El SO pot ser Windows o Linux (64 o 32 bits). En el cas de Windows, les versions compatibles són:

- Windows 10 64 bits.
- Windows server 2012 R2 64 bits.
- Windows server 2016 R2 64 bits.
- Windows server 2019 R2 64 bits.

En el cas de Linux, les versions compatibles són:

En aquest cas concret, per aquest treball optem per la versió de Windows, però en cas que fos un servidor, la millor opció seria un sistema Linux, ja que la versió en Windows necessita una llicència d'ús. Una versió de Linux amb Debian o Ubuntu, seria gratuïta.

El SGDB ha de ser Oracle per requisits del client.

El client per la connexió a la BD serà *sqldeveloper*. És un client gratuït d'Oracle que es pot descarregar des de la seva pàgina web.

### 3.5 Disseny conceptual

En aquesta primera fase es vol dissenyar les principals entitats i les seves relacions per establir la base del producte[14].

Existeixen els següents requisits que no estan reflectits al disseny ER:

- El CIC ha de fer una primera valoració i acceptar la inclusió de la companyia a la base de dades perquè pugui penjar ofertes de treball.
- Les ofertes de treball enviades per les empreses les ha de publicar la universitat aproximadament uns tres mesos abans de l'inici de les pràctiques.

[10] <https://www.microsoft.com/es-es/windows>

[11] <https://www.linux.com/what-is-linux/>

[12] <https://www.oracle.com/es/database/technologies/appdev/xe.html>

[14] Llibre: CASAS ROMA, Jordi. Disseny conceptual de base de dades. UOC - PID\_00220512





caràcter, i un number(1) correspon amb un número compres entre el 0 i el 9, he optat pel number per la facilitat que es té en veure si la dada es verdadera o falsa gràcies als llenguatges de programació, on el 0 correspon amb fals. A més, les cerques per aquesta columna són més eficients per un número que per un caràcter.

- Un treballador només pot pertànyer a un departament de cada empresa. És així perquè normalment un treballador pertany a un departament, ja sigui direcció, RRHH... En cas contrari s'hauria de crear una taula que tingui la relació entre treballador i departament.
- Un treballador de qualsevol departament pot tenir el permís per validar i seleccionar qualsevol candidat. Per fer-ho s'ha creat un paràmetre anomenat *permís* a la taula treballador que té el següent funcionament:
  - És un valor numèric
  - Pot prendre els valors 0 (només lectura), 1 (pot validar candidatures) i 2 (pot seleccionar candidatures).
  - Funciona fent operacions AND i OR lògiques[15] per obtenir el permís que es necessita. Per exemple, un treballador amb permís 0 només pot veure candidatures, però no pot validar ni seleccionar. Un treballador amb permís 3 ( $2+1$ ) pot validar i seleccionar candidatures.
  - Es ampliable sempre que els següents valors coincideixin amb una potència de 2, és a dir,  $2^n$  (0,1,2,4,8...). És així per poder fer les operacions AND i OR. Per exemple, si volem crear un nou permís amb el número 4 que pugui crear programes de formació, un treballador amb permís 6 podrà crear programes de formació
  - ( $6 \text{ AND } 4 = 4$ ) i seleccionar candidatures ( $6 \text{ AND } 2 = 2$ ), però no podrà validar-ne cap ( $6 \text{ AND } 1 = 0$ ).
- Un estudiant només pot cursar una titulació. Si un estudiant pogués matricular-se de més d'una titulació, s'hauria de fer igual que amb els treballadors i departaments. En una futura revisió, es podria separar els estudiants per matrícula i titulació, de forma que si un estudiant es matricula de diverses titulacions, quedi enregistrat sense repetir informació.
- No s'ha creat una entitat *ProgramaFormacio* perquè es considera que no és necessari la creació d'una entitat. Els atributs d'aquesta entitat podrien ser *id*, *treballador\_nif* i *contracte\_practiques\_id*, dades que ja es tenen a l'entitat *ContractePractiques*.

[15] Llibre: HUERTAS SÁNCHEZ, M. Antònia. Lògica i àlgebra de boole. UOC - PID\_00149503

- La informació de l'informe final de pràctiques s'ha deixat en *ContractePractiques* perquè es considera que no és necessari la creació d'una entitat. Tota la informació pertany a l'entitat *ContractePractiques* que és la que rep l'avaluació final.
- Es podria haver creat una entitat *Persona* amb els atributs *id*, *nom* i *cognom*, i fer que les entitats *Treballador*, *Estudiant* i *Professor* siguin una especialització d'aquesta, però per raons de seguretat no s'ha fet, ja que són dades personals sensibles, sobretot si en un futur, per exemple, es decideix posar el número de seguretat social.
- El nom de la columna que correspon amb una clau forana està composta pel nom de la taula de la clau forana seguida d'un guió baix i el nom de la columna que correspon amb la clau primària. Per exemple *companyia\_cif*.
- Els noms dels procediments comencen per *afegir* quan es vol fer afegir una dada a la base de dades, *eliminar* quan es vol eliminar i *modificar* quan es vol editar.
- A conseqüència de la restricció dels noms tant de les claus primàries com les foranies (i qualsevol alta), ja que els noms són comuns a tota la base de dades i no es poden repetir, s'ha optat per posar com a nom la següent regla:
  - Per les claus primàries, nom de la taula seguit de PK.
  - Per les claus foranies, nom de la taula seguit pel nom de la taula que conté la clau seguida de FK
- Quan s'elimina una dada, com a conseqüència de la integritat referencial, també s'eliminen les dades associades. Es fa així per poder mantenir la integritat de les dades al sistema.

```

create table professor(
  nif char(9) not null,
  universitat_id INTEGER not null,
  nom varchar(50) not null,
  cognoms varchar2(255) not null,
  CONSTRAINT professorPK PRIMARY KEY (nif),
  CONSTRAINT professorUniversitatFK FOREIGN KEY (universitat_id) REFERENCES universitat (id) ON DELETE CASCADE ENABLE
);

```

Figura 6: SQL de taula amb integritat referencial d'eliminació en cascada

### 3.6 Disseny lògic

En aquest apartat es defineix com serà la creació del disseny lògic[16] a partir del disseny conceptual. Com que es tracta d'un projecte on un dels requisits és que s'ha de fer en una BD relacional, el disseny lògic també serà relacional. Per definir les relacions entre les diferents entitats seguirem el següent esquema:

[16] Llibre: BURGUES ILLA, Xavier. Disseny lògic de base de dades. UOC - PID\_00220510

- Els atributs que pertanyen a claus primàries (PK) estan en format subratllat
- Els atributs NOT NULL estan en format **negreta**
- Els atributs que pertanyen a claus foranes (FK) estan en format subratllat amb punts

Existeixen les següents entitats:

- Universitat( id, **nom**, **te\_estudiants** )
- Titulacio( id, **nom** )
- TitulacioUniversitat(universitat\_id, titulacio\_id )
  - universitat\_id es clau forana de Universitat (id)
  - titulació\_id es clau forana de Titulacio (id)
- Professor( nif , universitat\_id, **nom**, **cognoms** )
  - universitat\_id es clau forana de Universitat(id)
- Estudiant( nif, universitat\_id, titulacio\_id, **nom**, **cognoms** )
  - universitat\_id es clau forana de Universitat (id)
  - titulació\_id es clau forana de Titulacio (id)
- Companyia(cif, **nom**, **data\_peticio\_inscripcio** )
- Departament(id, **nom**)
- Treballador(nif, companyia\_cif, departament\_id, **permis**, **nom**, **cognoms**)
  - companyia\_cif es clau forana de Companyia (cif)
  - departament\_id es clau forana de Departament (id)
- OfertaTreball(id, universitat\_id, companyia\_cif, titulacio\_id, **nom**, **descripció**, **data\_creacio**)
  - universitat\_id es clau forana de Universitat (id)
  - companyia\_cif es clau forana de Companyia (cif)
  - titulació\_id es clau forana de Titulacio (id)
- Candidat(id, estudiant\_nif, oferta\_treball\_id, **requisits\_empresa**, **data\_inscripcio**)
  - estudiant\_nif es clau forana de Estudiant (nif)
  - oferta\_treball\_id es clau forana de OfertaTreball (id)
- PreguntaEntrevista(id, candidat\_id, **pregunta**, **data\_pregunta**, **resposta**, **data\_resposta** )
  - candidat\_id es clau forana de Candidat (id)

- Contracte( id, oferta\_treball\_id, treballador\_nif, estudiant\_nif, professor\_nif, data\_inici, data\_fi, sou, comentari\_tutor, es\_apte, data\_informe, comentari\_alumne)
  - oferta\_treball\_id es clau forana de OfertaTreball (id)
  - treballador\_nif es clau forana de Treballador (nif)
  - estudiant\_nif es clau forana de Estudiant (nif)
  - professor\_nif es clau forana de Professor (nif)
- TemaFormacio( id, contracte\_practiques\_id, titol\_tema, descripció, valoració, data\_valoracio)
  - contracte\_practiques\_id es clau forana de Contracte (id)
- InformeParcial( id, contracte\_practiques\_id, comentari\_tutor, data\_informe)
  - contracte\_practiques\_id es clau forana de Contracte (id)
- InspeccioTreball(id, contracte\_practiques\_id , es\_favorable , data\_inspecció)
  - contracte\_practiques\_id es clau forana de Contracte (id)

### 3.6 Normalització

Existeixen certes normes[16] per la creació de les BD. Aquestes normes indiquen com separar la informació de les dades a les taules i evitar-ne la redundància. També indica bones pràctiques que permeten evitar la pèrdua d'informació o evitar múltiples actualitzacions de les dades per un mal disseny. Aquestes normes estan fixades a la teoria de la normalització, i es distribueixen en nivells anomenats formes normals. Com més formes normals compleix l'estructura de les dades, menys redundància de dades i més optimització. Per complir una forma normal s'ha de complir l'anterior, és a dir, per complir la forma normal 2 també s'ha de complir la forma normal 1.

#### 3.6.1 Primera Forma Normal (1FN)

La primera forma normal especifica que tots els valors han de ser atòmics, es a dir, no poden haver-hi més d'un valor per registre (clau primària) ni valors null.

Per exemple, si a la taula *universitat* existís una columna anomenada titulació amb dos valors, no estaria en 1FN.

Es compleix la 1FN

[16] Llibre: BURGUES ILLA, Xavier. Disseny lògic de base de dades. UOC - PID\_00220510

### 3.6.2 Segona Forma Normal (2FN)

La segona forma normal especifica que tots els atributs que no són clau candidata, depenen exclusivament de la clau primària (PK).

Per exemple, els atributs de la taula *treballador* depenen exclusivament de la PK *nif* (del treballador). Aquesta taula no estaria en 2FN si un treballador pogués estar en més d'una companyia o en més d'un departament.

Es compleix la 2FN

### 3.6.3 Tercera Forma Normal (3FN)

La tercera forma normal especifica que tots els atributs que no són clau candidata, depenen de la PK. És a dir, que tots els atributs que no són clau primària (PRIMARY KEY), úniques (UNIQUE) o foranies (FOREIGN KEY) depenen de la clau primària i només dona informació referent a la PK.

Per exemple, si la taula *treballador* tingues com a clau primària composta per *nif* i *companyia\_cif*, no estaria en 3FN perquè permís depèn de *nif* però no de companyia. En una altra companyia podria tenir un altre permís.

Es compleix la 3FN

Una extensió d'aquesta forma normal és la forma normal de Boyce-Codd (FNBC). La diferència respecte a la 3FN està en el fet que també ha de complir que els determinants, que són atributs que depenen d'una clau candidata, depenen únicament d'aquella clau candidata. És a dir, cap dels determinants es repeteix amb la mateixa clau candidata de la taula. Normalment, es dona en taules on la clau primària es composta.

Es compleix la FNBC

### 3.6.4 Quarta Forma Normal(4FN)

La quarta forma normal especifica que els fets multivaluats, que són les dades relacionades entre múltiples taules, no estigui repetida en diferents files. Això fa que per un concepte s'hagin d'afegir tantes línies a la taula com dades repetides hi existeixen.

Per exemple, si els estudiants no tinguessin la relació amb la titulació, i ho tinguessin a la mateixa taula com un nom, no podria complir amb la 4FN, ja que s'hauria de repetir cada línia (registre) de la taula per cada titulació a la qual hagués fet la matrícula. A més, en cas que de canvi del nom de la titulació, s'hauria d'actualitzar tantes files com estudiants amb aquesta titulació.

Es compleix la 4FN

### 3.6.5 Quinta Forma Normal (1FN)

La quinta forma normal especifica que no hi ha d'haver dependències de projecció-combinació. Les dependències de projecció-combinació es donen quan una taula pot repetir conceptes per la mateixa línia, és a dir, hi ha atributs que es poden repetir per la mateixa clau candidata. Normalment es dona en taules amb clau composta.

Es compleix la 5FN

### 3.7 Disseny físic

El disseny físic[17] consisteix en la creació en el SGBD de les taules, procediments i funcions del disseny conceptual i lògic.

El primer pas és la creació d'una base de dades. Per fer-ho, obrim el sqldeveloper i creem una nova BD i una nova connexió per aquesta BD. Les BD es creen amb un usuari i una contrasenya. Per poder utilitzar la BD s'ha d'utilitzar una connexió, que és un arxiu de configuració on s'especifica totes les dades necessàries per a la comunicació amb el SGBD, com per exemple usuari, contrasenya, port, nom de la base de dades...

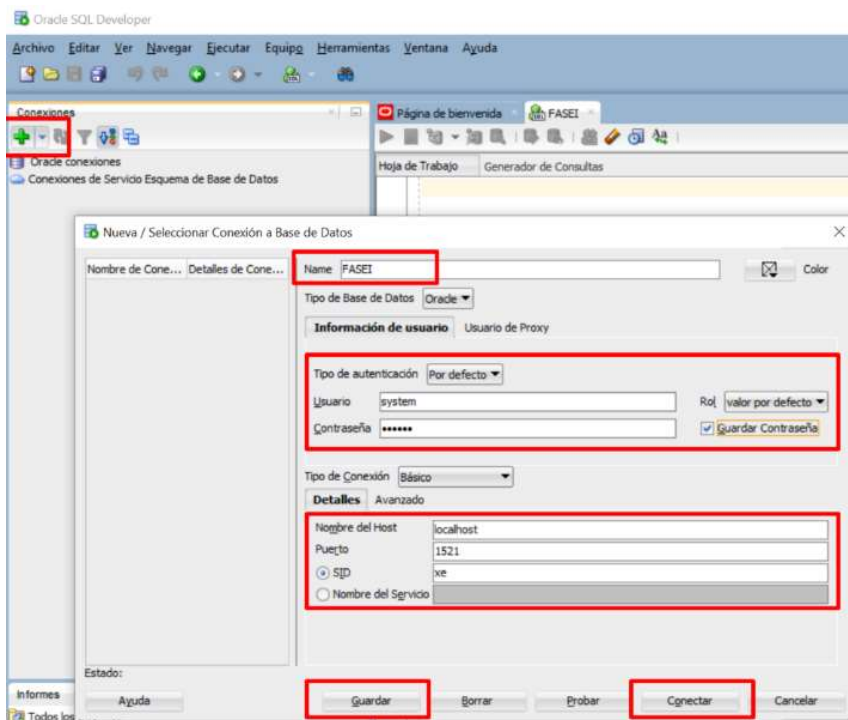


Figura 7: Creació d'una BD amb sqldeveloper d'Oracle

Un cop creada, podem accedir-hi per començar a crear les taules, els procediments i les funcions necessàries.

[17] Llibre: CABRÉ I SEGARRA, Blai. Disseny físic de base de dades. UOC - PID\_00223662

Per crear les taules tenim diverses opcions. Les més comunes són dos: o bé mitjançant el GUI del programa, o mitjançant llenguatge SQL. Per fer-ho mitjançant el GUI, anem a l'opció *Taules*, premem el *bot dret del ratolí* i seleccionem *nova taula*. Un cop obert afegim les dades necessàries.

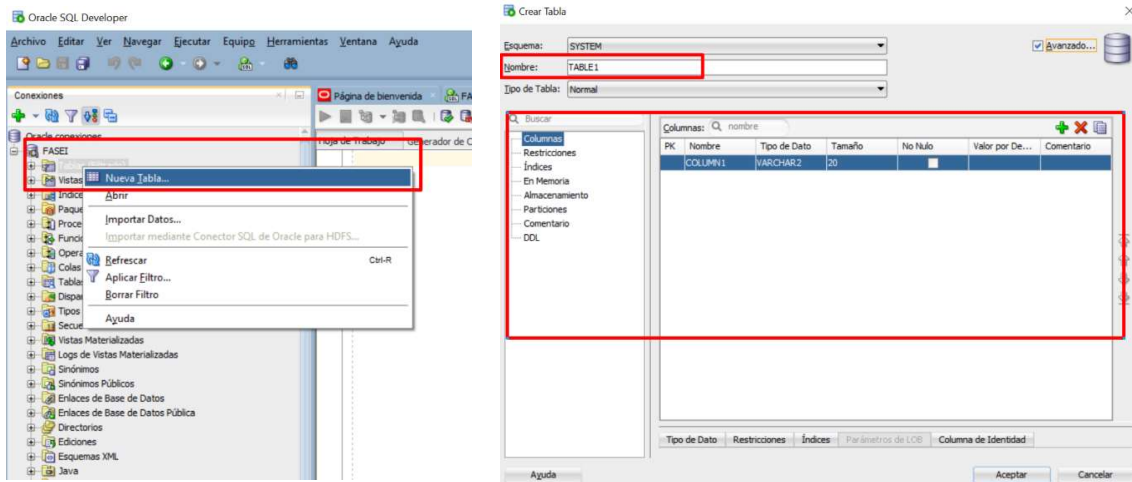


Figura 8: Creació d'una taula amb sqldeveloper d'Oracle mitjançant GUI

Per fer-ho mitjançant codi SQL, hem d'obrir un arxiu per poder executar codi. Un cop obert posem el codi i l'executem

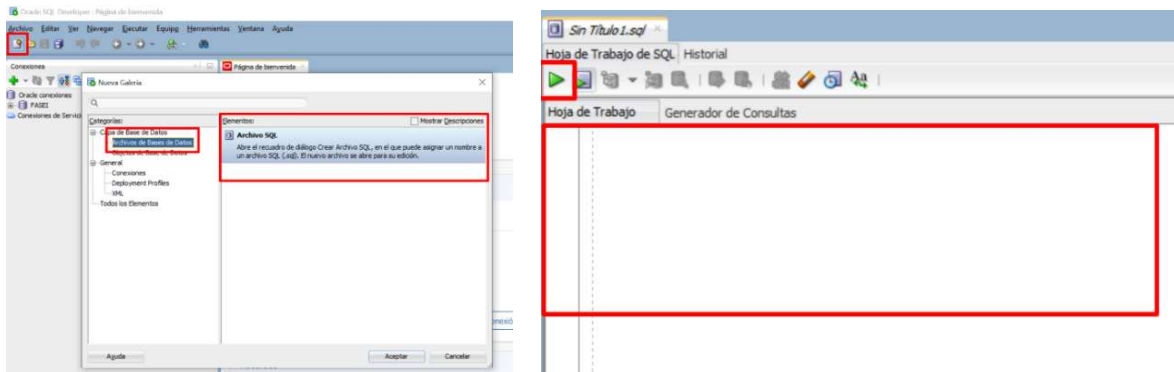


Figura 9: Creació d'un arxiu de comandos SQL amb sqldeveloper d'Oracle



Mitjançant l'arbre situat a l'esquerra del programa sqldeveloper, es pot veure la llista de les taules i procediments creats.

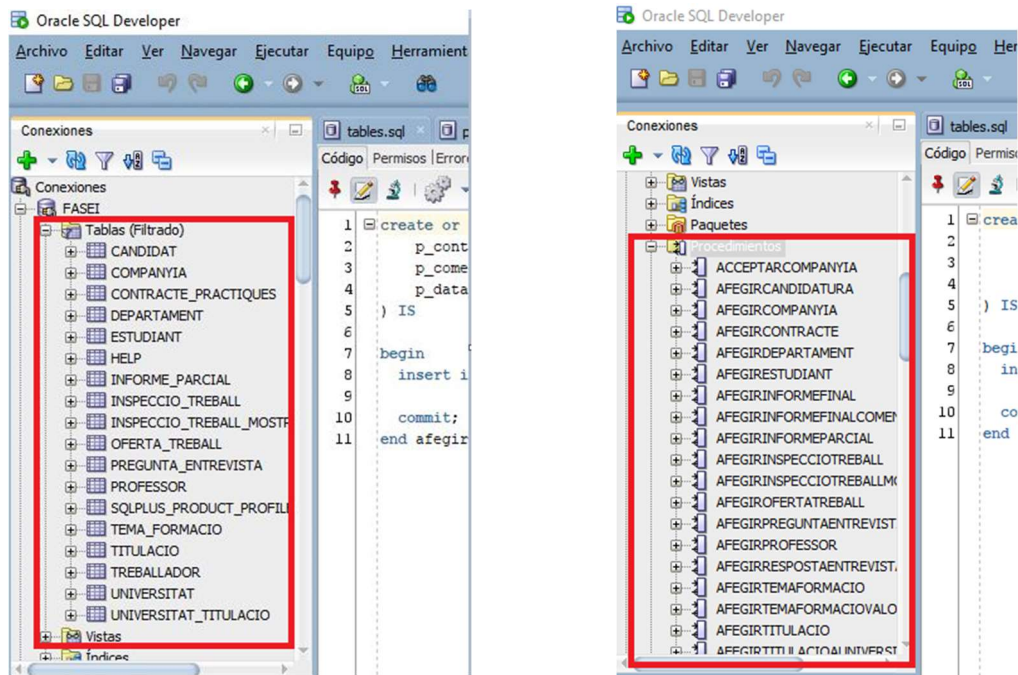


Figura 10: Llista de l'arbre de taules i procediments a l'sqldeveloper

### 3.7.1 PL/SQL

PL/SQL és un llenguatge de programació per procediments que complementa l'ús del llenguatge SQL. Mentre que amb l'SQL es poden executar sentències per interactuar amb la base de dades i consultar, actualitzar o eliminar dades, taules..., amb el PL/SQL es pot executar diversos blocs de codi, incloent-hi codi SQL, per obtenir un resultat concret.

PL/SQL també permet la creació de variables, constants, cursors... que permeten múltiples operacions per obtenir el resultat desitjat. A més de tot això, també es poden identificar els possibles errors de dades que hi hagi, assegurant-ne la integritat de les dades, i donar un missatge personalitzat per poder resoldre'l.

Un exemple seria la creació d'una universitat amb els diversos títols acadèmics que imparteix. Mitjançant PL/SQL es podria crear un procediment amb els paràmetres d'entrada del nom d'universitat i una llista dels títols acadèmics. Aquest procediment asseguraria que la universitat no existeix, els títols acadèmics existeixen i ho pot donar tot d'alta. Sense aquest llenguatge, el procés s'hauria de fer de forma manual mitjançant llenguatge SQL i en diversos passos.

Per la creació dels procediments es fa servir PL/SQL.

### 3.7.2 Creació de procediments

Per crear procediments mitjançant el GUI anem al sqldeveloper, i a la columna esquerra mitjançant el botó esquerre del ratolí, seleccionem “Nou Procediment”. S’obre una finestra per poder posar el nom del procediment i el nombre de paràmetres d’entrada i/o sortida (paràmetre Mode IN/OUT).

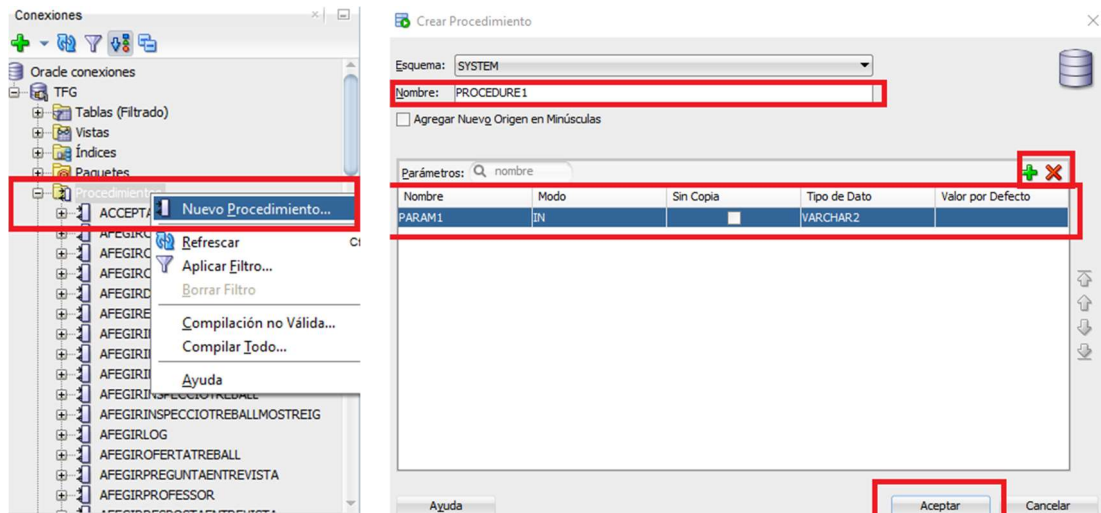


Figura 11: Creació de procediments mitjançant GUI a l’sqldeveloper

Per crear els procediments mitjançant codi SQL, s’ha d’obrir un nou arxiu de codi i escriure el codi SQL com s’indica a la documentació d’oracle[18][19].

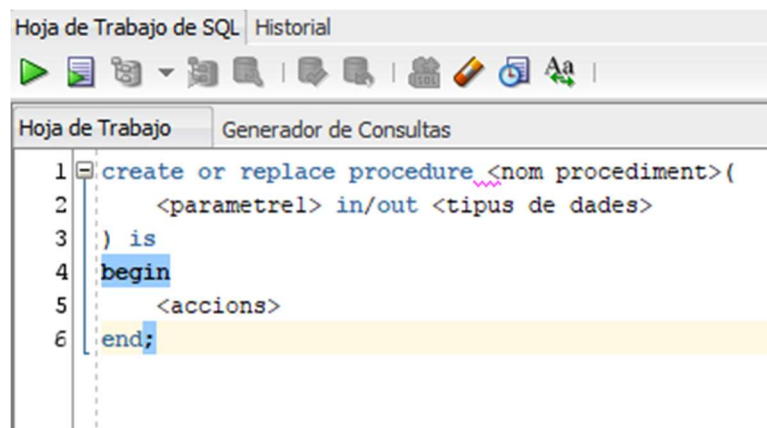


Figura 12: Exemple de creació de procediments mitjançant codi SQL a l’sqldeveloper

Els procediments creats són de tres tipus:

- **Afegir**: corresponen amb els procediments que afegixen dades al sistema. Tots comencen per la paraula afegir seguida del nom de l’entitat, com per exemple *afegirUniversitat*.

[18] <https://docs.oracle.com/en/database/oracle/oracle-database/21/Inpls/CREATE-PROCEDURE-statement.html#GUID-5F84DB47-B5BE-4292-848F-756BF365EC54>

[19] <https://www.oracletutorial.com/plsql-tutorial/plsql-procedure>

- **Modificar:** corresponen amb els procediments que modifiquen les dades entrades al sistema. Tots comencen per la paraula modificar seguida del nom de l'entitat, com per exemple *modificarUniversitat*
- **Eliminar:** corresponen amb els procediments que eliminen les dades entrades al sistema. Tots comencen per la paraula eliminar seguida del nom de l'entitat, com per exemple *eliminarUniversitat*.

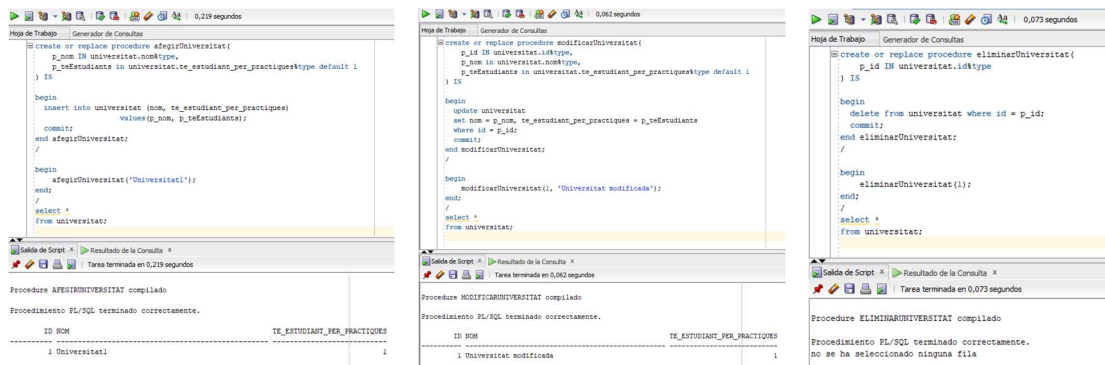


Figura 13: Procediments d'afegir, modificar i eliminar de la Fase I

### 3.8 Joc de dades

El joc de dades és un joc complet amb dades per totes les entitats i també les seves relacions. La importació de les dades es fa mitjançant els procediments creats en la base de dades. Encara que és un joc complet, té algunes incongruències, és a dir, és un joc que permet comprovar les relacions entre les diferents entitats mitjançant les PK i les FK, però algunes dades com per exemple les dates entre la inscripció de l'alumne a l'oferta de treball i la publicació de l'oferta de treball per part de la universitat, poden no coincidir.

La creació de les taules es troba a [taules.sql](#) dins l'arxiu fase1.zip.

La creació dels procediments es troba a [procediments.sql](#) dins l'arxiu fase1.zip.

El joc de dades es troba a [dades.sql](#) dins l'arxiu fase1.zip.

### 3.9 Proves i optimitzacions

En aquesta primera fase, en les proves i optimitzacions es vol comprovar que les relacions entre les entitats és correcta. Per aquest motiu, les consultes són bàsiques i permeten veure la informació que, per altra banda, podria correspondre amb les diferents opcions de l'aplicatiu a alt nivell, com per exemple els estudiants d'una universitat amb la seva titulació, el llistat de contractes en pràctiques que hi ha, etc.

Les consultes es troben a [consultes.sql](#) dins l'arxiu fase1.zip.

## 4. Fase II

### 4.1 Introducció

En aquesta segona fase del treball es vol ampliar l'estructura creada a la fase I per afegir un mòdul estadístic i un registre d'error aplicant tècniques de magatzem de dades (DataWarehouse[31,32,33,34,35,36]).

La finalitat és aconseguir una versió beta del producte per poder avaluar-lo i arreglar els possibles problemes que es puguin detectar abans de fer l'entrega final. A més, també permet provar si les modificacions i actualitzacions són senzilles i ràpides d'implementar al sistema.

El registre d'error ha de ser capaç d'emmagatzemar la crida del procediment amb totes les dades enviades, la data, el nom del procediment i el resultat de l'operació. En cas d'error, també ha de poder emmagatzemar el codi d'error.

El mòdul estadístic ha de ser capaç d'emmagatzemar les dades necessàries per complir amb el requisit de donar resultats en temps constant 1 de forma automàtica i transparent per l'usuari que fa servir el producte.

### 4.2 Anàlisi de requisits.

S'han detectat els següents requisits que s'han d'afegir als detectats a la fase I:

Codi	Descripció
<b>R011</b>	El mòdul estadístic ha de donar el resultat en temps constant 1
<b>R012</b>	Els procediments tenen que d'incorporar un paràmetre de sortida anomenat RSP amb les següents condicions: <ul style="list-style-type: none"><li>• Si l'execució del procediment ha finalitzat amb èxit, retornarà 'OK'.</li><li>• Si ha donat algun error, retornarà 'ERROR &lt;codi d'error&gt;'</li></ul>
<b>R013</b>	Els procediments han de disposar de tractament d'excepcions per recollir els errors
<b>R014</b>	Ha de poder emmagatzemar grans volums de dades
<b>R015</b>	Les actualitzacions i modificacions del sistema han de ser ràpides i senzilles

[31] <https://www.guru99.com/data-warehouse-architecture.html>

[32] [https://ca.wikipedia.org/wiki/Magatzem\\_de\\_dades](https://ca.wikipedia.org/wiki/Magatzem_de_dades)

[33] <https://insightsoftware.com/es/blog/database-vs-data-warehouse-whats-the-difference/>

[34] <https://www.inesem.es/revistadigital/informatica-y-tics/guia-construir-datawarehouse>

[35] <http://fcea.unicauca.edu.co/old/datawarehouse.htm>

[36] <https://www.ionos.es/digitalguide/online-marketing/analisis-web/los-data-warehouses-en-la-business-intelligence/>

### 4.3 Criteris d'acceptació

S'han de complir aquests criteris per poder avaluar els resultats del treball

Codi	Descripció
<b>C001</b>	El repositori estadístic ha de donar resultats en temps constant 1
<b>C002</b>	Les modificacions i millores del producte són fàcils d'implementar

### 4.4 Disseny Conceptual

Per al disseny de les entitats noves, s'ha de diferenciar en dos tipus:

- Les que no tenen relació directa amb les entitats i les dades ja creades, que correspon amb una actualització del sistema per implementar el registre d'accions
- Les que tenen relació directa amb les entitats i les dades ja creades, que correspon amb una millora del sistema per implementar el mòdul estadístic.

#### 4.4.1 Entitats del registre d'accions

Les entitats pel registre d'errors són dues, el llistat de les accions disponibles, i les dades corresponents a la crida del procediment, la data i el resultat de la crida.

El l·listat d'accions disponibles emmagatzema els noms dels procediments de les taules. El procés és automàtic, ja que s'agafa el nom del procediment amb el codi.

Les dades corresponents a la crida registra a la base de dades totes les dades relacionades amb la crida als procediments. Les dades són:

- Paràmetres d'entrada
- Data i hora de la crida
- Nom del procediment cridat
- Resultat de l'execució. En cas d'error també emmagatzema l'error que ha donat

El model ER és el següent:

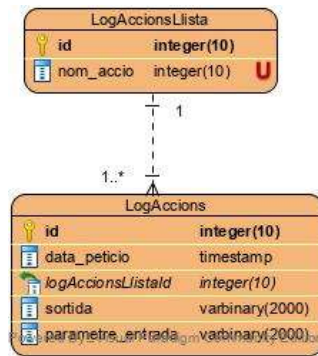


Figura 14: Model ER del registre d'accions

#### 4.4.2 Entitats del mòdul estadístic

Les entitats del mòdul estadístic comprenen totes aquelles entitats que són necessàries per poder fer les consultes amb temps constant 1. Aquestes taules van recollint les dades de forma automàtica mitjançant disparadors (*triggers en anglès [20]*).

El model ER del mòdul estadístic és el següent:

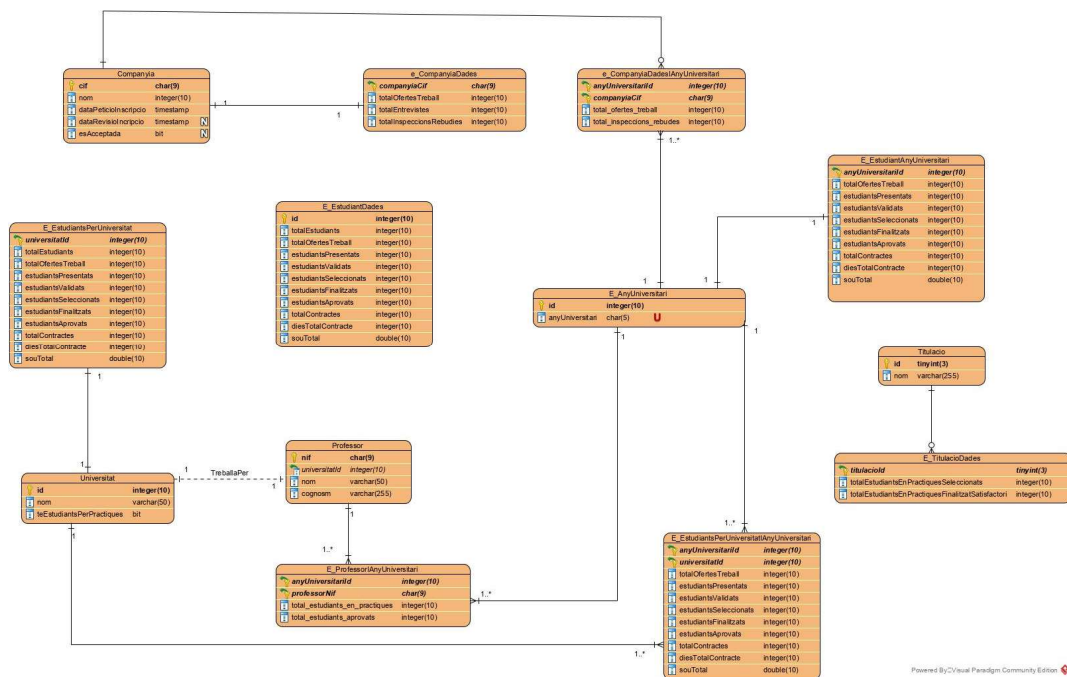


Figura 15: Model ER del mòdul estadístic

[20] <https://docs.oracle.com/en/database/oracle/oracle-database/21/npls/plsql-triggers.html#GUID-217E8B13-29EF-45F3-8D0F-2384F9F1D231>





- La columna *parametre\_entrada* de la taula *log\_accions* és del tipus `varchar2`, però les dades estan en format `JSON[21][22][26]`. Això permet fer cerques dins de les dades per poder trobar una dada concreta mitjançant una sentència `SELECT`.
- S'ha creat una taula anomenada *e\_any\_universitari* que permet emmagatzemar de forma automàtica els anys universitaris per les dades estadístiques. Es considera any universitari el període comprès entre setembre i juliol, de forma que la data 26/10/2021 correspon a l'any universitari 21/22, mentre que la data 15/03/2021 correspon amb l'any universitari 20/21. El format de l'any universitari és `<any1>/<any2>`.
- Les claus foranes de les taules d'estadística es fan de forma lògica, ja que es trenca la integritat referencial de les taules quan es fan eliminacions de taula d'entitats principals, com podria ser esborrar totes les dades de la taula universitat. Per aquest motiu no s'han afegit al disseny físic de les taules, i per complir amb la integritat referencial, s'han creat disparadors que eliminen la dada de les taules d'estadística per complir amb la integritat referencial.
- S'han creat alguns índexs en taules per millorar-ne la velocitat en les consultes a les dades. Els índexs creats són d'una sola dada, com podrien ser les claus primàries (`id`, `nif...`) o compostes, com podrien ser claus compostes (`any_universitari_id` - `universitat_id`). Quan es crea un índex s'ha de tenir en compte que no poden haver-hi dos noms d'índexs iguals, per això s'ha seguit la següent nomenclatura per crear-los:
  - Nom de l'entitat seguit de la paraula clau índex seguit d'un comptador. Per exemple, `candidat_index_1`

#### 4.5 Disseny lògic

El disseny lògic de les entitats del registre d'accions i del mòdul estadístic segueix el mateix patró que a la fase I.

S'ha de tenir en compte, que les relacions foranes entre les taules del mòdul estadístic i les de les entitats que no corresponent amb el mòdul estadístic, es fa de forma lògica i no física, és a dir, es fa mitjançant codi `PS/SQL` i no amb la creació de la taula mitjançant `SQL`.



#### 4.5.1 Registre d'accions

La implementació d'aquest mòdul permet comprovar si les actualitzacions del sistema són senzilles.

El disseny lògic del registre d'errors de les entitats és el següent:

- LogAccionsLLista( **id**, **nom\_accio**)
  - nom\_accio: correspon amb el nom del procediment cridat.
- LogAccions( **id**, **data\_peticio**, **log\_accio\_llista\_id**, **sortida**, **parametre\_entrada** )
  - data\_peticio: correspon amb la data de quan es fa la petició al procediment.
  - log\_accio\_llista\_id: identificador que correspon amb el nom del procediment cridat
  - sortida: correspon amb el resultat de l'execució del procediment. El resultat és OK per a les execucions amb èxit, i ERROR|<missatge d'error> en cas que s'hagi produït un error.
  - parametre\_d'entrada: correspon amb tots els paràmetres d'entrada del procediment quan es fa la crida. L'emmagatzematge de les dades es fa mitjançant JSON.

#### 4.5.2 Mòdul estadístic

La implementació d'aquest mòdul permet comprovar si les modificacions del sistema són senzilles.

El disseny lògic del mòdul estadístic de les entitats és el següent:

- E\_AnyUniversitari( **id**, **any\_universitari**)
- ECompanyiaDades( **companyia\_cif**, **total\_ofertes\_treball**, **total\_entrevistes**, **total\_inspeccions\_rebudes**)
  - total\_ofertes\_treball: Total d'ofertes de treball creades per companyia
  - total\_entrevistes: total entrevistes fetes a candidats. S'entén per entrevista un cop la universitat ha enviat la candidatura a la companyia i aquesta l'ha validat.
- ECompanyiaDadeslAnyUniversitari( **companyia\_cif**, **any\_universitari\_id**, **total\_ofertes\_treball**, **total\_inspeccions\_rebudes**)
  - total\_ofertes\_treball: Total d'ofertes de treball creades per companyia i any universitari
  - total\_inspeccions\_rebudes: Total d'inspeccions rebudes per companyia i any universitari.

- **E\_EstudiantsAnyUniversitari( any\_universitari\_id, total\_ofertes\_treball, estudiants\_presentats, estudiants\_validats, estudiants\_seleccionats, estudiants\_finalitzats, estudiants\_aprovats, total\_contractes, dies\_total\_contracte, sou\_total)**
  - total\_ofertes\_treball: total d'ofertes de treball creades per les companyies per any universitari
  - estudiants\_presentats: total de candidatures presentades pels estudiants per any universitari
  - estudiants\_validats: total d'estudiants validats per les empreses per any universitari
  - estudiants\_seleccionats: total d'estudiants seleccionats per les empreses per signar un contracte de pràctiques per any universitari.
  - estudiants\_finalitzats: total d'estudiants en pràctiques que han finalitzat el contracte en pràctiques per any universitari.
  - estudiants\_aprovats: total d'estudiants en pràctiques que han finalitzat i aprovat les pràctiques d'empresa per any universitari.
  - total\_contractes: total contractes signats per any universitari. S'ha de tenir en compte que pot haver-hi un estudiant seleccionat que no hagi signat el contracte per qualsevol motiu
  - dies\_total\_contracte: total de dies de durada de cada contracte. Correspon amb la suma dels dies de durada de cada contracte per any universitari.
  - sou\_total: sou total dels contractes signats per any universitari. Correspon amb la suma de tots els sous.
  
- **E\_EstudiantsDades( id, total\_estudiants, total\_ofertes\_treball, estudiants\_presentats, estudiants\_validats, estudiants\_seleccionats, estudiants\_finalitzats, estudiants\_aprovats, total\_contractes, dies\_total\_contracte, sou\_total)**
  - total\_estudiants: Total d'estudiants a la base de dades.
  - total\_ofertes\_treball: total d'ofertes de treball a la base de dades
  - estudiants\_presentats: total de candidatures presentades pels estudiants a la base de dades
  - estudiants\_validats: total d'estudiants validats per les empreses a la base de dades
  - estudiants\_seleccionats: total d'estudiants seleccionats per les empreses per signar un contracte de pràctiques a la base de dades.
  - estudiants\_finalitzats: total d'estudiants en pràctiques que han finalitzat el contracte en pràctiques a la base de dades.
  - estudiants\_aprovats: total d'estudiants en pràctiques que han finalitzat i aprovat les pràctiques d'empresa a la base de dades.
  - total\_contractes: total contractes signats per any universitària a la base de dades.
  - dies\_total\_contracte: total de dies de durada de cada contracte a la base de dades
  - sou\_total: sou total dels contractes signats per any universitari a la base de dades

- **E\_EstudiantsPerUniversitat( universitat\_id, , total\_estudiants, total\_ofertes\_treball, estudiants\_presentats, estudiants\_validats, estudiants\_seleccionats, estudiants\_finalitzats, estudiants\_aprovats, total\_contractes, dies\_total\_contracte, sou\_total)**
  - total\_estudiants: total d'estudiants per universitat
  - total\_ofertes\_treball: total d'ofertes de treball creades per les companyies per universitat
  - estudiants\_presentats: total de candidatures presentades pels estudiants per universitat
  - estudiants\_validats: total d'estudiants validats per les empreses per universitat
  - estudiants\_seleccionats: total d'estudiants seleccionats per les empreses per signar un contracte de pràctiques per universitat.
  - estudiants\_finalitzats: total d'estudiants en pràctiques que han finalitzat el contracte en pràctiques per universitat.
  - estudiants\_aprovats: total d'estudiants en pràctiques que han finalitzat i aprovat les pràctiques d'empresa per universitat.
  - total\_contractes: total contractes signats per universitat
  - dies\_total\_contracte: total de dies de durada de cada contracte per universitat.
  - sou\_total: sou total dels contractes signats per universitat.
  
- **E\_EstudiantsPerUniversitatIAnyUniversitari( any\_universitari\_id, universitat\_id, total\_ofertes\_treball, estudiants\_presentats, estudiants\_validats, estudiants\_seleccionats, estudiants\_finalitzats, estudiants\_aprovats, total\_contractes, dies\_total\_contracte, sou\_total)**
  - total\_ofertes\_treball: total d'ofertes de treball creades per les companyies per universitat i any universitari
  - estudiants\_presentats: total de candidatures presentades pels estudiants per universitat i any universitari
  - estudiants\_validats: total d'estudiants validats per les empreses per universitat i any universitari
  - estudiants\_seleccionats: total d'estudiants seleccionats per les empreses per signar un contracte de pràctiques per universitat i any universitari.
  - estudiants\_finalitzats: total d'estudiants en pràctiques que han finalitzat el contracte en pràctiques per universitat i any universitari.
  - estudiants\_aprovats: total d'estudiants en pràctiques que han finalitzat i aprovat les pràctiques d'empresa per universitat i any universitari.
  - total\_contractes: total contractes signats per universitat i any universitari
  - dies\_total\_contracte: total de dies de durada de cada contracte per universitat i any universitari.
  - sou\_total: sou total dels contractes signats per universitat i any universitari.

- **E\_ProfessorlAnyUniversitari**(any\_universitari\_id, professor\_nif, **total\_estudiants\_en\_practiques**, **total\_estudiants\_aprovats**)
  - total\_estudiants\_en\_practiques: total d'estudiants en pràctiques per any universitari i professor
  - total\_estudiants\_aprovats: total d'estudiants que han aprovat les pràctiques d'empresa per any universitari i professor.
- **ETitulacioDades**( titulacio\_id, **total\_estudiants\_en\_practiques\_seleccionats**, **total\_estudiants\_en\_practiques\_finalitzat\_satisfactori** )
  - total\_estudiants\_en\_practiques\_seleccionats:

#### 4.6 Disseny físic

En aquesta fase, a més de la creació de les taules necessàries per al funcionament del registre d'errors i les estadístiques, també s'han creat:

- Funcions: Necessàries per poder obtenir el resultat de dades necessàries per a l'execució de consultes, i també per poder tenir un registre d'errors exhaustiu.
- Disparadors: Necessaris per poder omplir de dades les taules d'estadística de forma transparent per l'usuari que utilitza el producte. També són necessaris per poder mantenir la integritat referencial de les taules d'estadística.
- Índex de taules: Necessaris per poder millorar el temps de resposta de les dades quan es fan actualitzacions, eliminacions i consultes.

Hi ha dues maneres fàcils per la creació de qualsevol dels elements citats: mitjançant el GUI o mitjançant codi SQL.

##### 4.6.1 Funcions

S'han creat un conjunt de funcions que permeten:

- Millorar la sortida d'errors del mòdul quan es fa una crida a un procediment donant informació en temps real quan s'executa una crida a un procediment del resultat de l'execució, i l'error amb el codi en cas que la crida no finalitza amb èxit.
- Millora la cohesió de les dades. Això és perquè la integritat referencial s'encarrega del fet que les dades existeixin, però no que siguin coherents. Amb les funcions ens assegurem que les dades que es volen emmagatzemar al sistema coincideixen al 100%. Un exemple seria quan una companyia publica una oferta de treball per una universitat Y. Aquesta universitat Y té els estudiants Y1, Y2 i Y3. Si un estudiant de la universitat X, anomenat X1 es vol apuntar a l'oferta, si només tenim en compte la integritat referencial, no donaria cap error i el deixaria apuntar perquè

l'estudiant X1 existeix encara que no estigui relacionat a la universitat Y. La funció comprova, a més, que l'estudiant X1 pertany a la universitat Y, sinó dona error.

PROCEDIMENT	RESULTAT
AFEGIRCONTRACTE	ERROR NO S'HA SELECCIONAT CAP CANDIDAT PER LA OFERTA DE TREBALL I

Figura 17: Sortida d'error per dada no coherent

- Ajuden en la comprovació de l'existència de dades.
- Mantenen l'ordre i la qualitat del codi, ja sigui en funcions, procediments, disparadors...

```

/**
Activa o desactiva el fet que una universitat té estudiants en practiques
@param p_id identificador de la universitat a modificar
@param RSP OUT retorna el resultat de l'execució del procediment
**/
create or replace procedure universitatTeEstudiantsPerPractiques(
    p_id IN universitat.id%type,
    p_teEstudiants in universitat.te_estudiant_per_practiques%type,
    RSP out varchar2
) IS
no_universitat exception;
no_valors_booleans exception;
params varchar2(3000);
begin
    -- Inicialitzem la variable RSP
    RSP := 'OK';
    -- Creem el JSON per al registre
    params := JSON_OBJECT('p_id' value p_id, 'p_teEstudiants' value p_teEstudiants);

    -- Cerquem si els valors del paràmetre teEstudiants es vàlid
    if (comprovaValorsBooleans(p_teEstudiants) = false) then
        raise no_valors_booleans;
    end if;

    -- Cerquem si existeix la universitat
    if (existeixUniversitat(p_id) is null) then
        raise no_universitat;
    end if;

    update universitat
    set te_estudiant_per_practiques = p_teEstudiants
    where id = p_id;
    commit;

    afegirLog($plsql_unit, params, RSP);

exception
when no_valors_booleans then
    RSP := 'ERROR|EL VALOR PER TE ESTUDIANTS NO ES VÀLID [0,1]';
    afegirLog($plsql_unit, params, RSP);

when no_universitat then
    RSP := 'ERROR|NO EXISTEIX LA UNIVERSITAT';
    afegirLog($plsql_unit, params, RSP);

when others then
    RSP := concat('ERROR|', SQLERRM);
    afegirLog($plsql_unit, params, RSP);
end universitatTeEstudiantsPerPractiques;

```

Figura 18: Vista d'un procediment amb funcions

#### 4.6.2 Creació de funcions

Per crear funcions mitjançant el GUI anem al sqldeveloper, i a la columna esquerra mitjançant el botó esquerre del ratolí, seleccionem “Nova funció”. S'obre una finestra per poder posar el nom de la funció, tipus de dades que ha de retornar i el nombre de paràmetres d'entrada i/o sortida (paràmetre Mode IN/OUT).

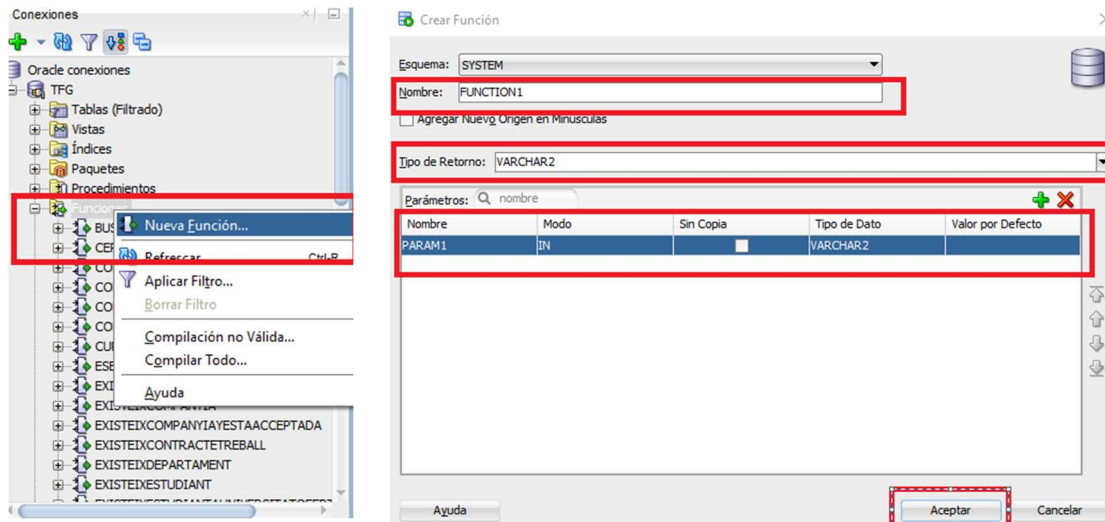


Figura 19: Creació d'una funció mitjançant el GUI de l'sqldeveloper

Per crear les funcions mitjançant codi SQL, s'ha d'obrir un nou arxiu de codi i escriure el codi SQL com s'indica a la documentació d'oracle[23].

```

1 create or replace function <nom funcio>(
2   <parametre1> in/out <tipus de dades>
3 ) return number as
4   <nom parametre retorn> <tipus de retorn>
5 begin
6   <accions>
7
8   return <nom parametre retorn>
9 end;

```

Figura 20: Exemple de codi per la creació d'una funció mitjançant codi SQL de l'sqldeveloper

#### 4.6.3 Creació de disparadors

Per crear funcions mitjançant el GUI anem al sqldeveloper, i a la columna esquerra mitjançant el botó esquerre del ratolí, seleccionem "Nova funció". S'obre una finestra per poder posar el nom de la funció, tipus de dades que ha de retornar i el nombre de paràmetres d'entrada i/o sortida (paràmetre Mode IN/OUT).

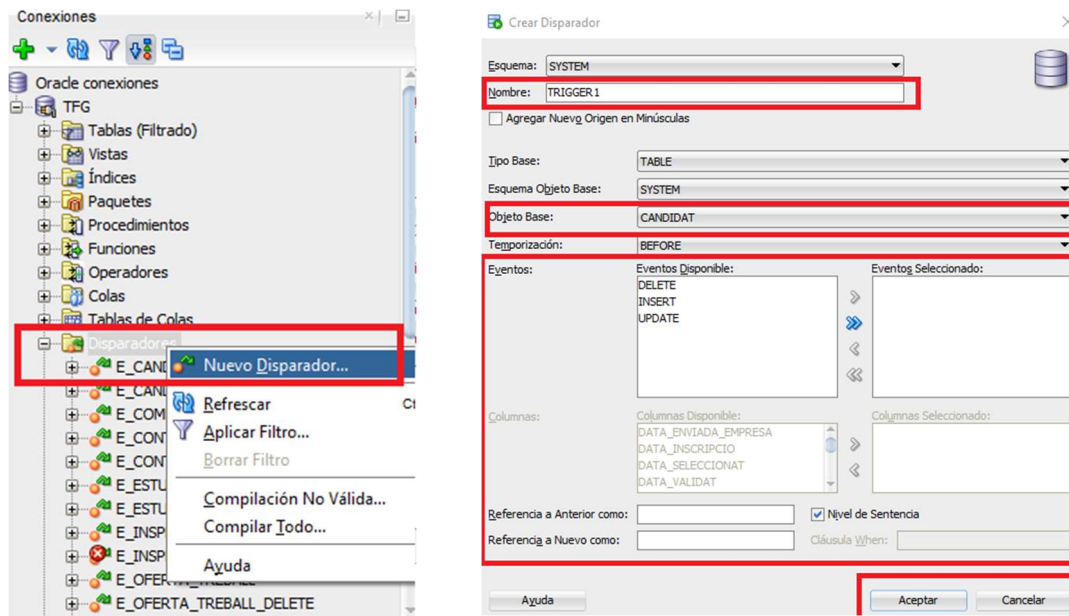


Figura 21: Creació d'un disparador mitjançant el GUI de l'sqldeveloper

Per crear els disparadors mitjançant codi SQL, s'ha d'obrir un nou arxiu de codi i escriure el codi SQL com s'indica a la documentació d'oracle[20][24].

#### 4.6.4 Creació d'índex

Per crear un índex mitjançant el GUI anem al sqldeveloper, i a la columna esquerra mitjançant el botó esquerre del ratolí, seleccionem "Nou índex". S'obre una finestra per poder posar el nom de la funció, tipus de dades que ha de retornar i el nombre de paràmetres d'entrada i/o sortida (paràmetre Mode IN/OUT).

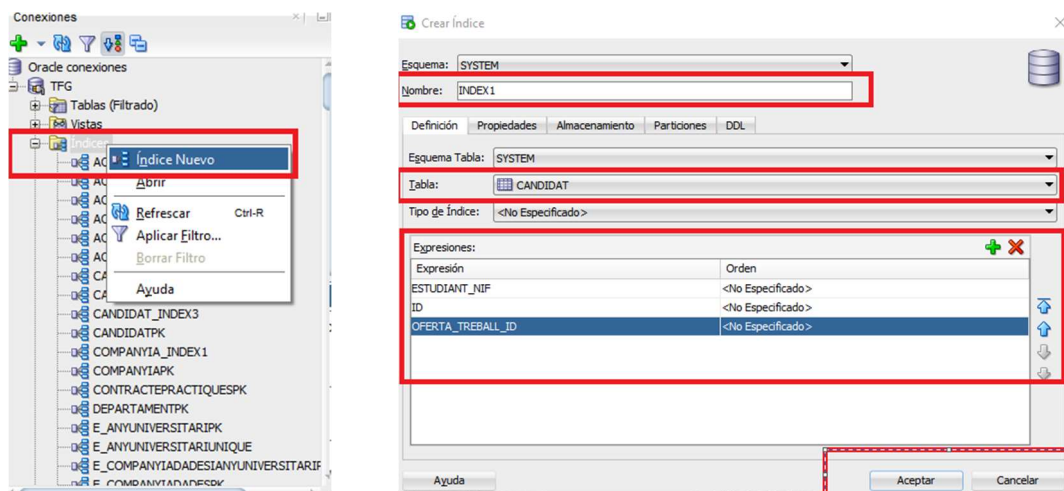


Figura 22: Creació d'un índex mitjançant el GUI de l'sqldeveloper

[20] <https://docs.oracle.com/en/database/oracle/oracle-database/21/npls/plsql-triggers.html#GUID-217E8B13-29EF-45F3-8D0F-2384F9F1D231>

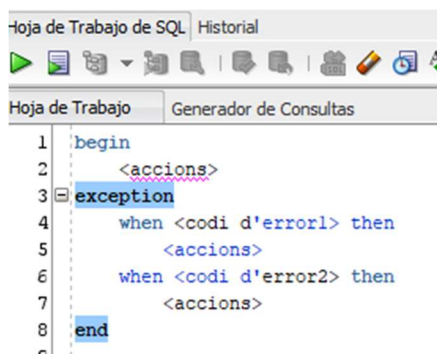
[23] <https://docs.oracle.com/en/database/oracle/oracle-database/21/npls/CREATE-FUNCTION-statement.html>

[24] <https://docs.oracle.com/en/database/oracle/oracle-database/21/npls/CREATE-TRIGGER-statement.html#GUID-AF9E33F1-64D1-4382-A6A4-EC33C36F237B>

Per crear els índexs mitjançant codi SQL, s'ha d'obrir un nou arxiu de codi i escriure el codi SQL com s'indica a la documentació d'oracle[25].

#### 4.6.5 Captura d'errors

Per poder capturar els errors que es poden produir al codi SQL, com per exemple la consulta a una taula sense dades dins un procediment, s'ha d'escriure el codi SQL com s'indica a la documentació d'oracle[27][28].



```
1 begin
2     <accions>
3 exception
4     when <codi d'error1> then
5         <accions>
6     when <codi d'error2> then
7         <accions>
8 end
```

Figura 23: Exemple de codi per la captura d'un error mitjançant codi SQL de l'sqldeveloper

De codis d'error n'hi ha que ja existeixen al sistema d'Oracle. També es poden crear de personalitzats per poder tenir més control sobre les dades

#### 4.7 Mòdul estadístic

El mòdul estadístic el compren un conjunt de dades, que separades per si mateixes poden no tenir gran contingut semàntic, però quan les juntes i les relacions donen un resultat que té sentit.

Les dades estadístiques són dades que es van calculant de forma dinàmica i automàtica, de tal forma que l'usuari que utilitza el producte no té coneixement de com funciona.

##### 4.7.1 Disparadors (Triggers)

Les dades estadístiques necessàries per a les consultes s'emmagatzemen de forma dinàmica. Això vol dir que les dades s'han de calcular de forma automàtica i transparent per l'usuari que utilitza el producte, i han d'estar disponibles en tot moment. A més, hi ha dades que necessiten una transformació per poder tenir el sentit que es desitja.

[25] <https://www.oracletutorial.com/oracle-index/oracle-create-index/>

[27] <https://docs.oracle.com/en/database/oracle/oracle-database/21/lnpls/plsql-error-handling.html#GUID-343E0653-9BCE-48F5-A00A-795D77B96B44>

[28] [https://es.wikipedia.org/wiki/Tiempo\\_UNIX](https://es.wikipedia.org/wiki/Tiempo_UNIX)



Per aconseguir-ho, s'han creat diferents disparadors en les diferents entitats. Un disparador és una "funció" que s'executa en el moment en què una entitat rep un canvi, ja sigui un registre nou (insert), una actualització general o a una columna concreta (update) o una eliminació de registre (delete).

Els disparadors també s'executen en el cas en què un registre rep un canvi a conseqüència d'una alteració de dades d'una taula aliena (FK) per mantenir la integritat de les dades a la base de dades, i a conseqüència de la restricció de la clau forana que elimina les dades si s'elimina la clau principal.

#### 4.7.2 Llistat d'estadístiques bàsiques

S'han implementat les següents dades estadístiques:

Codi	Descripció	Estat
<b>E001</b>	Universitat que, analitzant totes les dades de què disposem, té un percentatge més alt d'estudiants que superen les pràctiques en empreses. Es consideren superades quan la persona responsable de l'estudiant a l'empresa el qualifica com a apte	Implementat
<b>E002</b>	Top5 d'empreses que han ofert més places per estudiants en pràctiques al CIC	Implementat
<b>E003</b>	Empresa que, d'entre totes les dades de què disposem, ha rebut més inspeccions de treball (sense considerar el resultat de la inspecció).	Implementat
<b>E004</b>	Donat un any universitari en concret, percentatge d'estudiants, de qualsevol universitat, que no superen les pràctiques en empreses	Implementat
<b>E005</b>	En l'any universitari en curs, sou mitjà que reben els estudiants durant les serves pràctiques	Implementat
<b>E006</b>	Nombre total d'estudiants en pràctiques en el moment de fer la consulta. Es consideraran els estudiants que han començat les pràctiques i no les tenen acabades en aquell moment	Implementat
<b>E007</b>	Durant el darrer any universitari, percentatge d'universitats controlades pel CIC que no tenen cap estudiant en pràctiques	Implementat
<b>E008</b>	Titulació universitària amb un major nombre d'estudiants que han acabat satisfactòriament les pràctiques en empreses. Es tindran en compte totes les dades de què es disposa a la BD	Implementat

<b>E009</b>	Nombre mitjà d'estudiants que es presenten a cada oferta rebuda per les diferents universitats durant l'any universitari en curs	Implementat
<b>E010</b>	Tenint en compte totes les dades de què es disposa, durada mitjana dels contractes d'estudiants amb les empreses.	Implementat
<b>E011</b>	Empresa que, en un any concret, ha realitzat el major nombre d'entrevistes a estudiants.	Implementat
<b>E012</b>	En l'any universitari en curs, professor que porta un major nombre d'estudiants que estan fent pràctiques en empreses	Implementat
<b>E013</b>	En el moment d'executar la consulta, nombre total d'informes de pràctiques pendents de ser realitzats per les persones responsables de les empreses.	Implementat
<b>E014</b>	Tenint en compte l'any universitari en curs i l'anterior, percentatge d'empreses que van oferir places l'any anterior i no n'han ofert en l'actual	Implementat

#### 4.7.3 Decisions de disseny

S'han pres les següents decisions de disseny referents a les taules estadístiques:

- Restricció de clau forana de forma lògica.

Pel correcte funcionament del mòdul estadístic, s'han eliminat les claus foranes d'aquestes taules que les relaciona amb les entitats que no pertanyen al mòdul estadístic. Per poder mantenir la integritat de les dades al mòdul estadístic, la restricció es fa de forma lògica mitjançant disparadors a les taules principals.

El motiu és que les claus foranes actualment tenen la restricció que si s'elimina la dada, les taules que apunten a aquestes claus foranes també l'eliminen (*on delete cascade*), cosa que fa que les files que corresponen amb les claus també s'esborrin. Això porta un problema derivat d'aquesta restricció, ja que hi ha valors que són acumulatius, com podrien ser els totals d'estudiants que pertanyen a una universitat, que s'han de tornar a calcular. Aquests disparadors que calculen aquestes dades, necessiten anar a buscar les dades a les taules d'estadístiques per actualitzar el valor.

És en aquest moment on es produeix l'error, ja que el disparador està intentant accedir-hi a una taula per modificar o eliminar el valor, però com que està sent modificada per l'eliminació de l'entitat principal (es una altra transacció i no es poden veure per la propietat d'aïllament), potser el valor ja s'ha eliminat, o la fila que correspon amb la dada està bloquejada fins que finalitzi la transacció.

Per exemple, suposem que tenim les entitats principals Universitat i Estudiant, i la taula d'estadístiques *e\_total\_estudiants\_universitat*, amb una clau forana *universitat\_id* amb restricció d'eliminació en cascada (on delete cascade). Si hi ha un disparador a la taula Estudiant que suma 1 a cada universitat quan es fa un *insert* i resta 1 quan es fa un *delete*, si eliminem la universitat amb identificador 1 a la taula universitat, per la restricció de la clau forana també s'eliminen tots els registres que pertanyen als estudiants de la universitat 1, de la taula Estudiants, i els registres de la universitat 1, de la taula *e\_total\_estudiants\_universitat*. Quan comença a eliminar els registres de la taula estudiant, fa que els disparadors de *delete* s'executin, intentant actualitzar una dada que no es pot modificar perquè està bloquejada (la fila que correspon a la universitat 1 de la taula d'estadístiques), o ja s'ha eliminat (un disparador anterior ha eliminat la fila). Això fa que es produeixi un problema d'integritat entre les dades i les taules, o que la transacció quedi a l'espera de la finalització d'una altra transacció.

Perquè això no passi, no s'ha definit a nivell físic de la taula la restricció, però sí a nivell lògic. Això es fa eliminant la restricció de la taula física, i afegint un disparador en les entitats principals que elimini de la taula d'estadístiques la fila que és necessària.

- Nom de les taules

Les taules d'estadística tenen una nomenclatura que permet trobar-les fàcilment. Totes les taules comencen pe *e\_*. D'aquesta manera estan 'separades' de la resta de taules.

- Dates en POSIX (epoch time)

Per fer alguns càlculs amb dates, aquestes s'han convertit a POSIX (epoch time<sup>[29]</sup><sup>[30]</sup>). POSIX és la mesura de temps en segons que han transcorregut des de la data 01/01/1970 a les 00:00:00 fins al dia d'avui. És més fàcil treballar amb segons i convertir-los després que amb anys, mesos, dies , hores, minuts i segons.

[29] [https://es.wikipedia.org/wiki/Tiempo\\_Unix](https://es.wikipedia.org/wiki/Tiempo_Unix)

[30] <https://www.epochconverter.com/>

The current Unix epoch time is **1671437678**

Convert epoch to human-readable date and vice versa

[\[batch convert\]](#)

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **seconds**:  
**GMT: Saturday, 1 January 2022 0:00:00**  
**Your time zone:** sábado, 1 de enero de 2022 1:00:00 GMT+01:00  
**Relative:** A year ago

Yr Mon Day Hr Min Sec  
 -  -   :  :

**Epoch timestamp:** 1640995200  
Timestamp in milliseconds: 1640995200000  
**Date and time (GMT):** Saturday, 1 January 2022 0:00:00  
Date and time (your time zone): sábado, 1 de enero de 2022 1:00:00 GMT+01:00

Figura 24: Exemple de conversió data-POSIX

## 4.8 Tècniques DataWarehouse (magatzem de dades)

En aquest producte s'han introduït algunes tècniques dels magatzems de dades.

Els magatzems de dades són bases de dades on s'emmagatzema informació històrica, estadística... que serveix per donar informació útil (presa de decisions, estadístiques, predicció d'esdeveniments...) i permet guardar-ne l'històric de dades per una consulta posterior.

Els magatzems de dades té les següents característiques:

- **Integrat:** Es refereix al fet que les dades han de ser coherents amb la informació que conté. Dit d'una altra forma, les dades del mòdul estadístic han d'estar relacionades i ser correctes amb les dades del sistema.
- **Històric:** Es refereix al fet que s'han d'emmagatzemar els canvis de les dades a mesura que s'utilitza el producte i les dades es van modificant. Dit d'una altra forma, totes les dades han d'estar disponibles per la seva consulta amb tots els canvis o estats que va tenint al llarg del temps.
- **No volàtil:** Es refereix al fet que totes les dades han d'estar accessibles per consultar en qualsevol moment, i no s'ha de perdre informació. Dit d'una altra forma, les dades eliminades no s'han d'esborrar del sistema.
- **Orientat a temes:** Es refereix al fet que la informació del sistema, encara que tinguin diferent tipus i dades, es puguin trobar totes de la mateixa manera i tinguin la mateixa referència. Dit d'una altra forma, totes les dades que tenen la mateixa semàntica o valors dobles, com podria ser [0,1] o [Y,N], que són valors tipus booleà, han de ser accessibles com si es tractés del mateix valor.

#### 4.8.1 Integrat

En aquest producte, la informació s'ha integrat de tal forma que les dades siguin coherents i tinguin sentit.

Per fer-ho, la informació s'ha de manipular mitjançant les funcions creades. Són aquestes funcions les que s'encarreguen que totes les dades que es volen manipular siguin coherents.

Un exemple seria la funció per afegir un candidat a una oferta de treball. El candidat ha de pertànyer a la universitat a qui s'ha assignat aquella oferta de treball. A més, l'estudiant ha de complir amb la titulació que està estudiant per poder-se apuntar. De tot això s'encarrega la funció indicada, que en cas que alguna cosa no sigui coherent, retorna un missatge d'error.

#### 4.8.2 Històric

En algunes taules d'aquest producte s'han utilitzat tècniques per mantenir l'històric de les dades per poder consultar-lo quan es necessiti.

Per fer-ho, s'han creat columnes a les taules on s'emmagatzema tant la data de la modificació com el resultat obtingut. D'aquesta manera es pot saber quan i quin és el resultat de la modificació.

Un exemple seria la taula *candidat*, on mitjançant les columnes *data\_inscripcio* es pot saber quan l'estudiant es va inscriure a l'oferta de treball, *data\_enviada\_empresa* quan la universitat va enviar la candidatura a l'empresa, *data\_validat* per saber quan l'empresa va validar la candidatura i qui ho va fer, i *data\_seleccionat* per saber quan l'empresa va seleccionar o no a l'estudiant i qui ho va fer.

#### 4.8.3 No Volàtil

A la taula *companyia* s'utilitza aquesta tècnica. Un cop la companyia ha enviat la petició d'inscripció al CIC i aquest l'ha validat, si no s'ha acceptat la inscripció de la companyia, en comptes d'eliminar la fila de la taula, es manté i es posa a 0 el valor de la columna *es\_aceptada*. D'aquesta manera es pot veure quantes empreses s'han acceptat i quantes s'han denegat.

Falta aplicar aquesta tècnica a més taules del sistema perquè sigui més eficient i no hi hagi pèrdua de dades a conseqüència de l'eliminació d'aquestes.

#### 4.8.4 Orientat a temes

Aquesta tècnica consisteix a agafar totes les dades que tenen un mateix significat i extreure'n la informació que es necessita. Per exemple, si tinguéssim una columna a la taula *Estudiant* anomenada *genere*, amb els valors possibles [H,D], on H és home i d és dona, i una columna a la taula *Treballador* anomenada *genere*, però amb els possibles valors [0,1], on 0 és home i 1 és dona, i volguéssim saber quants homes i dones hi ha al sistema, hauríem d'utilitzar aquesta tècnica per saber-ho.

Al projecte s'utilitza per trobar l'any acadèmic a partir d'una data. Existeix una funció anomenada *buscaAnyUniversitari* que té un paràmetre d'entrada que és una data i retorna l'any universitari que pertany a aquella data.

```
/**
Comprova si existeix l'any universitari. Un any universitari comença en setembre
i acaba a l'agost. Per exemple, per la data 22/10/2022, l'any universitari
correspon amb el 22/23, per contra, per la data 22/05/2022, l'any universitari
correspon amb el 21/22. Si no existeix el crea i retorna l'identificador
@param p_data data per comprovar l'any universitari
@return number Si existeix retorna l'identificador, si no existeix retorna null
**/
create or replace function buscaAnyUniversitari(
  p_data in date
) return number as
  p_month number;
  p_any number;
  p_any_universitari char(5);
  p_any_universitari_id number;
begin
  p_month := to_number(to_char(p_data, 'mm'));
  p_any := to_number(to_char(p_data, 'yy'));

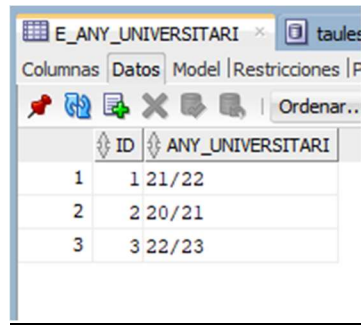
  if (p_month>8) then
    p_any_universitari := (p_any-1) || '/' || p_any;
  else
    p_any_universitari := p_any || '/' || (p_any+1);
  end if;

  select id into p_any_universitari_id
  from e_any_universitari
  where any_universitari = p_any_universitari;

  return p_any_universitari_id;
exception
  when no_data_found then
    insert into e_any_universitari(any_universitari)
    values(p_any_universitari)
    returning id into p_any_universitari_id;
    return p_any_universitari_id;
end;
```

Figura 25: Exemple d'implementació de tècniques DataWarehouse

Un cop tenim l'any universitari el guardem a les taules d'estadística per poder fer les consultes que necessitem.



ID	ANY_UNIVERSITARI
1	1 21/22
2	2 20/21
3	3 22/23

Figura 26: Exemple de taula de dades agrupades per temes

#### 4.9 Joc de dades

El joc de dades s'ha millorat gràcies a la implementació del registre d'accions. El registre ha permès la detecció de dades i relacions errònies a les taules d'una manera senzilla.

El mòdul estadístic s'omple de forma transparent per l'usuari, així que el no hi ha cap referència de forma directa a les taules d'estadística.

Els arxius que componen aquesta segona fase són:

- taules.sql : correspon amb tot el codi que permet la creació de les taules d'entitats, així com el registre d'accions i les taules estadístiques. També crea els índexs necessaris per millorar el temps de resposta de les consultes a les dades.
- funcions.sql : correspon amb tot el codi que permet la creació de les funcions necessàries tant pel mòdul estadístic com el registre d'accions.
- triggers.sql: correspon amb tot el codi que permet la creació dels disparadors, que són necessaris per omplir les dades del mòdul estadístic.
- procediments.sql: correspon amb tot el codi que permet la creació dels procediments que són necessaris per omplir les taules d'entitats de la base de dades.
- dades.sql: omple totes les taules de la base de dades mitjançant la crida als procediments creats a l'arxiu procediments.sql
- consultes estadistiques.sql: conte les consultes al mòdul estadístic demanats pel client. Cada secció està separada per poder fer la comprovació per separat. A més, també s'han creat consultes que no estan en temps constant 1 que són per fer les comprovacions de les dades d'estadístiques.
- bd.sql: exportació de la base de dades de prova de forma completa.

Per la creació de la base de dades es pot fer de dues formes:

1. Mitjançant l'execució dels arxius *taules.sql* + *funcions.sql* + *triggers.sql* + *procediments.sql* + *dades.sql* en aquest ordre. S'ha de tenir en compte que un cop executat els arxius *taules.sql*, *funcions.sql*, *triggers.sql* i *procediments.sql* no cal tornar-los a executar per fer les proves que es considerin adequades. L'arxiu *dades.sql* es pot executar tantes vegades com es vulgui, ja que cada cop que s'executa fa una neteja de les dades de les taules de la base de dades, cosa que permet fer canvis a les dades que es desitgi per veure un resultat diferent en cada execució.
2. Mitjançant l'execució de l'arxiu *Bd.sql*. **Després es pot executar l'arxiu *dades.sql* tantes vegades com es vulgui per fer les proves que es considerin.**

#### 4.10 Proves i optimitzacions

Les proves s'han realitzat mitjançant l'execució continuada de l'arxiu *dades.sql*. Aquest arxiu fa una neteja completa de les dades de la base de dades i la torna a omplir. Un exemple clar d'aquesta forma de treball és mitjançant la implementació del registre d'accions, on s'anava millorant les dades gràcies a l'execució continuada del contingut d'aquest arxiu.

A més de tot això, s'han revisat les consultes que es fan a les taules per afegir els índexs que s'han considerat adients per la millora de les consultes d'actualització, modificació, eliminació i selecció de dades.



## 5. Entrega Final

En aquesta última fase del treball s'ha revisat el producte resultant per poder assegurar-ne l'entrega al client final.

També s'ha creat una presentació del producte per poder veure de forma resumida els principals aspectes i funcionalitats del producte mitjançant una introducció general al producte i el seu funcionament. Aquesta presentació s'ha fet mitjançant un vídeo amb una durada màxima de 20 minuts.

Per poder fer la importació correcta del joc de proves, s'ha preparat un manual d'instruccions que indica com fer les importacions i les comprovacions necessàries sobre les dades i el joc de proves.

Finalment, s'ha revisat la memòria per afegir amb més detall informació referent a cadascuna de les fases que componen aquest projecte.

## 6. Futur

### 6.1 Introducció

Aquest producte compleix amb els requisits demanats, però li falta una sèrie de millores per poder estar en producció. Aquestes millores són:

- Seguretat del sistema
- Aplicació i millora de més tècniques de magatzem de dades

### 6.1 Seguretat del Sistema

És molt necessari afegir una millora de seguretat al sistema. Com que l'aplicació la faran servir molts usuaris i amb diferents permisos, és necessària aquesta millora abans de posar-la en producció.

Tant els permisos com la seguretat permetran que tant la consulta com la interacció amb les dades es faci d'una forma segura, a més de protegir les dades sensibles (personals, confidencials...) del sistema. Les dades sensibles poden ser contrasenyes, números de seguretat social...

En el cas de les dades sensibles que no es necessita saber quin és l'origen, només comprovar si és correcta o no com en el cas de les contrasenyes, la millor opció és codificar el text en pla mitjançant una funció criptogràfica de HASH, i emmagatzemar el resultat a la taula, com per exemple sha1 o sha256. La forma de comprovar la contrasenya és comparar el resultat de codificar el text entrat amb la codificació de la contrasenya de l'usuari, i si donen el mateix resultat, és correcte.

En el cas de les dades que és necessari consultar i que s'ha de saber l'origen, la millor opció es emmagatzemar el resultat de fer un xifratge simètric mitjançant una clau, o un xifratge asimètric mitjançant un parell de claus pública/privada. D'aquesta manera es pot desxifrar la dada per veure-la.

Pel que fa a les sessions, la millor manera és la de crear una taula de sessions i una especialització de les entitats Usuari, Professor i Treballador.

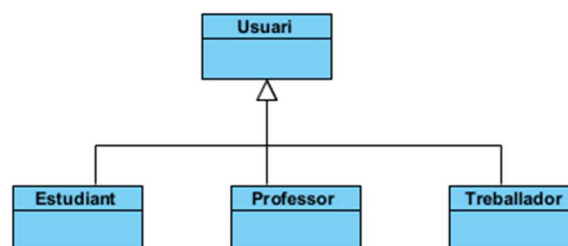


Figura 27: Especialització de les entitats Estudiant, Professor i Treballador

Mitjançant aquesta entitat, es pot fer que un cop validat l'usuari retorni una variable de sessió. Aquesta sessió ha de tenir una durada determinada, i un cop finalitzada ha de caducar donant un missatge d'error de sessió caducada.

Aquesta variable de sessió s'ha de passar a tots els procediments, i els procediments, mitjançant una funció de comprovació de sessió, l'ha de validar i permetre o no l'acció en funció de:

- La sessió ha caducat
- L'usuari té permís per fer aquella operació, sigui llegir, modificar o eliminar.

```

create or replace function comprovaSessio(
  p_sessio in sessio_usuaris.sessio%type,
  p_permis_tipus in number
) return number as
  p_te_permis number;
begin
  -- Busquem l'acció a la taula
  select permis into p_te_permis
  from sessio_usuaris
  where sessio = p_sessio;

  if(bitand(p_te_permis, p_permis_tipus) = p_permis_tipus) then
    return 1;
  else
    return 0;
  end if;
-- si no la troba dona excepció
exception
  when no_data_found then
    return null;
end;

create or replace procedure afegirUniversitat(
  p_sessio in sessio_usuaris.sessio%type,
  p_nom in universitat.nom%type,
  p_te_estudiants in universitat.te_estudiant_per_practiques%type default 1,
  RSP out varchar2
) IS
  id number;
  te_permis number;
  params json_object_t;

  no_sessio exception;
  no_permis exception;
begin
  -- Comprovem la sessio
  te_permis := comprovaSessio(p_sessio, 3); --Permis d'escriptura
  if(te_permis is null) then
    raise no_sessio;
  elsif (te_permis <> 1) then
    raise no_permis;
  end if;

  -- Inicialitzem la variable RSP
  RSP := 'OK';
  -- Creem el JSON per al registre
  params := JSON_OBJECT_T('{"id":null}');
  params.put('p_nom', p_nom);
  params.put('p_te_estudiants', p_te_estudiants);

  -- afegim el registre
  insert into universitat (nom, te_estudiant_per_practiques)
  values(p_nom, p_te_estudiants)
  returning id into id;
  commit;

  -- afegim l'identificador al JSON
  params.put('id', id);
  afegirLog($psql_unit, params.to_string, RSP);
exception
  when others then
    RSP := concat('ERROR|', SQLERRM);
    afegirLog($psql_unit, params.to_string, RSP);
  when no_sessio then
    RSP := 'ERROR|NO EXISTEIX LA SESSIO';
    afegirLog($psql_unit, params.to_string, RSP);
  when no_permis then
    RSP := 'ERROR|NO TENS PERMIS PER REALITZAR AQUESTA OPERACIO';
    afegirLog($psql_unit, params.to_string, RSP);
end afegirUniversitat;

```

Figura 28: Exemple de funció i procediment amb sessions

## 6.2 Aplicació i millora de tècniques de magatzem de dades

Es poden ampliar i millorar les tècniques de magatzem de dades. La principal seria la modificació de les taules per permetre marcar les dades com a eliminades. Aplicant aquesta millora podem aconseguir que les dades no desapareguin, incloent-hi les dades relacionades i les dades estadístiques.

Per exemple, actualment si s'elimina una universitat, totes les dades relacionades amb la universitat s'eliminen o es modifiquen, de la següent forma:

- Integritat Referencial: Per integritat referencial totes les dades que tenen la clau de la dada eliminada s'eliminen. En el cas d'una universitat, també s'eliminen les candidatures, els estudiants, les ofertes de treball...

- Dades estadístiques: Com que a conseqüència de la integritat referencial també s'eliminen dades d'altres taules, les dades estadístiques han de modificar-se perquè siguin coherents amb el sistema. D'aquesta manera, una dada amb un percentatge d'un 80% per exemple de les candidatures, queda modificada per ser coherent amb les dades que resten al sistema, però per un altre costat, la dada en si és incoherent perquè no reflecteix la realitat de les dades que hi havia abans de l'eliminació de la dada.
- Pèrdua d'informació històrica: Quan una dada s'elimina es perd aquella dada, i si en un futur torna a estar disponible, com en el cas d'una universitat, s'hauria de donar d'alta un altre cop perdent tot l'històric d'aquella universitat, incloent-hi la data de baixa i la data d'alta.

Per aplicar aquesta actualització, la millor forma és afegir a totes les taules una columna *eliminat* amb valors compresos entre [0,1] i la data de quan va ser eliminada.

ID	NOM	TE_ESTUDIANT_PER_PRACTIQUES	ELIMINAT	DATA_ELIMINACIO
3	Universitat3		1	0 (null)
4	Universitat4		1	0 (null)
5	Universitat5		1	11/05/22 00:16:07
6	Universitat6		1	0 (null)

Figura 29: Exemple de millora amb tècniques DataWarehouse

D'aquesta forma es pot tenir l'històric i les dades estadístiques tenen la coherència durant tot el funcionament del sistema.

També es poden aplicar i ampliar més tècniques de magatzem de dades, com la creació de funcions i disparadors per afegir dades d'estadística a les taules, l'agrupació de dades per temes o la millora de la informació de les dades afegint-hi més estats d'aquestes, com podria ser els estats que ha passat una companyia per poder ser acceptada al sistema i poder publicar ofertes de treball en pràctiques.

### 6.3 Altres millores

Altres millores que es poden implantar, que no són necessàries per posar en producció el producte podrien ser:

- Ampliació del mòdul estadístic: Això passa per la creació de més taules per poder emmagatzemar més dades estadístiques per poder consultar en qualsevol moment.
- Ampliació del registre d'errors: Afegir el registre d'errors a les funcions i els disparadors per poder tenir millor control sobre els possibles errors que es poden produir al sistema. Això també pot permetre resoldre d'una forma fàcil i ràpida els possibles errors que es poden produir al sistema, ja que es tenen les dades (d'entrada, data i hora, resultat...) de l'error produït.

- Creació de funcions de consulta: Les consultes més habituals es podrien crear com funcions per poder millorar-ne l'estabilitat del sistema. El fet de tenir les consultes més comunes disponibles també millora la seguretat i l'estabilitat, ja que els canvis produïts com una possible millora de les dades, es poden aplicar a la funció sense que l'aplicació que la fa servir tingui la necessitat de fer canvis amb les consultes quan es connecta per executar-les a la base de dades.

## 7. Conclusions

S'han extret les següents conclusions segons l'evolució del projecte:

### 7.1 Fase I

Respecte al desenvolupament de la fase I, s'han deduït les següents conclusions:

- Planificació: Encara que s'ha seguit la planificació, hi ha tasques que no estaven ben planificades. Per exemple, el temps dedicat al disseny físic ha estat menor del que s'esperava (de 21 hores a 7 hores). Per contra, la creació del joc de dades ha estat bastant superior (de 18 hores a 25 hores). La resta d'hores ha estat per la documentació i les proves i optimitzacions. També faltaven tasques per afegir a la planificació, com podria ser les hores de millores proposades pel consultor.
- Objectius aconseguits: S'han aconseguit alguns per aquesta primera fase, encara que s'ha utilitzat més temps del que s'havia planejat, però no s'ha complert amb tots els objectius previstos. El joc de dades té dades que poden contenir incongruències i les proves són molt bàsiques i només serveixen per comprovar les relacions entre entitats. S'ha de solucionar les incongruències de les dades i ampliar-ne el joc de proves.

De forma general, en aquesta primera fase es pot comprovar que fer una planificació correcta és molt complicat. Depenent de l'experiència i l'ús de les tecnologies que s'utilitzen, els temps estimats es poden escurçar o ampliar. Una persona amb més experiència amb un programari pot fer el mateix treball que una amb poca experiència en menys temps. A més, sempre es poden deixar de comptar tasques que, encara que requereixin un temps mínim, es necessita temps per fer-les. A tot això, s'ha d'afegir que hi ha tasques que es poden donar per "enteses" per la persona que ho planifica, però quan una altra veu la planificació, no sap quines funcionalitats estan incloses.

També hi ha hagut una estimació molt a la baixa a l'hora de crear un joc de dades. El fet de tenir una aplicació d'alt nivell que gestiona les dades simplifica el temps que es necessita per la seva introducció, i assegura el compliment de la integritat referencial entre les taules. A més, s'ha de tenir en compte que no només s'ha de complir la integritat referencial, sinó que també han de coincidir les altres dades, com per exemple les dates, ja que es pot trobar un estudiant que es va inscriure per fer les pràctiques l'any 2022, però la seva candidatura sigui aprovada l'any 2021, o s'hagi inscrit en una candidatura que no correspon amb la titulació que estudia.

## 7.2 Fase II

Aquesta segona fase ha donat més problemes del que s'havia previst inicialment. La principal dificultat derivava del trencament de la integritat de les taules a conseqüència de l'eliminació d'entitats principals i l'eliminació posterior de totes les dades de totes les taules que la tenien com a clau forana. L'error només el donaven les taules d'estadística quan s'intentaven actualitzar dades que corresponien a una clau forana que s'estava eliminant. La solució ha estat trencar la relació física de la clau forana i fer-la de forma lògica.

Un altre problema referent a les estadístiques ha estat la decisió de com es calculava un any acadèmic. Finalment, s'ha optat per fer una funció que calcula l'any acadèmic mitjançant la data d'entrada.

El registre d'accions ha estat correcte tant en temps com en planificació sense més conseqüències. L'única dificultat que s'ha trobat ha estat a l'hora de decidir el format per emmagatzemar els paràmetres d'entrada de dades. De totes les opcions que s'han pensat, la més fàcil i que dona més opcions per fer cerques i emmagatzemar sense que doni problemes ha estat el format JSON. Després de fer proves i comprovar que Oracle el suporta s'ha implementat. A més, la implementació del registre ha permès d'una forma ràpida i senzilla la millora de les dades de prova detectant errors en les relacions de dades (estudiants que superen les pràctiques sense haver-se presentat a aquella oferta de treball)

Finalment, s'ha revisat la documentació de la Fase I per afegir els comentaris del consultor.

Totes les tasques excepte la del disseny físic han finalitzat per sota del temps previst. Aquest temps extra s'ha utilitzat per resoldre els imprevistos que s'han trobat a l'hora de crear el mòdul d'estadístiques.

S'ha complert l'objectiu d'aquesta segona Fase

## 7.3 Entrega final

L'entrega final ha anat com s'havia previst al calendari. S'han revisat totes les fases del desenvolupament del projecte i millorada la documentació.

El calendari s'ha complert sense cap imprevist

La creació del PowerPoint del vídeo de presentació ha complert els temps establerts, encara que el vídeo resultant de la presentació tenia una mida massa gran, cosa que ha fet que hagués de recodificar perquè tingues una mida més acceptable.

Tenia dubtes a l'hora de si feia un manual per la importació de la base de dades de prova, però finalment l'he fet, ja que considero que la guia pot ajudar a fer les proves d'una forma més ràpida.

S'ha complert l'objectiu de l'entrega final.

## 8. Glossari

- **DB:** Són les sigles de Base de dades
- **ER:** Són les sigles de Model Entitat-Relació.
- **RRHH:** Departament de Recursos Humans d'una companyia.
- **CIC :** Consell Interuniversitari de Catalunya
- **BD :** Base de dades
- **Gantt:** És un cronograma que permet establir la durada determinada d'una activitat. El cronograma de Gantt permet la visualització de les activitats de forma vertical, i el temps requerit per cadascuna en horitzontal.
- **SGBD:** Correspon amb Sistema Gestor de Base de Dades. És el sistema que s'encarrega de gestionar les bases de dades que té emmagatzemades.
- **MySQL:** És un SGBD gratuït en la seva versió comunitat.
- **MariaDB:** És un SGBD gratuït en la seva versió comunitat.
- **noSQL:** Són DB no relacionals. La principal diferència amb les BD relacionals es que no té relacions directes amb altres dades (relacions) sinó que són objectes complets (documents).
- **MongoDB:** És un SGBD no relacional.
- **SO:** Correspon amb Sistema Operatiu.
- **Windows:** És un SO propietat de Microsoft.
- **Linux:** És un SO de programari lliure.
- **RedHat:** És un SO basat en Linux
- **Debian:** És un SO basat en Linux
- **Ubuntu:** És un SO basat en Linux
- **VM:** Correspon amb Virtual Machine. És un servei d'algunes empreses que donen un espai virtual limitat en processador, RAM i espai de disc, en un centre de dades perquè puguis crear la teva màquina.



- **Amazon EC2:** Correspon amb Amazon Elastic Cloud2. És un servei que proporciona Amazon per la gestió de VM.
- **Amazon RDS:** Correspon amb Amazon Relational Databases. És un servei que proporciona Amazon de BD relacional.
- **UOC:** Correspon amb Universitat Oberta de Catalunya.
- **Google Drive:** Servei de Google que permet la sincronització de arxius en diferents ordinadors a diferents llocs.
- **Microsoft One Drive:** Servei de Microsoft que permet la sincronització d'arxius en diferents ordinadors a diferents ubicacions.
- **ER:** Correspon amb Entitat-Relació. És un model que permet veure les entitats que componen una BD i les seves relacions.
- **PK:** Correspon amb Primary Key (clau primària). Indica que tota la informació associada a aquella fila és única. No pot haver-hi dues claus primàries iguals a la mateixa taula.
- **FK:** Correspon amb Foreign Key. Indica que un valor de la fila d'una taula X, és la clau primària d'un valor de la taula Y. Serveix per fer la relació entre les taules.
- **UNIQUE:** És una propietat que pot tenir un valor d'una taula. Indica que aquell valor és únic respecte a tots els altres registres de les taules, és a dir, no hi ha dos iguals. Per exemple, la PK és única.
- **1FN:** Correspon amb la Primera Forma Normal.
- **2FN:** Correspon amb la Segona Forma Normal.
- **3FN:** Correspon amb la Tercera Forma Normal.
- **FNBC:** Correspon amb la Forma Normal de Boyce-Codd. És una especificació de la 3FN
- **4FN:** Correspon amb la Quarta Forma Normal.
- **5FN:** Correspon amb la Cinquena Forma Normal.
- **GUI :** Correspon amb Graphical User Interface. La Interfície gràfica és un conjunt d'eines que permet als usuaris interactuar amb l'aplicació d'una forma més senzilla i visual.
- **POSIX o Unix Time o Epoch:** Mesura de temps que compta els segons transcorreguts des de la data 01/01/1970 a les 00:00:00 fins la data seleccionada.

- **JSON:** Format d'intercanvi de dades que permet intercanviar i interactuar amb les dades mitjançant múltiples llenguatges de programació. Té una estructura estàndard basat en el llenguatge Javascript.
- **Trigger:** nom en anglès que significa disparador. S'utilitza per executar codi de forma automàtica quan es produeix un canvi en les dades d'una taula.
- **Insert/Update/Delete/Select:** operacions que es poden fer a les dades de les taules. Permeten afegir dades (insert), actualitzar dades (update), eliminar dades (delete) i fer consultes sobre les dades (Select).

## 9. Bibliografia

[1] **Web:** Pàgina web calendari laboral del 2022 de la Generalitat de Catalunya: <https://web.gencat.cat/ca/actualitat/reportatges/calendarilaboral/calendari-laboral-2022/>, data de consulta: 10 d'octubre del 2022.

[2] **Web:** Pàgina web calendari laboral del 2023 de la Generalitat de Catalunya: <https://web.gencat.cat/es/actualitat/reportatges/calendarilaboral/calendari-laboral-2023/>, data de consulta: 10 d'octubre del 2022.

[3] **Web:** Pàgina de RedHat <https://www.redhat.com/es/technologies/linux-platforms/enterprise-linux>, data de consulta 11 d'octubre del 2022.

[4] **Web:** Pàgina web de Debian: <https://www.debian.org/index.es.html>, data de consulta: 11 d'octubre del 2022.

[5] **Web:** Pàgina web de productes d'Amazon AWS – EC2 (cloud computing): <https://aws.amazon.com/es/ec2/>, data de consulta: 10 d'octubre del 2022.

[6] **Web:** Pàgina web d'Oracle per fer el càlcul de costos d'una VM: <https://www.oracle.com/cloud/costestimator.html?source=:ow:o:h:po:OHPPanel1nav0625>, data de consulta: 11 d'octubre del 2022.

[7] **Web:** Pàgina web del servei d'Oracle a Amazon: [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Oracle.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Oracle.html), data de consulta: 11 d'octubre del 2022 .

[8] **Web:** Pàgina web per fer els càlculs de costos del servei d'Oracle a Amazon: <https://calculator.aws/#/estimate>, data de consulta: 11 d'octubre del 2022.

[9] **Llibre:** RODRÍGUEZ, JOSE RAMÓN i MARINÉ JOVÉ, Pere. *Planificació del projecte*. UOC - PID\_00247936, data de consulta: 19 d'octubre del 2022.

[10] **Web:** Pàgina web Windows: <https://www.microsoft.com/es-es/windows>, data de consulta: 21 d'octubre del 2022.

[11] **Web:** Pàgina web Linux: <https://www.linux.com/what-is-linux/>, data de consulta: 21 d'octubre del 2022.

[12] **Web:** Pàgina web Oracle XE: <https://www.oracle.com/es/database/technologies/appdev/xe.html>, data de consulta: 21 d'octubre del 2022.

[12] **Web:** Pàgina web Oracle PL/SQL: <https://www.oracle.com/es/database/technologies/appdev/plsql.html>, data de consulta: 21 d'octubre del 2022.

[13] **Web:** Pàgina web Oracle sqldeveloper: <https://www.oracle.com/database/sqldeveloper/>, data de consulta: 21 d'octubre del 2022.

[14] **Llibre:** CASAS ROMA, Jordi. *Disseny conceptual de base de dades*. UOC - PID\_00220512, data de consulta: 22 d'octubre del 2022.

[15] **Llibre:** HUERTAS SÁNCHEZ, M. Antònia. *Lògica i àlgebra de boole*. UOC - PID\_00149503, data de consulta: 26 d'octubre del 2022.

[16] **Llibre:** BURGUES ILLA, Xavier. *Disseny lògic de base de dades*. UOC - PID\_00220510, data de consulta: 26 d'octubre del 2022.

[17] **Llibre:** CABRÉ I SEGARRA, Blai. *Disseny físic de base de dades*. UOC - PID\_00223662, data de consulta: 27 d'octubre del 2022.

[18] Pàgina web de la documentació d'Oracle amb informació dels procediments: <https://docs.oracle.com/en/database/oracle/oracle-database/21/Inpls/CREATE-PROCEDURE-statement.html#GUID-5F84DB47-B5BE-4292-848F-756BF365EC54> , data de consulta: 25 d'octubre del 2022

[19] **Web:** Pàgina web de tutorials d'Oracle amb informació sobre els procediments: <https://www.oracletutorial.com/plsql-tutorial/plsql-procedure/> , data de consulta: 25 d'octubre del 2022

[20] Pàgina web de la documentació d'Oracle amb informació dels disparadors: <https://docs.oracle.com/en/database/oracle/oracle-database/21/Inpls/plsql-triggers.html#GUID-217E8B13-29EF-45F3-8D0F-2384F9F1D231> , data de consulta: 30 de novembre del 2022

[21] **Web:** Pàgina web d'informació sobre JSON: <https://www.json.org/json-es.html>, data de consulta: 30 de novembre del 2022

[22] **Web:** Pàgina web d'un blog que pertany a Oracle amb informació sobre JSON: <https://blogs.oracle.com/sql/post/how-to-store-query-and-create-json-documents-in-oracle-database#json-table>, data de consulta: 30 de novembre del 2022

[23] **Web:** Pàgina web de la documentació d'Oracle amb informació de les funcions: <https://docs.oracle.com/en/database/oracle/oracle-database/21/Inpls/CREATE-FUNCTION-statement.html>, data de consulta: 02 de desembre del 2022

[24] **Web:** Pàgina web de la documentació d'Oracle amb informació sobre els disparadors: <https://docs.oracle.com/en/database/oracle/oracle-database/21/Inpls/CREATE-TRIGGER-statement.html#GUID-AF9E33F1-64D1-4382-A6A4-EC33C36F237B>, data de consulta: 06 de desembre del 2022

[25] **Web:** Pàgina web de tutorials d'Oracle amb informació sobre els índexs: <https://www.oracletutorial.com/oracle-index/oracle-create-index/>, data de consulta: 19 de desembre del 2022

[26] **Web:** Pàgina web de la documentació d'Oracle amb informació sobre JSON: [https://docs.oracle.com/en/database/oracle/oracle-database/12.2/sqlrf/JSON\\_OBJECT.html#GUID-1EF347AE-7FDA-4B41-AFE0-DD5A49E8B370](https://docs.oracle.com/en/database/oracle/oracle-database/12.2/sqlrf/JSON_OBJECT.html#GUID-1EF347AE-7FDA-4B41-AFE0-DD5A49E8B370), data de consulta: 02 de desembre del 2022

[27] **Web:** Pàgina web de la documentació d'Oracle amb informació sobre les excepcions: <https://docs.oracle.com/en/database/oracle/oracle-database/21/lnpls/plsql-error-handling.html#GUID-343E0653-9BCE-48F5-A00A-795D77B96B44>, data de consulta: 02 de desembre del 2022

[28] **Web:** Pàgina web de la documentació d'Oracle amb informació sobre les excepcions: <https://docs.oracle.com/en/database/oracle/oracle-database/21/lnpls/exception-handler.html#GUID-3FECF29B-A240-4191-A635-92C612D00C4D>, data de consulta: 02 de desembre del 2022

[29] **Web:** Pàgina web de Wikipedia: [https://es.wikipedia.org/wiki/Tiempo\\_Unix](https://es.wikipedia.org/wiki/Tiempo_Unix), data de consulta: 08 de desembre del 2022

[30] **Web:** Pàgina web d'un convertidor online de dates a unix time: <https://www.epochconverter.com/>, data de consulta: 08 de desembre del 2022

[31] **Web:** Pàgina web guru99 per tècniques de DataWarehouse: <https://www.guru99.com/data-warehouse-architecture.html>, data de consulta: 12 de desembre del 2022

[32] **Web:** Pàgina web de Wikipedia amb informació DataWarehouse: [https://ca.wikipedia.org/wiki/Magatzem\\_de\\_dades](https://ca.wikipedia.org/wiki/Magatzem_de_dades), data de consulta: 15 de desembre del 2022

[33] **Web:** Pàgina web amb informació referent al DataWarehouse: <https://insightsoftware.com/es/blog/database-vs-data-warehouse-whats-the-difference/>, data de consulta: 09 de gener del 2023

[34] **Web:** Pàgina web de l'escola INESEM amb informació referent a DataWarehouse <https://www.inesem.es/revistadigital/informatica-y-tics/guia-construir-datawarehouse/>, data de consulta: 09 de gener del 2023

[35] **Web:** Pàgina web amb informació referent a DataWarehouse <http://fccea.unicauca.edu.co/old/datawarehouse.htm>, data de consulta: 09 de gener del 2023

[36] **Web:** Pàgina web d'ionos amb informació referent a DataWarehouse <https://www.ionos.es/digitalguide/online-marketing/analisis-web/los-data-warehouses-en-la-business-intelligence/>, data de consulta: 09 de gener del 2023

## 10. Annexos

### 10.1 Manual d'importació del joc de proves

S'ha creat un manual d'instruccions on s'explica com importar i utilitzar tant el joc de proves com exemples a les consultes de les taules estadístiques.

Juntament amb aquestes consultes estan les consultes sense temps constant 1 (amb *group by*, *sum...*) per poder comprovar que les dades del mòdul estan ben calculades mitjançant els disparadors.

El manual s'anomena *manual importacio bd.pdf*, i es troba dins l'arxiu comprimit anomenat *jlara\_annexes.zip*