

# Disseny i implementació de la base de dades per a una aplicació de control de canvis

**Edgar Ureña Sanjaume**  
Pla d'estudis de l'estudiant  
Àrea de treball final

**Jordi Ferrer Duran**  
**Josep Cobarsí Morales**

Data Lliurament



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

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Disseny i implementació de la base de dades per a una aplicació de control de canvis</i>
<b>Nom de l'autor:</b>	<i>Edgar Ureña Sanjaume</i>
<b>Nom del consultor/a:</b>	<i>Jordi Ferrer Duran</i>
<b>Nom del PRA:</b>	<i>Josep Cobarsí Morales</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>01/2024</i>
<b>Titulació o programa:</b>	<i>Pla d'estudis de l'estudiant</i>
<b>Àrea del Treball Final:</b>	<i>Base de dades</i>
<b>Idioma del treball:</b>	<i>Català, castellà o anglès</i>
<b>Paraules clau</b>	<i>Processos ITIL, gestió de canvis, magatzem de dades</i>

**Resum del Treball (màxim 250 paraules):** *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

Aquesta aplicació ha sorgit de la necessitat detectada per el departament comercial de l'empresa en que les aplicacions actuals que gestionen el processos ITIL més importants com la gestió de canvis en las aplicacions informàtiques són massa generalistes.

La aplicació sorgeix amb la finalitat de cobrir aquesta necessitat i pretén ser una aplicació dedicada exclusivament a la gestió de canvis en les aplicacions informàtiques

Constarà de dos mòduls, el primer dedicat al funcionament de l'aplicació per a realitzar el control de gestió de canvis, és a dir, controlar les aplicacions que són gestionades, les necessitats per a validar un canvi, les persones responsables, les tasques a realitzar si un canvi es realitza malament.

L'altre mòdul estarà dedicat a les consultes estadístiques on es guardaran les dades més significatives i importants de la aplicació en un magatzem de dades i podran ser consultades en qualsevol moment de la forma més eficient.

Els resultats de la realització d'aquest projecte han set l'obtenció d'una base de dades amb un esquema per els mòduls de la aplicació on es troben totes les taules, funcions, disparadors que permeten el bon funcionament de

l'aplicació. També un sistema de "logs" que ens permet saber tot el que fa la aplicació

Per a concloure, s'ha realitzat una base de dades operativa per les necessitats esmentades on tot el disseny ha set basat per a ser un sistema escalable, que compleixi les necessitats actual i que amb pocs canvis, pugui absorbir necessitats futures del sistema

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

This application is created due to the need founded by commercial department This need is focused on the actual applications which manages the ITIL process are too generics.

This application wants to solve this need and it wants to be a specific application for the changes management in software applications.

This application will have 2 modules, the first will manage the working flow of the application, it's dedicated to changes management.

The other module is destined to statistical queries where is based on data warehouse technology to store all the information and it could be consulted whenever in an efficient way.

The results of this project are a database with a different scheme for the different modules which has each table, functions, triggers. With this architecture the good performance of the application is guaranteed. All the information of what the application are doing is stored in a system log.

To sum up, a functional database to solve all the needs commented below is created. The design and the implementation are based on scalability and guarantee the actual needs and could be extend with few changes for the future ampliatiions of the system.

## Índex

1.	Introducció.....	1
1.1	Context i justificació del Treball.....	1
1.2	Objectius del Treball.....	1
1.3	Enfocament i mètode seguit.....	2
1.4	Planificació del Treball .....	3
1.4.1	Planificació inicial.....	3
1.4.2	Planificació en PAC 2 .....	3
1.4.3	Planificació en PAC 3 .....	4
1.4.4	Planificació en lliurament final .....	4
1.5	Breu sumari de productes obtinguts.....	5
1.5.1	Entregues parcials .....	5
1.5.2	Lliurament final .....	6
1.6	Breu descripció dels altres capítols de la memòria .....	7
2.	Resta de capítols .....	8
2.1	Disseny de la base de dades. ....	8
2.2	Descripció dels mòduls de la base de dades .....	9
2.3	Diagrama UML mòdul funcional. ....	10
2.3.1	Classe ApplicationInventory .....	11
2.3.2	Classe DevelopmentTechnology .....	12
2.3.3	Classe InfraestructureType.....	13
2.3.4	Classe Person .....	13
2.3.5	Classe Role .....	15
2.3.6	Classe Absence.....	16
2.3.7	Classe TypePlannedAbsence.....	18
2.3.8	Classe TypeUnplannedAbsence.....	19
2.3.9	Classe Change .....	19
2.3.10	Classe ChangeCategory .....	22
2.3.11	Classe GeographicalCost.....	22
2.3.12	Classe GeographicaRegion.....	24
2.3.13	Classe GeographicaZone.....	24
2.3.14	Classe Approval .....	25
2.3.15	Classe Audit .....	27
2.3.16	Classe NonCompilance .....	29
2.3.17	Classe ChangeStartup .....	30
2.3.18	Classe ChangeStartupQualification.....	31
2.3.19	Classe ActionsPlan .....	32
2.3.20	Classe Action .....	33
2.3.21	Decisions de disseny.....	34
2.4	Diagrama UML mòdul magatzem de dades “DataWarehouse” .....	38
2.4.1	Classe DWH_Changes.....	38
2.4.2	Classe DWH_ActionsPlan .....	39
2.4.3	Classe DWH_Audit .....	39
2.4.4	Classe DWH_Absence .....	40
2.4.5	Decisions de disseny .....	40
2.5	Diagrama UML mòdul consultes. ....	42
2.5.1	Classe QueryPendingChanges.....	42
2.5.2	Classe QueryBestTechnicalManager .....	42

2.5.3	Classe QueryOpenedChangesTechnicalManager.....	43
2.5.4	Classe QueryFailedReplannedChangesByRegion .....	43
2.5.5	Classe QueryOpenedActionsPlan .....	43
2.5.6	Classe QueryMostUnplannedReplacedAprover .....	44
2.5.7	Classe QueryFailedExecutionLastYear .....	44
2.5.8	Classe QueryActionsPercentagePerYear.....	44
2.5.9	Classe QueryApprovedChangesByGCABCCurrentYear.....	45
2.5.10	Classe QueryNonComplianceLastSixMonths.....	45
2.5.11	Decisions de disseny.....	45
2.6	Disseny implementacions ABM.....	47
2.6.1	Procediments ApplicationInventory.....	47
2.6.2	Procediments DevelopmentTechnology .....	48
2.6.3	Procediments InfraestructureType.....	48
2.6.4	Procediments Person .....	49
2.6.5	Procediments Role .....	50
2.6.6	Procediments Absence.....	51
2.6.7	Procediments TypePlannedAbsence.....	52
2.6.8	Procediments TypeUnplannedAbsence.....	52
2.6.9	Procediments Change .....	53
2.6.10	Procediments ChangeCategory .....	53
2.6.11	Procediments GeographicalCost.....	54
2.6.12	Procediments GeographicalRegion.....	55
2.6.13	Procediments GeographicalZone .....	55
2.6.14	Procediments Approval .....	56
2.6.15	Procediments Audit .....	57
2.6.16	Procediments NonCompliance .....	57
2.6.17	Procediments ChangeStartup .....	58
2.6.18	Procediments ChangeStartupQualification.....	58
2.6.19	Procediments ActionsPlan.....	59
2.6.20	Procediments Action.....	60
2.6.21	Altres procediments.....	60
2.7	Disseny i implementació de les funcions disparadors per magatzem de dades	61
2.7.1	Funció Maintenance_DWH_Changes.....	61
2.7.2	Funció Maintenance_DWH_Absence .....	61
2.7.3	Funció Maintenance_DWH_ActionsPlan .....	61
2.7.4	Funció Maintenance_DWH_Audit.....	61
2.8	Disseny i implementació de les funcions disparadors per consultes	61
2.8.1	Funció Maintenance_QueryPendingChanges .....	61
2.8.2	Funció Maintenance_QueryBestTechnicalManager .....	61
2.8.3	Funció Maintenance_QueryOpenedChangesTechnicalManager ..	62
2.8.4	Funció Maintenance_QueryFailedReplannedChangesByRegion ..	62
2.8.5	Funció Maintenance_QueryOpenedActionsPlan .....	62
2.8.6	Funció Maintenance_QueryMostUnplannedReplacedAprover .....	62
2.8.7	Funció Maintenance_QueryMostUnplannedReplacedAprover .....	62
2.8.8	Funció Maintenance_QueryFailedExecutionLastYear .....	62
2.8.9	Funció Maintenance_QueryActionsPercentagePerYear.....	62
2.8.10	Funció Maintenance_QueryApprovedChangesByGCABCCurrentYear.....	62
2.8.11	Funció Maintenance_NonComplianceLastSixMonths .....	62

2.9	Disseny i implementacions dels logs.....	63
2.10	Disseny implementacions del joc de proves .....	63
2.11	Disseny implementacions de la prova de funcions .....	64
3.	Conclusions.....	65
4.	Glossari .....	66
5.	Bibliografia.....	67
6.	Annexos .....	68

## Llista de figures

Figura 1. Planificació inicial .....	3
Figura 2. Planificació PAC 2.....	4
Figura 3. Planificació PAC 3.....	4
Figura 4. Planificació entrega final .....	4
Figura 5. Mòduls base de dades .....	9
Figura 6. Diagrama UML mòdul funcional .....	10
Figura 7. Classe ApplicationInventory .....	11
Figura 8. Relació 1 ApplicationInventory .....	11
Figura 9. Relació 2 ApplicationInventory .....	12
Figura 10. Classe DevelopmentTechnology.....	12
Figura 11. Classe InfraestructureType .....	13
Figura 12. Classe Person .....	13
Figura 13. Relació 1 Person .....	14
Figura 14. Classe Role .....	15
Figura 15. Classe Absence .....	16
Figura 16. Relació 1 Absence .....	16
Figura 17. Relació 2 Absence .....	17
Figura 18. Classe TypePlannedAbsence .....	18
Figura 19. Classe TypeUnplannedAbsence .....	19
Figura 20. Classe Change.....	19
Figura 21. Relació 1 Change.....	20
Figura 22. Relació 2 Change.....	20
Figura 23. Classe ChangeCategory .....	22
Figura 24. Classe GeographicalCost.....	22
Figura 25. Relació 1 GeographicalCost.....	23
Figura 26. Relació 2 GeographicalCost.....	23
Figura 27. Classe GeographicalRegion.....	24
Figura 28. Relació 1 GeographicalRegion.....	24
Figura 29. Classe GeographicalZone.....	24
Figura 30. Classe Approval .....	25
Figura 31. Relació 1 Approval .....	26
Figura 32. Relació 2 Approval .....	26
Figura 33. Relació 3 Approval .....	27
Figura 34. Classe Audit .....	27
Figura 35. Relació 1 Audit .....	28
Figura 36. Relació 2 Audit .....	28
Figura 37. Relació 3 Audit .....	29
Figura 38. Classe NonCompliance.....	29
Figura 39. Classe ChangeStartup .....	30
Figura 40. Relació 1 ChangeStartup .....	30
Figura 41. Relació 2 ChangeStartup .....	31
Figura 42. Classe ChangeStartupQualification.....	31
Figura 43. Classe ActionsPlan .....	32
Figura 44. Relació 1 ActionsPlan .....	33
Figura 45. Classe Action .....	33
Figura 46. Relació 1 Action .....	34



Figura 47. Decisions absencies.....	34
Figura 48. Decisions Change .....	35
Figura 49. Decisions GeographicalCost .....	35
Figura 50. Decisions startup.....	36
Figura 51. Decisions Approval.....	36
Figura 52. Decisions GeographicalZone .....	37
Figura 53. Decisions ActionsPlan.....	37
Figura 54. Classe DWH_Changes .....	38
Figura 55. Classe DWH_ActionsPlan.....	39
Figura 56. Classe DWH_Audit.....	39
Figura 57. Classe DWH_Absence.....	40
Figura 58. Decisions DWH_Changes.....	41
Figura 59. Decisions DWH_Absence .....	41
Figura 60. Classe Query_PendingChanges .....	42
Figura 61. Classe Query_BestTechnicalManager .....	42
Figura 62. Classe Query_OpenedChangesTechnicalManager .....	43
Figura 63. Classe Query_FailedReplannedChangesByRegion.....	43
Figura 64. Classe Query_OpenendActionsPlan .....	43
Figura 65. Classe Query_MostUnplannedReplacedAprover.....	44
Figura 66. Classe Query_FailedExecutionLastYear .....	44
Figura 67. Classe Query_ActionsPercentagePerYear .....	44
Figura 68. Classe Query_ApprovedChangesByGCABCCurrentYear .....	45
Figura 69. Classe Query_NonComplianceLastSixMonths.....	45
Figura 70. Decisions disseny consultes 1 .....	46
Figura 71. Decisions disseny consultes 2 .....	46

# 1.Introducció

## 1.1 Context i justificació del Treball

Aquest treball té l'objectiu de realitzar una aplicació que controli la gestió de canvis en aplicacions informàtiques. Aquest tipus d'aplicació pretén gestionar un dels processos ITIL(Information Technology Infrastructure Library) més importants.

Els processos ITIL són un conjunt de bones pràctiques àmpliament reconegudes en el sector de la gestió de serveis de tecnologies de la informació. El seus objectius es basen en la millora continua, els objectius del negoci, la gestió de canvis, d'incidències, problemes amb la finalitat de fer que els serveis TI sigui eficients i de qualitat.

Per tant la temàtica del treball inclosa en aquests processos es un tema rellevant ja que i esta inclòs i té l'objectiu de millorar el procés de gestió de canvis.

Actualment ja existeixen aplicacions enfocades en la gestió de canvis, el problema actual que és el que vols millorar és que aquests programaris són genèrics per a poder donar abast a varis processos ITIL i no cobreixen alguna necessita especifica com ho faria una aplicació dedicada. Per tant, el que es vol aconseguir es realitzar un programari exclusiu per al gestió de canvis que inclogui totes les necessitats per assegurar una bona gestió de canvis i la consulta d'aquests.

## 1.2 Objectius del Treball

Desenvolupar una base de dades capaç de realitzar totes les funcionalitats requerides.

Desenvolupar una base de dades capaç de donar resposta a consultes estadístiques de forma eficient en termes de temps de resposta.

Documentar tot el procés de realització del projecte en la memòria tècnica.

Durant tot el desenvolupament, es tindrà molt en compte aspectes ètics, s'actuarà de forma honesta i respectuosa amb els drets humans i la diversitat.

[1]Referencia bibliogràfica sobre Processos ITIL

*Que es ITIL? Conceptes y principios* [en línia] [Consulta: 10 de Octubre de 2023]. Disponible a : <https://www.servicetonic.com/es/itil/3-itil-conceptos-y-principios/>

### 1.3 Enfocament i mètode seguit

L'estratègia per a dur a terme el treball es basa en partir dels requeriments que s'han recollit dels diferents stakeholders i s'han traslladat a l'enunciat d'aquest treball.

Partint d'aquesta base, es consideren dues estratègies a seguir.

La primera és desenvolupar un producte nou partint de la base de les especificacions comentades abans.

La segona es utilitzar, si existeix, una aplicació amb un funcionament semblant al desitjat i modificar-la.

La estratègia triada i que es considera la més òptima és la primera ja que moltes vegades és més simple començar des de el principi una aplicació ven definida que modificar una aplicació actual, ja que el més probable es que moltes taules siguin bastant diferents i s'hagin d'eliminar i fer de nou.

La modificació dels atributs de les columnes existents per canviar per exemple el tipus de dades podria no estar permesa i tornar-nos a obligar a eliminar i tornar a crear la taula.

Les restriccions, claus primàries i claus foranies també poden donar molts problemes durant la manipulació dels atributs de les taules.

Per tots aquests problemes que poden sorgir és millor començar en un projecte en blanc, tenint en compte que si existeixen scripts guardats d'algun altre projecte per a realitzar funcions, crear taules, que puguin servir per la creació de la nova aplicació és millor manipular-los o consultar-los que no fer-ho de zero.

## 1.4 Planificació del Treball

En el següents subapartats s'indica i s'actualitza la planificació dels treballs a realitzar en les entregues del projecte

### 1.4.1 Planificació inicial

En la següent imatge es pot veure la planificació inicial que es va fer durant el pla de treball per assegurar el correcte desenvolupament de l'aplicació en els terminis establerts

id	Tasca	Durada (Hores)	Data d'inici	Data de fi	Estat	25/09/2023	02/10/2023	09/10/2023	16/10/2023	23/10/2023	30/10/2023	06/11/2023	13/11/2023	20/11/2023	27/11/2023	04/12/2023	11/12/2023	18/12/2023	25/12/2023	01/01/2024	08/01/2024	15/01/2024
1	Repàs del disseny de bases de dades en magatzems de dades (Data Warehouse)	6	05/10/2023	08/10/2023	Pendent																	
2	Elecció de motor de base de dades més indicat	5	08/10/2023	11/10/2023	Pendent																	
3	Disseny UML de les classes principals de la base de dades.	30	11/10/2023	21/10/2023	Pendent																	
4	Definició de les restriccions de les classes principals de la base de dades.	25	21/10/2023	28/10/2023	Pendent																	
5	Definició de les funcions de cada classe principal.	18	21/10/2023	28/10/2023	Pendent																	
6	Desenvolupament taules i procediments de les classes principals	25	28/10/2023	07/11/2023	Pendent																	
7	Disseny UML de les classes per les consultes estadístiques.	35	07/11/2023	17/11/2023	Pendent																	
8	Definició de les restriccions de les classes per les consultes estadístiques.	18	17/11/2023	24/11/2023	Pendent																	
9	Definició de les funcions de cada classe estadística.	25	17/11/2023	24/11/2023	Pendent																	
10	Desenvolupament taules i procediments de les classes estadístiques	25	24/11/2023	04/12/2023	Pendent																	
11	Realització dels jocs de proves per a provar l'aplicació.	15	04/12/2023	09/12/2023	Pendent																	
12	Realització de les proves en l'aplicació.	30	09/12/2023	18/12/2023	Pendent																	
13	Realització de la memòria.	180	09/10/2023	23/12/2023	Pendent																	
14	Realització de la presentació virtual	20	24/12/2023	08/01/2024	Pendent																	

Figura 1. Planificació inicial

Com es pot observar en el Gantt, per la PAC2 estarà definit i implementat tant el diagrama UML del mòdul funcional com la seva implementació a la base de dades. També hauria d'estar finalitzat o quasi el disseny UML de les classes estadístiques.

Per la PAC3 el projecte estarà acabat i la memòria estarà acabada o a punt de finalitzar i es podrà començar a preparar la presentació virtual.

### 1.4.2 Planificació en PAC 2

Durant en desenvolupament de la aplicació s'ha detectat que per la correcte definició de l'arquitectura de la aplicació i perquè no comporti més feina en següents mòduls, es decideix primer realitzar tota la arquitectura en UML abans de començar el desenvolupament de la base de dades.

Es torna a planificar el diagrama de Gantt amb la nova organització de tasques i assegurant complir les fites en les dades previstes.

També es canvia l'estat de les fites segons el estat actual, s'actualitzen les durades de les tasques ja realitzades amb la duració real i es re planifiquen les tasques pendents revalorar algunes de les durades ja que durant el desenvolupament de les primeres tasques s'ha observat que algunes poden necessitar més temps i altres menys.

En al següent imatge es pot observar l'actualització de la planificació tant amb les dates com l'ordre d'execució d'aquestes.

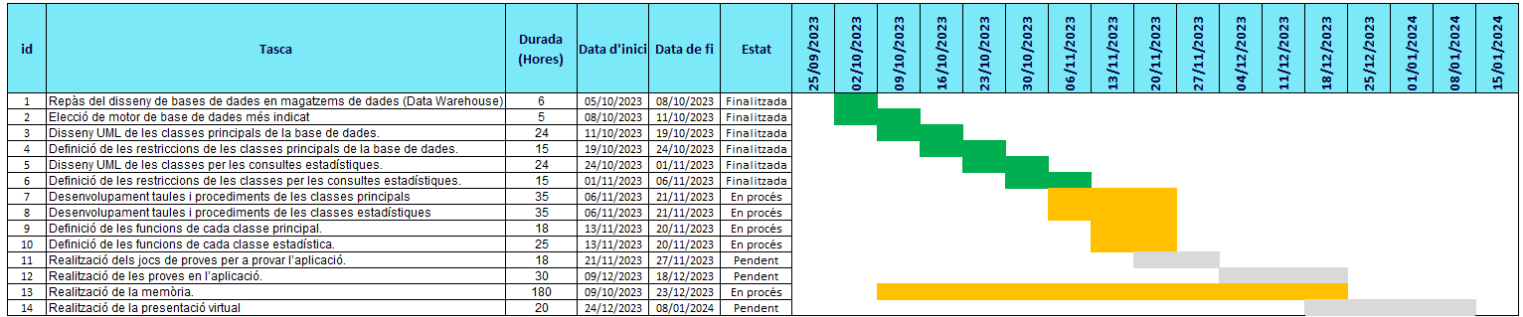


Figura 2. Planificació PAC 2

### 1.4.3 Planificació en PAC 3

Durant el desenvolupament de la PAC 2, s'ha donat prioritat el joc de proves per a provar que tot el mòdul funcional funciona correctament i al sistema de logs, cosa que ha pausat el desenvolupament del mòdul estadístic.

Tot i aquests canvis, no afecta a les dates finals d'entrega

En al següent imatge es pot observar l'actualització de la planificació

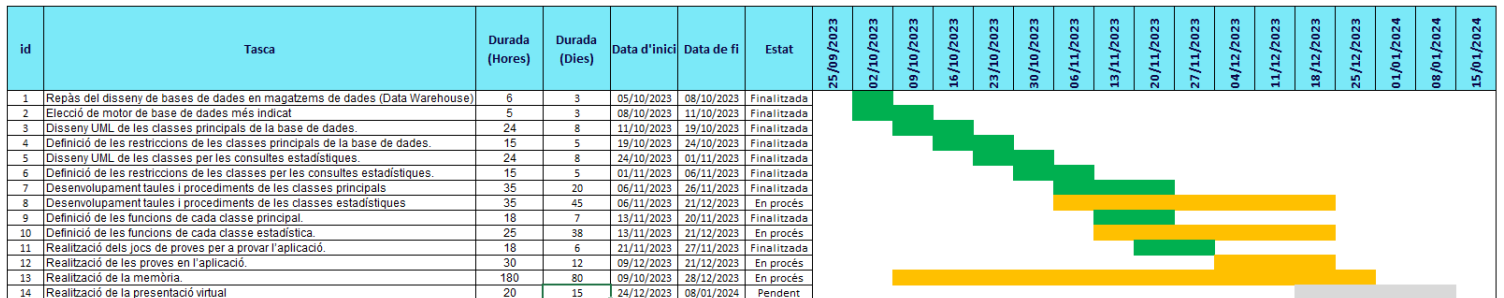


Figura 3. Planificació PAC 3

### 1.4.4 Planificació en lliurament final

Durant la finalització de l'entrega final s'han modificat la data de finalització de la memòria, ja que s'ha esperat a finalitzar la presentació virtual i s'ha actualitzat la data de fi de la presentació, mantenint les fites finals del 15 de gener com a màxim.

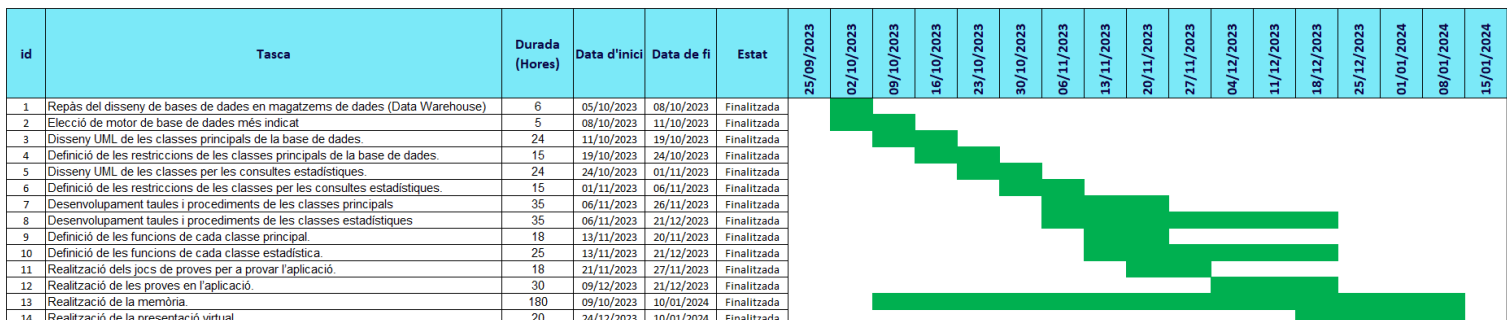


Figura 4. Planificació entrega final

## 1.5 Breu sumari de productes obtinguts

Aquest apartat es divideix en dos apartats, les entregues parcials i el lliurament final.

En les entregues parcials es detallarà que s'ha entregat en les diferents PACS durant el curs.

En el lliurament final es detallarà els documents i arxius que s'han entregat.

### 1.5.1 Entregues parcials

En la PAC 1 s'entrega el pla de treball on es determinen els següents punts

- Problemes que pretén resoldre el projecte
- Treball concret que es portarà a terme
- Objectius a assolir en el projecte
- Descomposició del treball en tasques
- Planificació de les tasques en fites temporals
- Anàlisi de riscos i pla de contingència

En la PAC 2 s'entrega la memòria actualitzada amb totes les tasques realitzades durant la PAC i amb la actualització de la planificació i les instruccions SQL per a crear i eliminar totes les taules del sistema.

Les tasques realitzades i documentades són les següents

- Modelització UML del mòdul funcional
- Modelització UML del magatzem de dades del mòdul estadístic.
- Modelització UML del sistema de consultes del mòdul estadístic.
- Explicació dels dubtes i decisions de disseny del UML.
- SQL per crear i eliminar els diferents esquemes de base de dades
- SQL per crear i eliminar totes les taules del mòdul funcional.
- SQL per crear i eliminar totes les taules del magatzem de dades del mòdul estadístic.
- SQL per crear i eliminar totes les taules de les consultes del mòdul estadístic.

En la PAC 3 s'entrega la memòria actualitzada amb totes les tasques realitzades durant la PAC i amb la actualització de la planificació i les instruccions SQL per a crear i eliminar totes les taules del sistema.

Les noves tasques realitzades i documentades són les següents

- Documentació de totes les funcions
- Documentació del sistema de logs
- Documentació del joc de proves
- Documentació del test de funcions
- SQL per crear totes les funcions
- SQL per el joc de proves
- SQL per la prova de funcions

## 1.5.2 Lliurament final

En el lliurament final s'entrega la memòria finalitzada amb tota la documentació dels mòduls del sistema, totes les funcions i les diferents instruccions SQL per a crear la base de dades, realitzar el joc de proves i provar totes les funcionalitats del sistema. També en els annexos es troba informació com la planificació final i els diagrames UML

La documentació entregada és la següent

- Planificació final
- Diagrames UML
- Memòria definitiva
- SQL per crear els esquemes
- SQL per crear les taules
- SQL per crear les funcions
- SQL per crear els disparadors
- SQL per eliminar els esquemes
- SQL per eliminar les taules
- SQL per eliminar les funcions
- SQL per eliminar els disparadors
- SQL per crear el joc de proves
- SQL per executar les proves de funcions
- SQL per les proves del sistema
- Arxiu "llegeix-me" per explicar l'ordre d'execució dels SQL
- Vídeo presentació del treball
- Auto informe

## 1.6 Breu descripció dels altres capítols de la memòria

En els següents capítols de la memòria trobem tota la informació detallada del desenvolupament del projecte.

En el capítol 2.1 trobem la definició del disseny de la base de dades

En el capítol 2.2 trobem la definició dels mòduls de la base de dades

En el capítol 2.3 trobem el diagrama UML del mòdul funcional i la definició de totes les seves classes

En el capítol 2.4 trobem el diagrama UML del mòdul de magatzem de dades i la definició de totes les seves classes

En el capítol 2.5 trobem el diagrama UML del mòdul de consultes i la definició de totes les seves classes

En el capítol 2.6 trobem el disseny i implementació IBM del mòdul funcional.

En el capítol 2.7 trobem el disseny i implementació de les funcions disparadors per el magatzem de dades

En el capítol 2.8 trobem el disseny i implementació de les funcions disparadors per el mòdul de consultes

En el capítol 2.9 trobem el disseny i implementació del sistema de logs

En el capítol 2.10 trobem el disseny i implementació del joc de proves

En el capítol 2.10 trobem el disseny i implementació de la prova de funcions



## 2. Resta de capítols

### 2.1 Disseny de la base de dades.

La base de dades s'ha dissenyat, en tres mòduls específics per cobrir les necessitats del projecte i amb una estructura que pretén evitar un gran consum de recursos, es a dir, que es necessiti un ordinador molt potent o varis ordinadors perquè funcioni l'aplicació. Aquesta decisió s'ha pres per actuar de manera sostenible amb el medi ambient.

També es té molt en compte durant tot el desenvolupament no incloure atributs a les taules referent a races, ètnies, gèneres... ja que no té implicació amb l'objectiu del projecte i podria donar lloc a que en un futur, quan el projecte estigues entregat, que es poguessin generar consultes que no actuessin de forma ètica, socialment responsable o respectuosa amb els drets humans. Per tant, ja que no té implicació en l'objectiu del projecte, s'ha tingut en compte en el disseny de l'arquitectura no incloure camps d'aquests tipus.

L'aplicació té un sistema per auditar el correcte funcionament de la mateixa i així assegurar-nos de forma honesta si el funcionament es correcte o no. També es crearan rols d'usuaris per accedir a la base de dades perquè ningú que no sigui l'administrador, pot manipular les dades.

També te un sistema de logs, on s'indica el que esta fent l'aplicació, indicant la funció que s'executa, els seus paràmetres i el resultat d'execució.

Per el desenvolupament de la base de dades, s'ha escollit POSTGRES SQL 16 i per a realitzar la programació s'utilitza PgAdmin 4 per la creació de totes les taules i funcions.

La base de dades s'ha dividit en tres esquemes.

- “Functional” on es creen totes les taules i funcions referents al mòdul funcional
- “DataWarehouse” on es creen totes les taules i funcions referents al mòdul estadístic de magatzem de dades.
- “Queries” on es creen totes les taules i funcions referents al mòdul estadístic a les consultes.

## 2.2 Descripció dels mòduls de la base de dades

La base de dades esta diferenciada en dos mòduls on cadascun té funcionalitats diferents. Aquest mòduls són: el mòdul funcional, el mòdul estadístic que consta de les parts de magatzem de dades i de consultes.

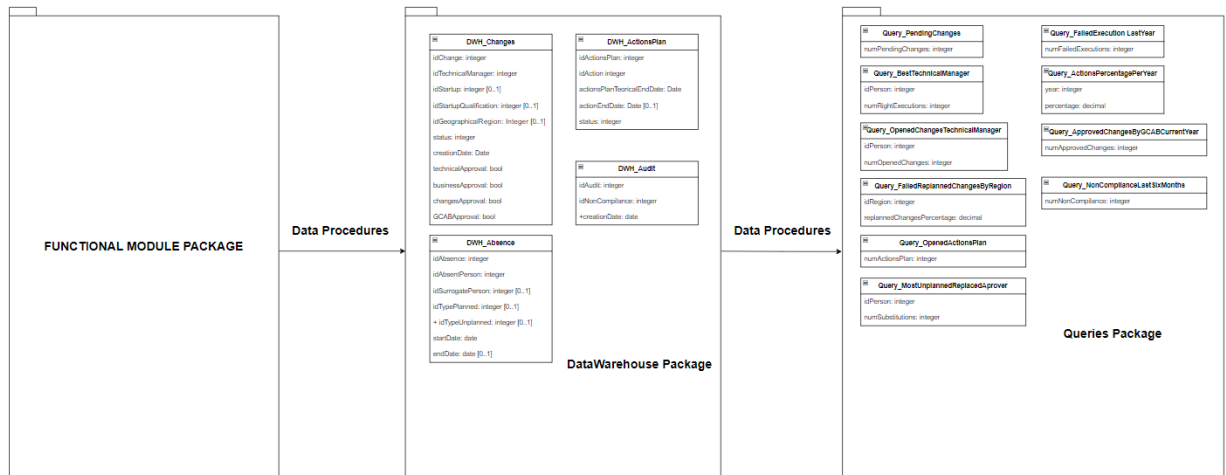


Figura 5. Mòduls base de dades

El mòdul funcional és el responsable de controlar el funcionament de l'aplicació, en aquest és on es s'emmagatzemaran totes les aplicacions, tots els canvis, les persones implicades i tota la informació relacionada amb la gestió dels canvis. També es controlarà tot l'avanç en els canvis i la seva informació derivada.

El submòdul de magatzem de dades és el responsable de guardar la informació més important del mòdul funcional per a poder donar tota la informació necessària al mòdul de consultes. La informació a emmagatzemar s'obté de funcions de traspàs d'informació que es realitzaran en els diferents canvis d'estat del mòdul funcional (Inserció o modificació).

El mòdul de consultes serà el responsable de tenir la informació guardada i organitzada en funció de les consultes que es necessitin del sistema. La informació a emmagatzemar s'obté de funcions de traspàs d'informació que es realitzaran en els diferents canvis d'estat del mòdul de magatzem de dades (Inserció o modificació).

## 2.3 Diagrama UML mòdul funcional.

En aquest apartat es defineix el diagrama UML per al mòdul funcional, cada classe amb els seus atributs, les restriccions de cada una d'elles i l'explicació de les decisions que s'han pres.

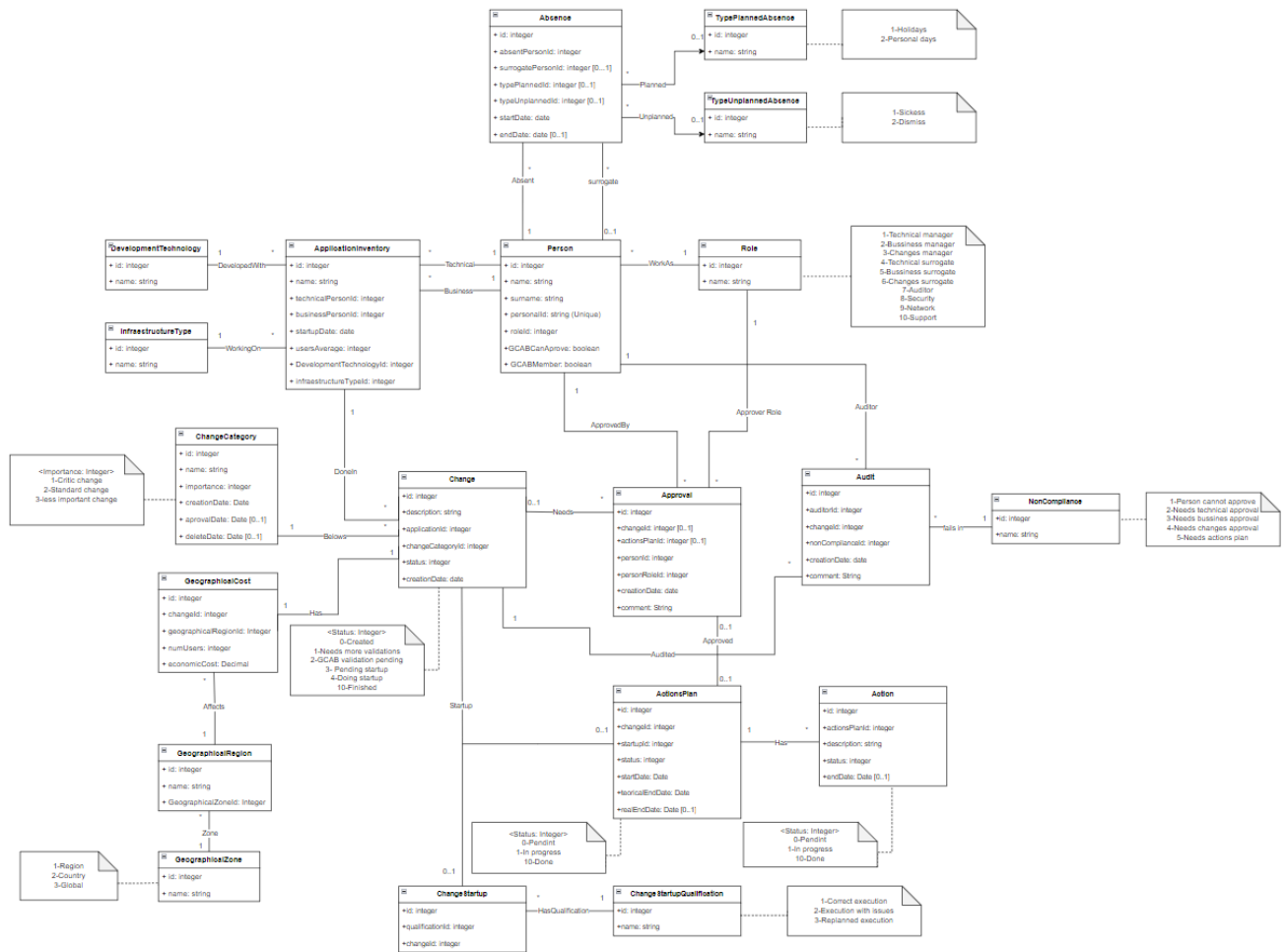


Figura 6. Diagrama UML mòdul funcional

A continuació definim cada classe, amb els seus atributs, les seves relacions i les seves restriccions que no s'han pogut representar en el diagrama

Les relacions només s'explicaran en la classe on es realitzarà la relació en la base de dades en comptes d'explicar-se en les diferents classes en que es crea per tal de no duplicar informació.

### 2.3.1 Classe ApplicationInventory

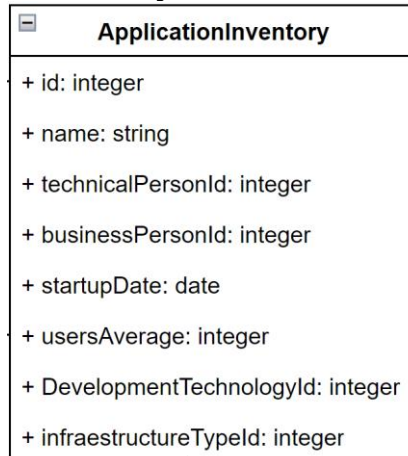


Figura 7. Classe ApplicationInventory

#### Definició

Representa l'inventari d'aplicacions de les quals es gestionen els canvis

#### Atributs

- Id: Per identificar l'aplicació i es auto incremental.
- Name: Nom únic per identificar l'aplicació
- TechnicalPersonId: Identifica la id de la persona que té el rol de cap tècnic per tant es la persona de contacte tècnica.
- BusinessPersonId: Identifica la id de la persona que té el rol de cap d'àrea de negoci implicada
- startupDate: Data de posada en producció de l'aplicació
- usersAverage: Mitjana d'usuaris que utilitzen la aplicació
- DevelopmentTechnologyId: Id que identifica la tecnologia que s'ha utilitzat per el seu desenvolupament
- InfrastructureTypeId: Id que identifica la infraestructura utilitzada per l'aplicació

#### Relacions

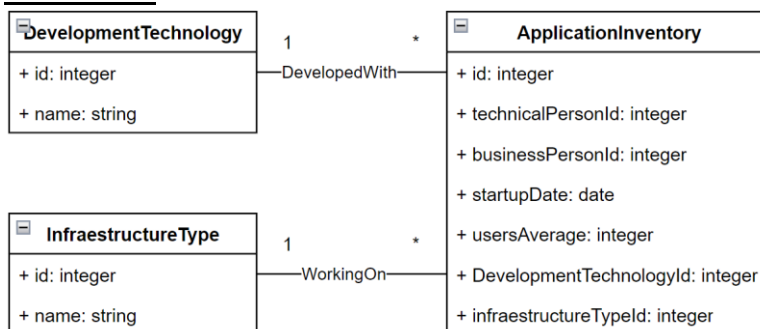
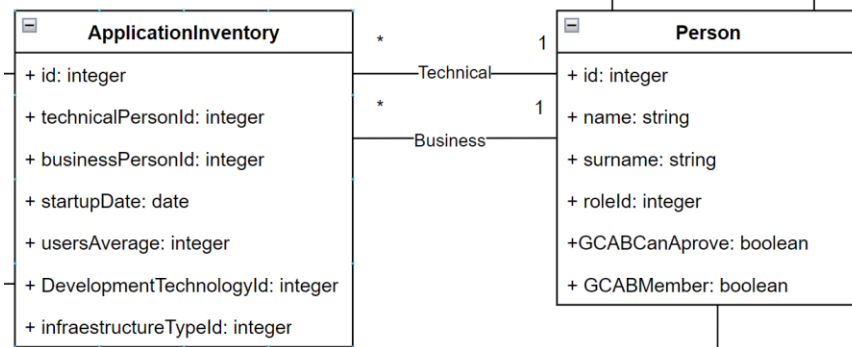


Figura 8. Relació 1 ApplicationInventory

La relació 1..\* "DevelopedWith" indica que una aplicació només pot ser desenvolupada amb una tecnologia, però una tecnologia pot ser utilitzada per desenvolupar múltiples aplicacions.

La relació 1..\* “WorkingOn” indica que una aplicació només pot funcionar amb un tipus d’infraestructura, però en un tipus d’infraestructura, poden funcionar múltiples aplicacions.



**Figura 9. Relació 2 ApplicationInventory**

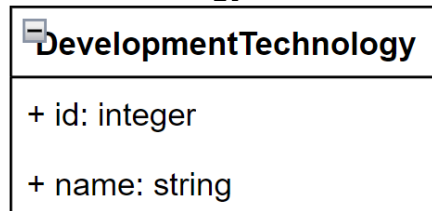
La relació 1..\* “Technical” indica que només una persona pot ser el responsable tècnic d’una aplicació, però un responsable tècnic ho pot ser de varies aplicacions.

La relació 1..\* “Business” indica que només una persona pot ser el responsable de negoci d’una aplicació, però un responsable de negoci ho pot ser de varies aplicacions.

**Restriccions no representades en el diagrama**

En aquesta classe tenim la restricció que l’atribut id ha de ser únic i l’atribut name també. L’atribut Id ha de ser auto incremental

**2.3.2 Classe DevelopmentTechnology**



**Figura 10. Classe DevelopmentTechnology**

**Definició**

Representa les tecnologies de desenvolupament de les aplicacions

**Atributs**

- Id: Per identificar la tecnologia utilitzada
- name: nom per identificar la tecnologia.

**Relacions**

No es contemplen relacions en aquesta classe

**Restriccions no representades en el diagrama**

En aquesta classe tenim la restricció que l’atribut id ha de ser únic i l’atribut name també.

### 2.3.3 Classe InfraestructureType

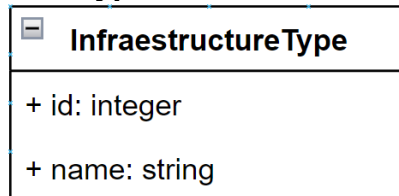


Figura 11. Classe InfraestructureType

#### Definició

Representa el tipus de infraestructura que utilitza l'aplicació.

#### Atributs

- Id: Per identificar la infraestructura
- name: nom per identificar la infraestructura.

#### Relacions

No es contemplen relacions en aquesta classe

#### Restriccions no representades en el diagrama

En aquesta classe tenim la restricció que l'atribut id ha de ser únic i l'atribut name també.

### 2.3.4 Classe Person

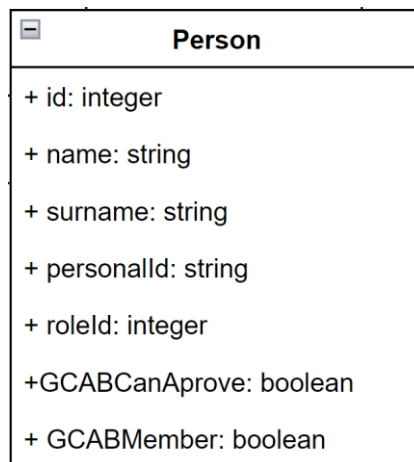


Figura 12. Classe Person

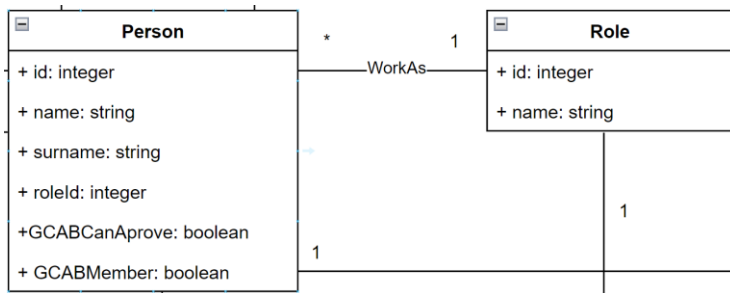
#### Definició

Representa les persones involucrades en el procés de gestió dels canvis

#### Atributs

- Id: Per identificar la persona i és auto incremental
- name: nom per de la persona
- surname: Cognom de la persona
- personalId: Identificador únic de la persona (DNI)
- roleId: identifica la id del rol de la persona
- GCABcanAprove: Indica si la persona pot fer aprovacions en GCAB
- GCABMember: Indica si la persona es membre de GCB

## Relacions.



**Figura 13. Relació 1 Person**

La relació 1..\* "WorkAs" indica una persona només pot tenir un rol, però un rol pot ser assignat a varies persones.

## Restriccions no representades en el diagrama

El camp personalId ha de ser únic per a cada persona.

L'atribuït id ha de ser auto incremental

Una persona només pot generar aprovacions dels canvis en les aplicacions en les que pot tenir el rol de responsable tècnic, responsable de negoci, ser el responsable de canvi, substitut d'aquests o membre amb dret a vot de GCAB.

Una persona no pot ser substituïda en una absència si té una absència oberta o si ja esta realitzat una substitució en el mateix període de temps.

El rol de la persona per a realitzar la substitució ha de ser el de substitut d'aquella àrea de la persona a substituir

Per a realitzar auditories, el rol de la persona ha de ser auditor.

### 2.3.5 Classe Role

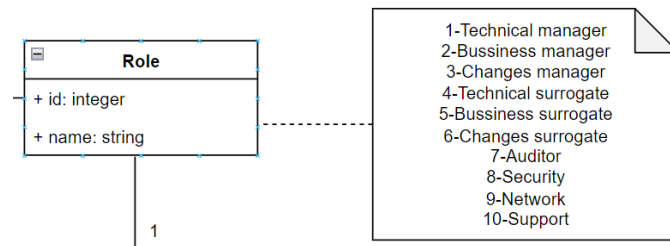


Figura 14. Classe Role

#### Definició

Representa el tipus de rol que pot tenir una persona

#### Atributs

- Id: Per identificar el rol
- name: nom per identificar el rol.

#### Valors per defecte

Aquesta classe té uns valors assignats per defecte, segons id i nom i serà possible ampliar els rols.

Els valors per defecte són els següents

- 1- Responsable tècnic,
- 2- Responsable de negoci
- 3- Responsable de canvis
- 4- Substitut tècnic
- 5- Substitut de negoci
- 6- Substitut de canvis
- 7- Auditor
- 8- Seguretat
- 9- Xarxes
- 10-Suport

#### Relacions

No es contemplen relacions en aquesta classe

#### Restriccions no representades en el diagrama

En aquesta classe tenim la restricció que l'atribut id ha de ser únic i l'atribut name també.



### 2.3.6 Classe Absence

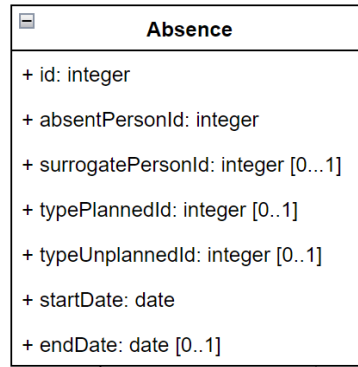


Figura 15. Classe Absence

#### Definició

Representa una absència d'una persona

#### Atributs

- Id: Per identificar la absència i és auto incremental
- absentPersonId: Per identificar la persona que genera la absència
- surrogatePersonId: Per identificar la persona que fa la substitució en cas de que sigui necessari.
- typePlannedId: Per identificar el tipus d'absència planificada en cas de que ho sigui
- typeUnplannedId: Per identificar el tipus d'absència no planificada en cas de que ho sigui
- startDate: Per identificar la data en que comença l'absència
- endDate: Per identificar quan va acabar l'absència, en cas de que hagi acabat.

#### Relacions

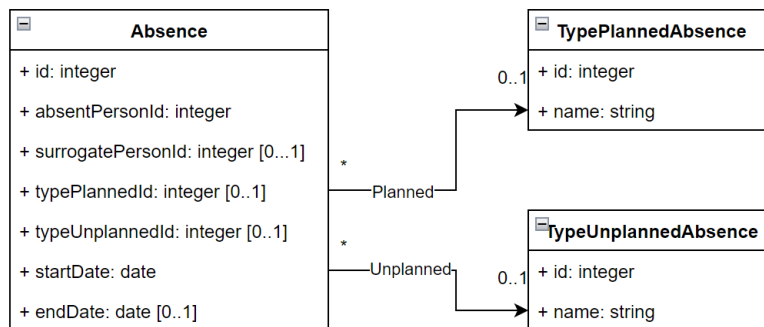
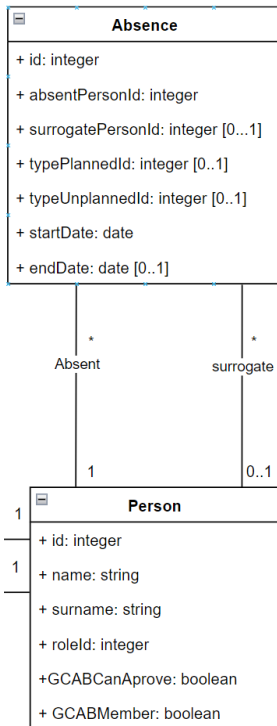


Figura 16. Relació 1 Absence

La relació 0..1..\* "Planned" indica que una absència pot tenir cap o una absència planificada però una absència planificada pot estar en varies absències.

La relació 0..1..\* "Unplanned" indica que una absència pot tenir cap o una absència no planificada però una absència no planificada pot estar en varies absències.



**Figura 17. Relació 2 Absence**

La relació 1..\* “Absent” indica que una persona pot tenir varies absències, però cada absència només correspon a una persona.

La relació 0..1..\* “surrogate” indica que una persona pot ser substituïda en varies absències, però una absència pot tenir cap o una persona substituïda. Això es degut a que no totes les absències necessiten un substituït. Només necessiten substituït aquelles persones que són responsables tècniques d’una aplicació, responsables de negoci, el responsable de canvi

### **Restriccions no representades en el diagrama**

L’atribut Id ha de ser auto incremental

Una absència ha de ser del tipus planificada o no planificada, No pot pertànyer als dos grups a la vegada ni a cap dels dos.

Una persona no pot ser substituïda en una absència si té una absència oberta o si ja esta realitzat una substitució en el mateix període de temps.

El rol de la persona per a realitzar la substitució ha de ser el de substituït d’aquella àrea de la persona a substituir

### 2.3.7 Classe TypePlannedAbsence

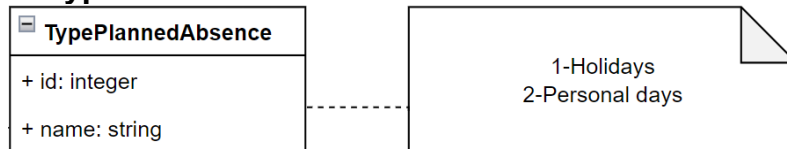


Figura 18. Classe TypePlannedAbsence

#### Definició

Representa el tipus de d'absència planificada que pot tenir una absència

#### Atributs

- Id: Per identificar el tipus d'absència
- name: nom per identificar la absència

#### Valors per defecte

Aquesta classe té uns valors assignats per defecte, segons id i nom i serà possible ampliar els rols.

Els valors per defecte són els següents

- 1- Vacances
- 2- Permisos

#### Relacions

No es contemplen relacions en aquesta classe

#### Restriccions no representades en el diagrama

En aquesta classe tenim la restricció que l'atribut id ha de ser únic i l'atribut name també.

### 2.3.8 Classe TypeUnplannedAbsence



Figura 19. Classe TypeUnplannedAbsence

#### Definició

Representa el tipus de d'absència no planificada que pot tenir una absència

#### Atributs

- Id: Per identificar el tipus d'absència
- name: nom per identificar la absència

#### Valors per defecte

Aquesta classe té uns valors assignats per defecte, segons id i nom i serà possible ampliar els rols.

Els valors per defecte són els següents

- 1- Malaltia
- 2- Acomiadament

#### Relacions

No es contemplen relacions en aquesta classe

#### Restriccions no representades en el diagrama

En aquesta classe tenim la restricció que l'atribut id ha de ser únic i l'atribut name també.

### 2.3.9 Classe Change

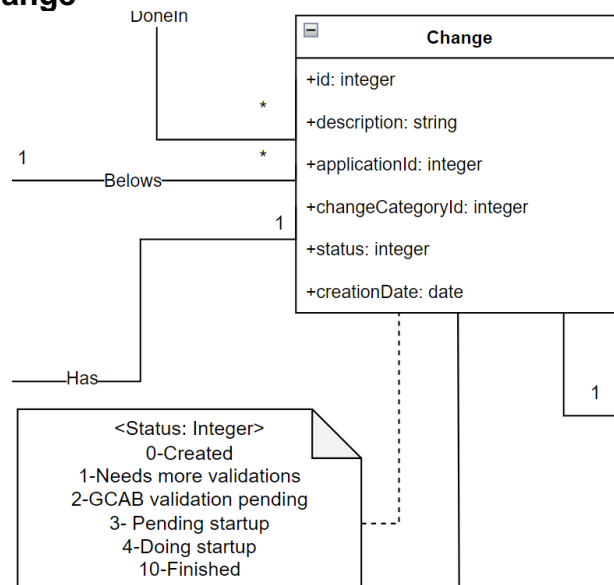


Figura 20. Classe Change

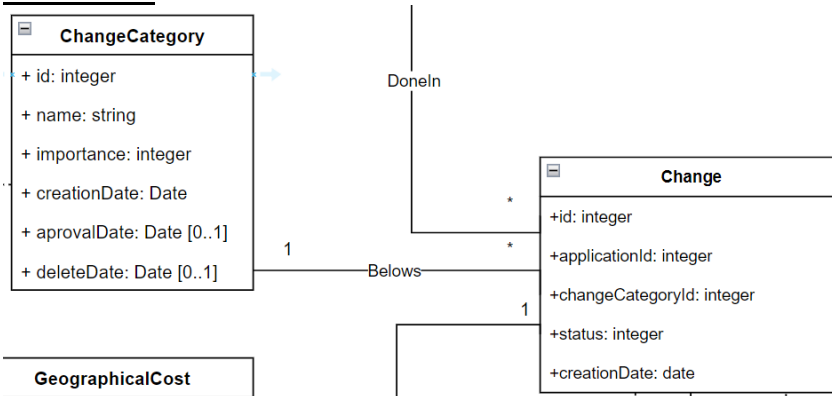
#### Definició

Representa un canvi en una aplicació

## Atributs

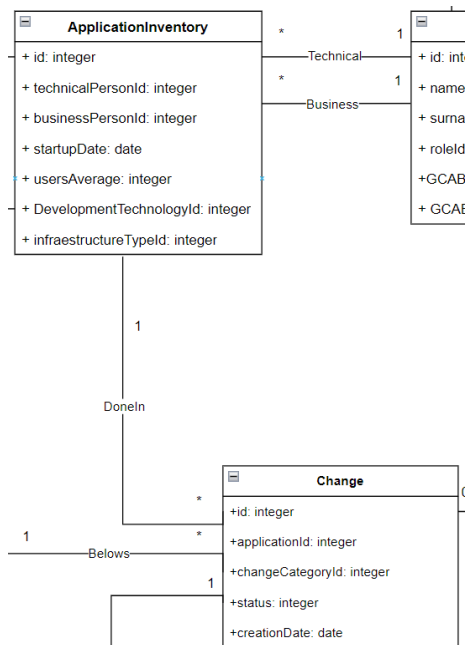
- Id: Per identificar el canvi i es auto incremental.
- Description: Per descriure el canvi que es realitza.
- ApplicationID: Identifica la id de la aplicació en la qual es realitza el canvi
- ChangeCategoryId: Identifica la id de la categoria del canvi
- status: Indica l'estat del canvi i podrà tenir els següents valors: 0-Canvi creat, 1-Validacions insuficients, 2-Pendent de validació GCAB, 3-Pendent de posta en marxa, 4-Posada en marxa en curs, 10-Finalitzat
- creationDate: Data de creació del canvi.

## Relacions



**Figura 21. Relació 1 Change**

La relació 1..\* "Belows" indica que un canvi pertany a una categoria de canvi, però una categoria pot contenir molts canvis.



**Figura 22. Relació 2 Change**

La relació 1..\* "Doneln" indica que un canvi només és pot realitzar en una aplicació, però en una aplicació es poden realitzar varis canvis.

### **Restriccions no representades en el diagrama**

L'atribut id ha de ser auto incremental.

La categoria del canvi que s'escull segons l'atribut d'importància, va directament relacionat amb el cost geogràfic, en concret amb el seu atribut de zona geogràfica, ja que un canvi amb un cost geogràfic alt, sempre tindrà més importància que un canvi que tingui un cost geogràfic baix.

La categoria del canvi ha de tenir data d'aprovació i no pot tenir data d'eliminació.

El número d'aprovacions que necessita un canvi com a mínim ha de ser, una validació d'una persona amb rol de "Technical Manager" i que estigui relacionat amb la classe "ApplicationInventari" en l'atribut "technicalPersonId". Una altre d'una persona amb rol de "Business Manager" i que estigui relacionat amb la classe "ApplicationInventari" en l'atribut "businessPersonId". Una altre d'una persona amb el rol de "ChangesManager".

Finalment cal tenir l'aprovació de totes aquelles persones que són membres de GCAB i tenen dret a vot.

L'atribut "stauts" es un identificador de canvis que s'ha de controlar, quan es crea el canvi, es crea amb valor 0(Canvi creat), quan es crea la primera validació, passa a tenir estat 2(Validacions insuficients). Quan es tenen les validacions abans comentades i l'aprovació de GCAB, passa a estat 4 (Pendent de posta en marxa). Si té les validacions però no té la validació de GCAB passa a estat 3(Pendent de validació GCAB).

Quan esta a estat 4, es pot escollir per fer-ne la posada en marxa, si s'executa passa a estat 5(Posada en marxa en curs). Quan aquesta es finalitza, s'acaba amb l'estat 10(Finalitzat).

### 2.3.10 Classe ChangeCategory

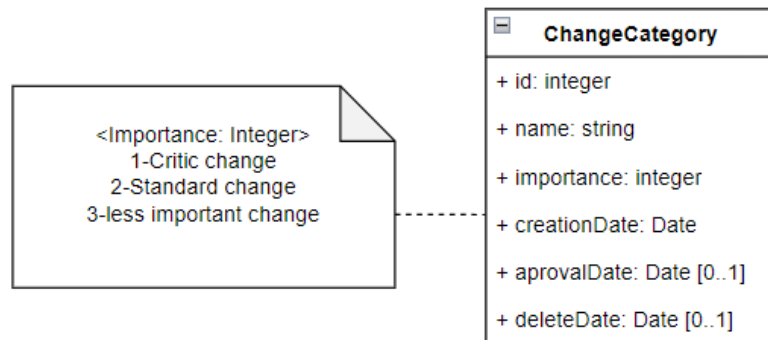


Figura 23. Classe ChangeCategory

#### Definició

Representa una categoria per el canvi de l'aplicació

#### Atributs

- Id: Per identificar el la categoria de canvi i es auto incremental
- name: nom únic que identifica la categoria del canvi
- importance: valor que identifica la importància del canvi. Per defecte tindrà els valors 1- Canvi crític, 2- Canvi estàndard, 3- canvi poc important.
- CreationDate: Data de creació de la categoria del canvi.
- AprovalDate: Data d'aprovació d'utilització de la categoria. Tindrà valor null fins que s'aprovi la categoria.
- DeleteDate: Data en que es va eliminar la categoria. Tindrà valor null fins que s'elimini la categoria.

#### Relacions

No es contempen relacions en aquesta classe

#### Restriccions no representades en el diagrama

L'atribut id i name han de ser únics

L'atribut id ha de ser auto incremental.

La data d'aprovació ha de ser igual o posterior a la data de creació.

La data d'eliminació ha de ser posterior a la data d'aprovació

### 2.3.11 Classe GeographicalCost

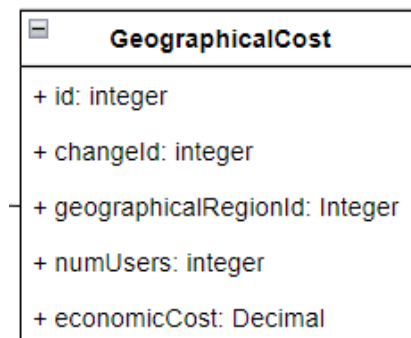


Figura 24. Classe GeographicalCost

## Definició

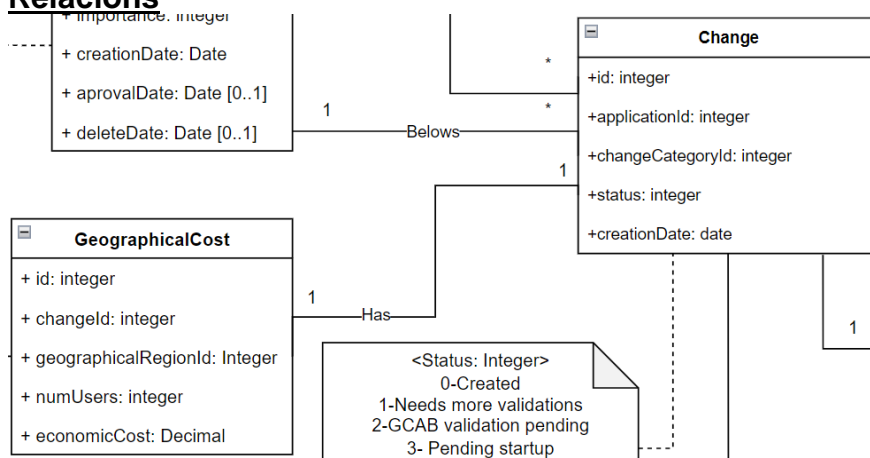
Representa el cost geografic del canvi.

S'ha escollit no incloure els atributs en la classe change sino crear una classe nova amb relació 1-1 perquè té suficient informació per identificar-la com una nova classe de domini.

## Atributs

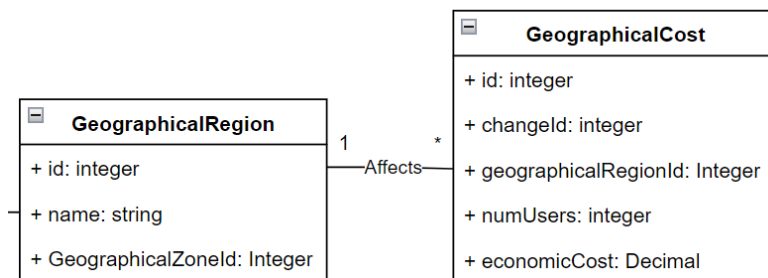
- Id: Per identificar el cost geogràfic i es auto incremental.
- Changeld: Per identificar el id del canvi al que representa.
- geographicalRegionId: Identifica la regió geogràfica d'afectació del canvi.
- NumUsers: Identifica el número d'usuaris que afecta el canvi.
- economicCost: identifica el cost en diners del canvi.

## Relacions



**Figura 25. Relació 1 GeographicalCost**

La relació 1..1 "has" indica que cada canvi té un cost geografic únic associat, aquesta relació s'ha creat perquè tot i que els atributs de "GeographicalCost" són propis de la classe change, conté prou informació per encapsular-se en una classe única i mantindre l'integritat de la base de dades.



**Figura 26. Relació 2 GeographicalCost**

La relació 1..\* "Affects" indica que un cost geografic té una regió geografica, en canvi una regió geografica, pot tenir diferents costos geografics.

## Restriccions no representades en el diagrama

L'atribut id ha de ser auto incremental



### 2.3.12 Classe GeographicalRegion

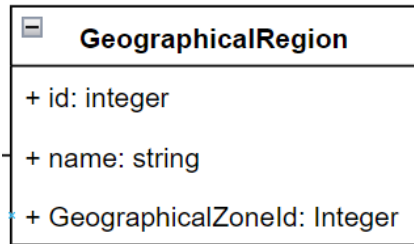


Figura 27. Classe GeographicalRegion

#### Definició

Representa la regió geogràfica a la que afecta el canvi.

#### Atributs

- Id: Per identificar la regió geogràfica i és auto incremental
- name: identifica el nom de la regió

#### Relacions

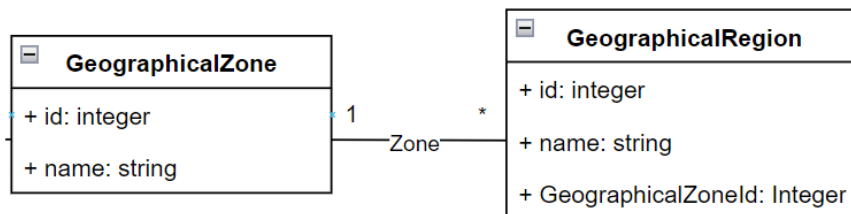


Figura 28. Relació 1..\* "Zone" GeographicalRegion

La relació 1..\* "Zone" indica que una regió geogràfica pertany a un tipus de zona, però una zona pot ser part de varies regions

#### Restriccions no representades en el diagrama

Els camps id i name han de ser únics i id ha de ser auto incremental. Només hi podrà haver una instància de zona geogràfica amb valor 3-Global. Ja que de regions i països si que té sentit que hi hagin varies instàncies, però com a zona global, no tindria sentit que hi haguessin diferents instàncies.

### 2.3.13 Classe GeographicalZone

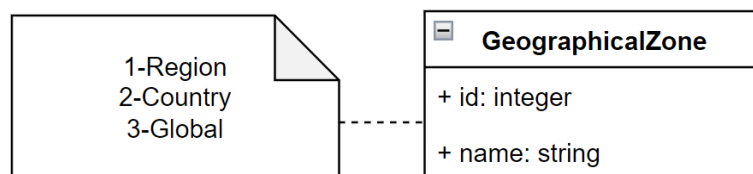


Figura 29. Classe GeographicalZone

#### Definició

Representa el tipus de zona geogràfica a la que afecta el canvi.

#### Atributs

- Id: Per identificar la zona geogràfica
- name: identifica el nom de la zona

### Valors per defecte

Aquesta classe té uns valors assignats per defecte, segons id i nom i no es té previst que es puguin ampliar ni eliminar, aquests són els següents

- 1- Regional
- 2- Estatal
- 3- Global

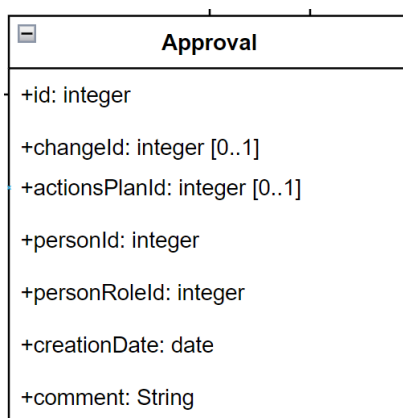
### Relacions

No es contemplen relacions en aquesta classe

### Restriccions no representades en el diagrama

Els atributs id i name han de ser únics.

#### 2.3.14 Classe Approval



**Figura 30. Classe Approval**

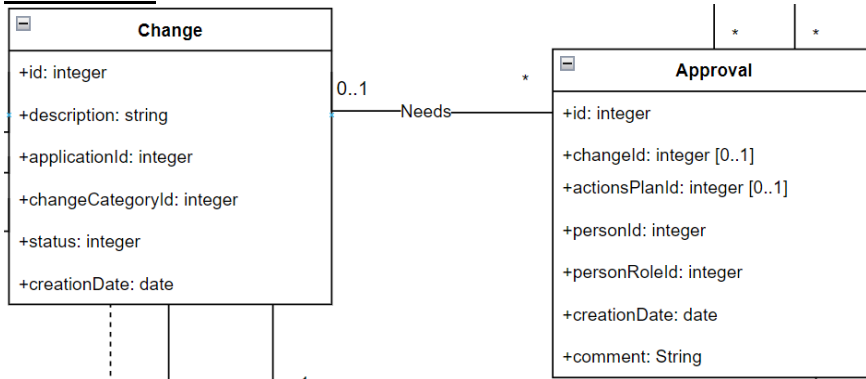
### Definició

Representa una aprovació d'un canvi o d'un pla d'accions

### Atributs

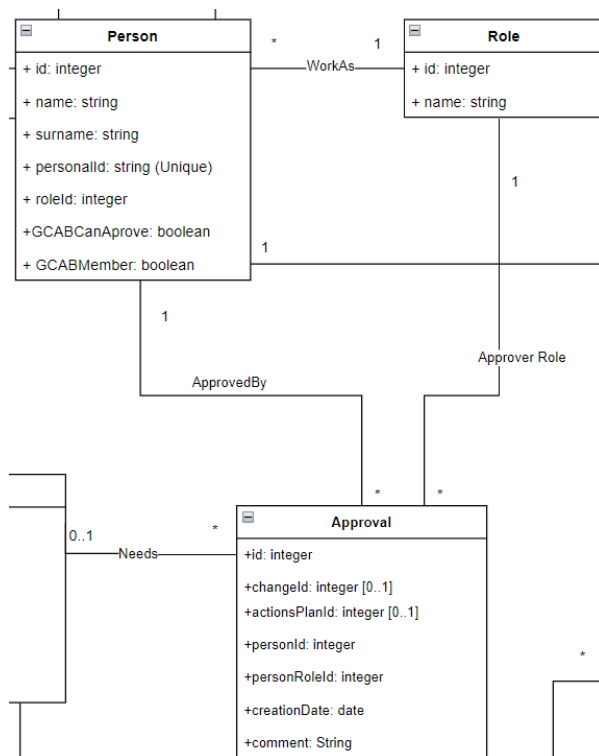
- Id: Per identificar l'aprovació i es auto incremental
- changeId: Per identificar el canvi en el qual s'ha fet la aprovació si l'aprovació és d'un canvi.
- actionsPlanId: Per identificar el pla d'accions en el qual s'ha fet la aprovació si l'aprovació és d'un pla d'accions.
- personId: Per identificar la persona que ha fet la aprovació
- personRoleId: Per identificar el rol de la persona en el moment de fer la aprovació.
- creationDate: Per identificar la data de creació de l'aprovació
- comment: Per guardar el comentari que pugui fer l'aprovador.

## Relacions



**Figura 31. Relació 1 Approval**

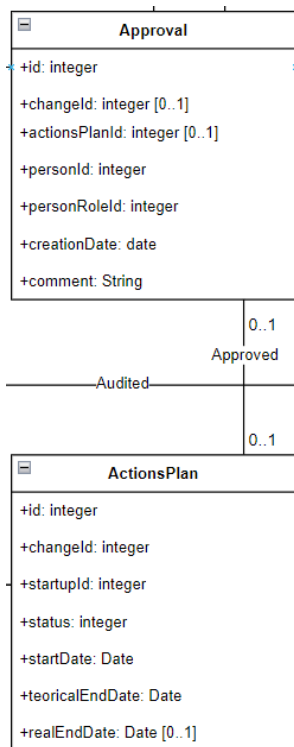
La relació 0..1..\* “Needs” indica que una aprovació sempre està relacionada amb cap o un canvi, però un canvi pot contenir varies aprovacions, el número d’aprovacions s’explica a l’apartat de restriccions no representades en el diagrama.



**Figura 32. Relació 2 Approval**

La relació 1..\* “ApprovedBy” indica una validació només pot estar feta per una persona, però una persona pot realitzar varies aprovacions

La relació 1..\* “ApproverRole” indica que una aprovació només pot contenir un rol per la persona que la genera, però un rol pot tenir moltes aprovacions.



**Figura 33. Relació 3 Approval**

La relació 0..1..0..1 “Approved” indica que una aprovació sempre esta relacionada amb cap o un pla d’accions, i un pla d’accions també pot tenir un o cap aprovació, depenen de l’estat del pla d’accions.

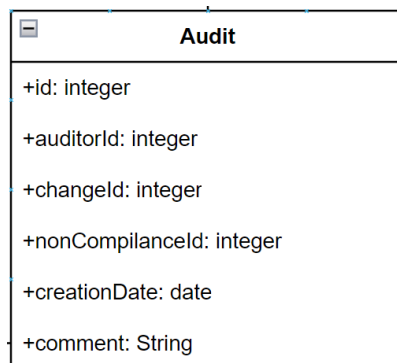
### Restriccions no representades en el diagrama

L’atribut id ha de ser únic i auto incremental

Una aprovació sempre ha de ser d’un canvi o d’un pla d’accions, no pot pertànyer als 2 de cop ni ha cap.

Una aprovació d’un pla d’accions només la pot fet un aprovador amb el rol de responsable de canvis i sempre que l’estat de pla sigui pendent d’aprovar.

### 2.3.15 Classe Audit



**Figura 34. Classe Audit**

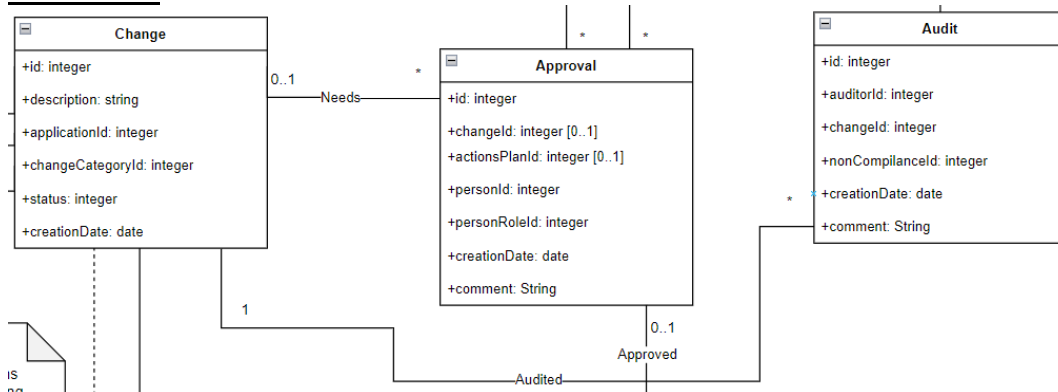
### Definició

Representa una auditoria del canvi

## Atributs

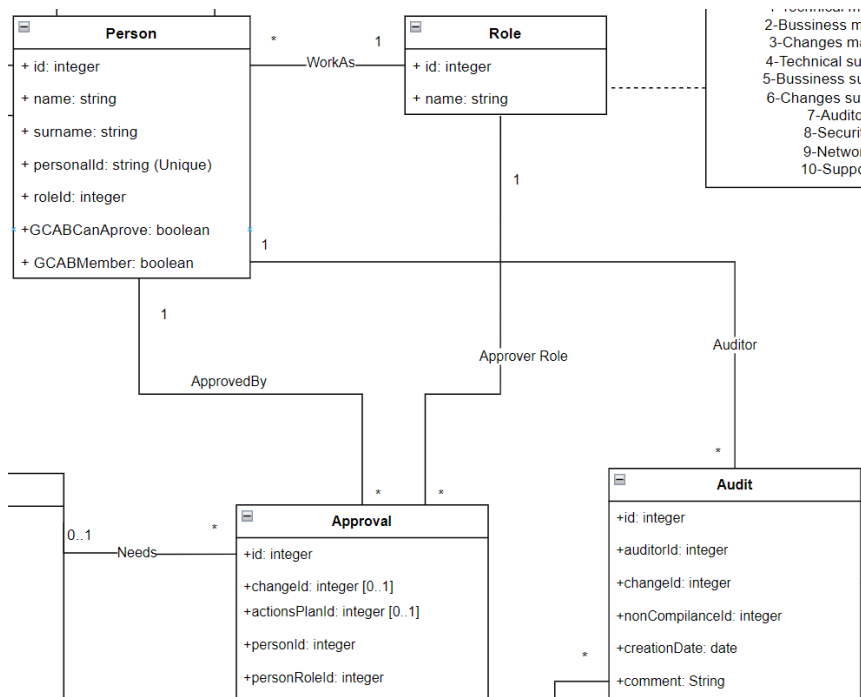
- Id: Per identificar l'auditoria i es auto incremental
- auditorId: Per identificar l'auditor
- changeld: Per identificar el canvi en el qual s'ha fet la auditoria
- nonComplanceld: Per identificar el incompliment per el qual s'ha generat l'auditoria
- creationDate: Per identificar el dia en la que s'ha realitzat la auditoria
- comment: Per guardar el comentari que pugui fer l'auditor.

## Relacions



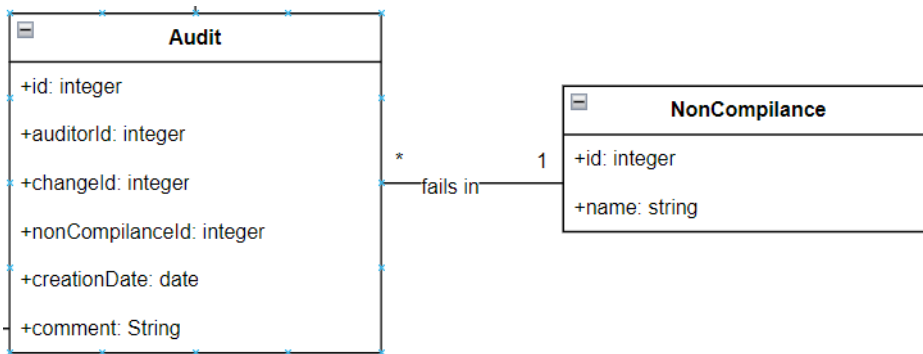
**Figura 35. Relació 1 Audit**

La relació 1..\* "audited", indica que un canvi pot tenir molts registres d'auditoria però cada registra d'auditoria, només pertany a un canvi



**Figura 36. Relació 2 Audit**

La relació 1..\* "Auditor" indica que una persona amb el rol d'auditor pot realitzar varies auditories, però cada auditoria només pot ser realitzada per una persona amb el rol d'auditor



**Figura 37. Relació 3 Audit**

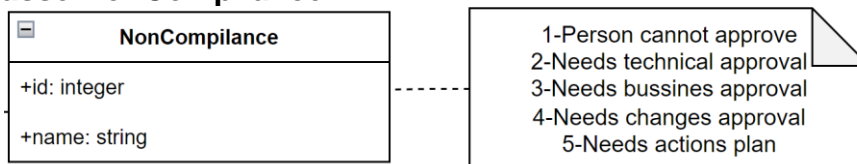
La relació 1..\* “fails in” indica una auditoria només pot contenir un incompliment, però un incompliment pot estar present en moltes auditories.

### **Restriccions no representades en el diagrama**

L'atribut id ha de ser únic i auto incremental

La persona auditora ha de tenir el rol d'auditor.

### **2.3.16 Classe NonCompliance**



**Figura 38. Classe NonCompliance**

### **Definició**

Representa un incompliment trobat en una auditoria

### **Atributs**

- Id: Per identificar l'incompliment
- Name: Per identificar el nom del incompliment i ha de ser únic

### **Valors per defecte**

Aquesta classe té uns valors assignats per defecte, segons id i nom i no es té previst que es puguin ampliar ni eliminar, aquests són els següents

- 1- La persona no tenia permisos per aprovar
- 2- Falta l'aprovació del responsable tècnic
- 3- Falta l'aprovació del responsable de negoci
- 4- Falta l'aprovació del responsable de canvis
- 5- Es necessita un pla d'accions

### **Relacions**

No es contemplen relacions en aquesta classe

### **Restriccions no representades en el diagrama**

Per aquesta classe no es contemplen restriccions d'integritat no representades en el diagrama.

### 2.3.17 Classe ChangeStartup

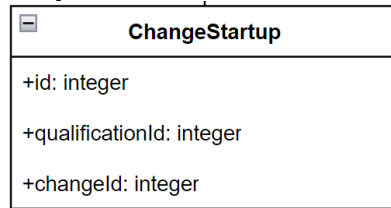


Figura 39. Classe ChangeStartup

#### Definició

Representa la posada en marxa del canvi

#### Atributs

- Id: Per identificar la posada en marxa i és auto incremental
- qualificationId: Per identificar la qualificació de com ha anat la posada en marxa
- ChangeId: Per identificar el canvi del qual s'ha realitzat la posada en marxa.

#### Relacions

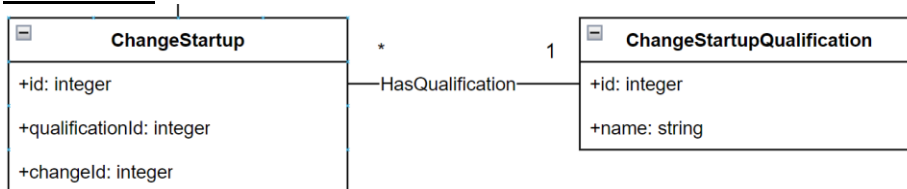
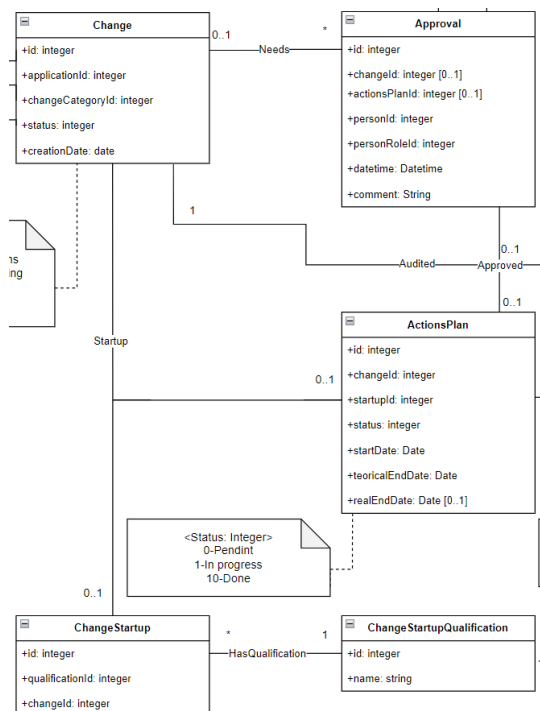


Figura 40. Relació 1 ChangeStartup

La relació 1..\* "HasQualification" indica que una posada en marxa només pot tenir una qualificació, però una qualificació pot estar present en varies posades en marxa.



**Figura 41. Relació 2 ChangeStartup**

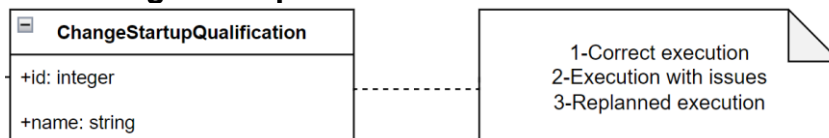
La relació 1..1 “startup”, indica que un canvi té només una posada en producció i cada posada en producció només pot contenir un canvi. D’aquesta relació, segons el valor de l’atribut “qualificatonId”, pot sorgir una instància de la classe “ActionsPlan”. Això succeeix quan “qualificatonId” té valor 2(Execució amb errors) o 3(Execució replanificada)

### **Restriccions no representades en el diagrama**

L’atribut id es auto incremental

En aquest cas, depenen del valor de l’atribut “qualificatonId” 2(Execució amb errors) o 3(Execució replanificada) sorgeix una instància de la classe “ActionsPlan”.

### **2.3.18 Classe ChangeStartupQualification**



**Figura 42. Classe ChangeStartupQualification**

### **Definició**

Representa la qualificació d’una posada en marxa.

### **Atributs**

- Id: Per identificar la qualificació
- Name: Per identificar el nom de la qualificació



### Valors per defecte

Aquesta classe té uns valors assignats per defecte, segons id i nom i no es té previst que es puguin ampliar ni eliminar, aquests són els següents

- 1- Execució correcta
- 2- Execució amb errades
- 3- Execució re planificada

### Relacions

No es contemplen relacions en aquesta classe

### Restriccions no representades en el diagrama

Els camps id i name han de ser únics

#### 2.3.19 Classe ActionsPlan

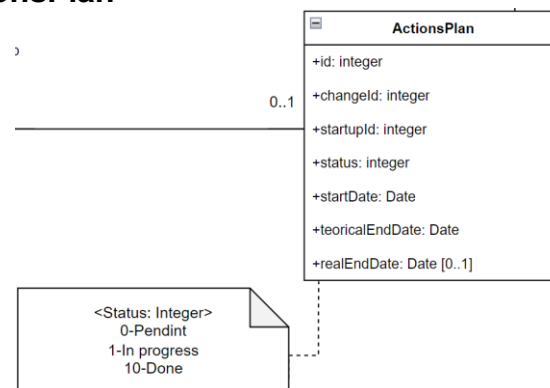


Figura 43. Classe ActionsPlan

### Definició

Representa un pla d'accions creat per una incorrecta posada en marxa d'un canvi d'una aplicació

### Atributs

- Id: Per identificar la el pla d'accions i és auto incremental
- changeID: Per identificar el canvi al qual pertany al pla
- startupId: identifica la id de la posada en marxa la qual ha generat el pla
- status: identifica l'estat del pla d'accions. Pot tenir vols com 0: Pendent, 1: En execució, 10: executat.
- startDate: Identifica l'inici del pla d'accions
- theoreticalEndDate: Identifica el fi de pla d'accions teòric
- realEndDate: Identifica el fi de pla d'accions real. Només té valor quan s'ha finalitzat.

## Relacions

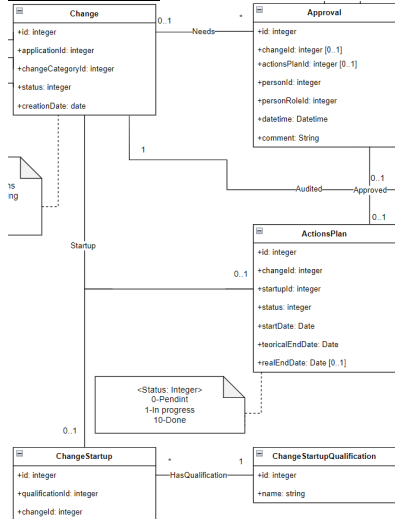


Figura 44. Relació 1 ActionsPlan

La relació 1..1 “startup”, indica que un canvi té només una posada en producció i cada posada en producció només pot contenir un canvi. D’aquesta relació, segons el valor de l’atribut “qualificatonId”, pot sorgir una instància de la classe “ActionsPlan”. Això succeeix quan “qualificatonId” té valor 2(Execució amb errors) o 3(Execució replanificada)

## Restriccions no representades en el diagrama

L’atribut id és auto incremental.

Segons l’estat de les accions que formen el pla d’accions, s’anirà actualitzat l’estat del pla d’accions.

### 2.3.20 Classe Action

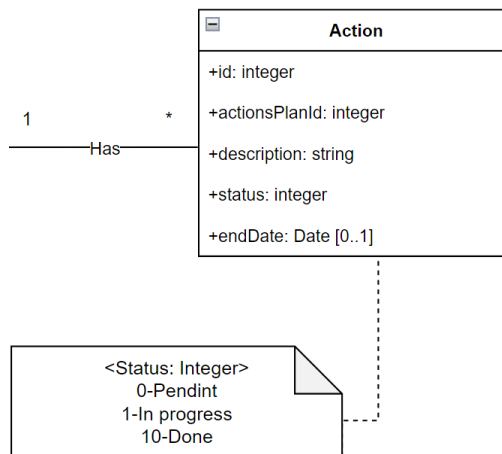


Figura 45. Classe Action

## Definició

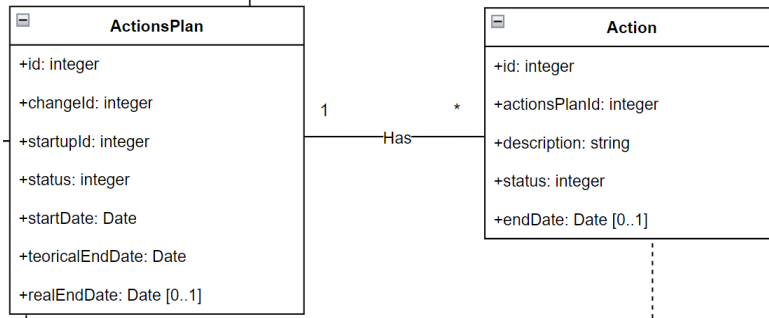
Representa una acció d’un pla d’accions

## Atributs

- Id: Per identificar la acció i és auto incremental
- actionsPlanId: Per identificar el pla al qual pertany l’acció
- description: Descripció de l’acció ha realitzat

- status: identifica l'estat de la acció. Pot tenir vols com 0: Pendent, 1: En execució, 10: executat.
- endDate: Indica la data de finalització de l'acció. Tindrà valor null fins que l'estat sigui 10-Executat

## Relacions



**Figura 46. Relació 1 Action**

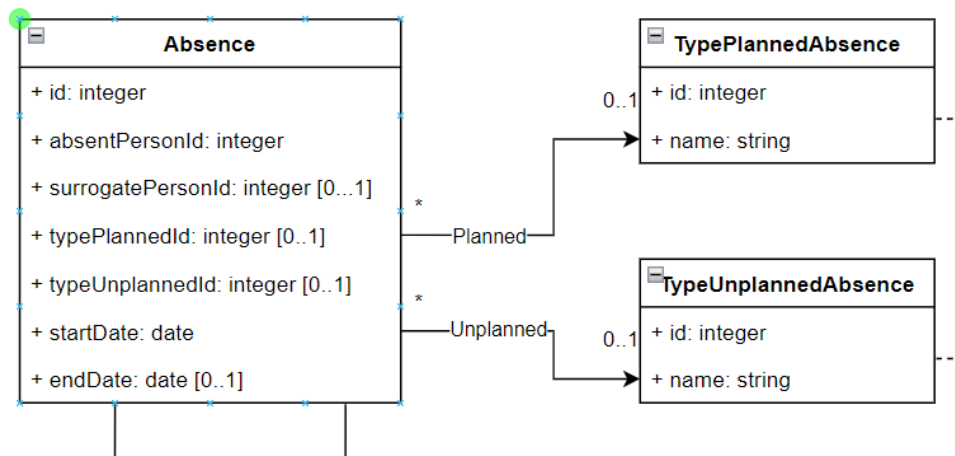
La relació 1..\* "Has" indica que una acció només pot pertanyer a un pla d'accions, però pla d'accions pot tenir varies accions.

## Restriccions no representades en el diagrama

L'atribut id és auto incremental

### 2.3.21 Decisions de disseny

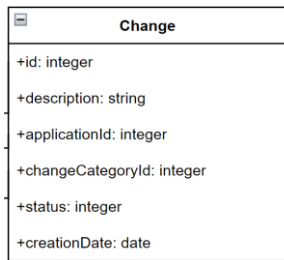
En aquest últim apartat del capítol, es definiran les decisions de disseny que s'han pres després d'analitzar els diferents problemes que s'han trobat durant el desenvolupament de la arquitectura.



**Figura 47. Decisions absencies**

Aquesta relació es podria haver fet només amb una classe D'absència i posar un atribut que fos de tipus, per tal de diferenciar si l'absència es planificada o no. En canvi s'ha escollit fer-ho d'aquesta manera amb dues classes diferents i en la classe absència els atributs de Id nullables perquè es considera que queda una estructura més clara i en cas d'augmentar el tipus d'absències, seguint aquesta estructura de classe nova i un id nou en la classe absència quedria tot millor estructurat i

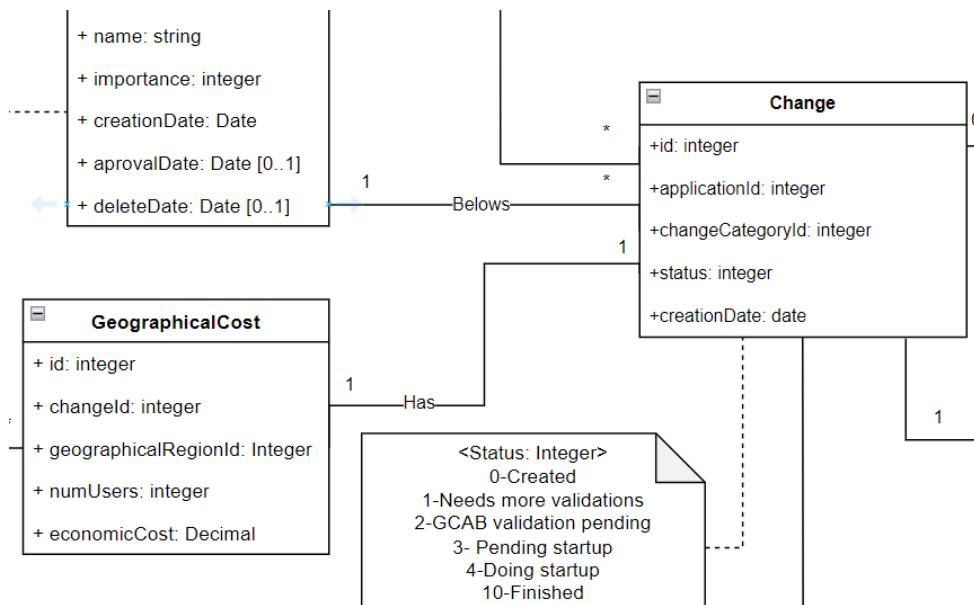
seria menys probable equivocar-se al tenir que definir una variable de tipus en cada absència diferent.



**Figura 48. Decisions Change**

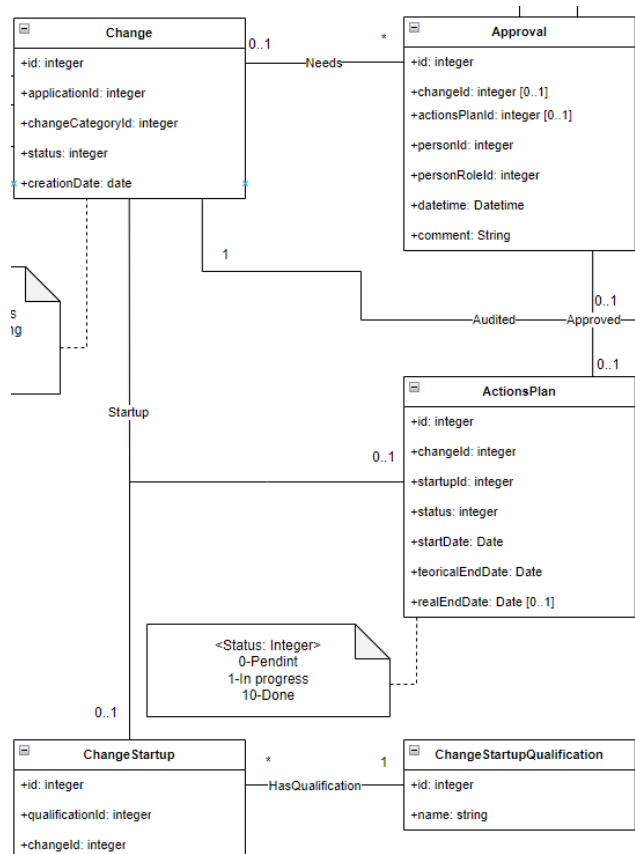
A la classe “Change” tot i que en l’enunciat no s’especifica s’ha afegit l’atribut creationDate per tal d’identificar la data en que es va crear el canvi ja per s’ha considerat que per a realitzar les consultes necessaries, es un atribut necessari.

També s’ha afegit el camp de description per a poder posar una descripció del canvi que s’esta realitzant



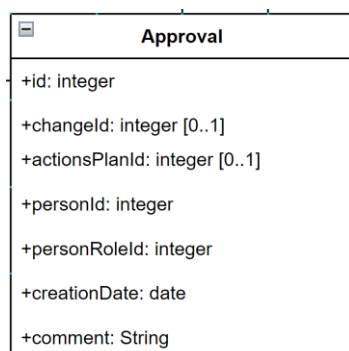
**Figura 49. Decisions GeographicalCost**

La relació “Has” entre les classes “Change” i “GeographicalCost” tot i ser 1..1 s’ha decidit posar la clau forana en la taula “GeographicalCost” ja que es dependent de la classe change i no es podria entendre sense aquesta, en canvi la classe “Change” tot i necessitar “GeographicalCost” per a tenir tota la informació necessaria s’enten per si sola.



**Figura 50. Decisions startup**

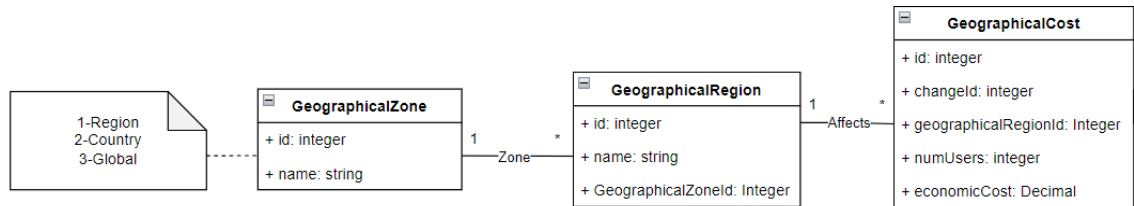
En la relació startup s'ha considerat que en la classe "Change" no es necessari tenir les claus foranes ni de "ChangeStartup" ni de "ActionsPlan" ja que la posada en marxa es realitza després de que la instància de canvi sigui creada i en aquesta si que es necessari especificar de quin canvi es tracta. En el pla d'accions com que es genera depenen de la qualificació de la posada en marxa s'hi ha inclòs tant la relació forana amb "ChangeStartup" com la de "Change", encara que aquesta si pogues accedir a través de "ChangeStartup" s'ha considerat convenient afegir-la i així tenir una relació directa entre el pla d'accions i al canvi al que correspon.



**Figura 51. Decisions Approval**

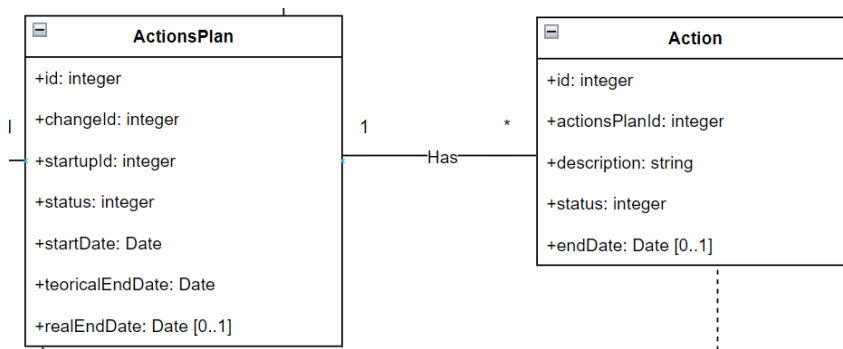
En les aprovacions, s'ha considerat posar la clau forana de canvi i del pla d'accions ja que els 2 necessiten una aprovació tot i que em necessitats diferents de tipus nullable per tal de diferenciar si una aprovació esta destinada a un canvi o al pla d'accions. En cap cas l'aprovació pot fer referència a les dues classes.

També s’ha decidit incorporar la relació forana amb la persona que fa la seva aprovació com el seu rol, ja que es considera que el rol d’una persona pot canviar durant el temps i es necessari saber quin rol tenir la persona en el moment de fer l’aprovació.



**Figura 52. Decisions GeographicalZone**

Es decideix crear les classes “GeographicalZone” “GeographicalRegion” degut a que es detecta que en les consultes s’ha de poder realitzar consultes segons regió, ja que la idea inicial era indicar en el cost geografic quin abast tenia el canvi, es a dir, regional, nacional o global, però degut a les necessitats de les consultes, s’ha creat “GeographicalRegion” per tal de poder identificar regions i paisos diferents, tot i que hi ha la restricció que per la zona global només podra existir una regió.



**Figura 53. Decisions ActionsPlan**

En la classe “ActionsPlan” s’ha determinat els atributs “teoricalEndDate” i “realEndDate” per tal de poder registrar quina es la data de fi teòrica de quan es genera i el pla i la data de fi real de quan s’han finalitzat totes les accions, aquesta és nullable ja que tindrà valor null fins que no s’hagin completat totes les accions.

Les accions també tenen un atribut “endDate” nullable per determinar quan s’han finalitzat les accions.

La intenció de fer-ho així es que quan es crea el pla d’acions, es creen totes les accions necessaries, i a mesura que es van posant en execució i finalitzat, es va actualitzant el camp de “status” de les accions i un cop es finalitza, s’inserta la data de fi i segons aquest camp de les accions i les dates de fi, s’actualitza el camp “status” del pla d’acions com la data real de fi quan s’han completat totes les accions.

## 2.4 Diagrama UML mòdul magatzem de dades “DataWarehouse”

En aquest mòdul trobem les diferents classes que formen part del magatzem de dades i recopilen la informació més important i rellevant del mòdul funcional per a poder nodrir el mòdul de consultes

### 2.4.1 Classe DWH\_Changes

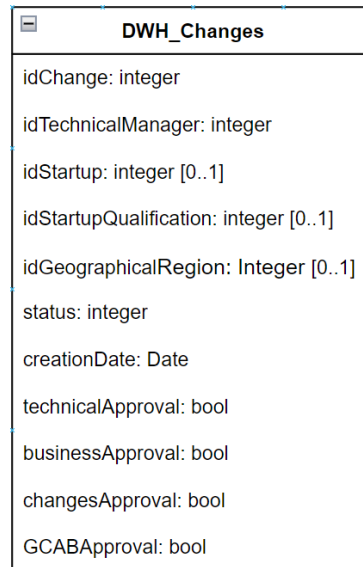


Figura 54. Classe DWH\_Changes

#### Definició

Classe que guarda la informació més rellevant dels canvis per tal de poder nodrir totes les consultes referent als canvis.

#### Atributs

- IdChange: Per identificar el canvi del qual guardem la informació
- idTechnicalManager: Per identificar el responsable tècnic del canvi
- idStartup: Per identificar la posada en marxa del canvi. Tindrà valor null fins que aquesta es realitzi
- idStartupQualification: Per identificar la qualificació de la posada en marxa del canvi. Tindrà valor null fins que es realitzi la posada en marxa.
- idGeographicalRegion: Per identificar la regió geogràfica del canvi. Tindrà valor null fins que el valor sigui entrat al sistema
- status: Per identificar l'estat del canvi
- creationDate: Per identificar la data de creació del canvi
- TechnicalApproval: Per identificar si el canvi té l'aprovació del responsable tècnic o el seu substitut
- bussinessApproval: Per identificar si el canvi té l'aprovació del responsable de negoci o el seu substitut
- changesAproval: Per identificar si el canvi té l'aprovació del responsable de canvis o del seu substitut:
- GCABApproval: Per identificar si el canvi té l'aprovació del GCAB.

## 2.4.2 Classe DWH\_ActionsPlan

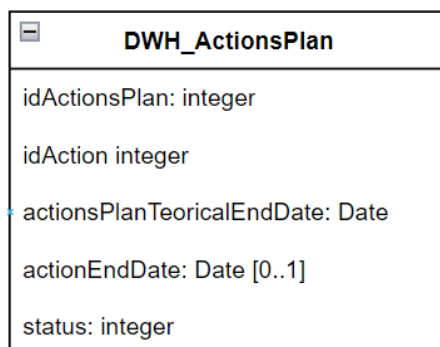


Figura 55. Classe DWH\_ActionsPlan

### Definició

Classe que guarda la informació més rellevant dels plans d'accions i les seves accions per tal de poder nodrir totes les consultes referent als plans d'accions i les accions.

### Atributs

- idActionsPlan: Per identificar el pla d'accions al que pertany la acció
- idAction: Per identificar l'acció del pla
- actionsPlanTeoricalEndDate: Per identificar la data de fi teòric del pla d'accions
- actionEndDate: Per identificar la data de fi d'una acció. Tindrà valor null fins que aquesta no s'hagi finalitzat.
- status: Per identificar l'estat de la acció

## 2.4.3 Classe DWH\_Audit

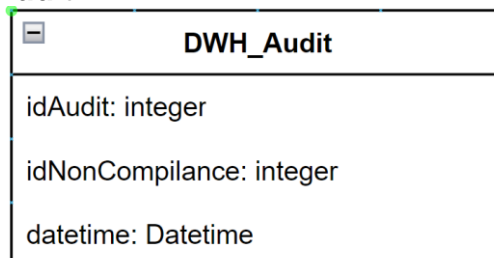


Figura 56. Classe DWH\_Audit

### Definició

Classe que guarda la informació més rellevant del les auditories i les seves no conformitats per tal de poder nodrir totes les consultes referent a les auditories i les seves no conformitats.

### Atributs

- idAudit: Per identificar l'auditoria a la que pertany la no conformitat
- idNonCompliance: Per identificar la no conformitat
- creationDate: Per identificar la data de creació



#### 2.4.4 Classe DWH\_Absence

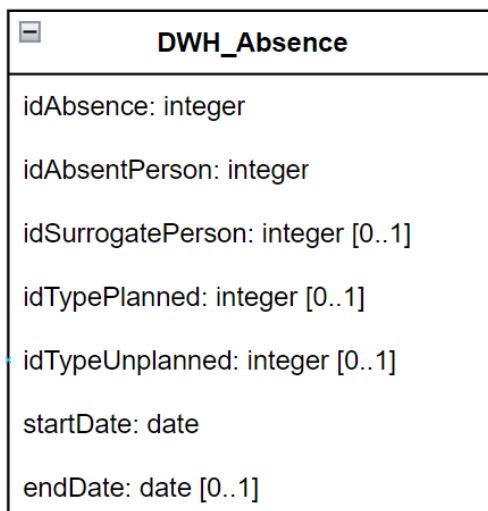


Figura 57. Classe DWH\_Absence

#### Definició

Classe que guarda la informació més rellevant del les absències per tal de poder nodrir totes les consultes referent a les absències

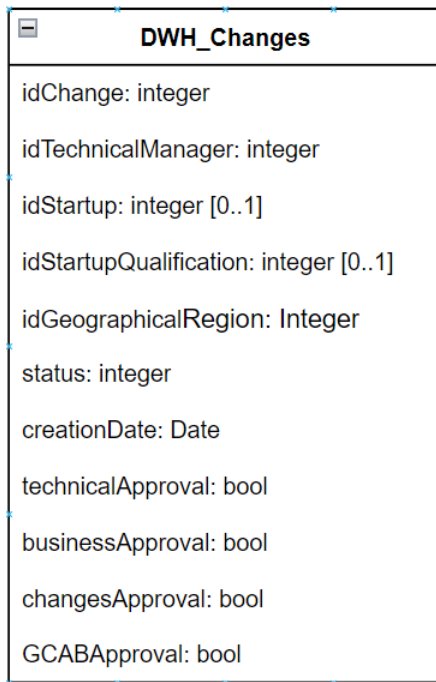
#### Atributs

- idAbsence: Per identificar l'absència
- idAbsentPerson: Per identificar la persona que genera l'absència
- idSurrogatePerson: Per identificar la persona que fa la substitució si és necessària.
- idTypePlanned: Per identificar el tipus d'absència planificada si es que és necessari
- idTypeUnplannedId: Per identificar el tipus d'absència no planificada si es que és necessari
- startDate: Per identificar la data d'inici de la absència
- endDate: Per identificar la data de fi de la absència

#### 2.4.5 Decisions de disseny

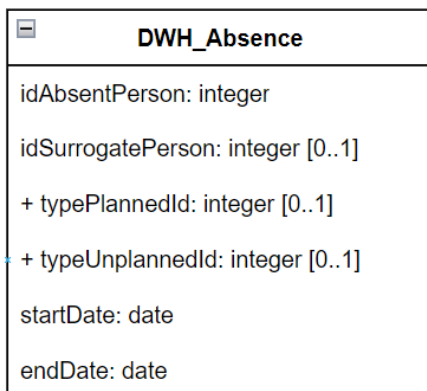
En aquest últim apartat del capítol, es definiran les decisions de disseny que s'han pres després d'analitzar els diferents problemes que s'han trobat durant el desenvolupament de la arquitectura.

S'ha decidit crear un magatzem de dades per cada classe amb les seves opcions més importants per les que es volen realitzar consultes, d'aquesta manera es podrien arribar a crear moltes consultes, sense haver de crear més magatzems de dades ni modificar-ne els atributs actuals.



**Figura 58. Decisions DWH\_Changes**

En aquest cas s'ha decidit crear atributs booleans per els diferents tipus d'aprovacions per tal de no haver d'anar consultant les aprovacions dels canvis cada cop que es fagi una modificació en aquesta classe.



**Figura 59. Decisions DWH\_Absence**

En aquest cas s'han decidit registrar totes les absències, ja siguin d'aprovadors o no, per si es necessita per les consultes, per tant s'ha registrar el substitut com a nullable, així es pot arribar a saber si aquella absència es d'aprovador o no, ja que aquests requereixen un substitut.

## 2.5 Diagrama UML mòdul consultes.

En aquest mòdul trobem les diferents classes que de les consultes que es poden fer al sistema, contenen tota la informació processada del mòdul de magatzem de dades

### 2.5.1 Classe QueryPendingChanges

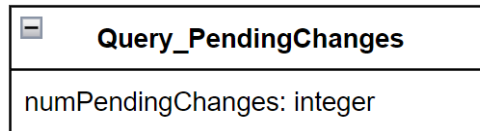


Figura 60. Classe Query\_PendingChanges

#### Definició

Classe que guarda la informació per a consultar el número de canvis pendents del sistema

#### Atributs

- numPendingChanges: Per identificar el número de canvis pendents en el sistema

### 2.5.2 Classe QueryBestTechnicalManager

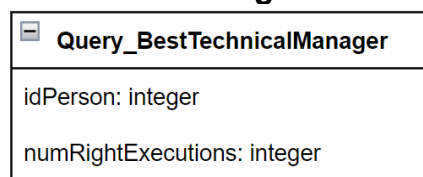


Figura 61. Classe Query\_BestTechnicalManager

#### Definició

Classe que guarda la informació per a consultar el responsable tècnic amb més execucions correctes

#### Atributs

- idPerson: Per identificar el responsable tècnic.
- NumRightExecutions: Per identificar el número d'execucions correctes del responsable.

### 2.5.3 Classe QueryOpenedChangesTechnicalManager

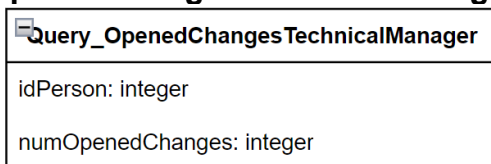


Figura 62. Classe Query\_OpenedChangesTechnicalManager

#### Definició

Classe que guarda la informació per a consultar el responsable tècnic amb més canvis en curs

#### Atributs

- idPerson: Per identificar el responsable tècnic.
- numOpenedChanges: Per identificar el número de canvis en curs del responsable.

### 2.5.4 Classe QueryFailedReplannedChangesByRegion

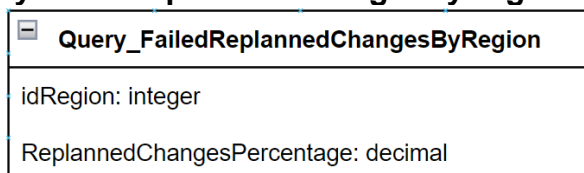


Figura 63. Classe Query\_FailedReplannedChangesByRegion

#### Definició

Classe que guarda la informació per a consultar la regió amb el percentatge de canvis re planificats més alt

#### Atributs

- regionId: Per identificar la regió
- replannedChangesPercentage: Per identificar el percentatge de canvis re planificats de la regió

### 2.5.5 Classe QueryOpenedActionsPlan

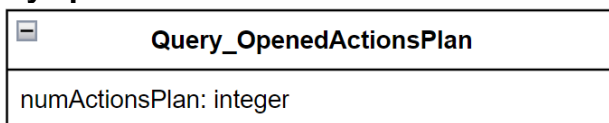


Figura 64. Classe Query\_OpenendActionsPlan

#### Definició

Classe que guarda la informació per a consultar el número de plans d'accions oberts en el sistema

#### Atributs

- numActionsPlan: Per identificar el número de plans d'acció oberts en el sistema

### 2.5.6 Classe QueryMostUnplannedReplacedAprover

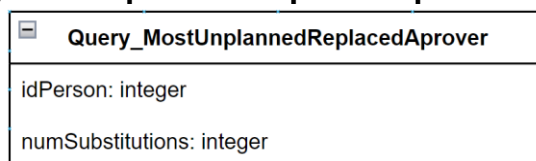


Figura 65. Classe Query\_MostUnplannedReplacedAprover

#### Definició

Classe que guarda la informació del aprovador que ha set substituït més cops per absències no planificades

#### Atributs

- idPerson: Per identificar l'aprovador.
- numSubstitutions: Per identificar el número de vegades que ha set substituït per absències no planificades.

### 2.5.7 Classe QueryFailedExecutionLastYear

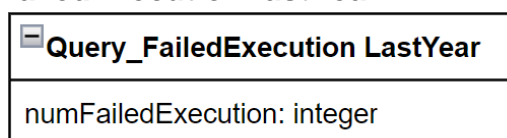


Figura 66. Classe Query\_FailedExecutionLastYear

#### Definició

Classe que guarda la informació del número d'execucions fallides en l'últim any complet.

#### Atributs

- numFailedExecution: Per identificar el número d'execucions fallides l'últim any.

### 2.5.8 Classe QueryActionsPercentagePerYear

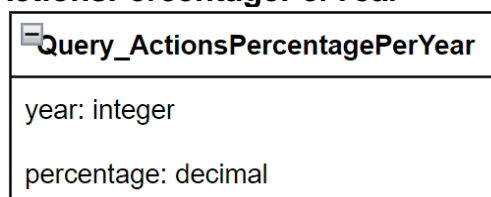


Figura 67. Classe Query\_ActionsPercentagePerYear

#### Definició

Classe que guarda la informació del percentatge d'accions definides per execucions no correctes tancades en el temps determinat per el seu pla d'accions en un any concret.

#### Atributs

- year: Per identificar l'any al que correspon el percentatge.
- Percentatge: Per identificar el percentatge d'accions no correctes en el temps determinat per el seu pla d'accions.

### 2.5.9 Classe QueryApprovedChangesByGCABCurrentYear

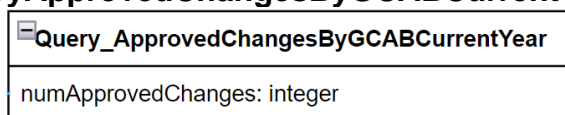


Figura 68. Classe Query\_AprovedChangesByGCABCurrentYear

#### Definició

Classe que guarda la informació del número de canvis aprovats per GCAB en el any actual.

#### Atributs

- numApprovedChanges: Per identificar el número de canvis aprovats per GCAB

### 2.5.10 Classe QueryNonComplianceLastSixMonths

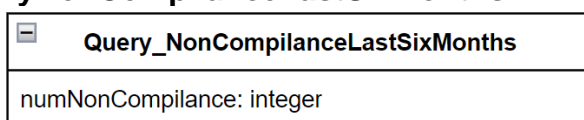


Figura 69. Classe Query\_NonComplianceLastSixMonths

#### Definició

Classe que guarda el número de no conformitats en els últims sis mesos

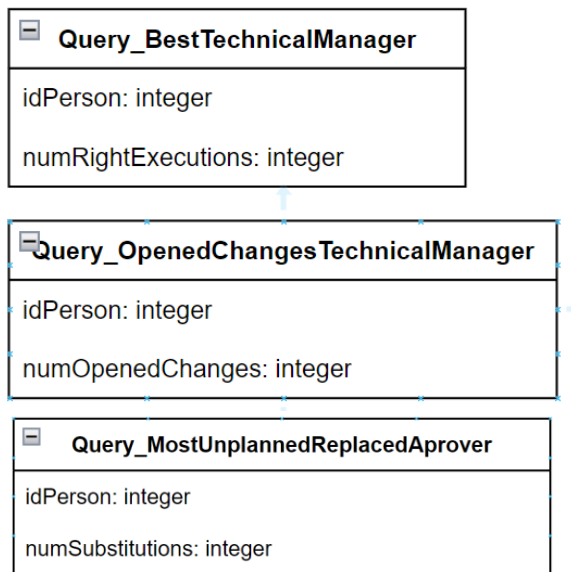
#### Atributs

- numNonCompliance: Per identificar el número de no conformitats en els últims 6 mesos.

### 2.5.11 Decisions de disseny

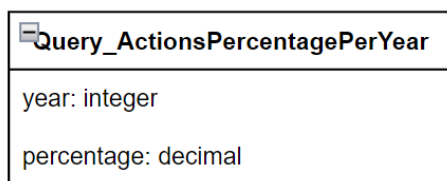
En aquest últim apartat del capítol, es definiran les decisions de disseny que s'han pres després d'analitzar els diferents problemes que s'han trobat durant el desenvolupament de la arquitectura.

En aquest mòdul s'ha pensat per tal de crear tantes taules com consultes específiques es necessitin. La informació d'aquesta consultes s'obté del magatzem de dades, que per exemple guarda molta informació dels canvis, en canvi no hi ha una sola consulta de canvis, sino que n'hi ha varies que consulten certa informació específica.



**Figura 70. Decisions disseny consultes 1**

En les consultes en que es vol saber la persona que és millor en quelcom o que té el màxim número d'alguna fet, s'ha decidit registrar apart del Identificar, un camp número amb el número per tal de poder quantificar magnituds.



**Figura 71. Decisions disseny consultes 2**

En aquesta consulta es registra l'any i el percentatge, de tal manera que si es vol consultar un any en concret, caldrà utilitzar la clausula "Where".

## 2.6 Disseny implementacions ABM

En el següent apartat es defineixen els procediments IBM (Alta, baixa i modificació) de les diferents classes del disseny. Algunes d'elles només tindran procediments d'alta i modificació, ja que no es podran eliminar registres en el sistema per a poder seguir una correcta traçabilitat d'aquest. Els procediments de baixa, no els eliminaran, sinó que els modificaran, deixant-los en un estat d'inactivitat en el sistema. S'explicarà cada cas en concret.

### 2.6.1 Procediments ApplicationInventory

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació dels canvis en el sistema de l'aplicació que s'elimini.

**Procediment:** Alta

**Nom:** CreateNewApplication

**Descripció:** Procediment per crear una nova aplicació, cal introduir tots els paràmetres obligatoris per a crear l'aplicació.

**Paràmetres obligatoris:**

- AppName: Nom per identificar la aplicació
- AppTechnicalPersonId: Identificador per el responsable tècnic de l'aplicació
- AppBusinessPersonId: Identificador per el responsable de negoci de l'aplicació
- AppStartupDate: Data de posada en marxa de l'aplicació.
- AppUsersAverage: Mitjana d'usuaris que utilitzen l'aplicació
- AppDevelopmentTechnologyId: Identificador de la tecnologia de desenvolupament
- AppInfrastructureTypeId: Identificador de la infraestructura de l'aplicació.

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

[2]Referencia bibliografia sobre procediments ABM

*Altas, bajas y modificaciones en base de datos* [en línia] [Consulta 28 de Octubre de 2023] Disponible a : <https://disenowebakus.net/altas-bajas-y-modificaciones-en-bases-de-datos-mysql-phpmyadmin-php.php>



**Procediment:** Modificació

**Nom:** ModifyApplication

**Descripció:** Procediment per a modificar una aplicació. L'únic camp que té sentit poder actualitzar és "usersAverage" ja que tots els altres són camps necessaris per la seva creació i no poden ser modificables per no deixar dades incorrectes en el sistema o que consultes posteriors puguin tenir valors erronis.

**Paràmetres obligatoris:**

- id: identificador de la aplicació a modificar
- usersAverage: Mitjana d'usuaris que utilitzen l'aplicació

Paràmetres de retorn:

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### 2.6.2 Procediments DevelopmentTechnology

En aquesta classe es preveu només el paràmetre d'alta, un cop inserit no es permetrà ni modificar ni eliminar

**Procediment:** Alta

**Nom:** CreateNewDevelopmentTechnology

**Descripció:** Procediment per crear una nova tecnologia de desenvolupament, cal introduir tots els paràmetres obligatoris per a crear l'aplicació.

**Paràmetres obligatoris:**

- Identifier: Per identificar la tecnologia
- TechnologyName: Nom per identificar la tecnologia

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### 2.6.3 Procediments InfrastructureType

En aquesta classe es preveu només el paràmetre d'alta, un cop inserit no es permetrà ni modificar ni eliminar

**Procediment:** Alta

**Nom:** CreateNewInfrastructureType

**Descripció:** Procediment per crear un nou tipus d'infraestructura, cal introduir tots els paràmetres obligatoris per a crear l'aplicació.

**Paràmetres obligatoris:**

- Identifier: Per identificar la infraestructura
- InfrastructureName: Nom per identificar la infraestructura

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

#### 2.6.4 Procediments Person

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació de la persona que ha realitzat alguna acció sobre el sistema

En el cas de les modificacions, es preveuen diferents procediments.

**Procediment:** Alta

**Nom:** CreateNewPerosn

**Descripció:** Procediment per crear una nova persona, cal introduir tots els paràmetres obligatoris per a crear l'aplicació.

**Paràmetres obligatoris:**

- PersonName: Nom per identificar el nom de la persona
- PersonSurname: Cognom per identificar el cognom de la persona
- PersonPersonalId: Identificador personal i únic de la persona (DNI)
- PersonRoleId: Identificador per el rol de la persona
- PersonGCABCanAprove: Indica que té dret a vot en el GCAB
- PersonGCABMember: Indica que és membre actual de GCAB

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyPersonCCABCanApprove

**Descripció:** Procediment per a modificar el dret a vot d'un membre de GCAB. Permetra indicar si una persona té dret a vot o no, sempre que sigui membre de GCAB.

**Paràmetres obligatoris:**

- PersonId: identificador de la persona en la que es fa la modificació
- PersonGCABCanAprove: Indica que té dret a vot en el GCAB

Paràmetres de retorn:

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyPersonCCABMember

**Descripció:** Procediment per a indicar si una persona es membre de GCAB. Permetrà indicar si una persona es membre de GCAB, si deixa de ser membre, i tenia dret a vot, aquest serà revocat.

**Paràmetres obligatoris:**

- PersonId: identificador de la persona en la que es fa la modificació
- PersonGCABMember: Indica que és membre actual de GCAB

Paràmetres de retorn:

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### 2.6.5 Procediments Role

En aquesta classe es preveu només el paràmetre d'alta, un cop inserit no es permetrà ni modificar ni eliminar

**Procediment:** Alta

**Nom:** CreateNewRole

**Descripció:** Procediment per crear un nou rol per una persona, cal introduir tots els paràmetres obligatoris per a crear-lo

**Paràmetres obligatoris:**

- identifier: Identificador del Rol
- roleName: Nom per identificar el rol

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

### 2.6.6 Procediments Absence

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació de les absències de les persones. En el cas de les modificacions, es preveuen diferents procediments.

**Procediment:** Alta

**Nom:** CreateAbsence

**Descripció:** Procediment per crear una nova absència, cal introduir tots els paràmetres obligatoris per a crear l'absència i els opcionals si es possible.

Els paràmetres opcionals de tipus d'absència, es obligatori que només un d'ells sigui seleccionat alhora de crear l'absència

**Paràmetres obligatoris:**

- AbsentPerson: Identificador de la persona que genera l'absència
- AbsenceStartDate: Data en la que comença l'absència

**Paràmetres opcionals:**

- SurrogatePerson: Identificador de la persona que fa la substitució, Aquest paràmetre s'utilitzarà si es coneix la persona en el moment de crear l'absència i també si aquella absència necessita un substitut.
- TypePlanned: Identificador del tipus d'absència planificada.
- TypeUnplanned: Identificador del tipus d'absència no planificada.
- AbsenceEndDate: Data de fi d'absència, en el cas de que en el moment de registrar l'absència en el sistema, aquesta ja hagi finalitzat.

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyAbsenceSurrogatePerson

**Descripció:** Procediment per a modificar la persona substituïda, en cas de que no es conegués quan s'ha creat l'absència o que aquesta hagi hagut de canviar.

**Paràmetres obligatoris:**

- ModifyAbsenceId: Identificador de la absència que es vol modificar
- ModifySurrogatePerson: Identificador de la persona que fa la substitució

Paràmetres de retorn:

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyAbsenceEndDate

**Descripció:** Procediment per actualitzar la data de fi de la absència.

**Paràmetres obligatoris:**

- ModifyAbsenceId: Identificador de la absència que es vol modificar
- ModifyEndDate: Indica la data de fi de la absència

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### 2.6.7 Procediments TypePlannedAbsence

En aquesta classe es preveu només el paràmetre d'alta, un cop inserit no es permetrà ni modificar ni eliminar

**Procediment:** Alta

**Nom:** CreateNewTypePlannedAbsence

**Descripció:** Procediment per crear un nou tipus d'absència planificada, cal introduir tots els paràmetres obligatoris per a crear-la.

**Paràmetres obligatoris:**

- identifier: Identificador del tipus d'absència
- TypeName: Nom per identificar el tipus d'absència

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

### 2.6.8 Procediments TypeUnplannedAbsence

En aquesta classe es preveu només el paràmetre d'alta, un cop inserit no es permetrà ni modificar ni eliminar

**Procediment:** Alta

**Nom:** CreateNewTypeUnplannedAbsence

**Descripció:** Procediment per crear un nou tipus d'absència no planificada, cal introduir tots els paràmetres obligatoris per a crear-la.

**Paràmetres obligatoris:**

- identifier: Identificador del tipus d'absència
- TypeName: Nom per identificar el tipus d'absència

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

### 2.6.9 Procediments Change

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació dels canvis en el sistema

**Procediment:** Alta

**Nom:** CreateChange

**Descripció:** Procediment per crear un nou canvi, cal introduir tots els paràmetres obligatoris per a crear el canvi

**Paràmetres obligatoris:**

- ChangeDescription: Descripció del canvi
- ChangeApplication: Identificador de la aplicació en la que es fa el canvi
- ChangeCategory: identificador de la categoria del canvi
- status: Estat del canvi. El valor en la alta ha de ser 0 (Creat)
- ChangeCreationDate: Data de creació del canvi

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyChangeStatus

**Descripció:** Procediment per a modificar el estat d'un canvi

**Paràmetres obligatoris:**

- ChangeId: Identificador del canvi a modificar
- ChangeStatus: Estat del canvi.

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

### 2.6.10 Procediments ChangeCategory

En aquesta classe es preveuen els procediments d'alta i modificació i baixa. El de baixa no eliminarà registres.

**Procediment:** Alta

**Nom:** CreateNewChangeCategory

**Descripció:** Procediment per crear una nova categoria de canvi, cal introduir tots els paràmetres obligatoris per a crear la categoria. També si es possible, cal introduir els paràmetres opcionals.

**Paràmetres obligatoris:**

- CategoryName: nom de la categoria
- CategoryImportance: identifica la importància de de la categoria del canvi
- CategoryCreationDate: Data de creació del canvi

**Paràmetres opcionals:**

- CategoryApprovalDate: Data d'aprovació per utilitzar la categoria.

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyChangeCategoryApprovalDate

**Descripció:** Procediment per a modificar la data d'aprovació de la categoria del canvi.

**Paràmetres obligatoris:**

- CategoryId: identificador de la categoria del canvi
- ModifyApprovalDate: Data d'aprovació de la categoria del canvi

Paràmetres de retorn:

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Baixa

**Nom:** DeleteChangeCategory

**Descripció:** Procediment per a eliminar una categoria de canvi

**Paràmetres obligatoris:**

- CategoryId: identificador de la categoria del canvi
- ModifyDeleteDate: Data d'eliminació de la categoria del canvi.

Paràmetres de retorn:

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

### 2.6.11 Procediments GeographicalCost

En aquesta classe es preveuen el procediment d'alta. No es permetrà ni modificar informació ni eliminar registres ja que aquesta informació es genera al crear un canvi i no pot ser modificable

**Procediment:** Alta

**Nom:** CreateGeographicalCost

**Descripció:** Procediment per crear un nou cost geogràfic, cal introduir tots els paràmetres obligatoris per a crear el cost.

**Paràmetres obligatoris:**

- Change: identificador del canvi
- Region: identificador de la regió geogràfica
- Users: Número d'usuaris als que pot afectar el canvi
- ChangeCost: Cost econòmic del canvi

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

### **2.6.12 Procediments GeographicalRegion**

En aquesta classe es preveuen el procediment d'alta. No es permetrà ni modificar informació ni eliminar registres ja que no hi ha cap atribut que pugui canviar en el temps

**Procediment:** Alta

**Nom:** CreateGeographicalRegion

**Descripció:** Procediment per crear una nova regió geogràfica, cal introduir tots els paràmetres obligatoris per a crear el cost.

**Paràmetres obligatoris:**

- RegionalName: nom per identificar la regió
- RegionalZone: identificador de la zona geogràfica

**Paràmetres de retorn:**

“OK”- Si la funció s'executa correctament

“ERROR + Descripció” Si hi ha algun error o algun paràmetre incorrecte

### **2.6.13 Procediments GeographicalZone**

En aquesta classe no es permetrà realitzar altes, modificacions ni baixes. Amb els paràmetres creats per defecte és suficient.



## 2.6.14 Procediments Approval

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació de les aprovacions en el sistema

**Procediment:** Alta

**Nom:** CreateApproval

**Descripció:** Procediment per crear una nova aprovació, cal introduir tots els paràmetres obligatoris per a crear el canvi i els opcionals si es possible. En aquest cas cal introduir un dels dos opcionals, és a dir, cal indicar si l'aprovació es d'un canvi o un pla d'accions.

**Paràmetres obligatoris:**

- Approver: Identificador del aprovador
- ApproverRole: Identificador del rol del aprovador
- ApprovalDate: data de l'aprovació
- ApprovalComment: comentari de la aprovació (Pot ser un text buit)

**Paràmetres opcionals:**

- Change: Identificador del canvi en que es fa l'aprovació
- ActionsPlan: Identificador de pla d'accions en que es fa l'aprovació

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyApprovalComment

**Descripció:** Procediment per a modificar el comentari d'una aprovació

**Paràmetres obligatoris:**

- Approval: Identificador de la aprovació
- Approval comment: comentari a modificar

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### 2.6.15 Procediments Audit

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació de les auditories en el sistema

**Procediment:** Alta

**Nom:** CreateAudit

**Descripció:** Procediment per crear una nova auditoria, cal introduir tots els paràmetres obligatoris

**Paràmetres obligatoris:**

- Auditor: Identificador del auditor
- Change: Identificador del canvi que s'audita
- NonCompliance: identificador de la no conformitat
- AuditDate: Data de l'auditoria
- AuditComment: Comentaris del auditor

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyAuditComment

**Descripció:** Procediment per a modificar el comentari d'una auditoria

**Paràmetres obligatoris:**

- id: Identificador de la auditoria
- comment: comentari a modificar

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### 2.6.16 Procediments NonCompliance

En aquesta classe es preveu només el paràmetre d'alta, un cop inserit no es permetrà ni modificar ni eliminar

**Procediment:** Alta

**Nom:** CreateNewNonCompliance

**Descripció:** Procediment per crear una nova no conformitat, cal introduir tots els paràmetres obligatoris per a crear-lo

**Paràmetres obligatoris:**

- NonComplianceId: Identificador de la no conformitat
- NonComplianceName: Nom per identificar la no conformitat.

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### **2.6.17 Procediments ChangeStartup**

En aquesta classe es preveu només el paràmetre d'alta, un cop inserit no es permetrà ni modificar ni eliminar

**Procediment:** Alta

**Nom:** CreateNewChangeStartup

**Descripció:** Procediment per crear una nova posada en marxa del canvi, cal introduir tots els paràmetres obligatoris per a crear-lo

**Paràmetres obligatoris:**

- Change: Identificador del canvi del qual es fa la posada en marxa
- Qualification: Identificador de la qualificació de la posada en marxa

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

### **2.6.18 Procediments ChangeStartupQualification**

En aquesta classe no es permetrà crear, modificar o eliminar cap registre. Només es necessiten els que es creen per defecte.

### 2.6.19 Procediments ActionsPlan

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació dels plans d'accions del sistema

**Procediment:** Alta

**Nom:** CreateActionsPlan

**Descripció:** Procediment per crear un nou pla d'accions, cal introduir tots els paràmetres obligatoris per a crear el canvi i els opcionals si es possible.

**Paràmetres obligatoris:**

- Change: Identificador del canvi en el que es fa el pla d'accions
- Startup: Identificador de la posada en marxa del pla d'accions
- status: estat del pla d'accions. Per defecte tindrà valor 0-Pendent.
- ActionsPlanStartDate: Data d'inici del pla d'accions
- ActionsPlanTeoricalEndDate; Data de fi teòrica del pla d'accions

**Paràmetres opcionals:**

- ActionsPlanRealEndDate: Data real de fi del pla d'accions

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyStatus

**Descripció:** Procediment per a modificar l'estat del pla d'accions

**Paràmetres obligatoris:**

- ActionsPlanId: Identificador del pla d'accions
- ActionsPlanStatus: estat del pla d'accions

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** ModifyRealEndDate

**Descripció:** Procediment per a modificar la data real de fi del pla d'accions

**Paràmetres obligatoris:**

- ActionsPlanId: Identificador del pla d'accions
- ActionsPlanRealEndDate: Data real de fi del pla d'accions

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

## 2.6.20 Procediments Action

En aquesta classe es preveuen els procediments d'alta i modificació, el de baixa no, degut a que es perdria tota la informació de les accions dels plans del sistema

**Procediment:** Alta

**Nom:** createAction

**Descripció:** Procediment per crear una nova acció, cal introduir tots els paràmetres obligatoris per a crear el canvi i els opcionals si es possible.

**Paràmetres obligatoris:**

- ActionPlan: Identificador del pla d'accions al que pertany
- ActionDescription: descripció de la acció
- ActionStatus: estat de la acció. Per defecte tindrà valor 0-Pendent.

**Paràmetres opcionals:**

- ActionEndDate: Data real del fi de la acció

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** modifyActionStatus

**Descripció:** Procediment per a modificar l'estat de la acció

**Paràmetres obligatoris:**

- ActionId: Identificador de la acció
- ActionStatus: estat de la acció

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

**Procediment:** Modificació

**Nom:** modifyActionEndDate

**Descripció:** Procediment per a modificar la data real de fi de la acció

**Paràmetres obligatoris:**

- ActionId: Identificador de la acció
- ActionEndDate: Data de fi de la acció

**Paràmetres de retorn:**

"OK"- Si la funció s'executa correctament

"ERROR + Descripció" Si hi ha algun error o algun paràmetre incorrecte

## 2.6.21 Altres procediments.

S'han desenvolupat procediments auxiliars per a reutilitzar el codi i no repetir el mateix en moltes funcions com:

- Aux\_CheckPersonExists: Donada una id de persona, retorna si aquesta existeix en el sistema o no mitjançant una resposta booleana
- Aux\_InsertLog: Registra una entrada en la taula de log.

## 2.7 Disseny i implementació de les funcions disparadors per magatzem de dades

En aquest apartat definirem i explicarem les funcions que s'utilitzen en els disparadors "triggers" per tal d'alimentar el magatzem de dades des de el mòdul funcional.

### 2.7.1 Funció Maintenance\_DWH\_Changes

Aquesta funció es la encarregada d'actualitzar la taula del magatzem de dades "DWH\_Changes" i s'utilitza a través de disparadors en les diferents taules que ens generen la informació per aquesta taula, en els processos d'inserció i actualització. Les taules que tenen els disparador són: "Changes", "GeographicalCost", "Approval" i "ChangeStartup".

S'utilitza els paràmetres de la funció del disparador per indicar a la funció quina taula el crida i així poder identificar el canvi el qual s'esta inserint o modificant.

### 2.7.2 Funció Maintenance\_DWH\_Absence

Aquesta funció es la encarregada d'actualitzar la taula del magatzem de dades "DWH\_Absence" i s'utilitza a través d'un sol disparador en la taula "absence" en els processos d'inserció i actualització.

### 2.7.3 Funció Maintenance\_DWH\_ActionsPlan

Aquesta funció es la encarregada d'actualitzar la taula del magatzem de dades "DWH\_ActionsPlan" i s'utilitza a través d'un sol disparador en la taula "ActionsPlan" en els processos d'inserció i actualització.

### 2.7.4 Funció Maintenance\_DWH\_Audit

Aquesta funció es la encarregada d'actualitzar la taula del magatzem de dades "DWH\_Audit" i s'utilitza a través d'un sol disparador en la taula "Audit" en els processos d'inserció i actualització.

## 2.8 Disseny i implementació de les funcions disparadors per consultes

En aquest apartat definirem i explicarem les funcions que s'utilitzen en els disparadors "triggers" per tal d'alimentar el submòdul de consultes des de el magatzem de dades

### 2.8.1 Funció Maintenance\_QueryPendingChanges

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_PendingChanges" i s'utilitza a través d'un sol disparador en la taula "DWH\_Changes" en els processos d'inserció i actualització.

### 2.8.2 Funció Maintenance\_QueryBestTechnicalManager

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_BestTechnicalManager" i s'utilitza a través d'un sol disparador en la taula "DWH\_Changes" en els processos d'inserció i actualització.

### **2.8.3 Funció Maintenance\_QueryOpenedChangesTechnicalManager**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_OpenedChangesTechnicalManager" i s'utilitza a través d'un sol disparador en la taula "DWH\_Changes" en els processos d'inserció i actualització.

### **2.8.4 Funció Maintenance\_QueryFailedReplannedChangesByRegion**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_FailedReplannedChangesByRegion" i s'utilitza a través d'un sol disparador en la taula "DWH\_Changes" en els processos d'inserció i actualització.

### **2.8.5 Funció Maintenance\_QueryOpenedActionsPlan**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_OpenedActionsPlan" i s'utilitza a través d'un sol disparador en la taula "DWH\_ActionsPlan" en els processos d'inserció i actualització.

### **2.8.6 Funció Maintenance\_QueryMostUnplannedReplacedAprover**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_MostUnplannedReplacedAprover" i s'utilitza a través d'un sol disparador en la taula "DWH\_Absence" en els processos d'inserció i actualització.

### **2.8.7 Funció Maintenance\_QueryMostUnplannedReplacedAprover**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_MostUnplannedReplacedAprover" i s'utilitza a través d'un sol disparador en la taula "DWH\_Absence" en els processos d'inserció i actualització.

### **2.8.8 Funció Maintenance\_QueryFailedExecutionLastYear**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_FailedExecutionLastYear" i s'utilitza a través d'un sol disparador en la taula "DWH\_Changes" en els processos d'inserció i actualització.

### **2.8.9 Funció Maintenance\_QueryActionsPercentagePerYear**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_ActionsPercentagePerYear" i s'utilitza a través d'un sol disparador en la taula "DWH\_ActionsPlan" en els processos d'inserció i actualització.

### **2.8.10 Funció Maintenance\_QueryApprovedChangesByGCABCurrentYear**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_ApprovedChangesByGCABCurrentYear" i s'utilitza a través d'un sol disparador en la taula "DWH\_Changes" en els processos d'inserció i actualització.

### **2.8.11 Funció Maintenance\_NonCompilanceLastSixMonths**

Aquesta funció es la encarregada d'actualitzar la taula de consultes "Query\_NonCompilanceLastSixMonths" i s'utilitza a través d'un sol disparador en la taula "DWH\_Audit" en els processos d'inserció i actualització.

## 2.9 Disseny i implementacions dels logs

Per el sistema de logs, es preveu una única taula on s'emmagatzemarà tota la informació dels procediments executats en el sistema.

Es preveuen els següents atributs:

- Id: Identificador auto incremental de log
- Procedure: Indica el nom del procediment que s'ha executat.
- InputValue1: Indica el valor del atribut d'entrada 1. Pot tenir valor null
- InputValue2: Indica el valor del atribut d'entrada 2. Pot tenir valor null
- InputValue3: Indica el valor del atribut d'entrada 3. Pot tenir valor null
- InputValue4: Indica el valor del atribut d'entrada 4. Pot tenir valor null
- InputValue5: Indica el valor del atribut d'entrada 5. Pot tenir valor null
- InputValue6: Indica el valor del atribut d'entrada 6. Pot tenir valor null
- InputValue7: Indica el valor del atribut d'entrada 7. Pot tenir valor null
- InputValue8: Indica el valor del atribut d'entrada 8. Pot tenir valor null
- InputValue9: Indica el valor del atribut d'entrada 9. Pot tenir valor null
- InputValue10: Indica el valor del atribut d'entrada 10. Pot tenir valor null
- TypeOutput: Indica el tipus de sortida del procediment INFO si ha set correcte o FATAL si ha finalitzat en error.
- OutputValue: Indica el valor de sortida de la execució del procediment. Tant si l'execució es correcte com si no. Si ha sigut correcte, tindrà el valor "OK" sinó "ERROR" + "tipus d'error"
- Timestamp: Data i hora en que s'ha executat el procediment.
- 

Tots els camps d'entrada seran tipus text per a facilitar la introducció de dades. Durant la implementació d'aquesta taula i de la funció per inserir les dades, has va intentar utilitzar paràmetres d'entrada de tipus "Any" per tal d'intentar poder introduir qualsevol tipus de dades i dins de la funció convertir-ho a text d'aquesta manera fer els inputValues de la taula fossin de tipus text i aquesta funció convertir qualsevol altre tipus de dada a text. Varen sorgir problemes quan s'introduïen valors de text en les camps tipus "Any" per la qual cosa es va optar per fer una conversió a text abans d'introduir els paràmetres a la funció per escriure a la taula de logs.

## 2.10 Disseny implementacions del joc de proves

Es realitza una funció SQL on s'utilitzen totes les funcions descrites anteriorment per a omplir la base de dades de registres suficients amb dades lo més semblants a la realitat possibles per a provar tot el sistema i que el mòdul estadístic pugui mostrar informació en la part de consultes.

Es decideix utilitzar una funció per tal de tingui un paràmetre de retorn amb 2 valors:

- OK: Si el joc de proves s'executa correctament
- ERROR + FUNCIO + ERROR: Indica si durant la realització del joc de proves es troba algun error en les funcions i indica quina funció ha donat error i el missatge d'error corresponent.



## 2.11 Disseny implementacions de la prova de funcions

Es realitza una funció SQL es comproven totes les funcions i totes les validacions que tenen cadascuna d'elles, utilitzant els registres del joc de proves i altres creats expressament per la validació però sempre sense modificar el joc de proves original i així no alterar les dades inicials.

Es necessari executar primer el joc de proves abans de la prova de funcions.

Es decideix utilitzar una funció per tal de tingui un paràmetre de retorn amb 2 valors:

- OK: Si totes les funcions i validacions s'executen correctament
- ERROR + VALIDACIO DE FUNCIO + ERROR: Indica si durant la realització de les proves de funcions es troba algun error en alguna de les validacions de les funcions i indica quina validació i quina funció ha donat error i el missatge d'error corresponent.

### 3. Conclusions

Per a concloure aquest treball definirem el que s'ha après durant la realització del projecte.

Per una banda s'ha pogut estudiar i aprendre el funcionament d'una de les aplicacions més importants dels processos ITIL, la gestió de canvis. S'ha estudiat i après el seu funcionament i amb el desenvolupament d'aquest projectes s'ha pogut comprendre molt millor aquest procés.

Per altre banda s'han reforçat els conceptes tècnics en quan el desenvolupament d'una base de dades funcional i de bases de dades de magatzem de dades.

La realització del sistema de logs ha set una part important i molt interessant d'aquest projecte.

Es considera que s'han assolit els objectius d'aquest projecte ja que s'ha aconseguit crear les bases de dades i taules necessàries per a la funció del projecte i els disparadors necessàries per actualitzar-les. També caldria repassar el sistema d'absències, substitucions i validacions de substituïts, tot i que és operatiu, la solució no es gaire elegant.

La part de planificació s'ha pogut seguir bastant bé respecte a les fites temporals, però l'ordre de les accions s'ha anat modificant durant la realització del projecte per ajustar-se millor en les necessitats de realització d'aquest.

També gracies a la metodologia seguida en la realització del projecte, s'han augmentat els coneixements de planificació de projectes i gestió del temps.

De cara a futur en aquest projecte, s'hauria de millorar com hem comentat abans, el sistema d'absències, substitucions i validacions de substituïts. També es considera que s'emmagatzema poca informació sobre la posada en marxa dels canvis, ja que es podria emmagatzemar més informació sobre qui ha realitzat la posada en marxa, quan l'ha realitzat, quins problemes han sorgit. També un cop s'ha realitzat el pla d'accions es dona per suposat que el canvi haurà set correcte, però aquest pla d'accions també seria interessant realitzar una posada en marxa, saber quin ha set el resultat.

Seguint aquest camí es podria traçar millor la posada en marxa dels canvis, s'obtidria més informació per a consultar quan un canvi ha fallat i això ajudaria a la presa de decisions i a la realització d'accions en nous canvis.

## 4. Glossari

ITIL: “Information Technology Infrastructure Library” Guia de bones pràctiques per la gestió de serveis de tecnologies de la informació.

ABM: Alta, baixa i modificació. Tipus de funcions que es creen en una classe

UML: “Unified Modeling Language” Llenguatge visual de programari per a realitzar arquitectures de programes.

SQL: “Structured Query Language”. Llenguatge de programació per construir, editar i eliminar taules i registres de bases de dades.

DWH: “Data WareHouse”. Tecnologia de magatzem de dades per a emmagatzemar grans volums de dades

Base de dades: Recopilació de dades electrònicament per a emmagatzemar i consultar

Esquema: Grup lògic per emmagatzemar taules en una base de dades

Classe: Representació lògica d'una entitat real.

Atribut: variable definida dins d'una classe que conté informació d'aquesta

Relació: Manera de representació quina i quin tipus d'unió i pot haver entre dues classes o més

Query: Instrucció que permet realitzar una consulta en una taula d'una base de dades

Procediment: Instrucció que permet realitzar una acció en una base de dades, aquesta pot tenir paràmetres d'entrada i paràmetres de sortida.

Disparador: Element que permet executar una funció quan s'afegeix, es modifica o elimina informació en una taula de la base de dades

Log: Registre d'activitat del sistema que indica les operacions que es realitzen en tot moment.

## 5. Bibliografia

[1] *Que es ITIL? Conceptes y principis* [en línia] [Consulta: 10 de Octubre de 2023]. Disponible a : <https://www.servicetonic.com/es/itil/3-itil-conceptos-y-principios/>

[2] *Altes, baixes y modificacions en base de dades* [en línia] [Consulta 28 de Octubre de 2023] Disponible a : <https://disenowebakus.net/altas-bajas-y-modificaciones-en-bases-de-datos-mysql-phpmyadmin-php.php>

[3] *Disseny conceptual de bases de dades* [en línia] [Consulta 15 de Octubre de 2023] Disponible a :  
[http://cv.uoc.edu/annotation/cb826b689abc472d8fb5b2519840058b/699689/PID\\_00213704/PID\\_00213704.html](http://cv.uoc.edu/annotation/cb826b689abc472d8fb5b2519840058b/699689/PID_00213704/PID_00213704.html)

[4] *Procediments emmagatzemats* [en línia] [Consulta 5 de Novembre de 2023] Disponible a : <https://learn.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver16>

[5] *Estil citació ISO 690* [en línia] [Consulta 20 de Desembre de 2023] Disponible a :  
<https://biblioteca.uoc.edu:8080/ca/plana/Estil-ISO-690/>

## 6. Annexos

Annex 1: Planificació final

Annex 2: Diagrama UML Mòdul funcional

Annex 3: Diagrama UML Mòdul estadístic