

Creació d'un Servidor de Catàlegs pel sector Sanitari

Memòria

Treball de final de grau de
Enginyeria Informàtica de Gestió i Sistemes
d'Informació

Fet per

Oriol Muñoz Melchor

Dirigit per

Eugeni Fernández Gonzalez

Agraïments a

Eugeni Fernández Gonzalez

Jose Carlos Ruiz

Carlos Palacios

Pierre-Yves Duquesnoy

David Gonzalez Buisan

Alberto Fuentes

Índex

1. Objecte del projecte	9
2. Estudi previ.....	11
2.1 Interoperabilitat.....	11
2.2 Estàndard mèdic	11
2.2.1 Que és un estàndard mèdic.....	11
2.2.2 Tipus d'estàndard	11
2.2.3 Obtenció d'estàndards	13
2.2.4 Components d'un estàndard.....	13
2.3 Catàlegs.....	13
2.3.1 Productors	13
2.3.2 Repositoris.....	14
2.3.3 Consumidors	14
3. Objectius i abast	15
3.1 Usuari final	15
3.2 Objectius	15
3.2.1 Objectius del producte	15
3.2.2 Objectius del client	16
3.3 Criteris d'Acceptació.....	16
3.3.1 Acceptació del projecte.....	16
3.3.2 Acceptació dels mòduls	16
4. Metodologia.....	17
4.1 Tasques	17
4.2 Etapes	17
4.2.1 Estudi.....	18
4.2.2 Inserció i persistència	18
4.2.3 Exportació i API.....	18
4.2.4 Test integral de catàleg.....	18
4.2.5 Test integral de l'aplicació.....	18
4.2.6 Desenvolupament de la part davantera	18
5. Requeriments.....	19
5.1 Casos d'ús.....	19
5.1.1 Cas d'ús d'inserir un estàndard	19

5.1.2	Cas d'ús de consultar un estàndard	20
5.1.3	Cas d'ús de modificar un estàndard	21
5.2	Requeriments funcionals	22
5.2.1	Requeriments del mòdul de persistència	22
5.2.2	Requeriments del mòdul d'introducció de estàndard	22
5.2.3	Requeriments del mòdul de carrega de dades.....	22
5.2.4	Requeriments del mòdul de gestor d'API.....	23
5.2.5	Requeriments del mòdul d'autorització d'API	23
5.2.6	Requeriments de la part davantera	23
5.3	Requeriments no funcionals	24
6.	Desenvolupament	25
6.1	Elecció de tecnologies	25
6.1.1	Backend:	25
6.1.1.1	Que es IRIS	25
6.1.2	Perquè IRIS.....	25
6.1.3	Avantatges tècniques	25
6.2	FrontEnd	26
6.3	Model de classes	26
6.3.1	Classes d'estructura	27
6.3.2	Classes d'autorització.....	30
6.4	Cicle de vida d'un estàndard i permisos d'usuaris	31
6.4.1	Cicle de vida d'un estàndard	31
6.4.2	Permisos per tipus d'usuari.....	33
6.5	Estructura del projecte	33
6.6	Esprints	34
6.6.1	Estudi.....	34
6.6.2	Creació de permisos i taules de persistència a Commons.....	34
6.6.3	Creació del mòdul de inserció d'estàndard	34
6.6.4	Creació del mòdul de persistència d'estàndard.....	34
6.6.5	Creació del mòdul d'etapa	35
6.6.6	Migració de metodologia	35
6.6.7	Creació del mòdul de autorització.....	35
6.6.8	Creació del mòdul de API	36

6.7 Problemes generals en el desenvolupament	40
6.7.1 Problemes en el plantejament	40
6.7.2 Problemes en la tecnologia	40
7. Guia d'ús del producte	43
7.1 Limitacions del sistema	43
7.2 Operacions	44
7.2.1 Inserció.....	44
7.2.2 Validació.....	45
7.2.3 Consulta.....	45
7.2.4 Extinció	46
7.3 Rols.....	46
7.3.1 Usuari IRIS:	46
7.3.2 Usuari d'estàndard	47
7.4 Implantació del sistema.....	47
7.4.1 Requeriments mínims	47
7.4.2 Arrencament del sistema	47
7.4.3 Part davantera	47
8. Possibles millores del producte	49
9. Conclusió	51
10. Bibliografia	53
10.1 Cites	53
10.2 Fonts.....	53

Índex de figures

1Classes estructurals del estàndard	27
2Classes d'autorització	30
3Cicle de vida d'un estàndard	31
4Permisos per tipus d'usuari i fase del estàndard	33

Creació d'un Servidor de Catàlegs pel sector Sanitari

Objecte del projecte

1. Objecte del projecte

L'objecte del projecte és el desenvolupament d'un software capaç d'emmagatzemar, actualitzar, traduir i possibilitar la consulta, per usuaris i/o per aplicacions, d'estàndards mèdics, facilitant així, la interoperabilitat entre sistemes, es a dir una comunicació clara i precisa amb un vocabulari de definicions estàndard i compartit.

Aquest catàleg d'estàndards permetrà que tot tipus d'aplicacions, principalment mèdiques, tot i que no exclusivament, pugin mantenir-se al dia amb els estàndards publicats, tant per entitats públiques com per privades, així com mantenir i actualitzar el seus propis estàndards. En forma d'un servei de descarregar i avis de canvis en els estàndards que aquestes aplicacions hagin expressat interès,

Creació d'un Servidor de Catàlegs pel sector Sanitari

Objecte del projecte

2. Estudi previ

2.1 Interoperabilitat

“La interoperabilidad es la capacidad de los sistemas de información y de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos.”[1]

La interoperabilitat és una necessitat que neix de tenir sistemes diferents, possiblement independents i desenvolupats per persones amb cultures, idiomes i idees diferents.

Això implica que, per a poder assegurar una comunicació clara i precisa entre els sistemes, s'ha d'estudiar i planificar tant estructures de dades com definicions compartides en les comunicacions.

Així neix, en l'àmbit de la medicina, la necessitat dels estàndards mèdics, que són els encarregats d'establir aquestes estructures i vocabularis, degut a la sensibilitat i importància de la interoperabilitat en els sistemes mèdic.

2.2 Estàndard mèdic

2.2.1 Que és un estàndard mèdic

Un estàndard mèdic es un vocabulari, d'estructura o de termes, normalitzat sobre un àrea específica de l'àmbit mèdic que elimina qualsevol tipus de subjectivitat en la interoperabilitat.

És a dir, és un conjunt d'entrades amb definicions, dades, esquemes, estructures, informació, versions i, normalment, algun tipus de codi alfanumèric sobre un tema específic. Aquest permet identificar qualsevol medicació, malaltia, símptoma, informe, imatge, exploració d'un pacient, etc, de forma única i inequívoca per a tot software o professional que atengui a l'estàndard, permeten així una comunicació clara i concisa sense importar barreres com el temps o el llenguatge.

2.2.2 Tipus d'estàndard

En la comunicació entre sistemes, hi ha necessitat de diferents tipus d'estàndard per a poder englobar tota la complexitat de la comunicació mèdica.

Així doncs, podem trobar tres principals varietats d'estàndard.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Estudi previ

2.2.2.1 Estàndard de missatgeria

Aquesta varietat d'estàndard està centrat en el missatge, es a dir, en l'estructura, els tipus de camps i els formats de les comunicacions entre sistemes, això ajuda a evitar errors de format, de rebre un camp per altre o d'ometre camps necessaris entre d'altres errors que l'estructura del missatge pot comportar.

Els estàndards més usats en aquesta categoria són DICOM i HL7.

DICOM és un estàndard per l'intercanvi d'imatges que emmagatzema dades del pacient junt amb la imatge, impedit així possibles confusions en qüestió de qui és la imatge, com podria ser el cas amb una radiografia.

També proporciona un protocol TCP/IP per la difusió d'imatges a través de la xarxa.

HL7 és l'estàndard més usat i és un descriptor de dades que usa DFDL per a descriure informes de probes, exploracions i processos, podent mantenir així el mateix arxiu de dades durant tot el cicle de vida del pacient.

HL7 presenta diferents versions on els camps varien i a partir de la V3 està permesa l'edició directa amb XML en comptes d'haver d'utilitzar l'idioma de descripció DFDL.

2.2.2.2 Estàndard de terminologia

Un estàndard de terminologia és un estàndard que proporciona definicions sobre un àrea clínica, eliminant així, ambigüitat sobre la terminologia intercanviada.

Un exemple seria el ICD-11 o SNOMED.

ICD-11 és un estàndard específic que engloba les malalties i causes de mortalitat que afecten a la població. És un estàndard publicat pel WHO i obligatori en l'àmbit internacional.

SNOMED, per contra, és un estàndard de caràcter general que tracta d'especificar i definir tot el vocabulari utilitzat en l'àmbit mèdic, englobant actualment més de 300.000 definicions.

2.2.2.3 Estàndard de documentació

Un estàndard de documentació és un estàndard similar al d'estructura centrat en els professionals del camp. Aquest està orientat a estructurar un document per un procés en particular fent que sigui llegible per un professional. Aquest estàndard ens assegura que qualsevol professional de la medicina

Creació d'un Servidor de Catàlegs pel sector Sanitari

Estudi previ

podrà llegir aquests documents.

2.2.3 Obtenció d'estàndards

Els estàndards, quan són utilitzats per una aplicació, s'han de, generalment, descarregar i mantenir en persistència, de forma que sigui accessible ràpidament pel sistema, això implica descarregar l'estàndard de webs d'entitats públiques o privades en algun dels formats disponibles. Generalment aquests formats solen ser CSV, JSON, XML o documents PDF que són els únics que un usuari pot arribar a entendre.

2.2.4 Components d'un estàndard

Un component es pot dividir en dos principals parts.

2.2.4.1 Metadades

Les metadades d'un estàndard és la informació que descriu el estàndard.

Descriu la versió, el nom, les columnes que cada definició de l'estàndard te, entre d'altres. Aquestes dades són las que permeten diferenciar un estàndard d'un altre.

2.2.4.2 Dades

Les dades són les definicions que un estàndard te, poden ser de qualsevol tipus, i son els valors usats en la interoperabilitat.

2.3 Catàlegs

Un catàleg és un servei que emmagatzema un o més estàndards mèdics, possiblement establint relacions de similitud.

Podem entendre que, en els catàlegs, tenim tres tipus d'interacció.

2.3.1 Productors

Són els que creen els estàndards mèdics i que, generalment, també faran de repositori.

Exemples serien els organismes públics que creen estàndards nacionals o empreses privades que creen catàlegs, per exemple, dels seus productes.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Estudi previ

2.3.2 Repositoris

Són els catàlegs pròpiament dits, contenen estàndards i en permeten la seva consulta i descarrega.

2.3.3 Consumidors

Són els sistemes que interoperan amb els estàndards, requereixen d'accés a repositoris i depenen d'ells per mantenir-se al dia.

Usen les definicions dels estàndards per a comunicar-se.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Objectius i abast

3. Objectius i abast

3.1 Usuari final

L'usuari objectiu d'aquest projecte es qualsevol tipus de sistema d'informació, centrat especialment en l'àmbit de la medicina, que tingui dependències en estàndards que han de ser mantinguts i actualitzats.

En la situació actual d'investigació i d'evolució constant en l'àmbit de la medicina, sorgeix la necessitat de mantenir les definicions dels diferents estàndards utilitzats per un sistema. Això implica el desenvolupament d'un mòdul o sistema separat que gestioni l'actualització i canvi dels estàndards.

En conseqüència, aquest projecte és el desenvolupament de mòduls lleugers capaços de ser acoblats, de forma senzilla i intuïtiva, a sistemes que requereixin de les funcionalitats que els catàlegs aporten, així simplificant i reduint tant el cost, com la complexitat del desenvolupament dels sistemes que requereixin d'aquesta interoperabilitat.

3.2 Objectius

3.2.1 Objectius del producte

- Crear, a través de fonts de dades, estàndards en forma de persistència en una base de dades.
- Mantenir estàndards, afegint, retirant o actualitzant entrades en les seves diferents versions.
- Consultar els estàndards, tant per part dels sistemes com dels usuaris, ja sigui amb accés directe a la persistència o de forma remota amb una API exposada.
- Connectar els diferents estàndards que existeixen en un sistema per les seves similituds.
- Notificar les actualitzacions o canvis que es presentin en un estàndard del qual un sistema hagi expressat la voluntat de rebre'n.
- Actualitza estàndards automàticament, havent informat prèviament d'un origen d'on obtenir la informació s'obre l'estàndard.
- Oferir un part davantera, senzill, per a interactuar amb els estàndards.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Objectius i abast

3.2.2 Objectius del client

- Regularitzar els catàlegs, es a dir, establir una única plataforma que els gestioni.
- Automatitzar l'actualització dels catàlegs en sistemes dependents.
- Simplificar la implementació dels catàlegs en projectes actuals i/o futurs.
- Facilitar el manteniment i actualització dels catàlegs propis.

3.3 Criteris d'Acceptació

3.3.1 Acceptació del projecte

Degut a que els objectius tenen un caràcter general que fa difícil determinar si s'han assolit, s'utilitza en el projecte, els estàndards: SNOMED, ICD, LOINC i VADEMECUM, com a referència, es a dir, si per tots els anteriorment mencionat estàndards, el sistema és capaç de complir tots els objectius, el projecte haurà estat un èxit.

3.3.2 Acceptació dels mòduls

Com es discuteix més endavant ([4.4.2 Modularitat](#)), el projecte és desenvolupa de forma modular, amb funcionalitats divisibles que són complertes per si mateixes, per això i per tal d'assegurar el correcte funcionament d'aquests mòduls independents, es desenvolupen tests individuals de funcionament per cada mòdul, que hauran de donar un resultat positiu per a que el mòdul es consideri correcte i funcional.

4. Metodologia

4.1 Tasques

1. Estudi dels estàndards esmentats en el objectius ([3.2 Criteris d'Acceptació](#)).
2. Estudi de la tecnologia.
3. Creació del mòdul de persistència.
4. Creació del mòdul d'introducció de estàndard.
5. Creació del mòdul d'exportació.
6. Creació del mòdul de gestor d'API.
7. Creació del mòdul d'autorització d'API.
8. Endpoints per a test.
9. Introducció dels estàndards.
10. Test integral del producte.
11. Desenvolupament d'un part davantera.

4.2 Etapes

El projecte estarà dividit en 7 principals etapes.

1. Estudi
2. Inserció i persistència
3. Exportació i API
4. Test integral de catàleg
5. Test Integral d'actualització
6. Part davantera

Creació d'un Servidor de Catàlegs pel sector Sanitari

Metodologia

4.2.1 Estudi

L'estudi compon la fase inicial del projecte i està majoritàriament, sinó totalment, compost per recerca i investigació per entendre la tecnologia necessària per dur a terme els objectius del projecte.

4.2.2 Inserció i persistència

La inserció i persistència compon la segona fase del projecte i és el desenvolupament de la introducció i comprovació de les dades que s'insereixen al sistema sobre un estàndard amb la seva posterior persistència en la base de dades.

També inclou la modificació d'un estàndard.

4.2.3 Exportació i API

L'exportació i API és el desenvolupament de la consulta de dades en forma de API o com a crida directa a les classes dels mòduls.

4.2.4 Test integral de catàleg

El test integral serà la comprovació completa de les funcionalitats desenvolupades, el test total es realitza en aquest punt ja que s'ha assolit els mínims per a un catàleg funcional, es a dir, introducció, permanència, manteniment i consulta.

4.2.5 Test integral de l'aplicació

Aquest segon test general comprova el adequat funcionament de tota l'aplicació incloent les noves classes afegides en avís i actualització.

4.2.6 Desenvolupament de la part davantera

La part davantera és un model senzill que inclourà tres vistes, una vista de inici de sessió, una vista de taula i una vista de detall.

5. Requeriments

5.1 Casos d'ús

Per a poder extreure amb precisió les necessitats de cada mòdul s'han descrit casos d'ús que descriuen les interaccions que l'usuari pot tenir amb el sistema.

Aquests casos d'ús no contemplen la part davantera, ja que és prioritari extreure únicament les necessitats del sistema, la part davantera es discuteix més endavant.

Cal denotar que l'usuari, en aquest cas, pot ser tant una persona, com un sistema, no hi haurà diferència en el funcionament.

La representació dels casos d'ús es troba en un drive ubicat en la bibliografia.

5.1.1 Cas d'ús d'inserir un estàndard

Descripció: Un usuari, amb rol d'administrador, insereix un estàndard en el sistema instal·lat en la seva màquina.

Actor: Usuari amb rol d'administrador.

Pre-requisits: L'usuari té credencials i està autoritzat com administrador en el sistema.

Post-condicions en cas de èxit: L'usuari haurà introduït un estàndard nou al sistema. **Post-condicions en cas de fracàs:** L'usuari rep una resposta indicant l'error.

Flux:

1. L'usuari envia una petició API amb les metadades del estàndard.
2. La consulta entra al sistema pel mòdul d'API.
3. El sistema verifica que el cos de la petició conté les dades necessàries {1}.
4. El sistema envia la petició al mòdul d'introducció d'estàndard.
5. El mòdul d'introducció, posterior a comprovar que no existeix un estàndard igual en el sistema {2}, insereix el nou estàndard com una nova classe de metadades d'estàndard.
6. El mòdul envia una confirmació a l'usuari.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Requeriments

7. L'usuari rep la confirmació i comença a enviar l'estàndard en peticions subseqüents.
8. L'estàndard arriba al mòdul d'API que comprova que la estructura general del cos de la petició sigui correcte i que tingui el paràmetre amb l'indicador de l'estàndard {3}, en cas afirmatiu el dirigeix al mòdul de persistència d'estàndard.
9. El mòdul de persistència emmagatzema les entrades una per una, emmagatzemant s'hi ha aquelles que no compleixen la definició de l'estàndard .
10. El sistema retorna una confirmació al usuari i informació de entrades que han donat error.

Flux d'error:

{1}: El sistema retorna un avís indicant l'error del cos.

{2}: El sistema retorna informació que l'estàndard ja existeix en el sistema junt amb l'identificador del mateix.

{3}: El sistema retorna un avís indicant com el cos per a introduir entrades en un estàndard ha de ser.

5.1.2 Cas d'ús de consultar un estàndard

Descripció: Un usuari, amb qualsevol rol, consulta un catàleg.

Actor: Usuari

Pre-requisits: L'usuari té credencials i està autoritzat en el sistema.

Post-condicions en cas de èxit: L'usuari podrà consultar el estàndard. **Post-**

condicions en cas de fracàs: l'usuari rep una resposta indicant l'error. **Flux:**

1. L'usuari forma una petició API amb el identificador de l'estàndard i les opcions amb un valor opcional de paginació.
2. La consulta accedeix al sistema pel mòdul de API que l'envia al mòdul d'exportació que carrega les dades en un objecte JSON.
3. El sistema retorna l'estàndard, en cas de que la paginació no estigui definida retorna la primera pàgina de valors, junt amb els enllaços al següent i anterior pàgina, si hi ha. (S'ha modificat, no retorna enllaç ja que és per paràmetre no per URL)

Creació d'un Servidor de Catàlegs pel sector Sanitari

Requeriments

Flux alternatiu:

1. L'usuari fa una crida directa a un mètode del mòdul d'exportació amb els arguments pertinents.
2. La consulta carrega tots els estàndards a traves de la classe de persistència del estàndard.
3. El sistema retorna l'estàndard com una llista de instàncies de classe.

5.1.3 Cas d'ús de modificar un estàndard

Descripció: Un usuari, amb un rol d'editor o administrador, modifica un l'entrada d'un catàleg.

Actor: Usuari

Pre-requisits: L'usuari te credencials i esta autoritzat en el sistema.

Post-condicions en cas de èxit: L'usuari podrà modificar una entrada del estàndard.

Post-condicions en cas de fracàs: l'usuari rep una resposta indicant l'error.

Flux:

1. L'usuari envia una petició de modificació amb l'identificador de l'estàndard, de l'entrada i els valors a canviar.
2. El mòdul d'API rep la petició i l'envia al mòdul d'introducció d'estàndard.
3. El mòdul d'introducció persisteix els canvis en el sistema.
4. El mòdul retorna un missatge al usuari informant del resultat de l'operació.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Requeriments

5.2 Requeriments funcionals

5.2.1 Requeriments del mòdul de persistència

1. El mòdul ha de poder rebre una llista d'entrades d'un estàndard en JSON.
2. El mòdul ha de generar una classe amb el nom de l'estàndard, els camps extrets de la informació i els índexs necessaris.
3. El mòdul ha de persistir cada entrada rebuda en el JSON en la base de dades en forma d'instància de la classe que ha generat.
4. El mòdul ha de, donada la prèvia informació, relacionar les entrades dels estàndards per ser traduccions totals o parcials a traves de relacions en les classes.
5. El mòdul ha de poder documentar, en un registre, quines insercions han estat satisfactòries i quines han retornat un error, per a posteriorment informar a l'usuari.
6. El mòdul ha de permetre modificar una entrada.

5.2.2 Requeriments del mòdul d'introducció de estàndard

1. El mòdul ha de rebre una arxiu CSV i processar-la a JSON.
2. El mòdul ha de poder avisar sobre entrades descartades en el CSV.
3. El mòdul ha de poder identificar si l'estàndard ja existeix en la persistència.
4. El mòdul ha de poder enregistrar la introducció d'un nou estàndard, es adir, las metadades de l'estàndard, a la base de dades.
5. El mòdul ha de poder enviar les dades de l'estàndard, les entrades, a un altre mòdul.

5.2.3 Requeriments del mòdul de carrega de dades

1. El mòdul ha de poder rebre els paràmetres d'opcions i una identificació de l'estàndard.
2. El mòdul ha de poder processar els paràmetres d'opcions com a un filtre definit.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Requeriments

3. El mòdul ha de poder executar una petició a persistència amb filtres.
4. El mòdul ha de poder retornar les dades en una cadena JSON.
5. El mòdul ha de poder retornar les dades com a una llista d'objectes.

5.2.4 Requeriments del mòdul de gestor d'API

1. El mòdul ha de poder rebre informació a través de rutes API definides.
2. El mòdul ha de poder fer crides a API definides.
3. El mòdul ha de poder gestionar l'autorització i permisos de cada usuari.

5.2.5 Requeriments del mòdul d'autorització d'API

1. El mòdul ha de poder emmagatzemar els usuaris en la base de dades.
2. El mòdul ha de poder emmagatzemar els permisos dels usuaris respecte als estàndards en base de dades.
3. El mòdul ha de poder confirmar si les credencials rebudes corresponen a un usuari.
4. El mòdul ha de poder confirmar si un usuari té permisos suficients per a una acció amb un estàndard específic.

5.2.6 Requeriments de la part davantera

1. La part davantera ha de poder rebre una cadena JSON amb la informació i extreure les dades de la representació del estàndard.
2. La part davantera ha de permetre a l'usuari iniciar sessió.
3. La part davantera ha de poder interpretar la cadena JSON i extreure les especificacions de representació i generar una vista de taula i una de detall.
4. La part davantera ha de permetre navegar entre la vista de taula a la detall i envers.
5. La part davantera ha de permetre enviar peticions d'inserció d'estàndard. (Cancel·lat* Llegir tasques ->[])
6. La part davantera ha de permetre enviar peticions de modificació d'estàndard.
7. La part davantera ha de permetre marcar estàndards i versions com a validades, deprecades o en desús.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Requeriments

5.3 Requeriments no funcionals

1. El sistema ha de poder determinar si un usuari té autorització per a realitzar una acció.
2. El sistema ha de tenir un certificat SSL per a poder tenir una connexió web xifrada.
3. El sistema ha de poder reportar errors en el sistema de forma que en quedi constància.
4. El sistema ha de ser fàcilment instal·lable en nous servidors.
5. El sistema ha de poder rebre un volum considerable de peticions per HTTP.

6. Desenvolupament

6.1 Elecció de tecnologies

6.1.1 Backend:

El desenvolupament del backend s'ha realitzat en InterSystems IRIS for Health .

6.1.1.1 Que es IRIS

IRIS és una plataforma de desenvolupament d'InterSystems per al sector sanitari. Aquesta plataforma incorpora el seu propi llenguatge, ObjectScript, base de dades integrada i un gestor visual de projecte en les seves produccions.

6.1.2 Perquè IRIS

S'ha decidit usar IRIS ja que és una tecnologia de referència en l'àmbit mèdic, usada en centres sanitaris de tot el món, amb una gran presència a Europa. A més, ja que esta centrada en l'àmbit sanitari, incorpora moltes eines d'utilitat i molts sistemes que requereixen de solucions que depenen d'estàndards mèdics i que, en conseqüència, necessiten d'un catàleg per a gestionar els estàndards.

6.1.3 Avantatges tècniques

6.1.3.1 Gestió de base de dades

La tecnologia IRIS for Health incorpora la base de dades directament, així eliminant la necessitat de gestionar connexió.

Seguidament, ObjectScript incorpora la generació, eliminació i gestió del canvi de les taules de classes que es defineixin a través de la classe %Persistent, cosa que elimina la necessitat de crear un model de base de dades. A més a més gestiona el canvi entre els tipus de dades presents en ObjectScript i els presents en SQL facilitant la persistència de dades.

6.1.3.2 Gestió del projecte a través de produccions

Iris incorpora un portal de gestió que permet crear produccions. Aquestes produccions són gestors de projectes basades en missatges, permeten definir punts d'entrada pel projecte (Business service), lògica de transformació de dades i de redirecció de missatges (Business procés) i finalment necessitats del projecte i/o sortides externes (Business operations).

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

Permetent així mantenir una traça dels missatges que passen pel sistema a més a més de aïllar errors i gestionar volum de peticions i dades.

El projecte es desenvolupa en una producció on hi ha Business processes que realitzen les operacions que requereixen canvis en la persistència, Business services que implementen una façana REST per a poder rebre peticions i Business operations on es requereixin per a adaptar les dades entre les fases.

6.1.3.3 Integració d'API

Iris ofereix dos principals avantatges per al desenvolupament d'API:

Gestió d'usuaris: Iris permet tenir usuaris registrats amb diferents nivells d'accés tant al portal de gestió com a les diferents aplicacions web que el servidor implementi.

Aplicacions web i gestió de gateway: Iris permet tenir varies aplicacions web, que són URL exposades sobre les quals podem crear varis Endpoints per a que els usuaris pugin fer peticions.

Aquestes aplicacions també implementen mesures d'autenticació amb els usuaris anteriorment esmentats. Així eliminant una part important de la complexitat d'una API.

6.2 FrontEnd

El FrontEnd s'ha realitzat amb Angular 16.

Angular és un framework de TypeScript amb una orientació a objectes més forta que altres ambients de desenvolupament web, fent així un desenvolupament basat en components reciclables més fàcil que amb altres solucions. Degut a la naturalesa del projecte, un catàleg d'estàndards, l'usuari crearà contingut, específicament objectes, pels quals un frontend

específic no ha estat dissenyat, amb lo qual per a poder tenir una representació més accessible a usuaris menys tècnics en la forma d'un frontend s'han desenvolupat components genèrics en Angular que responen a una varietat de casos tractant de ser reutilitzables.

6.3 Model de classes

Degut a que en ObjectScript la definició d'una classe que estengui la classe %Persistent ja implica la creació d'una taula en base de dades s'ha definit, en les figures a continuació, la informació de les classes i de base de dades.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

Denotar que aquí no es troben totes les classes que es desenvolupen en el projecte, sinó només aquelles que generen una persistència en la base de dades.

En el projecte es divideixen en dos tipus les classes de persistència que s'utilitzen:

- Classes d'estructura
- Classes d'autorització

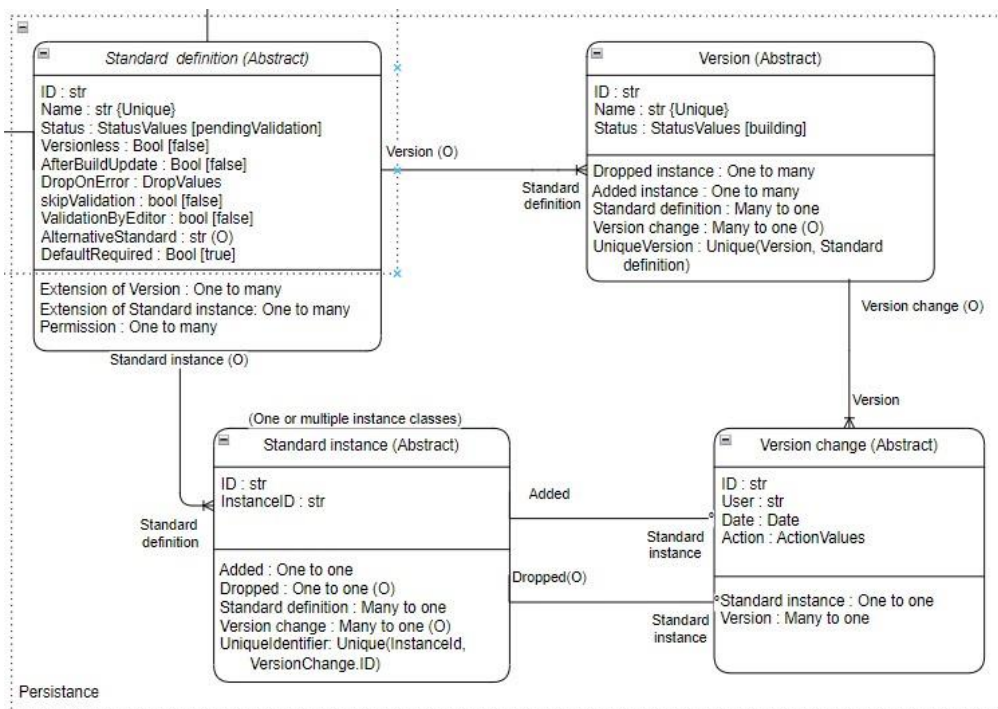
6.3.1 Classes d'estructura

Les classes d'estructura són aquelles que conformen la pròpia estructura de l'estàndard, contenint tant la seva informació com la seva meta informació.

Totes aquestes classes no són utilitzades de forma directa, sinó que defineixen els mínims requeriments que un estàndard ha de complir per a poder funcionar en el sistema. Tot estàndard ha d'estendre totes aquestes classes i definir la informació que hagi d'emmagatzemar en les classes que generi.

En el catàleg d'estàndards, un estàndard contindrà aquestes classes estructurals:

- Standard definition
- Version
- Standard instance
- Version change



1Classes estructurals del estàndard

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

6.3.1.1 Standard definition

Defineix el estàndard en si mateix, serveix per a que el sistema pugi identificar-lo i serveix de connexió amb la resta de classes persistents en el sistema.

Propietats:

1. Id: Id de base de dades
2. Name: Nom de l'estàndard, ha de ser únic.
3. Status: Defineix el punt del cicle de vida en que es troba l'estàndard. Pot tenir els valor: [pendingValidation, inUse, deprecated, obsolete]. Per defecte: "Pending validation"]
4. DropOnError: Defineix com s'actua en cas de que en la inserció una o múltiples files no siguin correctes estructuralment. Pot tenir els valors: [none,clearFrame,clearAll]. Per defecte: "none". None ignora en cas de error, clearFrame elimina totes les dades introduïdes en la mateixa petició, clearAll elimina la versió al complert.
5. skipValidation: Defineix si s'omet la fase de validació. Per defecte fals
6. AlternativeStandard: Informació opcional per al final de vida de l'estàndard indicar al consumidor un estàndard substitutori.
7. DefaultRequired: Fa que les propietats de l'estàndard, les de la classe que estengui aquesta siguin per defecte obligatòries a menys que s'indiqui lo contrari. Per defecte: cert.

Relacions:

1. Standrd instance: un estàndard te una o molts tipus d'instàncies. (No entrades, tipus d'instàncies, un estàndard pot tenir diferents tipus d'entrada).
2. Version: un estàndard te zero o varies versions.

6.3.1.2 Version

Defineix els requeriments mínims d'una versió d'un estàndard.

Propietats:

1. Id: Id de la base de dades.
2. Name: nom, ha de ser únic.
3. Status: Etapa del cicle de vida en la que es troba la versió. Pot tenir els valor: [building, pendingValidation, inUse, deprecated, obsolete, legacy]. Per defecte: "Building"

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

Relacions:

1. StandardDefinition: Una versió té un estàndard.
2. VersionChange: Una versió té zero o varis canvis de versió.

6.3.1.3 StandardInstance:

Propietats:

1. Id: Id de la base de dades.
2. InstanceId: Id d'instància en el context de versió i estàndard, ha de ser única per la canvi. Relacions:
 1. Added: Instància que defineix en quina versió la instància va ser afegida.
 2. Dropped: Opcional, defineix en quina versió la instància va ser retirada.
 3. StandardDefinition: Una instància va lligada a una definició d'estàndard.

6.3.1.4 Version change:

Defineix un canvi que s'ha produït en una versió, documentant la instància que ha patit el canvi i la versió en la qual s'ha realitzat.

Propietats:

1. Id: Id de base de dades.
2. User: Nom del usuari que ha realitzat el canvi.
3. Date: Data en la que s'ha realitzat el canvi.
4. Action: Acció que s'ha realitzat. Valor possibles:[Added,Dropped]

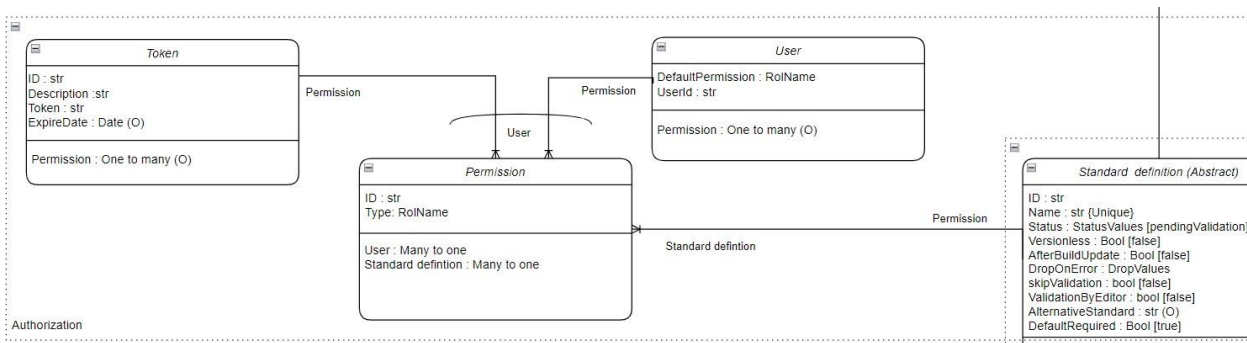
Relacions:

1. StandardInstance: Un canvi de versió és sobre una entrada de l'estàndard.
2. Version: Indica la versió en la qual s'ha realitzat el canvi.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

6.3.2 Classes d'autorització



2Classes d'autorització

6.3.2.1 Token

Token és una de les possibles formes d'autenticació, principalment pensada per sistemes o aplicacions.

Propietats :

1. Id: Id de base de dades.
2. Description: Descripció de l'ús que se li dona al token.
3. Token: La cadena que autoritza.
4. ExpireDate: Data d'expiració.

Relacions:

1. Permission: Un token pot tenir varis permisos per a diferents estàndards.

6.3.2.2 User

Classe connectora de permission i un usuari de Iris. Propietats:

1. DefaultPermission: Permís de base que se li donarà al usuari quan es creï un nou estàndard, pot ser nul.
2. UserId: Id d'un usuari de IRIS.

Relacions:

1. Permission: Un usuari pot tenir varis permisos per a diferents estàndards.

6.3.2.3 Permission

Permission és una classe d'autorització per a interactuar amb un estàndard. Es a dir, estableix si un token o usuari pot consumir o modificar un estàndard.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

Propietats:

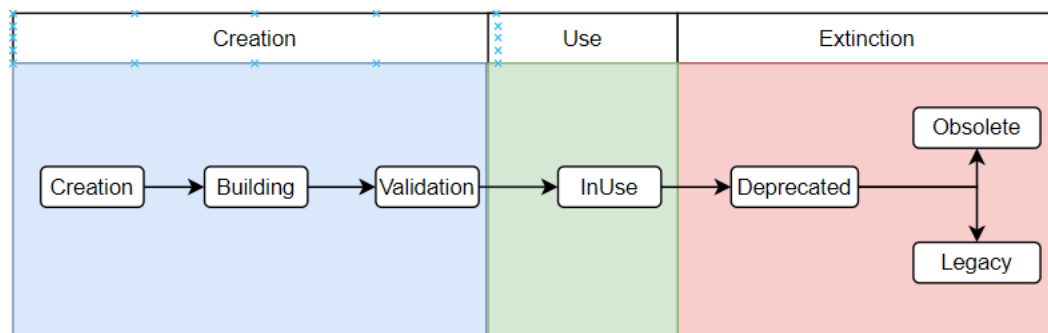
1. Id: Id de base de dades.
2. Type: Tipus de rol que te en el estàndard. Pot tenir valors de [Consumer, Editor,Admin] Relacions:
 1. StandardDefinition: Un permís és d'un estàndard i un estàndard te a varis, usuaris, permetis.

6.4 Cicle de vida d'un estàndard i permisos d'usuaris

6.4.1 Cicle de vida d'un estàndard

Degut a la necessitat d'actualització dels diferents termes y conceptes que un estàndard defineix és contempla que tot estàndard i versió d'estàndard passa per un cicle de vida compost per tres fases: creació, ús i extinció.

Així doncs, tot estàndard i versió dels mateixos, en el projecte, observa el cicle de vida definit en la següent figura.



3Cicle de vida d'un estàndard

6.4.1.1 Creació

La creació és l'inici del cicle de vida d'un estàndard i involucra tres fases:

1. Creation: Aquesta fase és exclusiva a un estàndard i compon la definició de l'estructura del estàndard, es a dir, les classes i les estructures de dades que contindran les dades de l'estàndard. Denotar que, ja que una versió esta continguda en un estàndard, aquesta te la estructura de dades definida prèviament, i la comparteix amb totes les altres versions de l'estàndard.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

2. **Building:** Introducció de les dades de l'estàndard o versió, definicions, Ids, noms, descripcions, entre d'altres, a l'estructura prèviament creada.
3. **Validation:** Degut a la sensibilitat de les dades i la importància de que aquestes siguin veraces i correctes, es defineix una fase de validació on un expert en el camp, en el cas d'estàndard mèdic podria ser un doctor, valida les dades i confirma que són correctes.

6.4.1.2 Ús

Aquesta etapa involucra únicament un fase, InUse. Aquesta fase és el moment en que consumidors podran usar el estàndard o versió que el catàleg ofereix. Un estàndard o versió en aquesta fase implica que les seves dades són correctes i actuals, a més a més de que el servei d'accés és estable y es mantindrà en un futur proper.

6.4.1.3 Extinció

Aquesta etapa és el final de vida d'un estàndard o versió i compona la notificació a usuaris del fi d'usabilitat del estàndard o versió i terminació dels serveis. Te dos fases:

1. **Deprecated:** Aquesta fase s'usa per a indicar als consumidors el final de suport d'un estàndard o versió, els consumidors no han de començar a utilitzar aquest estàndard o versió, i aquells que ja l'utilitzin haurien de migrar a un estàndard o versió més nova.
2. **Obsolete:** Aquesta ultima fase és la retirada dels serveis, es a dir, de l'accessibilitat del consumidor a les dades.
3. **Legacy:** Degut a la dificultat de migració i la reticència de molts desenvolupadors a migrar a noves versions de serveis, s'ha donat una opció alternativa a la retirada completa. Legacy, implica el manteniment de serveis tot i l'antiguitat del estàndard o versió. Cal indicar que s'ofereix per a serveis de traducció de dades que estiguin en versions anteriors i per a no causar mal funcionament en sistemes que segueixin consumint l'estàndard o versió.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

6.4.2 Permisos per tipus d'usuari

En la següent figura, es descriu com un usuari interactuarà segons el seu rol en l'estàndard i l'etapa de vida en la que l'estàndard es trobi.

CAN RETRIEVE THE STANDARD/VERSION						
USER/STATUS	building	pendingValidation	inUse	deprecated	obsolete	legacy
Admin	YES	YES	YES	YES	YES	YES
Editor	YES	YES	YES	YES with warning	YES with warning	YES with warning
Consumer	NO	NO	YES	YES with warning	NO	YES with warning

CAN MOVE THE STAGE OF THE STANDARD/VERSION						
USER/STATUS	building	pendingValidation	inUse	deprecated	obsolete	legacy
Admin	YES	YES	YES	YES	YES	YES
Editor	YES	Validation by editor or rollback	Forward only	Forward only	NO	NO
Consumer	NO	NO	NO	NO	NO	NO

CAN EDIT THE STANDARD/VERSION						
USER/STATUS	building	pendingValidation	inUse	deprecated	obsolete	legacy
Admin	YES	YES	NO	NO	NO	NO
Editor	YES	NO	NO	NO	NO	NO
Consumer	NO	NO	NO	NO	NO	NO

4Permisos per tipus d'usuari i fase del estàndard

6.5 Estructura del projecte

El projecte es divideix per funcionalitats, prenen una estructura de carpetes amb una carpeta Commons que conté tot el que sigui compartit o porti funcionalitats de suport a les funcionalitats del projecte que tenen una carpeta individual al mateix nivell que la carpeta de Commons.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

6.6 Esprints

6.6.1 Estudi

Treball realitzat: S'ha estudiat l'estructura dels estàndards en especial SNOMED i s'ha fet una aproximació inicial de la tecnologia.

Resultats:

- Es requereix d'una estructura de base que permeti tenir varis tipus d'estructura d'instància d'estàndard.
- Creació de l'estructura de classes i establiment del cicle de vida i permisos per a usuaris definitius.
- La classe %JSON.Adaptor que estava contemplada per a fer de pont del cos de petició en JSON a objecte de persistència és inadequada al no permetre editar objectes, s'ha de crear un nou adaptador.

Contratemps: S'ha hagut de crear les classes per a fer una nova implementació de JSON.

6.6.2 Creació de permisos i taules de persistència a Commons

Treball realitzat: S'ha creat una classe ajudant que determini si una acció és possible segons les condicions prèviament determinades. S'ha creat totes les taules de persistència, tant estructurals com d'autorització, s'ha verificat la seva integritat relacional. Resultats:

- Estructures de dades de persistència creades.
- Es pot determinar si una acció de modificació o consulta es possible.

Contratemps: N/A

6.6.3 Creació del mòdul de inserció d'estàndard

Treball realitzat: S'ha creat el mòdul per a definir un estàndard en el catàleg. Resultats:

- Es pot definir un estàndard al sistema i es generen les subclasses de forma correcta.

Contratemps: N/A

6.6.4 Creació del mòdul de persistència d'estàndard

Treball realitzat: S'ha creat el mòdul per a poder persistir dades en la estructura de classes anteriorment creada.

Resultats:

- S'ha creat un adaptador JSON per a traduir objectes de JSON a persistència.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

- S'ha creat les classes per a poder guardar les dades.
- S'ha creat mètodes que eliminen les dades en cas d'error segons el definit a la classe de definició d'estàndard.
- S'ha decidit desplaçar l'etapa al seu propi mòdul.

Contratemp: N/A

6.6.5 Creació del mòdul d'etapa

Treball realitzat: S'ha decidit separa el mòdul d'etapa i donar-li el seu propi mòdul, s'ha creat el mòdul i s'han implementat els mètodes que permeten moure tant versions com estàndards pel seu cicle de vida.

Resultats:

- S'ha creat el servei que permet moure les etapes, retorna un estatus que indica l'èxit de l'operació.

Contratemp: N/A

6.6.6 Migració de metodologia

Treball realitzat: Degut a la falta de transformació de dades i al desenvolupament orientats en serveis abstracts de la producció en IRIS, s'ha decidit que es prescindirà de la producció per als mòduls, la inserció d'estàndards seguirà mantenint un mòdul per a que el usuari pugui implementar els seues propis serveis i afegir entrades al sistema.

Resultats:

- Canvi en la metodologia, es manté el test individual de cada mòdul.

Contratemp: N/A

6.6.7 Creació del mòdul de autorització

Treball realitzat: S'ha creat el mòdul d'autorització, que permet donar i retirar permisos per a usuaris d'estàndards així poden variar el seu accés a consultar o modificar l'estàndard.

Resultats:

- S'ha creat el servei de permisos.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

- S'ha creat utilitats per a retornar un estat més descriptiu de la possibilitat de l'acció segons el rol i l'estàndard.

Contratemp: N/A

6.6.8 Creació del mòdul de API

Treball realitzat: S'ha creat el mòdul d'API descrit en las següents subtasques. Cada divisió de peticions esta formada per un controlador que rep la petició i retorna el codi d'estat, i un servei que comprova els permisos i crida o implementa el respectiu servei necessitat. S'ha testat amb l'eina Postman. L'API està dividida per operacions amb l'estàndard, mòdul de permís i mòdul de l'estat de l'estàndard.

6.6.8.1 Creació del controlador general

Treball realitzat: S'ha creat el controlador que redirigirà ala resta de controladors.

Resultats:

- S'ha creat el controlador general.
- S'ha creat la "web application" en el portal de IRIS.

Contratemp: N/A

6.6.8.2 Creació dels GET

Treball realitzat: S'ha creat el controlador i el servei de obtenció de l'estàndard.

Resultats:

- S'ha creat el servei i el controlador.
- S'ha creat mètodes per a obtenir: els estàndards disponibles de l'usuari, versió d'un estàndard i instàncies d'un estàndard.

Contratemp: S'ha hagut de modificar la implementació del exportador a JSON degut a que exportava les relacions de forma errònia.

6.6.8.3 Creació del permís

Treball realitzat: S'ha creat el controlador i el servei de permís d'estàndard.

Resultats:

- S'ha creat el servei i el controlador amb els mètodes per a donar i retirar permisos.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

- Un usuari pot retirar o donar permisos si el seu usuari te l'opció habilitada.

Contratemp: N/A

6.6.8.4 Creació del POST

Treball realitzat: S'ha creat el controlador i el servei de creació de l'estàndard.

Resultats:

- S'ha permès afegir versions i instàncies, no definicions d'estàndard, es discutirà més endavant en la guia d'ús.

Contratemp: N/A

6.6.8.5 Creació del DELETE

Treball realitzat: S'ha creat el controlador i el servei de eliminació de l'estàndard.

Resultats:

- S'ha permès eliminar versions i instàncies, no definicions d'estàndard, les versions es poden eliminar sempre i quant no tinguin canvis associats es discutirà més endavant en la guia d'ús.

Contratemp: N/A

6.6.8.6 Creació del PATCH

Treball realitzat: S'ha creat el controlador i el servei de modificació de l'estàndard.

Resultats:

- S'ha permès editar camps no exclosos per la opció de variable 'skip' en instàncies, versions i definicions.

Contratemp: N/A

6.6.8.7 Creació del estat

Treball realitzat: S'ha creat el controlador i el servei per a desplaçar tant la versió com l'estàndard.

Resultats:

- S'ha permès moure l'estat de versió i estàndards.

Contratemp: N/A

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

6.6.8.8 Creació de la carga per CSV

Treball realitzat: S'ha creat la possibilitat de afegir entrades i versions a un estàndard a través de la producció d'IRIS.

Resultats:

- S'ha creat un BO(Business operation) en la producció d'IRIS, que rep una instància i la inserta en l'estàndard i versió especificats(Crea la versió si no existeix).
- S'ha creat un BS(Business service) en la producció d'IRIS que llegeix arxius txt o csv introduïts en el directori especificat en la configuració del component i crida al BO anterior per a crear l'instància, en cas d'error es comporta com se li hagi definit en la configuració de l'estàndard (propietat dropOnError)

Contratemp: N/A

6.6.8.9 Planificació de la part davantera

Treball realitzat: S'ha planificat, en conseqüència a l'API prèviament creada, les interfícies de la part davantera.

Resultats:

- Veure annex.

Contratemp: N/A

6.6.8.10 Creació del projecte Angular i inici de sessió

Treball realitzat: S'ha creat el projecte en angular i s'ha creat l'inici de sessió, com l'usuari, actualment, s'identifica amb autenticació bàsica, és fa una petició de obtenir tots els estàndards, però no es reaprofitja per a una possible adaptació a usar OAuth en el futur.

Resultats:

- L'usuari pot iniciar sessió amb el seu usuari IRIS en el sistema.

Contratemp: Error 500 en el 'Preflight' degut a comprovació CORS.

6.6.8.11 Creació de la pantalla d'estàndard i menú

Treball realitzat: S'ha creat tres components, el menú, especificat directament ja que no es veu possibilitat de reutilitzar-lo, component de detall d'objecte, component de llista d'objecte i l'especificació per a la pantalla de definició d'estàndard

Resultats:

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

- Component que fa la funció de menú
- Component genèric que explora i mostra un objecte JSON.
- Component genèric que itera sobre una col·lecció d'objectes JSON.
- Implementació específica dels anteriors components per a definició d'estàndard.

Contratemps: Hi ha hagut complicacions amb la navegació de URL i recarrega del navegador.

6.6.8.11 Creació de component d'estat d'estàndard/versió i component de notificació al usuari

Treball realitzat: S'ha creat dos components, un que mostra l'etapa del cicle de vida de l'estàndard i permet canviar-lo, i un altre que notifica d'errors i de operacions realitzades a l'usuari.

Resultats:

- Component genèric de missatgeria que dona 'feedback' al usuari de accions que s'hagin realitzat.
- Component genèric tant per a versió com per a definició d'estàndard que gestiona el cicle de vida dels mateixos.

Contratemps: N/A

6.6.8.11 Creació del component de detall i llista de versió

Treball realitzat: S'ha reutilitzat els dos components anteriors de detall d'objecte i de llista per a crear una pantalla que mostra les propietats i les instàncies associades a una versió

Resultats:

- Component de detall i llista de versió

Contratemps: N/A

6.6.8.11 Creació del component de validació

Treball realitzat: S'ha reutilitzat el component de detall de l'objecte i s'ha creat un component que permet navegar per totes les instàncies de la versió.

Resultats:

- Component de validació.
- Component de navegació de instàncies carregades.

Contratemps: N/A

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

6.6.8.12 Test parcial de LOINC

Treball realitzat: S'ha pujat una part de instàncies del estàndard LOINC a través de CSV.

Resultats:

- S'ha pogut pujar l'estàndard sense problemes

Contratemp: N/A

6.7 Problemes generals en el desenvolupament

6.7.1 Problemes en el plantejament

S'han trobat alguns problemes amb el plantejament inicial del projecte, especialment respecte a la complexitat dels estàndards i la metodologia de treball.

En quant als estàndards, primerament, es planteja un únic tipus d'instància, això s'ha trobat insuficient degut a que molts estàndards presenten no només diferents tipus d'entrades amb diferents camps sinó també extensions o versions localitzades per l'idioma, cosa que ha requerit de modificar el plantejament en quant a instàncies es refereix, permeten que una definició posseeixi una o varies varietats d'instància.

Seguidament, en quant a la metodologia, la metodologia plantejada inicialment és més funcional per un equip que per a desenvolupament de forma individual. Així doncs, s'ha continuat utilitzant un gestor de tasques, però els test de component s'han eliminat degut a testejar directament el produït en el mateix moment.

6.7.2 Problemes en la tecnologia

Iris i el llenguatge ObjectScript són una tecnologia molt potent però molt especialitzada, el domini de la mateixa no només implica un coneixement d'estructures i les particularitats d'ObjectScript sinó també el funcionament del portal de gestió.

ObjectScript al ser un llenguatge centrat en la gestió de dades ofereix molta potència en quant a integració de base de dades, gestió del canvi de taules i relacions entre entitats però per contra, és limitat en quant a tipus de classes, no tenint ni interfases ni enumeracions (Hi ha macros que realitzen una funció semblant), a més a més

Creació d'un Servidor de Catàlegs pel sector Sanitari

Desenvolupament

de poder originar problemes no relacionats amb el propi programari sinó amb el sistema en general.

Per últim, degut a una migració de IDE que esta realitzant InterSystems en el moment de redacció d'aquest treball, el corrector, detector d'errors i d'avisos del compilador de Visual Studio és limitat cosa que ha afegit més temps del esperat en moltes tasques en les fases de test i de compilació.

Creació d'un Servidor de Catàlegs pel sector Sanitari
Desenvolupament

7. Guia d'ús del producte

7.1 Limitacions del sistema

- El sistema només permet tenir relacions de 1 -> M, i només permet inserir a través de referència de la ID en el costat de M de la relació, és a dir el costat que en base de dades guarda la clau primària de l'altre objecte. Això es degut a limitacions de la classe de JSONEnabled i per a assegurar que tota instància de l'estàndard registra la pujada o eliminació de si mateixa.
- JSONEnabled contempla els objectes i les relacions com dos entitats diferents, si s'exporta un objecte amb l'opció de exportar relacions, les relacions s'exportaran en profunditat u, mentre que si la propietat d'objecte no té la opció 'deepExport' no s'exportarà. En cas de no exportar relacions cap s'exportarà.
- El sistema està pensat per funcionar, principalment amb cadenes, números, booleans i dates, degut a ser les opcions de dades més comunes en estàndards, tot i això es recomana treballar únicament amb cadenes.
- El sistema no contempla concurrència en els objectes de l'estàndard, delegant el evitar que se solapin operacions a la producció d'IRIS i a la implementació API d'IRIS.
- Degut al funcionament d'eliminació de classes d'IRIS el sistema, tot i eliminar les classes, no pot assegurar que s'hagi eliminat totes les dades y que recrear un objecte de mateix nom (per exemple un estàndard) no pugi dur problemes (Veure 7.2.4 Extinció).
- Per a motius de seguretat i aprofitant de límits d'IRIS els noms de les classes d'estàndard no poden superar els 50 caràcters.
- El sistema identifica cada instància amb una combinació del 'InstanceID', la definició d'estàndard i la versió en la qual a estat afegit provocant que, tot i que siguin de diferents tipus d'instància, dos instàncies no pugin compartir el 'InstanceID' si pertanyen al mateix estàndard i ocupen el mateix rang de versions.
- El sistema permet a un administrador canviar un estàndard de estat saltant la validació, però això pot trencar la validació.(Veure 7.2.2 Validació)

Creació d'un Servidor de Catàlegs pel sector Sanitari

Ús del producte

7.2 Operacions

7.2.1 Inserció

Irellevament del mètode usat, per a inserir un estàndard primer s'ha de crear les classes de persistència. Tot i que l'API implementa un mètode per poder crear un estàndard aquest, per seguretat és limitat en quant el que permet fer, així que es recomana invocar-lo sense propietats i, posteriorment afegir les propietats a las classe creades.

Això es pot aconseguir cridant el mètode:

```
“##class(TFG.InsertionModule.Generators.GeneratorCoordinator).createNewStandard(request As TFG.Message.Request.MedicalStandardRequest)”
```

El mètode afegeix les instàncies si no existeixen, marca com eliminades i crea una nova si han estat modificades en la versió i, per últim, marcarà com eliminades aquelles que no hagi trobat en la versió.

7.2.1.1 CSV

Per a inserir un estàndard per CSV haurem de primer haver creat el estàndard, i posteriorment copiar (important que IRIS elimina els arxius després de processar-los) un arxiu amb el nom tal que:

'[nom_standard]_[versió_standard]_[tipus d'instància].txt' al directori que el BS(Business service) estigui llegint.

El format del text es pot modificar en les opcions del BS(Business service).

L'arxiu pot contenir dos línies especials: '##header##' (obligatoria) que indicarà sense espai, posterior a la cadena indicada i separats per comes, els valors que equivalen a cada columna. I '##CloseVersion##' (opcional) que indica al sistema que aquesta versió s'ha pujat al complert i pot marca totes aquelles instàncies que apareguin en versions anteriors i no hagin estat pujades en aquesta com a eliminades.

Important que una de les columnes tingui el valor obligatori de 'InstanceID'

La resta de línies han de contenir els valors que han de prendre les instàncies sent una instància una línia i els valors separats per ','.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Ús del producte

Les relacions han de determinar la ID a la que apunten abans d'arribar al BO, al BO han de arribar com un objecte dynamic i ha de tenir el 'InstanceID' definit per separat (Utilitzar la classe de Ens.Request del paquet). (InstanceID és l'identificador de cada instància d'estàndard.)

7.2.1.2 Personalitzat

Per a afegir versions a un estàndard de forma personalitzada s'ha de, a part de prèviament haver creat el estàndard, crear un BS(Business service) que implementi el comportament en error i comprovació de l'existència de l'estàndard versió i tipus d'instància, es poden cridar als mètodes de la carpeta d'utilitats en el mòdul de 'BulkInsertion', a més a més de resoldre relacions i enviar-ho al BO(Business operation) en un format que pugui processar, recomanat usar BP(Business process) per a les transformacions.

7.2.2 Validació

Les dades poden ser validades individualment a través de la part davantera, o canviant el camp en la base de dades per a cada instància.

El sistema impedirà que una versió canviï d'estat de 'pendingValidation' a 'inUse' fins que totes les instàncies que han estat afegides o modificades en aquella versió hagin estat validades, excepte que l'estàndard tingui declarat el 'skipValidation' que farà que el sistema no ho comprovi.

7.2.3 Consulta

7.2.3.1 API

Les dades es poden consultar a través de l'API descrita en la col·lecció de Postman.

7.2.3.2 Local

Les dades es poden consultar localment a través dels serveis de l'API anteriorment anomenats o directament des de la base de dades. Si s'utilitzen les classes de 'TFG.Commons.Persistence' s'obtidran totes les instàncies sense importar a l'estàndard que pertanyin. Es pot evitar això consultant directament a les classes 'TFG.Generated', hi ha una classe de macros en les utilitats de 'Commons' de nom: 'PathOfStandard' que pot determinar la classe o taula a la qual s'ha de cridar segons el nom de l'estàndard i el nom del tipus d'instància si s'escau.

Per últim indicar que les instàncies tenen la mateixa ID a la taula de la superclasse: 'TFG.Commons.Persistence' com a la subclasse específica de l'estàndard.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Ús del producte

7.2.4 Extinció

7.2.4.1 Extinció d'estàndard

Per a poder eliminar un estàndard del sistema s'ha de, primerament moure a un estat en el qual els consumidors ja no hi pugin accedir, recomanablement durant un temps per a evitar mal funcionaments del sistema.

Seguidament s'ha de netejar totes les taules de base de dades d'aquell estàndard en particular, primerament esborrar de la taula d'autorització 'Permission' totes les instàncies d'aquell estàndard (Exemple de query): "delete from TFG_Commons_Authorization_Persistence.Permission where standardInstance = :stID"

Seguidament caldrà eliminar totes les instàncies de estàndard, seguit dels canvis de versió, per a acabar amb eliminant les versions.

Per últim cal eliminar la definició. Concluint es pot continuar eliminant les classes de la carpeta 'Generated' que pertanyen al estàndard.

Finalment s'ha de eliminar les dades que romanen a les globals d'IRIS des de el portal de gestió.

7.2.4.2 Extinció de versió

Per a poder eliminar una versió s'ha de eliminar totes les instàncies d'estàndard que tinguin ja sigui com ha 'added' o 'removed' un canvi de versió de la versió que es vol eliminar. Seguidament s'ha d'eliminar tots els canvis de versió, i finalment la versió. Pel pas final es pot utilitzar el mètode de eliminar versió del servei de eliminar de l'API, que impedirà eliminar la versió si encara existeixen canvis relacionats amb l'entitat.

7.3 Rols

El sistema separa rols a dos nivells: usuari IRIS i usuari d'estàndard.

7.3.1 Usuari IRIS:

L'usuari d'IRIS ve determinat en el portal de gestió i és necessari per a que pugui utilitzar l'API i la part

Creació d'un Servidor de Catàlegs pel sector Sanitari

Ús del producte

davantera, ha de tenir permisos per a les taules: totes les '%Dictionary', només consulta, totes les taules que inicien amb 'TFG_', consulta i edició. Si es possible és més recomanable donar aquests permisos a la 'Web application' que implementa l'API així tots els usuaris rebran els permisos necessaris.

Les credencials d'aquest usuari són les usades en l'inici de sessió de la part davantera.

7.3.2 Usuari d'estàndard

És l'usuari registrat usat en determinar l'accés a estàndards i versions, en concedir un permís a un usuari per a un estàndard és crearà automàticament.

7.4 Implantació del sistema

7.4.1 Requeriments mínims

Requeriments mínims de maquinari i programari:

- Una maquina amb una instància de IRIS 2023 o superior.
- Una maquina amb Docker.

Requeriments mínims de personal:

- Coneixements d'IRIS y bases de dades.
- Coneixements d'Angular.
- Coneixements de REST.

7.4.2 Arrencament del sistema

Per arrancar el projecte s'ha d'importar les classes a IRIS des de el portal de gestió. Seguidament s'ha de afegir el BS(Business service) i BO(Business operation) a la producció (Els XML es troben junt amb el codi).

7.4.3 Part davantera

Per a utilitzar la part davantera en Angular s'ha de crear un 'docker' de la mateixa, i afegir-ho a un contenidor en un servidor, la 'BASE_URL' de 'petition-service' ha de ser actualitzada a la del servidor en que existeixi la instància de IRIS, junt amb el prefix de la 'Web application'.

8. Possibles millores del producte

- Projecta els recursos com a instància de recurs FHIR. FHIR és un estàndard que defineix l'estructura per a consumir i/o utilitzar recursos REST relacionats amb tema mèdic, entre d'altres requeriments, FHIR requereix que es defineixin tots els components del missatge com a recursos FHIR, una opció de ampliació seria oferir les instàncies d'estàndard també com un recurs FHIR.
- Oferir més filtres a la part davantera.
- Oferir poder revisar, des de la part davantera, quines entitats del sistema estan relacionades amb quins estàndards. Així oferint una vista clara de quines classes o fluxos usen quins estàndards.
- Afegir la possibilitat d'obrir la API, exclusivament a consulta, per part del públic.
- Afegir BS(Business service) creats per a estàndards comuns.
- Crear base per a BS(Business service) establint els requeriments i funcionalitats necessàries per a que pugui servir creats per diferents estàndards pugin connectar-se amb facilitat al BO(Business operation)
- Afegir més traçabilitat opcionalment, actualment el sistema té traçabilitat limitada en quant a operacions a l'estàndard com canviar una instància, versió o moure l'estat, es pot oferir registra aquests canvis a través de crear entitats actives i marcar-les desactivades quant pateixin un canvi. Important fer-ho opcional ja que pot consumir molt d'espai en disc en poc temps en sistemes de alt volum de dades.
- Generació de índexs automàtics per a classes generades de persistència del estàndard.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Conclusions

9. Conclusió

El desenvolupament del treball ha estat dificultat pel desconeixement de la tecnologia i per una planificació que ha pecat d'optimista en varis aspectes.

S'ha subestimat la complexitat que alguns estàndards arriben, en quant a estructures, que requereixen de múltiples tipus d'instància i de camps no obligatoris.

També la dificultat en quan a l'aprenentatge d'IRIS, aquest és tot un ambient, no només el llenguatge d'ObjectScript això a provocat estancament i retards en moltes tasques.

Es creu fermament que es podria haver desenvolupat un producte més testat si s'hagués tingut coneixement previ de l'àmbit, dels estàndards mèdics o de la tecnologia d'IRIS. Tot i això s'ha aconseguit desenvolupar un producte que compleix els mínims per a ser funcional.

Creació d'un Servidor de Catàlegs pel sector Sanitari

Conclusions

Creació d'un Servidor de Catàlegs pel sector Sanitari

Bibliografia

10. Bibliografia

10.1 Cites

Govern d'Espanya. (sense data). *Real Decreto 4/2010*. Recollit de <https://www.boe.es/buscar/pdf/2010/BOE-A-2010-1331-consolidado.pdf>

10.2 Fonts

Estàndards mèdics:

Títol: Estandarización Semántica y Codificación de Dispositivos Médicos.

Autor: Govern de Colòmbia. Tipus:

Web.

Enllaç:

<https://www.minsalud.gov.co/salud/MT/Paginas/estandarizacion-dispositivos-medicos.aspx>

Títol: Políticas de Normalización y Terminología.

Autor: Govern d'Espanya.

Tipus: Web.

Enllaç: <https://www.sanidad.gob.es/areas/saludDigital/interoperabilidadSemantica/politicasNormalizacion/home.htm>

Títol: Estándares de interoperabilidad en salud: guía esencial.

Autor: Caduceus.

Tipus: Web.

Enllaç: <https://www.caduceus.es/estandares-interoperabilidad-salud/#estandares-interoperabilidad-que-son>

Interoperabilitat:

Títol: Real Decreto 4/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica.

Autor: Govern d'Espanya.

Tipus: Document.

Enllaç:

Creació d'un Servidor de Catàlegs pel sector Sanitari

Bibliografia

<https://www.boe.es/buscar/pdf/2010/BOE-A-2010-1331-consolidado.pdf>

o

<https://drive.google.com/drive/u/1/folders/1sBsoKYvEVAEzyl7xzKxkAPDbGQCMMgTV>

DICOM:

Títol: **DICOM.**

Autor: Wikipedia.

Tipus: Web.

Enllaç: <https://es.wikipedia.org/wiki/DICOM>

ICD:

Títol: **ICD.**

Autor: WHO.

Tipus: Web.

Enllaç: <https://www.who.int/standards/classifications/classification-of-diseases>

HL7:

Títol: **Estándar de mensajería HL7.**

Autor: IBM.

Tipus: Web.

Enllaç:

<https://www.ibm.com/docs/es/integration-bus/10.0?topic=formats-hl7-messaging-standard>

Títol: **Lenguaje de descripción de formato de datos (DFDL)**

Autor: IBM.

Tipus: Web.

Enllaç:

<https://www.ibm.com/docs/es/integration-bus/10.0?topic=model-data-format-description-language-dfdl>

Creació d'un Servidor de Catàlegs pel sector Sanitari

Bibliografia

Títol: Figures en Drive

Autor: Oriol Muñoz Melchor

Tipus: Carpeta en Drive.

Enllaç:<https://drive.google.com/drive/folders/1hA73vNk6vBYjBEJ-2a3gWjh4zqG65H-4?usp=sharing>