

Enginyeria del Software
Pràctiques
Curs 2024-25
UAB

Document:
S.R.S. Template
(versió 1.0)
(Data: 2024-11-22)

Historial de revisions

Date	Version	Description	Author
20/09/2023	1.0	Preparació document SRS	Equip SoftKraft
22/11/2024	2.0	Inclusió dels requisits i casos d'ús	Equip SoftKraft
22/11/2024	2.1	Creació de diagrames de casos d'ús i descripcions DCUS	Equip SoftKraft
22/11/2024	2.2	Incorporació de diagrames d'activitats i seqüència i wireframe	Equip SoftKraft
12/12/2024	2.3	Incorporació de diagrames UML i explicació de l'aplicació	Equip SoftKraft

Descripció del document

Aquest document especifica els requisits del sistema **SeniorLife**, una aplicació creada per millorar la qualitat de vida de les persones grans que viuen soles, aprofitant tecnologies modernes com els dispositius IoT i les videoconferències. Aquesta solució està orientada a abordar problemàtiques socials i mèdiques mitjançant funcionalitats que reforcen les relacions personals, proporcionen seguiment mèdic remot i fomenten la independència dels usuaris. El document segueix l'estàndard SRS (Software Requirements Specification), incloent-hi descripcions completes dels requisits del sistema, diagrames de casos d'ús i dissenys preliminars d'interfícies d'usuari.

Objectius del Document

- Detallar les funcionalitats del sistema SeniorLife, assegurant-ne el compliment dels requisits funcionals i no funcionals.
- Proporcionar una guia per al desenvolupament tècnic del sistema, incloent els casos d'ús, diagrames, i disseny d'interfícies.
- Garantir la traçabilitat dels requisits i la consistència entre la descripció funcional i el desenvolupament.
- Proporcionar una guia per al desenvolupament tècnic del sistema mitjançant diagrames UML.
- Facilitar al client una descripció del codi de l'aplicació per que entengui el funcionament

Taula de continguts

1.Requeriments de sistema	5
1.1.Requeriments d'usuari.....	5
1.2.Requirements list	5
1.3.System requirements description	5
2.Use case diagram	8
2.1.Casos d'ús seleccionats	8
3.Diagrames d'activitat i seqüència	14
3.1.Registre mèdic	14
3.2.Consultar activitat diària	16
3.3.Concertar cita metge	18
4. Diagrames UML.....	20
4.1. Diagrama Xarxes socials	20
4.2. Diagrama paràmetres salut.....	22
.....	23
5. Justificació de Bones Pràctiques en el Desenvolupament de Programari.....	24
5.1 Introducció.....	24
5.2 Bones Pràctiques Implementades.....	24
5.2.1 Ús de Patrons de Disseny	24
5.2.1.1 Patró MVC (Model-View-Controller).....	24
5.2.1.2 Patró Singleton.....	24
5.2.2 Principis SOLID	24
5.2.3 Altres Aspectes Destacats	25
5.3 Testers.....	25
5.3.1 Importància de les Proves	25
5.3.2 Tipus de Proves Implementades	25
5.3.3 Tecnologia Utilitzada.....	25
6. Links	27

1.Requeriments de sistema

1.1.Requeriments d'usuari

Llista de requeriments recollits de l'usuari:

- Requeriment 1: Interfície d'usuari bàsica
- Requeriment 2: Accessible per a persones d'edat avançada
- Requeriment 3: Disponible per a dispositius mòbils
- Requeriment 4: Servei de comunicació amb amics, familiars i voluntaris
- Requeriment 5: Monitorització dels paràmetres de salut
- Requeriment 6: Consultar estadístiques de temps d'ús de les diferents funcionalitats de l'aplicació
- Requeriment 7: Consultar posologia dels medicaments i rebre recordatoris

1.2.Requirements list

Taula de la llista de requeriments classificats segons si són funcionals o no funcionals.

Function Requirements	Non-Functional Requirements
F001 – Requeriment 4	NF001 – Requeriment 1
F002 – Requeriment 5	NF002 – Requeriment 2
F003 – Requeriment 6	NF003 – Requeriment 3
F004 – Requeriment 7	

1.3.System requirements description

Descripció completa dels requeriments de sistema:

1.3.1: Requeriments no funcionals

Requisit	NF001 – Interfície d'usuari bàsica
Descripció	La idea és que després d'una breu explicació, els usuaris siguin capaços de recordar com executar les accions principals i que aquestes es puguin executar en menys de 10 segons
Tipus	Objectiu de Disseny
Stakeholders	Carla Qurban, emprenedora i Directora Executiva

Requisit	NF002 – Accessible per a persones d'edat avançada
Descripció	L'aplicació ha d'estar preparada per a l'ús de persones amb diversitat funcional propia de gent gran, com a referencia s'hauria de prendre les indicacions del WCAG per a dispositius mòbils
Tipus	Restriccions de disseny – Acompliment dels estàndards
Stakeholders	Carla Qurban, emprenedora i Directora Executiva

Requisit	NF003 – Disponible per a dispositius mòbils
Descripció	Crear una aplicació per a tablet Android per a la gent gran i una aplicació mòbil, compatible tant amb smartphones iOS com Android i també amb la resta
Tipus	Restriccions de Disseny – Limitació Hardware
Stakeholders	Alejandra Popa, enginyera en informàtica i CTO, Antonio Garcia, familiar d'un usuari potencial

1.3.2: Requeriments funcionals

F001: Com a persona gran vull poder comunicar-me amb familiars, amics i voluntaris mitjançant missatges de text, fotografies i videoconferències per tal de millorar les meves relacions socials.

F002: Com a usuari vull que es puguin monitoritzar els meus paràmetres de salut (passos, freqüència cardíaca, etc.) mitjançant dispositius wearables per tal que es pugui fer un seguiment del meu estat de salut.

F003: Com a CFO vull poder consultar estadístiques sobre el temps d'ús de l'aplicació per cada tipus d'usuari, les funcionalitats més usades i la interacció amb els anuncis per tal de millorar el producte i justificar futures inversions.

F004: Com a persona gran o familiar vull poder visualitzar la posologia dels meus medicaments i rebre recordatoris per tal de complir amb el tractament mèdic prescrit.

1.3.3: Requeriments en conflicte

A partir de l'anàlisi elaborada anteriorment, a continuació es detallen els conflictes entre requisits que s'han identificat:

Conflicte 1: Compatibilitat amb dispositius mòbils

- **Requisits en conflicte:**
 - Alejandra Popa creu que el més adequat és crear únicament una aplicació compatible tant amb tablets com amb mòbils.
 - Antonio Garcia no vol cap restricció tecnològica sobre l'aplicació, i no vol gastar-se més diners en altres aparells electrònics.
- **Stakeholders:** Antonio Garcia (familiar d'un usuari en potència) i Alejandra Popa (CTO)
- **Raó del conflicte:** L'Alejandra creu que per que la gent gran faci servir l'aplicació només és necessari que sigui compatible amb dispositius mòbils i tablets, mentre que l'Antonio està interessat en poder fer servir l'aplicació en qualsevol

dispositiu i que no hi hagi cap restricció tecnològica. Diu que no vol gastar-se més diners en aparells.

- **Solució:** Donat que el pressupost del que es disposa no és el suficient, hem considerat que la millor opció és que l'aplicació sigui compatible només amb tablets i mòbils. Creiem que avui dia la gran majoria de gent posseeix algun d'aquests dispositius i per tant hem considerat que fent l'aplicació compatible amb ells tothom podrà fer-la servir.

Conflicte 2: Garantir la privacitat de l'usuari per sobre de tot

- **Requisits en conflicte:**
 - Antonio Garcia vol que els usuaris tinguin dret a decidir quan volen ser vigilats i quan no.
 - Pascual Peña creu que la millor funcionalitat de l'aplicació és la monitorització constant dels paràmetres vitals de l'usuari.
- **Stakeholders:** Antonio Garcia (familiar d'un usuari en potència) i Pascual Peña (metge)
- **Raó del conflicte:** Pascual Peña considera que el major benefici de l'aplicació és poder crear un perfil mèdic que permeti monitoritzar la salut física i mental dels usuaris. L'accés constant a aquestes dades garantiria un bon temps de resposta en cas de que algun dels paràmetres es surti de lo normal. Antonio Garcia, en canvia, creu que la privacitat és més important i que les persones grans haurien de poder veure qui els està monitoritzant i haurien de tenir la possibilitat de desactivar la monitorització
- **Solució:** Considerem que la privacitat és un dret essencial de tot ésser humà i no s'ha de treure aquest dret a ningú. Per tant hem decidit que en cas de que l'usuari vulgui desactivar la monitorització podrà fer-ho.

1.3.4: Requeriments dependents entre si

La **Carla Qurban, emprenedora i Directora Executiva** ens explica que abans de començar a utilitzar les funcionalitats de l'aplicació, l'usuari ha d'activar aquesta possibilitat mitjançant un registre, tal i com ho explica ella: *Això implica que abans de començar a utilitzar aquestes funcionalitats, l'usuari haurà d'activar aquesta secció mitjançant un "registre mèdic".*

Aquest requisit té una dependència amb l'explicat pel **Pascual Peña, metge i amic de Carla**: *La idea seria que, en donar-se d'alta, l'usuari respongués a un qüestionari breu sobre la seva condició mèdica. Per exemple: medicació que pren, malalties conegudes, historial.*

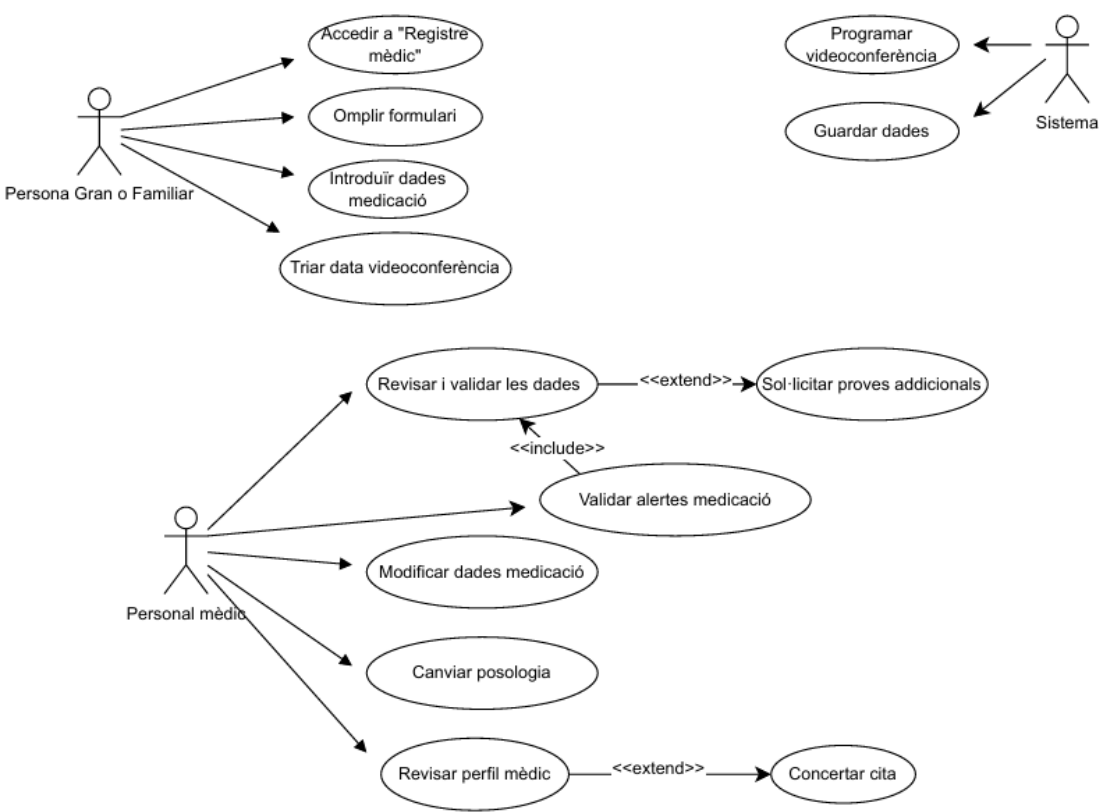
Aquests dos requisits contenen una dependència entre si ja que per tal de que el metge pugui accedir a la informació del usuari, és necessari que aquest s'hagi donat d'alta i introduït la informació. I per tant, fins que no es dugui a terme el registre, el personal mèdic no podrà accedir a la informació del usuari.

2. Use case diagram

2.1. Casos d'ús seleccionats

Nom	Actor iniciador
Realitzar una videoconferència amb un familiar	Persona gran
Omplir el formulari del registre mèdic	Persona gran o familiar
Introduir dades sobre la medicació actual	Persona gran o metge
Programar una cita amb el metge	Persona gran
Validar les dades mèdiques de l'usuari	Metge
Canviar dades de la posologia	Metge
Validar les alertes de medicació	Metge
Visualitzar posologia dels medicaments	Persona gran o familiar
Consultar estadístiques de temps d'ús de l'aplicació	CFO
Consultar dades de l'activitat diària de l'usuari	Persona gran o familiar
Desactivar el monitoratge dels dispositius IoT	Persona gran

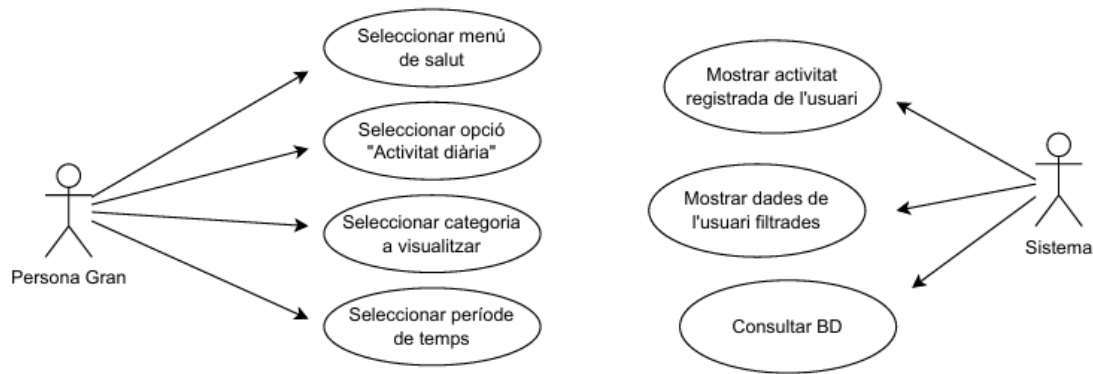
2.2.1 Realitzar el registre mèdic



CAS D'ÚS	Registre mèdic
Versió	1.0
Data	21/11/2024
Autors	Equip SoftKraft
Descripció	Aquest cas d'ús descriu com una persona gran o el seu familiar pot completar el registre mèdic necessari per activar les funcionalitats de monitoratge de salut. Inclou l'ompliment d'un formulari, la introducció de la medicació actual i la videoconferència amb personal mèdic per validar les dades.
Actors	Persona gran, Familiar, Personal mèdic, Sistema.
Precondició	L'usuari ha creat un perfil social a l'aplicació. L'usuari disposa d'un dispositiu compatible per realitzar la videoconferència.
Flux principal	1. L'usuari accedeix a l'opció "Registre mèdic" al menú principal. 2. Omple el formulari amb informació personal i de salut.

	<p>3. L'usuari introdueix les dades sobre la seva medicació actual.</p> <p>4. Les dades es guarden a la BD</p> <p>5. Tria una data per a una videoconferència amb el personal mèdic.</p> <p>6. Durant la videoconferència, el personal mèdic revisa i valida la informació.</p> <p>7. El personal mèdic configurarà els paràmetres a monitoritzar i les alertes associades.</p> <p>8. Si tot és correcte, el sistema confirma que el registre mèdic ha estat completat i activa les funcionalitats de monitoratge de salut.</p>
Subfluxos	<p>1. El personal mèdic configurarà paràmetres específics segons les necessitats del pacient durant al videoconferència.</p>
Fluxos alternatius	<p>Falta de dades: Si el personal mèdic considera que no te prou informació en el formulari, realitzarà tantes preguntes com cregui necessari i introduirà les dades en el sistema</p> <p>Falta d'informació: Si el personal mèdic considera que es necessita mñes informació o proves per poder validar el perfil mèdic demanarà a l'usuari que concertí un cita.</p> <p>Canviar posologia: Si el personal mèdic ho considera necessari té la possibilitat de canviar la posologia dels medicaments del pacient</p>
Postcondició	<p>El registre mèdic queda complert i el sistema activa totes les funcionalitats de salut de l'aplicació.</p>
Requeriments no funcionals	
Prioritat	Urgent
Comentaris	<p>Aquest cas d'ús és essencial per activar les funcionalitats de salut del sistema.</p>

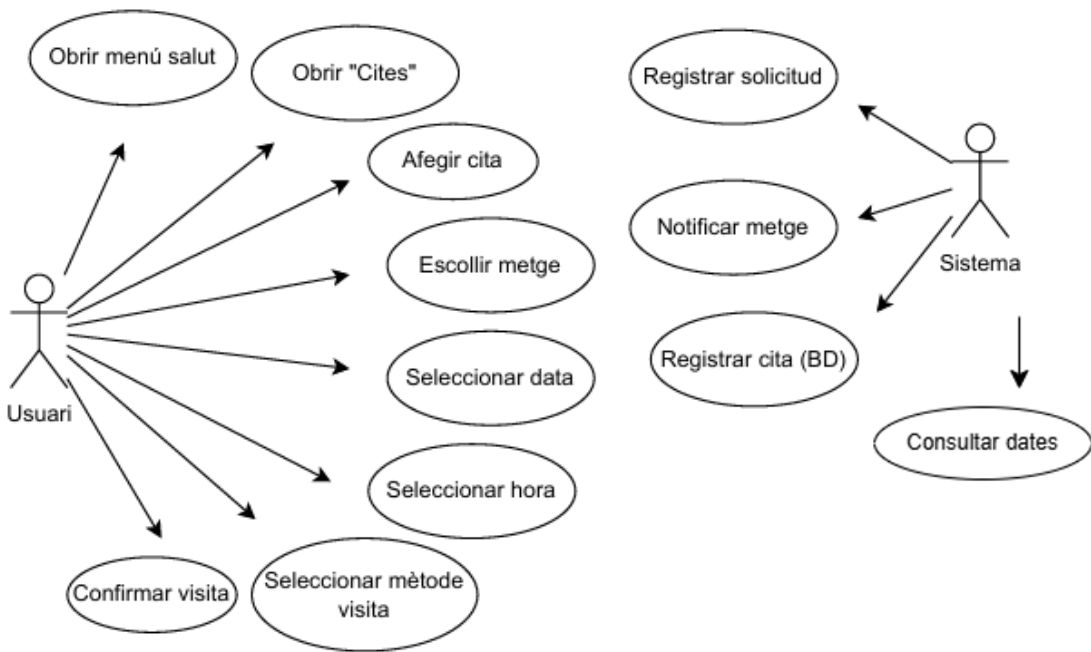
2.2.2 Consultar activitat diària



CAS D'ÚS	Consultar dades de l'activitat diària de l'usuari
Versió	1.0
Data	21/11/2024
Autors	Equip SoftKraft
Descripció	Aquest cas d'ús permet a la persona gran o al seu familiar consultar les dades recollides sobre l'activitat diària, incloent passos, hores de son, freqüència cardíaca i altres paràmetres monitoritzats pels dispositius IoT.
Actors	Persona gran, Familiar, Sistema.
Precondició	1. L'usuari ha iniciat sessió al sistema (Ha completat el registre mèdic). 2. Els dispositius IoT han registrat dades vàlides (No poden estar desconnectats)
Flux principal	1. L'usuari selecciona el "Menú de salut" 2. L'usuari selecciona l'opció "Activitat diària" al menú principal. 3. El sistema consulta les dades de l'usuari a la BD 3. El sistema mostra un resum general de l'activitat registrada (passos, son, etc...). 4. L'usuari pot seleccionar categories específiques per veure'n els detalls. 5. El sistema carrega les dades filtrades.

	6. L'usuari consulta la informació i tanca la vista.
Subfluxos	1. Visualitzar un període específic de temps. 2. Comparar dades amb períodes anteriors.
Fluxos alternatius	Falta de dades: Si no hi ha dades disponibles es mostra un missatge informatiu a l'usuari.
Postcondició	
Requeriments no funcionals	Temps de resposta màxim de 2 segons per a consultes estàndard.
Prioritat	Normal
Comentaris	

2.2.3 Concertar cita metge



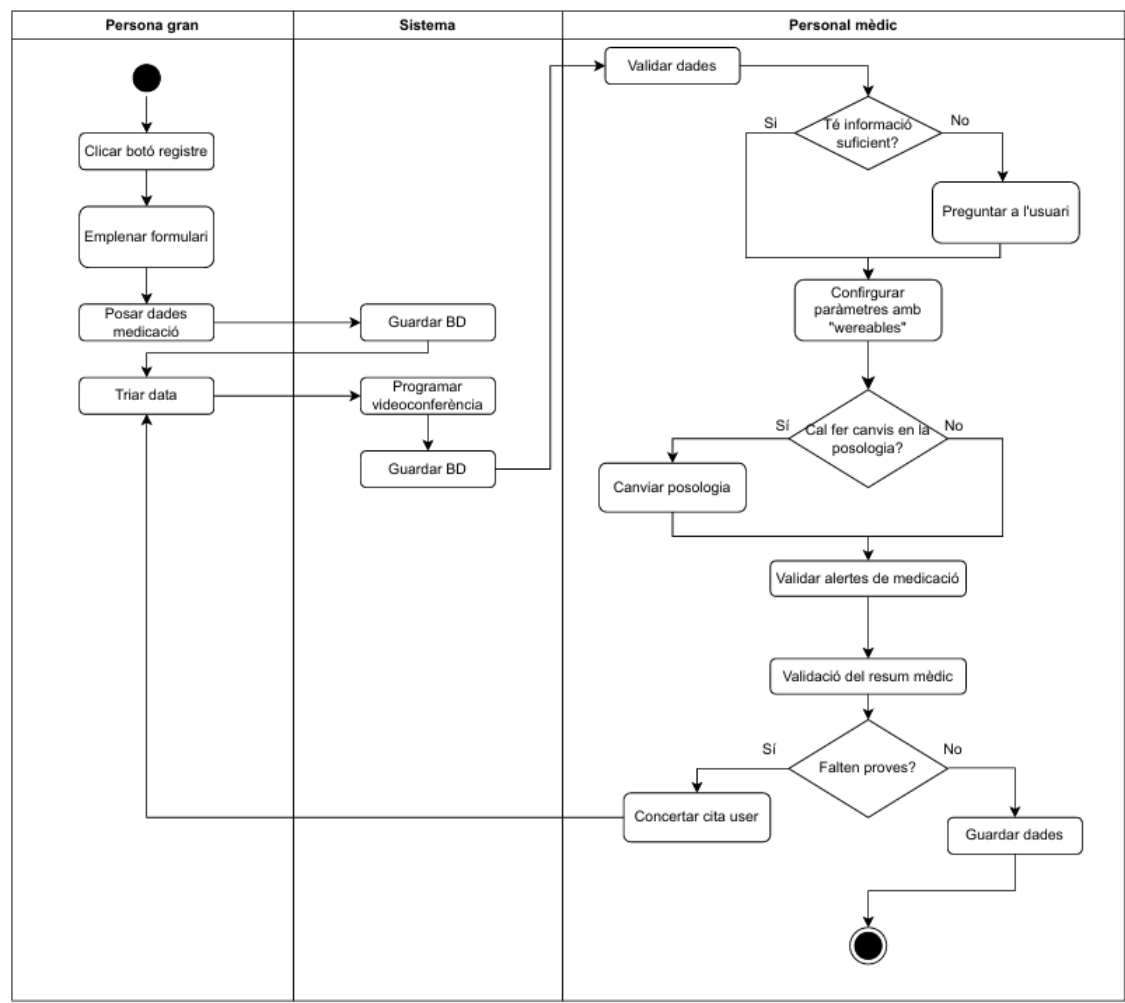
CAS D'ÚS	Sol·licitar cita amb el metge
Versió	1.0
Data	21/11/2024

Autors	Equip SoftKraft
Descripció	Aquest cas d'ús descriu com una persona gran o el seu familiar pot sol·licitar una cita amb un metge mitjançant l'aplicació, seleccionant una data disponible per a consultes presencials o virtuals.
Actors	Persona gran, Familiar, Sistema
Precondició	1. L'usuari ha iniciat sessió al sistema (Ha completat el registre mèdic).
Flux principal	<ol style="list-style-type: none"> 1. L'usuari selecciona l'opció "Salut" al menú principal. 2. L'usuari selecciona l'opció "Cites" al menú de salut 3. L'usuari selecciona "Afegir cita" 4. El sistema demana a l'usuari que esculli metge per la visita 5. L'usuari selecciona el metge amb el que es vol visitar 6. El sistema consulta les dates disponibles a la BD 7. El sistema mostra un calendari amb les dates disponibles. 8. L'usuari selecciona una data 9. El sistema mostra les hores disponibles per aquella data 10. L'usuari selecciona la hora de la cita 11. El sistema demana a l'usuari que esculli entre visitar-se en persona o online. 12. L'usuari selecciona el mètode amb que es vol visitar. 13. L'usuari confirma la cita 14. El sistema registra la sol·licitud. 15. El sistema registra la cita a la BD 15. El sistema notifica al metge sobre la cita programada.
Subfluxos	
Fluxos alternatius	Cap cita disponible: Si no hi ha dates disponibles no es mostrarà per pantalla cap data
Postcondició	La cita queda registrada i s'envia una notificació al metge.
Requeriments no funcionals	
Prioritat	Normal
Comentaris	

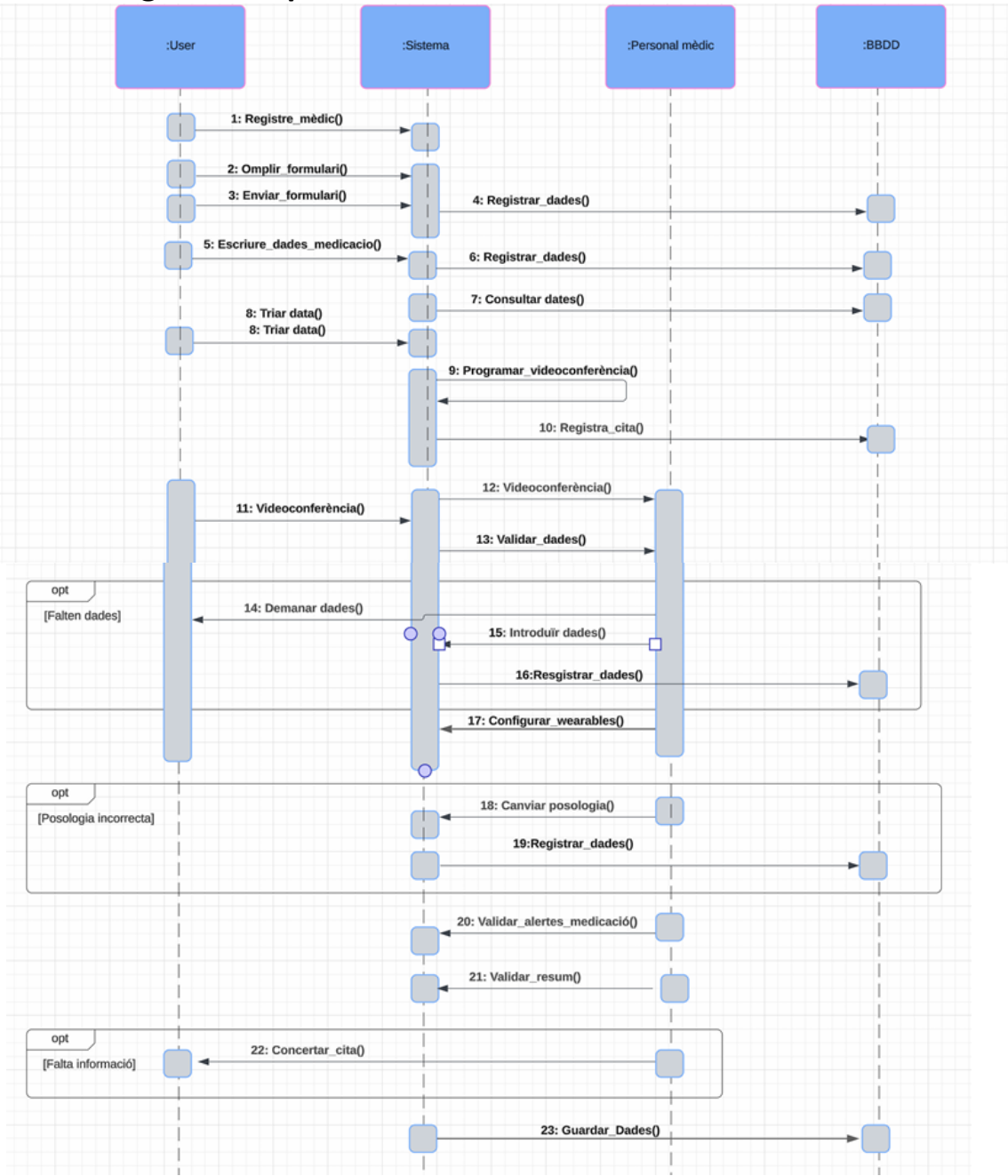
3.Diagrames d'activitat i seqüència

3.1.Registre mèdic

3.1.1: Diagrama Activitats

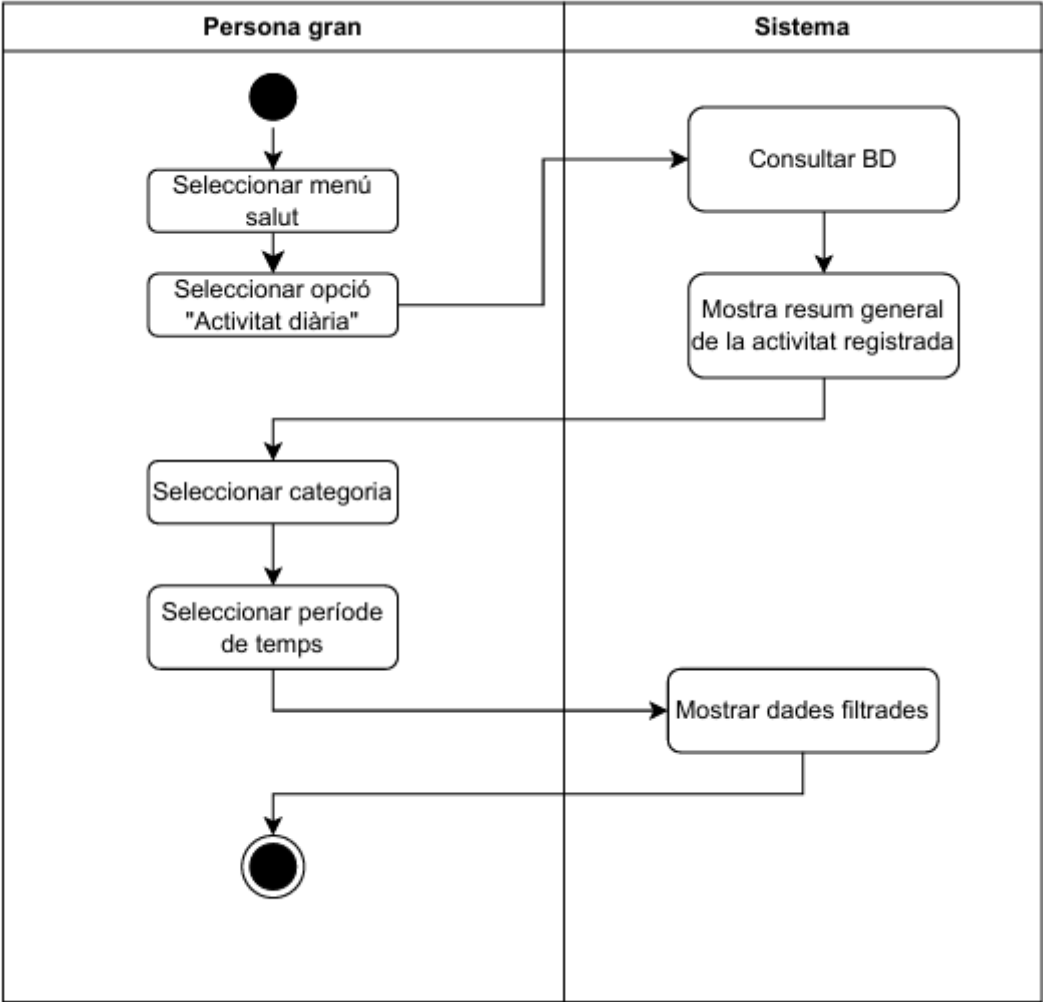


3.1.2: Diagrama Seqüència

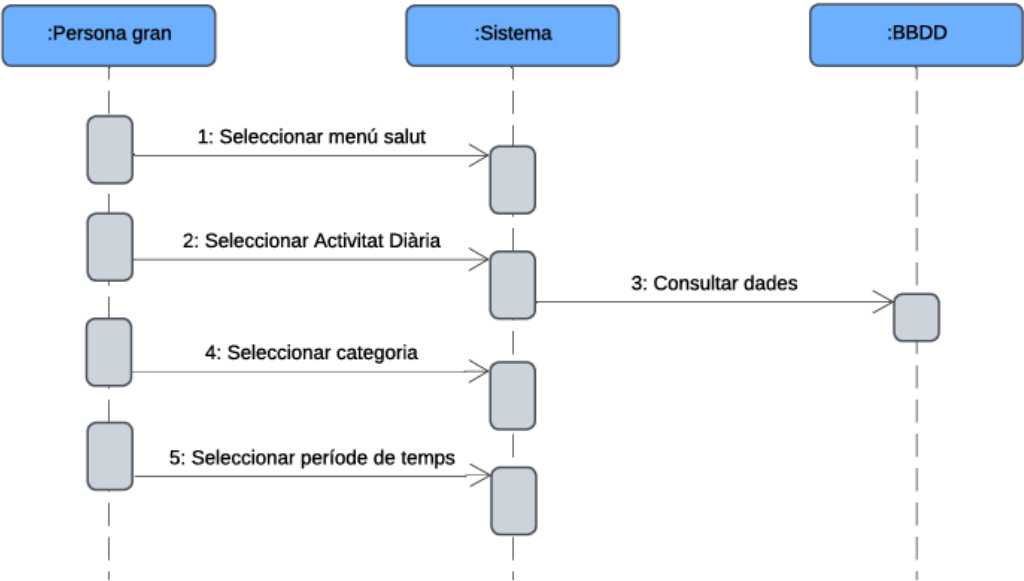


3.2.Consultar activitat diària

3.2.1: Diagrama Activitats

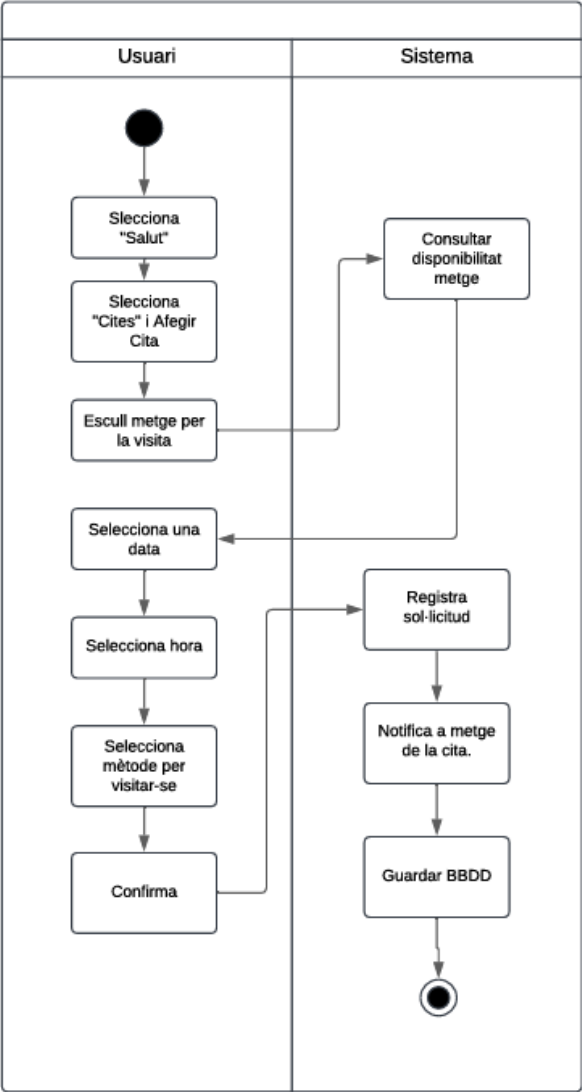


3.2.2: Diagrama Seqüència

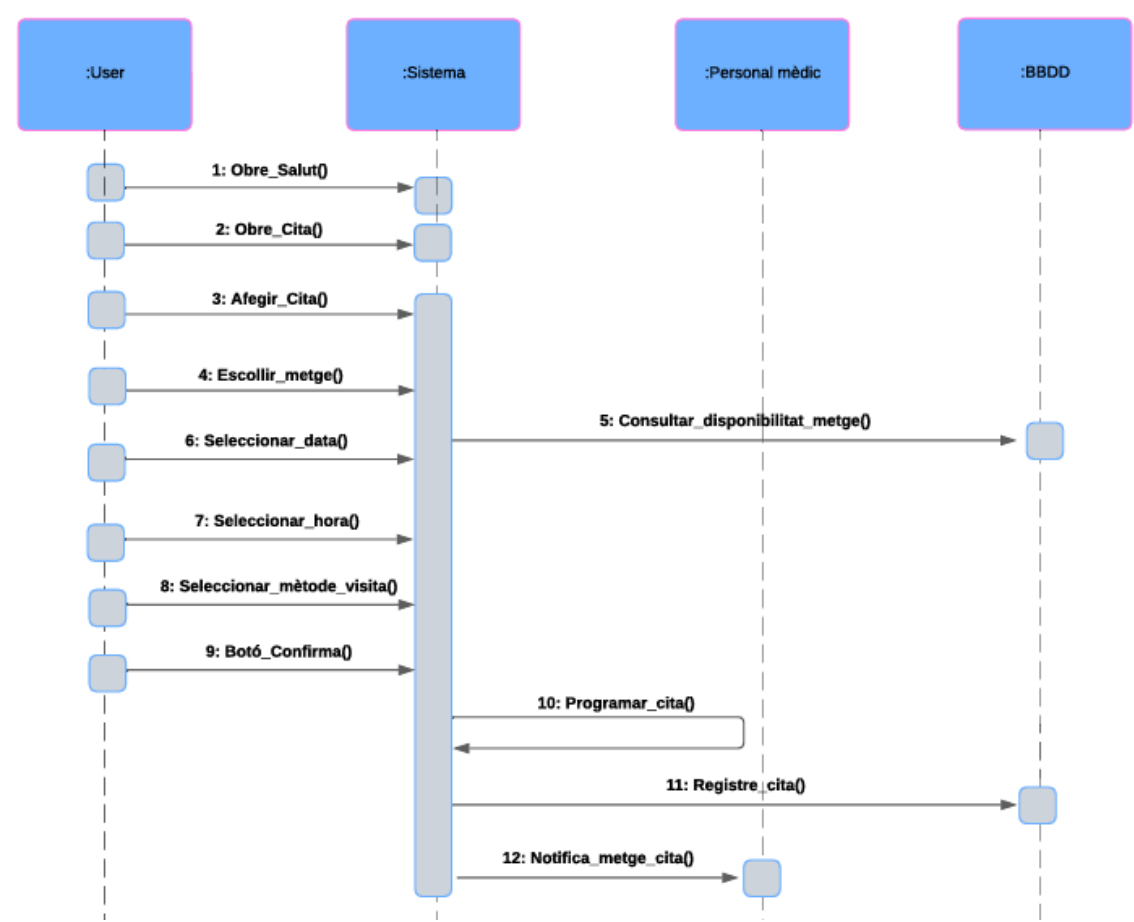


3.3.Concertar cita metge

3.3.1: Diagrama Activitats



3.3.1: Diagrama Seqüència



4. Diagrames UML

4.1. Diagrama Xarxes socials

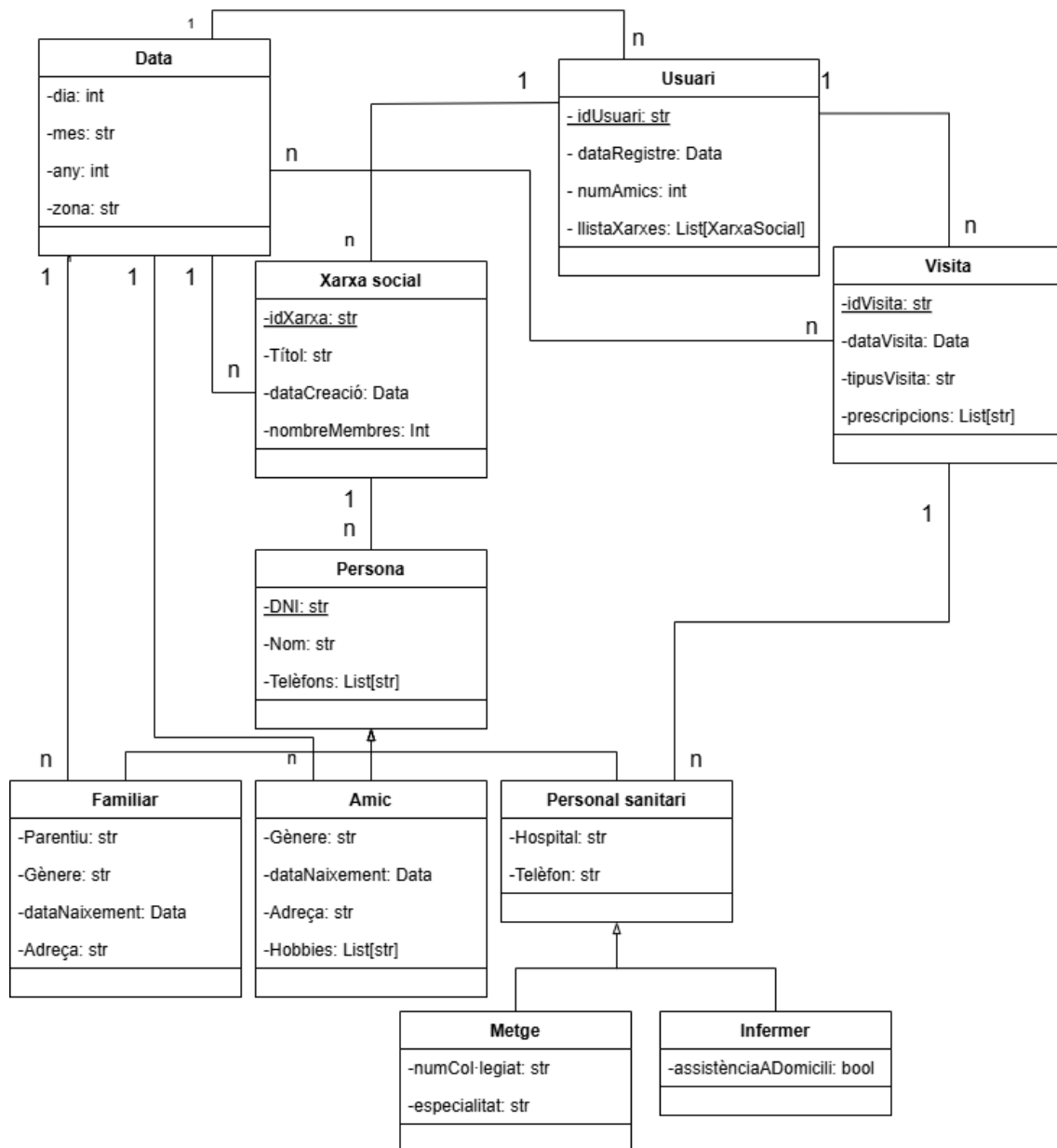
Aquest primer diagrama hem establert les classes especificades pel client. La classe **Xarxa Social**, que conté l'identificador de la xarxa, el títol d'aquesta, la data de creació de la xarxa i finalment el nombre de membres que formen part.

La següent classe creada és **Usuari** amb l'identificador del usuari, la data en que es va registrar, la llista de xarxes socials de les que forma part el client i finalment, el nombre d'amics que té el usuari. Aquest últim atribut ha sigut afegit com a petició del client.

La classe **Visita**, conté l'identificador de la mateixa, la data en que es va fer la visita, el tipus i la llista de prescripcions donades pel metge.

En quant a la classe **Persona** volem guardar el seu DNI, nom i telèfons si en té més d'un, i per cada tipus de relació del usuari amb la persona es dona una relació d'herència amb les subclasses **Familiar**, **Amic** i **Personal Sanitari**. Del familiar volem guardar a més dels atributs d'herència, el parentiu, gènere, data de naixement i finalment l'adreça. Igualment amb la classe amic, guardarem la seva data de naixement i adreça, gènere i una llista amb els seus "hobbies". Finalment, la classe personal sanitari és una herència amb les subclasses **Metge** i **Infermer**, de les quals volem guardar, a més del hospital al que treballa i el telèfon, el número de col·legiat i especialitat en el cas del metge i per l'infermer un atribut que indiqui si realitza assistència a domicili.

Per últim, cal afegir que la classe Data s'ha creat com a especificació del client ja que d'aquesta manera podem guardar les dates en el format correcte i demanat.

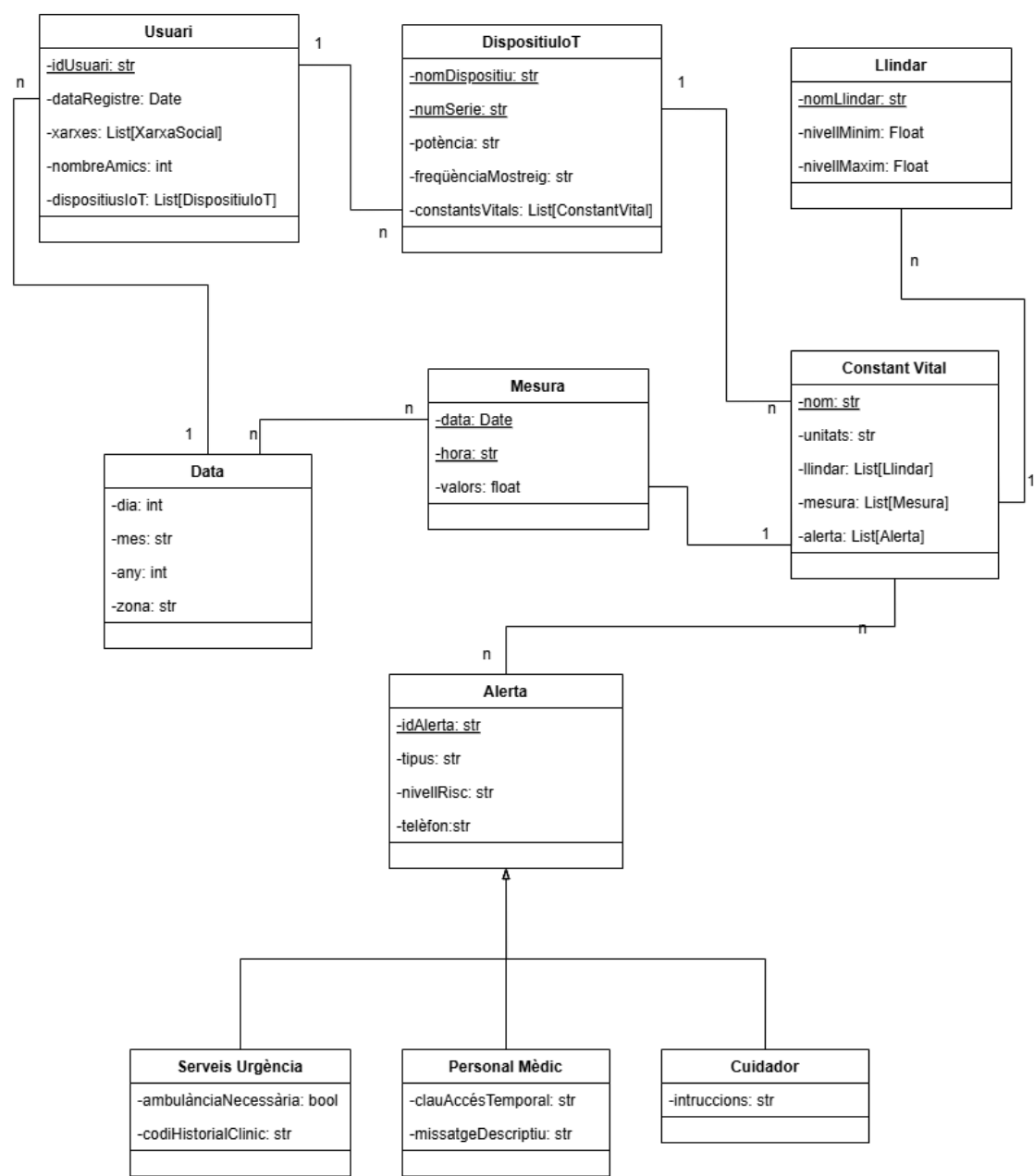


4.2. Diagrama paràmetres salut

Per aquest segon diagrama afegim a la classe **Usuari** una llista amb els dispositius IoT que té. A més a més, creem la classe d'aquests **dispositius IoT** amb el nombre i número de sèrie, la seva potència i freqüència de mostreig, i com a últim paràmetre, una llista de constants vitals que mesura.

Parlant de mesura, hem creat una classe **Mesura** amb la data, hora i valors que va prendre. La classe **Constant Vital** disposa de paràmetres com el nom de la constant, les seves unitats, el llindar, la mesura i també, l'alerta.

Com a classe **Alerta** tenim una relació d'herència amb els diferents tipus d'alertes de que disposa l'aplicació: els **Serveis d'Urgència, Personal mèdic i Cuidador**. Aquestes subclasses contenen els atributs d'herència d'alerta que són l'identificador de la alerta, el tipus, el nivell de risc i un telèfon. A més d'aquests atributs cada classe conté els seus propis: serveis d'urgència conté una variable de necessitat d'ambulància i el codi de l'historial clínic. Personal mèdic conté la clau d'accés temporal i un missatge descriptiu. Finalment, el cuidador conté una variable amb les instruccions a seguir. Aquest diagrama també conté la classe data ja que guardem les dates en el format especificat pel client



5. Justificació de Bones Pràctiques en el Desenvolupament de Programari

5.1 Introducció

A la nostra empresa adoptem bones pràctiques d'enginyeria de programari per garantir que les nostres aplicacions siguin d'alta qualitat, escalables i fàcils de mantenir. Aquest informe explica les decisions preses durant el desenvolupament, ressaltant els patrons de disseny aplicats i justificant l'ús de proves automatitzades.

5.2 Bones Pràctiques Implementades

5.2.1 Ús de Patrons de Disseny

5.2.1.1 Patró MVC (Model-View-Controller)

Al fitxer `classes.py`, es va organitzar el codi seguint el patró MVC per dividir les responsabilitats de l'aplicació:

- **Model (Model):** Conté la lògica de negoci i l'estructura de dades.
- **View (Vista):** S'encarrega de la interfície i la interacció amb l'usuari (encara que en aquest cas no es defineix directament al fitxer analitzat).
- **Controller (Controlador):** Coordina les accions entre la vista i el model.
- Aquesta separació millora l'organització del codi, facilita les proves i permet realitzar canvis en un component sense afectar els altres.

5.2.1.2 Patró Singleton

Al fitxer `aplicació.py`, es va utilitzar el patró Singleton per assegurar que certes classes (com gestors de configuració o serveis compartits) tinguin una única instància a tot el sistema. Això redueix el risc d'inconsistències i optimitza l'ús de recursos.

5.2.2 Principis SOLID

Es van implementar els principis SOLID per garantir un disseny modular i sostenible:

- **Responsabilitat única:** Cada classe té un propòsit clar.
- **Obert/Tancat:** Les classes es poden estendre sense necessitat de modificar el seu codi original.
- **Inversió de Dependències:** Es prioritzen les abstraccions sobre les implementacions concretes, reduint el acoblament entre components.

5.2.3 Altres Aspectes Destacats

- **Documentació:** Els comentaris al codi expliquen clarament les funcions i classes, facilitant-ne la comprensió.
- **Control de Versions:** El projecte està allotjat a GitHub, cosa que permet registrar canvis, col·laborar amb l'equip i revertir errors si cal.
- **Estàndards de Codificació:** Se segueix l'estàndard PEP8 a Python, garantint codi net i llegible.

5.3 Testers

5.3.1 Importància de les Proves

Les proves automatitzades són essencials perquè:

- Ajuden a detectar errors en les primeres etapes del desenvolupament.
- Asseguren que el programari compleixi amb els requisits definits.
- Eviten que errors antics reapareguin (regressions).
- Milloren la confiança en el sistema provant escenaris crítics.

5.3.2 Tipus de Proves Implementades

El fitxer `tester.py` conté diverses proves dissenyades per validar el funcionament correcte del sistema:

5.3.2.1 Proves Unitàries

Avalen cada mètode i classe de manera individual per assegurar que compleixen la seva funció correctament. Són el primer nivell de defensa contra errors.

5.3.2.2 Proves d'Integració

Verifiquen que els diferents mòduls de l'aplicació funcionin conjuntament de manera correcta. Això és vital per detectar problemes de comunicació entre components.

5.3.2.3 Proves de Casos Límit

Simulen situacions extremes o atípiques per comprovar com el sistema gestiona entrades inesperades.

5.3.3 Tecnologia Utilitzada

Es va utilitzar el framework `unittest` de Python per automatitzar les proves. Aquest framework permet:

- Crear i organitzar casos de prova.
- Automatitzar-ne l'execució.
- Generar informes detallats amb els resultats.

5.4 Conclusió

L'ús de bones pràctiques de desenvolupament, com el patró MVC, el patró Singleton i els principis SOLID, assegura que el programari sigui robust, mantenible i escalable. A més, la implementació de proves exhaustives garanteix que el sistema compleixi amb els requisits establerts, minimitzant riscos i millorant l'experiència de l'usuari. Aquestes estratègies reforcen la qualitat del producte final i optimitzen el procés de desenvolupament.

6. Links

Jira:

<https://lauragispertcortes1.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog>