



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Sistema de monitoratge autoadaptable heterogeni i distribuït

Informe de seguiment

Autor:

Joaquim MOTGER DE LA
ENCARNACIÓN

Director:

Xavier FRANCH GUTIERREZ

Codirector:

Marc ORIOL HILARI

Treball de final de grau presentat sota el marc del

Grau d'Enginyeria Informàtica

en l'especialitat de

Enginyeria del Software

Índex

1	Introducció	1
2	Contextualització	3
2.1	Presentació i justificació de la temàtica	3
2.1.1	Identificació dels stakeholders	4
2.2	Estat de l'art	4
2.3	Projecte SUPERSEDE	6
3	Gestió del desenvolupament i alternatives	9
3.1	Metodologia de desenvolupament	9
3.2	Planificació temporal	9
3.3	Previsió d'alternatives i pla d'acció	10
4	Integració de coneixements	13
5	Integració de lleis i regulacions	15
6	Conclusions	17
	Bibliografia	19

Capítol 1

Introducció

El present document consisteix en l'informe de seguiment del Treball de Final de Grau (TFG) del Grau en Enginyeria Informàtica titulat *Sistema de monitoratge autoadaptable, heterogeni i distribuït*. Com a document presentat durant el procés de desenvolupament del mateix projecte, els objectius principals d'aquest en són dos.

En primer lloc, establir una "fotografia" documentada del projecte: és a dir, projectar en quin estat es troba el projecte, quina és la feina que s'ha realitzat fins al moment, en quin punt del desenvolupament es troba, i quines són les tasques que, d'acord amb la contextualització i plantejaments presentats, queda per fer. D'aquesta manera, aquest document pretèn reflectir la realitat del projecte en el moment de la seva presentació.

En segon lloc, aquest document serveix de garantia pel desenvolupament i finalització del projecte. D'acord amb els terminis establerts, i a les tasques realitzades i per realitzar, els detalls del treball realitzat i les consideracions pertinents han de ser una prova de la viabilitat de presentació del projecte.

Per englobar aquests objectius, es presenten primerament la contextualització i detalls de l'entorn on desenvolupem el treball, així com la informació relacionada amb la gestió i planificació en relació a tot allò establert a GEP. En aquest sentit, s'inclouen consideracions com el sorgiment de desviacions i alternatives, i la integració dels coneixements i aspectes teòrics que han calgut considerar pel desenvolupament del projecte.

Capítol 2

Contextualització

2.1 Presentació i justificació de la temàtica

En les darreres dècades els sistemes software han evolucionat fins al punt d'esdevenir elements clau i imprescindibles de les activitats primàries de qualsevol empresa, organització o institució. La gestió de la informació, els protocols i controls de seguretat, els processos de negoci, etc., són els reptes als quals els Chief Information Officers (CIO) de moltes empreses s'han d'enfrontar. Aquests reptes i els seus resultats depenen, en gran mesura, del comportament dels sistemes software que entren en joc dins aquestes activitats. Addicionalment, la quantitat de productes software exposats com a serveis o aplicacions mòbils ha incrementat dràsticament. Fet que deriva en el sorgiment d'una gran varietat de contexts i entorns d'execució entre els grans volums d'usuaris que aquests sistemes poden tenir.

És per aquest motiu que eventualment ha anat prenent força un concepte basat en l'estudi i control de qualitat dels sistemes software: el monitoratge. Com a part de la vida professional d'un enginyer de software, la supervisió i control dels components i sistemes amb els què treballa és un concepte clau amb el qual, d'una forma o altra, ha d'estar familiaritzat. Però el problema que plantegem aquí va més enllà: després del repte de monitorar els sistemes, ens hem de plantejar com dissenyar, gestionar i adaptar aquest monitoratge.

Els reptes que aquestes tasques plantegen i que aquest projecte treballa són diversos. Entre d'altres, cal valorar el disseny i les característiques tècniques dels monitors, la seva configuració i la capacitat d'adaptabilitat. En relació amb aquest últim aspecte, també cal valorar com s'emmagatzemen i es gestionen els detalls relacionats amb la configuració dels monitors, i establir interaccions de la manera més genèrica possible per facilitar l'extensibilitat.

Tal i com veurem més endavant a l'apartat 2.3. *Estat de l'art*, existeix una àmplia recerca que actualment treballa i desenvolupa projectes en relació a aquest àmbit. El potencial d'estudi que ofereix resulta d'un alt interès a causa de la possibilitat de recerca i síntesi i als diferents aspectes i criteris sobre els quals es pot treballar.

Així, tant com estudiant com a futur professional del sector de l'enginyeria del software, es poden contemplar diversos criteris per treballar en aquesta temàtica:

- Un aprofundiment en els coneixements de l'enginyeria i els sistemes software
- Treball i recerca en conceptes de control de qualitat, fiabilitat i millora de l'experiència de l'usuari

- Possibilitat de col·laborar i aprofundir en un tema de recerca d'actualitat dins l'enginyeria de serveis i els sistemes d'informació
- Plantejament d'un projecte complet que pugui servir a tercers interessats en l'estudi de sistemes de monitoratge autoadaptatius

2.1.1 Identificació dels stakeholders

Les diferents fases que engloben aquest projecte deriven en l'obtenció d'un producte final, orientat a la seva aplicació pràctica. Com a tal, els documents generats i els components dissenyats i implementats esdevenen productes propis dins el context del monitoratge de sistemes software. Com a tals, aspectes que s'exposaran al llarg d'aquest document (el disseny i implementació d'una arquitectura genèrica pels monitors, la gestió de les configuracions, etc.) poden resultar d'utilitat per a agents externs a la pròpia autoria del projecte.

Podem considerar, per tant, que podrà ser una eina d'interès pels principals stakeholders, que vindrien a ser:

- **Desenvolupadors i enginyers software.** Aquells agents al càrrec del control de qualitat de sistemes softwares de diverses naturaleses. Els conceptes treballats, el plantejament de problemàtiques, i el producte generat, poden aportar valor de qualitat al sector treballant, per una banda, en la síntesi i recopilació de la informació actual, i per altra banda, aportant propostes i solucions pròpies basades en l'experiència del desenvolupament del projecte.
- **Gestors de projecte i experts en Sistemes d'Informació.** La gestió de la informació, el tractament i el seu potencial poden resultar afectats gràcies a la capacitat de recol·lecció de dades del sistema de monitoratge, així com els criteris de revisió i control de qualitat, que permeten analitzar i obtenir informació fiable.
- **Usuaris finals dels sistemes monitorats.** De forma indirecta, es veuran afectats degut a les conseqüències del monitoratge dut a terme per aquest sistema de monitoratge o d'altres derivats dels conceptes treballats al llarg d'aquest projecte.

2.2 Estat de l'art

Per tal de plantejar les necessitats i projeccions del treball, així com les vies de desenvolupament del projecte, és necessari conèixer quina és la situació del monitoratge autoadaptatiu de sistemes software en la recerca actual.

En primer lloc, cal conèixer l'entorn referent als sistemes software autoadaptatius: és a dir, aquells que seran l'objectiu de monitoratge del nostre sistema de monitoratge (que no deixa de ser un altre sistema software autoadaptatiu). És interessant veure com la recerca i la investigació planteja, de forma pràcticament correlacionada, els conceptes de sistema autoadaptatiu i monitoratge.

De fet, documents de caràcter acadèmic tals com [1] plantegen una arquitectura d'autoadaptabilitat per sistemes software basada en la supervisió o monitoratge. Tal com podem veure a la Figura 2.1, destaquem dos components principals: l'aplicació

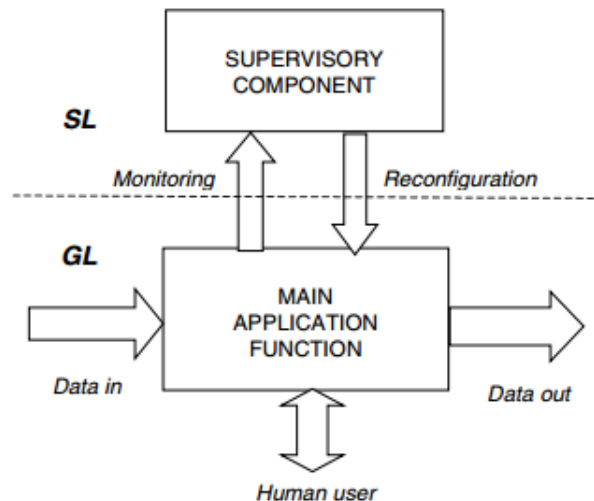


FIGURA 2.1: Arquitectura de sistemes autoadaptables monitorats

principal, o **Main Application Function**, equivalent al sistema software (servei web, aplicació mòbil, etc.) encarregat d'una funcionalitat específica i sobre el qual volem realitzar el control de qualitat; i el component supervisor, o **Supervisory Component**, que s'encarrega d'executar aquest control de qualitat.

La interacció i arquitectura plantejada en la figura és relativament senzilla. El component software a avaluar forma part del domini del sistema a avaluar (*ground-level*, GL). En la seva activitat, aquesta aplicació o sistema rep una sèrie de dades (que dependrà de la naturalesa i objectiu del sistema), i produeix uns resultats en base a aquest *input*. Paral·lelament, com qualsevol sistema orientat a l'ús, existeix una interacció sistema-usuari. Més enllà del domini i les característiques pròpies dels sistemes software, es presenta l'entorn corresponent al sistema de supervisió o monitoratge (*supervisory-level*, SL). En aquest nivell trobem des d'un punt de vista lògic el component de supervisió o monitor, que interacciona amb l'aplicació principal de dues formes diferents:

- **Monitoratge.** Procés d'interacció entre el component principal i el monitor on el primer envia informació al segon. L'aplicatiu produeix, com a conseqüència de la seva activitat, informació i dades que envia com a *output* al monitor. Aquesta informació ha d'estar prèviament definida i estructurada, de tal manera que el monitor sigui capaç d'interpretar-la.
- **Reconfiguració.** El monitor procesa les dades que rep com a input del monitoratge i, segons els criteris d'adaptabilitat establerts, aplica els canvis o reconfiguracions pertinents en el sistema software monitorat. D'aquesta manera, la lògica encarregada d'interpretar la informació i prendre decisions d'adaptabilitat (SL) queda totalment separada de la lògica del domini de l'aplicatiu (GL).

Partint d'aquest punt, aquest document estableix les premisses del monitoratge i les necessitats de reconfiguració d'aquests tercers sistemes softwares, la naturalesa dels quals és diversa i heterogènia. Noti's que, de fet, l'anterior disseny presenta una arquitectura genèrica aplicable a qualsevol entorn software, independentment de la naturalesa del domini monitorat. Tot i així, existeix una àmplia recerca especialitzada: publicacions com [3] es centren en analitzar els requisits, necessitats i funcions

principals del monitoratge i reconfiguració de sistemes i recursos al núvol (p.e. serveis web).

La documentació i bibliografia referent als sistemes software autoadaptables i als sistemes de supervisió i monitoratge és molt àmplia. Tot i així, si focalitzem la recerca a la problemàtica a resoldre, és a dir, l'autoadaptabilitat i reconfiguració dels sistemes de monitoratge, no trobem un treball tan profunditzat i específic com en el cas prèviament explicat.

En alguns documents, com per exemple el ja esmentat publicat per la University of Nashville, es mencionen criteris del disseny de la capa de supervisió, entre els quals entren en joc, p.e., factors com la previsió d'errors a la capa de l'aplicatiu principal, o ve esdeveniments/disparadors inesperats. I, com és lògic, una reacció i canvis per part del sistema de monitoratge.

En altres documents, tals com [2] es defineixen models autònoms d'autoadaptabilitat i reconfiguració autònoma, basats en tècniques de detecció d'esdeveniments i canvis en el propi sistema. Conceptes com l'anàlisi de l'estat del sistema, el monitoratge i l'execució d'una reconfiguració entren en joc dins d'aquest pla.

2.3 Projecte SUPERSEDE

Com a part de l'estat de l'art i punt de partida pel desenvolupament del sistema, cal introduir el projecte SUPERSEDE (<https://www.supersede.eu/>). Aquest projecte forma part del *Horizon 2020 Programme*, un programa de recerca i innovació finançat i gestionat per la Unió Europea. Actualment, compta amb la participació de diverses empreses, fundacions i universitats, entre les quals s'inclou la pròpia UPC.

Aquest projecte planteja una proposta del cicle de vida i la gestió dels serveis software i les aplicacions, amb l'objectiu final similar al plantejat com a premisa d'aquest Treball de Final de Grau: millorar la qualitat de l'execució dels sistemes software i, en conseqüència, l'experiència de l'usuari final en l'interacció amb aquests sistemes.

Dins aquest cicle de vida, orientat al control de qualitat dels sistemes software, es proposen 4 fases:

1. **Col·lecció.** L'obtenció i emmagatzematge de dades que puguin resultar d'interès pel control de qualitat. La naturalesa d'aquestes dades (així com el format i altres criteris) dependran de l'objectiu d'aquest anàlisi i el tipus de dades tractat. Així, aquestes poden incloure desde dades purament analítiques (p.e. % de disponibilitat del sistema) o bé contextuais (p.e. missatges o continguts introduïts al sistema).
2. **Anàlisi.** Les dades obtingudes en la fase anterior tenen un significat, una informació que el sistema ha de ser capaç d'extreure i comprendre. En aquesta fase, les dades es transformen en coneixement en relació a l'estat del sistema, a través de diverses tècniques analítiques, de nou en funció del marc d'estudi i el context. Per exemple, es podrien valorar tècniques d'anàlisi de llenguatge natural per estudiar les valoracions d'usuaris introduïdes a un sistema.

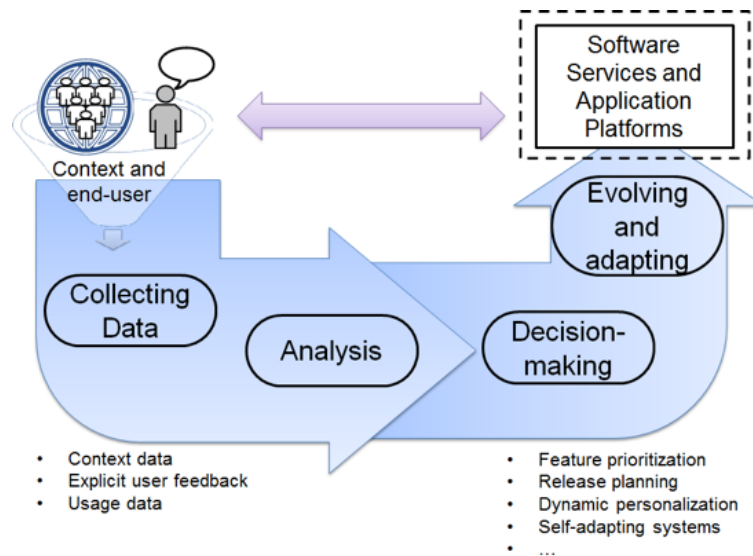


FIGURA 2.2: Cicle de vida i entorn proposat per SUPERSEDE

3. **Decisió.** El coneixement produït a l'anterior fase genera la capacitat de prendre decisions de millora i actuació sobre el sistema software. És a dir, deriva en una o diverses suggerències d'adaptació. Les eines de presa de decisions entren en joc en aquesta fase, rebent com a entrada la informació i, a partir dels criteris i paràmetres definits en relació a aquesta informació, es produeix la suggerència d'adaptació del sistema.
4. **Adaptació.** Un cop el sistema ha estat capaç de produir de forma automàtica una suggerència de millora o adaptació del sistema, aquesta s'aplica sobre el component monitoritzat amb l'objectiu de millorar l'experiència de l'usuari. Arribats a aquest punt, es tanca el cicle de control de qualitat, reflectint la transformació de l'input de la fase de col·lecció, les dades, fins a l'output d'aquesta darrera fase, l'adaptació del sistema.

La Figura 2.2 resumeix aquest cicle de vida i les característiques del context de les seves fases. Tal i com es pot observar, la principal font de dades amb les quals aquest control de qualitat es nodreix és l'experiència de l'usuari final, el context on s'executa aquesta aplicació o sistema i les dades generades del propi ús i funcionament del sistema. És, per tant, un aspecte clau l'obtenció de **dades en execució real**, que seran les que ens permetin aplicar aquest cicle regularment. Regularitat que serà crucial per satisfer l'objectiu d'anàlisi de la qualitat del sistema: només amb dades actuals i constants serem capaços de conèixer l'activitat real del nostre sistema i actuar en conseqüència.

En aquest context identifiquem per tant 4 sectors o subsistemes amb un objectiu específic que, integrats, serveixen a un propòsit genèric. Si ens plantegem com encaixa aquest model dins el nostre tema d'estudi (és a dir, l'adaptabilitat dels sistemes de monitoratge) podem establir una relació directa amb les fases de **col·lecció** de dades i **adaptació**. Dins aquesta primera fase de col·lecció, necessitem definir un sistema capaç d'obtenir aquestes dades, en funció dels sistemes a controlar i de l'interès que tinguem sobre aquests sistemes. Aquest sistema de **monitoratge** estarà compost per un conjunt definit de monitors, encarregats de recollir aquesta informació. Per

altra banda, si volem dotar a aquest sistema de monitoratge d'adaptabilitat, i per tant, garantir un control de qualitat sobre aquests monitors, necessitem establir un sistema que gestioni l'activitat dels monitors i defineixi adaptacions a aplicar sobre aquests monitors.

Pel desenvolupament d'aquest projecte, ens centrarem en aquestes dues parts: el sistema encarregat de la fase de col·lecció de dades (monitors), i el sistema encarregat de gestionar i aplicar les adaptacions sobre aquest sistema de col·lecció. De la mateixa manera, es treballa la integració entre aquests dos components, per generar com a resultat final l'objectiu d'aquest projecte: un sistema de monitoratge autoadaptable, heterogeni i distribuït.

Capítol 3

Gestió del desenvolupament i alternatives

A continuació, procedim a incloure els detalls relacionats amb la planificació i metodologia de desenvolupament.

3.1 Metodologia de desenvolupament

Tal i com es va decidir inicialment, el treball s'està desenvolupant mitjançant una simplificació de la metodologia de desenvolupament àgil **Kanban**. D'aquesta manera, s'ha respectat el desenvolupament de les diferents tasques del projecte en 3 fases:

1. **Identificació de tasques.** En base als requisits inicials del projecte, s'han anat afegint les diferents tasques a realitzar mitjançant l'ús de l'eina **Trello** amb el nivell de detall necessari per considerar-les en una fase prèvia a la seva implementació (*To-Do*).
2. **Implementació.** Conforme disponibilitat, s'ha anat realitzant la implementació de les diferents tasques en paral·lel (sempre amb un màxim de 2-3 tasques alhora, garantint així l'efectivitat de la feina realitzada). En aquest estat (*Doing*), es realitza la feina corresponent a la realització de la tasca segons la seva naturalesa (disseny, implementació...).
3. **Validació.** Un cop s'han complert cadascuna de les tasques (*Done*) s'ha acceptat i validat d'acord amb els requisits considerats.

3.2 Planificació temporal

Segons les necessitats d'aquest projecte i les seves característiques, es va classificar el seu desenvolupament en 4 fases: la planificació, el disseny, la implementació, i la documentació i entrega del projecte. A continuació, procedim a detallar l'estat actual del projecte i desviacions viscudes.

Planificació

Gràcies al curs de GEP es va arribar a completar un percentatge elevat d'aquesta fase, de manera que l'arrencada i desenvolupament del projecte va ser més factible i fluida. Actualment aquesta fase ja es troba totalment acabada. Respecte a les tasques inicials no hi ha hagut desviacions destacables.

Disseny

Totes les tasques de configuració, disseny del sistema, estudi d'arquitectura, i primeres documentacions de memòria i components a desenvolupar s'han complert satisfactòriament. En aquesta fase, si bé no s'han viscut modificacions pròpies de les activitats, si que s'han produït algunes alteracions en quant al moment del desenvolupament. En alguns casos, aspectes de disseny i implementació d'alguns dels components han anat més lligat de la mà, seguint estils més propis de les metodologies àgils, ja que s'ha preferit anar dissenyant i implementant de forma paral·lela a mesura que le desenvolupament i la recerca avançava.

Tot així, aquestes desviacions no han suposat problemes en el desenvolupament del projecte, ja que tal i com es plantejava a la planificació inicial, la gestió i quantització de la dedicació d'algunes d'aquestes tasques no eren dependents entre elles, i per tant s'ha pogut garantir la continuïtat de desenvolupament.

Implementació

En general s'han finalitzat les tasques d'**implementació** dels components principals del projecte. Actualment, s'estan aplicant darreres correccions i validacions d'alguns dels components, però el sistema ja disposa d'un nivell de maduresa suficient com per poder realitzar proves i validacions.

L'únic punt que encara es troba en les primeres fases de desenvolupament és la **implementació del dashboard**. Tot i així, això no suposa un problema, ja que per tal de garantir l'èxit del desenvolupament del projecte és va establir que el nivell de detall i funcionalitats d'aquest seria proporcional al temps disponible per dedicar-hi temps i recursos, al tractar-se d'una eina més aviat complementària i orientada a la presentació que no pas al funcionament i satisfacció dels requisits del sistema.

En qualsevol cas, desviacions pròpies d'aspectes com p.e. nombre d'hores o quantitat de recursos a destinar no s'han produït.

Fase final

Tot i que pròpiament no s'han realitzat tasques específiques d'aquesta fase (de fet, si recordem la planificació inicial, aquesta estava planejada per la 1a setmana de juny), això ja s'escau a la planificació del projecte. De fet, en aquest sentit s'han produït possibles desviacions a favor del temps de desenvolupament, ja que tasques com la **validació de components** (testing) o la **documentació de la memòria** ja s'han començat i, per tant, no requeriran una dedicació tan elevada en aquesta fase.

3.3 Previsió d'alternatives i pla d'acció

Considerem, a continuació, les previsions d'alternatives que es va realitzar durant el curs de GEP i, si s'escau, els detalls d'aquests desviaments.

- **Increment del nº d'hores necessàries.** Fins al punt de desenvolupament actual, no s'ha produït un increment del nº d'hores (més enllà de petites desviacions que s'han anat autoregulant).

Pla d'acció. *No aplicat*

- **Avaria en el hardware.** No hi ha hagut cap imprevist relacionat amb avaries del hardware utilitzat.

Pla d'acció. *No aplicat*

- **Canvis en els requisits del projecte.** En aquest sentit sí que s'han produït algunes modificacions. En general, s'ha realitzat una **modificació de prioritats** en els requisits i s'ha afegit una nova part dins l'àmbit de desenvolupament del projecte, corresponent a l'adaptació dels monitors i la gestió d'un sistema d'adaptabilitat automàtic mitjançant l'ús de diagrames UML.

Pla d'acció. Tot i que ha suposat un canvi en els requisits, no s'han produït modificacions en la càrrega de feina ja que simplement s'ha fet un plantejament diferent dels requisits i necessitats del nostre projecte. Al prendre aquesta decisió amb la suficient antel·lació, no ha suposat un problema de gestió del projecte.

Capítol 4

Integració de coneixements

Un dels objectius del Treball de Final de Grau és constatar l'assimilació dels coneixements estudiants durant el Grau en Enginyeria Informàtica i, concretament en aquest cas, els conceptes propis de l'especialitat d'Enginyeria del software. Aquests conceptes engloben diverses disciplines, que el desenvolupament del projecte treballen, i a continuació s'exposen.

Disseny de sistemes

La solució proposada per aquest treball consisteix en un **sistema de monitoratge dotat d'autoadaptabilitat**. Aquesta premisa genèrica, tal i com es plasmarà a la memòria, engloba en realitat un sistema complex format per un conjunt de components independents i amb activitat pròpia que, integrats dins el nostre sistema i definint una interacció específica entre aquests, serveixen a un propòsit superior orientat a la gestió i adaptabilitat de sistemes de monitoratge (és a dir, sistemes encarregats del control de qualitat).

En definitiva, la solució plantejada i exposada presenta el **disseny, implementació i integració de components**, i per tant l'anàlisi i estudi de sistemes software complexos, un dels pilars de l'enginyeria del software.

Disseny arquitectura software

Cadascun dels components que integren el sistema ha de satisfer uns requisits des d'un punt de vista d'arquitectura del software. Per exemple, la implementació dels monitors ha estat basada en una proposta d'arquitectura genèrica que dota al sistema de l'extensibilitat i heterogeneïtat que volíem assolir.

Per aquest motiu, durant el disseny dels components s'han aplicat coneixements relacionats amb el disseny software i l'ús de patrons arquitectònics, conceptes treballats al llarg de l'especialitat cursada.

Gestió i desenvolupament de projecte

De forma explícita, el desenvolupament d'un Treball de Fi de Grau inclou aspectes relacionats amb la gestió d'un projecte, coneixements també treballats durant el curs de l'especialitat d'enginyeria del software. La prioritització de requisits, la transició de fases, etc., s'han aplicat des d'un punt de vista teòric i pràctic.

Disseny i implementació de serveis RESTful

Els components del sistema s'exposen com a serveis RESTful per tal de permetre'n una integració i dotar al sistema de la característica de distribució que volíem assolir. Per fer-ho, coneixements d'arquitectura de serveis web s'han hagut d'aplicar per dissenyar i implementar aquests components.

Disseny i implementació de bases de dades

De forma més genèrica i superficial que els altres conceptes, també ha calgut dissenyar i implementar bases de dades per gestionar el sistema, tot i que la complexitat i els esforços dedicats a aquesta part han estat menors, ja que l'únic propòsit ha estat satisfer la necessitat de persistència de models UML i les dades de monitoratge.

Altres coneixements d'implementació

Per suposat, tots aquells conceptes propis de les capacitats com a *software developer* (coneixements en programació amb Java, ús de gestor de versions, ús de gestió de dependències, etc.) han estat necessaris.

Capítol 5

Integració de lleis i regulacions

Els principals aspectes relacionats amb la integració de lleis i regulacions ha estat la gestió de les llicències dels components desenvolupats. Al tractar-se d'un sistema desenvolupat dins del context del projecte SUPERSEDE, del qual la UPC n'és *partner*, el copyright de tot el codi font implementat pertany a la Universitat Politècnica de Catalunya. Tanmateix, al tractar-se de components dels quals jo mateix n'he estat el desenvolupador, hi figuro com a *main development contribution*. En alguns casos aquesta contribució no ha estat exclusiva, i així es documenta al llarg de la memòria per deixar de forma explícita aquells components d'autoria completa pròpia, i aquells en els quals hi he col·laborat de forma parcial.

Tots els components estan sota la *Apache License 2.0* (els detalls de la mateixa es poden consultar a <http://www.apache.org/licenses/LICENSE-2.0>). Tal i com s'estableix en les condicions d'aquesta llicència, el codi és lliure d'utilitzar-se, reaprofitar-se, distribuir-se i modificar-se (a diferència d'altres llicències de tipus *copy-left*).

Capítol 6

Conclusions

Amb la informació presentada anteriorment, podem arribar a les següents conclusions que garanteixen la viabilitat de finalització del projecte:

- **Definició i contextualització del sistema.** L'aprofundiment en la temàtica i la recerca i contextualització de l'àmbit d'estudi garanteixen un punt de partida fort per realitzar un treball vàlid i contrastat sota un marc de desenvolupament específic.
- **Seguiment acurat de la planificació.** Tal i com s'ha exposat prèviament, no s'han produït modificacions que hagin causat desviacions crítiques en el desenvolupament del projecte. Les poques desviacions viscudes han estat tractades i en cap cas han compromés la integritat del desenvolupament, sinó que han permet una millora o simplement un canvi en els objectius i els resultats finals.
- **Integració de coneixements.** Com a Treball de Final de Grau amb l'objectiu d'integrar els coneixements estudiats durant el curs del mateix, aquest document valida breument els conceptes treballats per tal de garantir no només que es tracta d'un projecte amb potencial propi del grau, sinó també de l'especialitzat sota la qual es desenvolupa.
- **Integració de lleis i regulacions.** Com a part fonamental de la gestió d'un projecte, es presenten els aspectes relacionats amb lleis i regulacions que poden afectar al desenvolupament (en aquest cas, bàsicament la gestió de llicències del codi).

En base a aquests punts, podem garantir que el projecte segueix una via satisfactòria, que garanteix els objectius principals del projecte, i que la seva realització portarà a una cloenda completa dels coneixements i l'experiència obtinguda durant el curs del Grau en Enginyeria Informàtica.

Bibliografia

- [1] Gabor Karsai et al. "An Approach to Self-adaptive Software Based on Supervisory Control". A: *Self-Adaptive Software: Applications (IWSAS)* (2001).
- [2] Ying Li et al. "Self-reconfiguration of service-based systems: A case study for service level agreements and resource optimization." A: *International Conference on Web Services* (2005).
- [3] Salvatore Venticinque. "Agent based services for negotiation, monitoring and reconfiguration of cloud resources." A: *European Research Activities in Cloud Computing* (2012).