



UNIVERSITAT DE
BARCELONA

Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques i Informàtica
Universitat de Barcelona**

PAMTool, Premium Analysis and Monitor Tool

Jordi Armengol Bosch

Director: Sergio Sayago
Realitzat a: Departament de
Matemàtiques i Informàtica
Barcelona, 25 de juny de 2019

Resum

Premium, SA és l'empresa on s'ha dut a terme el treball present. Es dedica al disseny i producció de sistemes de conversió d'energia.

Actualment, la tecnologia avança a passos agegantats i Premium no n'és l'excepció. Per tal de mantenir els productes i dispositius a l'avantguarda del sector, no és suficient realitzar tasques com a carregadors, onduladors, convertidors, etc. Per aquest motiu, el mercat demana productes configurables amb gestió d'errors i que es puguin monitoritzar sense la necessitat d'eines especialitzades més enllà d'un ordinador.

Per realitzar la transició del paper a la realitat d'aquestes necessitats, entra en joc l'equip de software. Per una banda, estan presents els dispositius que aporten habilitats comunicatives amb l'exterior i per l'altra, la part externa que requereix una aplicació de sobretaula que gestiona les comunicacions i totes les possibilitats que ofereixen.

En aquest projecte la contribució principal a l'empresa és el desenvolupament de l'aplicació de sobretaula i tota la seva planificació i disseny. Aquesta aplicació ha de complir certs requisits imposats per l'empresa:

- Disponibilitat per a diferents sistemes operatius.
- Existència d'una sola aplicació per tots els dispositius, incloent els dispositius amb un protocol més antic i senzill.
- Disposar d'una interfície gràfica per als usuaris més inexperts.
- Disposar d'una interfície per línia de comandes per a usuaris avançats i/o possibles automatitzacions.

Tot i els problemes trobats en la planificació, el desenvolupament de l'aplicació es pot realitzar i es troba operativa a la web de l'empresa disponible per a els usuaris de forma gratuïta.

Resumen

Premium, SA es la empresa donde se ha llevado a término el trabajo presente. Se dedica al diseño y producción de sistemas de conversión de energía.

Actualmente, la tecnología avanza a pasos de gigante y Premium no ha sido la excepción. Para mantener los productos y dispositivos de la empresa en la vanguardia del sector, no es suficiente realizar su función como cargadores, onduladores, convertidores, etc. Por este motivo, el mercado reclama productos configurables, con gestión de errores, que se puedan monitorizar sin necesidad de herramientas especializadas más que un ordenador.

Para llevar estas necesidades del papel a la realidad entra en juego el equipo de software. Por un lado, están presentes los dispositivos que aportan habilidades comunicativas con el exterior y por el otro lado, la parte externa que requiere una aplicación de sobremesa que gestiona las comunicaciones y todas las posibilidades que ofrecen.

En este trabajo nuestra aportación a la empresa será el desarrollo de la aplicación de sobremesa y toda su planificación y diseño. Esta aplicación tiene que cumplir con ciertos requisitos impuestos por la empresa:

- Disponibilidad para diferentes sistemas operativos.
- Existencia de una sola aplicación para todos los dispositivos, incluyendo los dispositivos con un protocolo de comunicación más antiguo y sencillo.
- Disponer de una interfaz gráfica para los usuarios más inexpertos.
- Disponer de una interfaz por la línea de comandos para usuarios avanzados y/o posibles automatizaciones.

A pesar de los problemas encontrados con la planificación, el desarrollo de la aplicación se completa y se encuentra operativa en la propia web de la empresa disponible de forma gratuita para todos los usuarios.

Abstract

Premium, SA is a company dedicated to designing and producing power conversion systems, and this project will be executed. Technology is quickly moving forward, that is a fact. If we talk about Premium, they are not an exception and with the goal to keep their products at the avant-garde of the sector, it is not enough that they comply with they're function as chargers, inverters, converter, etc. The market demands configurable products, with error management, with the ability to be monitored without the need of specialized tools except a computer.

To bring these needs from paper to reality the software team comes into play, firstly the devices need some "intelligence" so they can communicate with the outside world and secondly, a desktop application to manage this communication and all the possibilities they offer.

In this project our main contribution to the company will be the desktop application and all its planning, design and development. This application has some requirements specified by the company that it must comply:

- Available for different operating systems.
- Only one application for all the company devices, including the ones that use an older and more simple communication protocol.
- It must have a graphical user interface for the inexperienced users.
- It must have a command line interface for the advanced user and all the possible automatizations.

Even with all the problems found due to the planification, the development of the application can be completed, and it can be found in the enterprise website free for all users.

Índex

Resum	2
Resumen	3
Abstract	4
Índex.....	5
1. Introducció i motivació.....	7
1.1 L'empresa	7
1.2 El problema.....	8
2. Objectius.....	9
2.1 Llista de requisits i funcionalitats	9
3. Antecedents.....	10
3.1 Estat de l'art.....	10
3.1.1 Java	10
3.1.2 SWT Standard Widget Toolkit i Swing.....	11
3.1.3 jSerialComm	11
3.1.4 Comunicació sèrie	11
3.1.5 BUS CAN.....	12
4. Planificació.....	13
4.1 Planificació estimada.....	13
4.1.1 Fase 1 (20/08/2018):.....	13
4.1.2 Fase 2 (30/12/2018).....	14
4.1.3 Fase 3 (03/02/2019).....	14
4.2 Temps invertit real.....	15
4.2.1 Fase 1 (20/08/2018):.....	15
4.2.2 Fase 2 (25/12/2018).....	15
4.2.3 Fase 3 (10/02/2019).....	16
4.3 Diagrama de Gant.....	17
5. Valoració econòmica.....	18

5.1 Estimació econòmica	18
6. Anàlisis	19
6.1 Diagrama de casos d'ús.....	19
6.2 Casos d'ús textuais	20
6.3 Model de domini.....	28
6.4 Interfícies	29
6.4.1 Pestanya "General"	29
6.4.2 Pestanya "Detail"	30
6.4.3 Pestanya "Connection".....	32
6.4.4 Pestanya "Information".....	33
6.4.5 Pestanya "Update"	34
6.4.5 Pestanya "Log".....	35
6.4.5 Pestanya "Developer"	36
6.4.5 Pestanya "Calibration".....	37
6.4.6 CLI.....	38
7. Disseny i implementació	39
7.1 Patró observador.....	39
7.2 Patró singleton	39
7.2 Patró factory.....	40
7.3 Patró Facade	40
7.3 Distribució de l'aplicació	41
7.3.1 Java 11	41
7.3.2 Maven.....	42
7.3.3 jlink	43
8. Discussió	44
9. Conclusions i futures ampliacions	46
Bibliografia.....	47

1. Introducció i motivació

1.1 L'empresa

Premium és una empresa dedicada al disseny i producció de sistemes de conversió d'energia. Principalment són productes i solucions personalitzades per a segments especials del mercat i en general requereixen una qualitat i fiabilitat molt alta tant com productes resistents.

Com a proveïdors de mercats tecnològics amb una exigència molt elevada, la qualitat dels seus productes i el servei d'atenció al client són el seu compromís més important. La satisfacció total del client és el seu objectiu principal. Els seus objectius són la constant ampliació i modernització dels seus productes estàndard, així com ser un dels principals fabricants de productes a mesura.

En un principi, el seu mercat integrava principalment àrees industrials (instrumentació, maquinària, entre altres), però més tard es va ampliar a altres camps com:

- **Comunicacions.** Fonts d'alimentació, sistemes rectificadors i convertidors per a equips de comunicació a través de línies d'alta tensió, control de tràfic aeri, etc.
- **Fabricants de maquinària d'alta tecnologia.** Fonts d'alimentació i convertidors per a màquines de litografia, recobriment PVD PACVD, indústria òptica, sistemes de test, sistemes d'alimentació, etc.
- **Ferrocarrils.** Fonts d'alimentació, convertidors i onduladors tant per a equips de control en terra com per a servei mòbil. Sistemes de senyalització ATP i ATO, aire condicionat, llums de marxa, etc.
- **Energia.** Carregadors de bateries, sistemes rectificadors i convertidors per a instal·lacions de AT i MT, automatització de processos i control, subestacions, centrals nuclears, etc.
- **Electromedicina.** Escàners de raigs-X, aparells de depilació làser, etc.

La presència de l'empresa en els diferents sectors es podria repartir tal i com es mostra en la [Taula 1](#) [1]:

Taula 1. Presencia de productes segons sectors

Comunicacions	10%
Maquinària d'alta tecnologia	40%
Ferrocarrils	20%
Energia	6%
Electromedicina	6%
Varis	4%

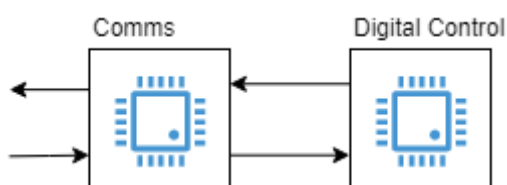
1.2 El problema

Tot el relacionat amb la tecnologia està evolucionant a velocitats cada vegada més grans. Això es tradueix en l'aparició de grans canvis en les tecnologies i mercat en un interval de temps reduït.

Molts dels productes de Premium ja inclouen un microcontrolador a dintre per controlar de forma digital el hardware. A partir d'ara, aquest microcontrolador l'anomenarem com a l'encarregat de la part de potència.

Fins fa relativament poc, els productes només suportaven la configuració d'alguns dels seus paràmetres. El fet que el microcontrolador de potència s'encarregués de controlar el hardware, complicava molt que aquest mateix pogués comunicar-se amb l'exterior. Sobretot per temes de memòria, potència i gestió del temps. La solució més senzilla ha estat incorporar un altre microcontrolador, el qual va enfocat a les comunicacions externes i comunicacions amb el microcontrolador de potència. Es pot apreciar un exemple a la *Il·lustració 1*.

Il·lustració 1. Esquema microcontroladors



A partir d'aquest fet, cada microcontrolador pot especialitzar-se molt més i ampliar les seves funcionalitats dins de les seves tasques. El microcontrolador de comunicacions pot incloure noves funcions com ara:

- Augment de la velocitat de transmissió i processament de comandes al exterior.
- Creació de *logs* interns dels esdeveniments que succeeixen al microcontrolador.
- Disposar de un *bootloader* encarregat de les actualitzacions de *firmware*.

El resultat que s'aconsegueix és l'adquisició d'un producte molt més atractiu de cara al client. Així doncs, actualment amb una sola compra inicial es pot disposar d'un producte configurable, reduint el cost i ampliant les possibilitats d'utilització d'un mateix dispositiu. Tot això ens porta al centre d'aquest treball que és desenvolupar una aplicació que configuri aquests nous dispositius intel·ligents. Per aquest motiu, ha de complir certes característiques imposades per l'empresa:

- Disponible per a diferents sistemes operatius.
- Una sola aplicació per tots els dispositius de l'empresa que suportin comunicacions, tant els que feien servir el protocol antic, com els nous dispositius amb les noves funcionalitats i un protocol totalment renovat.
- Ha de disposar d'una interfície gràfica que sigui còmoda, intuïtiva i senzilla.
- Ha de disposar d'una interfície per línia de comandes per entorns on no hi hagi un sistema operatiu amb interfície gràfica o per donar la possibilitat d'automatitzacions en el procés de fabricació.

A banda d'aquestes premisses, les funcionalitats que permet l'aplicació depenen de cada producte i del protocol que facin servir. Al ser un producte en fase de desenvolupament actiu, si el client o enginyers del departament sol·liciten una nova funcionalitat, es debat internament al departament de software el com i quan s'implementarà aquesta funcionalitat si és possible.

2. Objectius

L'objectiu fonamental d'aquest projecte és desenvolupar una aplicació de sobretaula amb suport per diferents sistemes operatius i amb suport per a tots els dispositius amb comunicacions de Premium.

2.1 Llista de requisits i funcionalitats

Per tal de definir fins a quin punt arribaria aquesta aplicació era necessari establir una llista de requisits i funcionalitats que s'havien de complir i així, poder donar-la com a vàlida.

Requisits:

- Ser multi plataforma amb un sol arxiu executable.
- Llenguatge de programació Java.
- Interfície gràfica (GUI).
- Interfície per línia de comandes (CLI).
- Fer ús de la llibreria desenvolupada per la pròpia empresa per comunicar-se amb els dispositius.

Funcionalitats a implementar:

- Detecció automàtica del dispositiu i la seva velocitat de connexió.*
- Secció d'ajuda dintre de l'aplicació.
- Enllaç a la documentació del dispositiu des de l'aplicació.
- Auto actualització de les dades de monitorització en diferents intervals de temps.
- Control sobre quines dades s'actualitzen quan s'està monitoritzant.
- Filtrat de dades de monitoratge.
- Configuració de paràmetres que ho permetin.*
- Representació d'informació respecte el dispositiu com ara la data de fabricació, número de sèrie, etc.
- *Log* intern de successos de l'aplicació.*
- Importació i exportació de configuració mitjançant un fitxer XML.*
- Creació de fitxer CSV amb dades de monitoratge.
- *Update* de *firmware* a través de l'aplicació.*
- Pestanya que actuï com una aplicació de comandes com "*TeraTerm*", "*Putty*", etc.
- Configuració de paràmetres de calibratge.*
- Poder canviar entre *bootloader* i aplicació en el dispositiu.

*Funcionalitats disponibles amb CLI.

3. Antecedents

3.1 Estat de l'art

Per saber com fer una aplicació de qualitat és important conèixer l'estat actual de la tecnologia que es farà servir, així com les seves limitacions, punts forts i punts febles.

3.1.1 Java

Com a base d'un projecte basat en una aplicació multi-plataforma, és important definir les tecnologies que es faran servir i perquè.

En aquest concretament, es tria Java com a base del projecte. Java és un llenguatge de programació nascut els anys 90 amb una premissa molt clara en el seu moment de concepció, "*Write once, run everywhere*" [2]. Aquest eslògan és cert, però amb matisos. Per fer córrer un programa desenvolupat amb Java és necessari disposar de la màquina virtual de Java inclosa dintre del "*Java Runtime Environment o JRE*". Per tant, si es vol córrer un programa desenvolupat amb aquesta tecnologia, s'ha de distribuir un JRE vàlid juntament amb ella, o bé el sistema l'ha de tenir instal·lat prèviament. De primeres, això pot semblar un desavantatge, però Java compta amb una gran presència en el maquinari de sobretaula i no és habitual que una màquina no el tingui instal·lat.



Il·lustració 2. Logotip Java

Java és un llenguatge de programació orientat a objectes gratuït que sempre ha tingut el recolzament de grans empreses com Sun Microsystems, Oracle, Azul, etc. Durant el transcurs d'aquest projecte, algunes de les premisses més importants de Java han anat variant pel canvi de versions i/o decisions de la principal empresa que mantenia Java, Oracle, però això es detallarà més endavant.

Característiques de Java:

- **Senzill:** Java es va crear perquè sigui un llenguatge fàcil amb una sintaxis clara i entenedora.
- **Orientat a objectes:** Tal i com s'ha comentat anteriorment, Java segueix els paradigmes de la programació orientada a objectes, ja que la programació amb Java es centra en la manipulació, creació i construcció d'objectes.
- **Distribuït:** Java permet la construcció d'aplicacions distribuïdes per mitjà d'una col·lecció específica de classes.
- **Interpretat:** Es necessita un intèrpret per executar els programes de Java, això alenteix els programes però els hi dona flexibilitat. En el nostre cas les tasques a realitzar no són crítiques i per tant, aquesta falta de velocitat no és rellevant i es dóna prioritat al fet que sigui multi-plataforma.
- **Robust:** Java és un llenguatge robust i fiable, s'ha escrit pensant en poder verificar errors i està molt tipificat.
- **Segur:** Java produeix pocs problemes de seguretat, característica molt important en les aplicacions distribuïdes d'internet.
- **Arquitectura neutral:** Java és independent de la plataforma final on s'executarà el programa.
- **Portable:** Java és un llenguatge d'alt nivell i de plataforma independent.
- **Concurrent:** Java permet l'execució de múltiples fils d'execució, o diverses tasques de forma simultània.
- **Dinàmic:** En temps d'execució, l'entorn Java es pot ampliar mitjançant enllaços a classes que poden estar localitzades en servidors remots o en xarxa.

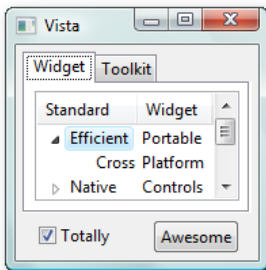
A banda d'aquestes característiques pròpies del llenguatge, s'ha de tenir en compte altres factors externs com la immensa comunitat que suporta aquest llenguatge, la gran quantitat de llibreries ja desenvolupades i la flexibilitat del llenguatge. Tots ells han estat punts decisius a l'hora d'escollir-lo com a base del projecte.

Tot i això, els canvis que ha dut a terme Oracle com a principal i més coneguda empresa que suporta Java han comportat un retard i canvis importants en el projecte. Aquests canvis es detallaran a l'apartat [!Error! No se encuentra el origen de la referencia.].

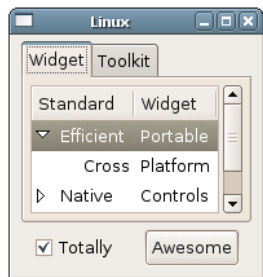
3.1.2 SWT Standard Widget Toolkit i Swing

SWT és un conjunt d'eines per Java que implementen i desenvolupen interfícies gràfiques natives del sistema operatiu de forma eficient i portable. És un projecte "open source" nascut degut a la necessitat d'implementar una interfície per a *Eclipse*, ja que aquest està escrit en Java [3].

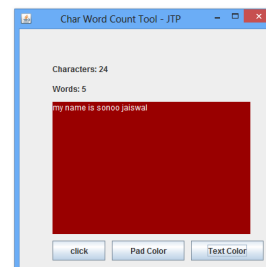
Swing també és un conjunt d'eines de Java per implementar i desenvolupar interfícies gràfiques, però en aquest cas va ser desenvolupat per oferir un conjunt de components més sofisticats que els seus predecessors. *Swing* mescla l'aspecte i funcionalitats de diferents plataformes [4].



Il·lustració 3. SWT win



Il·lustració 5. SWT linux



Il·lustració 4. Swing

3.1.3 jSerialComm

jSerialComm és una llibreria de codi obert multi-plataforma de Java dissenyada per a oferir un accés estàndard als ports sèrie de l'equip. Està pensada per facilitar al màxim el seu ús, donar suport avançat de *timeouts* i aportar la habilitat d'obrir simultàniament diferents ports [5].

Algunes de les seves funcionalitats són:

- Implementació lleugera i eficient.
- Llista tots els ports sèrie del maquinari.
- Proporciona el nom del sistema del port i un nom "amigable" del mateix.
- Configuració del *baud rate*, bits de les dades, bits de parada i paritat dels ports.
- Paràmetres de control dels ports com CTS, RTS/CTS, DSR, etc.
- Possibilitat d'escriure i llegir "bytes" directament del port sèrie.

3.1.4 Comunicació sèrie

La comunicació sèrie és la forma de transmetre informació entre dispositius i perifèrics més usada. Es tracta de la transmissió de dades mitjançant polsos binaris de forma cablejada. Tota transmissió té un receptor i un remitent. Com a mínim es necessiten 3 canals de connexió, TX (transmissió), RX (recepció) i GND (terra).

3.1.5 BUS CAN

El “*bus-CAN*” és un protocol de comunicacions sèrie per a control distribuït en temps real amb un alt nivell de seguretat [6]. Descriu dues capes de l'entorn OSI:

- Física: On no es defineix cap medi de transport físic. Bàsicament es defineix els estats lògics: bit Dominant-Recessiu.
- Enllaç: Es defineixen tots els serveis de les subcapes MAC i LLC:
 - Codificació/Descodificació de Trames.
 - Filtrat de Missatges.
 - Detecció i Notificació d'errors.
 - Arbitratge del bus, *Timing*...

Les seves característiques principals són:

- Protocol orientat a missatges no a dispositiu.
- Multimestre.
- Multicast.
- Protocol d'Arbitratge de bus tipus CSMA-CA¹.
- Velocitat màxima 1Mb/s.
- Es defineixen estats de baix consum pels nodes.
- Sistema de detecció/Senyalització d'errors potent:
 - CRC².
 - Monitorització dels busos mentre envia dades.
 - Comprovació de format de trames.
 - “*Bit Stuffing*”.

¹ “*Carrier Sense Multiple Access With collision Avoidance*”

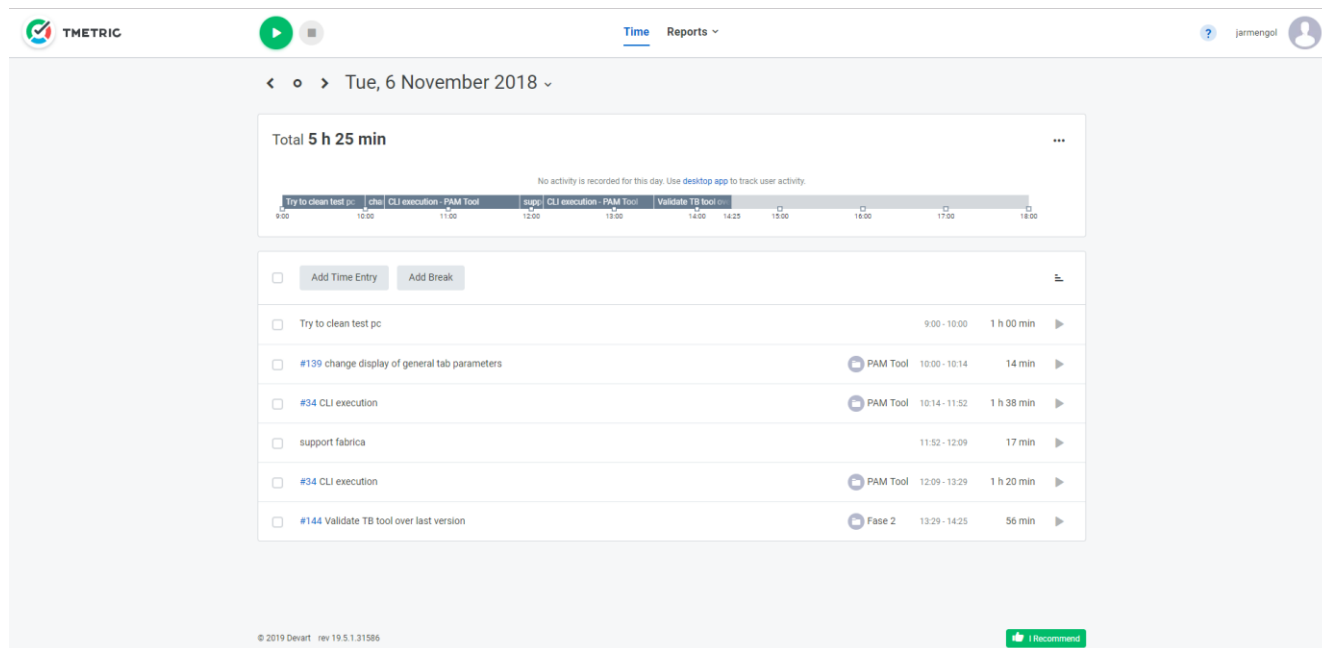
² “*Cyclic Redundancy Check*”

4. Planificació

Com el treball s'ha realitzat en una empresa, la planificació ha estat un dels punts més significatius i controlats del projecte. Malgrat no tenir una data d'entrega fixa, a mesura que es definien les tasques per a cada versió, es planificava quin temps aproximadament costaria dur a terme cada tasca.

Tot aquest control de la planificació del temps es duia a terme amb l'ajuda del servidor propi de "Redmine" [7] i l'aplicació de control del temps, "Tmetric".

Il·lustració 6. Tmetric



4.1 Planificació estimada

Al ser una aplicació pensada per ser usada tant bon punt estigui operativa, era de vital importància el fet de segmentar el projecte en diferents fases de desenvolupament. El sistema de versions seguirà la nomenclatura estàndard de vLX.Y.Z on, L = A(Alpha), B(Beta), T(Test), etc. X = "Major release". Y = "Minor release". Z = "Bugfixes". En aquest cas es va decidir dividir-ho en 3 fases.

4.1.1 Fase 1 (20/08/2018):

En aquesta fase la prioritat i objectiu numero 1 és tenir una **aplicació operativa, usable i estable**. Per a tal objectiu els requisits a complir són els següents:

- El disseny de l'aplicació es durà a terme en aquesta fase.
- L'aplicació ha de permetre monitoritzar les dades del dispositiu.
- L'aplicació ha d'incloure una finestra d'ajuda dintre de la pròpia interfície.
- L'aplicació ha de permetre configurar paràmetres del dispositiu.
- L'aplicació ha de disposar d'una CLI i una GUI.
- L'aplicació ha de ser multi-plataforma amb un sol fitxer executable.
- L'aplicació ha de permetre la configuració de paràmetres de calibratge.
- L'aplicació ha de permetre extreure "logs" i informació per poder oferir un servei tècnic a distància el més eficient possible.
- L'aplicació ha de permetre l'actualització del "firmware" del dispositiu connectat.
- L'aplicació ha de suportar els dispositius amb "Protocol".

Aquesta fase és la que s'ha planificat una duració més llarga, ja que implica molt temps el disseny, proves i desenvolupament d'una bona base que més tard implementa funcionalitats més complexes.

En finalitzar aquesta etapa, l'aplicació es considera apta per a ser utilitzada per clients i es distribuïx als que disposin de dispositius suportats en aquest punt.

4.1.2 Fase 2 (30/12/2018)

La fase dos és la fase amb menys temps previst, això és degut a que les tasques planificades en aquesta fase **no són extenses, però tenen molta prioritat**. En aquest punt té més importància la velocitat en poder distribuir la solució a la quantitat de noves característiques a oferir:

- L'aplicació ha de suportar tots els dispositius actualment disponibles a Premium amb comunicacions.
- L'aplicació ha d'auto detectar el dispositiu connectat en el port seleccionat i adaptar el protocol utilitzat a aquest.
- L'aplicació ha de detectar automàticament la velocitat de transmissió suportada pel dispositiu, adaptar-s'hi i mostrar per la interfície quina és aquesta.
- L'aplicació ha de permetre extreure un fitxer de configuració XML del dispositiu connectat.
- L'aplicació ha de permetre importar un fitxer de configuració XML del dispositiu connectat.

En aquesta fase la prioritat era clara, **augmentar el nombre d'usuaris** que la poden fer servir, ja que actualment la gran majoria dels dispositius amb comunicacions són producte estàndard amb el protocol antic.

4.1.3 Fase 3 (03/02/2019)

La fase 3 és una fase de refinament. En aquesta fase el producte és ja molt complet i ja és apte per a tots els dispositius de l'empresa que el suportin. Ens centrarem en polir i oferir funcionalitats més específiques i menys enfocades al usuari general.

- L'aplicació ha de permetre filtrar les dades de monitoratge.
- L'aplicació ha de permetre crear un fitxer de "log" del monitoratge de les dades.
- L'aplicació ha de suportar auto detecció de velocitat màxima de mostreig de monitoratge.
- L'aplicació ha de permetre la creació de gràfics en temps reals sobre valors com, voltatge d'entrada, de sortida, intensitat de sortida, etc.

En el moment que aquests requisits es compleixin es donarà l'aplicació com a finalitzada i simplement s'haurà d'anar afegint els nous dispositius que vagi creant l'empresa dins de l'aplicació.

4.2 Temps invertit real

Com és evident, la planificació d'un projecte està subjecta a moltes variacions amb el temps com poden ser canvis de prioritats en projectes, correcció de "bugs", problemes amb tecnologies usades, etc. Aquestes variacions s'intenten tenir en compte a l'hora de planificar, però tot i així sempre està el factor sorpresa i sobretot, l'experiència. Aprendre a planificar no és trivial, es necessita molt temps planificant i controlant quan es tarda aproximadament en cada tasca segons la seva complexitat i la jurisprudència que hi hagi. En el cas d'aquest projecte un dels **factors claus en la desviació** de la planificació han sigut els **canvis en les condicions de la versió de Java** distribuïda per Oracle (la més habitual).

En el moment d'iniciar aquest projecte es decideix per tecnologia principal de desenvolupament Java pels motius comentats anteriorment en la secció de [3.1.1 Java]. Per explicar aquest esdeveniment i les seves implicacions dividirem el temps invertit real en les 3 fases vistes anteriorment i d'aquesta manera serà més fàcil veure on s'ha desviat en major mesura la planificació.

4.2.1 Fase 1 (20/08/2018):

Taula 2 Comparació hores Fase 1

Comparació de les hores	
Estimació	Real
393	313
Desviació	-20.3 %

En aquesta fase la data inicial es manté ja que és la data d'inici i per tant les desviacions poden aparèixer a partir d'aquesta data. Eren moltes les tasques a realitzar dintre d'aquesta fase, però també es va optar per una planificació molt conservadora, ja que al ser la base de tot el projecte, es volia tenir per segur que no s'escatimaria amb les hores de disseny i desenvolupament. D'aquesta manera, aquesta fase es va finalitzar abans del previst i es va poder emprendre la segona fase abans del previst.

Concretament, l'estimació de temps de desenvolupament d'aquesta fase era de 393 hores incloent reunions per definir requisits, documentació, disseny i desenvolupament. Finalment, el cost real en hores ha estat de 313 hores. Aparentment, pot semblar positiu el fet d'haver finalitzat la feina prevista amb menys hores, però al final l'error segueix sent una falta de previsió i estimació adequada.

4.2.2 Fase 2 (25/12/2018)

Taula 3. Comparació hores fase 2

Comparació de les hores	
Estimació	Real
91	121
Desviació	33.0 %

Degut a l'avançament de la data de finalització de la fase 1, la data de inici de la fase 2 es va poder avançar aproximadament una setmana. Contràriament, als successos de la fase 1, en aquest cas la desviació és positiva en el sentit que les hores reals han estat superiors a les hores estimades. Aquesta

desviació és deguda a la subestimació del cost de suportar tots els dispositius. El protocol de comunicació utilitzat pels dispositius més antics és completament diferent al nou “*Protocol*” i per tant, es va haver de modificar la llibreria de comunicacions per tal de suportar-la. A més, un mateix dispositiu del protocol antic pot tenir diverses versions de “*firmware*” i cada una d’elles disposar alhora de petites variacions. Si a tot això li sumem que no està correctament documentat, fa que en aquest cas s’hagi invertit més hores de les que realment serien necessàries si la documentació fos correcta i actualitzada.

4.2.3 Fase 3 (10/02/2019)

Taula 4. Comparació hores fase 3

Comparació de les hores	
Estimació	Real
154	279
Desviació	81.2 %

Tal i com s’explica a la introducció d’aquest punt i com es pot apreciar a la Taula 4. Comparació hores fase 3, la desviació en aquesta fase és desmesurada. Això és degut a que en aquest punt l’estat de l’aplicació es troba en què mitjançant la tecnologia d’automatització [8] *Ant* es genera un sol fitxer *.jar* que tria en temps d’execució quina llibreria de SWT carregar, ja que al ser una llibreria dependent del sistema operatiu, és necessari que sigui la mateixa.

En aquest punt comencen els problemes, Oracle informa que a partir del primer quadrimestre del 2019 les versions que oferiria serien sota llicència comercial, és a dir, que es poden fer servir per ús personal, però no com a eina monetària. Després de dedicar molts dies a investigar fins a quin punt aquest canvi ens afectava, parlar amb comercials d’Oracle i moltes hores de recerca, es va arribar a la conclusió que es migraria a versions amb llicència gratuïta de Java. Per què es va prendre aquesta decisió? Segons el que vam concloure, el desenvolupament de java és de codi obert a “*openjdk*” [9] i després cada companyia li afegeix els seus paquets personalitzats i distribueix les imatges sota la seva llicència, en aquest cas passant a ser comercial.

En aquest punt el problema podia estar solucionat, però al estar investigant sobre tot el relacionat amb les noves versions de java, es va descobrir que a partir de la versió 9 s’introdueixen millores importants que podien ser molt beneficioses pel projecte. Un dels grans punts febles de l’aplicació és que era necessari tenir la màquina virtual de java instal·lada en l’ordinador per poder-la fer servir, i justament en java 9 s’introdueixen noves mecàniques de programació com els mòduls [10], i l’eina “*jlink*” [11]. Aquestes dues característiques obren una nova possibilitat que és poder distribuir l’aplicació sense dependre de la màquina virtual instal·lada en l’entorn del usuari sense haver de distribuir el JRE sencer que ocupa uns 200MB aproximadament. De totes maneres, en lloc de java 9 es decideix prosseguir amb java 11, ja que és la versió LTS³.

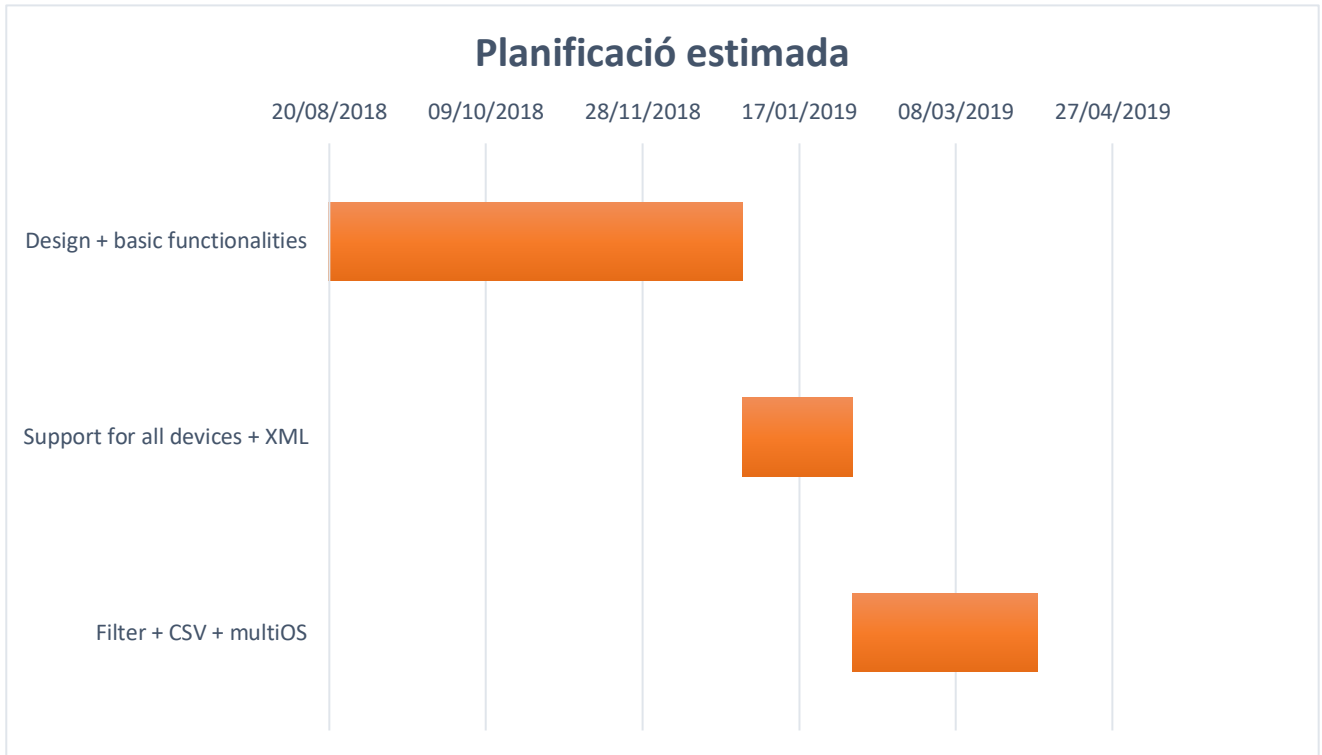
La suma d’aquests fets ens porta al retard del projecte, però alhora ens aporta una millora substancial, ja que l’aplicació totalment independent del sistema operatiu ens ocupa “només” 40 MB. En aquest punt es va haver de prioritzar tasques i les dues últimes de la fase 3 s’han deixat per més endavant, ja que les hores invertides excedeixen amb un gran marge la previsió i hores teòriques de dedicació al TFG.

³ “*Long-term support*”

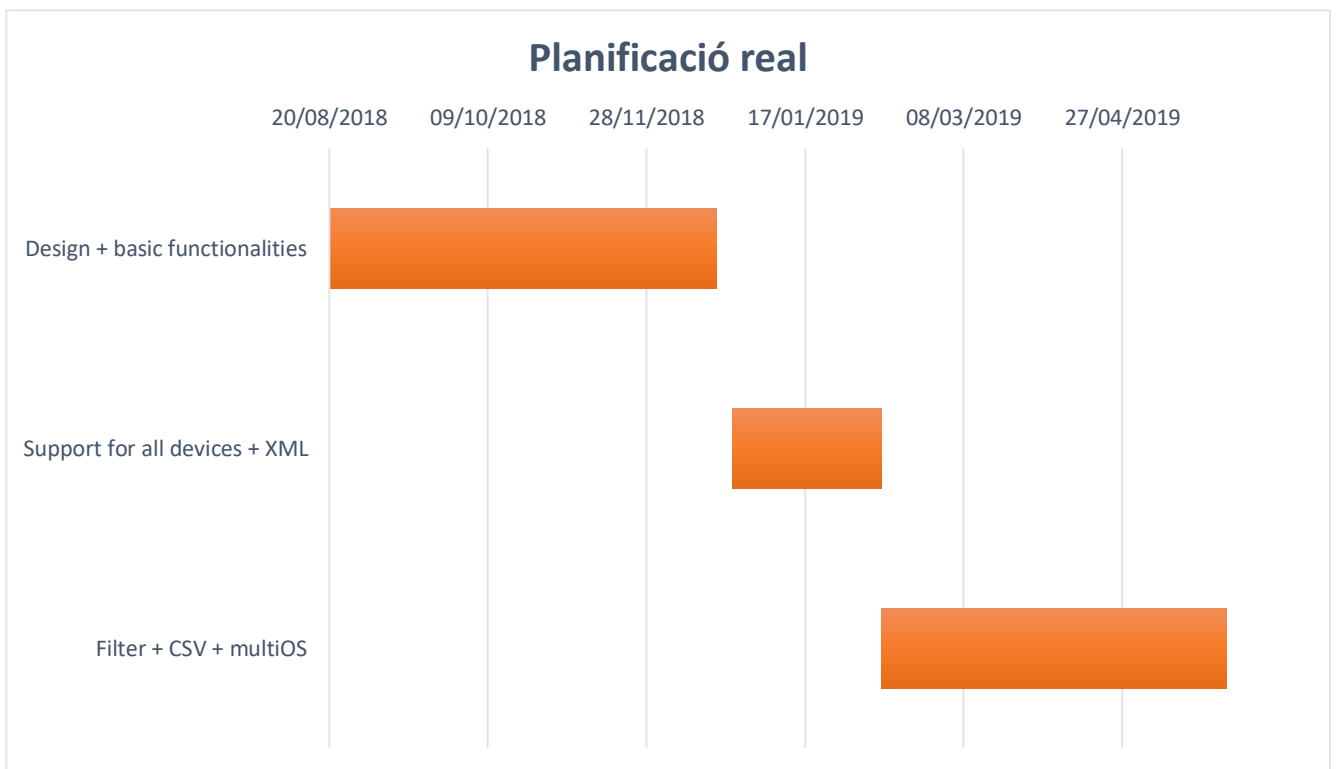
4.3 Diagrama de Gant

Tota la planificació comentada en els punts 4.1 Planificació estimada i 4.2 Temps invertit real és pot plasmar en dos diagrames de Gant.

II-lustració 7. Planificació estimada



II-lustració 8. Planificació real



5. Valoració econòmica

Com tot projecte que es vulgui dur a terme, s'ha de valorar el cost de realització. En aquest cas es tracta d'una aplicació que s'entregarà de forma gratuïta a tots els clients a través de la pròpia web de l'empresa com a valor afegit. Per tant, no es pot calcular el rendiment que se li traurà a la inversió.

El cost aproximat de realització el calcularem mitjançant les hores invertides d'enginyer, el cost de les llicències, si se'n fes servir alguna de pagament, el cost del maquinari per desenvolupar, etc.

5.1 Estimació econòmica

Taula 5. Estimació econòmica

Cost econòmic PAMTool			
Concepte	Quantitat	Preu unitat	Total
Hores desenvolupament	532	15 €	7980 €
Hores disseny	144	15 €	2160 €
Hores documentació	36	15 €	540 €
Equipament informàtic	1	1300 €	1300 €
Total			11980 €

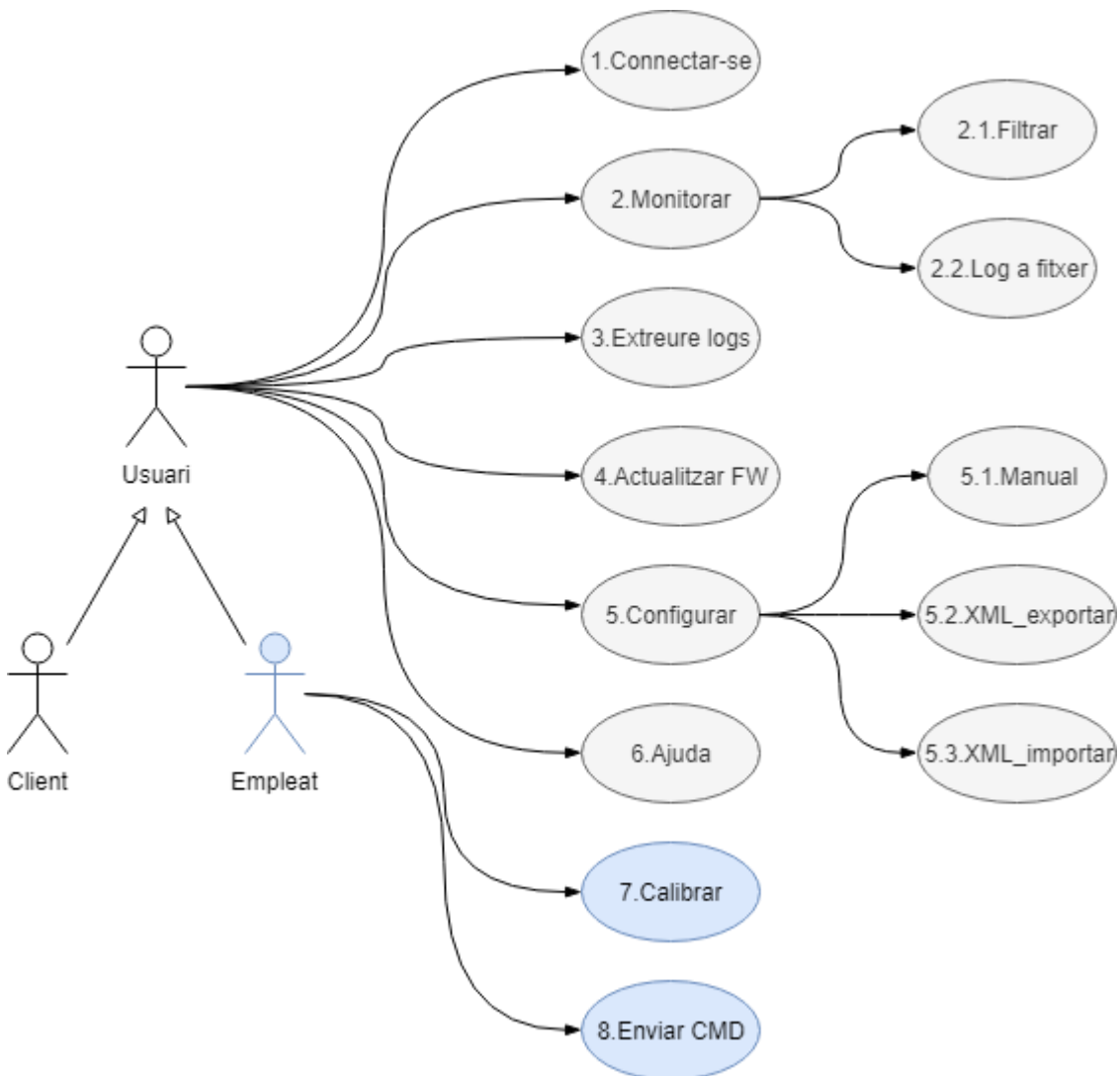
El preu de la hora invertida de 15 € surt del càlcul sobre el cost mensual de l'empresa sobre un treballador de grup 1 del conveni d'indústria siderometal·lúrgica de la província de Barcelona [12]. Referent a l'equipament informàtic, els 1300€ és el preu del "MSI P65 Creator 8RD-095XES", ja que és l'equip que s'ha fet servir en el desenvolupament de l'aplicació.

6. Anàlisi

6.1 Diagrama de casos d'ús

La *Il·lustració 9. Diagrama de casos d'ús* ens explica visualment quines són les interaccions i possibilitats dels diferents actors del sistema en el nostre projecte. En el nostre cas tenim un únic actor principal que seria l'usuari, el qual pot fer gairebé totes les interaccions, excepte les reservades per a empleats, que seria una extensió de l'usuari. Aquest usuari serà un client o un empleat.

Il·lustració 9. Diagrama de casos d'ús



6.2 Casos d'ús textuais

A banda de llistar els diferents casos d'ús també és necessari definir-los un per un i detallar com serà la interacció usuari – sistema.

Cas d'ús 1: Connectar-se	
Descripció	L'usuari estableix una connexió amb el dispositiu desitjat.
Actors	Usuari.
Pre-condició	Tenir un canal físic de connexió amb el dispositiu. (<i>Bluetooth, rs232, CAN...</i>)
Flux bàsic	<ol style="list-style-type: none"> 1. El cas d'ús comença quan l'usuari inicia l'aplicació. 2. L'usuari selecciona quin tipus de connexió té amb el dispositiu. En el moment d'entrega d'aquest projecte només es suporta: Sèrie o CAN-bus 3. L'usuari selecciona el COM on té la connexió física amb el dispositiu. 4. L'usuari prem el botó de "Connect". 5. El sistema establirà una connexió amb el dispositiu 6. El sistema mostrarà el codi, la imatge i paràmetres més importants del dispositiu a la pestanya "General".
Flux alternatiu	<p>3,5</p> <p>3) 3.1 Si el canal físic no apareix al desplegable de COM, l'usuari revisarà si el canal físic està ben establert i clicarà el botó de "Refresh". Es torna al pas 3.</p> <p>5) 5.1 El sistema no pot establir una connexió amb el dispositiu. Es torna al pas 4.</p>
Post-condició	El sistema té una connexió establerta amb el dispositiu.

Cas d'ús 2: Monitorar	
Descripció	L'usuari monitoritza els paràmetres del dispositiu
Actors	Usuari.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari navega a la pestanya "General" o la pestanya "Detail". 2. El sistema mostra la pestanya seleccionada. 3. L'usuari selecciona la cadència d'actualització de les dades monitoritzables. 4. El sistema demanarà les dades al dispositiu amb la cadència seleccionada. 5. El sistema refrescarà la interfície amb les dades demanades al dispositiu. 6. Dependent de la cadència seleccionada es tornarà al pas 4.
Post-condició	El sistema mostrarà les dades actualitzades

Cas d'ús 2.1: Filtrar	
Descripció	L'usuari filtra les dades que vol monitorar
Actors	Usuari.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu.
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari navega a la pestanya "Detail". 2. El sistema mostra la pestanya seleccionada. 3. L'usuari selecciona o des selecciona les dades que vol monitoritzar de la taula "Monitoring". 4. L'usuari selecciona l'opció "Filter". 5. El sistema amaga els paràmetres que no estiguessin seleccionats.
Flux alternatiu	<p>3</p> <ol style="list-style-type: none"> 3) 3.1 L'usuari des selecciona l'opció de "Select all" 3.2 El sistema des selecciona totes les dades de la taula "Monitoring"
Post-condició	El sistema mostra les dades monitoritzables de la taula "Monitoring" filtrades.

Cas d'ús 2.2: "Log" a fitxer	
Descripció	L'usuari selecciona quines dades vol guardar en un fitxer durant un període de temps o un número determinat de iteracions.
Actors	Usuari.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu.
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari navega a la pestanya "Detail" 2. El sistema mostra la pestanya seleccionada. 3. L'usuari filtra les dades que vol enregistrar al fitxer. 4. L'usuari prem el boto "Log to file" 5. El sistema mostra la finestra de "Log to file" 6. L'usuari introdueix la localització on vol que es generi el fitxer i el nom que vol que tingui aquest. 7. L'usuari selecciona quina vol que sigui la condició de finalització de l'enregistrament. 8. L'usuari ha seleccionat límit de temps com a condició, per tant introdueix quants dies, hores, minuts i/o segons vol que duri l'enregistrament. 9. L'usuari prem el botó "Save". 10. El sistema guarda la configuració i inicia l'enregistrament. 11. L'usuari selecciona la cadència d'actualització de les dades que enregistrarà. 12. El sistema enregistrarà les dades al fitxer fins que el temps seleccionat s'esgoti.
Flux alternatiu	<p>8</p> <ol style="list-style-type: none"> 8) 8.1 L'usuari selecciona l'opció de límit per número de lectures.

	8.2 L'usuari introdueix el numero de lectures que vol que s'enregistrin. Es salta al pas 9.
Post-condició	El sistema ha generat un fitxer amb extensió .csv amb les dades enregistrades.

Cas d'ús 3: Extreure "logs"	
Descripció	Demana els "logs" interns del dispositiu i els comprimeix juntament amb els "logs" de l'aplicació i el fitxer <i>xml</i> de la configuració actual.
Actors	Usuari.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu.
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari clica sobre el menú desplegable superior "Help". 2. El sistema mostra totes les opcions dintre del menú desplegable. 3. L'usuari prem sobre l'opció "Export logs" 4. El sistema demana a l'usuari la ubicació on desitja els "logs". 5. L'usuari tria la ubicació i el nom del fitxer amb extensió ".zip" de "logs". 6. El sistema demana els "logs" interns al dispositiu. 7. El sistema demana els valors de configuració del dispositiu i genera un fitxer XML. 8. El sistema comprimeix el fitxer de "logs" interns del dispositiu, el fitxer de configuració en format XML i els "logs" de l'aplicació a la ubicació seleccionada.
Flux alternatiu	<p>6</p> <p>6) 6.1 Si el dispositiu no suporta la comanda el pas 6 no es realitza.</p>
Post-condició	S'ha creat un fitxer amb extensió ".zip" a la ubicació seleccionada amb informació sobre el funcionament del sistema.

Cas d'ús 4: Actualitzar "firmware"	
Descripció	S'envia un nou fitxer de "firmware" al dispositiu per actualitzar la seva versió.
Actors	Usuari.
Pre-condició	<p>El sistema necessita tenir una connexió establerta amb el dispositiu.</p> <p>El dispositiu ha de suportar el "Protocol", en cas negatiu no suporta la funció d'actualitzar el <i>firmware</i>.</p>
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari navega fins la pestanya de "Update". 2. El sistema mostra la pestanya seleccionada. 3. L'usuari clica sobre el botó "Browse file" i selecciona el fitxer de <i>firmware</i> corresponent. 4. El sistema mostra la versió de <i>firmware</i> a enviar. 5. L'usuari prem sobre el botó "Update".

	6. El tema inicia la transferència del fitxer al dispositiu i espera a que el dispositiu s'hagi actualitzat.
Flux alternatiu	6 6) 6.1 El dispositiu rebutja l'actualització i es torna al pas 3.
Post-condició	El dispositiu està actualitzat a la versió del fitxer enviat.

Cas d'ús 5.1 : Configurar Manual	
Descripció	Configura el dispositiu de forma manual modificant un per un els paràmetres desitjats.
Actors	Usuari.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu.
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari navega fins a la pestanya "Detail". 2. El sistema mostra els continguts de la pestanya seleccionada. 3. L'usuari clica dues vegades sobre la cel·la "Value" de la fila de la dada que vol configurar. 4. L'usuari introdueix un valor i prem la tecla "Enter" per confirmar la seva selecció. 5. L'usuari repeteix el pas 4 per tots els valors configurables que vulgui enviar. 6. L'usuari prem el botó "Apply changes". 7. El sistema envia les noves dades al dispositiu 8. El dispositiu confirma que les dades s'han rebut correctament. 9. El sistema informa a l'usuari que la configuració s'ha aplicat correctament.
Flux alternatiu	<p>3, 4, 8</p> <p>3) 3.1 Alternativament alguns paràmetres tenen un desplegable en la cel·la "Value" i s'ha d'escollir entre les opcions disponibles. Es salta al pas 5.</p> <p>4) 4.1 El sistema detecta que el valor introduït no compleix el format i mostra un missatge d'error. Es torna al pas 3.</p> <p>4) 4.1 El sistema detecta que el valor introduït no està dins dels límits superiors o inferiors de la dada a enviar i es mostra la cel·la de color vermell. Es torna al pas 4.</p> <p>8) 8.1 El dispositiu informa d'algun error en les noves dades a configurar.</p> <p>8.2 El sistema informa a l'usuari de l'error. Es torna al pas 3.</p>
Post-condició	El dispositiu es troba configurat amb la nova configuració enviada per l'usuari.

Cas d'ús 5.2: Configurar, XML_exportar	
Descripció	El sistema genera un fitxer XML amb la configuració actual del dispositiu
Actors	Usuari.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu.
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari clica sobre el menú desplegable superior "Tools". 2. El sistema mostra totes les opcions dintre del menú desplegable. 3. L'usuari prem sobre l'opció "Export configuration". 4. El sistema demana a l'usuari la ubicació i nom del fitxer a generar. 5. L'usuari selecciona on vol el fitxer i el seu nom i prem el botó "Guardar" 6. El sistema demana la configuració actual al dispositiu 7. El dispositiu respon la seva configuració. 8. El sistema genera un fitxer XML que conté, entre altra informació rellevant, la configuració actual del dispositiu.
Flux alternatiu	<p>7</p> <p>7) 7.1 El dispositiu contesta algun tipus d'error</p> <p>7.2 El sistema informa que no s'ha pogut generar el fitxer XML de configuració. Es torna al pas 1.</p>
Post-condició	S'ha generat i guardat un fitxer XML de configuració en la ubicació escollida per l'usuari.

Cas d'ús 5.3: Configurar, XML_importar	
Descripció	S'importa una configuració prèviament exportada a un fitxer XML i es configura el dispositiu.
Actors	Usuari.
Pre-condició	<p>El sistema necessita tenir una connexió establerta amb el dispositiu.</p> <p>Disposar d'un fitxer de configuració XML.</p>
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari clica sobre el menú desplegable superior "Tools". 2. El sistema mostra totes les opcions dintre del menú desplegable. 3. L'usuari prem sobre l'opció "Import configuration". 4. El sistema mostra una nova finestra per importar la configuració. 5. L'usuari prem sobre el botó "Browse". 6. L'usuari busca i tria el fitxer XML amb la configuració a importar. 7. El sistema mostra a la taula el nom i el valor que es configurarà de cada paràmetre. 8. L'usuari prem el botó "Apply changes". 9. El sistema envia la nova configuració al dispositiu. 10. El dispositiu es configura amb els nous paràmetres enviats. 11. El sistema informa que la configuració s'ha completat correctament.

Flux alternatiu	<p>7</p> <p>7)</p> <p>7.1 El sistema alternativament pot mostrar un signe de creu vermella al costat d'un paràmetre per informar que aquest no està dintre dels límits configurables.</p> <p>7.2 El sistema alternativament pot mostrar un signe d'igual taronja al costat del paràmetre per informar que aquest és igual a la configuració actual del dispositiu i per tant, no s'aplicarà.</p>
Post-condició	El sistema queda configurat amb la nova configuració importada per l'arxiu XML.

Cas d'ús 6: Ajuda	
Descripció	L'usuari accedeix a l'ajuda de l'aplicació.
Actors	Usuari.
Pre-condició	Cap.
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari clica sobre el menú desplegable superior "Help". 2. El sistema mostra totes les opcions dintre del menú desplegable. 3. L'usuari prem sobre l'opció "Help". 4. El sistema mostra una nova finestra per mostrar l'Ajuda. 5. L'usuari pot navegar i llegir informació sobre cada pestanya per conèixer les seves funcionalitats bàsiques. 6. L'usuari pot obrir l'arbre anomenat "Supported devices" per visualitzar quins dispositius suporta la versió actual de l'aplicació.
Flux alternatiu	Cap.
Post-condició	L'usuari queda informat de les funcionalitats bàsiques de l'aplicació.

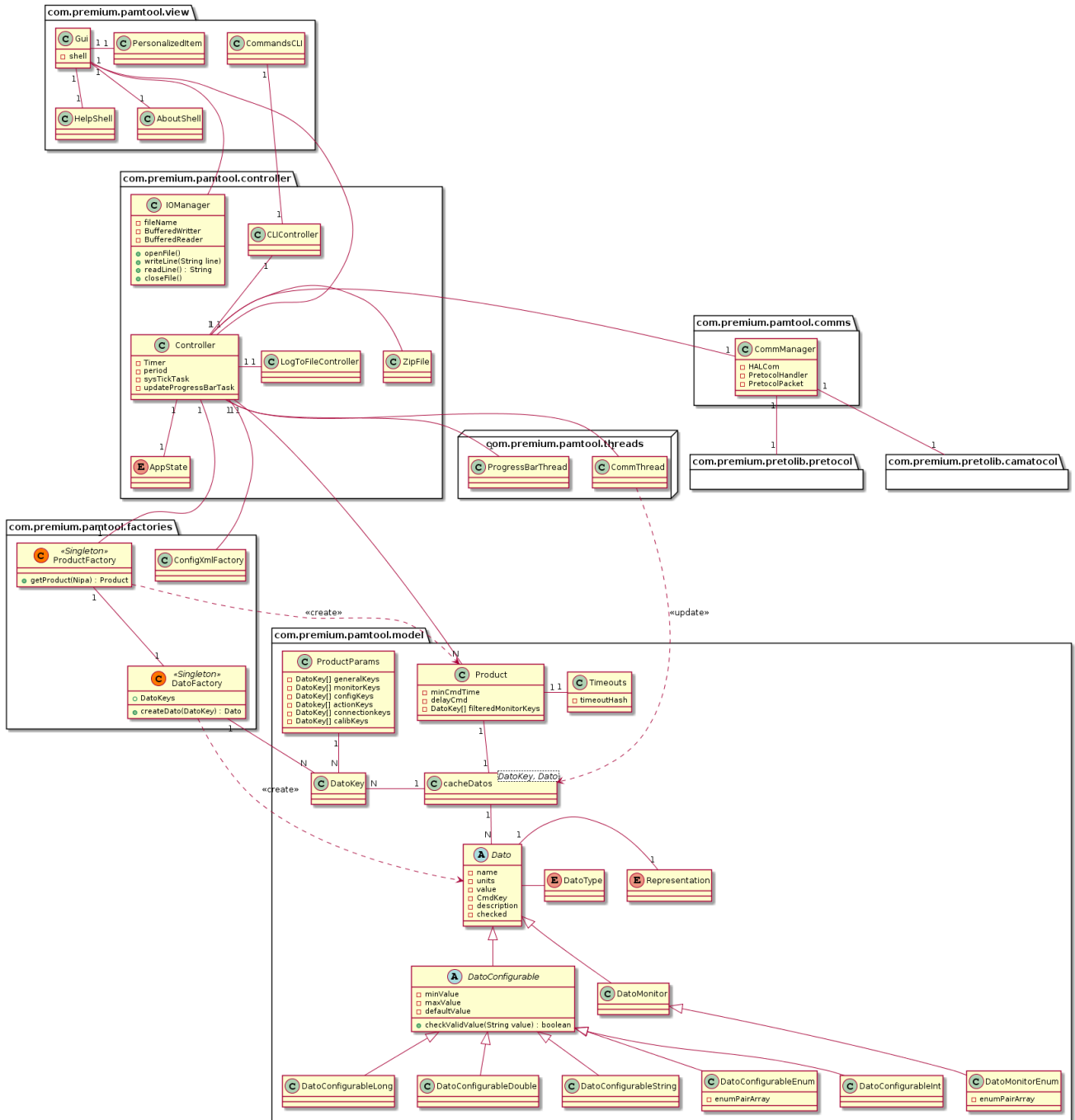
Cas d'ús 7: Calibrar	
Descripció	Es configura el dispositiu amb paràmetres protegits que només es poden configurar un cop.
Actors	Empleat.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu. El sistema necessita estar obert amb la versió per desenvolupadors, normalment anomenada “*_dev”.
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari navega fins la pestanya “Calibration”. 2. El sistema mostra els continguts de la pestanya seleccionada. 3. L'usuari clica dues vegades sobre la cel·la “Value” de la fila de la dada que vol configurar. 4. L'usuari introdueix un valor i prem la tecla “Enter” per confirmar la seva selecció. 5. L'usuari repeteix el pas 4 per tots els valors calibrables que vulgui enviar. 6. L'usuari prem el botó “Apply changes”. 7. El sistema envia les noves dades al dispositiu. 8. El dispositiu confirma que les dades s'han rebut correctament. 9. El sistema informa a l'usuari que la calibració s'ha aplicat correctament.
Flux alternatiu	<p>3, 4, 8</p> <p>3) 3.1 Alternativament, alguns paràmetres tenen un desplegable a la cel·la “Value” i s'ha d'escollir entre les opcions disponibles. Es salta al pas 5.</p> <p>4) 4.1 El sistema detecta que el valor introduït no compleix el format i mostra un missatge d'error. Es torna al pas 3.</p> <p>4) 4.1 El sistema detecta que el valor introduït no està dintre dels límits superiors o inferiors de la dada a enviar i es mostra la cel·la de color vermell. Es torna al pas 4.</p> <p>8) 8.1 El dispositiu informa d'algun error a les noves dades a calibrar.</p> <p>8.2 El sistema informa a l'usuari de l'error. Es torna al pas 3.</p>
Post-condició	El dispositiu queda calibrat amb les dades noves enviades i no acceptarà que aquestes es tornin a modificar si no es programa de nou el dispositiu.

Cas d'ús 8: Enviar CMD	
Descripció	S'envia una comanda amb caràcters ASCII al dispositiu i es rep una resposta.
Actors	Empleat.
Pre-condició	El sistema necessita tenir una connexió establerta amb el dispositiu. El sistema necessita estar obert amb la versió per desenvolupadors, normalment anomenada "*_dev".
Flux bàsic	<ol style="list-style-type: none"> 1. L'usuari navega fins la pestanya "Developer". 2. El sistema mostra els continguts de la pestanya seleccionada. 3. L'usuari introdueix una comanda a l'entrada de text inferior de la pestanya. 4. L'usuari prem la tecla "Enter" per enviar la comanda. 5. El sistema mostra la comanda a la "consola" superior de la pestanya i envia la comanda al dispositiu. 6. El dispositiu rep la comanda i la processa. 7. El dispositiu respon a la comanda. 8. El sistema mostra la resposta a la comanda a la "consola" superior de la pestanya.
Flux alternatiu	Cap.
Post-condició	El dispositiu ha rebut i respost una comanda per la línia de comandos de l'aplicació.

6.3 Model de domini

II-lustració 10. Model de domini

PAMTool - Class Diagram



Aquest model de domini és una versió simplificada del real, ja que no seria possible plasmar-lo en una pagina dinA4. A més, dificultaria molt la seva comprensió. Tal i com es pot apreciar, l'aplicació està segmentada entre diferents paquets seguint el patró de disseny MVC⁴.

⁴ Model-View-Controller

6.4 Interfícies

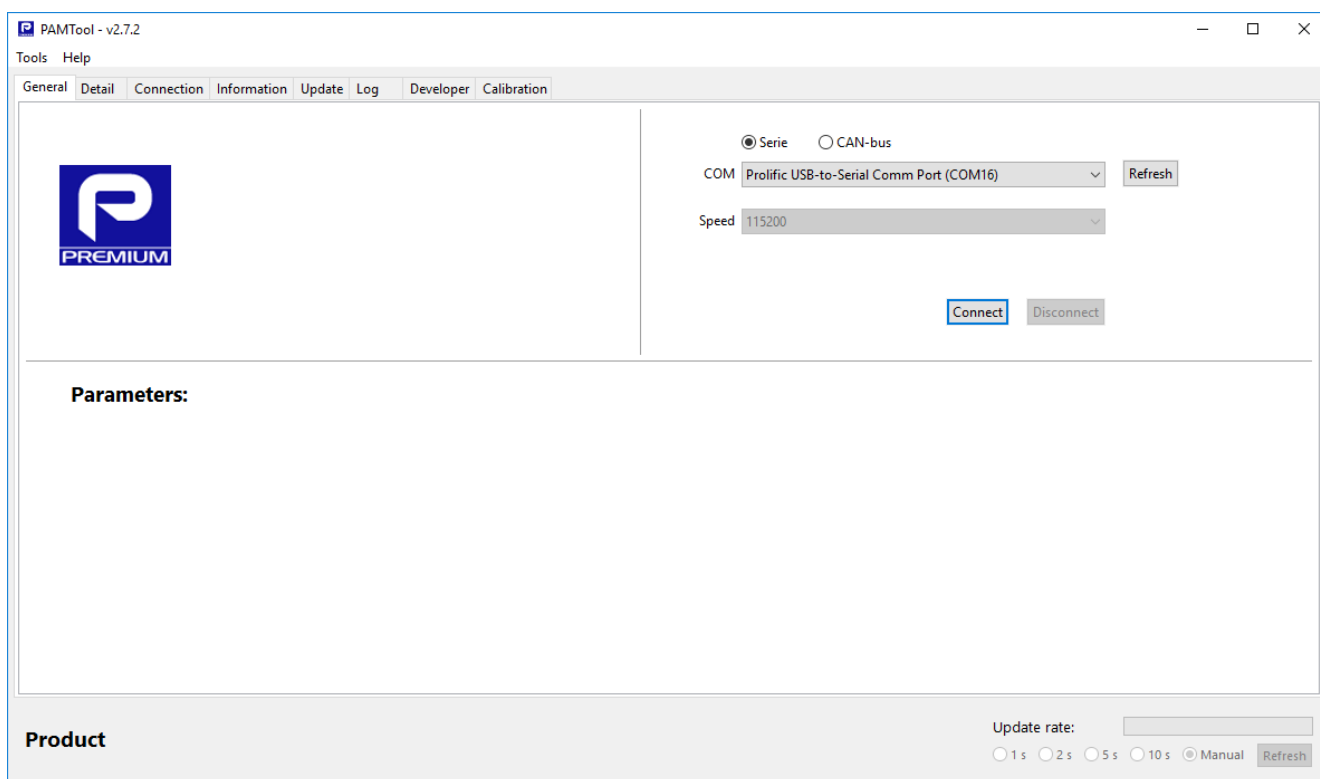
En una aplicació on l'usuari serà tant un enginyer amb experiència com un client que no necessàriament haurà de tenir coneixement de la matèria, la interfície gràfica de l'aplicació és tant important com que el seu funcionament sigui correcte.

En el cas de la GUI s'ha escollit SWT com a tecnologia per desenvolupar-la, ja que la seva interfície utilitza les eines natives del sistema. Per tant, dóna una gran sensació de familiaritat. En tot moment s'ha dissenyat amb la intenció que la aplicació sigui auto-explicativa i es pugui prescindir del manual.

6.4.1 Pestanya "General"

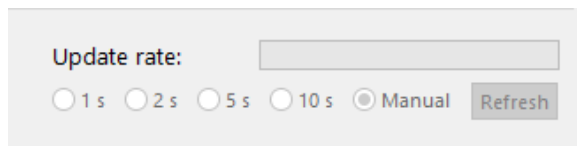
En la *Il·lustració 11. PAMTool, pestanya "General"* es poden dur a terme varies funcions com ara establir una connexió, escollir els paràmetres adequats, finalitzar una connexió i canviar la freqüència d'actualització de dades de monitoratge mitjançant el grup de botons inferior dret. Aquest grup de botons és independent de la pestanya en què estem ubicats.

Il·lustració 11. PAMTool, pestanya "General"

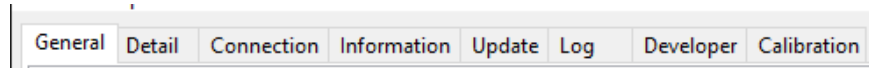


Es pot apreciar clarament una variable visual en el botó "Connect". En el disseny, no només d'aquesta pestanya sinó de totes, s'ha fet ús de l'espai en blanc per agrupar i/o separar components de la interfície. En aquest cas concret es pot veure amb els botons de "Connect" i "Disconnect" que és gracies a l'**espai en blanc** aplicant el **principi de Gesalt de proximitat, similitud i continuïtat**. Per altra banda, tot el bloc referent a la connexió està separat per més espais i un marge per denotar la seva diferència a la resta de la pestanya i agrupar en aquest punt tot el relacionat a la connexió amb el dispositiu.

A la part inferior de l'aplicació hi ha dos grups importants que es mantenen independents de la pestanya on siguem. A la cantonada inferior esquerra tenim una etiqueta on es mostrarà el codi del producte i el seu nom, si en té. A la cantonada inferior dreta tenim un altre grup clarament aïllat, els botons de tria de freqüència d'actualització i una barra de progrés que ens informa de l'estat.



II·lustració 12. Botons de tria de freqüència d'actualització



II·lustració 13. Barra de pestanyes

A la II·lustració 12. Botons de tria de freqüència d'actualització i la II·lustració 13. Barra de pestanyes es pot observar que amb un simple cop d'ull el cervell humà entén que estan agrupats i que tenen relació entre ells. En el cas de la II·lustració 13. Barra de pestanyes s'aconsegueix gràcies als principis de Gestalt mencionats anteriorment, similitud i continuïtat.

6.4.2 Pestanya “Detail”

II·lustració 14. Pestanya "Detail"

Configuration

Name	Value	Configured	Min	Max	Default	Units
Output state		0	0	0	0	
UVLO(dc)		-	16.8	30.0	-	V
UVLOAlrm(dc)		-	0.0	21.6	-	V
VoutRMS(ac)		-	184.0	241.5	-	V
IoutMax(ac)		-	1.3	6.5	-	A
FoutNom			0	0	0	Hz
IoutAlrm(ac)		-	0.0	6.5	-	A

Monitoring Select all Filter

Name	Value	Units
<input checked="" type="checkbox"/> VoutRMS(ac)	200.0	V
<input checked="" type="checkbox"/> VoutSts(ac)	ON	V
<input checked="" type="checkbox"/> FoutNom(ac)	50.0	Hz
<input checked="" type="checkbox"/> Vi(dc)	24.1	V
<input checked="" type="checkbox"/> IoutRMS(ac)	0.1	A
<input checked="" type="checkbox"/> Pout(ac)	3	W
<input checked="" type="checkbox"/> Temp (int)	25.0	°C

7113-ODS1500

Update rate: 1 s 2 s 5 s 10 s Manual

En aquesta pestanya es poden diferenciar dos blocs. A l'esquerra el bloc de configuració i a la dreta el de monitorització.

El bloc de configuració consta d'una taula i 3 botons. La taula ens llista tots els paràmetres ajustables del dispositiu, el botó per aplicar els canvis de la taula, el botó “Commands” per mostrar les accions a enviar al dispositiu (“Hard reset”, “Soft reset”, etc) i el botó “Refresh” que ens refresca el valor configurat de cada dada del dispositiu.

Per altra banda, el bloc de monitorització ens permet fer tot el relacionat amb el monitoratge. A través de la casella de cada dada es pot triar si es vol refrescar aquesta dada o no. Si s'activa la casella "Filter", totes les dades amb la casella des seleccionada es trauran de la taula fins que es torni a des seleccionar la casella "Filter". D'aquesta manera, si el paràmetre disposa de desenes de dades de monitoratge, l'usuari es pot centrar en les que ell necessiti. Seguint amb la dinàmica de les dades seleccionades hi ha l'opció de generar el fitxer de "logs" mitjançant el botó "Log to file".

Il·lustració 15. Finestra "Log to file"

Log to file config - PAMTool

Location: C:\Users\jarmengo\Desktop Folder...

File name: test_log_YYYYMMDD_HHMMSS.csv

STOP CONDITION

Time Num of readings

Time limit

Days Hours Minutes Seconds

0 0 1 0

Readings limit

Readings

0

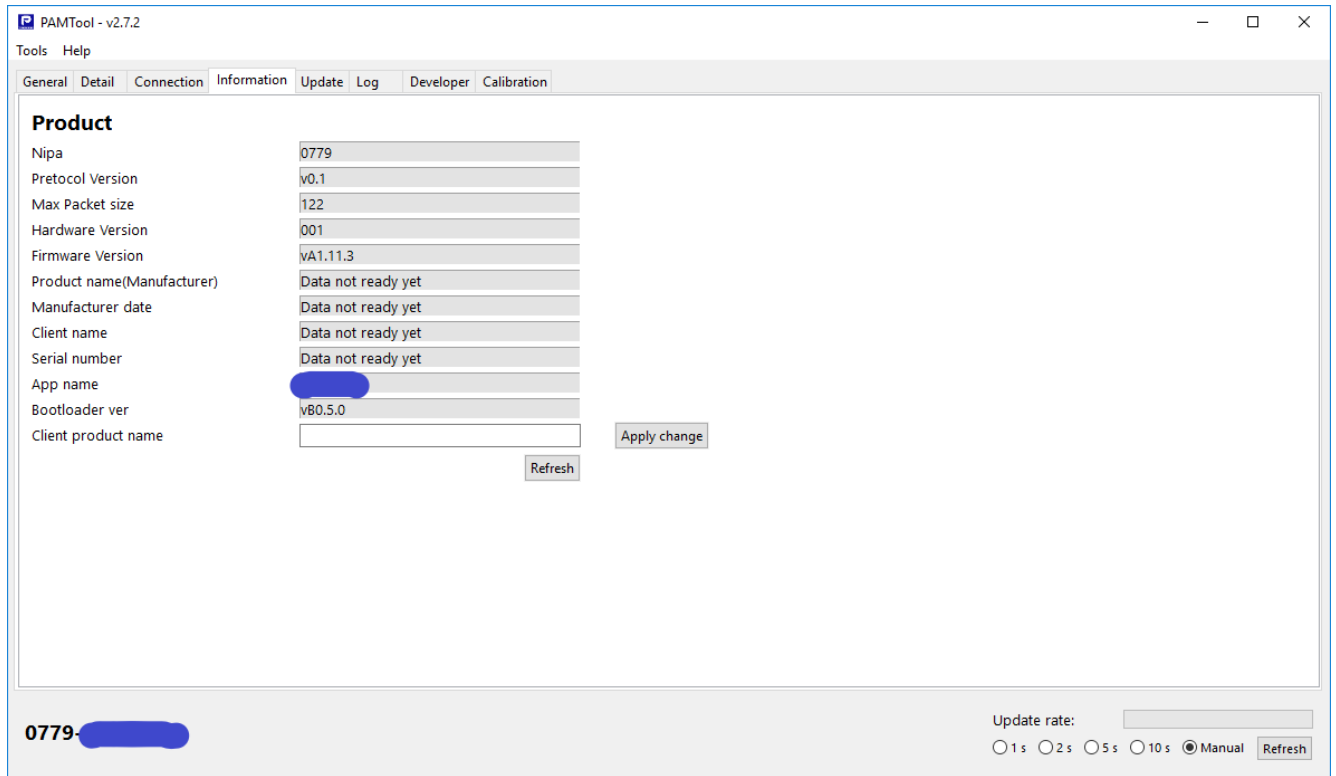
Save Cancel

Al pressionar el botó "Log to file" s'obre una nova finestra de configuració amb un formulari i diverses opcions. Per poder iniciar l'enregistrament és necessari emplenar-lo i en quan li donem al botó de guardar, iniciarà l'enregistrament de les dades que estiguin seleccionades en aquell instant.

6.4.4 Pestanya "Information"

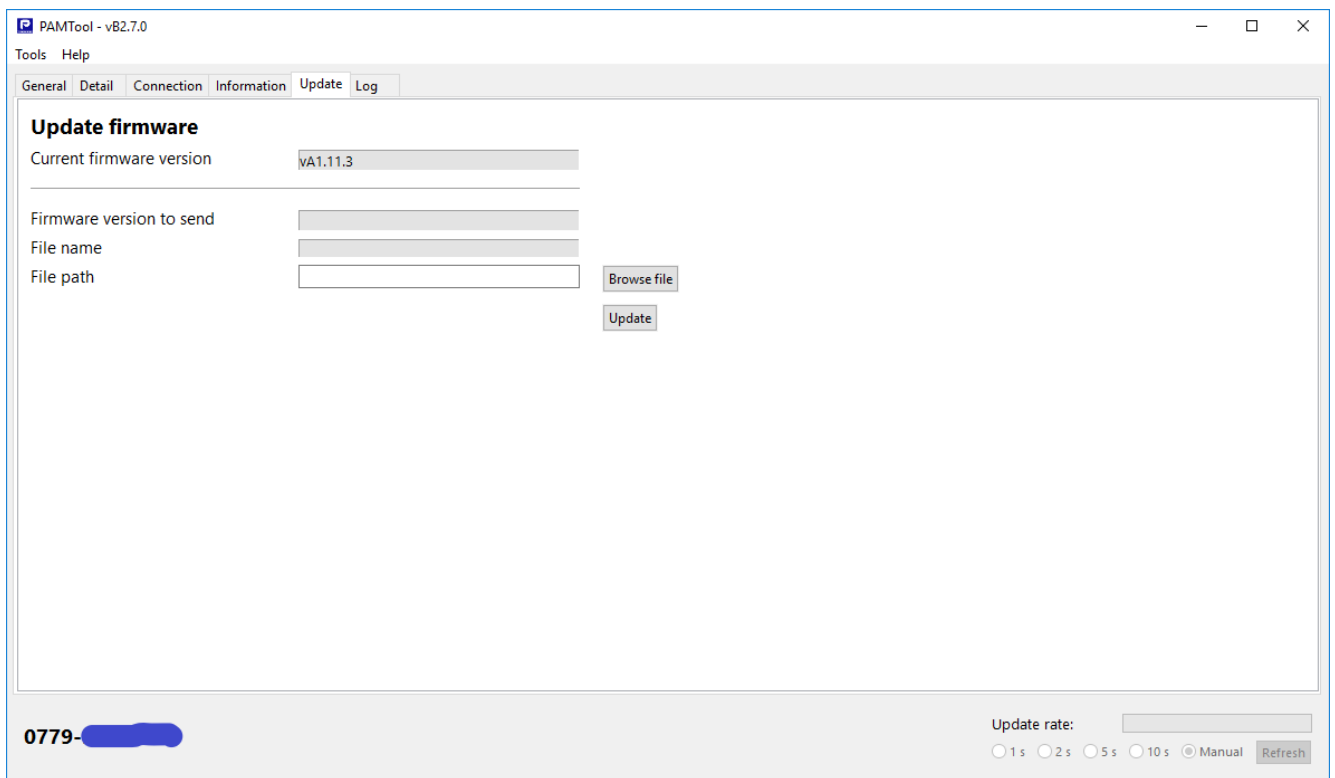
En aquesta pestanya es mostra un seguit de dades referents a informació tècnica del dispositiu. Depenent del protocol del dispositiu es mostrarà el número de referència del dispositiu (NIPA) i versió de *firmware* si suporta el protocol antic o bé molts més paràmetres com número de sèrie, data de fabricació, versió del protocol, ... si suporta el pretocol.

Il·lustració 17. Pestanya "Information"



6.4.5 Pestanya “Update”

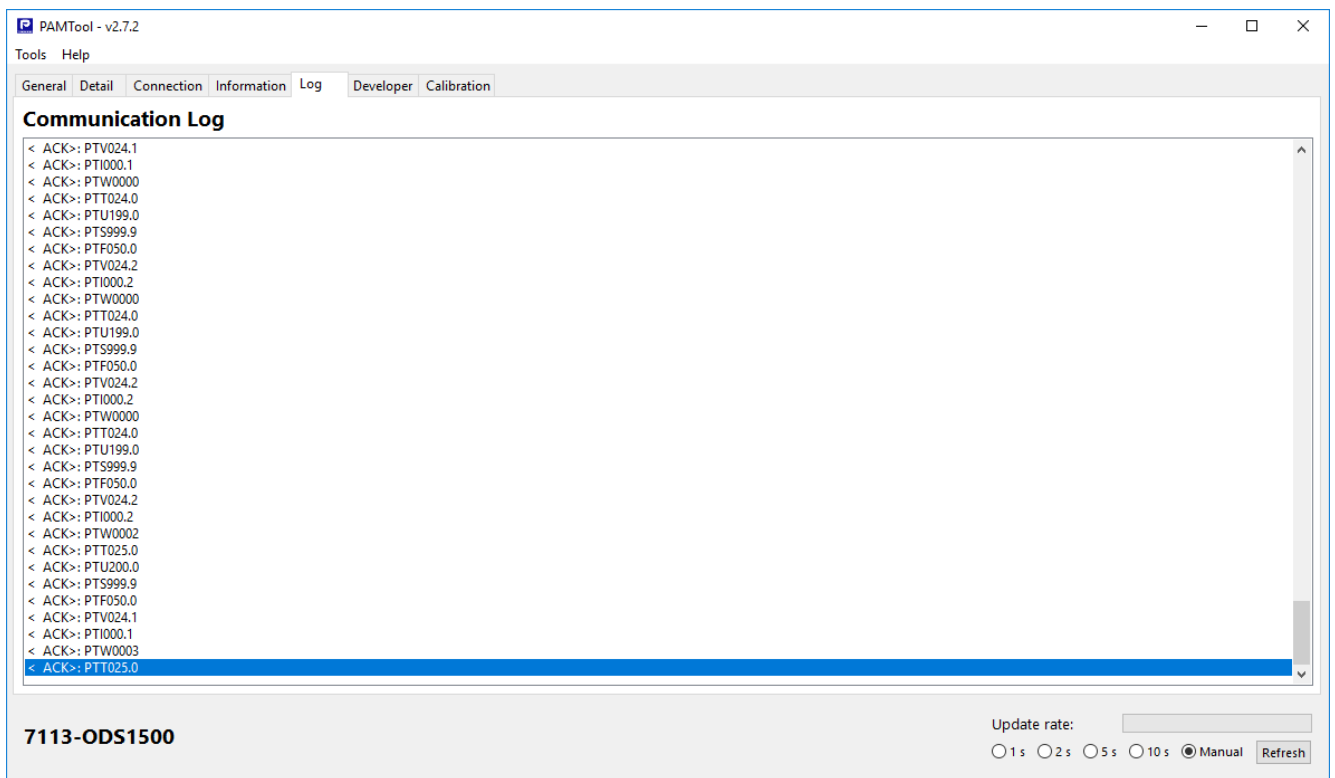
Il·lustració 18. Pestanya "Update"



Aquesta pestanya només serà visible si el dispositiu connectat ho suporta. És aquí on es mostra la versió actual de *firmware* i la informació relativa a l'actualització, ambdues separades per una línia. Consta de dos botons, un per obrir un diàleg de Windows de cerca i selecció de fitxer i un altre per iniciar l'actualització. Mentre l'actualització estigui en curs, la barra inferior esquerra ens mostrarà un progrés indefinit fins que el procés s'acabi i una finestra emergent ens informi.

6.4.5 Pestanya "Log"

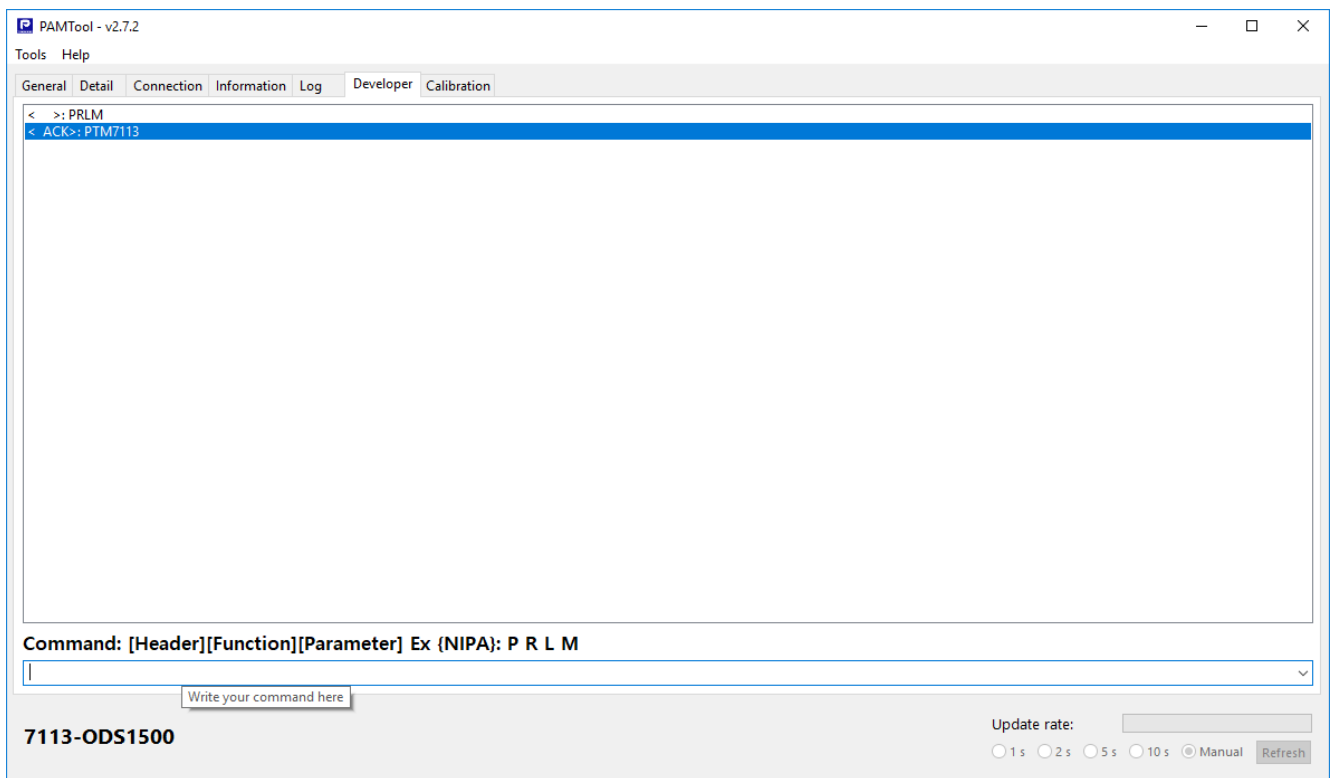
Il·lustració 19. Pestanya "Log"



En aquesta pestanya es mostra un sol requadre on a mesura que vagin enviant-se i rebent-se trames seran mostrades per aquí. D'aquesta manera si hi hagués algun problema amb una dada en concret, serà molt senzill tant pel client com per l'empleat trobar l'error que podria estar passant.

6.4.5 Pestanya “Developer”

Il·lustració 20. Pestanya "Developer"



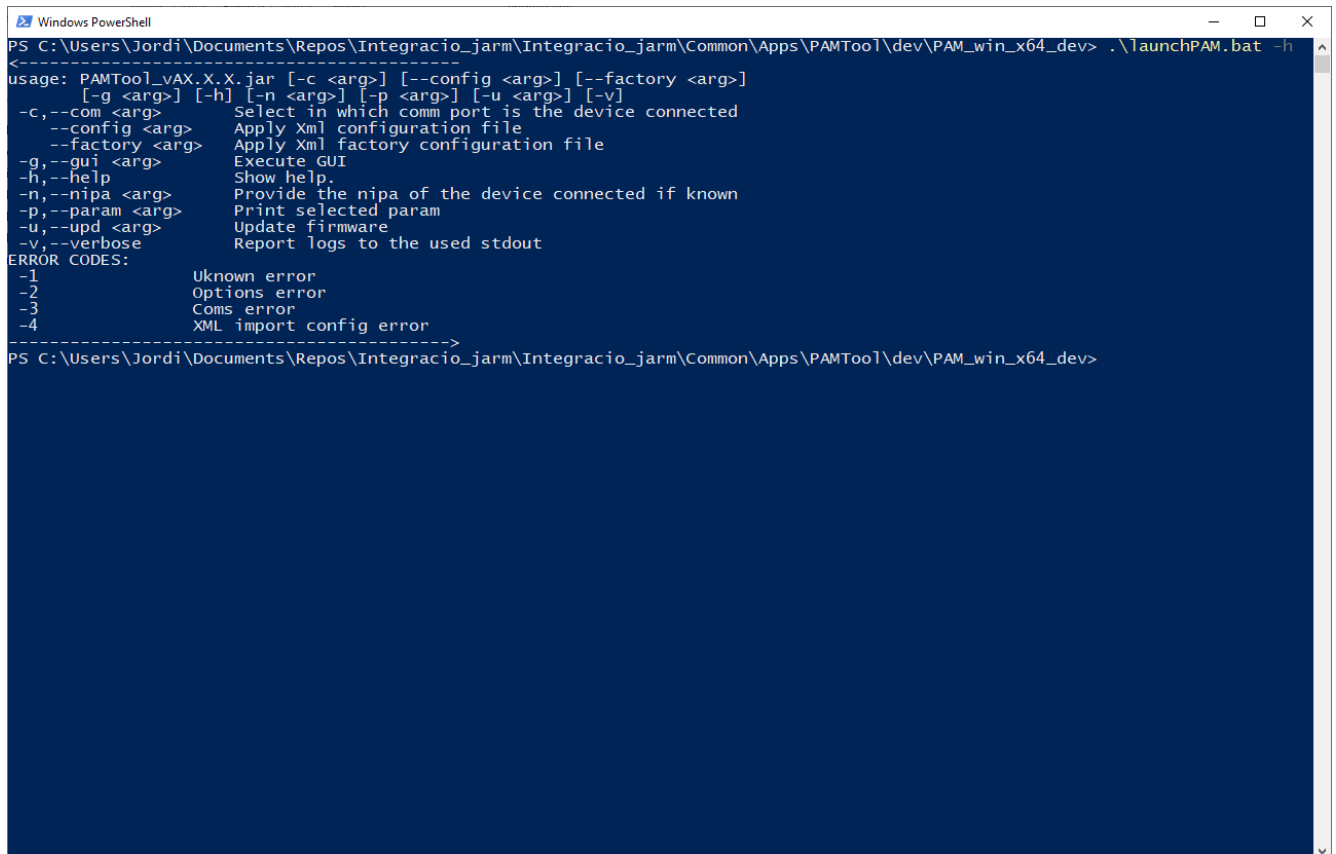
En aquesta pestanya s'emula el comportament que es podria tenir amb altres aplicacions que ens proporcionen un terminal de comunicacions via COM ports com Teraterm, Putty, etc. D'aquesta manera, si escrivim una comanda dintre del requadre baix de "Command:" i pressionem la tecla "Enter", s'enviarà la comanda al dispositiu i al requadre superior se'ns mostrarà tant la comanda enviada com la resposta del dispositiu. Aquesta pestanya només està disponible per a les versions generades per a empleats.

6.4.6 CLI

En el cas de la CLI està més pensada per a usuaris avançats. Disposa de moltes funcions i possibilitat per automatitzacions. La capa del controlador i model és la mateixa que amb la GUI i per tant, les funcions són un subgrup de les totals.

Si hi ha algun tipus d'error durant l'execució, la aplicació retornarà un codi d'error de sistema amb un dels valors indicats a l'ajuda (-h). Això li proporciona una gran integració amb el procés de fabricació de dispositius per poder controlar molt bé una tasca tant crítica.

Il·lustració 22. CLI help



```
Windows PowerShell
PS C:\Users\Jordi\Documents\Repos\Integracio_jarm\Integracio_jarm\Common\Apps\PAMTool\dev\PAM_win_x64_dev> .\launchPAM.bat -h
usage: PAMTool_vAX.X.X.jar [-c <arg>] [--config <arg>] [--factory <arg>]
      [-g <arg>] [-h] [-n <arg>] [-p <arg>] [-u <arg>] [-v]
-c,--com <arg>      Select in which comm port is the device connected
--config <arg>     Apply Xml configuration file
--factory <arg>    Apply Xml factory configuration file
-g,--gui <arg>     Execute GUI
-h,--help           Show help
-n,--nipa <arg>    Provide the nipa of the device connected if known
-p,--param <arg>   Print selected param
-u,--upd <arg>     Update firmware
-v,--verbose       Report logs to the used stdout
ERROR CODES:
-1                Unknown error
-2                Options error
-3                Coms error
-4                XML import config error
PS C:\Users\Jordi\Documents\Repos\Integracio_jarm\Integracio_jarm\Common\Apps\PAMTool\dev\PAM_win_x64_dev>
```

El comportament decidit en la CLI implica que només es pot executar una de les principals funcions per cada cop que s'executa l'aplicació. En cas contrari, mostra un error si s'ha seleccionat l'opció -v i si no retornarà un codi d'error -2.

7. Disseny i implementació

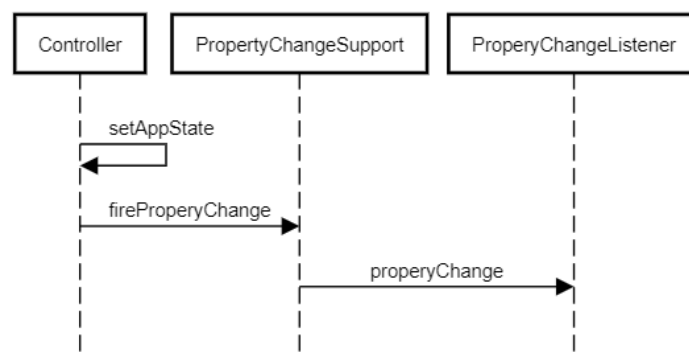
En aquest apartat s'explicarà els fonaments més importants i rellevants del projecte a nivell de disseny i implementació. En una aplicació que ha de tenir dues opcions de vista del model MVC és molt important fer una bona segmentació de les classes per responsabilitats. D'aquesta manera, el mateix model i controlador actuen per les dues vistes indiferentment.

7.1 Patró observador

Tal i com es comenta a la introducció d'aquest punt, tenim dos vistes diferents del model MVC. Per obtenir un comportament adequat i una bona reutilització del codi s'implementa el patró observador. En aquest patró de disseny hi ha una classe observable i altres que es poden subscriure als seus esdeveniments. D'aquesta manera, si s'inicia l'aplicació amb GUI, aquesta es subscriurà als esdeveniments del controlador. Per altra banda, si s'inicialitza amb la CLI, serà aquesta qui es subscriurà.

Per implementar aquest patró es fa servir la interfície *PropertyChangeListener* com a subjecte observador i la classe *PropertyChangeSupport* com a atribut que emmagatzema variables a observar tal i com es pot observar a la [II·l·lustració 23. Patró Observador](#). Addicionalment, es fa servir un enumerat per definir els diferents estats de l'aplicació i un *String* de missatge d'error. Si algun d'aquests dos es canvia a la capa controlador, es notificarà a les capes de la vista i actuaran en conseqüència.

II·l·lustració 23. Patró Observador



7.2 Patró singleton

Hi ha diverses classes del projecte que només poden estar instanciades un cop, ja que molta informació rellevant es troba emmagatzemada en aquesta. Per tant, per evitar que es pugui tenir més d'una instància per causa d'un error humà, s'implementa el patró *singleton* en totes aquelles que en qualsevol situació només n'hi pot haver una.

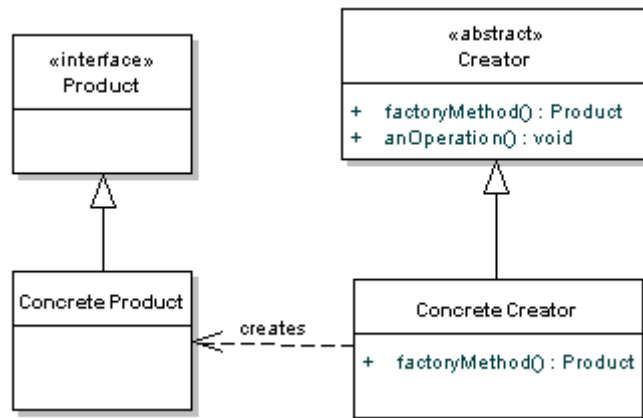
Per implementar aquest patró és tant senzill com guardar una instància de la pròpia classe com a atribut dintre de la classe, tenir el constructor com a privat, i un mètode estàtic i públic que ens la retorni. D'aquesta manera, cada cop que vulguem fer-la servir de la classe en qüestió, al demanar la seva instància, comprovarà si aquesta ja ha estat instanciada. En cas que no ho estigui crearà una nova instància de la classe i la guardarà a l'atribut comentat anteriorment. Si la instància ja està creada, simplement la retornarà.

Alguns exemples de classes del projecte que fan servir aquest patró són la classe Controller, CLIController, DatoFactory, ProductFactory, CommsManager, etc.

7.2 Patró factory

Un altre patró de disseny dels més coneguts és el del *factory*. En aquest patró es busca la especialització d'una classe en generar subclasses. En el cas del projecte s'utilitza en dos parts claus: la creació de *Products* i la de *Datos*. Quan un usuari inicia l'aplicació no sabem quin producte connectarà ni quins paràmetres suportarà aquest. Per tant, l'encarregat de generar aquestes dades seran les seves respectives factories amb un mètode on se li enviaran dades rellevants del dispositiu i aquestes s'encarregaran que les classes generades siguin les adequades.

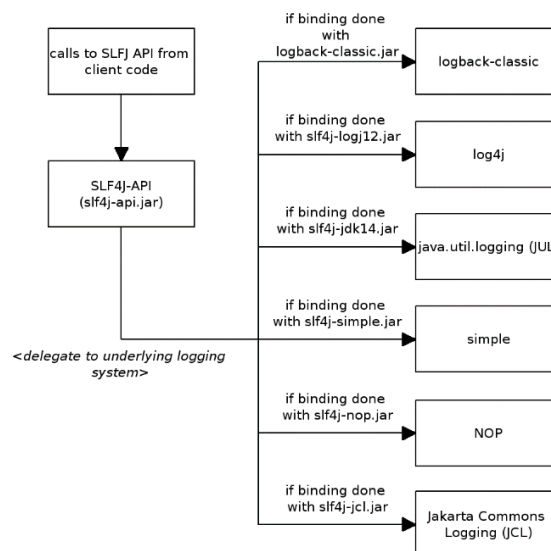
II·lustració 24. Model de domini factoria



7.3 Patró Facade

El patró *Facade* és un patró estructural pensat per evitar l'alt encapsulament de classes i la simplificació de la interacció amb un subsistema donant una capa externa que interactua amb les capes inferiors. En el cas del projecte, un dels clars exemples on es pot apreciar aquest patró és a les llibreries externes de *logging* fetes servir. Java té diverses opcions en quant a llibreries per generar *logs*, les quals tenen funcionalitats molt similars i completes. Per tal d'evitar estar lligat a una llibreria externa i poder triar en qualsevol moment quina fer servir, es desenvolupa amb *slf4j*, *Simple Logging Facade for java* [13]. Aquesta llibreria ofereix una capa externa per comunicar-se amb una de les llibreries com podrien ser *Log4j2*, *java.util.logging* o *logback*.

II·lustració 25. Diagrama SLF4J



7.3 Distribució de l'aplicació

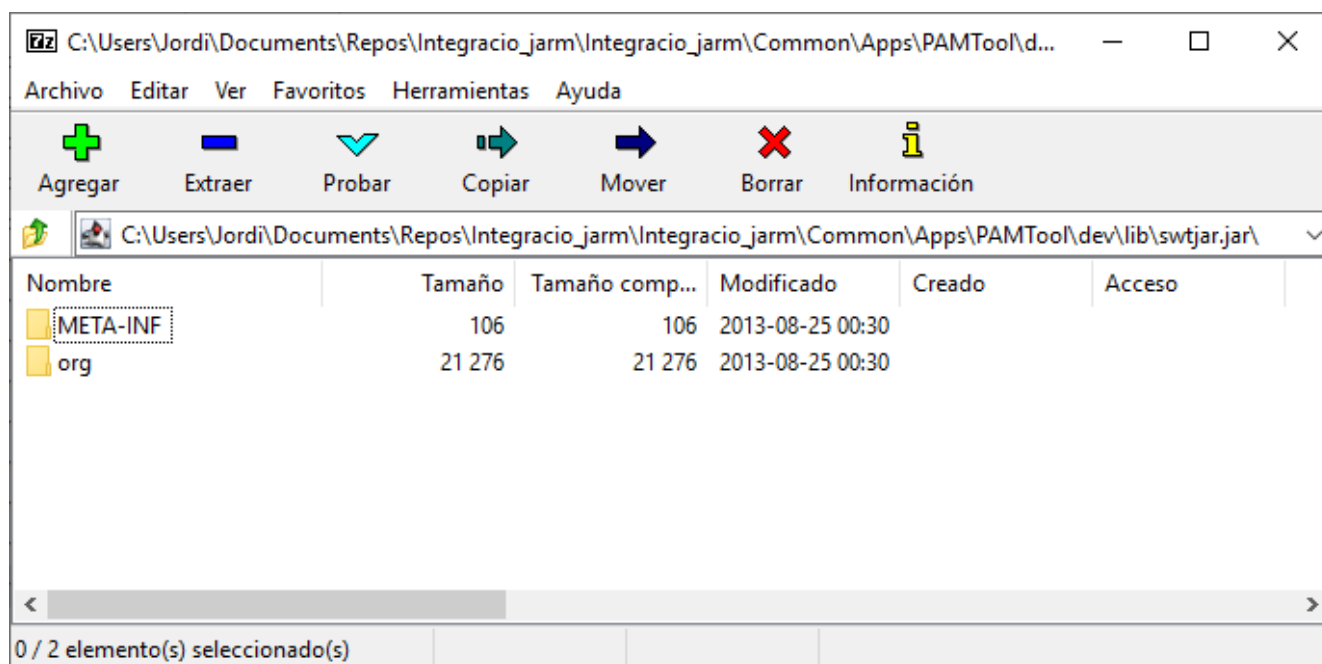
Quan es va dissenyar l'aplicació hi havia un requisit fonamental sempre en ment i consistia en què havia de ser multi-plataforma amb un sol fitxer. En el moment de prendre la decisió de migrar de Java 8 a Java > 9 es va decidir que el fet de no dependre de si la màquina on s'executaria disposaria de un JRE o de haver de distribuir-lo amb l'aplicació (>150MB) era més rellevant.

La combinació de Java 11, Maven i jlink és una combinació perfecta pel nostre objectiu. Però primer expliquem què aporta cada una d'aquestes tres tecnologies/eines al nostre objectiu i com interactuen.

7.3.1 Java 11

A partir de Java 9 apareix un nou paradigma d'organització de projectes amb Java que són els mòduls. Fins a Java 8 sempre s'havien organitzat les classes d'un projecte en paquets. Senzillament les classes que tenien relació amb aquell paquet o que es podien agrupar s'organitzaven amb una jerarquia. En el moment de creació del fitxer .jar es guardaven els diferents paquets com a primer nivell dins d'aquest.

II-lustració 26. Fitxer .jar



A la [II-lustració 26. Fitxer .jar](#) es pot veure el directori "org" que seria el nivell més alt d'un paquet de aquest jar. Aquest mètode ha acabat essent insuficient, ja que és necessari tenir més organització i modularitat a l'hora de treballar en grups de classes i les seves dependències. Per exemple, classes d'un mateix paquet podrien estar ubicades en dos .jar diferents. A més, els propis .jar de Java inclouen centenars de paquets i això és molt aproximat al concepte de monòlit.

Per donar una solució a aquests problemes recurrents, Java introdueix el projecte *Jigsaw* on es defineixen els mòduls. Són una solució molt esperada per gran part de la comunitat de Java i que altres plataformes ja feien servir, com *Node*. Un mòdul és un conjunt de classes que poden contenir un o varis paquets i que defineix les dependències amb la resta de mòduls així com la visibilitat de les classes que conté. Per definir-ho, és indispensable tenir un fitxer `module-info.java` on s'especifiqui quins *packages* es requereixen per a aquest mòdul, quins n'exporta, quins són només requerits en temps de compilació, etc.

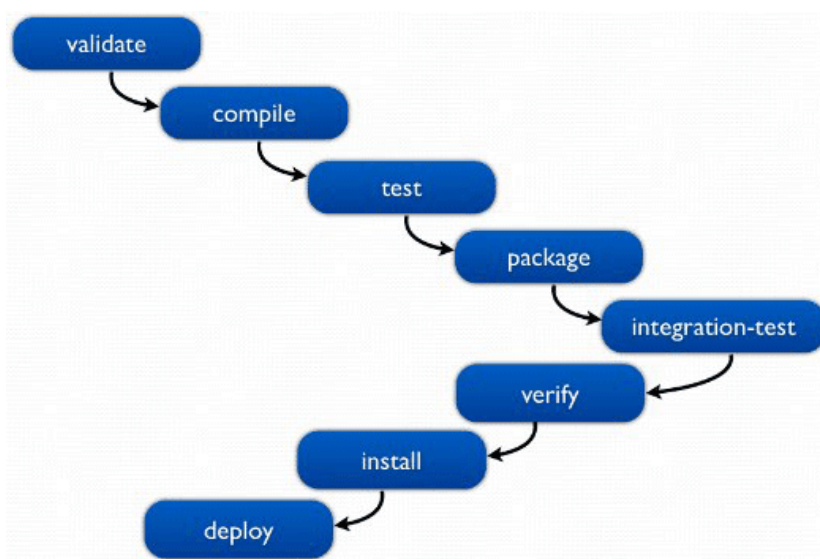
Per tal de poder fer servir els mòduls en el projecte principal implica que la resta de dependències han d'estar també gestionades per mòduls. Això suposa convertir tots els projectes que es feien servir com a llibreries a modulars. Amb els projectes interns de l'empresa és una feina relativament fàcil, ja que

són coneguts i no tenen una mida desmesurada. El problema és amb les llibreries de 3rs: si no estan mantingudes, pot suposar un camí sense sortida i haver de retrocedir. Afortunadament la gran majoria de llibreries estan mantingudes i només un petit grup d'aquestes ha donat problemes per passar-les al sistema de mòduls. Per a passar-les a modulars es fa servir l'eina "jdeps" [14] que ens genera un fitxer module-info.java amb les dependències de la llibreria. Es compila aquest fitxer i s'injecta dins del .jar i ja tenim la nostra llibreria modularitzada.

7.3.2 Maven

Maven és una eina de gestió de projectes basada en el concepte "*Project Object Model*" POM, el qual pot gestionar des del processament dels fitxers fins a la construcció del .jar passant per fases com la compilació o el test tal i com es mostra a la *il·lustració 27. Cicle de vida de Maven*.

Il·lustració 27. Cicle de vida de Maven



Els objectius principals de Maven son:

- Fer el procés de compilació i "link" fàcil.
- Proporcionar un sistema de "build" uniforme.
- Proporcionar informació rellevant del projecte.
- Proporcionar directrius de bones practiques de desenvolupament.
- Permetre una migració transparent a noves funcionalitats.

Però la funció potser més coneguda d'aquesta eina és la de gestor de dependències.

Per treballar amb Maven es necessita un fitxer pom.xml amb la configuració i dependències del projecte i a més, distribuir el projecte en diferents carpetes:

- Src/main/java: En aquesta carpeta estaran continguts tots els paquets i classes del codi de l'aplicació o llibreria.
- Src/main/resources: En aquesta carpeta estaran continguts tots aquells fitxers que no siguin classes, però siguin utilitzats en el codi de l'aplicació.
- Src/test/java: En aquesta carpeta estaran continguts tots els paquets i classes de test ja siguin unitaris o de sistema per testejar el contingut de les classes de src/main/java.
- Src/test/resources: En aquesta carpeta estaran continguts tots aquells fitxers que no siguin classes de test, però siguin utilitzats per a testos unitaris o de sistema.

7.3.3 jlink

Java es una plataforma de desenvolupament que ofereix de forma nativa moltes utilitats i llibreries per tal de facilitar la vida al usuari. Però què passa si no volem tot el paquet? Fins l'aparició de java 9 el JRE/JDK era un sol bloc enorme, però des d'aquest punt s'ha segmentat i gràcies a jlink es poden crear imatges amb només aquells components que es facin servir.

Gràcies a aquesta eina i al projecte compilat anteriorment amb Maven podem generar una imatge de la nostra aplicació que per executar-se no necessita res addicional.

8. Discussió

En aquest apartat ens centrarem en establir una relació entre els coneixements aplicats al projecte i les diferents assignatures del grau. En un projecte d'aquestes característiques es toquen moltes àrees de l'enginyeria informàtica diferents. Per tal de seguir un ordre determinat, s'explicarà de la capa més baixa fins la més alta.

Tal i com s'explica al punt 1.1 L'empresa, aquesta es dedica al sector de l'electrònica de potència. Per aquest senzill motiu els coneixements adquirits a l'assignatura d'**Electrònica** han sigut una base per entendre quina era la finalitat del projecte, entendre conceptes teòrics, etc.

L'objectiu principal del projecte era comunicar-se amb microcontroladors integrats a dispositius de l'empresa. Per aquest fet era de vital importància tenir unes nocions bàsiques de diferents protocols de comunicació. Els coneixements adquirits en aquesta àrea dintre del grau s'imparteixen en diferents assignatures, però ens centrarem en la més específica, **Programació d'arquitectures encastades**. En aquesta assignatura s'introdueix la programació i creació d'aplicacions per a microcontroladors. Entre les lliçons s'inclou l'explicació de diferents protocols de comunicació com sèrie, i2c, CAN, etc. Les lliçons sobre el protocol i sobretot les classes pràctiques al laboratori d'electrònica han aportat que pugui solucionar problemes que sense els coneixements sobre la configuració d'un oscil·loscopi, o la forma i configuració de les trames enviades no hagués sigut possible.

L'aplicació s'ha desenvolupat amb Java, un llenguatge de programació orientada a objectes (POO). Les bases sobre aquesta programació es van anar adquirint de forma estructurada en diferents assignatures com **Programació I**, **Programació II** i altres en un terme més secundari. En aquestes s'aprèn l'estructura d'un programa, algunes bones pràctiques de desenvolupament, el concepte de classes de programació, els diferents tipus de variables, entre altres. Sense una base sòlida de programació, el dur a terme un projecte d'aquesta envergadura i sobretot, mantenir-lo seria una tasca molt més complexa i laboriosa.

Per generar una aplicació eficient en memòria i temps és important tenir uns bons coneixements sobre les estructures de dades i els seus avantatges i desavantatges. L'assignatura on s'han adquirit i aplicat coneixements sobre aquesta matèria és la que comparteix el nom amb el seu àmbit, **Estructura de dades**. En aquesta assignatura es van aprendre diferents estructures de dades com cues, cues amb prioritat, *heap*, arbres, *hashmaps*, etc. En un projecte on hi ha tantes dades i relacions entre entitats, ha sigut un dels aspectes on s'ha millorat l'eficiència, gràcies al fet de conèixer quina és l'estructura indicada per a una estructura que tant sols es construeix un cop, però s'hi fa molts accessos o a l'inrevés, que es modifica diversos cops, etc.

Un cop es té una base sòlida sobre conceptes de programació i estructures de dades disponibles ja es pot deixar d'escriure codi sense control, i es pot entrar en una nova etapa on es dissenya abans de desenvolupar. En **Disseny de Software** s'explica en detall la importància de cada fase del desenvolupament d'un projecte i el cost que pot tenir un error en cada fase. Per una banda, s'introdueix el concepte de patrons de disseny i els més importants i útils d'aquests. Per altra banda, es posa molt èmfasi en la correcta realització de l'anàlisi dels casos d'ús textuais, la realització del model de domini, el diagrama de classes i els diagrames de seqüència. L'ús correcte d'aquestes eines suposa que el programa desenvolupat és molt més robust, més fàcil de mantenir, més ampliable, etc.

Com a frontera entre l'usuari i el programa sempre hi ha d'haver una interfície, ja sigui gràfica o per línia de comandes. La interacció entre usuari i màquina i com aquesta última presenta les dades al usuari pot fer que dues aplicacions amb les mateixes funcionalitats siguin completament diferents a nivell de satisfacció i facilitat d'ús. A l'assignatura de **Factors humans i computació** s'introdueixen diferents regles o normes recomanables a l'hora de desenvolupar una interfície per tal de fer-la més atractiva i fàcil d'utilitzar. Al fi i al cap el primer que veu un usuari d'un programa amb interacció és la seva interfície i si aquesta genera un impacte negatiu, és possible que l'aplicació caigui en desús.

Tot tipus de projecte és susceptible a la corrupció de dades o senzillament, per tal de poder treballar en equip sobre un mateix projecte és vital fer servir una eina de control de versions. En aquest cas les assignatures que han aportat més a nivell de estructuració de tasques i sobretot, la introducció a git han sigut **Projecte integrat de software** i **Enginyeria del software**. Tot i que l'aplicació ha sigut pràcticament desenvolupada de forma individual, per tal de poder gestionar i planificar l'estat del projecte s'han generat tasques a partir de *User stories*. Per altra banda, la introducció i posterior aprofundiment a git [15] es va impartir en les dues assignatures citades, una habilitat fonamental per al desenvolupament de software.

9. Conclusions i futures ampliacions

L'objectiu principal d'aquest projecte era el desenvolupament d'una aplicació per donar suport als dispositius de l'empresa Premium SA. Dintre d'aquest objectiu n'hi havia més a nivell de funcionalitats. L'objectiu principal s'ha assolit, ja que l'aplicació és completament operativa i estable, es pot trobar actualment dintre de la mateixa web de l'empresa quan cerques informació d'un producte estàndard. Tot i això, els diferents objectius a nivell de funcionalitats a implementar no s'han aconseguit assolir, ja que la càrrega de temps era molt superior al assumible en un treball de final de grau i per tant, es va haver de prioritzar altres tasques.

La planificació en un projecte d'envergadura reduïda com aquest pot patir grans canvis amb els alt i baixos de la tecnologia. Sobretot si ens centrem en l'àrea de desenvolupament de software, és molt perillós dependre de tecnologies o llibreries de tercers, ja que si per algun motiu decideixen abandonar el projecte o realitzar un canvi de llicència podrien generar una variació a les hores estimades.

En aquest projecte ha quedat pendent el desenvolupament de funcionalitats com que es pugui mostrar una gràfica en temps real d'alguna de les dades del dispositiu, o detectar la màxima velocitat d'actualització de dades. A més, una de les futures línies de treball d'aquesta aplicació podria ser el migrar-la a dispositius mòbils. Al estar desenvolupada amb Java, la migració a dispositius *Android* no seria molt complexa i majoritàriament s'hauria de redissenyar la capa Vista del MVC, però la resta del codi seria molt reutilitzable.

Bibliografia

- [1] Premium SA, «Compañía: Los mercados,» Premium SA, [En línea]. Available: http://www.premium.es/web/es/premium_mercados.php. [Últim accés: 18 Junio 2019].
- [2] Oracle, «Java y tu,» Oracle, [En línea]. Available: <https://www.java.com/>. [Último acceso: 18 Junio 2019].
- [3] Eclipse Foundation, Inc, «SWT: The Standard Widget Toolkit,» [En línea]. Available: <https://www.eclipse.org/swt/>. [Últim accés: 18 Juny 2019].
- [4] Oracle, «Swing APIs and Developer Guides,» [En línea]. Available: <https://docs.oracle.com/javase/8/docs/technotes/guides/swing/index.html>. [Últim accés: 18 Juny 2019].
- [5] W. Hedgecock, «Fazecast/jSerialComm Wiki,» 6 May 2019. [En línea]. Available: <https://github.com/Fazecast/jSerialComm/wiki>. [Últim accés: 18 Juny 2019].
- [6] J. B. Estrada, «Microcontroladors i Sistemes Empotrats,» [En línea]. Available: https://campusvirtual2.ub.edu/pluginfile.php/6294729/mod_resource/content/4/MiSE%2002-7%20Comunicacions%20Serie%20BUS-CAN.pdf. [Últim accés: 18 Juny 2019].
- [7] Devart, «Free time tracking software & app,» Devart, 2019. [En línea]. Available: <https://tmetric.com/>. [Últim accés: 20 06 2019].
- [8] T. a. s. foundation, «Apache Ant,» 08 Maig 2019. [En línea]. Available: <https://ant.apache.org/>. [Últim accés: 20 Juny 2019].
- [9] Oracle, «OpenJDK,» 2019. [En línea]. Available: <https://openjdk.java.net/>. [Últim accés: 20 Junio 2019].
- [10] PaulDeitel, «Understanding Java 9 Modules,» Oracle, Octubre 2017. [En línea]. Available: <https://www.oracle.com/corporate/features/understanding-java-9-modules.html>. [Últim accés: 20 Juny 2019].
- [11] Oracle, «jlink,» Oracle, [En línea]. Available: <https://docs.oracle.com/javase/9/tools/jlink.htm#JSWOR-GUID-CECAC52B-CFEE-46CB-8166-F17A8E9280E9>. [Últim accés: 20 Juny 2019].
- [12] Federació d'Indústria de CCOO de Catalunya, «04_taulas_salariats_2019.pdf,» 1 Gener 2019. [En línea]. Available: https://www.ccoo.cat/industria/convenis/metall_bcn/01_documents_vigents/04_taulas_salariats_2019.pdf. [Últim accés: 18 Juny 2019].
- [13] SLF4J, «SLF4J,» 2019. [En línea]. Available: <https://www.slf4j.org/>. [Últim accés: 21 Juny 2019].

- [14] Oracle, «jdeps,» Oracle, 2019. [En línia]. Available: <https://docs.oracle.com/javase/8/docs/technotes/tools/unix/jdeps.html>. [Últim accés: 21 Juny 2019].
- [15] Software Freedom Conservancy, «Git,» 2019. [En línia]. Available: <https://git-scm.com/>. [Últim accés: 25 Juny 2019].