

Escola Universitària Politécnica de Mataró

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA**

Grau en Enginyeria Informàtica

VISOR, EDITOR I ANALITZADOR D'ARXIUS SVC

Memòria

**ROGER BERTRAN CAPDEVILA
PONENT: Dr. ENRIC SESA NOGUERAS**

PRIMAVERA 2015



**TecnoCampus
Mataró-Maresme**

Dedicatòria

Aquest treball va dedicat a totes aquelles persones que en algun moment han influït de manera positiva en la meva vida tant professional, com estudiantil, com humana.

Gràcies a tots.

Agraïments

Aquest treball no hauria estat possible sense la col·laboració de moltes persones. En especial agrair l'ajuda i l'esforç al meu tutor el Dr. Enric Sesa Nogueras i al responsable del GTS. També donar les gràcies a l'Andrés, al Bernat, al Jose, a la Sara, al Sergi i als Joseps per la seva col·laboració. Gràcies.

Resum

Aquest projecte té com a objectiu principal el desenvolupament d'una aplicació capaç de llegir, mostrar, editar i analitzar qualsevol arxiu SVC. Amb aquesta aplicació es pretén donar una nova eina per a què metges i investigadors puguin extreure'n resultats. Aprofitant tots els coneixements adquirits al grau, l'aplicació serà basada en Java. D'aquesta manera s'aprofita l'oportunitat de seguir augmentant els coneixements sobre aquest llenguatge i la seva estructura.

Resumen

Este proyecto tiene como objetivo principal el desarrollo de una aplicación capaz de leer, mostrar, editar y analizar cualquier archivo SVC. Con esta aplicación se pretende dar una nueva herramienta para que médicos e investigadores puedan extraer resultados. Aprovechando todos los conocimientos adquiridos en el grado, la aplicación será basada en Java. De esta manera se aprovecha la oportunidad de seguir aumentando los conocimientos sobre este lenguaje y su estructura.

Abstract

The main purpose of this project is the development of an application used to read, display, edit and analyse any SVC file. This application aims to provide a new tool that doctors and researchers can use to extract results. Taking advantage of the knowledge acquired in the degree, the application will be based on Java. Thus the opportunity is used to continue increasing our knowledge of this language and its structure.

Índex

Índex de figures.....	V
Índex de taules	IX
Glossari d'acrònims	XI
1. Objectius.....	1
1.1. Propòsit	1
1.2. Finalitat	1
1.3. Objecte	1
1.4. Abast	1
2. Introducció.....	3
2.1. Visió general de l'aplicació.....	3
2.1.1. L'aplicació.....	3
2.1.2. Els arxius.....	4
2.2. Especificacions tècniques.....	8
2.2.1. Usuari.....	8
2.2.2. Desenvolupador.....	8
2.3. Planificació.....	10
3. Context	11
3.1. Context Històric.....	11
3.2. Context del projecte	12
4. Anàlisi	13
4.1. Representacions gràfiques o dibuixos.....	13
4.2. Detall dels dibuixos	14
4.3. Reproducció dels dibuixos	14
4.4. Mesures dels dibuixos.....	15

4.5.	Modificació dels arxius SVC	15
4.6.	Exportació d'informació	16
4.7.	Persistència de paràmetres	16
5.	Disseny	19
5.1.	Lectura i escriptura de fitxers	19
5.2.	Classe svcData, arxiu SVC en temps d'execució.....	20
5.3.	Estructura de Finestres.....	21
5.4.	Representacions Gràfiques o dibuixos	23
5.5.	Detall dels dibuixos	27
5.6.	Reproducció dels dibuixos.....	31
5.7.	Mesures dels dibuixos	34
5.7.1.	Mesures Simples	35
5.7.2.	Mesures Avançades.....	40
5.7.3.	Gràfiques.....	45
5.8.	Correcció d'errors en els dibuixos	47
5.9.	Exportació d'informació	49
5.10.	Persistència de paràmetres	53
6.	Implementació	55
6.1.	Llibreries.....	55
7.	Conclusions	57
8.	Possibles Ampliacions	59
9.	Referències	61
Annex A.	Estudi Econòmic	63
A.1.	Cost del prototip.....	63
A.1.1.	Costos de recursos humans	63
A.1.2.	Amortització equips i software	64

A.1.3. Despeses indirectes	65
A.1.4. Cost total del projecte	65
A.2 Preu de venda en el mercat.....	66
Annex B. Diagrames	67
B.1. Diagrama de Classes de la capa Application.....	67
B.2. Diagrama de Classes de la capa Domain	68
B.3. Diagrama de Classes de la capa Persistence	68
B.4. Diagrama de Classes de la capa Presentation.....	69
B.4.1. Diagrama de Classes de la sub-capa Detail.....	70
B.4.2 Diagrama de Classes de la sub-capa Drawing.....	71
Annex C. Càlcul de rangs de valors.....	73
C.1. Rang de valors de la pressió	74
C.2. Rang de valors de l'angle altitud	77
C.3. Rang de valors de l'angle azimuth.....	79
C.4. Distància màxima de cobertura del bolígraf.....	82
C.5. Rang de valors de l'estat del bolígraf.....	83
Annex D. Contingut del DVD	85

Índex de figures

Fig. 2.1. Mostra d'eina digitalitzadora.....	4
Fig. 2.2. Exemple d'arxiu SVC.....	5
Fig. 2.3. Representació de l'angle Azimut.....	6
Fig. 2.4. Representació de l'angle Altitud.....	7
Fig. 2.5. Diagrama de Gantt de la Planificació.....	10
Fig. 5.1. Diagrama de Domini.....	20
Fig. 5.2. Captura de la Finestra principal d'una mostra.....	21
Fig. 5.3. Captura de la Finestra de detall d'un dibuix concret.....	22
Fig. 5.4. Diagrama de classes parcial, herència de la classe Drawing.....	23
Fig. 5.5. Esquema dels eixos de coordenades i un dibuix en relació a un panell.....	25
Fig. 5.6. Esquema dels eixos de coordenades i un dibuix en relació a un panell, amb tractament.....	25
Fig. 5.7. Funcionalitats en la finestra de detall.....	27
Fig. 5.8. Algoritme de mida automàtica d'un dibuix.....	28
Fig. 5.9. Dibuix Escalat per factor i posicionat a la cantonada.....	29
Fig. 5.10. Dibuix Escalat per factor i posicionat al centre.....	30
Fig. 5.11. Diagrama de classes parcial, herència de la classe Drawing.....	31
Fig. 5.12. Algoritme de reproducció d'un dibuix.....	32
Fig. 5.13. Diagrama d'estats de la reproducció d'un dibuix.....	33
Fig. 5.14. Captura de la part de Mesures.....	34
Fig. 5.15. Algoritme de càlcul de temps "en-l'aire" i "en-el-paper".....	35

Fig. 5.16. Algoritme de càlcul de la velocitat mitjana “en-l’aire” i “en-el-paper”.....	36
Fig. 5.17. Algoritme de càlcul de les pressions.....	38
Fig. 5.18. Algoritme de càlcul de les longituds de les traces.....	39
Fig. 5.19. Algoritme de càlcul de les entropies.....	41
Fig. 5.20. Algoritme de càlcul de les dimensions fractals.....	43
Fig. 5.21. Algoritme de càlcul de la freqüència d’aparició d’un valor a una col·lecció.....	43
Fig. 5.22. Possibles dimensions dins el càlcul de la Dimensió Fractal.....	44
Fig. 5.23. Captura de la Gràfica de pressió en relació al temps.....	45
Fig. 5.24. Captura de la Gràfica de la velocitat en relació al temps.....	46
Fig. 5.25. Algoritme d’aplicació d’efecte mirall al dibuix.....	48
Fig. 5.26. Captura de la finestra de selecció dels camps per al Informe.....	49
Fig. 5.27. Codi per a la creació de l’informe en PDF.....	50
Fig. 5.28. Codi per a la exportació de dibuixos en format PNG.....	51
Fig. 5.29. Captura del menú per a exportar un gràfic en format PNG.....	52
Fig. 5.30. Exemple d’arxiu de configuració.....	54
Fig. 5.31. Exemple de patró Singleton.....	54
Fig. B.1. Diagrama de Classes de la capa Application.....	67
Fig. B.2. Diagrama de Classes de la capa Domain.....	68
Fig. B.3. Diagrama de Classes de la capa Persistence.....	68
Fig. B.4. Diagrama de Classes de la capa Presentation.....	69
Fig. B.5. Diagrama de Classes de la sub-capça Detail.....	70
Fig. B.6. Diagrama de Classes de la sub-capça Drawing.....	71

Fig. C.1. Dibuix de la mostra u00001s00002_hw00002.svc.....	74
Fig. C.2. Gràfic de la pressió en funció del temps de la mostra.....	74
Fig. C.3. Gràfic de la pressió en funció del temps de la mostra.....	75
Fig. C.4. Gràfic de la pressió en funció del temps de la mostra.....	76
Fig. C.5. Representació de com s'ha realitzat la mostra u00001s00002_hw00004.svc.....	77
Fig. C.6. Dibuix de la mostra u00001s00002_hw00004.svc.....	77
Fig. C.7. Gràfic de l'angle altitud en funció del temps de la mostra.....	77
Fig. C.8. Gràfic de l'angle altitud en funció del temps de la mostra.....	78
Fig. C.9. Representació de com s'ha realitzat la mostra u00001s00002_hw00005.svc.....	79
Fig. C.10. Dibuix de la mostra u00001s00002_hw00005.svc.....	80
Fig. C.11. Gràfic de l'angle azimut en funció del temps de la mostra.....	80
Fig. C.12. Dibuix de la mostra u00001s00002_hw00006.svc.....	81
Fig. C.13. Gràfic de l'angle azimut en funció del temps de la mostra.....	81
Fig. C.14. Representació de com s'ha calculat la distància.....	82
Fig. C.15. Representació de la nomenclatura dels botons.....	83

Índex de taules

Taula 2.1. Distribució de columnes d'un arxiu SVC.....	4
Taula 2.2. Rang de valors de la columna Estat.....	6
Taula 4.1. Quadre resum dels requeriments de l'aplicació.....	17
Taula A.1. Preus dels recursos humans utilitzats.....	63
Taula A.2. Preus del Software i els equips utilitzats.....	64
Taula A.3. Totals de les Amortitzacions.....	64
Taula A.4. Despeses directes del projecte.....	65
Taula A.5. Costos totals del projecte.....	65
Taula A.6. Preu unitari del software.....	66
Taula C.1. Pressions de la mostra u00001s00002_hw00002.svc.....	74
Taula C.2. Pressions de la mostra u00004s00001_hw00002.svc.....	75
Taula C.3. Pressions de la mostra u00001s00001_hw00003.svc.....	76
Taula C.4. Angles de la mostra u00001s00002_hw00004.svc.....	77
Taula C.5. Angles de la mostra u00001s00002_hw00005.svc.....	80
Taula C.6. Angles de la mostra u00001s00002_hw00006.svc.....	81
Taula C.7. Possibles valors d'estat i la seva causa.....	84

Glossari d'acrònims

API	Application Programming Interface
CASE	Computer-Aided Software Engineering
ESUPT	Escola Superior Politècnica del TecnoCampus
EUPMT	Escola Universitària Politècnica de Mataro
FAC	Face Authentication Competiton
FIFO	First In, First Out
FVC	Fingerprint Verification Competition
GRC	Consolidated Research Group
GTS	Grup de Tractament del Senyal
PDF	Portable Document Format
PNG	Portable Network Graphics
RAM	Random-Access Memory
SVC	Signature Verification Competition
UPC	Universitat Politècnica de Catalunya

1. Objectius

1.1. Propòsit

Analitzar, dissenyar i implementar una aplicació capaç de llegir, visualitzar, editar i analitzar fitxers SVC.

1.2. Finalitat

Donar nous recursos i nous procediments per a agilitzar la feina dels professionals que, actualment, han de realitzar les anàlisis de manera manual i que no tenen cap eina que els ajudi, cosa que automàticament provoca uns resultats més subjectius.

1.3. Objecte

Creació d'una aplicació que implementi les funcionalitats que els professionals troben necessàries per poder treballar de manera còmoda i eficaç extraient el màxim de dades útils dels arxius SVC.

1.4. Abast

Desenvolupament d'una aplicació senzilla, usable i amigable, per tal que la feina del professional sigui el més àgil possible.

Implementar totes les funcionalitats més crítiques, a fi de poder extreure'n conclusions interessants a partir dels resultats que presenta l'aplicació.

Ha de ser un software fàcil de mantenir, ampliable, estructurat i optimitzat, perquè en possibles futures fases del projecte, aquest pugui ser ampliat i millorat.

No forma part d'aquest projecte l'obtenció de mostres manuscrites ni dels arxius que les contenen.

2. Introducció

L'objectiu d'aquest projecte és crear una aplicació que faciliti el treball amb arxius SVC, des de la visió fins a l'anàlisi, passant per l'edició. Una possible aplicació d'aquesta eina seria per al diagnòstic de malalties neurodegeneratives o trastorns en l'escriptura tant a metges com a investigadors.

Mitjançant aquest projecte, es vol donar totes les eines necessàries per a poder extreure'n informació significativa de l'escriptura o de qualsevol manuscrit que pugui fer l'usuari. És per aquest motiu que s'ha necessitat la participació de tercers per tal d'identificar i concretar els requeriments de l'aplicació.

A partir de la llista de requeriments que s'ha anat identificant, el que es vol és, donades les mostres d'arxius SVC, poder-les visualitzar, editar i analitzar mitjançant certes mesures.

2.1. Visió general de l'aplicació

En aquest apartat es descriurà de manera global dos dels factors més importants del projecte. Per una banda hi ha l'aplicació que es desenvoluparà i per altra banda hi ha els arxius que s'utilitzaran i que seran l'origen de totes les dades.

2.1.1. L'aplicació

L'aplicació està enfocada en una primera fase, a donar totes les eines necessàries per poder llegir i interpretar els arxius SVC i en una segona fase a poder-los analitzar i extreure'n informació i conclusions ben fonamentades.

Dins l'aplicació, les funcionalitats més importants seran:

- Representació gràfica dels arxius SVC
- Detall de les representacions gràfiques
- Reproducció de les representacions gràfiques
- Mesures de les dades dels arxius
- Modificació dels arxius
- Exportació d'informació de l'aplicació

2.1.2. Els arxius

Els arxius SVC són arxius de text que recullen dades provinents d'una eina de digitalització com, per exemple, una tauleta. Més concretament, aquestes dades sempre seran una seqüència de punts, els quals formen el dibuix digitalitzat.



Fig. 2.1. Mostra d'eina digitalitzadora.

Aquesta seqüència de punts, no només serà la que genera el bolígraf quan està en contacte amb l'eina digitalitzadora (traça “en-el-paper”), sinó també la que genera quan el bolígraf es mou a una distància menor a 1,5 cm de la tauleta (traça “en-l'aire”). D'aquesta manera s'aconsegueix registrar la totalitat dels moviments del bolígraf des que es comença a dibuixar i s'acosta a la tauleta fins que s'acaba i se separa.

L'estructura dels arxius SVC sempre comença amb una primera línia amb un sol atribut, que és el recompte de línies que conté el fitxer (sense incloure aquesta primera). Seguidament cada línia conte un punt i les seves característiques amb la següent estructura.

Posició de l'atribut	0	1	2	3	4	5	6
Nom de l'atribut	X	Y	TimeStamp	Estat	Angle Azimut	Angle Altitud	Pressió

Taula 2.1. Distribució de columnes d'un arxiu SVC

Així doncs un arxiu SVC tindrà la següent forma:

```
419
14191 20455 77885702 2 2880 870 0
14196 20455 77885710 2 2880 870 0
14196 20455 77885717 2 2880 870 0
14197 20455 77885725 2 2880 870 0
14202 20455 77885732 2 2880 870 0
.
.
.
25635 18458 77888211 3 3200 670 496
25682 18712 77888219 3 3200 670 494
25737 18949 77888226 3 3200 670 497
25805 19169 77888234 3 3190 680 494
25884 19372 77888241 3 3190 680 495
25976 19552 77888249 3 3190 680 495
29366 16548 77888827 3 3100 660 248
29366 16548 77888835 2 3100 660 0
29588 17106 77888842 2 3100 670 0
29669 17440 77888850 2 3090 680 0
29726 17786 77888858 2 3110 710 0
```

Fig. 2.2. Exemple d'arxiu SVC

- **Descripcions dels atributs**

- X i Y

Són les coordenades del píxel que s'ha dibuixat. Si se simulés que estem en un espai de coordenades cartesianes, serien les coordenades del punt concret en relació a l'origen de coordenades que seria la cantonada inferior esquerra.

- Timestamp

És una variable que ens dóna un valor temporal relatiu (no absolut) de quan s'ha dibuixat el punt concret. És un valor relatiu al punt d'inici de gravació. En la tauleta utilitzada en el projecte, l'interval és de 10 mili-segons tot i que no és massa exacte.

- Estat

Aquest atribut és l'encarregat de definir l'estat de la tauleta i el bolígraf en el punt concret. Defineix si el bolígraf i la tauleta estan en contacte o no, així com si el bolígraf té algun botó

premut o no i si el bolígraf està posat del dret o del revés. Els valors que pot agafar són els següents:

Estat del Bolígraf	Valor columna Estat
Dibuixant “en-el-paper” amb la punta sense cap botó	1
Dibuixant “en-l’aire” amb la punta sense cap botó	0
Dibuixant “en-el-paper” amb la punta i prement el botó B	3
Dibuixant “en-l’aire” amb la punta i prement el botó B	2
Dibuixant “en-el-paper” amb la punta i prement el botó A	5
Dibuixant “en-l’aire” amb la punta i prement el botó A	4
Dibuixant “en-el-paper” amb la punta i prement els dos botons	7
Dibuixant “en-l’aire” amb la punta i prement els dos botons	6
Dibuixant “en-el-paper” amb la goma sense cap botó	1
Dibuixant “en-l’aire” amb la goma sense cap botó	0
Dibuixant “en-el-paper” amb la goma i prement el botó B	3
Dibuixant “en-l’aire” amb la goma i prement el botó B	2
Dibuixant “en-el-paper” amb la goma i prement el botó A	5
Dibuixant “en-l’aire” amb la goma i prement el botó A	4
Dibuixant “en-el-paper” amb la goma i prement els dos botons	7
Dibuixant “en-l’aire” amb la goma i prement els dos botons	6

Taula 2.2. Rang de valors de la columna Estat.

Per veure el calibratge d'aquests valors, veure annex.

○ Angle Azimut

És l'angle pla relatiu a l'orientació del bolígraf amb la tauleta. Es pot observar de manera més clara en el següent esquema:

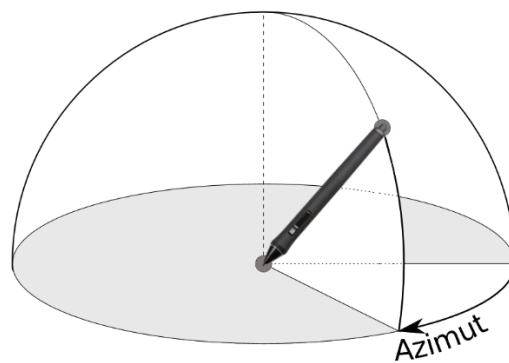


Fig. 2.3. Representació de l'angle Azimut

- Angle Altitud

És l'angle relatiu a l'altitud del bolígraf amb la tauleta. Es pot observar de manera més clara en el següent esquema:

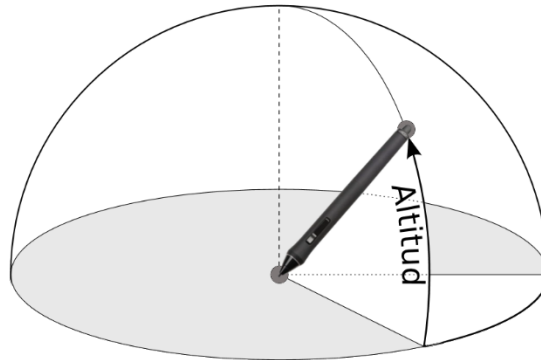


Fig. 2.4. Representació de l'angle Altitud

- Pressió

Dóna el valor de la pressió a què l'usuari sotmet el bolígraf sobre la tauleta. Com que no té una magnitud fixa, ja que cada tauleta té els seus propis límits, s'haurà d'especificar el valor màxim dins l'aplicació.

2.2. Especificacions tècniques

Per al correcte funcionament del software, en aquest apartat es detallen les especificacions utilitzades, tant en l'àmbit de Hardware, com de software.

2.2.1. Usuari

- Hardware

L'usuari per poder executar l'aplicació haurà de disposar d'un ordinador. L'ordinador no haurà de ser de gamma alta, però haurà de tenir com a mínim un processador Intel i3 amb 4 nuclis a 2.53 GHz i 4 GB de memòria RAM. No es pot garantir que amb prestacions inferiors l'aplicació funcioni correctament.

- Software

L'ordinador haurà de tenir instal·lada la màquina virtual Java amb versió 8.0 mínima, per evitar possibles problemes de versions i compatibilitats.

Respecte al sistema operatiu, no hi ha cap limitació pròpia, sinó les de la màquina virtual Java, qualsevol sistema operatiu que pugui tenir la màquina virtual, podrà executar correctament l'aplicació.

2.2.2. Desenvolupador

- Hardware

S'ha utilitzat un ordinador portàtil amb un processador Intel i3 amb 4 nuclis a 2.53 GHz, 4 GB de memòria RAM i 256 GB de memòria SSD. Amb una pantalla de 15.5'' amb retro il·luminació LED.

Per poder generar arxius SVC propis i poder fer proves dins l'aplicació, s'ha utilitzat una tauleta digitalitzadora Wacom intuos 5 touch large amb el bolígraf que porta de sèrie i amb un segon bolígraf, intuos 4 Cintiq 21UX.

- Software

S'ha treballat sobre un sistema operatiu Windows 7 Ultimate de 64 bits.

Com a entorn de desenvolupament s'ha utilitzat Eclipse Modeling Tools versió Luna 4.4.1, amb diversos complements; WindowBuilder, Git Team Provider i Jasper Reports Library and Tools.

Com a control de versions, s'ha utilitzat un repositori al servidor de Bitbucket i l'aplicació de Git instal·lada a l'ordinador.

Per a realitzar diagrames, s'ha utilitzat l'eina CASE Sybase PowerDesigner versió 16.1 amb llicència Standalone local.

Per a la redacció s'ha utilitzat la suite de Microsoft, Office Professional Plus 2013. Majoritàriament les eines Word i Excel.

Per a la visualització dels arxius SVC i poder llegir-ne les dades sense interpretar-les s'ha utilitzat l'editor de text Sublime Text, tot i que es pot llegir amb qualsevol editor de text convencional.

Per a la creació dels arxius SVC propis i poder provar l'aplicació, s'ha utilitzat un software que s'ha desenvolupat dins el mateix marc de recerca. L'aplicació anomenada HandWritingCapture.

2.3. Planificació

Des de bon principi i per tal de poder treballar de manera estructurada i precisa, s'ha treballat entorn d'una planificació de les tasques i el temps disponible. Aquesta planificació s'ha anat actualitzant segons l'avanç de l'aplicació i les seves dificultats, tot i en cap moment s'ha hagut de fer cap canvi significatiu.

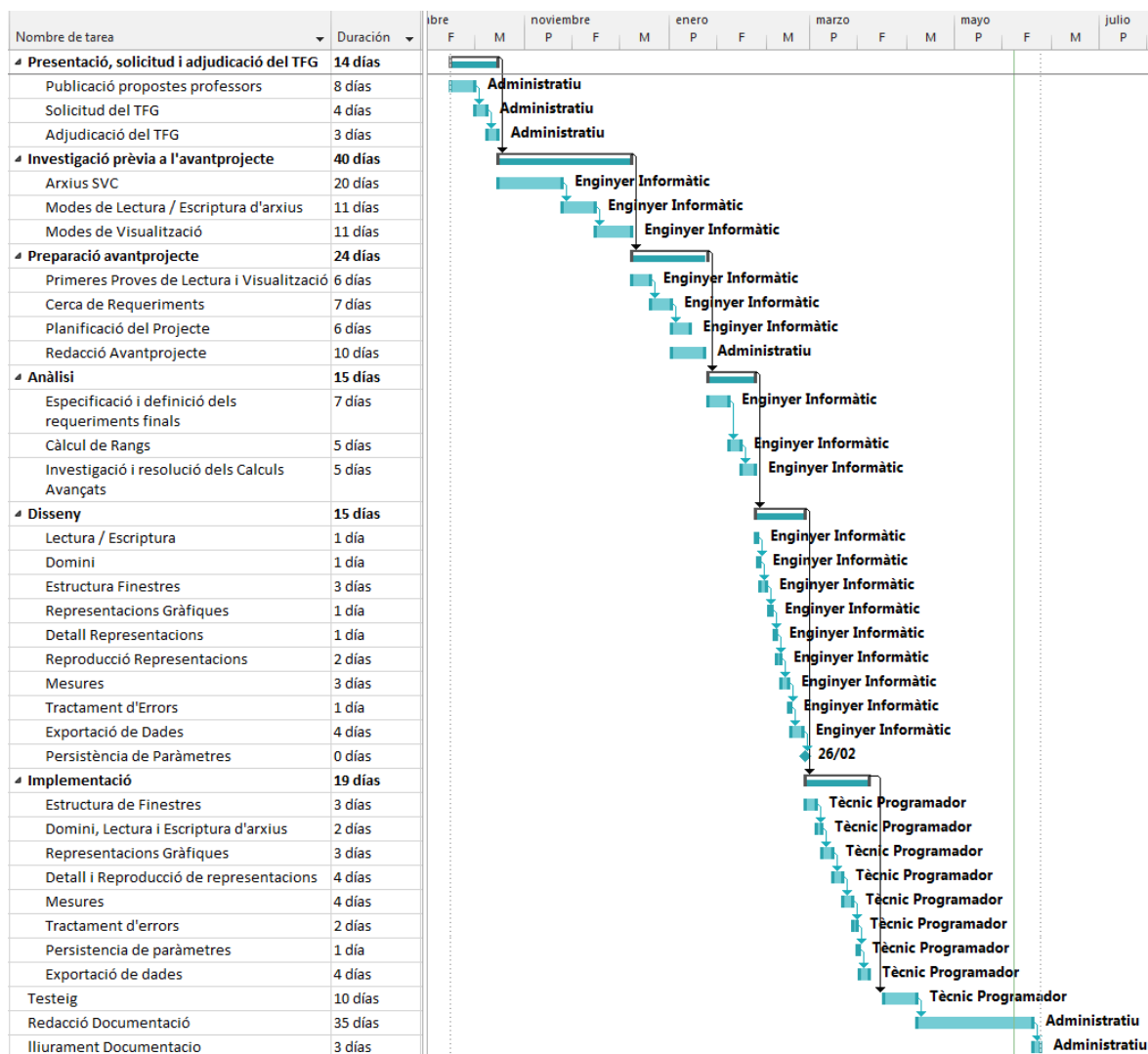


Fig. 2.5. Diagrama de Gantt de la Planificació.

3. Context

3.1. Context Històric

SVC, Signature Verification Competition, o traduït “Competició de verificació de signatures” és una competició que comença el 2004 i té com a objectiu la confirmació de la identitat d’un usuari a partir del manuscrit de la seva signatura i el comportament biomètric d’aquesta.

L’any 2004 en el sector del reconeixement biomètric de persones, hi havia tres competicions la de signatura (SVC – Signature Verification Competition), la d’empremta digital (FVC – Fingerprint Verification Competition), i la facial (FAC – Face Authentication Competition). La competició de signatures tenia un avantatge sobre les altres, ja que és el tipus d’identificació més usat i acceptat a tot el món.

Tot i aquesta acceptació i aquest ús de la signatura com a mètode de verificació, no hi havia hagut cap esforç a escala internacional per analitzar i millorar els diferents sistemes de verificació fins al 2004, que es va crear la competició per tal de posar en contacte les dues parts (investigació i indústria) i poder millorar els sistemes [9-11].

La competició va ser per equips on es combinaven les persones del món acadèmic amb les persones del món de la indústria. L’equip podia romandre en l’anonimat fins que rebien els seus resultats. Aleshores tenien l’opció de revelar la identitat.

Cada equip es podia inscriure a dues proves diferents. La diferència entre les dues competicions eren les dades que incloïen les signatures. En la primera només tenia les coordenades dels punts. En canvi la segona tenia informació addicional així com la pressió l’orientació del bolígraf, etc.

Un cop cada equip s’havia apuntat a les competicions, l’objectiu era desenvolupar un procediment el més exacte possible per tal d’aconseguir una verificació correcta el 100% de les execucions [7-10].

Les dades de cada signatura es guardaven en un fitxer de text on cada línia era un punt, seguit de tots els seus atributs. A la primera línia hi anava un recompte de totes les línies que es guardaven en el fitxer. D’aquest format d’arxiu és d’on sorgeixen els arxius SVC i la seva estructura.

3.2. Context del projecte

El projecte s'ha desenvolupat per al Grup de Tractament del Senyal (GTS), grup de recerca consolidat, reconegut per la Generalitat de Catalunya –GRC-1318-, dirigit pel Dr. Marcos Faúndez i del qual el ponent d'aquest treball, el Dr. Enric Sesa, n'és membre. El GTS es va crear l'any 1995 a l'Escola Universitària Politècnica de Mataró (EUMPT-UPC) avui Escola Superior Politècnica del TecnoCampus (ESUPT-Pompeu Fabra) per a fer recerca en l'àmbit dels senyals de veu. Posteriorment va ampliar el seu àmbit de recerca a altres camps també relacionats amb la biometria (reconeixement facial, reconeixement per signatura, reconeixement per escriptura manuscrita, imatges tèrmiques, ...). En el camp específic de l'escriptura manuscrita, el GTS ha publicat, treballs sobre el reconeixement i la verificació de persones, la generació sintètica d'escriptura, el reconeixement del gènere de l'escriptor, l'ús de la informació continguda a les traces “en-l'aire”, l'efecte de l'envelliment, etc. Darrerament, una part de la recerca del grup s'ha traslladat a qüestions relacionades amb la salut i, més concretament, amb la detecció de les malalties neurodegeneratives (Parkinson, Alzheimer, demència senil, etc.) I dins d'aquest àmbit hi trobem l'estudi de la possibilitat d'utilitzar l'escriptura manuscrita com a eina de detecció d'algunes d'aquestes malalties. Aquest estudi es desenvolupa en col·laboració amb d'altres ens de recerca, com la Universitat de Brno (República Txeca), la Universitat del País Basc, la Universitat de Vic l'Hospital de Mataró i l'Hospital Psiquiàtric de Salt.

Per tal de dur a terme la seva tasca de recerca en l'esmentat àmbit de la detecció de malalties neurodegeneratives a partir de l'anàlisi de l'escriptura manuscrita, el GTS requeria una eina que facilités les tasques de visualització, anàlisi i modificació dels arxius on resideixen les mostres d'escriptura que s'usen. Aquestes mostres s'obtenen d'orígens diversos (sovint de centres hospitalaris) i calia una eina que fos útil tant per als professionals encarregats de la seva obtenció com per als professionals i investigadors que n'han d'extreure conclusions.

4. Anàlisi

Per tal que l'aplicació desenvolupada fos el més ajustada al que les persones expertes en aquest camp necessiten, des de bon principi s'ha volgut mantenir la comunicació amb elles. L'objectiu sempre ha estat cobrir al màxim les seves expectatives, de manera que en un futur aquest projecte pogués ser usat.

Per aquest motiu, des de bon principi ja es va anar creant tota una llista de requeriments més o menys complexos, que amb l'avanç del projecte s'ha anat ampliant i concretant. La llista de requeriments està formada pels següents apartats.

4.1. Representacions gràfiques o dibuixos

La primera fita primordial és que l'aplicació ha de poder extreure les dades de qualsevol arxiu SVC. Aquestes en el seu conjunt, formen part d'una representació gràfica; Un escrit, un dibuix, etc. Dins aquest projecte a la representació gràfica l'anomenarem *Dibuix*.

Un dels requeriments més bàsics és poder mostrar aquests dibuixos en pantalla. Com ja s'ha explicat, dins les dades del fitxer SVC hi pot haver diferents tipus de traces. En aquest cas concret només es distingirà entre dos grans grups. Les traces “en-el-paper” o les traces “en-l'aire” analíticament estan diferenciades, de manera que aquesta distinció s'aplica també als Dibuixos. La traça “en-el-paper” se li ha assignat el color blau i a la “en-l'aire” el color vermell. Referent al punt anterior també s'ha decidit poder separar les traces en dibuixos diferents, per tenir-ne una visió separada o individualitzada.

4.2. Detall dels dibuixos

En aquest bloc en referència a la part dels dibuixos es vol tenir la possibilitat de compararlos, analitzar-los i modificar-los per tal de poder extreure'n més informació.

Una característica molt útil és poder modificar la mida del dibuix, ja bé de forma automàtica perquè agafi la mida de la pantalla, o bé poder utilitzar un factor concret a partir del qual poder comparar dos dibuixos diferents.

Relacionat amb l'arranjament de la mida a partir d'un factor concret, també s'ha vist oportú poder modificar la posició del dibuix dins la pantalla. Les dues posicions més interessants són la cantonada inferior esquerra (què és l'origen de coordenades) i el centre de la pantalla.

Amb aquestes dues característiques, l'usuari ja té les eines necessàries per poder mostrar dos dibuixos simultàniament i mitjançant els factors de mida i el posicionament, poder comparar de manera representativa i gràfica les dades dels dos arxius.

4.3. Reproducció dels dibuixos

Dins de cada arxiu SVC, com ja s'ha explicat, cada línia o punt té un atribut temporal. Però en les representacions gràfiques o dibuixos, aquest atribut no es pot utilitzar, i per tant només queden representats de manera bidimensional. Per aquest motiu es dóna la possibilitat de reproduir un arxiu SVC com si es tractés d'un vídeo, aprofitant d'aquesta manera l'atribut temporal.

Aquesta funcionalitat aporta molt valor degut que no només les dades en si contenen molta informació, sinó que si aquestes dades es posen relatives al temps, la informació s'enriqueix significativament. En el vídeo, l'usuari tindrà el control (play, pause/resume, stop) de la reproducció.

A partir de la visualització de diversos arxius reals, que no s'havien creat de manera correcta (cada arxiu contenia dos dibuixos seguits) s'ha vist oportú poder dividir l'arxiu en dos de diferents i poder treballar amb els dos dibuixos per separat. Un cop pausat el vídeo, es dóna l'opció d'exportar, ja sigui la part prèvia al punt on s'ha parat l'usuari, com la part posterior i crear un nou arxiu SVC amb la part que l'usuari ha decidit.

Per agilitzar la feina, l'usuari pot modificar la velocitat de reproducció, ja sigui per poder veure amb més detall una part concreta, com poder passar ràpidament una part no tan interessant.

4.4. Mesures dels dibuixos

A part de tot un seguit de requeriments gràfics, també es calculen algunes mesures concretes que poden aportar més valor afegit a la informació que es visualitza gràficament. Aquestes mesures es poden subdividir en dos grups. Les mesures simples i les mesures més avançades.

Dins les mesures simples s'inclouen el temps "en-l'aire", el temps "en-el-paper", la velocitat mitjana "en-l'aire", la velocitat mitjana "en-el-paper", la pressió mínima, màxima i mitjana i per últim les longituds totals de les traces "en-l'aire" i "en-el-paper".

El subgrup de les mesures més avançades inclou l'entropia de la pressió, l'entropia de l'angle azimut i l'entropia de l'angle altitud així com la dimensió fractal del dibuix "en-el-paper", la dimensió fractal del dibuix "en-l'aire" i la dimensió fractal del dibuix complet (traça "en-el-paper" i traça "en-l'aire" combinades).

Per últim en aquest apartat de mesures també s'ha vist interessant mostrar dues gràfiques; la pressió en funció del temps i la velocitat en funció del temps. Donant així una visió més general d'aquests dos atributs en funció de la variable temporal.

4.5. Modificació dels arxius SVC

A causa de la possibilitat que un arxiu sigui generat de manera errònia pel que fa a la posició de la tauleta, etc., es dona la funcionalitat per poder fer efecte mirall sobre els dibuixos. Podran ser en direcció vertical o horitzontal, o la combinació de les dues. D'aquesta manera el dibuix es pot corregir i després tornar a ser guardat per tal de seguir-hi treballant.

4.6. Exportació d'informació

Un cop l'aplicació ha pogut llegir un fitxer i ha mostrat tots els dibuixos, mesures, etc. una funcionalitat molt important és poder extreure'n informació fora d'aquesta per poder guardar-la, seguir-hi treballant, etc.

Per facilitar aquesta tasca, s'han plantejat diferents opcions. La primera és poder crear un document PDF on queden registrats tots els camps disponibles a l'aplicació. Donant l'opció a l'usuari de seleccionar quins camps vol o no vol mostrar.

La segona és exportar qualsevol dels dibuixos o dels gràfics en format d'imatge.

L'última és exportar una part d'un dibuix o la seva totalitat a un arxiu SVC, ja estigui modificat o no, per tal de guardar les dades en l'estat que es tenen.

4.7. Persistència de paràmetres

Per a la facilitat d'ús de l'usuari, l'aplicació pot emmagatzemar certs paràmetres, per tal de no haver-los d'introduir cada cop que s'utilitza l'aplicació.

Aquests paràmetres són:

- La carpeta per defecte on es van a buscar els arxius SVC.
- Les dimensions de la tauleta usada, en píxels.
- Les dimensions de la tauleta usada, en mil·límetres.
- El valor màxim de la pressió per a la tauleta i el bolígraf usats.

Apartat de Requeriments	Subapartat	Requeriments
Representacions Gràfiques		Dibuix “en-el-paper” (blau) Dibuix “en-l’aire” (vermell) Dibuix complet (blau+ vermell)
Detall	Posicionament	Posició del dibuix centrada amb la finestra Posició del dibuix a la cantonada del la finestra
	Escalat	Mida dibuix automàtica i relativa a la finestra Mida dibuix escalada
Reproducció de Dibuixos		Reproducció de qualsevol dels tres dibuixos anteriors Control de la reproducció Modificació de la velocitat de reproducció
Mesures		Temps “en-l’aire” Temps “en-el-paper” Velocitat mitjana “en-l’aire” Velocitat mitjana “en-el-paper” Pressió Màxima Pressió Mínima Pressió Mitjana Longitud total de la traça “en-el-paper” Longitud total de la traça “en-l’aire” Entropia de la pressió Entropia de l'angle azimut Entropia de l'angle altitud Dimensió Fractal del dibuix “en-el-paper” Dimensió Fractal del dibuix “en-l’aire” Dimensió Fractal del dibuix complet (dibuix “en-el-paper” + dibuix “en-l’aire”)
	Gràfiques	Gràfica pressió en funció del temps Gràfica velocitat en funció del temps
Exportació		Exportar Informe a PDF amb opció de seleccionar quins caps es volen exportar. Exportar Dibuixos a PNG Exportar SVC totals i parcials, modificats o no
Modificació dels arxius SVC		Flip dels Dibuixos Vertical Flip dels Dibuixos Horitzontal
Persistència de paràmetres		Carpeta per defecte a l’hora de buscar mostres Dimensions de la tauleta usada, en mil·límetres Dimensions de la tauleta usada, en píxels el valor màxim de la pressió per a la tauleta i el bolígraf usats

Taula 4.1. Quadre resum dels requeriments de l'aplicació.

5. Disseny

A partir dels requeriments acordats, i tenint en compte la prioritat o importància de cadascun, s'ha anat buscant una solució o implementació efectiva, ampliable i usable formant d'aquesta manera el software en la seva totalitat.

5.1. Lectura i escriptura de fitxers

El primer que es necessita fer un cop l'usuari ha seleccionat el fitxer que vol analitzar, és poder llegir-lo, i en cas que sigui modificat, poder guardar-lo de nou.

Per llegir un fitxer, es treballa amb la classe `BufferedReader`, amb la qual es va llegint línia a línia fins al final del fitxer. La primera línia es tracta de manera diferent, ja que és el recompte de línies, però la resta es tracten de manera idèntica.

Cada cop que es comença a llegir un fitxer, a l'hora també es crea una instància de la classe `svcData`, per tal de poder-hi anar posant les línies que es llegeixen. Cada cop que es llegeix una línia nova es comprova si es tracta d'un recompte o d'un punt. En cas que es tracti d'un recompte, es tanca la instància `svcData` que s'estava utilitzant i se'n crea una de nova. D'aquesta manera els dibuixos que comparteixen fitxer queden separats en conjunts de dades diferents.

En cas que la línia no sigui un recompte de línies, s'agafa aquesta línia (en format de text, que és com s'ha llegit) acompanyada del número de línia a la qual correspon dins el fitxer, i es passa a la instància `svcData` oberta. Aquesta ja s'encarrega de convertir la línia de text en dades numèriques separades i de guardar-les cadascuna a la seva col·lecció.

Per escriure un fitxer `SVC`, es treballa amb la classe d'escriptura equivalent a la classe de lectura que s'ha utilitzat, `BufferedWriter`. Amb aquesta classe el que es fa és a partir de la instància `svcData` que conté les dades que es volen guardar, se li va demanant d'una a una les línies que volem escriure. La instància ja s'encarrega de posar les dades en ordre i de retornar-les en format de text.

Un cop es tenen aquestes dades en format de text, d'una a una es van escrivint a l'arxiu, mitjançant la classe `BufferedWriter`.

5.2. Classe svcData, arxiu SVC en temps d'execució

La classe svcData és l'encarregada de fer de “magatzem” o de “fitxer” en temps d'execució, és a dir, serà l'encarregada de guardar totes les dades de l'arxiu en temps d'execució. D'aquesta manera queden les dades accessibles mitjançant mètodes sense haver d'anar al fitxer guardat a disc, que acostuma a ser una tasca més costosa.

svcData s'estructura de la següent manera:

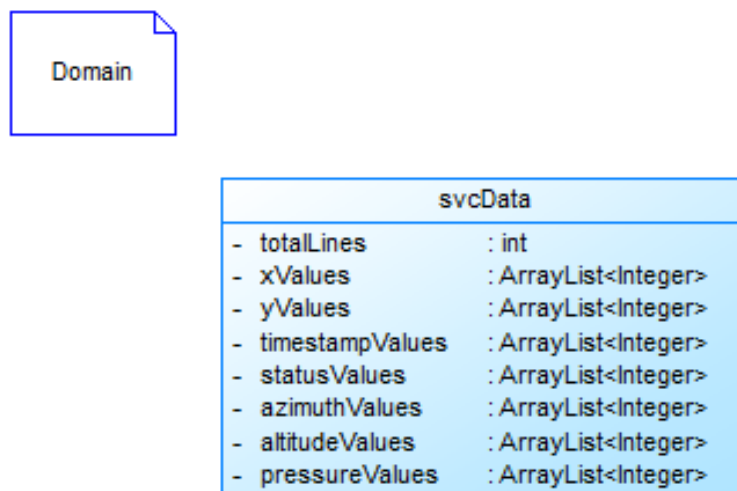


Fig. 5.1. Diagrama de Domini.

Per cada columna o atribut del fitxer SVC a la classe hi ha una col·lecció de la classe ArrayList on el valor serà l'atribut que es llegeix del fitxer. Com que el funcionament de la classe ArrayList és FIFO (First in - First out, el primer que entra és el primer que surt) [13], l'índex que s'assigna a cada valor a l'afegir-lo a la col·lecció, ja serveix per a identificar el número de línia del fitxer. També s'emmagatzema el nombre total de línies, pels casos en què cal recórrer la totalitat de les dades.

D'aquesta manera el contingut del fitxer continua ordenat, però de manera dinàmica. Facilitant l'ús de les dades dins l'aplicació. Cada fitxer que s'obri, se li crearà una instància svcData associada amb la qual es treballarà.

5.3. Estructura de Finestres

La interfície gràfica de l'aplicació s'ha pensat per facilitar la feina a l'usuari. Per aportar-li senzillesa, tota l'aplicació treballarà amb una sola finestra. Aquesta podrà tenir finestres internes dins ella per poder mostrar els diferents continguts.

Aquestes finestres internes podran ser de dos tipus. El primer tipus és la finestra que mostra la visualització general de les dades d'un arxiu. Se'n podran obrir tantes com dibuixos continguin els arxius que s'han obert. El disseny d'aquesta s'ha pensat perquè sense que l'usuari tingui necessitat de fer cap acció (excepte desplaçar la barra de desplaçament amunt i avall), es pugui tenir una visió general de tots els dibuixos, els resultats de les mesures i els gràfics.

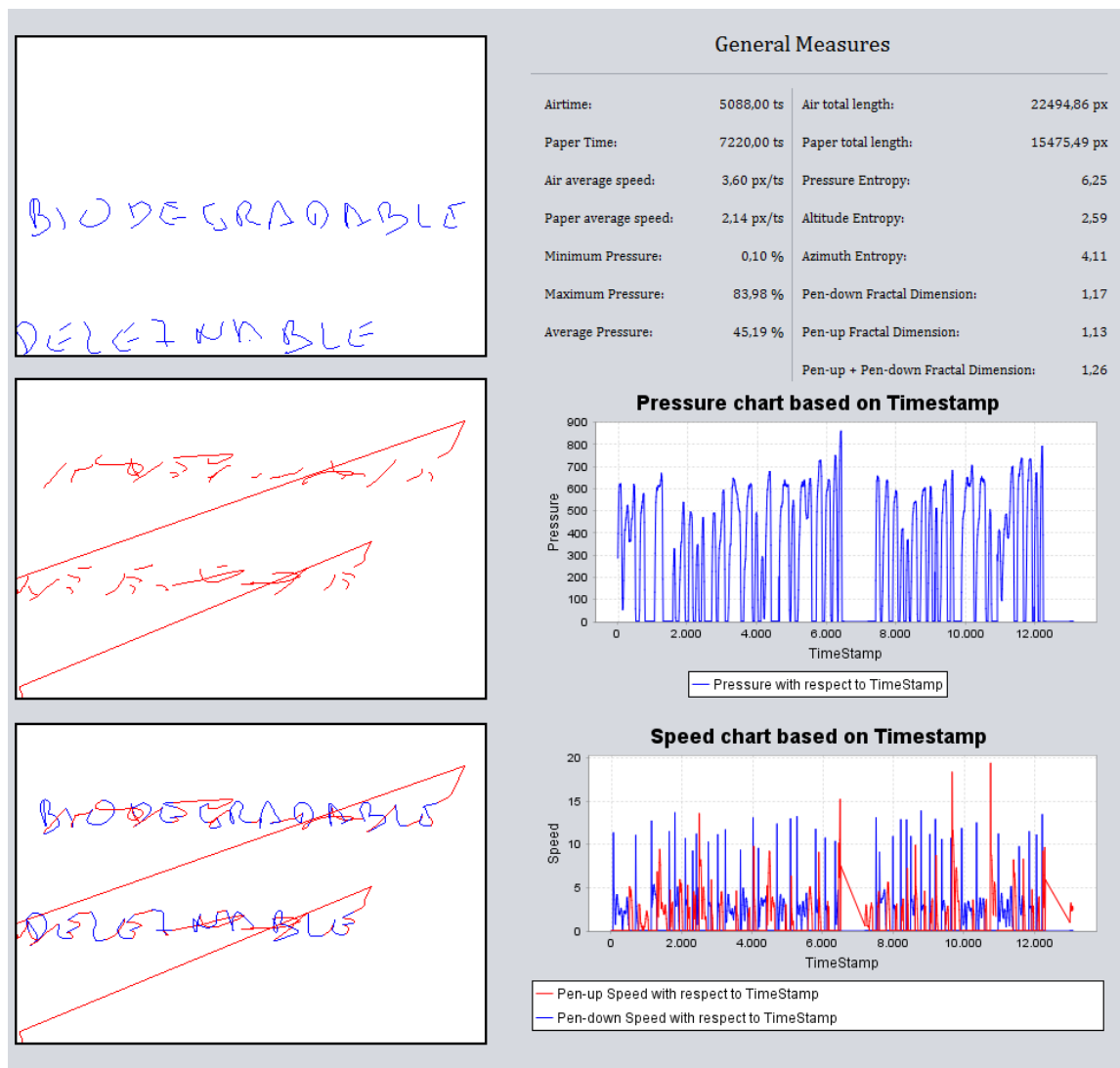


Fig. 5.2. Captura de la Finestra principal d'una mostra.

El segon tipus de finestra interna que hi pot haver és la finestra de detall, la qual dóna totes les opcions que s'han explicat a l'anàlisi per tal de poder inspeccionar de manera més profunda un dibuix concret. Consta de dues pestanyes, una per al detall del dibuix i una per a la part de la reproducció del dibuix.

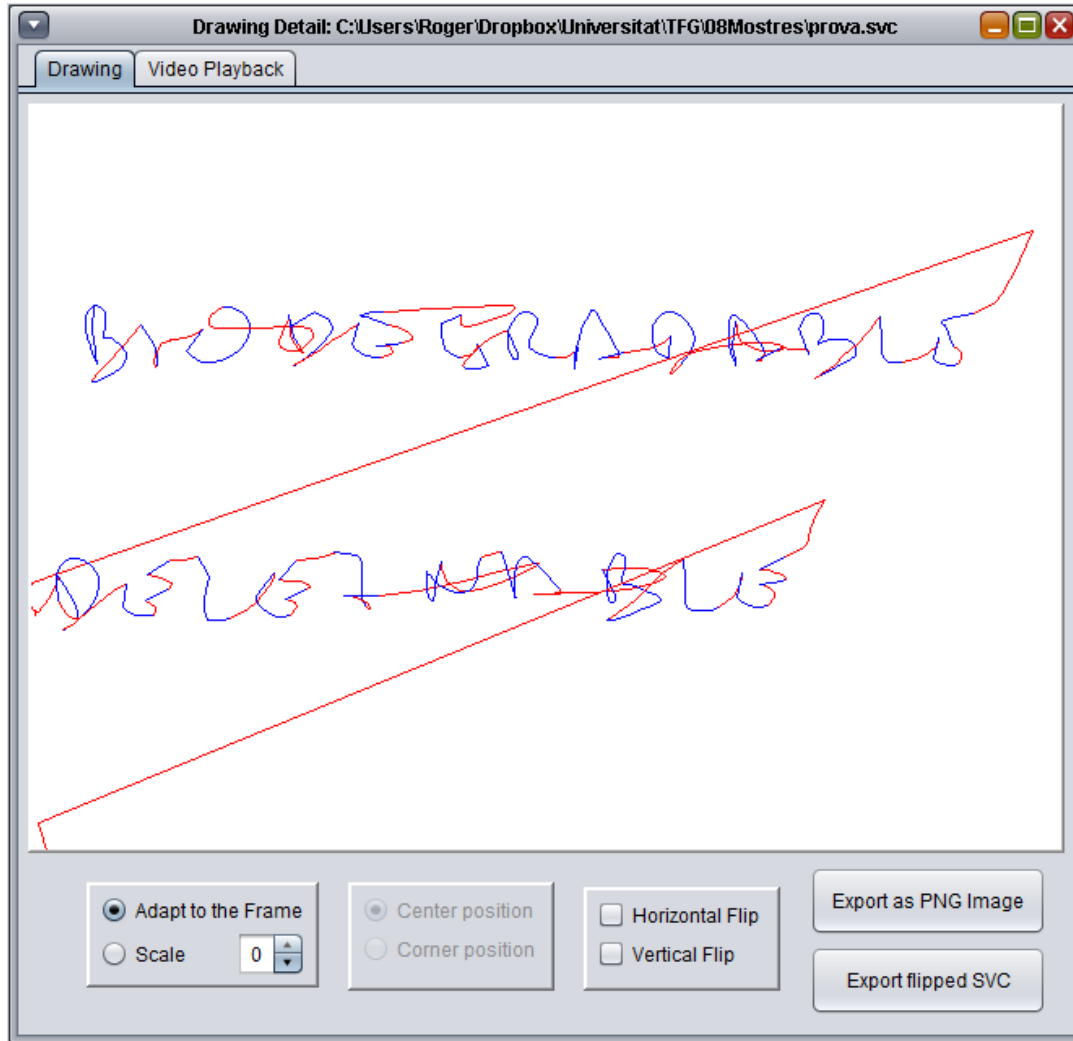


Fig. 5.3. Captura de la Finestra de detall d'un dibuix concret.

5.4. Representacions Gràfiques o dibuixos

Un cop ja s'han carregat les dades a l'objecte `svcData` en temps d'execució, ja es pot començar a representar gràficament els diferents dibuixos. Per poder-ho fer, s'ha creat la classe `Drawing`, i d'aquesta hereten els diferents tipus de dibuixos.

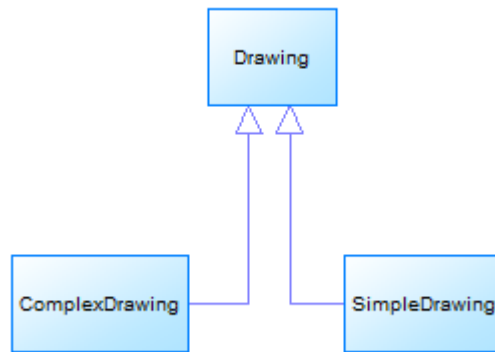


Fig. 5.4. Diagrama de classes parcial, herència de la classe `Drawing`.

La classe `Drawing` estén de `JPanel`, per tal de poder tenir tota la funcionalitat gràfica i poder mostrar els dibuixos. Hi ha diverses maneres de fer-ho. Com que aquests dibuixos no tenen una gran complexitat, ja que només són punts, el que s'ha fet és aprofitar la classe de Java `Graphics2D` [18].

`Graphics2D` és una classe que ens proporciona tota una col·lecció de mètodes per a fer representacions gràfiques en dues dimensions. Es poden fer des de formes geomètriques fins a efectes sobre una imatge passant per escriptura de text. Tot sempre de manera gràfica.

A partir d'aquesta classe, es reescriu el mètode `paintComponent(Graphics g)` des d'on es cridarà tot el codi per poder representar les dades necessàries per a formar els dibuixos, així com el mètode `paintComponent` de la super-classe. Més concretament a partir de tres mètodes:

- `create()`, per poder començar a pintar sobre el panell.
- `setColor(Color)`, amb el qual es pot anar definint el color per tal de poder diferenciar les traces “en-l'aire” de les traces “en-el-paper”.

- `drawLine(X0, Y0, X1, Y1)`, per tal de poder dibuixar els punts definits a l'arxiu. Tot i que a les dades el que hi ha definit són punts, a la realitat, la nostra escriptura és una línia. És per això que a l'hora de dibuixar, el que es fa és agafar un punt (X_0, Y_0) i el seu successor (X_1, Y_1) i dibuixar la línia entre aquests. D'aquesta manera el dibuix s'acosta més a la realitat que si dibuixem només punts.

En cas que el punt i el seu successor tinguin valors a l'atribut estat diferents, s'agafa com a origen i fi el mateix punt, ja que en aquest cas el punt representarà el final d'un traç i el següent punt serà el principi d'un altre traç.

D'aquesta manera a l'hora de crear els dibuixos el que es fa és anar demanant les dades a l'objecte `svcData` i anar-los dibuixant sobre l'objecte `Graphics2D`.

Per defecte la classe `Graphics2D` posa l'origen de coordenades a la cantonada superior esquerra del panell. Això vol dir que la part visible dins el panell és la part negativa de l'eix Y cartesià. Si les dades es passen a gràfic directament, el que passa és que no es visualitza l'arxiu, ja que la part positiva de l'eix Y no és visible.

Per solucionar aquesta problemàtica hi ha dues opcions. Una és canviar l'origen de coordenades a la cantonada inferior esquerra del panell i poder treballar normal, l'altre és simular que es canvia l'origen de coordenades mitjançant una conversió de la coordenada Y de cada punt.

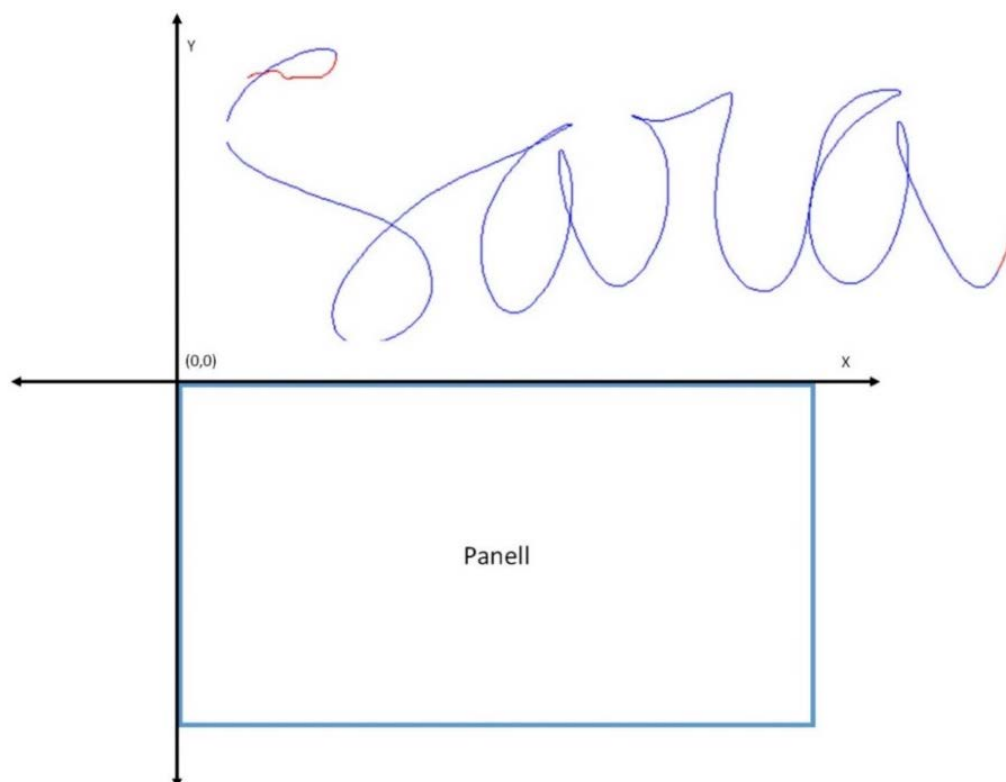


Fig. 5.5. Esquema dels eixos de coordenades i un dibuix en relació a un panell, sense tractar.

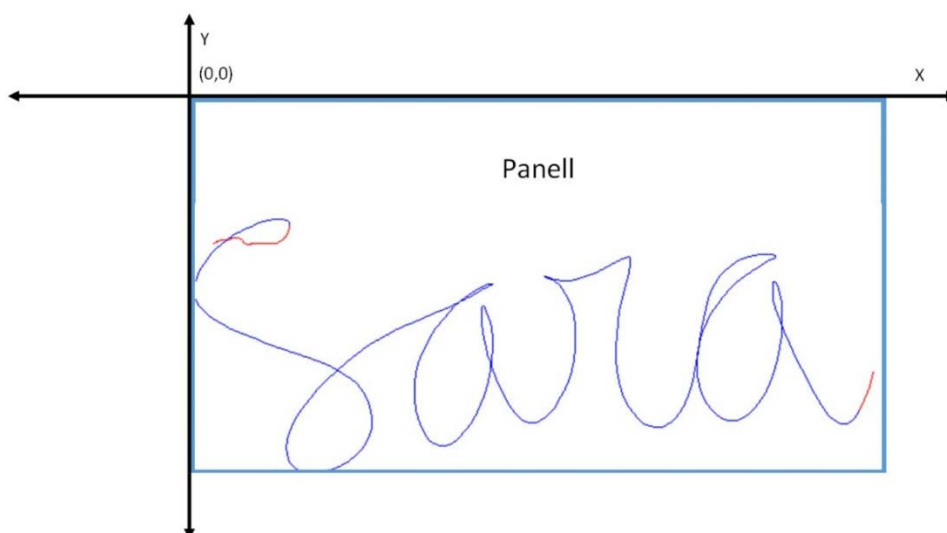


Fig. 5.6. Esquema dels eixos de coordenades i un dibuix en relació a un panell, amb tractament.

S'ha optat per la segona solució per motius visuals, ja que al canviar l'origen de coordenades a la part inferior, en cas de necessitar barres de desplaçament, aquestes no funcionen bé.

Utilitzant la segona opció, en tot moment, abans de dibuixar una línia a la instància de Graphics2D, s'agafa la mida actual del panell i se li resta la coordenada Y. D'aquesta manera se simula que l'origen de coordenades és el correcte per poder tenir els dos eixos positius a la part del panell que està a la vista.

Per construir els diferents tipus de dibuixos el que s'ha fet és aprofitar l'herència de Drawing. Aquesta classe contindrà tot l'algoritme per mostrar un dibuix qualsevol i només es desenvoluparan petites particularitats de cada tipus de dibuix, cadascuna en la seva subclasse. Aquestes particularitats seran, el filtre per la columna Estat i poder mostrar solament la part que correspon al dibuix concret.

Al fer les primeres proves, es va veure que la densitat de píxels per centímetre quadrat de la tauleta i dels panells és molt diferent. Això provoca que si les dades no són tractades prèviament a la creació de dibuix, aquest es crea amb una magnitud força més gran que el panell (encara que aquest estigui a pantalla completa) i el dibuix no es visualitza.

Aquest tractament previ, per defecte serà l'ajustament automàtic de la mida, a la pantalla. D'aquesta manera, si es vol mostrar un dibuix per pantalla, per defecte aquest es posarà a la mida del panell.

5.5. Detall dels dibuixos

Per tal de mostrar un dibuix de manera més detallada i donant més opcions a l'usuari per treballar amb aquest, com ja s'ha explicat, cada dibuix es podrà obrir per separat amb una finestra nova i es podrà obrir tantes vegades com es vulgui o es necessiti.

Dins la pestanya corresponent de la finestra interna de detall, dalt de tot es mostra el dibuix seleccionat, en un espai tan gran com es faci la finestra. Sota el dibuix, es donen les diferents opcions de visualització, modificació i exportació del dibuix.

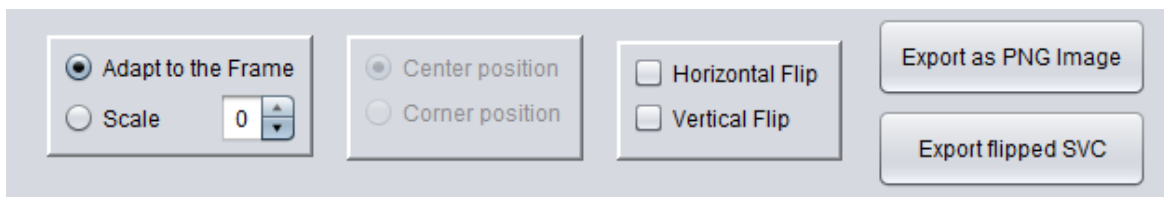


Fig. 5.7. Funcionalitats en la finestra de detall.

Quant a les opcions d'escalat del dibuix, la que s'utilitza per defecte és l'escalat automàtic a la mida de la finestra. Aquest escalat, mitjançant uns procediments, el que fa és reduir les dimensions del dibuix original a les dimensions màximes de la finestra, sempre deixant uns marges.

Cada cop que es vol mostrar de nou un dibuix, el primer que es fa és calcular un seguit de valors; valors màxims de X i Y, valors mínims de X i Y, relació d'aspecte, proporció entre les mides de la finestra i les mides del dibuix, etc.

Tots aquests valors, es calculen per tal de poder tenir una relació entre els valors reals i els valors que es necessiten per poder mostrar correctament un dibuix. El primer grup de valors que es busca són els màxims i mínims d'entre els valors de X i Y. Un cop es tenen aquests valors, es calcula l'increment de X i l'increment de Y per tal de saber les mides reals del dibuix. Aquestes mides del dibuix es comparen per saber si el dibuix és més llarg que alt o a la inversa. Seguidament depenent del cas es calcula la proporció del dibuix amb la pantalla, dividint la mida de la pantalla per la mida real del dibuix. La mida que no predomini (alçada o amplada), es calcularà a partir de la relació d'aspecte del dibuix i la mida escalada que predomini. La relació d'aspecte, s'ha calculat a partir de la divisió de les mides originals del mateix dibuix.

Amb aquests valors, seguidament es pot anar escalant punt a punt tot el dibuix, aconseguint així un dibuix a la mida de la finestra.

Per cada punt, el procediment d'escalat és el següent:

Valor de X = valor de X original - valor mínim de X del dibuix (per suprimir marges en blanc)
Valor de Y= valor de Y original - valor mínim de Y del dibuix (per suprimir marges en blanc)
Valor de X= valor de X * proporció X amb la pantalla
Valor de Y= valor de Y * proporció Y amb la pantalla

Fig. 5.8. Algoritme de mida automàtica d'un dibuix.

Per altra banda l'escalat a partir d'un factor, segueix el següent procediment. Directament a partir de les coordenades de cada punt, aplica el factor de reducció introduït per l'usuari.

Quan l'usuari selecciona com a escalat, un factor concret, aleshores la funcionalitat de posicionament s'activa. Quan l'escalat és automàtic, no es dona la possibilitat de modificar el posicionament, ja que el dibuix s'adapta a la finestra i per tant el posicionament dependrà de la forma del dibuix.

Com ja s'ha comentat, just abans de començar a dibuixar, es realitzen un seguit de càlculs per tal de tenir les relacions entre dibuix i finestra. Dins aquests càlculs també s'inclou el codi que busca les coordenades del punt central del dibuix i les coordenades del punt central de la finestra.

Per tal de calcular el centre del dibuix, es determina el mínim rectangle envoltant i es busca el centre d'aquest. Per al centre de la finestra, en ser un rectangle, es busca el centre directament.

Quan les opcions de posicionament s'activen, es donen dues possibilitats. La primera és posicionar el dibuix a l'origen de coordenades. Per tal de poder portar a terme aquesta acció el que es fa és buscar la diferència entre els valors mínim de X i Y dins el dibuix amb la coordenada (0,0). Aquesta diferència s'aplica a tots els punts del dibuix i així el dibuix queda traslladat.

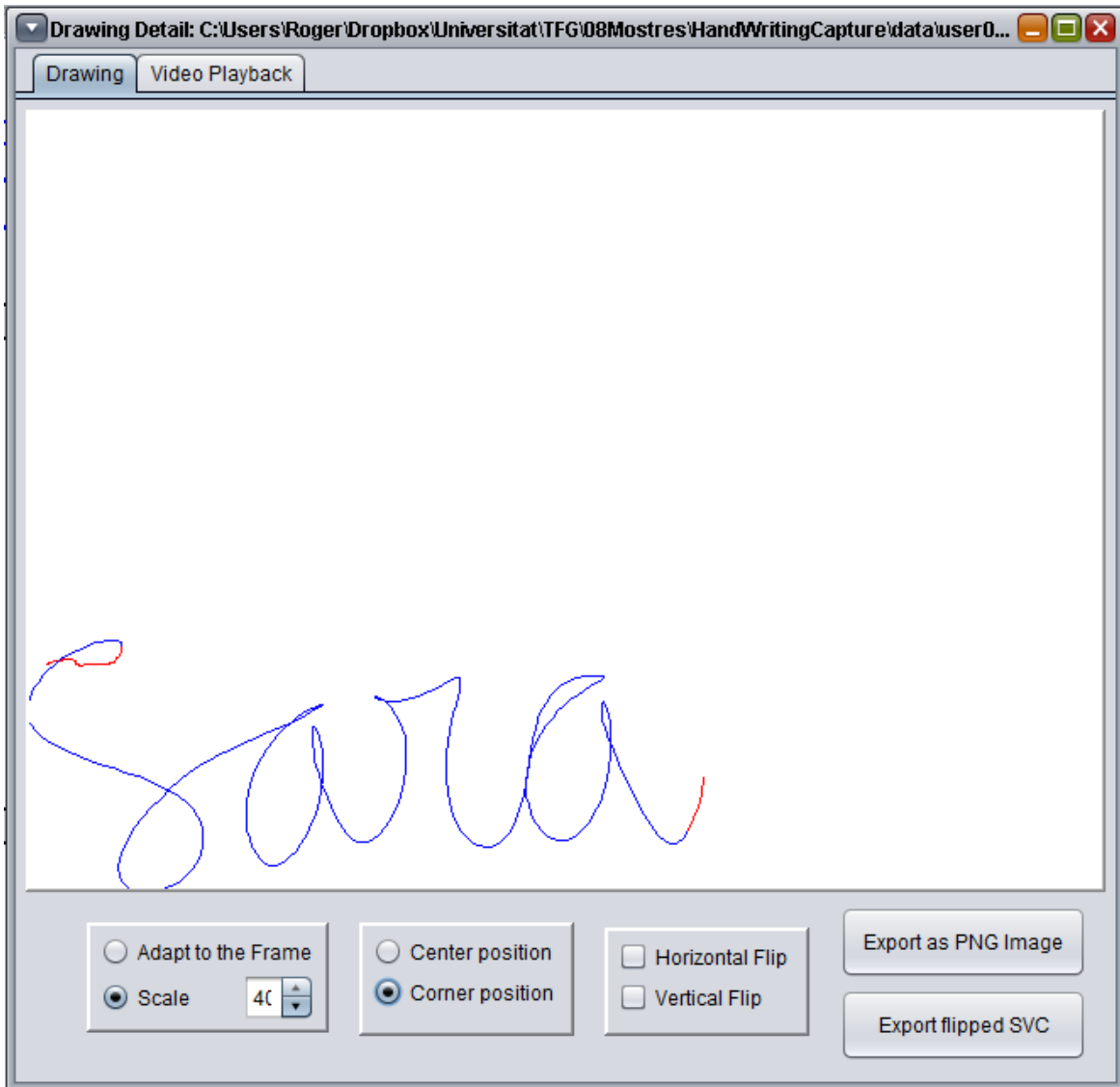


Fig. 5.9. Dibuix Escalat per factor i posionat a la cantonada.

Per altra banda també hi ha la possibilitat de centrar el dibuix a la finestra. Aquesta acció es duu a terme a partir del càlcul dels centres que s'ha fet prèviament. Un cop s'han calculat els dos centres, el que es fa és buscar la diferència de posició entre centres. Aquesta diferència s'aplica a tots els punts del dibuix per tal de canviar la posició.

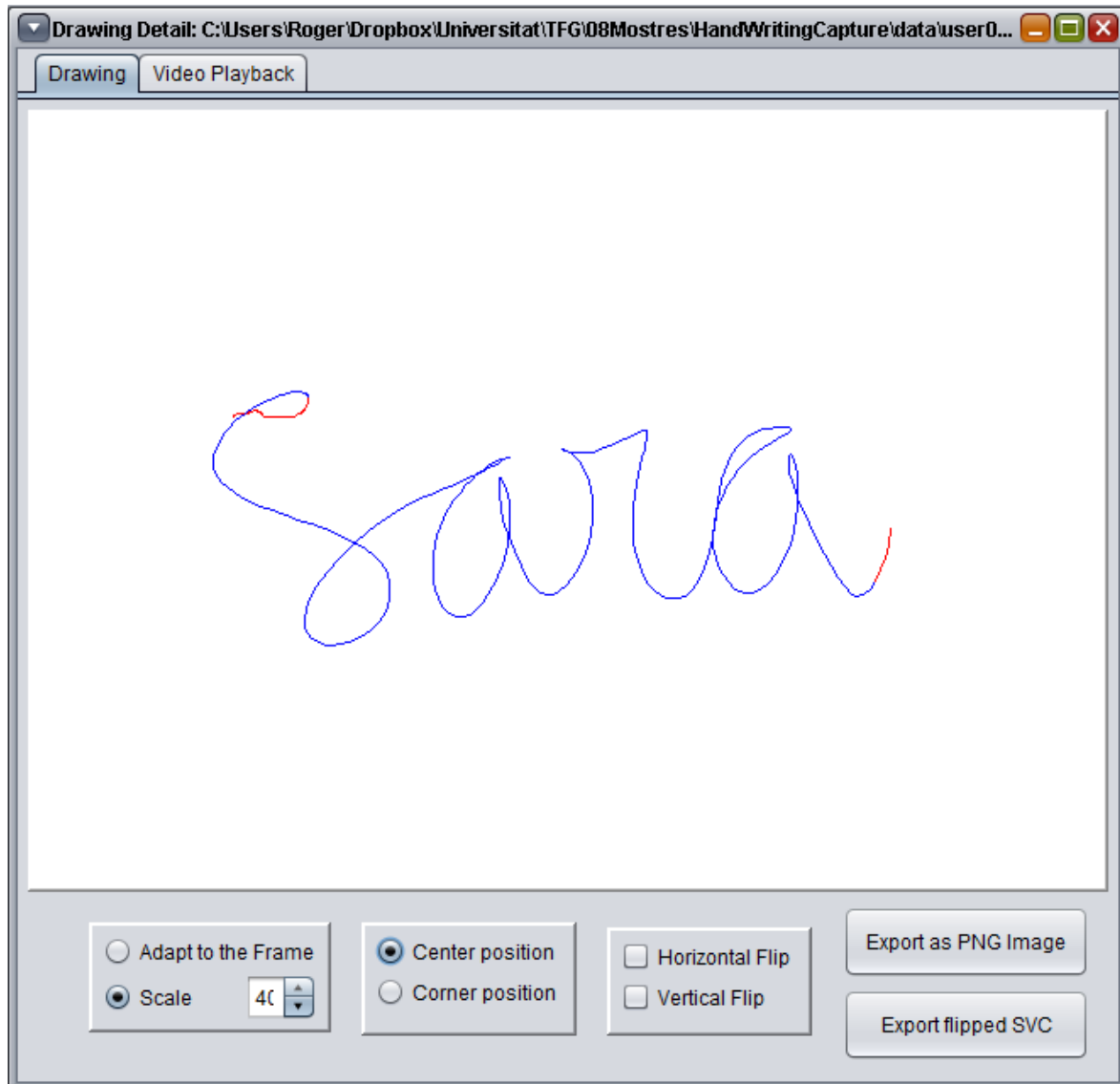


Fig. 5.10. Dibuix Escalat per factor i posicionat al centre.

5.6. Reproducció dels dibuixos

Per poder convertir un dibuix en un *vídeo*, el primer que es necessita és poder donar-li la funcionalitat de reproducció o el que seria equivalent, fer que el dibuix sigui *executable*.

Aquesta funcionalitat ens l'aportarà la interfície de la API de Java, Runnable. Runnable és una interfície que en ser implementada, converteix les instàncies de la classe en qüestió en instàncies executables. Per tal de què es puguin executar, Runnable obliga a implementar un mètode: `public void run()`.

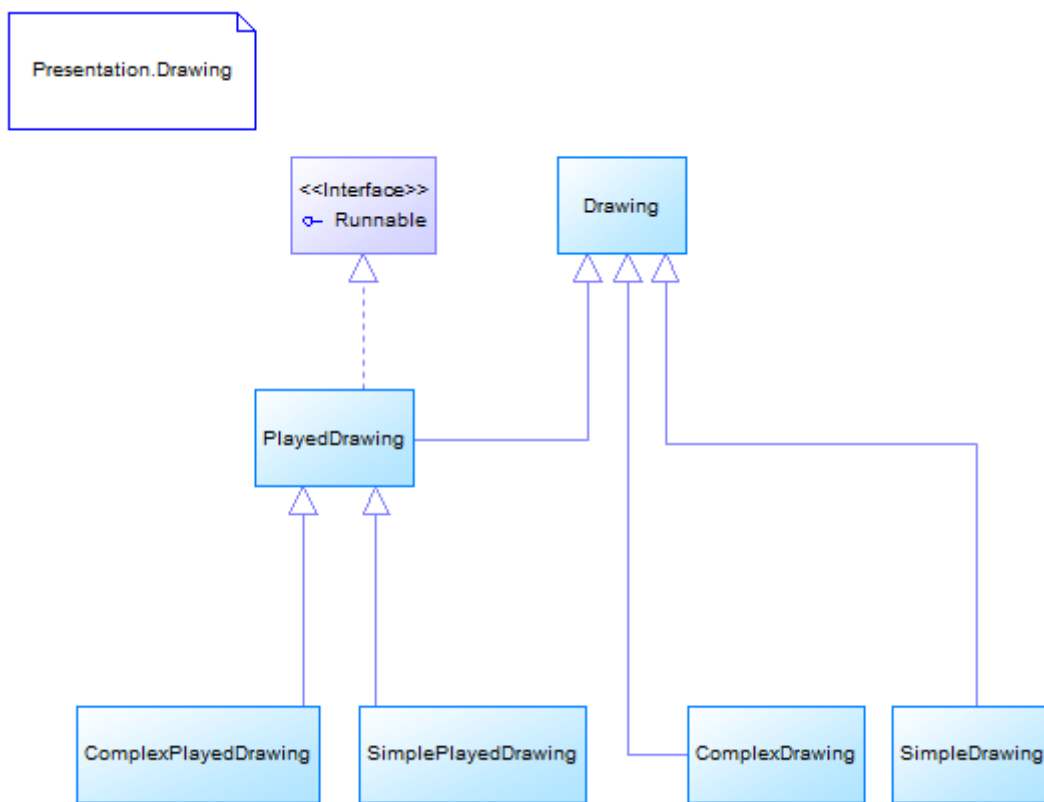


Fig. 5.11. Diagrama de classes parcial, herència de la classe Drawing.

Aquest mètode serà el que executarà tot el codi encarregat de donar “vida” a un dibuix.

Cada cop que s’executi el dibuix, s’executarà el mètode `run()`. El primer que es farà sempre és crear la finestra. Com que la variable instant estarà inicialitzada a zero, la finestra es mostrarà buida. A partir d’aquest punt, s’executa el bucle que va afegint punts al dibuix per tal d’anar-lo construint.

```
public void run() {
    mostrar finestra;
    loop (mentre no s'acabi la reproducció) {
        instant actual ++;
        actualitzar barra de progrés (instant actual);
        tornar a pintar la finestra ;
        temps= calcular increment de temps;
        if (temps > 0) {
            adormir fil d'execució actual(temps);
        }
        if (número total de punts del dibuix < instant) {
            acabat = true;
            acabar reproducció;
        }
    }
}
```

Fig. 5.12. Algorisme de reproducció d'un dibuix.

D'aquesta manera es va marcant l'instant de temps on es troba la reproducció i tot seguit es torna a pintar tota la finestra. Aquest fet el que fa és cridar al mètode que mostra el dibuix per pantalla igual que en la resta de dibuixos, amb la diferència que tindrà una barrera a l'hora de mostrar els punts, ja que només podrà mostrar fins al punt que marca la variable instant. Seguint aquest algorisme, cada cop que es refresca la finestra, s'actualitza tot el dibuix. Afegint els punts nous i refrescant els que ja s'havien mostrat.

Per donar a l'usuari una reproducció real, un cop ja s'ha actualitzat el dibuix, no es pot seguir amb la següent iteració, ja que aniria massa ràpid. El que es fa és calcular l'increment de la variable timestamp entre el punt actual i el seu següent i el fil d'execució s'adorm durant aquest valor calculat, respectant així, la velocitat de reproducció real.

Quant al control de la reproducció, s'ha implementat les accions bàsiques i necessàries:

- Play: dóna inici a la reproducció i automàticament queda desactivat fins que finalitza la reproducció, o bé es fa stop.
- Pause: s'activa en el moment que comença la reproducció i aporta la possibilitat de parar temporalment la reproducció. En cas que acabi la reproducció, ja sigui de manera natural o accionada per l'usuari, l'opció es desactiva.

- Resume: lligada a l'opció Pause, cada cop que Pause s'activa, s'activa l'opció Resume per tal de poder prosseguir la reproducció des del punt on s'ha deixat. Si l'usuari acciona la funció Stop, es desactiva.
- Stop: sempre que la reproducció està executant-se o està pausada, stop estarà activa. Stop finalitza la reproducció i la retorna al seu punt inicial.

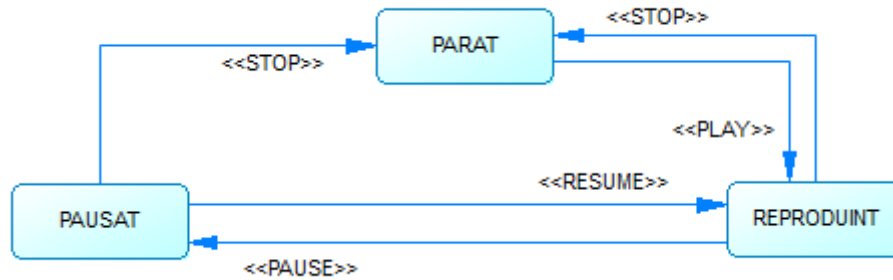


Fig. 5.13. Diagrama d'estats de la reproducció d'un dibuix.

Per tenir més joc a l'hora de reproduir un dibuix, s'ha implementat una barra lliscant, amb la qual es pot modificar la velocitat de reproducció del dibuix. La velocitat de reproducció, ve donada per l'increment de la variable timestamp, ja que és el temps que el fil d'execució s'adormirà cada iteració per tal de donar-li realisme. Al modificar la velocitat el que es fa és modificar aquest temps d'espera. En cas d'augmentar la velocitat el que es fa és dividir el temps d'espera entre el factor seleccionat per l'usuari. En cas contrari el que es fa és multiplicar-lo per poder reduir o augmentar el temps d'espera i per tant accelerar o frenar la reproducció.

En tot moment l'usuari necessita tenir una referència del punt on es troba la reproducció, és per això que també s'ha implementat una barra d'estat on a mesura que avança la reproducció, es va actualitzant el valor del percentatge. D'aquesta manera, visualment l'usuari pot estar situat. En la Fig. 5.12. es pot observar com el percentatge s'actualitza un cop per cada iteració del bucle.

Finalment, quan l'estat de la reproducció és pausat, s'activen les opcions d'exportar el dibuix actual de manera parcial. Per una banda es pot exportar la part fins a l'instant on s'ha parat la reproducció i per altra banda es pot exportar la part que queda a partir de l'instant i fins al final de la reproducció. En ambdós casos, el resultat és un nou arxiu SVC que només contindrà una part del dibuix original.

5.7. Mesures dels dibuixos

Tot el conjunt de mesures que es calculen per a cada dibuix, es calcula a partir de les dades que s'extreuen dels arxius SVC. Algunes de les mesures comparteixen càlculs, ja que el que canvien són les dades però no el procediment de càlcul. Seguidament s'explica com es calcula cadascuna d'elles.

General Measures			
Airtime:	647,00 ts	Air total length:	3454,11 px
Paper Time:	2480,00 ts	Paper total length:	59402,20 px
Air average speed:	5,36 px/ts	Pressure Entropy:	6,19
Paper average speed:	24,06 px/ts	Altitude Entropy:	3,62
Minimum Pressure:	24,22 %	Azimuth Entropy:	5,08
Maximum Pressure:	68,36 %	Pen-down Fractal Dimension:	1,09
Average Pressure:	48,36 %	Pen-up Fractal Dimension:	0,84
		Pen-up + Pen-down Fractal Dimension:	1,14

Fig. 5.14. Captura de la part de Mesures.

5.7.1. Mesures Simples

- Temps “en-l’aire” i “en-el-paper”

A partir d’una iteració, es van repassant tots els valors de l’arxiu. Per cada valor, es comprova a quin estat pertany (“en-l’aire” o “en-el-paper”), en cas que el punt pertanyi a l’estat que s’ha especificat, es compara per saber si és un valor d’inici de segment o de fi de segment. D’aquesta manera es van identificant els diferents segments que pertanyen a un mateix estat i per cada segment es mira l’increment entre els valors de timestamp. El valor del temps en un dels dos estats, serà la suma de tots els seus segments.

```
for (tots els punts de l'arxiu) {
    if (estat del punt == estat definit) {
        if (origen == 0) {
            origen = timestamp;
            fi = timestamp;
        } else {
            fi = timestamp;
        }
    } else {
        if (origen != 0 && fi != 0) {
            suma += fi - origen;
            origen = 0;
            fi = 0;
        }
    }
}
```

Fig. 5.15. Algoritme de càlcul de temps “en-l’aire” i “en-el-paper”.

Per poder calcular una de les dues mesures, es crida la funció encarregada del càlcul i se li passa l’estat que es vol calcular per paràmetre.

- Velocitat mitjana “en-l’aire” i “en-el-paper”

Igual que en el càlcul del temps “en-l’aire” i “en-el-paper”, aquest té com a base una iteració per totes les línies de l’arxiu. En aquest cas s’exclou l’última, ja que per calcular la velocitat instantània, es farà agafant parelles de punts (punt actual i punt següent) i calculant la velocitat entre la parella de punts, i per tant com que l’última fila no té punt següent, no tindrà velocitat. Per cada punt, el primer que es fa és mirar si el seu estat i l’estat del següent punt són iguals. En cas que comparteixin l’estat, es calcularà la velocitat instantània i es guardarà, així un cop iterats tots els punts, es tindran totes les velocitats instantànies en aquell estat i només s’haurà de fer la mitjana. En cas que no comparteixin el mateix estat, no es farà, ja que aquell punt és un punt de final de segment.

```

double suma = 0.0;
int total = 0;
for (totes les línies menys l'última) {
    int[] linia = getCoordenades(i);
    int[] liniaSeguent = getCoordenades(i + 1);

    if (estat linia == status && estat liniaSeguent == status) {
        int xLinia = linia [0];
        int yLinia = linia [1];
        int xLiniaSeguent = liniaSeguent [0];
        int yLiniaSeguent = liniaSeguent [1];
        int xVector = xLiniaSeguent - xLinia;
        int yVector = yLiniaSeguent - yLinia;
        double distancia = Math.sqrt(Math.pow(xVector, 2) +
Math.pow(yVector, 2));
        int tempsLinia = getTimestamp(i);
        int tempsLiniaSeguent = getTimestamp(i + 1);
        double temps = tempsLiniaSeguent - tempsLinia;
        double velocitat = distancia / temps;
        suma += velocitat;
        total++;
    }
}
double mitjana = suma / total;

```

Fig. 5.16. Algoritme de càlcul de la velocitat mitjana “en-l’aire” i “en-el-paper”.

Per calcular la velocitat instantània, es farà a partir de la següent fórmula:

$$velocitat = \frac{\Delta x}{\Delta t} \quad (5.1)$$

On Δx és l'increment de la distància i Δt és l'increment del temps. Per poder calcular Δx , s'utilitzaran vectors, ja que l'única dada referent a la distància que tenim són les coordenades dels punts. Per tant:

$$\vec{v} = (x_1 - x_0, y_1 - y_0) \quad (5.2)$$

$$\vec{v} = (x, y) \quad (5.3)$$

$$|\vec{v}| = \sqrt{x^2 + y^2} \quad (5.4)$$

$$|\vec{v}| = \Delta x \quad (5.5)$$

On x_n i y_n són les coordenades dels punts actual i següent, \vec{v} el vector que sorgeix de la unió dels dos punts i $|\vec{v}|$ el mòdul del vector o llargada del vector i per tant Δx . Per altra banda:

$$\Delta t = t_1 - t_0 \quad (5.6)$$

On t_n són els valors de la variable timestamp del punt actual i següent.

Un cop estan calculades totes les velocitats instantànies, l'únic que cal fer és calcular la mitjana de totes les velocitats:

$$mitjana = \frac{\sum \text{velocitat instantània}}{\text{nombre de velocitats calculades}} \quad (5.7)$$

Per poder calcular una de les dues mesures, es crida la funció encarregada del càlcul i se li passa l'estat que es vol calcular per paràmetre.

- Pressió Màxima, Mínima i Mitjana

```
double min = 999999999.0;
double max = 0.0;
double suma = 0.0;
int total = 0;
for (totes les línies de l'arxiu) {
    int pressio = this.data.getPressure(i);
    if (pressio < min && pressio != 0) {
        min = pressio;
    }
    if (pressio > max) {
        max = pressio;
    }
    if (pressio != 0) {
        suma += pressio;
        total++;
    }
}
double mitjana = suma / total;
double[] pressures = new double[3];
pressures[0] = percentage(min);
pressures[1] = percentage(max);
pressures[2] = percentage(avg);
```

Fig. 5.17. Algoritme de càlcul de les pressions.

Amb una sola iteració que recorre el valor de la pressió de cada línia de l'arxiu, es busca el màxim, el mínim i se suma a un total que s'utilitzarà per calcular la mitjana. Un cop es tenen els tres valors, es crida el mètode *percentage(valor)*, que s'encarrega de posar el valor que li passa en percentatge en relació al valor màxim de pressió que l'usuari ha especificat a la part de configuració de l'aplicació.

En aquest cas, es busquen les tres mesures de cop, degut que així s'evita fer la mateixa iteració tres cops.

- Longitud total de la traça “en-el-paper” i “en-l’aire”

```

double suma = 0.0;
for (totes les línies de l'arxiu menys la última) {
    int[] linia = getCoordenades(i);
    int[] liniaSeguent = getCoordenades(i + 1);
    if (estat linia == status && estat liniaSeguent == status) {
        int xLinia = linia [0];
        int yLinia = linia [1];
        int xLiniaSeguent = liniaSeguent [0];
        int yLiniaSeguent = liniaSeguent [1];
        int xVector = xLiniaSeguent - xLinia;
        int yVector = yLiniaSeguent - yLinia;
        double distancia = Math.sqrt(Math.pow(xVector, 2) +
            Math.pow(yVector, 2));
        suma += distancia;
    }
}

```

Fig. 5.18. Algorisme de càlcul de les longituds de les traces.

Aquest càlcul és semblant al de la velocitat mitjana, en la part de càlcul de vectors. Té com a base una iteració per totes les línies de l’arxiu. En aquest cas també s’exclou l’última, ja que, altre cop, es calcularà la distància entre dos punts (actual i següent) i l’última fila, com que no en té cap més a continuació (no té següent), no cal incloure-la al càlcul. Per cada punt, el primer que es fa és mirar si el seu estat i l’estat del següent punt són iguals. En cas que siguin iguals es calcularà la distància entre ells, en cas contrari no, ja que el punt serà un final de segment.

La distància entre els dos punts, es trobarà a partir del vector que es pot crear entre ells:

$$\vec{v} = (x_1 - x_0, y_1 - y_0) \quad (5.8)$$

$$\vec{v} = (x, y) \quad (5.9)$$

$$|\vec{v}| = \sqrt{x^2 + y^2} \quad (5.10)$$

$$|\vec{v}| = \Delta x \quad (5.11)$$

On x_n i y_n són les coordenades dels punts actual i següent, \vec{v} el vector que sorgeix de la unió dels dos punts i $|\vec{v}|$ el mòdul del vector o llargada del vector. Les diferents distàncies que es van calculant, es van guardant a una variable suma, que al final de la iteració tindrà el valor de la distància total en l’estat especificat (“en-l’aire” o “en-el-paper”).

Per poder calcular una de les dues mesures, es crida la funció encarregada del càlcul i es passa l'estat que es vol calcular per paràmetre.

5.7.2. Mesures Avançades

En aquest apartat s'utilitzen mesures que no són tan comunes ni conegudes, de manera que abans de la implementació d'aquestes s'explicarà de què es tracta cada una.

- Entropia de la pressió, de l'angle azimut i de l'angle altitud

En l'àmbit de la informació, l'entropia és una magnitud que defineix la incertesa de la informació o de la seva font, és a dir, quantifica la quantitat d'informació continguda en un recull de dades concret.

Per poder-la calcular partim de x que serà una variable aleatòria i discreta, altrament dit, que serà una variable que podrà tenir un valor qualsevol dins d'un rang definit. També tindrem una funció de probabilitat marginal, on $p(x)$ és la probabilitat d'aparició d'un determinat valor. Com que no es coneix la funció de probabilitat, el que es fa és aproximar la probabilitat a partir de la freqüència d'aparició d'un valor concret.

Així doncs definim l'entropia de x de la següent manera [1]:

$$H(x) = - \sum_{x \in X} p(x) \cdot \log_2(p(x)) \quad (5.12)$$

A l'hora de calcular, el que es farà és calcular $p(x) \cdot \log_2(p(x))$ on:

$$p(x) = \frac{\#d'aparicions\ de\ x\ a\ la\ mostra}{\#total\ de\ linies\ de\ la\ mostra} \quad (5.13)$$

Per cada valor de x que apareix a la mostra. Seguidament es fa el sumatori dels valors de cada punt i es nega.

En el nostre cas concret la variable x agafarà els valors de la columna que es vol calcular, en aquest cas Pressió, Angle Azimut o Angle Altitud.

```
ArrayList<Integer> usats = new ArrayList<Integer>();
double px, part, total, real, mida;
total = 0;
for (totes les línies de l'arxiu) {
    int valor = get(i);
    if (usats no conté (valor)) {
        usats afegir(valor);
        real = frecuencia(tots els valors, valor);
        mida = getTotalLinies();
        px = (real) / (mida);
        part = (px) * (java.lang.Math.log(px) / java.lang.Math.log(2));
        total += part;
    }
}
total = total * -1.0;
```

Fig. 5.19. Algoritme de càlcul de les entropies.

En la implementació, dins la iteració es van sumant els diferents blocs de $p(x) \cdot \log_2(p(x))$, on per poder calcular $p(x)$ es crida al mètode “*frecuencia(tots els valors, valor)*”, al qual es passa la col·lecció de valors i el valor concret.

El mètode “*frecuencia*”, el que farà és fer una altra iteració per tots els valors, per tal de poder contar en nombre d'aparicions d'aquest valor en la col·lecció.

- Dimensió Fractal del dibuix “en-el-paper”, del dibuix “en-l’aire” i del dibuix complet (traça “en-el-paper”+ traça “en-l’aire”)

És una magnitud que s'utilitza per quantificar la complexitat de les dades obtingudes. A partir d'aquesta definició i com es defineix a [2], en capturar les dades que l'usuari va dibuixant, ens dóna un seguit de punts. Si estan massa separats, es farà molt difícil de llegir, si estan massa junts, s'acabaran superposant i per tant es crea soroll en el dibuix.

Per poder mesurar aquesta *dificultat* a l'hora de llegir o entendre el dibuix, utilitzarem la Dimensió Fractal.

Per calcular-la es pot fer a partir de diferents orígens (punts, corbes, superfícies o 3D)[3] i mitjançant diferents mètodes (Àrea/perímetre, Box Counting, Divider reallion, etc.)[4].

Quant a l'origen, s'utilitzarà un conjunt de punts (combinació de les columnes X i Y de la mostra) i com a mètode s'utilitzarà “Box counting”[4] ja que en aquest cas, els punts seran píxels i són de forma quadrada.

El funcionament d'aquest mètode es basa en enquadrar la mostra en un rectangle i seguidament anar subdividint-lo en parts més petites. Aleshores comptar el número de subdivisions que estan “plenes”, és a dir, que contenen algun punt del dibuix. A partir d'aquí aplicar la fórmula per poder treure'n un valor resultant [2].

$$d = - \frac{\ln(N(r))}{\ln(r)} \quad (5.14)$$

On $N(r)$ seran el nombre de caselles pintades cadascuna escalada per un factor r [5, 6].

Depenent de la mida del costat de cada casella, el resultat serà més real o menys, és per això que s'intentarà treballar amb el costat més petit significatiu.

Aplicat a les mesures proporcionades per la tauleta, el costat serà de mida un píxel, per tant r serà la superfície en mil·límetres d'una de les caselles que subdivideixen el dibuix i $N(r)$ la suma de punts (columnes x i y) diferents entre si.

En utilitzar com a mida de costat 1 px, el que farem és passar-ho a mil·límetres. Per passar-ho el que es farà és buscar la densitat de píxels de la tauleta i a partir de la concentració, extreure'n la mida. Un cop trobades aquestes dades, només cal aplicar la fórmula i es troba el valor.

Tot el càlcul es pot veure en el següent algoritme:

```
double Nr;
if (type == 3) {
    Nr = puntsDiferents();
} else {
    Nr = puntsDiferents(type);
}
String[] dimension = dimensions de la tauleta en píxels;
double pXw = dimension[0];
dimension = dimensions de la tauleta en mil·límetres;
double mMw = dimension[0];
double r = (mMw / pXw);
double fractalDimension = (Math.log(Nr)) / (Math.log(r));
fractalDimension = fractalDimension * -1.0;
```

Fig. 5.20. Algoritme de càlcul de les dimensions fractals.

```
puntsDiferents () {
    ArrayList<String> usats = new ArrayList<String>();
    int num = 0;
    for (totes les línies de l'arxiu) {
        int[] linia = getCoordenades(i);
        String punt = linia [0] + " " + linia [1];
        if (usats no conté el punt) {
            afegir a usats(punt);
            num++;
        }
    }
    return num;
}
```

Fig. 5.21. Algoritme de càlcul de la freqüència d'aparició d'un valor a una col·lecció.

Depenent el tipus d'origen de dades que s'utilitzi i de la seva dimensió, el resultat haurà de tenir una magnitud o una altra. En aquest cas, com que s'està treballant amb dibuixos formats per línies, el resultat acostumarà a ser entre 1 i 2. En cas que s'acosti més a 1 voldrà dir que és un dibuix més aviat clar i entenedor (poc desordenat) i en cas que s'acosti més a 2 voldrà dir que té més desordre.

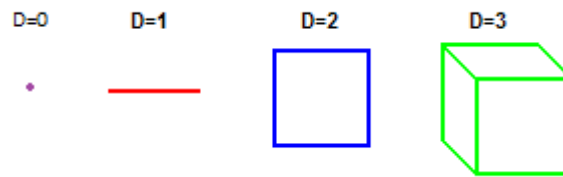


Fig. 5.22. Possibles dimensions dins el càlcul de la Dimensió Fractal.

- Logaritmes en Java

En el cas de les dues mesures avançades, a l'hora de realitzar el càlcul es necessita utilitzar logaritmes. Per al càlcul de l'entropia, més concretament es vol calcular el $\log_2 x$. Cosa que la llibreria *java.lang.Math* no inclou. És per això que s'ha buscat una igualtat amb una funció de la llibreria mitjançant el següent raonament:

$$y = \log_2 x \quad (5.15)$$

$$x = 2^y \quad (5.16)$$

$$\ln(x) = \ln(2^y) \quad (5.17)$$

$$\ln(x) = y * \ln(2) \quad (5.18)$$

$$y = \frac{\ln(x)}{\ln(2)} \quad (5.19)$$

En aquest cas el logaritme natural sí que està contemplat en la llibreria com a *java.lang.Math.log(valor)*.

En el cas de la dimensió Fractal, com que ja ve expressada en logaritmes naturals, es farà servir aquesta mateixa funció.

5.7.3. Gràfiques

Quant a les gràfiques, per poder donar el màxim de funcionalitats a l'usuari, s'ha buscat una llibreria que permetés modificar aspectes visuals del gràfic per part de l'usuari, així com exportar el gràfic en format d'imatge.

La llibreria escollida ha estat JFreeChart [12,15], una llibreria totalment gratuïta i desenvolupada íntegrament en Java. Per tal de poder mostrar els gràfics a la finestra, es crea una instància de l'objecte JFreeChart per cada gràfic que es vol mostrar. Aquestes instàncies es tractaran com a components dins la finestra.

Les instàncies que es van creant, necessiten alimentar-se d'un conjunt de dades. Aquest conjunt de dades s'anirà introduint dins una instància de la classe XYDataset i aquesta instància serà la que proporcionarà les dades al gràfic.

Per poder donar les dades a la instància de XYDataset, cada tipus de gràfic necessita el seu propi procediment.

- Gràfica pressió en funció del temps

En el cas de la pressió en funció del temps, el que es fa és recórrer tot l'arxiu i es van afegint parelles de valors (timestamp, pressió) a la sèrie del XYDataset. Com que el primer punt del dibuix no té un timestamp zero i només interessa l'increment d'aquesta variable, a tots els valors de timestamp se li restarà el valor del primer punt, per tal d'aconseguir aquest increment de temps.

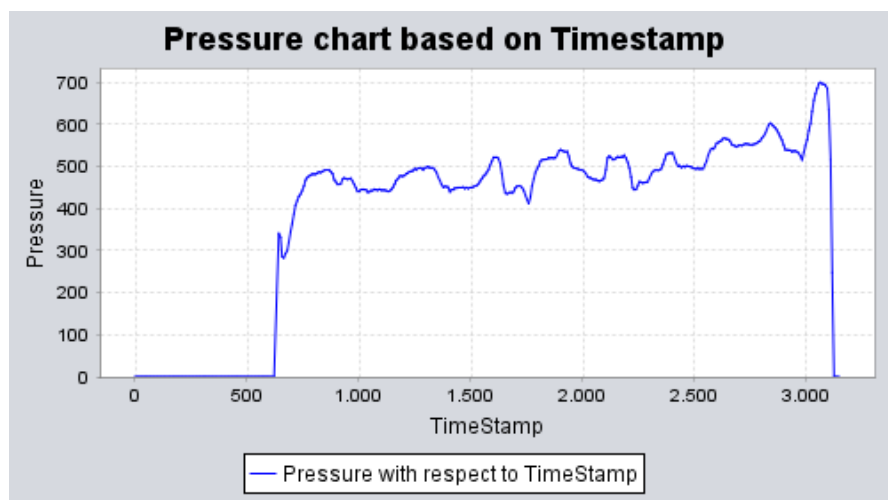


Fig. 5.23. Captura de la Gràfica de pressió en relació al temps.

- Gràfica velocitat en funció del temps

Per poder omplir el XYDataset d'aquest gràfic, primer de tot el que es farà és crear dues sèries diferents. D'aquesta manera una sèrie serà per marcar la velocitat del dibuix “en-el-paper” i una per “en-l'aire”. Aquestes dues sèries s'aniran omplint a l'hora aprofitant que es recorrerà tot l'arxiu calculant, les velocitats instantànies de cada punt.

Per poder alimentar cada sèrie es necessitaran parelles de valors (timestamp, velocitat), on el timestamp no serà el valor, sinó l'increment (igual que en el gràfic de la pressió en funció del temps) i la velocitat serà la velocitat instantània, que es calcula com s'ha explicat en l'apartat de mesures simples.

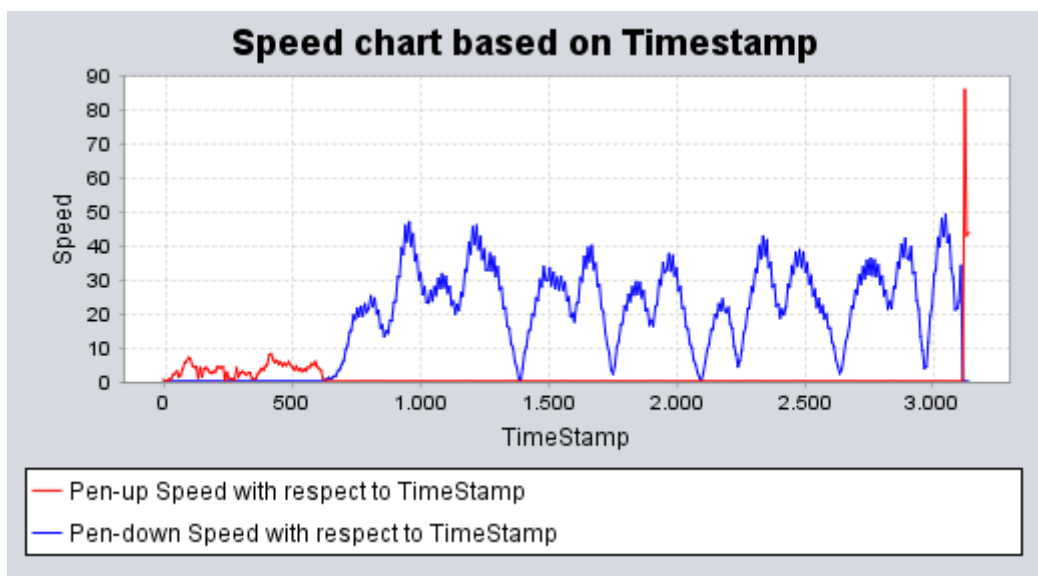


Fig. 5.24. Captura de la Gràfica de la velocitat en relació al temps.

5.8. Correcció d'errors en els dibuixos

A partir de la visualització de diverses col·leccions de mostres SVC, s'han detectat diferents errors que es donen dins els arxius SVC en ser creats. Les problemàtiques i les solucions han estat les següents:

- Arxius SVC sense punts a l'interior, només amb una sola línia amb la suma del total de línies a zero. S'ha decidit avisar a l'usuari que l'arxiu està buit i no obrir.
- Arxius SVC amb múltiples dibuixos en el seu interior. L'arxiu conté una estructura idèntica al que serien dos arxius SVC creats correctament però tots dos dins el mateix arxiu (concatenats). En lloc de contenir un sol dibuix, en conté dos, amb dos recomptes de files diferents, un al principi de cada dibuix. La solució ha estat avisar a l'usuari que l'arxiu conté múltiples dibuixos que es mostraran per separat. El sistema separa els dibuixos com si es tractés de dos arxius diferents i els mostrarà per pantalla com dos arxius diferents.
- Arxius SVC amb errors en els valors de l'atribut temporal. A causa d'un error desconegut, s'ha trobat punts concrets on el valor temporal té una magnitud totalment desorbitada. La solució que s'aplica és fer la mitja entre els dos valors predecessors al punt que conté l'error, per tal de poder tenir un valor coherent, i fer servir aquest valor calculat en lloc del valor corrupte.
- Arxius SVC que en el seu moment es van crear amb la tauleta mal posada i per tant el programa va registrar els dibuixos amb un efecte mirall. Tot i no ser un error, ja que el programa no té cap problema a l'hora de llegir les dades, sí que ho pot ser per l'usuari, ja que no té una representació real com es vol tenir. És per això que s'ha buscat la manera de poder jugar amb efectes mirall, tant en direcció vertical com horitzontal com ambdues simultàniament.

Per poder donar aquestes efectes al dibuix, es fa de la següent manera:

```
if (this.currentFlip == Drawing.NO_FLIP) {
    return new double[] { xCoord, (h) - yCoord };
} else if (this.currentFlip == Drawing.HORIZONTAL_FLIP) {
    return new double[] { (w) - xCoord, (h) - yCoord };
} else if (this.currentFlip == Drawing.VERTICAL_FLIP) {
    return new double[] { xCoord, yCoord };
} else if (this.currentFlip == Drawing.HORZ_VERT_FLIP) {
    return new double[] { (w) - xCoord, yCoord };
}
```

Fig. 5.25. Algorisme d'aplicació d'efecte mirall al dibuix.

On h és la mida vertical de la finestra, w la mida horitzontal de la finestra i xCoord i yCoord són les coordenades del punt concret amb què s'està treballant. currentFlip identifica l'acció seleccionada per l'usuari i NO_FLIP, HORIZONTAL_FLIP, VERTICAL_FLIP i HORZ_VERT_FLIP són les possibles accions amb un dibuix.

5.9. Exportació d'informació

Tota la informació que es mostra en l'aplicació és important que es pugui importar, per tal de poder-la guardar, o seguir utilitzant externament. S'han desenvolupat diferents opcions per a l'exportació:

- Informe PDF

Una manera senzilla i pràctica de donar a l'usuari totes les dades en un format amigable i sobretot útil, és creant un informe en PDF on queden reflectits tots els camps i dades utilitzats dins l'aplicació.

Com que la col·lecció de dades no és petita, prèviament a la creació de l'informe, es pregunta l'usuari on vol guardar aquest informe i quins són els camps que li interessa que es mostrin, evitant d'aquesta manera feina innecessària.

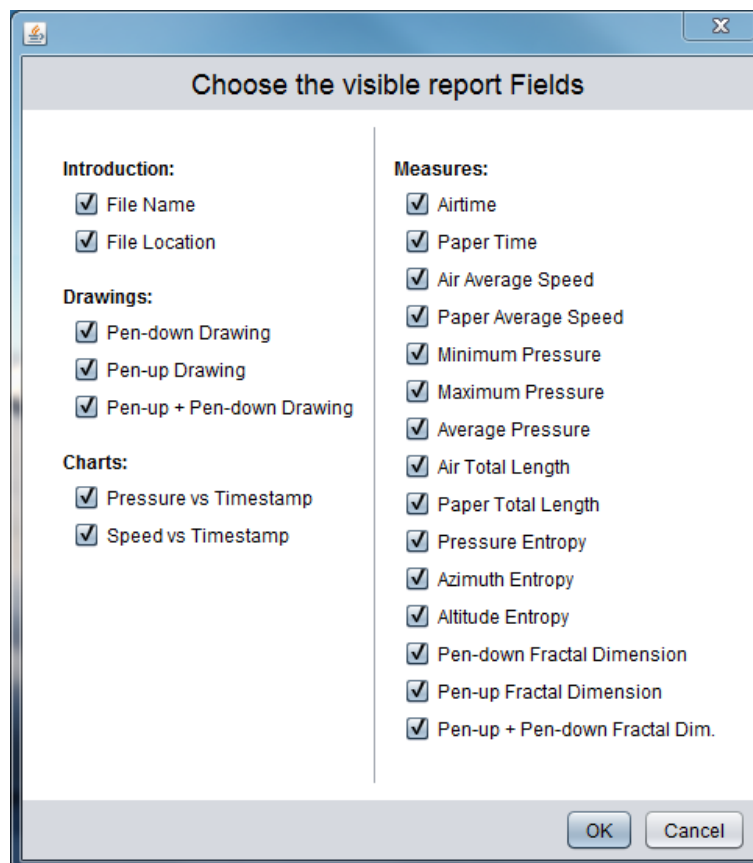


Fig. 5.26. Captura de la finestra de selecció dels camps per al Informe.

Un cop ja s'han especificat el destí i el conjunt de camps que formaran l'informe, comença el procés de creació d'aquest. Per poder crear un informe des de la mateixa aplicació, s'han utilitzat diferents llibreries i components per l'entorn de desenvolupament.

Per una banda, hi ha el disseny de l'informe que s'ha fet a partir de Jasper Reports [16,17]. Per poder fer-ne el disseny s'ha utilitzat el component de Jasper Reports per Eclipse, per poder-lo executar des de dins l'aplicació no s'ha fet ús del component per a Eclipse, sinó directament de la llibreria de Jasper Reports. El disseny s'ha pensat el més dinàmic possible, per aconseguir-ho, s'ha dividit l'informe en quatre apartats i cada apartat conté els seus camps. Cada camp té dues variables associades, una que conté el valor que ha de mostrar i una segona, de format Boolean, que informarà si el camp en qüestió ha estat seleccionat per l'usuari o no.

Cada cop que es crea l'informe, cada camp avalua les seves variables. En cas que s'hagin de mostrar, es mostren, en cas contrari es suprimeix l'espai que pertoca al camp. D'aquesta manera s'eviten espais en blanc i informes extensos amb poca informació. Aquest disseny farà de plantilla per a tots els informes que es creïn des de l'aplicació.

Per últim, la creació de l'informe es fa internament a l'aplicació. Tot i que el component per a Eclipse ja ve preparat per a l'exportació, per a fer tot el procés de creació des de la pròpia aplicació, implica tot un conjunt de biblioteques auxiliars a la de Jasper Reports.

Aquestes biblioteques són; commons beanutils [19], commons digester [20], commons logging [21], commons collections [22], groovy-all [23], itext-pdfa [24], itext-xtra [24], itextpdf [24], jasper-compiler-jdt [17] i jcommons [25]. Totes i cada una d'elles són necessàries, ja sigui per la creació del pdf, per a la compilació del report o per a la recollida de dades d'una col·lecció. Un cop les llibreries estan importades al projecte, el procediment de creació és relativament senzill.

```
JasperReport jR = JasperCompileManager.compileReport(url);  
  
JasperPrint jP = JasperFillManager.fillReport(jR, data, new  
JREmptyDataSource());  
  
JasperExportManager.exportReportToPdfFile(jP, this.fileDesti);
```

Fig. 5.27. Codi per a la creació de l'informe en PDF.

- Dibuixos en format Png

Dins la pantalla de detall, a la pestanya de dibuix, aprofitant que és l'espai de treball on l'usuari pot modificar visualment el dibuix, és on es presenta l'opció de poder exportar-ne el dibuix en format d'imatge (PNG).

Per a poder crear aquesta imatge s'aprofita la pròpia API de Java, ja que qualsevol classe que estén de la classe Component té la capacitat de poder-ne exportar les dades gràfiques. D'aquesta manera un cop es tenen les dades, aquestes es plasmen en una imatge.

```
BufferedImage image = new BufferedImage(component.getWidth(),
component.getHeight(), BufferedImage.TYPE_INT_RGB);

component.paint(image.getGraphics());

ImageIO.write(img, "png", new File(file));
```

Fig. 5.28. Codi per a la exportació de dibuixos en format PNG.

- Exportació de dibuixos parcials en arxius SVC

Com ja s'ha vist en l'apartat de Reproducció dels dibuixos, s'han implementat dues exportacions en format SVC. La primera és per exportar la part que s'ha reproduït del dibuix i la segona és per exportar la part restant del dibuix, és a dir, la que encara no s'ha reproduït.

Per a desenvolupar aquestes dues funcionalitats s'ha fet de manera molt similar o gairebé exacta. Igual que s'ha llegit l'arxiu, per començar a treballar amb l'aplicació, en aquest cas es fa el procés invers. Es va demanant a la instància svcData la fila que es vol i la instància ja retornarà la línia en una sola variable de text, perquè pugui ser escrita directament al fitxer. Seguidament mitjançant la classe BufferedWriter, s'escriu la línia al fitxer.

Per tal de marcar el punt fins on o des d'on s'ha d'exportar, s'utilitzarà la variable instant, on hi ha guardat el punt on s'ha parat la reproducció.

- Arxius SVC amb efectes

En la finestra de detall d'un dibuix, es dona l'opció de realitzar un efecte mirall al dibuix, sigui horitzontal, vertical o ambdós alhora. Quan es porten a terme aquestes accions, es modifiquen les dades que es mostren per pantalla (el dibuix) però mai les originals que estan a la instància svcData. Això es fa per evitar problemes a l'hora de fer accions sobre aquestes dades i evitar problemes en cas que es vulguin desfer.

És per això que si es vol exportar l'arxiu un cop se li ha donat un efecte mirall, és igual quin, el que es fa és tornar a la instància svcData i recuperar les dades originals a partir de les quals es tornarà a fer l'efecte mirall. Aquestes dades seran les que s'escriuran a l'arxiu SVC.

Per a l'escriptura se seguirà el mateix procediment que utilitza l'exportació de dibuixos parcials en arxius SVC, però sense fragmentar el dibuix, en aquest cas es donarà l'opció d'exportar una part o la totalitat del dibuix.

- Gràfiques en format PNG

Les gràfiques, com ja s'ha explicat, es creen a partir de la llibreria JFreeChart. Aquesta llibreria, al ser tan completa ja integra una funcionalitat per a l'exportació dels gràfics. Quan l'usuari fa botó dret sobre el gràfic, ja es despleguen les opcions que permet aquesta llibreria.

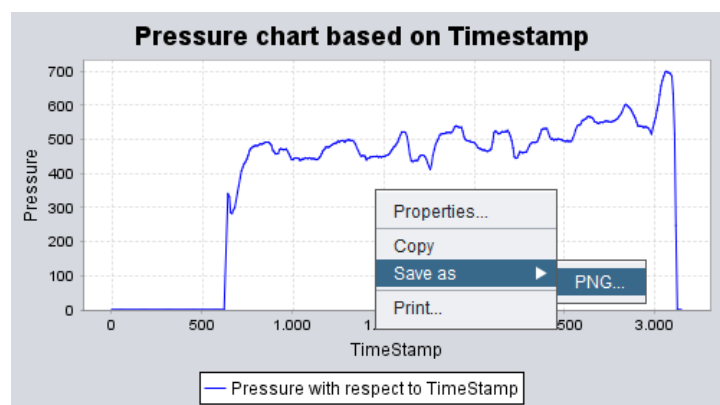


Fig. 5.29. Captura del menú per a exportar un gràfic en format PNG.

5.10. Persistència de paràmetres

Aquesta aplicació llegeix i genera moltes dades. En altres aplicacions, el flux normal seria que totes les dades que passen per l'aplicació, en cas de voler-les fer persistents, es guardessin en una base de dades. En el nostre cas, l'origen de les dades sempre serà un arxiu SVC i són dades que no tenen un destí. Normalment només són llegides, i en cap cas guardades. Si es dóna la situació que es volen guardar, ja hem vist que tenim opcions per exportar-les de diverses maneres.

És per això que aquesta aplicació no s'ha dissenyat acompanyada d'un sistema de base de dades. El volum de dades que es volen guardar dins l'aplicació és tan ínfim, que queda reduït a un seguit de paràmetres de configuració de la pròpia aplicació.

Per guardar aquesta configuració s'ha utilitzat la classe de Java Properties. Aquesta classe serà l'encarregada de manipular tots els paràmetres que es volen llegir o guardar quan s'utilitza l'aplicació. Ella mateixa s'ajuda d'un fitxer de text on guarda parelles de clau-valor per poder fer persistents aquests paràmetres i no perdre'ls quan es tanca l'aplicació.

Per tal de poder treballar amb la classe, s'han utilitzat els següents mètodes:

- `load(FileInputStream)`: a partir del canal de dades (que apunta a l'arxiu de text), la instància de Properties que s'ha creat, carrega totes les propietats en temps d'execució perquè puguin ser utilitzades.
- `getProperty(key)`: se li proporciona la clau que defineix un paràmetre i retorna el valor del paràmetre.
- `setProperty(key, value)`: se li passa la parella clau – valor i en cas que ja existeixi, actualitza els valors i en cas que no els escriu.
- `store(FileOutputStream, comments)`: un cop s'ha acabat de treballar, a partir del canal de dades que apunta al fitxer de text, guarda tots els paràmetres dins el fitxer, així com també els comentaris que se li passen.

```
tabletMmDimensions=325.1x203.2  
tabletMaxPressure=1024  
tabletPxDimensions=65024x40640
```

Fig. 5.30. Exemple d'arxiu de configuració.

Per eliminar possibles conflictes entre diferents instàncies de Properties interactuant amb el mateix fitxer de text, el que s'ha fet és aplicar el patró Singleton. El patró Singleton, és un patró que té com a finalitat donar un únic punt d'accés a una classe, global per tota l'aplicació.

Aquest patró s'implementa en tres passos:

- El constructor de la classe en qüestió ha de ser privat perquè cap classe pugui crear-ne una instància.
- Dins la classe hi ha d'haver un atribut de la pròpia classe però amb visibilitat privada i funcionament estàtic.
- Implementar un mètode que s'encarregui de subministrar la instància de la classe a les que la necessitin. El mètode haurà de ser sincronitzat i estàtic.

```
private static ApplicationSettings instance;  
  
private ApplicationSettings() {  
}  
  
public synchronized static ApplicationSettings getInstance() {  
    if (instance == null) {  
        instance = new ApplicationSettings();  
    }  
    return instance;  
}
```

Fig. 5.31. Exemple de patró Singleton.

6. Implementació

6.1. Llibreries

Durant el desenvolupament de l'aplicació, a causa dels múltiples requeriments, s'ha fet ús de llibreries externes a les pròpies de Java. D'aquestes llibreries, algunes són totalment necessàries, ja que la pròpia API de Java no conté cap característica semblant. Per altra banda n'hi ha algunes que sí que substitueixen alguna funcionalitat que ja té Java, però que a canvi aporten tot un recull de funcionalitats afegides, que són interessants de tenir.

Aquestes llibreries, totes de software lliure i gratuïtes, són les següents:

- Per al desenvolupament de Gràfiques
 - JFreeChart 10.0.19.jar [15]

- Per el desenvolupament d'informes en PDF
 - JCommon 10.23.jar [25]
 - Commons-logging 1.1.jar [21]
 - Commons-beanutils 1.8.2.jar [19]
 - Commons-digester 1.7.jar [20]
 - Commons.collections 3.2.1.jar [22]
 - Itextpdf 5.5.5.jar [24]
 - Itext-pdfa 5.5.5.jar [24]
 - Itext-xtra 5.5.5.jar [24]
 - Groovy-all 1.7.5.jar [23]
 - JasperReports 6.0.3.jar [17]
 - Jasper-compiler-jdt.jar [17]

7. Conclusions

Finalment, un cop ha estat desenvolupat tot el projecte, el resultat ha estat positiu. Amb la planificació establerta des de l'assignació del projecte fins al dia de l'entrega, s'ha pogut treballar de manera tranquil·la i eficaç. En tot moment s'ha tingut un marge de temps per possibles entrebancs que es poguessin trobar en el desenvolupament del projecte, cosa que ha estat molt positiva quan algun dels punts ha requerit més temps o més esforç del que en un primer moment es pensava.

Un cop acabat el projecte, s'ha pogut valorar molt positivament la planificació feta durant l'inici, ja que s'ha pogut comprovar que tot i petites desviacions de temps, en cap cas significatives, la feina s'ha pogut ajustar molt a la planificació definida.

A la versió final de l'aplicació s'ha pogut incloure, desenvolupats i testats, tots els requeriments classificats com a prioritaris, que s'han especificat tant en les reunions amb el tutor, com amb el responsable del GTS. Per altra banda hi ha hagut algun requeriment de poc valor afegit, que s'ha decidit no desenvolupar-lo i deixar-lo per a una futura versió de l'aplicació.

Referent a les problemàtiques sorgides durant el projecte, destacar que no s'ha trobat cap gran impediment. L'única cosa de la qual hi ha molt poca informació o gairebé inexistent és sobre els arxius SVC. Afortunadament, ha estat informació de la qual el tutor ja disposava i per tant no ha estat un problema massa greu a l'hora de començar el projecte. La resta de problemàtiques han estat per desconeixement de les llibreries i del seu funcionament, cosa que ha comportat dedicar-hi més hores de les previstes.

Quant a la documentació, com ja s'ha comentat, de bon principi va ser un problema a causa de la manca de coneixement sobre l'arxiu SVC. Un cop s'ha anat coneixent una mica més, tot i que el volum d'informació disponible no és massa extens, s'ha trobat la necessària. Per altra banda aquesta informació sempre ha estat d'autor contrastat, assegurant així la validesa del seu contingut.

Referent a l'aplicació, en tot moment s'ha treballat perquè el resultat fos un programa totalment emmotllat a la problemàtica del projecte. S'ha volgut presentar una aplicació senzilla, clara i sobretot útil per tal de facilitar la feina a tot usuari que pugui treure'n un benefici.

Pel que fa a coneixements, aquest projecte m'ha fet possible descobrir tota una branca, o tot un món, d'aplicació de la tecnologia i la informàtica que fins al moment desconeixia. La medicina, és una branca que malauradament encara no fa ús de moltes de les tecnologies que utilitzem al nostre dia a dia i que podria treure'n molt profit. Treballar-hi amb aquest projecte, m'ha fet plantejar la possibilitat de treballar-hi professionalment.

En resum, ha estat un projecte molt interessant, molt positiu i molt entretingut de realitzar. M'ha ajudat a descobrir noves branques professionals a les quals podria encarar la meua trajectòria i també m'ha servit per aprendre a treballar amb noves llibreries, molt útils per a qualsevol tipus de software. En general m'ha servit per obrir-me nous horitzons i noves possibilitats en el futur professional.

8. Possibles Ampliacions

Com ja s'ha explicat, aquest projecte forma part d'una recerca on s'està investigant per poder aplicar la tecnologia del reconeixement d'escriptors i poder-ne extreure conclusions.

Aquest projecte s'ha acotat a realitzar l'aplicació per a visió, l'edició i l'anàlisi dels arxius SVC, però tots aquests arxius s'han capturat amb un software que s'ha desenvolupat dins el mateix marc de recerca, l'aplicació anomenada HandWritingCapture.

És per això que aquest projecte té molt potencial per poder continuar treballant-hi i poder-ne crear noves versions. Per una banda es podrien afegir noves característiques a l'aplicació, com per exemple:

- Mostrar el dibuix de la traça en el paper, però en lloc de posar-hi un color sòlid, poder utilitzar una escala de blanc a negre, representant el blanc la pressió mínima i el negre la pressió màxima. D'aquesta manera es podria tenir una visió més directa i visual dels valors de la pressió
- Es podrien desenvolupar nous efectes per a rectificar possibles errors en la captura de dibuixos, així com d'altres mesures que els investigadors vegin escaients.

Per altra banda una possible fita interessant seria fusionar els dos softwares, el de captura i el visor, editor i analitzador, aconseguint d'aquesta manera un sol software amb totes les funcionalitats necessàries per a completar el cicle de vida de l'arxiu.

Per últim, aprofitant que avui en dia gairebé tothom disposa de smartphones i tauletes a casa, es podria desenvolupar una versió per a mòbil, donant d'aquesta manera les eines perquè la persona que necessita quantificar-se algun dels trastorns detectables amb aquest projecte, pogués realitzar-se un control diari sense haver d'anar a un especialista.

D'aquesta manera a l'hora de fer el control amb l'especialista, el volum de dades seria més gran i es podria tenir una idea més concreta de l'evolució del subjecte.

9. Referències

- [1] E. Sesa, M. Faundez i J. Mekyska, *An Information Analysis of In-Air and On-Surface Trajectories in Online Handwriting*. Cognitive Computation, Volume 4 Number 2, Springer 2009.
- [2] N. Vincent i B. Dorizzi, *A fractal justification of the normalization step for online handwriting recognition*. Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, International Unipen Foundation 2000.
- [3] V. Hotař i A. Hotař, *Surface profile evaluation by fractal dimension and statistic tools using matlab*. Technical University of Liberec.
- [4] B. Kiinkenbergh, *A Review of Methods Used to Determine the Fractal Dimension of Linear Features*. Mathematical Geology, Vol. 26, No. 1., International Association for Mathematical Geology, 1994.
- [5] B. Q.Al-Abudi, *Digit Recognition Using Fractal And Moment Invariants*. Iraqi Journal of Science, Department of Astronomy, College of Science, University of Baghdad.
- [6] H.E. Gutierrez, *Estudio De Geometria Fractal En Roca Fracturada*. Memoria para optar al titulo de ingeniero civil y series de tiempo. Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingenieria Civil, Santiago de Chile – 2008.
- [7] V.L. Blankers, C.E. van den Heuvel, K.Y. Franke i L.G. Vuurpijl, *The ICDAR 2009 Signature Verification Competition*. 2009 10th International Conference on Document Analysis and Recognition.
- [8] Dr. phil. nat. Marcus Liwicki, *ICDAR 2011 Signature Verification Competition*. German Research Center for Artificial Intelligence.
- [9] S. George, R. Kashi, T. Matsumoto i G. Rigoll, *Detailed Instructions for Participants*. SVC 2004: First International Signature Verification Competition.

- [10] D. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto i G. Rigoll, *SVC 2004: First International Signature Verification Competition*.
- [11] S. George, R. Kashi, T. Matsumoto i G. Rigoll, <http://www.cse.ust.hk/svc2004/>.
- [12] www.stackoverflow.com/, Question and answer site for professional programmers.
- [13] <http://docs.oracle.com/javase/8/docs/api/>, Java™ Platform, Standard Edition 8 API Specification.
- [14] www.java.com, informació sobre Java.
- [15] <http://www.jfree.org/jfreechart/>, JFreeChart.
- [16] <http://www.javaworld.com/article/2074594/java-security/reports-made-easy-with-jasperreports.html>, tutorial iniciació a reporting amb JasperReports.
- [17] <http://community.jaspersoft.com/project/jasperreports-library>, tutorial i informació sobre JasperReports.
- [18] <https://docs.oracle.com/javase/tutorial/2d/overview/text.html>, tutorial Java 2D.
- [19] <http://commons.apache.org/proper/commons-beanutils/>, documentació Beanutils.
- [20] <https://commons.apache.org/proper/commons-digester/>, documentació Digester.
- [21] <http://commons.apache.org/proper/commons-logging/>, documentació Logging.
- [22] <https://commons.apache.org/proper/commons-collections/>, docs collections.
- [23] <http://www.groovy-lang.org/>, documentació Groovy.
- [24] <http://itextpdf.com/>, documentació itext.
- [25] <http://www.jfree.org/jcommon/>, documentació jcommon.
- [26] <http://www.infosatsoluciones.es/tarifas.html>, quadre de preus per hora.
- [27] http://dugi-doc.udg.edu/bitstream/handle/10256/1381/7_Pressupost.pdf?sequence=7, tarifa concreta de consultoria.

Annex A. Estudi Econòmic

A.1. Cost del prototip

En aquest document es mostren les despeses econòmiques relacionades amb aquest projecte.

En l'estudi econòmic es consideren el següent factors:

- Costos de recursos humans.
- Amortització equips i software.
- Despeses indirectes.

A partir d'aquests factors basats en el que ha estat el desenvolupament real del projecte, es calcula el cost total d'aquest.

A.1.1. Costos de recursos humans

<u>Concepte</u>	<u>Hores</u>	<u>Preu/hora (€)</u>	<u>Total (€)</u>
Anàlisi i Disseny (Enginyer)	382,2	60	22.932
Implementació (Tècnic Programador)	109,2	30	3.276
Test (Tècnic Programador)	54,6	30	1.638
Redacció memòria (Administratiu)	104	25	2.600
TOTAL RECURSOS HUMANS			30.446

Taula A.1. Preus dels recursos humans utilitzats.

Els preus hores de cada perfil professional, s'han obtingut fent la mitja de diferents exemples que s'han trobat en empreses amb projectes amb un perfil semblant [26 - 27].

A.1.2. Amortització equips i software

<u>Equip utilitzat</u>	<u>Total (€)</u>
<u>Equips i programari informàtic</u>	
Software Sybase PowerDesigner	2.440
Microsoft Office	99
Ordinador Portàtil	800
TOTAL EQUIP	3.339

Taula A.2. Preus del Software i els equips utilitzats.

Comptant que es tracta d'un projecte tecnològic i que cal estar el dia, es fa una estimació de durada de 4 anys pel hardware i de 3 anys pel software. Un cop passat aquest període, caldrà canviar el material i el programes per versions més actuals. Tenint en compte que el projecte ha tingut una durada de aproximadament 6 mesos.

<u>Amortitzacions</u>	<u>Total (€)</u>
Amortització software	423,17
Amortització Hardware	100,00
TOTAL AMORTITZACIONS(€)	523,17

Taula A.3. Totals de les Amortitzacions.

A.1.3. Despeses indirectes

Les despeses indirectes d'aquest projecte, seran aquelles despeses provinents de l'espai utilitzat per a desenvolupar el projecte i els recursos que aquest ofereix.

Quant a l'espai, no es necessita un lloc específic, ja que les instal·lacions necessàries, són molt comunes a qualsevol local. Aquestes instal·lacions seran: Llum, aigua i internet. En aquest cas l'espai utilitzat ha estat una casa particular i els costos són els següents:

<u>Instal·lacions utilitzades</u>	<u>Total (€)</u>
Internet	120
Llum i aigua	100
TOTAL DESPESES INDIRECTES	220

Taula A.4. Despeses directes del projecte.

Aquests costos són una estimació respecte el que es consumeix en una vivenda particular, considerant 6 mesos d'Internet a un cost de 20€ mensuals i una imputació de 100 euros per els 6 mesos d'aigua, llum i altres despeses relacionades amb les instal·lacions.

A.1.4. Cost total del projecte

<u>Concepte</u>	<u>Total (€)</u>
Costos de recursos humans	30.446,00
Costos d'amortització	523,17
Despeses indirectes	220,00
TOTAL	31.189,17

Taula A.5. Costos totals del projecte.

A.2 Preu de venda en el mercat

Quant a la venda del software creat en aquest projecte, es tindran en compte dues coses, per una banda el preu de venda del propi software i per altre banda el preu del manteniment mensual.

Aquest software, s'ha estimat que podrà ser utilitzat en qualsevol centre de recerca o hospital relacionat amb el diagnòstic i/o control de malalties neurodegeneratives. En estar desenvolupat totalment en anglès, la llengua no serà un inconvenient perquè també pugui ser utilitzat fora del propi país.

Tot i aquestes condicions, s'ha volgut fer una estimació realista del nombre de vendes, que serien 100 unitats. A partir d'aquestes cent unitats, es calcula el següent preu de venda i manteniment.

<u>Concepte</u>	<u>Total (€)</u>
Costos total del projecte	31.189,17
Marge comercial (40%)	12.475,67
Subtotal	43.664,84
TOTAL Preu Unitari (100 unitats)	436,65

Taula A.6. Preu unitari del software.

Amb aquest preu, s'inclou tota l'aplicació en la versió actual. Afegit a aquest preu, hi ha la quota de manteniment, l'objectiu de la qual per la banda del client, és tenir assistència en cas de problemes / errors i dret a futures actualitzacions de l'aplicació. Per la banda del desenvolupador, l'objectiu de la quota de manteniment és tenir un ingrés constant per tal de poder seguir treballant en l'aplicació, poder ampliar-la, millorar-la, etc.

El preu de la quota de manteniment serà de 20€al mes, per cada client. D'aquesta manera:

$$20€ * 12 \text{ mesos} * 100 \text{ clients} = 24.000 \text{ €/any} \quad (\text{A.1})$$

Annex B. Diagrames

B.1. Diagrama de Classes de la capa Application

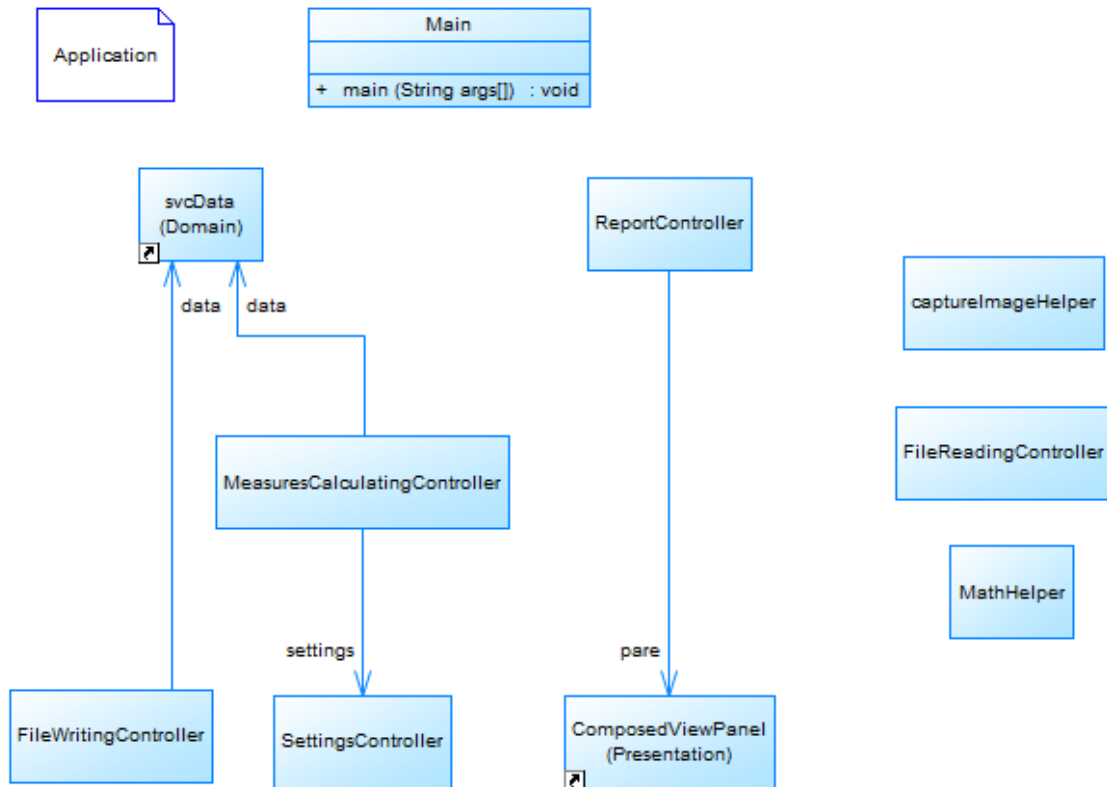


Fig. B.1. Diagrama de Classes de la capa Application.

B.2. Diagrama de Classes de la capa Domain

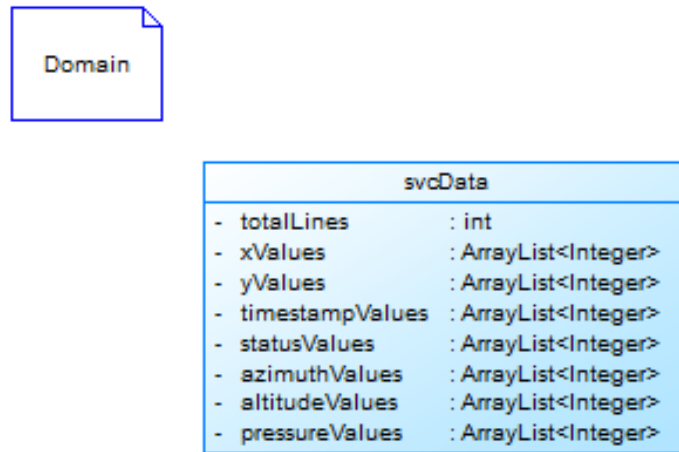


Fig. B.2. Diagrama de Classes de la capa Domain.

B.3. Diagrama de Classes de la capa Persistence

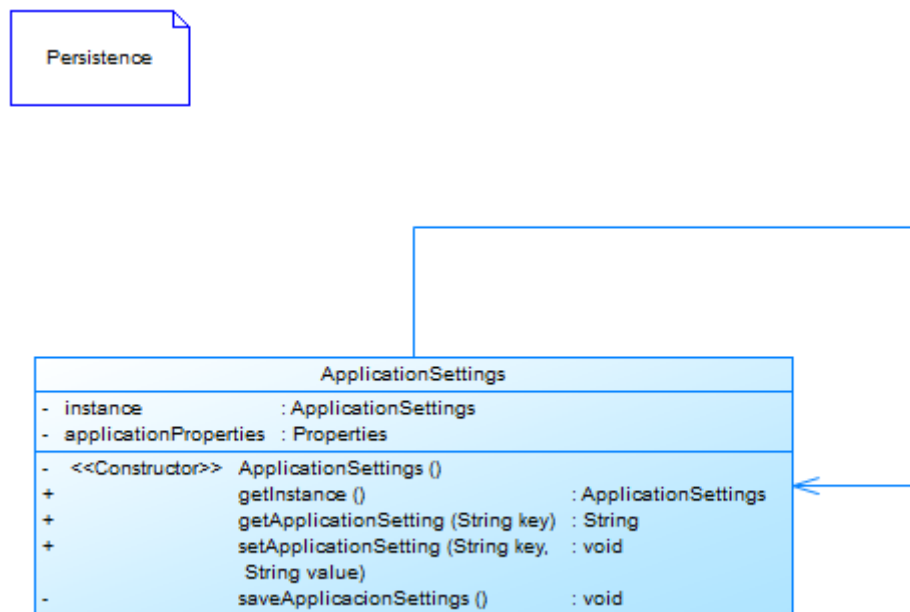


Fig. B.3. Diagrama de Classes de la capa Persistence.

B.4. Diagrama de Classes de la capa Presentation

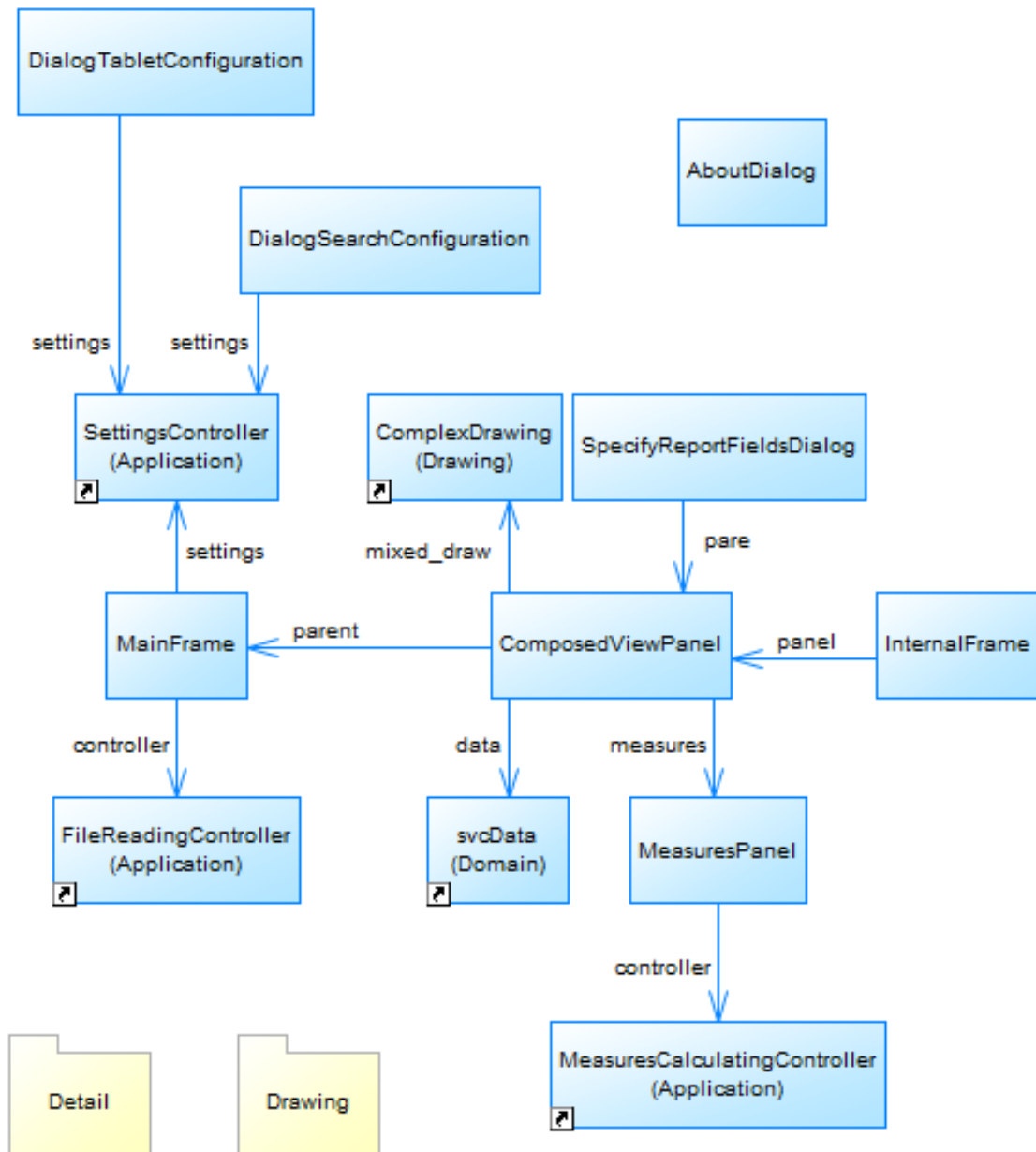


Fig. B.4. Diagrama de Classes de la capa Presentation.

B.4.1. Diagrama de Classes de la sub-capa Detail

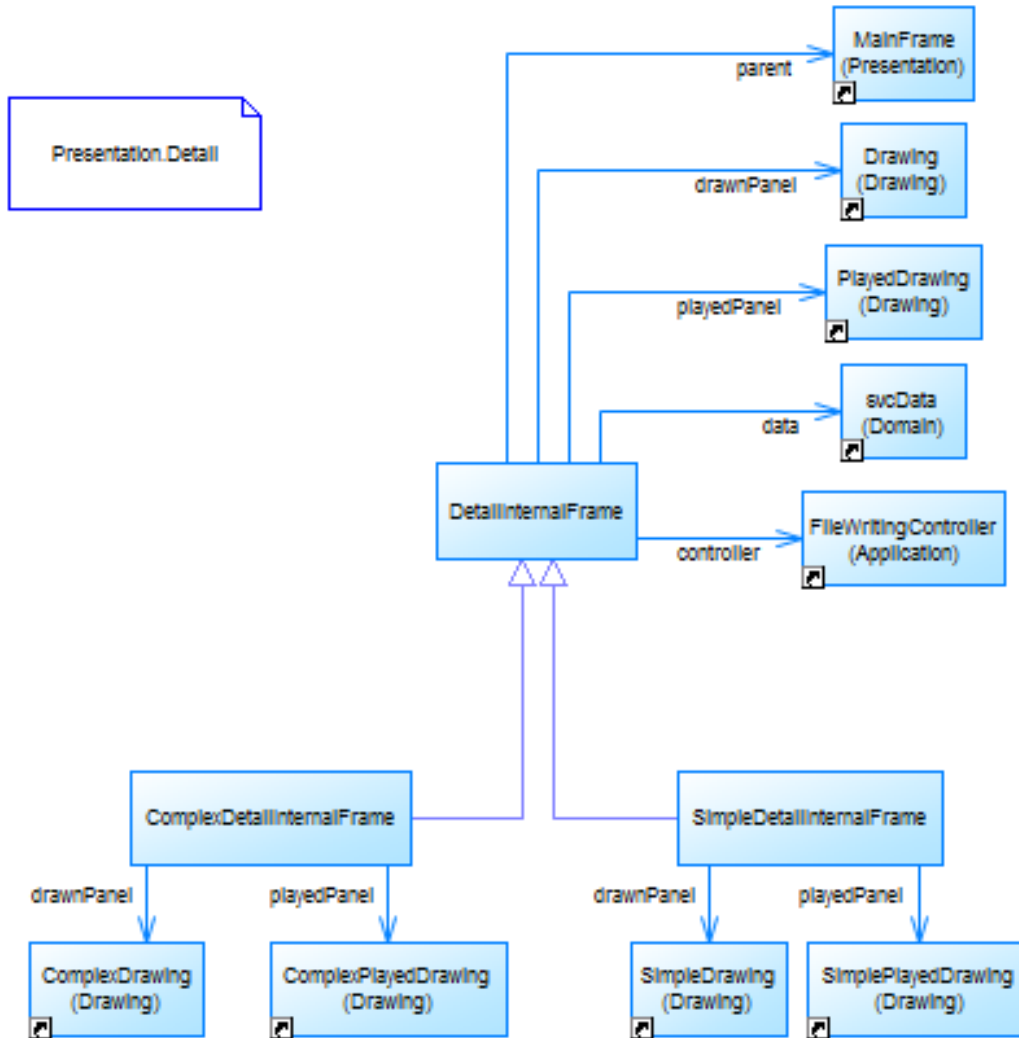


Fig. B.5. Diagrama de Classes de la sub-capa Detail.

B.4.2 Diagrama de Classes de la sub-capa Drawing

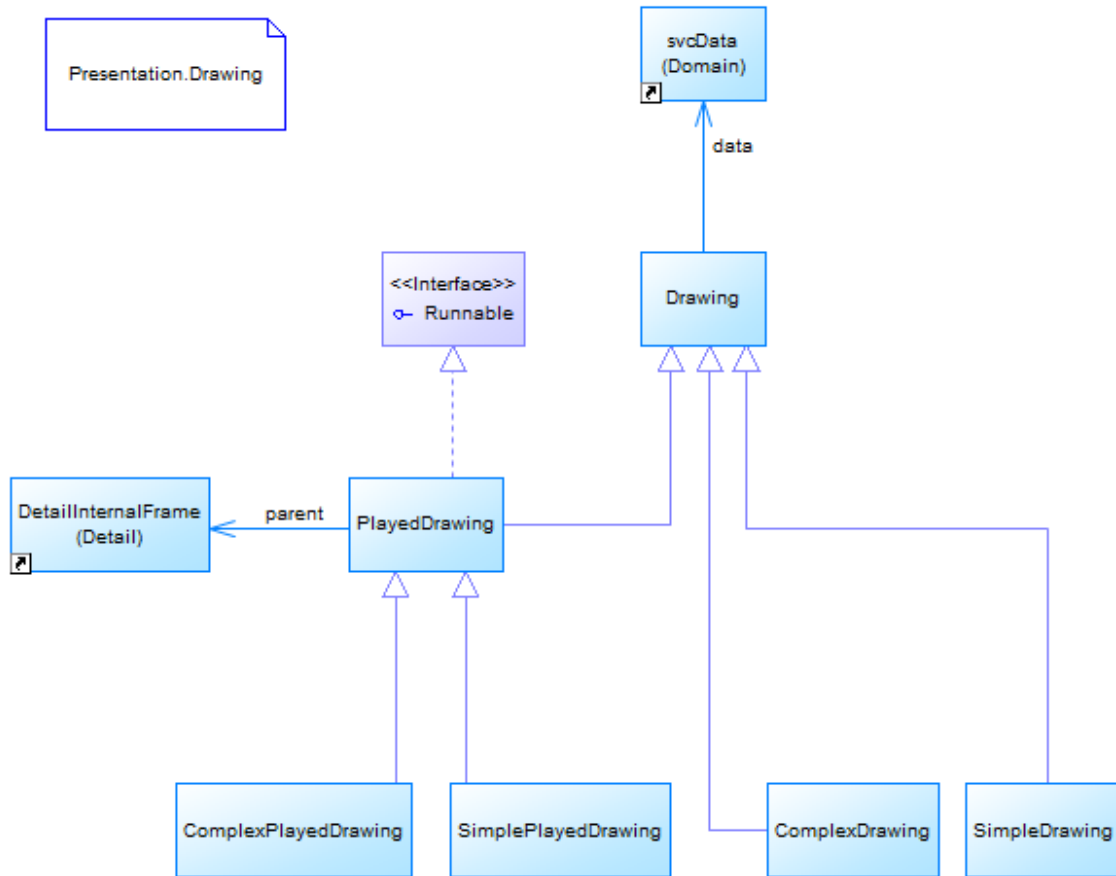


Fig. B.6. Diagrama de Classes de la sub-capa Drawing.

Annex C. Càlcul de rangs de valors

Aquest annex està dedicat a identificar els rangs de valors que pot retornar una tauleta, o més concretament la tauleta que s'ha utilitzat per realitzar el projecte. Aquesta identificació s'ha portat a terme per a poder realitzar un tractament més acurat o més real de les dades que retorna la tauleta.

Per poder calcular aquests rangs, el que s'ha fet és realitzar mostres d'arxius accentuant la magnitud que es vol analitzar. Per exemple, si es vol mirar la pressió, es fa una mostra on es passa de no pressionar gens, a pressionar al *màxim* el bolígraf, per tal de poder veure el rang de valors que retorna el model de tauleta utilitzat.

C.1. Rang de valors de la pressió

Mostra u00001s00002_hw00002

La mostra comença a dalt a l'esquerra amb una pressió mínima i es va augmentant fins al final de la línia a baix a l'esquerra.

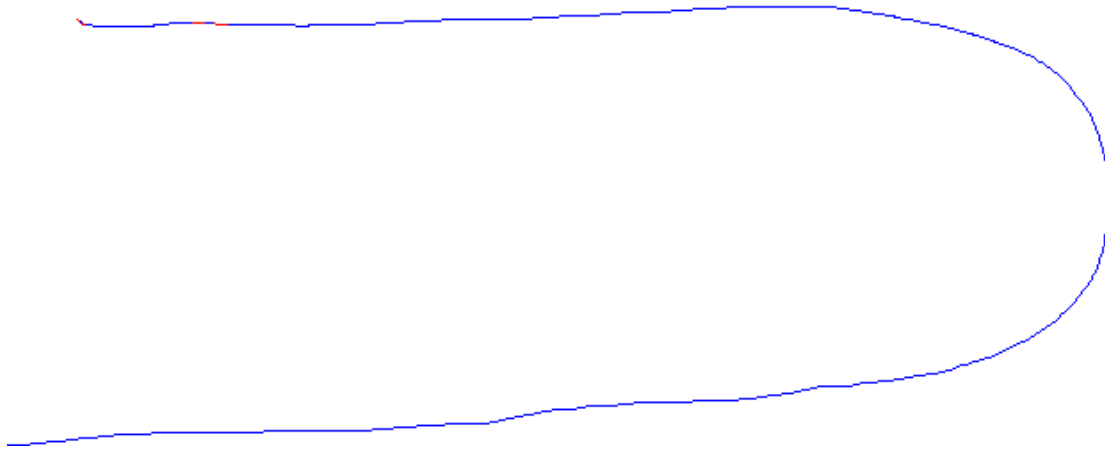


Fig. C.1. Dibuix de la mostra u00001s00002_hw00002.svc

Pressió Míxima	1
Pressió Míxima	1023

Taula C.1. Pressions de la mostra u00001s00002_hw00002.svc

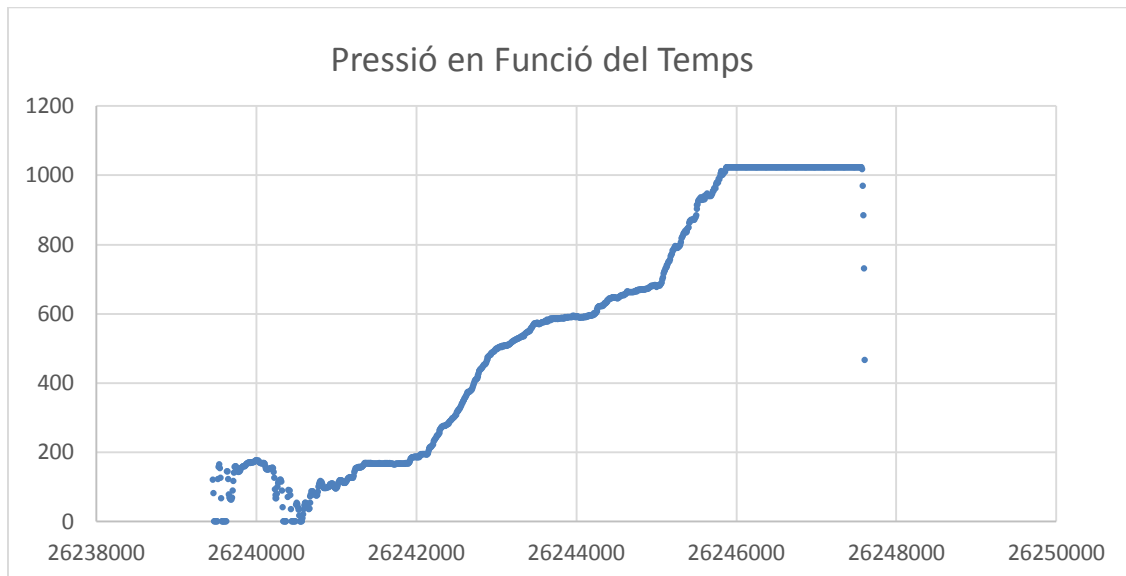


Fig. C.2. Gràfic de la pressió en funció del temps de la mostra

Calibratge de les pressions en funció del bolígraf utilitzat

S'han fet dos dibuixos iguals, anant de la mínima pressió a la màxima. Al primer s'ha fet amb el bolígraf que porta de sèrie la tauleta Wacom Intuos5 i el segon s'ha fet amb un bolígraf Wacom intuos 4, però el model Cintiq, que permet utilitzar una punta de tinta de bolígraf normal. Com es pot observar en el gràfic, els rangs de pressió són exactament iguals.

Mostra u00004s00001_hw00002

Pressió Míxima	1
Pressió Míxima	1023

Taula C.2. Pressions de la mostra u00004s00001_hw00002.svc

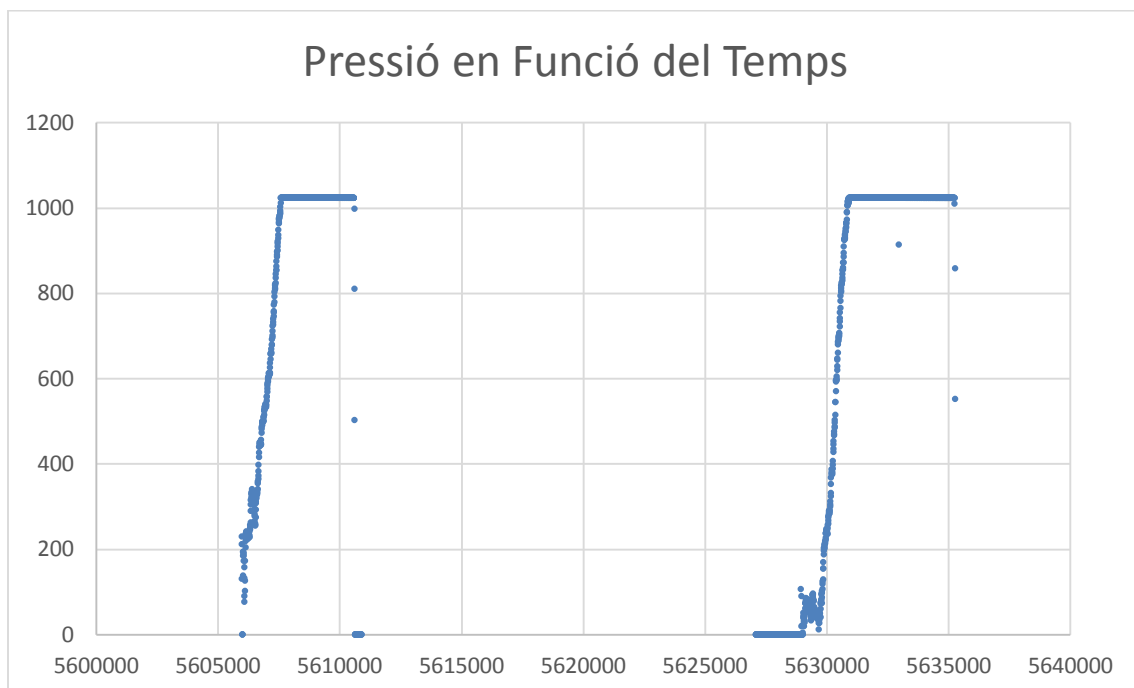


Fig. C.3. Gràfic de la pressió en funció del temps de la mostra

Anàlisi d'altres mostres realitzades amb altres tauletes

Entre les mostres que s'han analitzat, la que es mostra, és un exemple de que la pressió depèn de la tauleta i el bolígraf que s'utilitzi.

Mostra u00001s00001_hw00003

Pressió Míxima	1
Pressió Màxima	2048

Taula C.3. Pressions de la mostra u00001s00001_hw00003.svc

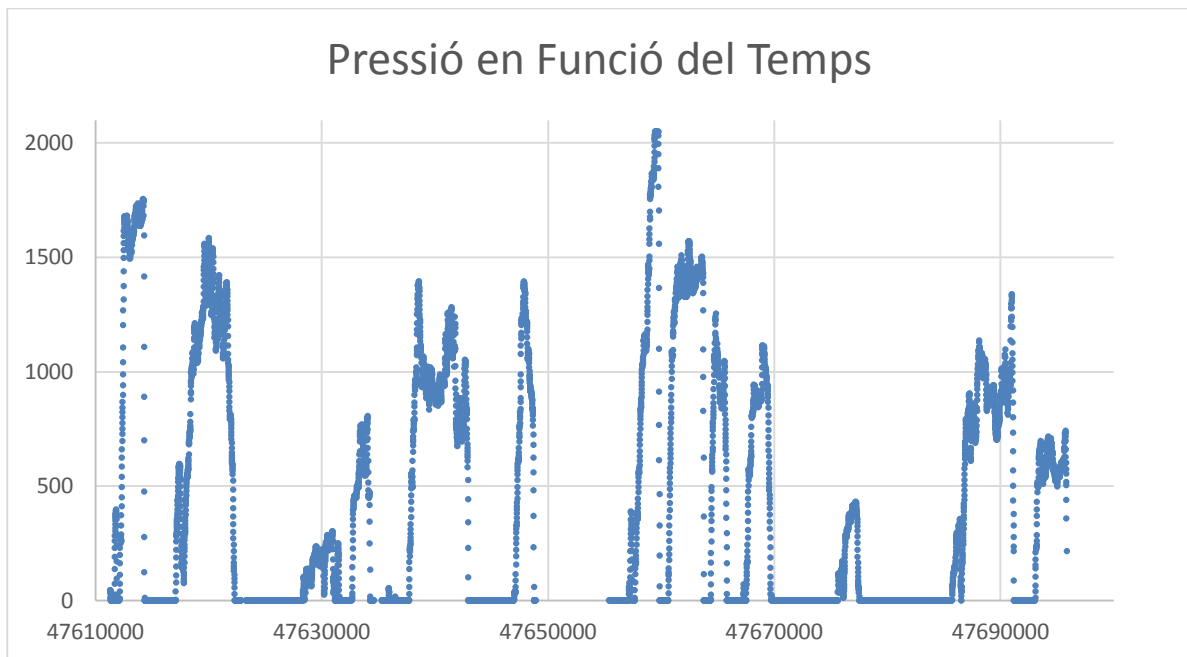


Fig. C.4. Gràfic de la pressió en funció del temps de la mostra

C.2. Rang de valors de l'angle altitud

Mostra u00001s00002_hw00004

La mostra comença a l'esquerra amb el bolígraf al màxim de tombat cap a l'esquerra i a mida que s'avança per el segment es va posant dret fins que arriba al centre. A partir d'aquí el bolígraf es va tombant cap a la dreta.



Fig. C.5. Representació de com s'ha realitzat la mostra u00001s00002_hw00004.svc



Fig. C.6. Dibuix de la mostra u00001s00002_hw00004.svc

Angle Mínim	300
Angle Màxim	890

Taula C.4. Angles de la mostra u00001s00002_hw00004.svc

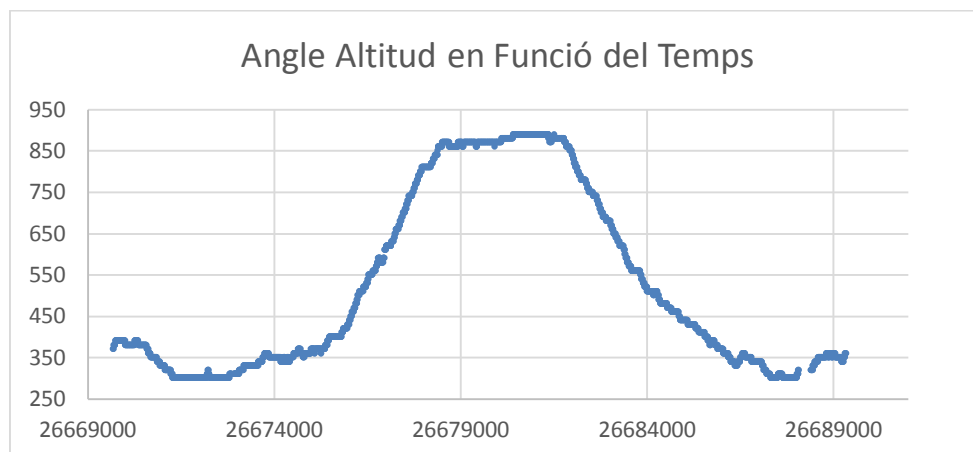


Fig. C.7. Gràfic de l'angle altitud en funció del temps de la mostra

Mostra u00004s00001_hw00001

A partir del gràfic següent, es pot veure com en el punt on l'angle té el valor de 300 pertany a l'interval de Timestamp 5300000 – 5305000 i mirant la pressió es pot veure com no hi havia pressió, això vol dir que és el moment en que el bolígraf estava totalment horitzontal, i per tant la punta no podia tocar la tauleta.

En els punts següents fins al final del dibuix el que s'ha intentat és tenir el bolígraf en un angle de 30° aproximadament i es pot veure que els valors de l'angle Altitud estan al voltant del 320.

Si es mira l'interval anterior 5290000, es pot veure com els valors oscil·len al voltant del 890. Aquest valor pertany a la posició totalment vertical (aproximadament) del bolígraf.

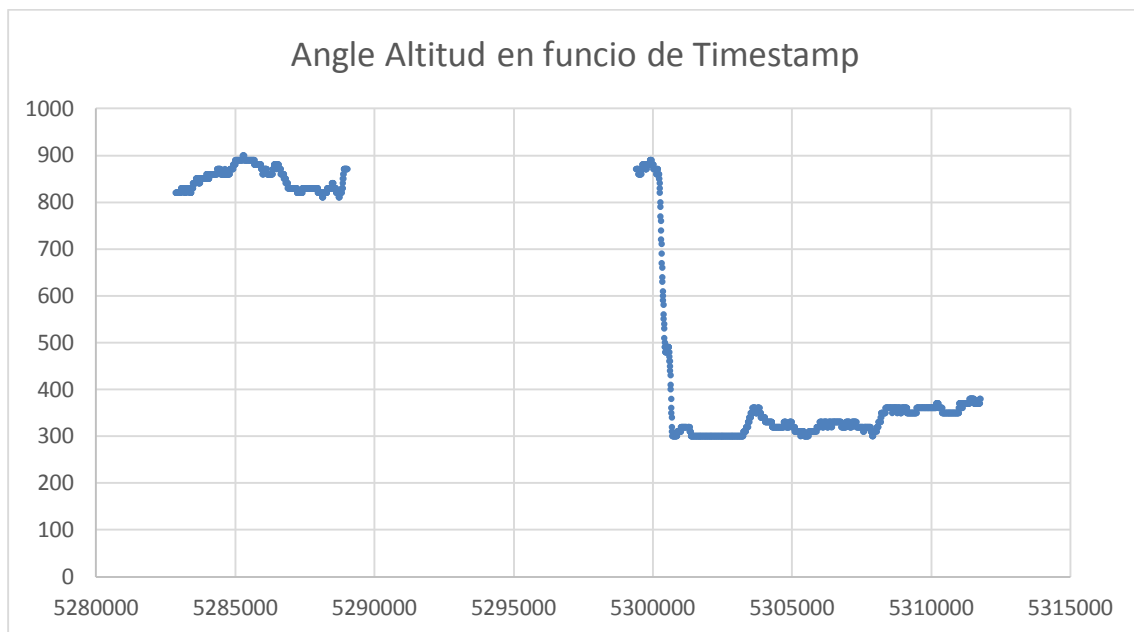


Fig. C.8. Gràfic de l'angle altitud en funció del temps de la mostra

C.3. Rang de valors de l'angle azimut

Mostra u00001s00002_hw00005

La mostra comença a l'equivalent de les 12:00 en un rellotge i fa tota una volta en sentit horari sempre amb el mateix angle d'altitud mirant cap a l'exterior de la circumferència

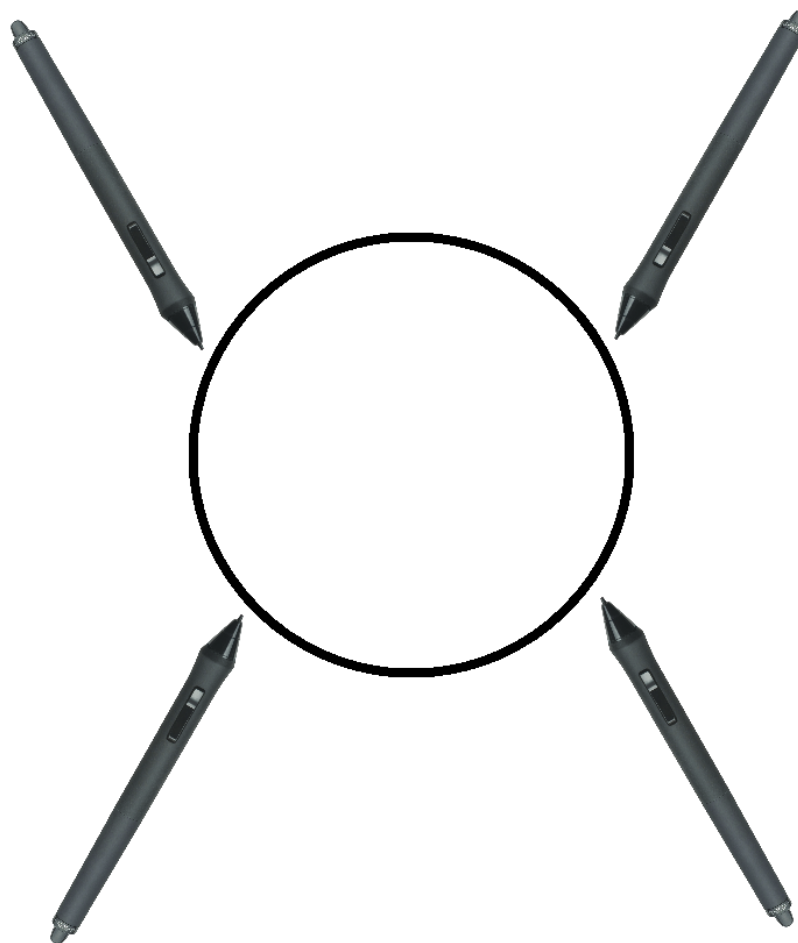


Fig. C.9. Representació de com s'ha realitzat la mostra u00001s00002_hw00005.svc

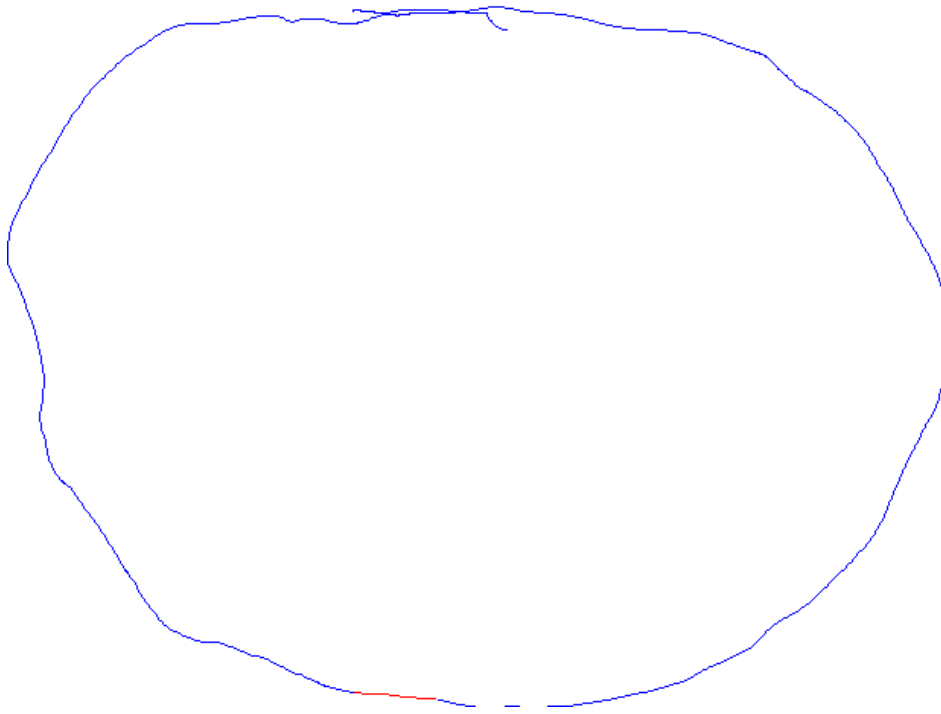


Fig. C.10. Dibuix de la mostra u00001s00002_hw00005.svc

Angle Mínim	0
Angle Màxim	3590

Taula C.5. Angles de la mostra u00001s00002_hw00005.svc

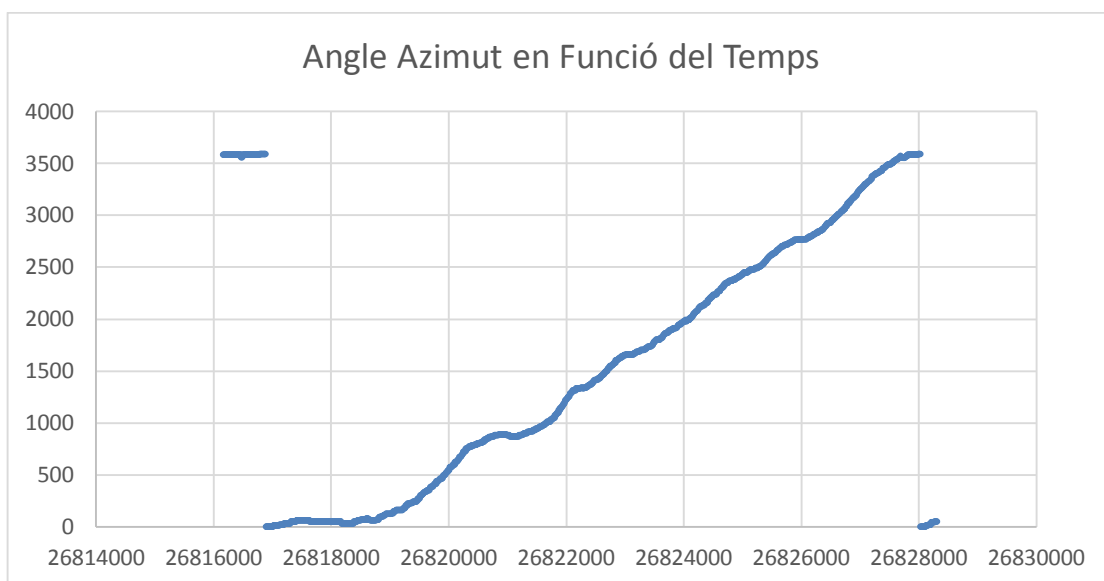


Fig. C.11. Gràfic de l'angle azimut en funció del temps de la mostra

Mostra u00001s00002_hw00006

La mostra segueix el mateix patró que l'anterior però fent dues voltes senceres.

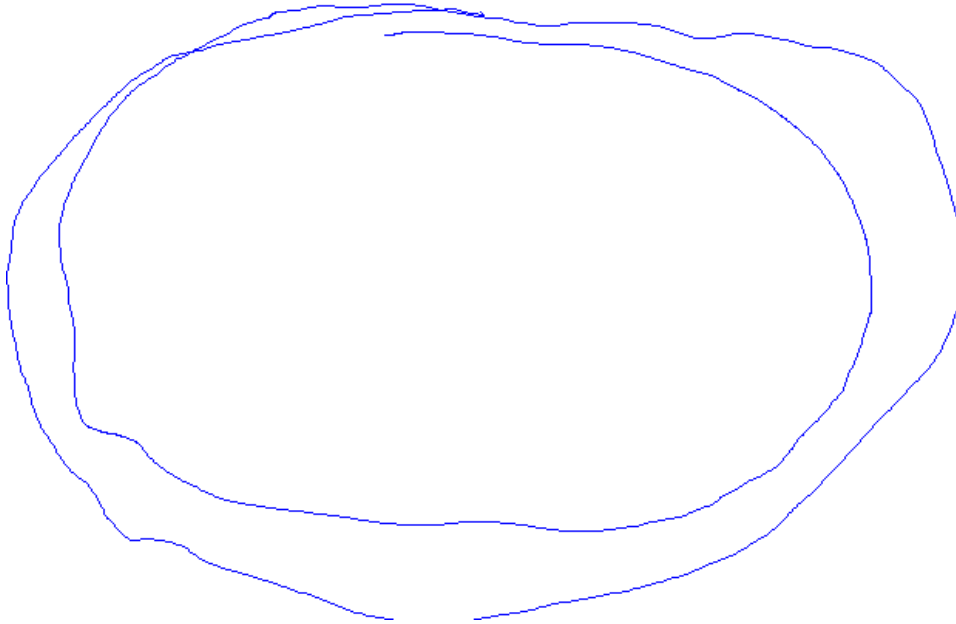


Fig. C.12. Dibuix de la mostra u00001s00002_hw00006.svc

Angle Mínim	0
Angle Màxim	3590

Taula C.6. Angles de la mostra u00001s00002_hw00006.svc

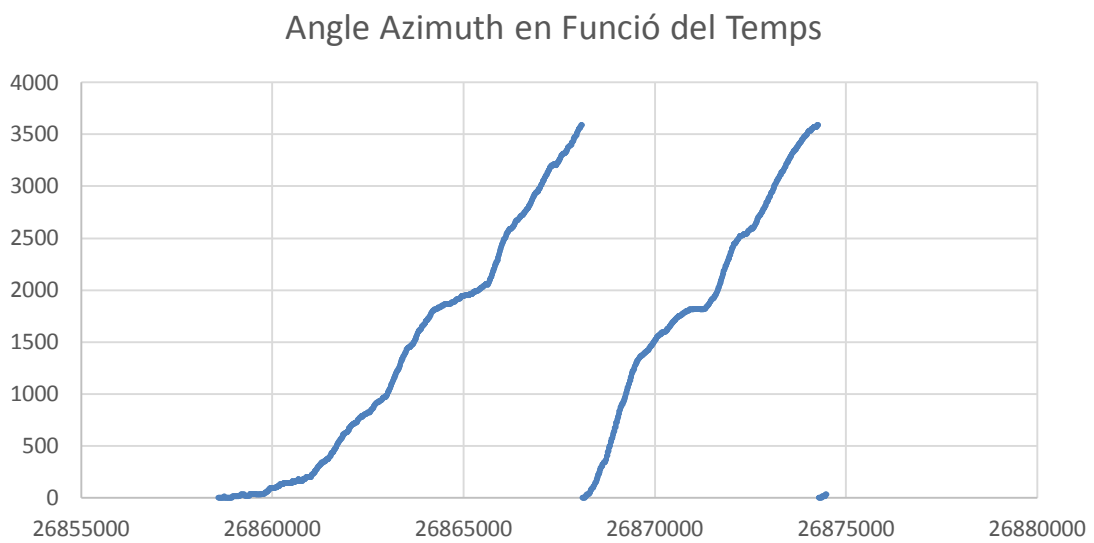


Fig. C.13. Gràfic de l'angle azimuth en funció del temps de la mostra

C.4. Distància màxima de cobertura del bolígraf

S'ha intentat fer una línia recta que va des de la distància zero amb la tauleta fins a la distància màxima, quan deixa de reconèixer el bolígraf. S'ha mesurat amb un regle mentre s'anava avançant en la recta. El resultat ha estat 1,5 cm.

Mostra u00004s00001_hw00003

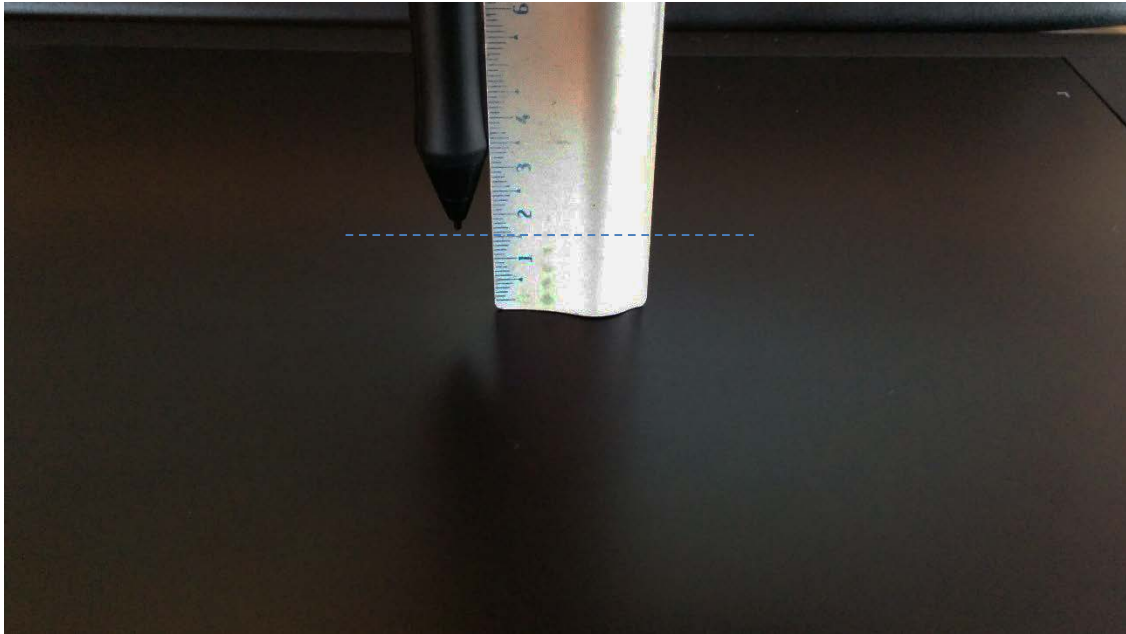


Fig. C.14. Representació de com s'ha calculat la distància.

C.5. Rang de valors de l'estat del bolígraf

En un principi es pensava que la columna Estat, només alternava dos valors 0 i 1, però degut a unes mostres realitzades durant el projecte, es va veure que apareixien nous valors. Aquest nous valors es donen a partir de la orientació del bolígraf (punta o goma) o de si es pitja algun botó (si en té) i de la combinació d'aquests.

Per tal d'identificar els diferents casos el que s'ha fet és crear una mostra dibuixant una línia recta, i en cada un dels casos, provocant un estat o valor diferent de la columna Estat.

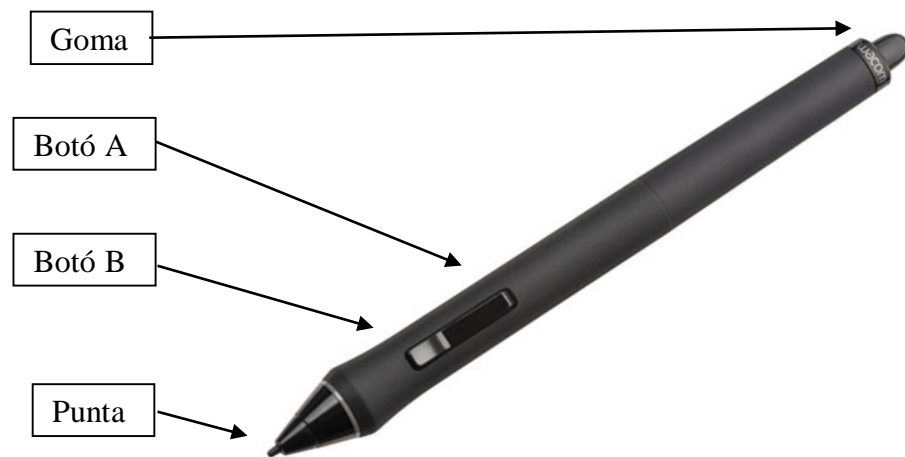


Fig. C.15. Representació de la nomenclatura dels botons.

Els valors han estat els següents:

Estat del Bolígraf	Valor Estat	Arxiu de la mostra
Dibuixant “en-el-paper” amb la punta sense cap botó	1	u00004s00003_hw00001.svc
Dibuixant “en-l’aire” amb la punta sense cap botó	0	u00004s00004_hw00001.svc
Dibuixant “en-el-paper” amb la punta i prement el botó B	3	u00004s00003_hw00003.svc
Dibuixant “en-l’aire” amb la punta i prement el botó B	2	u00004s00003_hw00004.svc
Dibuixant “en-el-paper” amb la punta i prement el botó A	5	u00004s00003_hw00005.svc
Dibuixant “en-l’aire” amb la punta i prement el botó A	4	u00004s00003_hw00006.svc
Dibuixant “en-el-paper” amb la punta i prement els dos botons	7	u00004s00003_hw00007.svc
Dibuixant “en-l’aire” amb la punta i prement els dos botons	6	u00004s00003_hw00008.svc
Dibuixant “en-el-paper” amb la goma sense cap botó	1	u00004s00003_hw00009.svc
Dibuixant “en-l’aire” amb la goma sense cap botó	0	u00004s00004_hw00002.svc
Dibuixant “en-el-paper” amb la goma i prement el botó B	3	u00004s00003_hw00011.svc
Dibuixant “en-l’aire” amb la goma i prement el botó B	2	u00004s00003_hw00012.svc
Dibuixant “en-el-paper” amb la goma i prement el botó A	5	u00004s00003_hw00013.svc
Dibuixant “en-l’aire” amb la goma i prement el botó A	4	u00004s00003_hw00014.svc
Dibuixant “en-el-paper” amb la goma i prement els dos botons	7	u00004s00003_hw00015.svc
Dibuixant “en-l’aire” amb la goma i prement els dos botons	6	u00004s00003_hw00016.svc

Taula C.7. Possibles valors d'estat i la seva causa

Com es pot observar en la columna Estat no es diferencia si es dibuixa amb la punta o amb la goma, però sí que es veu reflectit en la columna Angle Altitud, on en el cas de dibuixar amb la punta, l'angle serà positiu i en cas de dibuixar amb la goma l'angle serà negatiu.

Annex D. Contingut del DVD

- Documentació del projecte.
- Codi font de l'aplicació amb el projecte en Eclipse.
- Mostres anomenades dins els Annexos.