Títol: Plataforma per l'anàlisi de *performance* en navegació d'alt nivell
Volum: 1/1

Autor: Xavier Sagré Martínez

Data: 4 de Juny de 2011

Director: Alicia Ageno Pulido
Departament del director: Llenguatges i Sistemes Informàtics (LSI)

Titulació: Enginyeria en Informàtica

Centre: Facultat d'Informàtica de Barcelona (FIB)
Universitat: Universitat Politècnica de Catalunya (UPC)

DADES DEL PROJECTE

Títol del projecte: Plataforma per l'anàlisi de *performance* en navegació d'alt nivell

Nom de l'estudiant: Xavier Sagré Martínez

Titulació: Enginyeria en Informàtica

Crèdits: 37,5

Director: Alicia Ageno Pulido

Departament: Llenguatges i Sistemes Informàtics (LSI)

MEMBRES DEL TRIBUNAL (nom i signatura)

President: Horacio Rodríguez Hontoria

Vocal: Marta Jimenez Castells

Secretari: Alicia Ageno Pulido

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

A tu pare:

Et duc al cor. La vida que compartires amb mi modela l'home que sóc.

A tu mare:

Pou d'amabilitat i esperança. Dónes la força i mostres el rumb als teus fills.

A tu germana:

Persevera i mira sempre endavant perquè ens queda molt per compartir.

La navegació en aquesta última regata per assolir el títol d'enginyer informàtic ha esdevingut un autèntic repte. He afrontat les implacables condicions meteorològiques que dominaven la mar i per fi em trobo davant la línia d'arribada. Durant aquest temps moltes persones han pujat a bord del veler per fer-me companyia en la llarga travessia. A tots vosaltres, vull donar-vos les gràcies per ensenyar-me, recolzar-me, per compartir o simplement per gaudir al meu costat.

En primer lloc agrair a la meva directora de projecte Alicia Ageno Pulido la seva orientació sincera, paciència i comprensió. La seva col·laboració ha estat indispensable per tirar endavant aquest projecte que pertany a un món que fins ara m'era absolutament desconegut.

En segon lloc agrair als meus pares i la meva germana el seu suport constant i infatigable. Gràcies per ajudar-me amb el timó de la nau en les ocasions que podria haver perdut el rumb.

De la mateixa manera, vull agrair al meu amic Oscar la seva entrega desinteressada i les seves aportacions. M'has ajudat a mantenir el veler en un estat impecable, i ara puc dir-te que respon igual que el primer dia.

Finalment, gràcies a tots els que penseu en mi i somrieu recordant els bons moments que passem junts, aquesta victòria no és només meva també és vostra.

Índex

1 Introducció ... 13
1.1 Motivació del projecte ... 13
1.2 Situació actual ... 13
1.3 Abast ... 14
1.4 Objectius ... 15
1.4.1 Objectiu general ... 15
1.4.2 Objectius específics ... 15
1.5 Visió global del sistema ... 16
1.5.1 Usuaris del sistema ... 17
1.5.2 Arquitectura tècnica ... 17
1.5.3 Funcionalitats del sistema ... 18
1.5.4 Interfícies ... 19
2 Context del projecte ... 19
2.1 Descripció del projecte ... 20
2.1.1 Els vaixells de regata ... 20
2.1.2 Les regates ... 24
2.1.3 Sensors i recol·lecció d'informació ... 28
2.2 Estudi de mercat ... 31
2.2.1 Anàlisi de l'oferta ... 31
3 Planificació i pressupost econòmic ... 40
3.1 Fases del projecte ... 40
3.2 Pressupost del projecte ... 41
3.3 Planificació temporal del projecte ... 43
3.3.1 Planificació inicial ... 44
3.3.2 Planificació final ... 45
4 Anàlisi de Requeriments ... 46
4.1 Requeriments de client ... 46
4.1.1 Extensibilitat ... 46
4.1.2 Interfície d'usuari ... 46
4.1.3 Propietat de les tecnologies ... 46
4.1.4 Llenguatge de programació ... 46
4.1.5 Emmagatzematge persistent de la informació ... 46
4.2 Requeriments funcionals ... 47
4.2.1 Importació de fitxers de *Log* amb informació de regates ... 47
4.2.2 Conversió de dades durant la importació ... 47
4.2.3 Preparació i tractament de les dades importades ... 47
4.2.4 Gravació persistent de les dades llegides dels fitxers de *Log* ... 48
4.2.5 Inserció, consulta i esborrat de veles ... 48
4.2.6 Manteniment del temps d'ús de les veles ... 48
4.2.7 Consulta i esborrat de regates importades ... 48
4.2.8 Diari del sistema ... 48
4.2.9 Gràfiques d'anàlisi ... 48
4.2.10 Filtres ... 49
4.2.11 Creació i esborrat de la taula de regates ... 49
4.2.12 Creació i esborrat de la taula de veles ... 49
4.2.13 Fitxer bàsic de definició de la taula de regates de la base de dades ... 50
4.2.14 Incorporació de fitxers de *targets* pels vaixells tractats en el sistema ... 50

4.2.15 Manual d'usuari del sistema 50
4.3 Requeriments no funcionals 50
4.3.1 Interfície d'usuari i factors humans 50
4.3.2 Documentació 51
4.3.3 Característiques de rendiment 51
4.3.4 Tractament d'errors i condicions extremes 51
4.3.5 Interfície del sistema 52
4.3.6 Factors de qualitat 52
4.3.7 Temes de seguretat 53
4.3.8 Entorn de desenvolupament 53
4.3.9 Consistència 54
5 Especificació del sistema 55
5.1 Diagrama de casos d'ús 55
5.2 Especificació de casos d'ús 57
5.2.1 Especificació del cas d'ús "Importar regata" 57
5.2.2 Especificació del cas d'ús "Inserir noves columnes" 59
5.2.3 Especificació del cas d'ús "Filtre de qualitat" 61
5.2.4 Especificació del cas d'ús "Visualitzar regates" 62
5.2.5 Especificació del cas d'ús "Esborrar regates" 63
5.2.6 Especificació del cas d'ús "Inserir veles" 64
5.2.7 Especificació del cas d'ús "Visualitzar veles" 65
5.2.8 Especificació del cas d'ús "Esborrar veles" 66
5.2.9 Especificació del cas d'ús "Gràfica TWA vs Time" 68
5.2.10 Especificació del cas d'ús "Gràfiques X vs múltipes Y" 70
5.2.11 Especificació del cas d'ús "Filtre TW_speed" 73
5.2.12 Especificació del cas d'ús "Filtre TW_speed amb babord/estribord" 74
5.2.13 Especificació del cas d'ús "Filtre genèric babord/estribord" 75
5.2.14 Especificació del cas d'ús "Filtre genèric" 76
5.2.15 Especificació del cas d'ús "Diari del sistema" 77
5.2.16 Especificació del cas d'ús "Crear taula de regates" 77
5.2.17 Especificació del cas d'ús "Esborrar taula de regates" 78
5.2.18 Especificació del cas d'ús "Crear taula de veles" 80
5.2.19 Especificació del cas d'ús "Esborrar taula de veles" 80
5.2.20 Especificació del cas d'ús "Manual d'usuari" 82
5.2.21 Especificació del cas d'ús "Sobre Regaton" 82
5.3 Diagrama de classes 84
5.3.1 Diagrama de classes del domini 84
6 Disseny 86
6.1 Patrons arquitectònics 86
6.1.1 Patró Model – Vista – Controlador 86
6.1.2 Patró Arquitectura per Capes 94
6.2 Patrons de disseny 95
6.2.1 Patró Expert 95
6.2.2 Patró Controlador 97
6.2.3 Patró Observador 97
6.3 Realitzacions dels casos d'ús més importants 99
6.3.1 Realització del cas d'ús "Importar regata" 100
6.3.2 Realització del cas d'ús "Filtre de qualitat" 106
6.3.3 Realització del "Filtre TW_speed" 110
6.4 Disseny de la base de dades 112

7 Implementació 114
7.1.1 Diagrama de classes dels components gràfics 114
7.1.2 Diagrama de classes dels filtres per components *Java Swing* 116
7.1.3 Diagrama de classes dels filtres del sistema 117
7.1.4 Diagrama de classes de les vistes del sistema 119
7.1.5 Diagrama de classes de l'arrencada del sistema 120
7.1.6 Diagrama de classes del patró *Observador* 121
7.1.7 Diagrama de la classe *Controlador* 122
7.1.8 Diagrama de la classe *Acces* 123
7.1.9 Diagrama dels tipus de dades 124
7.1.10 Diagrama de classes amb la resta de classes i les seves relacions 125
7.1.11 Problemes detectats durant el desenvolupament 126
8 Conclusions 128
8.1 Conclusions 128
8.2 Balanç d'assoliment d'objectius 135
8.3 Proposta de millores 136
8.4 Conclusions personals 137
9 Glossari 139
10 Bibliografia 144
11 Apèndix A: Instal·lació 148
11.1 Programari requerit 148
11.2 Instal·lació del *SGBD* 149
11.3 Configuració del *SGBD* 151
11.4 Instal·lació de l'eina *Regaton* 154
11.5 Configuració de l'eina *Regaton* 158
12 Apèndix B: Manual d'usuari 159
12.1 Fitxers de *Log* per importació de regates 159
12.2 Fitxers de Targets 160
12.3 Fitxer bàsic de definició de la taula de regates 161
12.4 Manual d'usuari de la versió actual del sistema 164
13 Apèndix C: Estudi de les tecnologies disponibles 243
13.1 Llenguatges de programació 243
13.1.1 Característiques dels llenguatges de programació actuals més comuns 244
13.1.2 Taules comparatives dels llenguatges de programació estudiats 250
13.1.3 Conclusions de l'estudi 253

Índex de figures

Figura 1: Visió global del sistema 16
Figura 2: Arquitectura tècnica 17
Figura 3: Interfície d'usuari del sistema 19
Figura 4: Components i terminologia dels vaixells de regata 21
Figura 5: Rumbs relatius d'un vaixell 21
Figures 6 i 7: Maniobres de virada i trabujada 22
Figura 8: Gràfica de les polars d'un vaixell de regates 22
Figures 9 i 10: Orientació de les veles respecte del vent amb i sense turbulències 23
Figura 11: Suma de forces que resulten en el impuls vèlic 23
Figura 12: Bastó: Sortida – 1 – 2 – 1 – 2 – Arribada 25
Figura 13: Triangle: Sortida – 1 – 2 – 3 – 1 – 3 - Arribada 26
Figura 14 i 15: Recorregut Trapezoïdal: Sortida – 1 – 2 – 3 – 2 – 3 – Arribada i Recorregut Trapezoïdal: Sortida – 1 – 4 – 1 – 2 – 3 – Arribada 27
Figures 16,17 i 18: Exemples de recorregut costaners 27
Figura 19: Recorregut oceànic de la regata Volvo Ocean Race 2008 - 2009 28
Figura 20: Navegació tàctica 32
Figura 21: Funcionalitat meteorològica i de rutes 32
Figura 22: Recursos i anàlisis 33
Figura 23: Creació de rutes 34
Figura 24: Navegació 34
Figura 25: Ús de Polars 35
Figura 26: Sailect, selecció veles 35
Figura 27: Anàlisi d'un vaixell 36
Figura 28: Anàlisi per zones d'una regata 37
Figura 29: Calculadora tàctica resolent navegació ideal segons les polars del vaixell 37
Figura 30: Llista vaixells detectats 38
Figura 31: Funcionalitats diverses de l'eina GPS Race Driver 39
Figura 32: Planificació inicial del projecte 44
Figura 33: Planificació final del projecte 45
Figura 34 : Diagrama de casos d'ús 56
Figura 35: Diagrama de classes del domini 84
Figura 36: Esquema bàsic del Model – Vista - Controlador 87
Figura 37: Classe Controlador 88
Figura 38: Diagrama de les classes que formen part de la interfície d'usuari 89
Figura 39: Diagrama amb la generalització de les vistes del sistema 90
Figura 40: Diagrama de navegabilitat entre les vistes del sistema 91
Figura 41: Diagrama amb les vistes més importants del sistema 92
Figura 42: Diagrama amb les vistes gràfiques més importants del sistema 93
Figura 43: Arquitectura per capes 94
Figura 44: Exemples d'aplicació del Patró Expert 96
Figura 45: Diagrama del patró Observador 97
Figura 46: Diagrama amb les classes del sistema que usen el patró Observador 98
Figura 47: Realització del cas d'us "Importar regata" 100
Figura 48: Diagrama de seqüència de l'operació "iniciarImportació" 101
Figura 49: Diagrama de seqüència de l'operació "importarRegata" 102
Figura 50: Diagrama de seqüència de l'operació "finalitzarImportació" 103
Figura 51: Diagrama de seqüència de l'operació "inserirFila" 104

Figura 52: Diagrama d'estats del procés d'importació 105
Figura 53: Realització del cas d'ús "Filtre de qualitat" 106
Figura 54: Diagrama de seqüència de l'operació "executarFiltre" 106
Figura 55: Diagrama de seqüència de l'operació "tractarValor" 107
Figura 56: Diagrama de seqüència de l'operació "actualitzarQualitatAnteriorAlCanviDeSigne" 108
Figura 57: Diagrama d'estats de l'execució del filtre de qualitat 109
Figura 58: Realització del cas d'ús "Filtre TW_speed" 110
Figura 59: Diagrama de seqüència de l'operació "executarFiltre" 110
Figura 60: Diagrama de seqüència de l'operació "xyPolar" 111
Figura 61: Disseny de la base de dades del sistema 113
Figura 62: Diagrama d'implementació dels components gràfics 115
Figura 63: Diagrama d'implementació dels filtres per components Java Swing 116
Figura 64: Diagrama d'implementació dels filtres del sistema 117
Figura 65: Diagrama d'implementació de relació entre vistes i filtres del sistema 118
Figura 66: Diagrama d'implementació de la generalització de les vistes del sistema . 119
Figura 67: Diagrama d'implementació de l'arrencada del sistema 120
Figura 68: Diagrama d'implementació de les classes que usen el patró Observador... 121
Figura 69: Diagrama d'implementació de la classe Controlador 122
Figura 70 : Diagrama d'implementació de la classe Acces. 123
Figura 71 : Diagrama d'implementació de les classes dels tipus de dades 124
Figura 72: Diagrama d'implementació de la resta de classes del sistema 125
Figura 73: Gràfica d'anàlisi en funció TWS vs Boatspeed, versió scripts 129
Figura 74: Gràfica d'anàlisi en funció TWS vs Boatspeed, versió Regaton 129
Figura 75: Gràfica d'anàlisi en funció TWS vs Heel, versió scripts 130
Figura 76: Gràfica d'anàlisi en funció TWS vs Heel, versió Regaton 130
Figura 77: Gràfica d'anàlisi en funció TWS vs VMG, versió scripts 131
Figura 78: Gràfica d'anàlisi en funció TWS vs VMG, versió Regaton 131
Figura 79: Gràfica d'anàlisi en funció TWS vs Rudder, versió scripts 132
Figura 80: Gràfica d'anàlisi en funció TWS vs Rudder, versió Regaton 132
Figura 81: Gràfica d'anàlisi en funció TWS vs Boatspeed Down, versió scripts 133
Figura 82: Gràfica d'anàlisi en funció TWS vs Boatspeed Down, versió Regaton 133
Figura 83: Gràfica d'anàlisi en funció TWS vs Forestay, versió scripts 134
Figura 84: Gràfica d'anàlisi en funció TWS vs Forestay, versió Regaton 134
Figura 85: Creació d'una nova base de dades 152
Figura 86: Introducció del nom de la base de dades 152
Figura 87: Creació d'un nou usuari de treball 153
Figura 88: Introducció del nom i contrasenya del nou usuari 153
Figura 89: Alta de privilegis del nou usuari 154
Figura 90: Arbre de directoris del projecte 155
Figura 91: Contingut parcial d'un fitxer de Log 160
Figura 92: Fitxer de targets de la variable VMG 161
Figura 93: Exemple de fitxer de definició de la taula de regates 163

Índex de taules

Taula 1: Variables d'estudi dels fitxers de Log 30
Taula 2: Cost dels recursos humans 42
Taula 3: Cost del maquinari 42
Taula 4: Cost del programari 42
Taula 5: Cost total del projecte 42
Taula 6: Diferències entre llenguatges Turing - complets i dinàmics 244
Taula 7: Comparativa de llenguatges de programació Turing - complets 250
Taula 8: Comparativa de llenguatges dinàmics 252

1 Introducció

El present document correspon a la memòria del projecte "Plataforma per l'anàlisi de *performance* en navegació d'alt nivell", projecte de 37,5 crèdits de la Enginyeria Informàtica de la Facultat d'Informàtica de Barcelona, UPC.

El capítol mostra una pinzellada del concepte que embolcalla el projecte amb les argumentacions que m'han conduit a realitzar-lo.

1.1 Motivació del projecte

L'esperit de competició és el que impregna els vaixells de regata professionals. Acostumen a ser embarcacions sofisticades on el treball en equip adquireix una importància fonamental. A banda dels instruments comuns disponibles a bord com cronòmetres, compassos, prismàtics, etc., s'equipen amb sistemes de navegació sofisticats capaços de recollir dades d'un conjunt de sensors que aporten informació de variables com la velocitat i direcció del vent, l'escora, l'angle del timó, la situació del vaixell, etc.

La recol·lecció d'informació, mitjançant els sensors instal·lats, tant en les regates de competició com en les d'entrenament, es converteix en una eina indispensable per analitzar el rendiment dels vaixells.

Avui en dia, el concepte de rendiment és primordial en un vaixell de regata de competició i està estrictament relacionat amb l'habilitat de navegar a la seva màxima capacitat.

L'anàlisi de la informació obtinguda durant una regata permet conèixer en quines condicions l'ús d'unes o altres veles i de determinats reglatges ofereix els millors resultats per la navegació.

1.2 Situació actual

La vela és poder i saber controlar la dinàmica d'un vaixell que es mou per la força del vent sobre les seves veles. La navegació a vela és un esport tant d'entreteniment com de competició. Les competicions s'anomenen regates.

Els orígens de la navegació a vela no són molt clars. S'han trobat formes d'embarcacions impulsades per una vela en gravats egipcis. Es creu que navegaven pel Nil i el Mar Roig transportant animals i aliments. Els ibers, celtes i grecs van seguir els seus passos en la navegació. Els fenicis, un poble comerciant per excel·lència utilitzava embarcacions per desplaçar-se, mentre que els romans els usaven com armes de guerra.

És en el segle XVII quan als Països Baixos s'inicia la vela com una activat d'oci i com un esport. Carles II, després del seu exili a Holanda, implantarà la navegació a vela a Anglaterra. Però no és fins la meitat del segle XX que esdevindrà popular.

Les regates internacionals comencen el 1851 quan diversos membres del *New York Yacht Club* construïen una goleta de trenta metres d'eslora anomenada Amèrica. El veler guanyava el trofeu *Hundred Guineas* d'Anglaterra, després de participar en una regata al voltant de l'illa de Wright, organitzada pel *Royal Yacht Squadron*. Aquest trofeu posteriorment prendria per nom *Copa Amèrica*, en honor de la goleta, i no com es creu en honor als Estats Units. El trofeu va estar en possessió del *New York Yacht Club* fins el 1983 que passaria a mans d'un yacht Australià.

La vela esdevenia esport olímpic en els Jocs Olímpics de París de l'any 1900.

Avui en dia la navegació d'alt nivell és una competició molt renyida. El disseny dels vaixells es converteix en un fase de gran importància, on es persegueix assolir els millors paràmetres de rendiment possibles. La incorporació de tecnologia a les embarcacions és un fet. L'estratègia i la tàctica, eines imprescindibles per les tripulacions a l'hora d'afrontar les regates.

Mentre que l'estratègia defineix la forma de prendre les decisions a mig i llarg termini, com l'elecció de la ruta a seguir o els canvis realitzats depenent dels vents i els corrents que apareguin, la tàctica persegueix que el vaixell navegui el millor possible i superi els demès velers, mitjançant ajustos i canvis en la configuració de les veles.

La conjunció de tantes disciplines i metodologies tant en l'elaboració dels vaixells, com quan naveguen en les regates és una mostra innegable de l'aparició constant de noves inquietuds, necessitats i fites.

1.3 Abast

El sistema *Regaton* és una solució computacional per l'anàlisi de rendiment en regates d'alt nivell.

Mitjançant el sistema els usuaris poden importar regates realitzades amb un o altre vaixell, actualitzar la seva informació i analitzar les dades a partir de la generació de gràfiques d'una o diverses variables.

Regaton és una eina orientada a facilitar a l'equip de popa[1] de les embarcacions la presa de decisions tàctiques a partir de l'anàlisi de les dades recollides en les regates.

[1] L'equip de popa d'un vaixell de regates el formen el patró, el tàctic i el navegant. A vegades s'afegeix al grup un estrateg.

1.4 Objectius

1.4.1 Objectiu general

El projecte proposa l'anàlisi, disseny i implementació d'una nova aplicació amb un conjunt de funcionalitats (mecanismes) orientades a l'emmagatzematge i tractament de les dades recol·lectades pels sensors, amb l'objectiu que els resultats obtinguts ajudin en la presa de decisions tàctiques de l'equip durant les regates de competició.

El projecte requereix el desenvolupament de diversos processos automàtics, que anomenarem *filtres*, encarregats d'actualitzar les dades en base a diversos criteris, o bé de realitzar certs càlculs estadístics a petició de l'usuari. Gràcies als *filtres* l'usuari podrà analitzar aspectes específics relacionats amb el rendiment dels vaixells de regates.

Concloent, i tenint en compte que es requereix l'ús de programari lliure per la realització del projecte, el llenguatge escollit per a la programació de l'aplicació és *Java*, amb el suport del *SGBD*[2] *MySQL*[3] per a l'emmagatzematge persistent de la informació.

1.4.2 Objectius específics

1. Estudi bàsic dels conceptes nàutics necessaris per comprendre i realitzar la feina assignada.

2. Anàlisi, disseny i implementació d'una aplicació amb un conjunt de funcionalitats que permetin la importació, emmagatzematge, tractament i anàlisi de la informació recol·lectada durant les regates.

3. Creació de diversos *filtres* pel tractament de les dades. Han de ser independents del nucli del sistema.

4. Ús d'eines de programari lliure.

5. Creació d'un manual d'usuari de l'eina.

6. Redacció de la memòria del projecte.

[2] Sistema de gestió de bases de dades. Agrupació de programes que serveixen per definir, construir i manipular una base de dades.
[3] MySQL és un sistema de gestió de bases de dades relacional multi-fil i multiusuari que utilitza el llenguatge SQL.
SQL, de l'anglès Structured Query Language, és un llenguatge estàndard de comunicació amb bases de dades relacionals.

1.5 Visió global del sistema

El següent diagrama mostra de manera general l'estructura del sistema. És l'usuari qui indica les accions a realitzar al sistema mentre interactua amb ell.

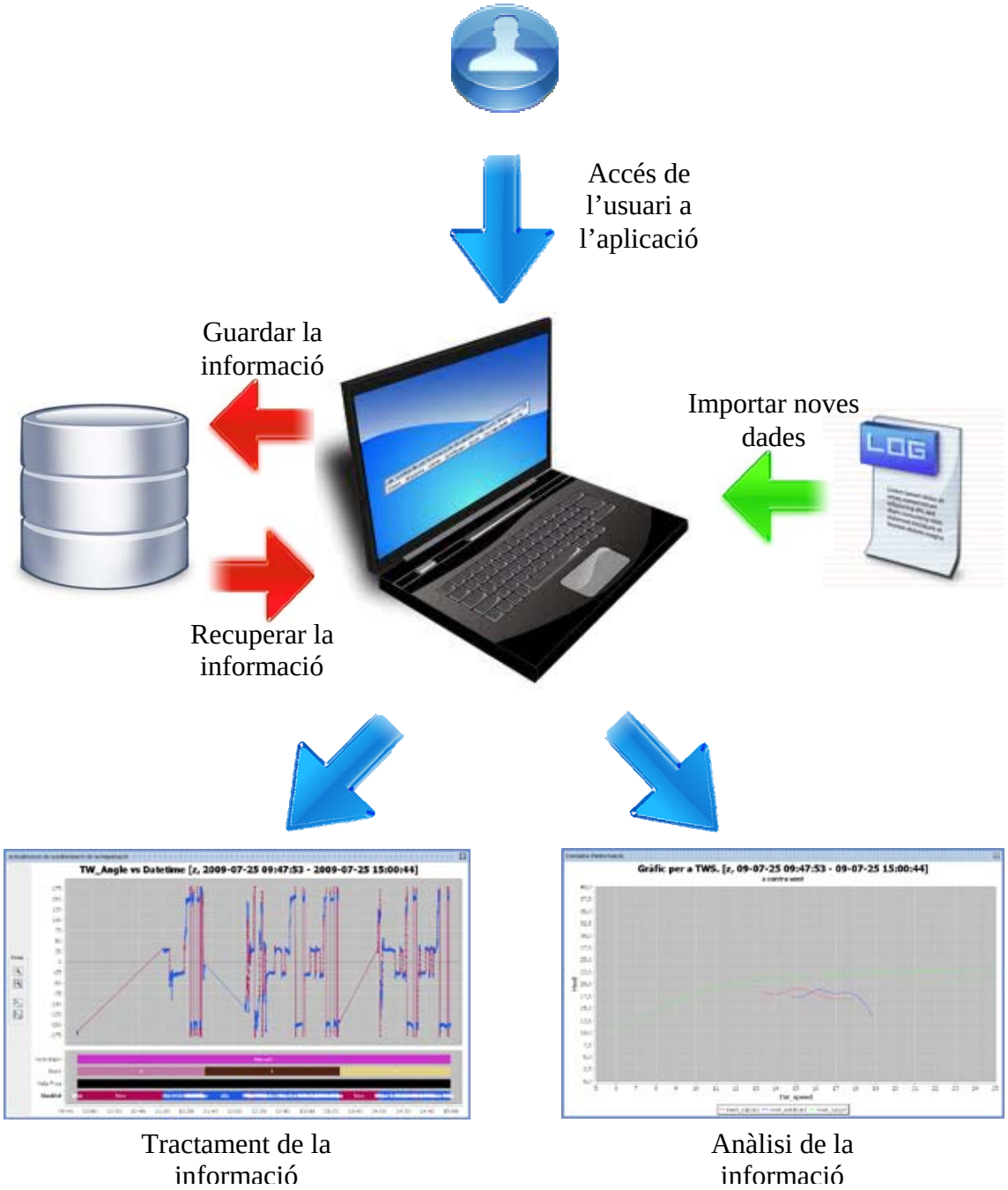


Figura 1: Visió global del sistema

1.5.1 Usuaris del sistema

Regaton és una eina específica que gestiona dades d'un àmbit molt concret. Els usuaris als que es dirigeix el sistema són persones amb bons coneixements nàutics i del món de les regates professionals.

L'eina permet la generació de gràfiques de diversos tipus com per exemple les gràfiques d'anàlisi d'una variable en funció de la velocitat real del vent o el tractament de la informació en funció de si és de babord o estribord. La interpretació d'aquest tipus de gràfiques requereix de coneixements nàutics i experiència en la navegació d'alt nivell per poder extreure'n conclusions útils que ajudin a maximitzar el rendiment de l'equip i el vaixell en les regates.

1.5.2 Arquitectura tècnica

El següent diagrama mostra l'estructura interna del sistema. Es tracta d'una arquitectura en tres capes que utilitza el patró Model – Vista – Controlador.

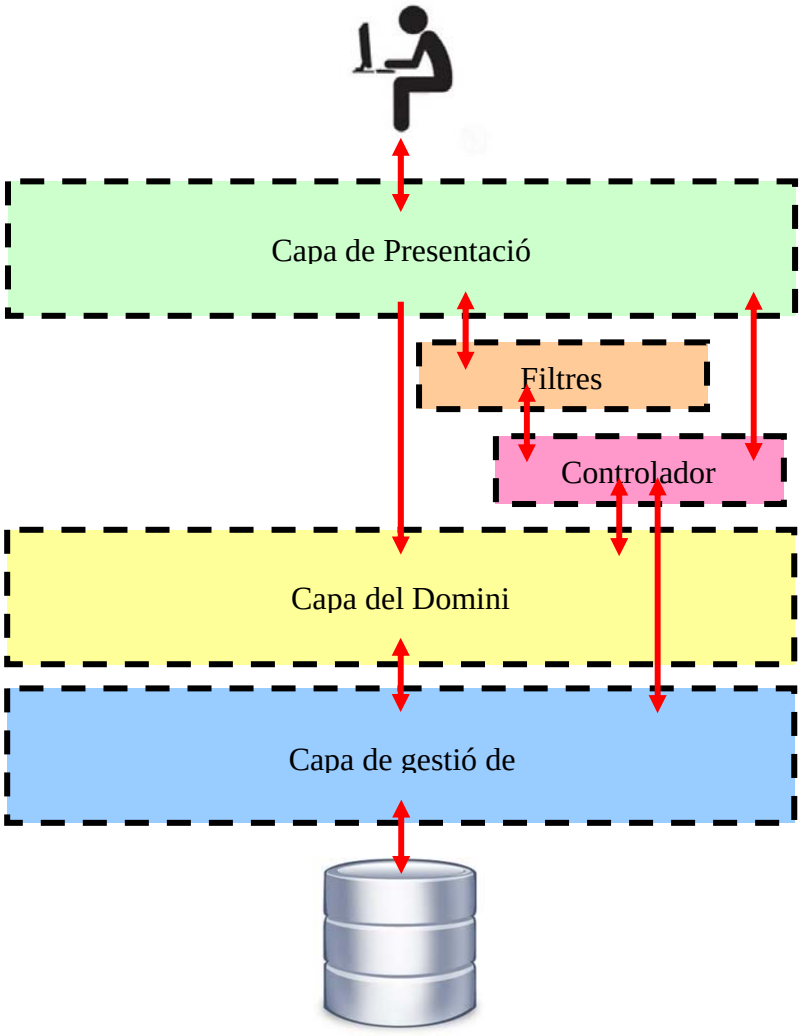


Figura 2: Arquitectura tècnica

1.5.3 Funcionalitats del sistema

El sistema desenvolupat posseeix diverses funcionalitats. Podem distingir-les en obligatòries i opcionals. Les opcionals són aquelles de les que el client no ha exigit la seva realització.

1.5.3.1 Funcionalitats obligatòries

1. Importació de fitxers de *Log* amb informació d'una o diverses regates.
2. Preparació i tractament de les dades importades per analitzar-les posteriorment.
3. Gravació persistent de les dades llegides dels fitxers de *Log* a una base de dades de regates.
4. Inserció, consulta i esborrat de veles.
5. Manteniment del temps d'ús de les veles.
6. Consulta i esborrat de regates importades.
7. Diari del sistema amb la història dels errors, avisos i missatges informatius produïts i les seves descripcions.
8. Gràfiques d'anàlisi:
    a. Gràfiques cronològiques.
    b. Gràfiques d'anàlisi de variables en funció de la velocitat real del vent (*TWS*).
9. *Filtres*:
    a. *Filtre de qualitat* sobre les dades.
    b. *Filtre* per a les gràfiques d'anàlisi de variables en funció de la velocitat real del vent.
10. Fitxers de *targets* per les gràfiques d'anàlisi en funció del *TW_speed*.
11. Manual d'usuari del sistema.

1.5.3.2 Funcionalitats opcionals

1. Creació i esborrat de la taula de regates.
2. Creació i esborrat de la taula de veles.

3. Gràfiques d'anàlisi:
    a. Gràfiques babord / estribord, per a qualsevol tipus de gràfica.
4. *Filtres*:
    a. *Filtre* babord / estribord per les gràfiques babord / estribord.
5. Fitxer bàsic de definició de la taula de regates de la base de dades.

1.5.4 Interfícies

La paraula interfície té diverses accepcions segons el context en que s'utilitzi. Dos són els sentits en els que exposaré les interfícies del sistema.

Per una banda, cal descriure quines són les interfícies d'entrada i sortida del sistema. La interfície d'entrada de dades de *Regaton* la formen els fitxers de *Log* amb les dades recol·lectades de les regates, que generen eines externes. Exceptuant la representació visual de la informació per l'anàlisi de les dades en la mateixa eina, *Regaton* no té cap interfície de sortida d'informació.

També podem parlar de la interfície d'usuari del sistema. La següent figura mostra la interfície principal:

Figura 3: Interfície d'usuari del sistema

Les funcionalitats s'han agrupat segons el seu objectiu i els recursos als que estan dirigides. Cadascun dels menús de la vista ofereix les funcionalitats relacionades amb el seu nom. Per exemple, el menú *Regates* permet importar, consultar i esborrar la informació recol·lectada durant les regates d'un vaixell. El menú de *Veles* permet inserir, consultar i esborrar les veles usades per l'equip que navega amb un vaixell durant la realització de les regates.

El manual d'usuari del sistema, que s'ha adjuntat al present document, detalla específicament cadascuna de les funcionalitats i les vistes associades als seus processos.

2 Context del projecte

El capítol persegueix atorgar als lectors que no han tingut l'oportunitat de realitzar cap incursió en el món nàutic, ni de les regates d'alt nivell, els coneixements necessaris per la correcta comprensió del treball realitzat en el projecte.

2.1 Descripció del projecte

Que coneguem els elements que nodreixen les regates de vaixells de vela ens permetrà entendre quines són les necessitats de l'equip de popa[1] de les tripulacions. Són aquestes necessitats les que donen sentit a la realització d'aquest projecte de fi de carrera.

És per això, que mostrarem en quines parts podem dividir una embarcació de vela i quins rumbs pot prendre mentre navega.

Les veles seran un altre concepte del que farà falta parlar. Elles són el motor, són les que aprofiten la incidència del vent i la força amb la que impacta per desplaçar les embarcacions de vela.

Un altre tema són els sensors instal·lats a bord. S'encarreguen de capturar en cada instant informació diversa associada a la navegació.

Per últim, són necessàries algunes nocions sobre les regates de vaixells de vela. Presentarem possibles circuits i la nomenclatura bàsica per poder comprendre, més endavant en la memòria, per què serveixen algunes de les funcionalitats de l'eina.

2.1.1 Els vaixells de regata

L'embarcació juntament amb la tripulació són l'essència de tota regata de vaixells de vela. El rendiment del vaixell es relaciona directament amb l'harmonia i eficàcia amb la que la tripulació afronta les seves funcions.

El llenguatge nàutic es va definir a partir de la necessitat de descriure les parts i moviments del vaixell respecte la direcció del vent.

Las parts d'una nau es defineixen considerant que l'observador està situat en el seu centre. La zona que queda davant de l'observador s'anomena *proa*, la que queda per darrera *popa*. La zona a la seva esquerra és *babord* i la que és a la dreta *estribord*.

[1] Dins de l'equip de popa, el navegant i el tàctic són els tripulants que s'ocupen d'analitzar, en temps real i després de les regates, la informació procedent dels sensors, la informació del transcurs de les regates, les dades atmosfèriques, etc. per decidir les accions a emprendre i assolir les condicions òptimes durant les regates.

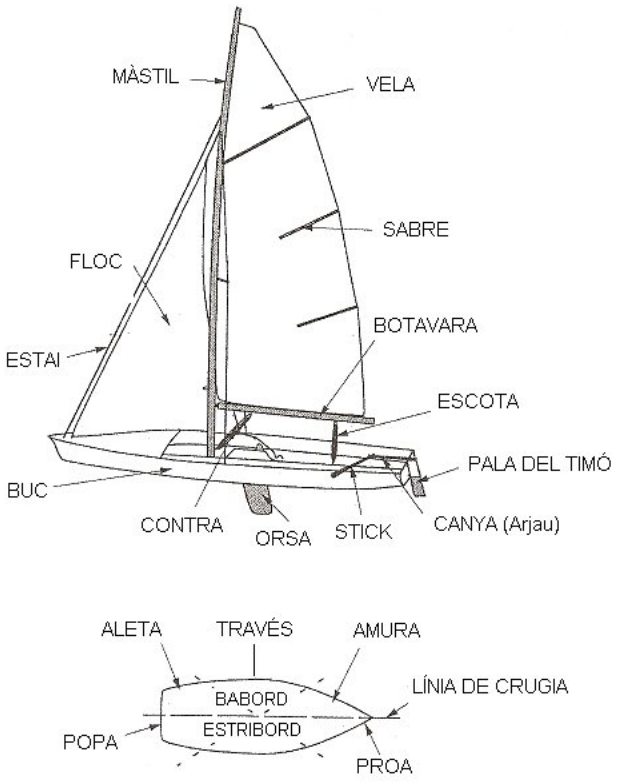


Figura 4: Components i terminologia dels vaixells de regata

El rumb relatiu[2] es mesura en graus en funció del desplaçament del vaixell respecte de la direcció del vent.

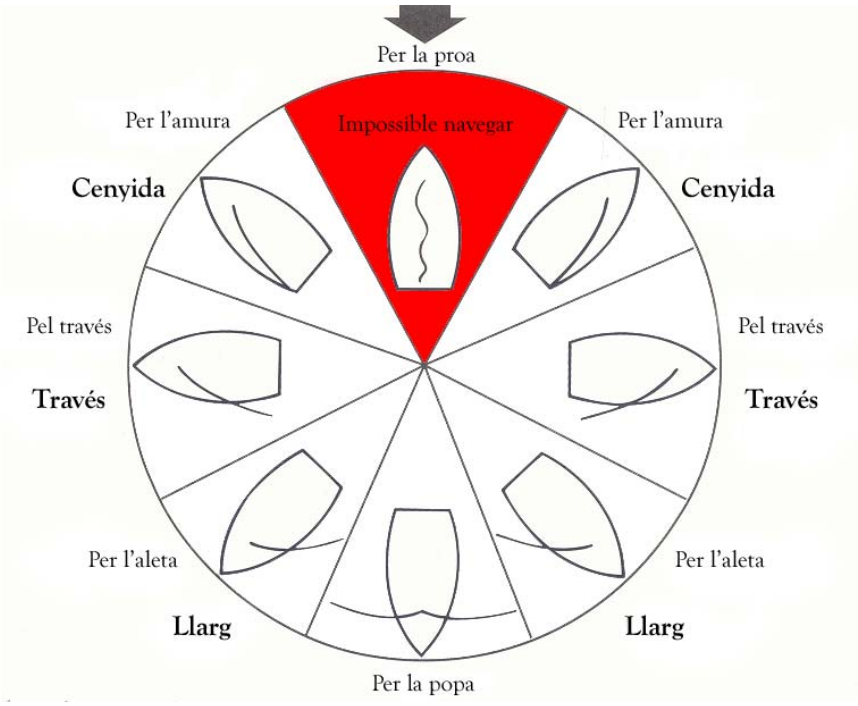
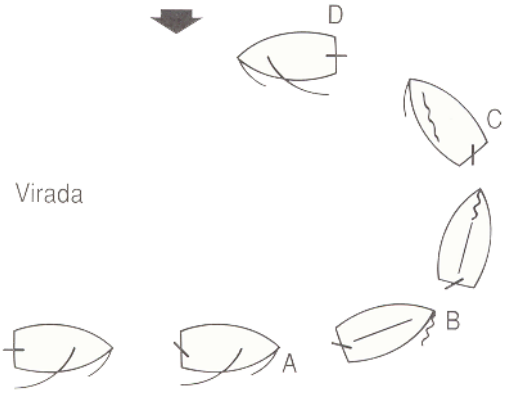
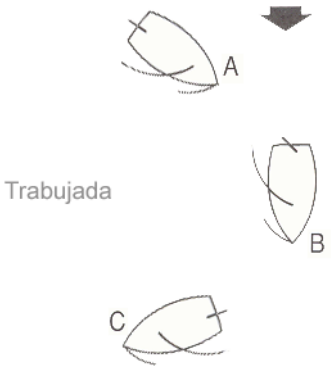


Figura 5: Rumbs relatius d'un vaixell

[2] El rumb absolut seria la direcció de desplaçament del vaixell en graus.

Existeixen dues paraules per designar els girs que pot realitzar un vaixell. S'anomena *virada* a un canvi de direcció en el que la proa passa per la direcció d'on ve el vent. En canvi, s'utilitza la paraula *trabujada* pel canvi de direcció en el que la popa passa per la direcció d'on ve el vent.





Figures 6 i 7: Maniobres de virada i trabujada

2.1.1.1 Polars del vaixell

Les taules Polars descriuen el rendiment del vaixell per totes les condicions de velocitat i angle real del vent. Representant radialment la velocitat de *corredera*[3]contra l'angle real del vent (*TWA*) i cada velocitat real del vent (*TWS*) s'obtenen les corbes polars de vent:

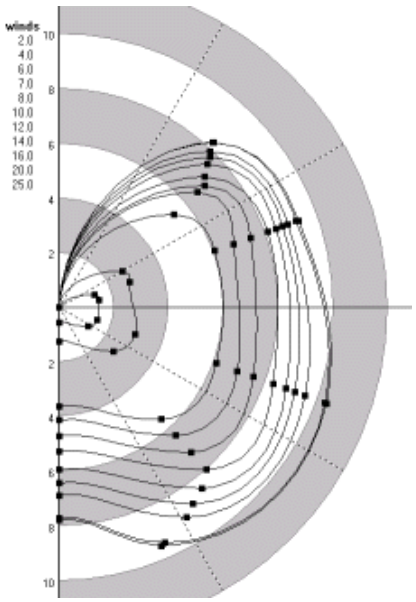


Figura 8: Gràfica de les polars d'un vaixell de regates

Les corbes Polars es poden deduir teòricament, per disseny predictiu, per certificació *IMS*[4] o a partir de l'anàlisi del rendiment del vaixell.

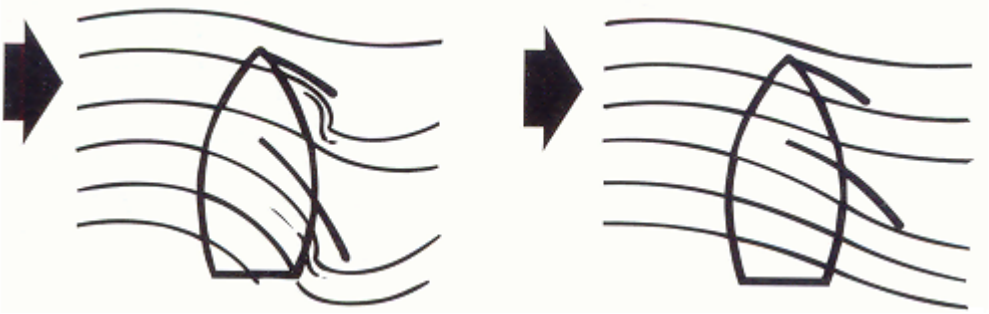
[3] Instrument per mesurar la velocitat d'una embarcació en marxa.
[4] Certificat de mesurament complet del vaixell de regata.

D'aquesta manera s'obté la velocitat de l'embarcació, el punt on una perpendicular des de *vent zero* incideix en la corba determina el primer punt de velocitat òptima per aquesta direcció del vent. Per aquesta velocitat de vent aquesta és converteix en la millor velocitat del vaixell, així com l'angle del vent es converteix en l'angle òptim.

2.1.1.2 Les veles

Les veles són el mecanisme que transforma l'energia de manera que, la força del vent es converteix en força de propulsió fent avançar l'embarcació.

Tot i que a simple vista pugui semblar que sigui el impuls directe del vent[5] sobre les veles el que genera la força propulsora no és així. Realment, la tècnica de la vela consisteix en orientar-les respecte el vent de manera que aquestes produeixin un petit canvi en la seva direcció, i aquest circuli a través d'elles sense produir turbulències.



Figures 9 i 10: Orientació de les veles respecte del vent amb i sense turbulències

La massa d'aire, que es desplaça per una vela corbada, produeix un augment de la pressió en el seu costat convex i una succió en el costat còncau. Com a resultat d'això, en tots i cadascun dels punts de la vela es produeixen petites forces, pràcticament perpendiculars a ella. Aquestes forces es poden considerar en una única força resultant anomenada impuls vèlic.

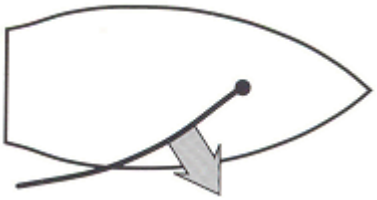


Figura 11: Suma de forces que resulten en el impuls vèlic

L'ús i trimat[6] d'unes o altres veles mentre es navega incideix directament en el rendiment que serà capaç d'assolir el vaixell.

[5] S'anomena *vent real* a el impuls real que reben les veles. Per altra banda *el vent aparent* o *relatiu*, és aquell que s'origina de la combinació del vent real i el moviment propi del vaixell.
[6] Ajustar les veles a les condicions del vent per aconseguir que treballin de forma òptima.

Però l'elecció de veles en el moment adequat no ho és tot, també cal tenir en compte el desgast que pateixen a mida que incrementa el seu temps d'ús. El desgast de les veles afecta principalment al seu rendiment.

Així, atenent el temps d'ús de les veles i el desgast acumulat, la tripulació pot decidir quan ha arribat el moment de substituir-les per unes altres.

2.1.2 Les regates

Una regata és una competició esportiva de velocitat, o una carrera entre embarcacions.

Les regates de vela poden ser de *vela lleugera* o de *vela de creuer*. Les regates de vela lleugera acostumen a realitzar-se sobre un recorregut marcat per boies o balises[7], normalment en forma de triangle, de la que la configuració més comuna és el *triangle olímpic*. Respecte la vela de creuer, a més dels recorreguts abalisats, es navega en regates de port a port. Quan el recorregut travessa diferents mars, la regata s'anomena *regata oceànica* o *transoceànica*.

Avui en dia existeixen regates de renom com la *Copa Amèrica*, *Barcelona World Race*, *Volvo Ocean Race*, etc.

2.1.2.1 Representació de recorreguts

A continuació es mostren diversos esquemes dels tipus de recorreguts més utilitzats. La trajectòria de la nau es mostra en línia contínua, de manera que cada esquema pot mostrar recorreguts amb diferent nombre de voltes.

2.1.2.1.1 Recorregut Sobrevent[8] – Sotavent[9], també anomenat Bastó

Les opcions del recorregut inclouen:

1. Augmentar o disminuir el nombre de voltes.
2. Suprimir l'últim tram cap a sobrevent.
3. Utilitzar una porta[10] envers d'una balisa de sotavent.
4. Utilitzar una balisa de desmarcatge a la balisa de sobrevent.
5. Utilitzar les balises de sotavent i sobrevent com balises de sortida i arribada respectivament.

[7] Element flotant de senyalització nàutica. Permeten senyalitzar i regular els circuits en els que es produeixen les regates.
[8] Costat cap on va el vent.
[9] Costat d'on ve el vent.
[10] Una porta està formada per dues boies.

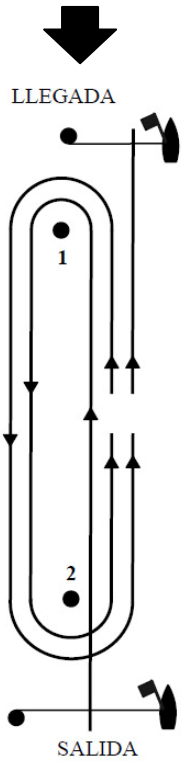


Figura 12: Bastó: Sortida – 1 – 2 – 1 – 2 – Arribada

2.1.2.1.2 Recorregut Sobrevent – Sotavent – Triangle, també anomenat triangle olímpic

Les opcions del recorregut inclouen:

1. Augmentar o disminuir el nombre de voltes.
2. Suprimir l'últim tram cap a sobrevent.
3. Variar els angles interiors del triangle (45º - 90º - 45º - 60º - 60º - 60º són usuals).
4. Utilitzar una porta envers d'una balisa de sotavent pels trams d'empopada[11].
5. Utilitzar una balisa de desmarcatge al principi dels trams d'empopada.
6. Utilitzar les balises de sotavent i sobrevent com balises de sortida i arribada respectivament.
7. Cal especificar l'angle intern de cada balisa.

[11] Navegació d'un vaixell de vela que rep el vent per la popa.

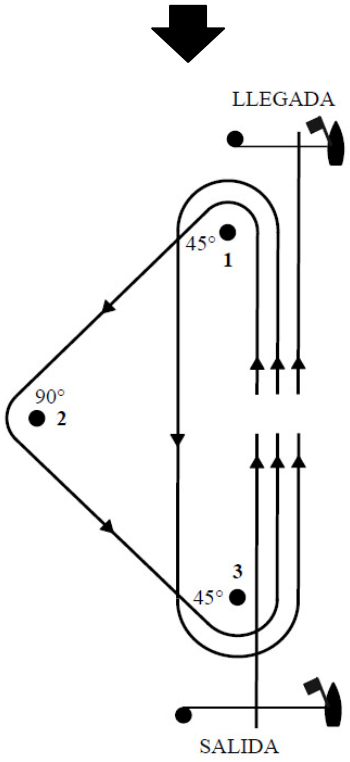


Figura 13: Triangle: Sortida – 1 – 2 – 3 – 1 – 3 - Arribada

2.1.2.1.3 Recorregut Trapezoïdal

Poc utilitzat en creuers.

Les opcions del recorregut inclouen:

1. Afegir trams addicionals.
2. Substituir la porta amb una balisa o utilitzar una porta també en el bastó de l'esquerra.
3. Variar els angles interiors dels trams de llarg[12].
4. Utilitzar una balisa de desmarcatge al començament dels trams d'empopada.
5. Finalitzar a sobrevent envers de en un llarg.
6. Cal especificar l'angle interior de cada tram de llarg.

[12] Cadascun dels trams del recorregut d'un veler coberts navegant per l'aleta.

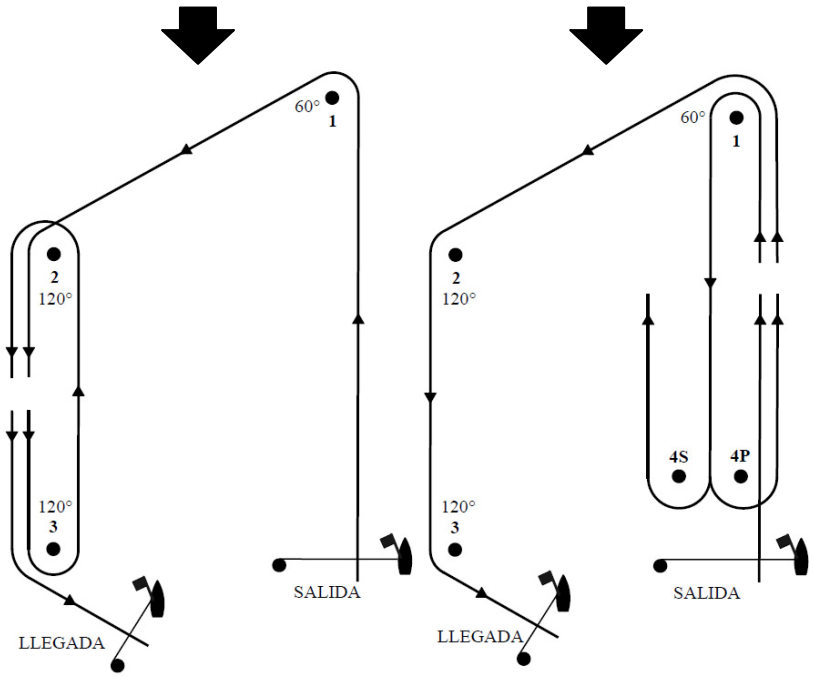
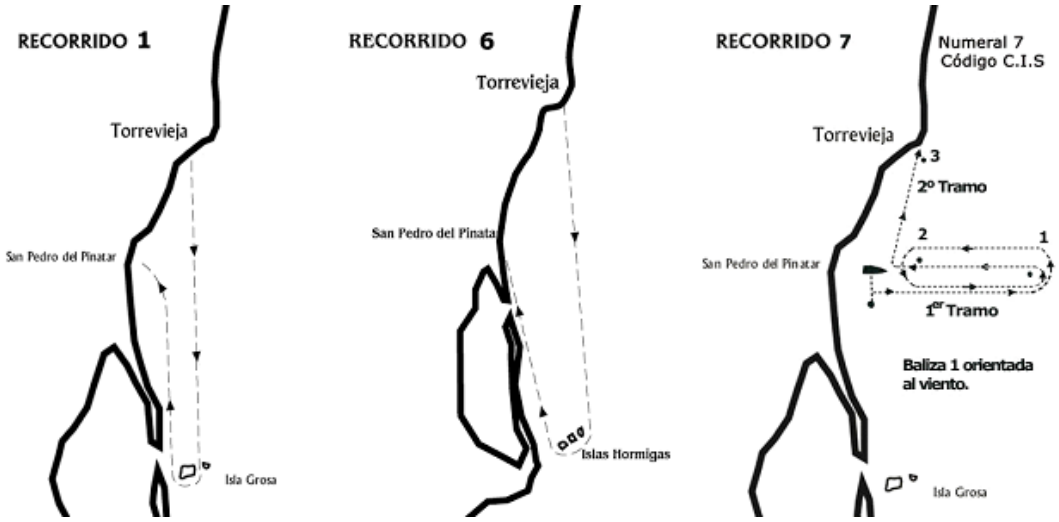


Figura 14 i 15: Recorregut Trapezoïdal: Sortida – 1 – 2 – 3 – 2 – 3 – Arribada i Recorregut Trapezoïdal: Sortida – 1 – 4 – 1 – 2 – 3 – Arribada

2.1.2.1.4 Recorreguts costaners

A continuació es mostren alguns exemples de recorreguts de la *Regata Estrella Levante'09*[13]:



Figures 16,17 i 18: Exemples de recorregut costaners

[13] Enllaç al web del Club Nàutic de Torrevieja on consular els recorreguts de les regates per l'any 2009 http://www.regataestrella.com/recorridos.php

2.1.2.1.5 Recorreguts oceànics

Recorregut de la regata *Volvo Ocean Race 2008 - 2009*, que es recorre al voltant del món:

Figura 19: Recorregut oceànic de la regata Volvo Ocean Race 2008 - 2009

2.1.3 Sensors i recol·lecció d'informació

Els sensors són components electrònics distribuïts arreu de la nau que permeten que el navegant i el tàctic, en temps real, disposin de mesuraments i representacions gràfiques de certs paràmetres. Normalment, entre ells s'hi inclouen indicadors com la posició, la velocitat i direcció del vent, velocitat i rumb del vaixell i la profunditat. Els sistemes més sofisticats estableixen aquestes mesures i a partir d'elles calculen i mostren informació addicional com velocitat aparent i real del vent, rumb sobre el fons[14], velocitat sobre el fons[15] i moltes altres dades.

El control dels sensors i de la resta de l'equip electrònic són responsabilitat del navegant que utilitza el sistema electrònic i *loggers*[16] per calibrar-los, configurar-los i recollir dades.

[14] En anglès *Course Over Ground*, o *COG*. Angle format per la línia proa - popa de l'embarcació amb el meridià geogràfic (nord geogràfic) del lloc.
[15] En anglès *Speed Over Ground*, o *SOG*. Velocitat de la nau mesurada sobre el fons marí.
[16] Equip electrònic que registra la informació dels sensors.

Mitjançant un *logger*, o un programa tàctic del mercat[17] compatible amb l'equip instal·lat s'obté i s'analitza la informació.

Entre les característiques d'aquests sistemes existeix la possibilitat de definir la freqüència de lectura dels sensors (que acostuma a ser de 1-10 MHz) i el format amb el que s'obté la informació (format tabulat, *XML*, ..., però també es poden escollir els sensors a tenir en compte).

Si l'arxiu és tabulat, aquest conté una capçalera amb els noms de les variables d'estudi dels sensors.

Degut a que la quantitat de sensors depèn en gran mesura de la mida del vaixell i sobretot de la informació que el tàctic necessiti analitzar no existeix un format estàndard pel nombre de camps que inclou el registre.

Per aquest motiu, es defineix el nombre mínim de sensors que ha de contenir qualsevol arxiu. Els camps són: data i hora, la posició de la nau, la velocitat real del vent (*TWS*), l'angle real del vent (*TWA*) i la velocitat real del vaixell (*Boatspeed*).

2.1.3.1 Descripció dels paràmetres dels sensors usats en el projecte

La taula descriu la següent informació:

1. *Log*: Nom de la variable en el fitxer de *Log*.
2. *Paràmetre*: Nom del paràmetre que representa el nom de la variable en el *Log*.
3. *Descripció*: Descripció del paràmetre.
4. *Unitat de mesura*: Unitat en que es mesura el paràmetre. Entre parèntesis s'indica el tipus de dades que li correspon.

Log	Paràmetre	Descripció	Unitat de mesura
Position	Position	Latitud i longitud segons el GPS.	Graus (real)
Datetime	Datetime	Data dd/mm/aa hh:mm:ss.	
TW_speed	True Wind Speed (TWS)	Velocitat real del vent.	Nusos (real)
Heel	Heel	Escora, inclinació del vaixell. Valors positius per estribord, negatius per babord.	Graus (real)
Heel_Targ	Heel Target	Escora objectiu. Varia amb la TWS. Tot i que es mesura sempre, aporta informació en cenyida.	Graus (real)
Ext_SOG	Speed Over Ground (SOG)	Velocitat del vaixell segons calcula el GPS.	Nusos (real)
Boatspeed	Boat speed	Velocitat de l'embarcació segons el sensor de velocitat.	Nusos (real)
Vs_targ	Velocitat Target	Velocitat objectiu, varia amb la TWS.	Nusos (real)
Vs_targ%	% Velocitat Target	Percentatge de Velocitat Target: *Boatspeed / Vs_targ * 100*	(enter)
TW_angle	True Wind Angle (TWA)	Angle real entre el rumb de l'embarcació i el rumb del vent.	Graus (enter) 0 <= |TWA| <= 180

[17] *Expedition*, *B&G Deckman* són dels programaris de navegació tàctica més famosos actualment. Altres sistemes existents són *Seatrack*, *Evolution*, etc.

		Valors positius per estribord, negatius per babord. Navegant en: cenyida (*upwind*): 0 <= |TWA| <= 120 popa (*downwind*): 120 <= |TWA| <= 180	
TWA_Targ	True Wind Angle Target	Angle real objectiu, varia amb la TW_speed i també segons s'estigui navegant en cenyida o popa.	Graus (enter)
AW_angle	Apparent Wind Angle (AWA)	Angle aparent entre el rumb de l'embarcació i el rumb del vent. S'anomena aparent perquè no té en compte que l'embarcació està en moviment, enfront del real que si ho considera. Valors positius per estribord, negatius per babord. En cenyida (*upwind*): 0 <= |AWA| <= 60 En popa (*downwind*): 60 <= |AWA| <= 180	Graus (enter) 0 <= |AWA| <= 180
AW_speed	Apparent Wind Speed (AWS)	Velocitat aparent del vent. S'anomena aparent perquè no té en compte que la embarcació està en moviment, enfront del real que si ho considera.	Nusos (real)
Leeway	Leeway	Deriva, ve a ser la diferència entre el rumb que fa la proa del vaixell i el rumb real al que s'està navegant (és a dir, quant el vaixell es mou "de costat"). Valors positius per babord (TWA < 0), negatius per estribord.	Graus (real)
Ext_COG	Course Over Ground (COG)	Rumb del vaixell segons el calcula el GPS.	Graus (enter) 0 <= COG <= 360
Course	Course	Rumb del vaixell tenint en compte la deriva: *Course = Heading + Leeway*	Graus (enter) 0 <= Course <= 360
Heading	Heading	Rumb (o direcció) del vaixell segons el sensor de rumb (compàs).	Graus (enter) 0 <= Heading <= 360
TW_Dirn	True Wind Direction (TWD)	Direcció o rumb del vent.	Graus (enter) 0 <= TW_dirn <= 360
VMG	Velocity Make Good (VMG)	Càlcul vectorial que mesura el rendiment del vaixell en funció de la seva velocitat i el seu angle: *VMG = Boatspeed * cos(TWA)* Valors positius en cenyida, negatius en popa (cosinus majors de 90º són negatius).	Nusos (real)
Forestay	Forestay	Pressió del sensor de l'estai[18]	Tones (real)
Rudder	Rudder	Angle del timó respecte la proa del vaixell. Valors positius quan el timó va cap a estribord, negatius quan va cap a babord.	Graus (real)
Rudder_Targ	Rudder Target	Angle del timó objectiu, varia amb la TWS. Tot i que es mesura sempre, només aporta informació en cenyida.	Graus (real)
Vs_Nav	Vs_Nav	Velocitat objectiu segons les polars del vaixell. Coincideix amb la velocitat objectiu per als TWA_Targ i també dóna la velocitat objectiu per angles diferents als de cenyida i popa pura. Varia amb la TWS o el TWA.	Nusos (real)
Vs_Nav%	Vs_Nav%	Percentatge de velocitat segons les polars: *Boatspeed / Vs_Nav * 100*	(enter)
MCurRate	MCurRate	Intensitat del corrent mesurat en el mar.	Nusos (real)
MCur_Dir	MCur_Dir	Rumb o direcció cap a on va el corrent.	Graus (enter) 0 <= MCur_Dir <= 360

Taula 1: Variables d'estudi dels fitxers de Log

[18] Cable que va de l'extrem del pal major a la proa del vaixell.

2.2 Estudi de mercat

Aquest estudi es centra en el sector de programari tàctic de navegació emmarcat dins del món de la competició nàutica d'alt nivell. Es tracta d'un sector amb patrons de consum i canals de comercialització molt específics, concentrats en les necessitats específiques de les embarcacions i les seves tripulacions.

Aquests sistemes operen sobre un micro-computador durant la navegació o una regata; integrant contínuament tota la informació que recol·lecten els sensors del vaixell amb el rendiment esperat.

Són eines que pretenen simplificar les tasques del navegant i permetre decisions en base a la millor informació disponible. També permeten la retroalimentació[19] amb els dissenyadors de vaixells, eixàrcia[20] i veles. Però alhora són d'elevat cost econòmic (per sobre dels 1000€ en la majoria de casos).

En el cas d'alguns equips de més alt nivell amb pressupostos més elevats, en regates com la *Copa Amèrica* o la *Volvo Ocean Race*, es construeixen eines a mida.

La informació obtinguda per l'estudi prové de les pàgines web de distribuïdors i venedors dels programaris i webs de caràcter nàutic.

2.2.1 Anàlisi de l'oferta

En aquest mercat predomina el programari tàctic amb marques internacionals de renom. És un mercat dominat per productes importats, degut a que la seva producció a Espanya és reduïda o nul·la. Per tant, la competència principal a la que hauria de fer front el nostre producte prové de Regne Unit, els EUA, França o altres.

La inversió en programari tàctic és un cost addicional al que suposa la inversió en veles, el manteniment del vaixell i altres aspectes en els que es gasten diners per millorar el rendiment d'un veler. Són aplicacions de cost elevat que alhora requereixen d'instruments electrònics de gamma alta capaços d'oferir la precisió esperada en les seves dades d'entrada. Però l'elecció i compra del programari adequat han d'anar estretament lligades a les necessitats de l'embarcació en funció del tipus de regates per les que s'ha construït.

Cal tenir en compte, que el programari necessitarà de l'atenció del navegant i el tàctic de la tripulació per tal de conèixer els valors requerits en cada moment.

Algunes marques presents en el mercat del programari tàctic de navegació són *B&G*, *Landfall*, *Adrena*, etc.

[19] Conjunt de reaccions o dades que l'equip d'una embarcació, mitjançant les eines tàctiques, retornen als dissenyadors per a que analitzin i modifiquin els seus vaixells, veles, etc. En anglès *feedback*.
[20] Conjunt de caps utilitzats a bord.

2.2.1.1 Deckman de B&G[21]

Programari en la versió 9.1 de desenvolupament. És una eina dissenyada per enginyers i dirigida a navegants.

Deckman és usat per molts dels competidors de la *Volvo Ocean Race* o la *Copa Amèrica*. El programari inclou totes les eines indispensables per curses de curta distància amb boies, regates a alta mar i de configuració de la nau.

2.2.1.1.1 Funcionalitats

Motor gràfic amb àmplies funcionalitats com la de *layline*[22], dades en temps real, posicionament, etc.

El navegador optimitza l'estratègia i la tàctica mentre s'està navegant en competició, i permet el test mentre s'està configurant i entrenant.

Figura 20: Navegació tàctica

Navegant a alta mar, en regates oceàniques el navegador té en compte les condicions meteorològiques calculant la ruta òptima.

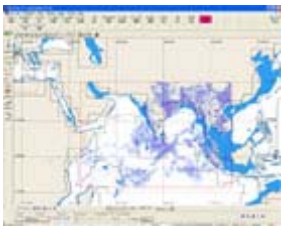


Figura 21: Funcionalitat meteorològica i de rutes

L'eina permet la càrrega de fitxers de dades meteorològiques, la configuració i creació d'animacions meteorològiques incloent vent, pressió, precipitació per realitzar prediccions.

Aporta un mòdul de càlcul de la ruta més ràpida.

[21] Pàgina web oficial de *B&G* on obtenir la informació relacionada amb el producte http://www.bandg.com/

[22] Línia recta des de la posició actual del vaixell fins la balisa.

Inclou l'ús de múltiples taules polars, una per objectius de rendiment, una altra per navegació, i una altra per *presortides*[23], permetent al navegant ajustar una polar per precisar-la al rendiment actual de l'embarcació.

Test de rendiment gràcies a un conjunt de *loggers* i eines d'anàlisi, per permetre la millora del rendiment de la tripulació i l'embarcació sota diferents condicions, durant les regates o els entrenaments o després de la seva finalització.

Test d'un sol vaixell, i comparació de múltiples vaixells, a partir de tests de velocitat per permetre comparacions estadístiques entre vaixells i l'ús de telemetria[24].

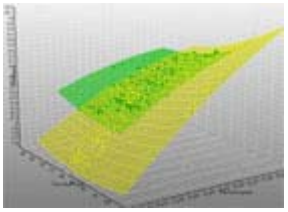


Figura 22: Recursos i anàlisis

2.2.1.2 Expedition de LandFall Navigation[25]

Programari tàctic de navegació en la versió 4.2. És una eina dissenyada per navegants i dirigida a navegants.

Aplicació creada per Nick White navegant veterà de la *Volvo Ocean Race*. La desenvolupava pel seu ús personal en la regata *Volvo Ocean Race* 2001 – 2002. Des de llavors ha estat en continu refinament per un equip de navegants de renom.

El resultat final és un sistema més avançat i útil que molts altres programaris tàctics, orientat tant a regates d'àmbit mundial com de curt recorregut.

2.2.1.2.1 Funcionalitats

Representació de gràfiques per l'anàlisi de dades.

Sistema avançat de tractament de meteorologia.

Seguiment des d'inici de les regates amb funcions avançades per regates de boies.

Instruments d'entrada i sortida per intercanviar dades amb tots els sistemes més populars, incloses les polars i els paràmetres de configuració.

[23] Preparació prèvia amb el vaixell per travessar la balisa o porta de sortida en l'instant que comença la regata navegant amb les condicions òptimes.
[24] Tecnologia que permet mesurar de manera remota magnituds físiques.
[25] Pàgina oficial del *Landfall* on obtenir la informació relacionada amb el producte http://www.landfallnavigation.com/

Funcions polars permeten extreure, analitzar i modificar les polars de rendiment dels vaixells.

Funcions pel seguiment dels competidors durant les regates.

Sistema de creació de rutes en funció de la meteorologia. Creació de rutes òptimes.

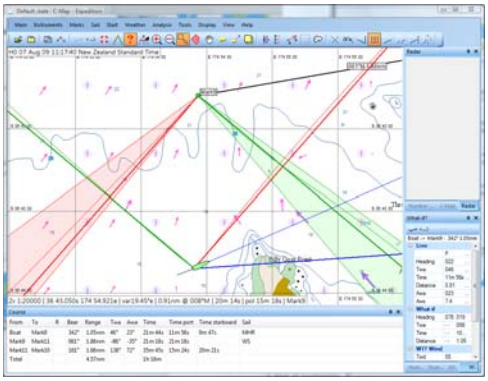


Figura 23: Creació de rutes

Funcions de gravació de les dades dels instruments de navegació per un posterior anàlisi o repetició de la regata.

Interfície *Google Earth*.

Figura 24: Navegació

2.2.1.3 Optima d'Adrena[26]

Programari en la versió 7 de desenvolupament.

És un sistema d'anàlisi de rendiment i creació de polars de velocitat. Es completa amb el mòdul *OptimaPro*, per incloure nous paràmetres de càlcul de polars.

[26] Pàgina oficial del *Adrena* on obtenir la informació relacionada amb el producte http://www.adrena.fr/

Programari d'alt nivell dissenyat específicament per regates que alhora aporta les funcionalitats clàssiques cartogràfiques i d'ajuda a la navegació. Utilitzat per les millors tripulacions franceses, és un programari d'innovadora ergonomia i molt fiable.

2.2.1.3.1 Funcionalitats

Calibratge simple dels instruments.

Durant la fase de gravació, Optima emmagatzema totes les dades dels instruments de navegació. El programa treballa independentment, i com més dades es recol·lecten més completes i fiables són les polars.

El programari carrega les dades gravades per realitzar el càlcul de les polars de velocitat excloent la informació no rellevant.

Permet utilitzar les polars per fer seguiment del rendiments dels vaixells, crear un fitxer de polars i usar-lo en la creació de rutes, o usar de les polars en la preparació de la sortida en les regates.

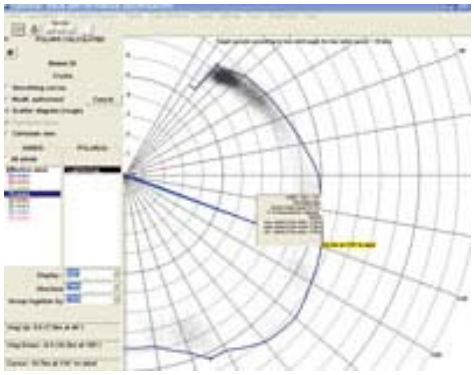


Figura 25: Ús de Polars

Sailect, programari que indica contínuament quines veles usar segons les condicions de la regata, i anticipa l'ús de les millors veles pels propers passos. És possible consultar les decisions preses per l'eina en la repetició de les regates.

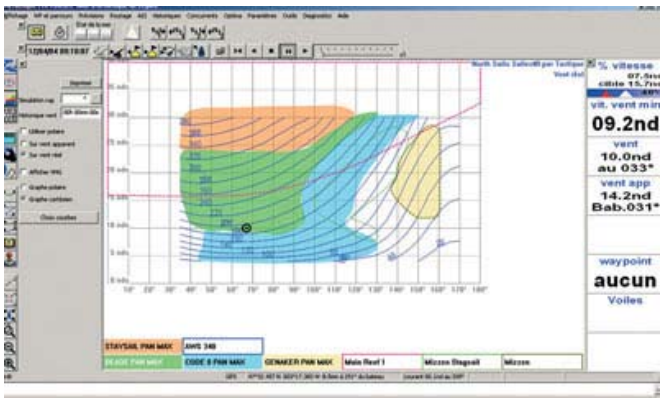


Figura 26: Sailect, selecció veles

Permet el càlcul i optimització de *Sailcharts*; són diagrames de selecció de veles per cada angle i intensitat del vent.

Funció de repetició de la regata per analitzar diferents passos, les opcions escollides i l'evolució de la meteorologia.

2.2.1.4 Seatrack[27]

Programari en la versió 11 de desenvolupament. És una eina dissenyada per navegants i dirigida a navegants.

Seatrack és un paquet de programes informàtics que proporciona un sistema tot en un, integració de funcionament, tàctic i navegació.

Pretén donar solucions que maximitzin l'eficàcia, funcionament i seguretat en el mar, mentre que alhora és senzill d'utilitzar.

El programa d'interès per aquest estudi és el *Performance & Tactical Data Analysis.*

2.2.1.4.1 Funcionalitats

Proporciona informació detallada pel l'anàlisi posterior de les regates per prendre decisions tàctiques i conèixer el rendiment actual del vaixell en contra dels rendiment teòric derivat de les seves polars.

Ofereix tres modes de treball:

1. Anàlisi d'un vaixell: rendiment i informació tàctica en un punt determinat del temps.

Figura 27: Anàlisi d'un vaixell

2. Anàlisi vaixell contra vaixell: Comparativa del rendiments i les decisions tàctiques preses respecte un vaixell competidor (per la qual són necessàries les dades recol·lectades del vaixell contrari).

[27] Pàgina oficial de *Seatrack* on obtenir la informació relacionada amb el producte http://www.seatrack.co.uk/Default.htm

3. Per zones d'una regata: Anàlisi de rendiment basat en les mitges de les dades en una secció discreta d'una regata.

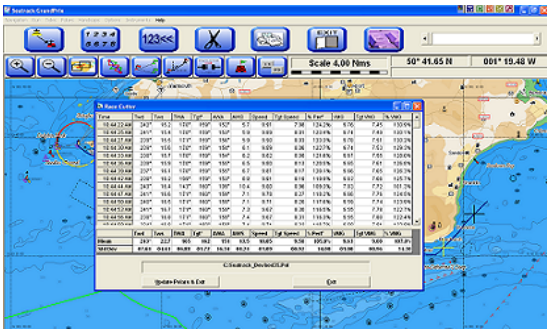


Figura 28: Anàlisi per zones d'una regata

2.2.1.5 Evolution[28]

Resol càlculs tàctics considerant les característiques de cada vaixell per qualsevol condició de vent i corrent, eliminant els factors d'incertesa a bord.

2.2.1.5.1 Funcionalitats

Considerant la posició de l'embarcació i el seu destí, per exemple la següent marca en el recorregut, queda establerta una situació tàctica per la que el sistema determina un model de navegació ideal sobre la base del vent i corrents que prevalen.

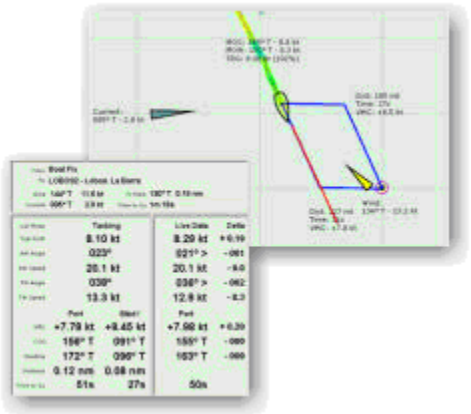


Figura 29: Calculadora tàctica resolent navegació ideal segons les polars del vaixell

El programari calcula contínuament el següents objectius de rendiment:

1. Velocitat objectiu del vaixell per qualsevol condició de *TWA*.

[28] Pàgina oficial d'*Evolution* on obtenir la informació relacionada amb el producte http://www.evolution-tactic.com/product/welcome/welcome.html

2. Velocitat del vaixell pels punts *VMG* òptims fent bordades[29] en cenyida o popa.

3. Angle del vent (*TWA* o *AWA*) pels punts de *VMG* òptims fent bordades en cenyida o popa.

4. Diferència entre l'escora recomanada pel dissenyador del vaixell i l'escora real.

Es pot realitzar l'anàlisi per cada tram del recorregut, permetent la selecció de veles i l'elecció d'una estratègia amb la necessària anticipació.

La informació gravada es pot analitzar de les següents maneres:

1. Amb reproductor, reproduint la regata de nou.

2. Ofereix l'ús de gràfiques comparatives per analitzar els paràmetres reals contra els objectius o altres regates realitzades prèviament.

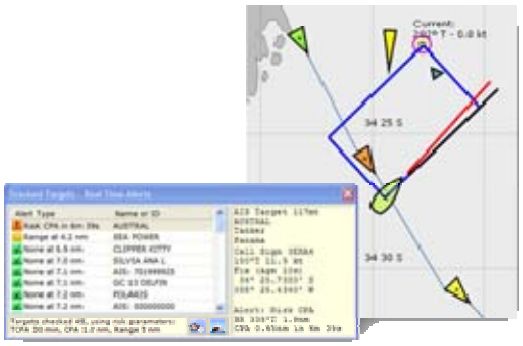


Figura 30: Llista vaixells detectats

2.2.1.6 Programari tàctic per a regates *amateurs*

En les regates de menys nivell els equips i les embarcacions tenen necessitats molt semblants a les d'alta competició.

Per donar resposta a aquests necessitats neix una altra família de programes tàctics de baix cost, que ajuden a realitzar les regates més eficientment. Sense cap dubte la generalització dels *GPS* en dispositius mòbils ha estat la clau. Aquests dispositius ofereixen costos raonables i possibilitats de programació més senzilles.

En conseqüència, avui poden trobar-se programes tàctics per sota dels 100€ que funcionen sobre dispositius que la majoria de nosaltres tenim. No són tan potents ni versàtils com els de gamma alta però donen resultats satisfactoris pels usuaris als que estan destinats.

[29] Una bordada és el camí recorregut per un veler entre dues virades en cenyir.

Quins valors afegits tenen aquestes eines? En primer lloc donen valors numèrics per avaluar el rendiment del vaixell, amb els que mesurar si un canvi de rumb millora l'aproximament a la boia, facilitació de maniobres que requereixen precisió, etc.

A més, l'ús d'aquestes eines atorga una navegació més organitzada aportant beneficis com:

1. millor situació en les regates
2. millor coneixement de les condicions de navegació
3. millor coneixement de les possibilitats de l'embarcació
4. una reacció més ràpida davant dels canvis de condicions que afecten al comportament del vaixell

Un exemple d'aquesta classe de programari és *GPS Race Driver*[30] per *Windows Mobile*.

Figura 31: Funcionalitats diverses de l'eina GPS Race Driver

[30] Pàgina oficial del *GPS Race Driver* on obtenir la informació relacionada amb el producte http://www.gpsracedriver.com/

3 Planificació i pressupost econòmic

Com en tot desenvolupament d'un projecte una de les primeres tasques a realitzar és la planificació inicial. A partir d'aquesta planificació aproximada l'equip de treball i el client poden fer-se una idea global del cost temporal de les fases del projecte i els permet abordar temes com el pressupost econòmic i el cost de cadascun dels recursos que s'assignaran a les fases del projecte.

El capítol detalla les planificacions temporals del projecte, la inicial i la final o real, explica en quines fases es divideix el desenvolupament i quins recursos se li assignen, i per últim, mostra el pressupost econòmic que es presentaria al client.

3.1 Fases del projecte

Les fases principals del projectes són les següents:

1. *Estudi previ*: Definició del projecte que es desitja construir. Quan el concepte del projecte ja és una realitat, es procedeix a l'estudi de les tecnologies disponibles i de les possibilitats que ofereixen.

2. *Preparació de l'entorn de treball*: Etapa en la que es preparen les eines i tecnologies necessàries pel desenvolupament del projecte.

3. *Aprenentatge*: Fase en la que l'equip s'adapta a les noves eines de treball i les tecnologies escollides. També és el moment de conèixer i comprendre els conceptes associats al projecte que seran necessaris per la correcta realització del projecte.

4. *Anàlisi de requeriments*: Definició de requeriments funcionals i no funcionals del projecte. A partir d'aquesta fase apareixen els objectius a assolir en el desenvolupament del projecte.

5. *Especificació*: Etapa en la que es defineix que ha de fer el sistema a partir dels casos d'ús. També s'estableixen els elements que el formaran a partir dels requeriments funcionals. La descripció del sistema és independent de l'arquitectura que s'utilitzi per la seva implementació.

6. *Disseny*: Es transforma la descripció del sistema a un model dependent de l'arquitectura i alhora es detalla com realitzarà els casos d'ús del projecte.

7. *Implementació*: Es construeix el sistema a partir del disseny realitzat.

8. *Proves*: Quan la implementació del projecte es troba en una fase avançada és l'hora de començar amb les proves. A partir de les proves es detecten errors en la implementació i es poden anar corregint mentre es comprova l'estabilitat del sistema.

9. *Redacció de la memòria i creació del manual d'usuari de l'eina*: Es redacta el document que detalla cadascuna de les fases del projecte, presenta els diagrames i esquemes de l'especificació, el disseny i la implementació necessaris per mostrar una visió completa i el més detallada possible de la realització del projecte. A més a més, en el cas de la implementació d'una aplicació, cal escriure el manual d'usuari.

10. *Preparació de les transparències per la defensa del projecte*: Es preparen les transparències per la lectura del projecte davant del tribunal.

3.2 Pressupost del projecte

El pressupost econòmic d'un projecte és essencial en qualsevol entorn empresarial. La valoració del cost dels recursos assignats al projecte, dels materials usats, les hores dedicades, les eines externes necessàries, etc. s'acumulen per obtenir el cost aproximat de la realització del projecte.

A partir de la proposta de solució i del pressupost el client avalua a quin proveïdor, empresa o equip de treball li comprarà la solució.

Per elaborar el càlcul del pressupost cal conèixer els recursos disponibles o que s'assignaran al projecte, el programari necessari i les hores que es dedicaran a cada etapa del desenvolupament.

El cost de l'equip de treball es calcula tenint en compte que són necessaris els següents perfils tècnics:

1. *Cap de projecte*: encarregat de realitzar les reunions amb el client, valorar el cost econòmic del projecte i dirigir l'equip de treball.

   Cost per hora del perfil: 75€

2. *Analista*: encarregat de l'especificació del sistema.

   Cost per hora del perfil: 42,5€

3. *Dissenyador*: encarregat del disseny del sistema i el seu desplegament.

   Cost per hora del perfil: 42,5€

4. *Programador*: encarregat de la implementació, les proves i la creació del manual d'usuari del sistema.

   Cost per hora del perfil: 20€

Tasca	Preu / hora	Hores	Preu	Preu (amb IVA)
Reunions etc.	75	25	1875	2175
Especificació	42,5	40	1700	1972
Disseny	42,5	100	4250	4930
Implementació	20	360	7200	8352
Proves	20	40	800	928
Desplegament	42,5	10	425	493
Documentació	20	120	2400	2784
Total		**695**	**18650**	**21634**

Taula 2: Cost dels recursos humans

El cost del material es divideix en dos grans grups, el de maquinari i el de programari:

Maquinari	Preu	Preu (amb IVA)
Estació de treball	250	290
Total	**250**	**290**

Taula 3: Cost del maquinari

Tot el programari excepte el sistema operatiu pel que es desenvolupa el projecte, Windows XP SP3, és gratuït.

Programari	Preu	Preu (amb IVA)
Servidor MySQL 5.0.32	Gratuït	Gratuït
Java SE	Gratuït	Gratuït
JFreeChart 1.0.12	Gratuït	Gratuït
JCalendar 1.3.3	Gratuït	Gratuït
JavaHelp 2.0.05	Gratuït	Gratuït
MySQL Connector 5.1.11	Gratuït	Gratuït
Eclipse IDE	Gratuït	Gratuït
ArgoUml	Gratuït	Gratuït
GanttProject	Gratuït	Gratuït
Total	**0**	**0**

Taula 4: Cost del programari

Els costos totals són els següents:

Tipus de cost	Preu	Preu (amb IVA)
Humà	18650	21634
Material – maquinari	250	290
Material – programari	0	0
Total	**18900**	**21924**

Taula 5: Cost total del projecte

3.3 Planificació temporal del projecte

La planificació temporal del projecte és una eina important per tenir un control adequat i el més precís possible sobre l'assoliment dels objectius marcats inicialment.

El primer dels diagrames temporals correspon a la planificació inicial del projecte. I mostra la previsió inicial del que es creia que requeririen cadascuna de les fases del projecte.

El segon diagrama mostra quina ha estat la realització real del projecte. El temps dedicat a les fases d'especificació, disseny s'ha adequat correctament a la previsió inicial.

La fase d'implementació és la que ha sofert més variacions que han provocat que incrementés el temps requerit per crear l'eina. Els següents punts expliquen els motius que han produït aquest increment temporal en la fase:

1. S'han hagut d'ampliar algunes de les classes de la llibreria gràfica *JFreeChart* per donar suport a les necessitats del sistema. La vista que ha requerit més modificacions de la llibreria de generació de gràfiques és la de la gràfica *TWA vs Datetime* per actualitzar la informació de les regates.

2. Durant les reunions de seguiment amb el client s'han detectat i acordat algunes millores o noves necessitats, com per exemple l'eliminació del biaix que apareix en intervals de temps menors a un minut entre dos canvis de signe de la variable *TWA* durant l'execució del *filtre de qualitat* sobre les dades o la incorporació de nous camps a les vistes per ampliar les seves funcionalitats.

Per últim, la fase de documentació s'ha vist afectada per motius aliens a la realització del projecte. Això a provocat que només es pogués desenvolupar la memòria i el manual d'usuari de l'eina durant els caps de setmana, endarrerint considerablement la finalització global del projecte.

3.3.1 Planificació inicial

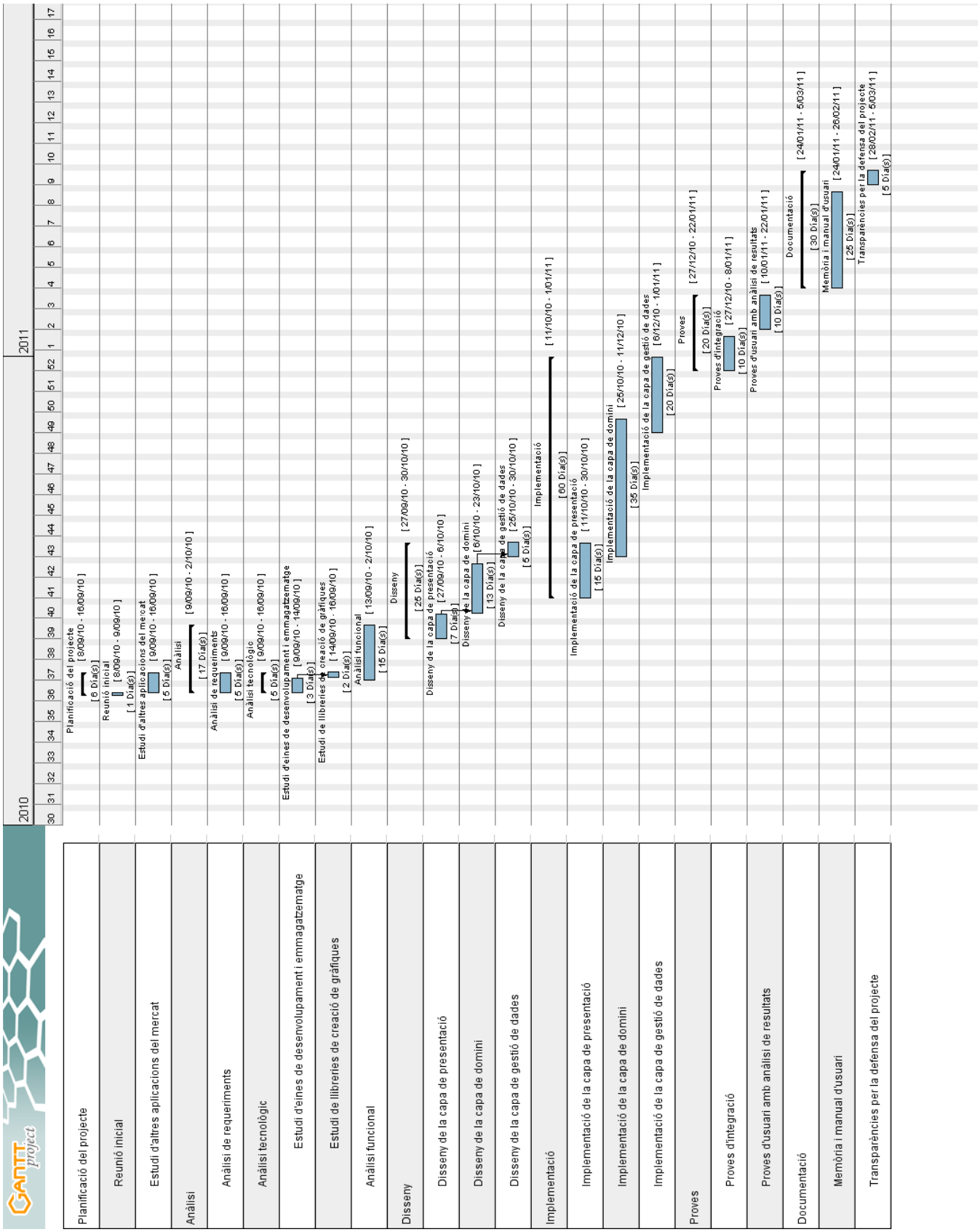


Figura 32: Planificació inicial del projecte

3.3.2 Planificació final

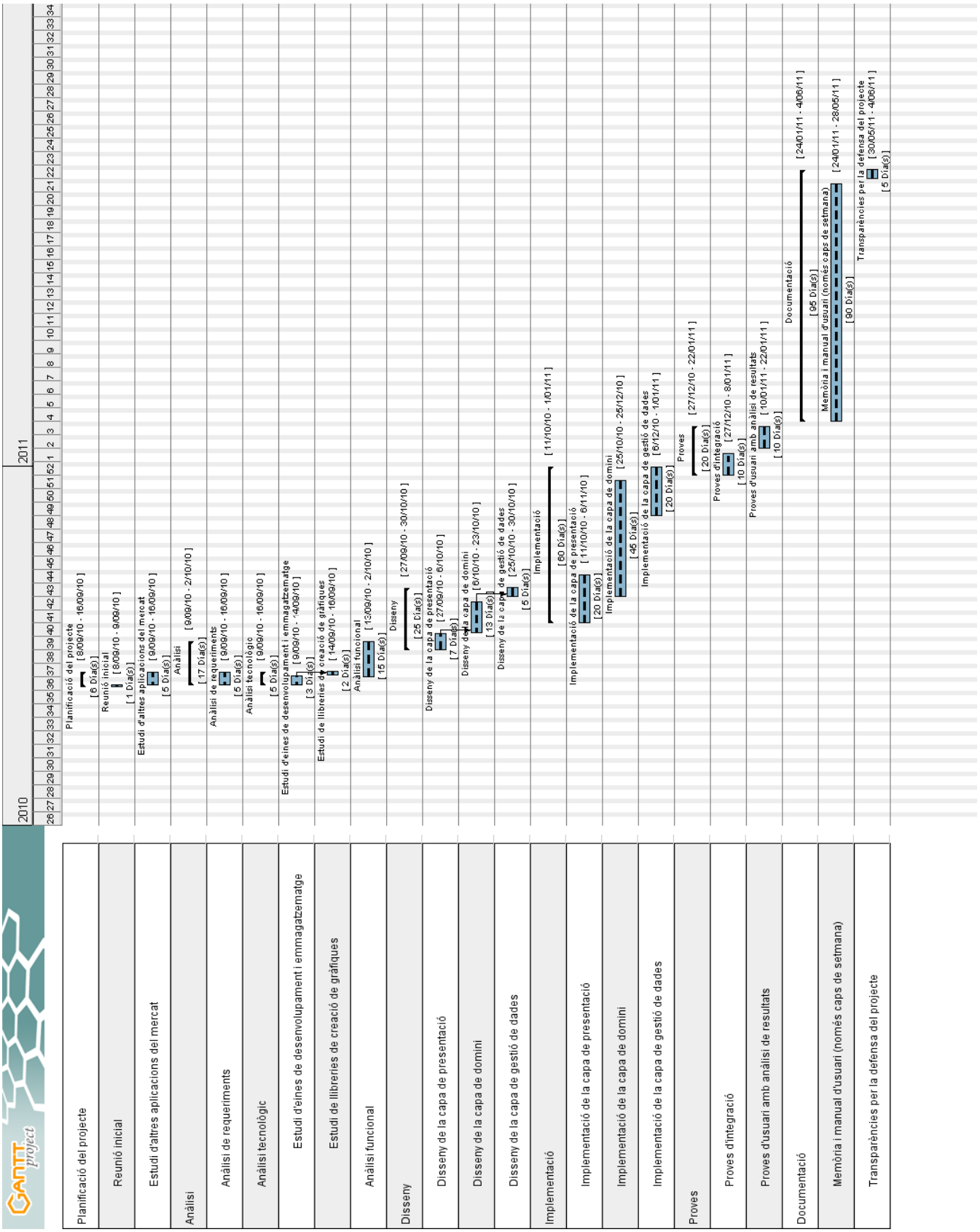


Figura 33: Planificació final del projecte

4 Anàlisi de Requeriments

L'anàlisi de requeriments és una de les diverses disciplines de l'enginyeria del programari que es preocupa de determinar els objectius, funcions i restriccions dels sistemes de maquinari i programari.

El capítol enregistra els requeriments determinats després de la comunicació, mitjançant entrevistes i reunions, amb el client i alhora usuari del sistema.

4.1 Requeriments de client

Les següents definicions de fet i assumpcions defineixen les expectatives del sistema en termes d'objectius, restriccions, etc.

4.1.1 Extensibilitat[1]

Les tecnologies escollides han de permetre que el sistema sigui fàcilment extensible.

4.1.2 Interfície d'usuari

No es desitja una interfície d'usuari en format web ni que involucri navegadors web. Ha de formar part d'una aplicació independent.

4.1.3 Propietat de les tecnologies

Totes les tecnologies utilitzades han de pertànyer al programari lliure.

4.1.4 Llenguatge de programació

A partir de la realització de l'estudi (no exhaustiu) dels llenguatges de programació i llibreries per la generació de gràfiques disponibles, i d'acord amb les preferències del client, s'escull *Java* com el llenguatge de desenvolupament.

4.1.5 Emmagatzematge persistent de la informació

El *SGBD* a utilitzar és *MySQL*[2]. L'elecció de la resta de tecnologia estarà lligada a la capacitat que tingui d'interactuar amb *MySQL*.

[1] Factor de qualitat del programari que consisteix en la facilitat d'adaptació del programari a nous requisits o canvis en l'especificació.
[2] *SGBD* que pertany al programari lliure i que forma part de l'arquitectura *LAMP* (*Linux Apache MySQL PHP*) per a diverses plataformes com *Unix*, *Linux*, *Windows*, etc.

4.2 Requeriments funcionals

Amb aquests requeriments expliquem les accions que ha de realitzar el sistema mitjançant les funcionalitats ofertes.

4.2.1 Importació de fitxers de *Log* amb informació de regates

És la funcionalitat principal d'entrada d'informació al sistema.

Els fitxers són plans i els genera una eina externa. L'extensió dels fitxers és *.log*. Sabent que una regata d'entrenament o de competició pot durar hores, i que els sensors del veler acostumen a enregistrar la informació cada segon[3], parlem de fitxers d'una mida considerable. Un valor aproximat del nombre de registres continguts seria de l'ordre de la desena de milers.

Cada registre de dades del fitxers correspon a un instant de temps determinat. Però el registre inicial és el de capçalera i descriu el nom dels camps continguts en cada registre de dades. Considerem com un requeriment que el primer registre de tot fitxer de *Log* sigui de capçalera, tot i que pot aparèixer de nou en altres registres dels fitxers.

El sistema és prou flexible per detectar si existeixen diferències entre la capçalera d'un fitxer i la informació considerada a la base de dades. Si es detecten nous camps en el fitxer s'ofereix la possibilitat d'afegir-los a la base de dades. Per altra banda, l'odre dels camps en el fitxer és irrellevant.

Tots els registres que continguin algun error de format s'obvien i s'informa de l'error al diari del sistema.

4.2.2 Conversió de dades durant la importació

Que els fitxer de *Log* siguin fitxers plans generats per aplicacions externes provoca que el format d'algunes de les dades pugui ser incompatible amb el sistema. Per això, durant la lectura dels registres es converteixen les dades que ho requereixin als formats que el sistema reconeix.

4.2.3 Preparació i tractament de les dades importades

Una vegada inserida una regata al sistema, l'usuari pot afinar la informació a partir d'una gràfica cronològica sobre l'angle real del vent. La gràfica permet seleccionar qualsevol instant de la regata i inicialitzar o actualitzar paràmetres com el nom del tram al que correspon la informació seleccionada, la vela major o la vela de proa utilitzades o també la qualitat de la informació que s'està tractant.

[3] La freqüència d'enregistrament dels sensors de l'embarcació és un paràmetre del programari de navegació, així que pot variar d'un fitxer a un altre.

4.2.4 Gravació persistent de les dades llegides dels fitxers de *Log*

La informació de les regates llegides dels fitxers de *Log* s'emmagatzema automàticament en la base de dades de l'eina. Tota actualització sobre les dades de navegació o de veles sempre es propaga a la base de dades.

4.2.5 Inserció, consulta i esborrat de veles

Són funcionalitats que el sistema ofereix per tal que l'usuari pugui gestionar i mantenir les veles que s'han utilitzat o s'utilitzen en les regates realitzades pels vaixells.

4.2.6 Manteniment del temps d'ús de les veles

Funcionalitat que el sistema calcula automàticament quan una vela es veu afectada per qualsevol acció que afecti al seu temps d'ús. La inserció, actualització o esborrat d'una regata implica l'actualització del temps d'ús de totes les veles implicades.

4.2.7 Consulta i esborrat de regates importades

El sistema permet a l'usuari consultar les regates existents i la data en que es van importar. Per altra banda, també existeix la possibilitat d'eliminar la informació d'una regata en el cas que es consideri oportú.

4.2.8 Diari del sistema

El sistema incorpora un diari d'errors, avisos i missatges informatius produïts mentre s'està executant.

Cada error que es detecta, bé per l'estat en que es troba la informació que s'incorpora al sistema, bé per un error intern de l'eina es grava en el diari amb la data i hora en que es produeix i el seu motiu.

4.2.9 Gràfiques d'anàlisi

Una altra funcionalitat del sistema és la que està orientada a l'anàlisi de les regates. L'eina aporta la creació de diversos tipus de gràfiques amb les que l'usuari pot estudiar el rendiment de l'embarcació.

Les gràfiques que actualment es poden generar amb aquesta funcionalitat són les següents:

1. Gràfiques cronològiques: Les gràfiques cronològiques serveixen per consultar el comportament de les variables d'estudi en funció del temps.

2. Gràfiques d'anàlisi de variables en funció de la velocitat real del vent: Permeten observar el comportament d'una variable en funció del *TWS*. Cada petició de generació d'una gràfica d'aquesta categoria implica l'execució implícita del *filtre TWS*.
   Si el sistema posseeix les dades objectiu de la variable d'estudi les mostra en la mateixa gràfica.

3. Gràfiques d'anàlisi de variables en funció d'una variable de domini diferent a la velocitat real del vent. També utilitzen el *filtre TWS*. Però el *filtre* no llegeix dades objectiu per a aquest tipus de gràfiques.

4. Gràfiques babord / estribord: Es poden realitzar gràfiques d'aquesta categoria per qualsevol tipus de gràfica d'anàlisi o cronològica. Permeten l'anàlisi de la informació en funció de si correspon a babord o estribord. La gràfica mostra la informació d'ambdues opcions alhora que incorpora les dades objectiu en el cas que existeixin.

El sistema permet a l'usuari, per a qualsevol de les gràfiques que es poden generar, seleccionar la informació a representar en funció de diversos criteris. Existeixen quatre criteris de selecció: tram de la regata, vela major usada, vela de proa usada i qualitat de les dades.

4.2.10 Filtres

1. *Filtre de qualitat* sobre les dades: Analitza les regates cercant les virades i trabujades de l'embarcació. Els instants immediatament anteriors i posteriors a l'execució d'aquestes maniobres es poden descartar en termes d'anàlisi de rendiment. Així, el sistema determina que les dades associades al temps que dura cada maniobra són de qualitat *Baixa*.

2. *Filtre* d'anàlisi de variables en funció de la velocitat real del vent: Calcula el valor mig que pren la variable d'estudi per cadascun dels intervals de la variable del domini que els sistema té en compte.

3. *Filtre babord / estribord*: Tracta per separat les dades que corresponen a babord de les d'estribord. El *filtre* es pot utilitzar alhora amb els anteriors.

4.2.11 Creació i esborrat de la taula de regates

El sistema ofereix la possibilitat d'esborrar i crear de nou les taules de regates de la base de dades. Permet una neteja completa del seu contingut en el cas que l'usuari així ho desitgi.

4.2.12 Creació i esborrat de la taula de veles

El sistema també dóna la possibilitat d'esborrar i crear de nou la taula de veles de la base de dades.

4.2.13 Fitxer bàsic de definició de la taula de regates de la base de dades

Fitxer tabulat amb la definició de les columnes de la taula de regates. Cada registre és tabulat i es descriu de manera bàsica el nom, el tipus i si parlem d'una columna clau de la taula.

Aquest fitxer s'utilitza en la funcionalitat de creació de la taula de regates.

4.2.14 Incorporació de fitxers de *targets* pels vaixells tractats en el sistema

El sistema és capaç de llegir els fitxer de *targets* de les variables d'estudi de qualsevol dels vaixells que s'estiguin tractant en el sistema. Quan l'usuari genera una gràfica d'anàlisi d'una variable en funció del *TW_speed* per un vaixell determinat el sistema llegeix el fitxer de *targets* associat i mostra la seva informació juntament amb les dades de la gràfica.

4.2.15 Manual d'usuari del sistema

Mostra a l'usuari les funcionalitats del sistema i com s'utilitzen. Incorpora imatges i exemples per facilitar la comprensió de l'eina.

4.3 Requeriments no funcionals

Conjunt de requisits que especifiquen els criteris amb els que valorar l'operació del sistema en el seu conjunt, més que els seus components específics.

4.3.1 Interfície d'usuari i factors humans

4.3.1.1 Tipus d'usuaris que interactuen amb el sistema

És una eina d'anàlisi de la informació obtinguda durant la navegació amb vaixells de regata d'alt nivell. Els usuaris als que pot interessar són tripulants de les embarcacions com el navegant o el tàctic per aprofitar l'ajuda que els resultats puguin donar en la presa de decisions tàctiques i la millora del rendiment. Els resultats dels anàlisis també són d'interès pels dissenyadors de vaixells o veles que necessiten conèixer el rendiment de certs paràmetres.

4.3.1.2 És important que l'eina sigui fàcil d'utilitzar

Totes les vistes de l'aplicació s'expressen en termes nàutics. Es tracta d'una eina específica, però senzilla d'utilitzar pels usuaris als que està destinada.

4.3.2 Documentació

4.3.2.1 Quins tipus de documents es necessiten

És necessari el manual d'usuari de l'eina per familiaritzar-se ràpida i fàcilment amb el seu ús.

4.3.2.2 Usuaris als que es dirigeix cada document

El manual d'usuari està destinat als usuaris que interactuaran amb el sistema.

4.3.3 Característiques de rendiment

4.3.3.1 Restriccions de velocitat, temps de resposta o capacitat de processament

Al no ser una eina d'anàlisi de la informació en temps real mentre és realitzen les regates, la velocitat i el temps de resposta no són requeriments imprescindibles. Si cal tenir en compte que es poden realitzar consultes sobre volums de dades considerables i que poden requerir temps de càlcul addicional en funció de la dificultat dels *filtres* a executar, repercutint en el temps de resposta de les consultes realitzades.

Per altra banda, la capacitat de processament si és important. Com més grans siguin els volums de dades a tractar més recursos de la màquina seran necessaris per obtenir un rendiment acceptable. Per exemple, la realització de consultes sobre un embarcació durant una regata requereix una capacitat de processament escassa. Però, si pel contrari, es desitja realitzar una consulta sobre totes les dades disponibles de totes les regates d'un vaixell, la capacitat de processament esdevindrà un factor transcendental.

4.3.4 Tractament d'errors i condicions extremes

4.3.4.1 Resposta del sistema davant d'errors d'entrada

L'entrada d'informació al sistema es realitza a partir de fitxers tabulats amb les dades recol·lectades de les regates. S'ha implementat certa tolerància a fallades en la lectura dels fitxers d'entrada. El sistema és capaç d'obviar els registres incorrectes informant de l'error detectat en el diari de l'eina, mentre importa tots els registres que són correctes.

L'usuari es capaç d'introduir i seleccionar informació mitjançant les vistes del sistema. S'han controlat els possibles errors derivats de seleccions o peticions mal realitzades per part de l'usuari.

4.3.4.2 Resposta del sistema davant de condicions extremes

Existeixen situacions de caràcter extrem, com la corrupció de la base de dades per motius externs, un error intern del *SGBD*, o un error intern de la mateixa eina, etc.

La informació amb la que treballa el programa no és única ni irrepetible i no estem parlant d'un sistema crític. El comportament esperat en condicions extremes només requereix que no es corrompi la base de dades, evitant a l'usuari que hagi de tractar de nou la informació prèviament processada.

4.3.5 Interfície del sistema

4.3.5.1 Entrades de sistemes externs al proposat

Els fitxers d'entrada que contenen la informació de les regates els genera un producte extern, encarregat de gravar en cada instant la informació que conté cadascun dels sensors instal·lats per la nau.

Un dels processos principals del nostre sistema s'encarrega d'importar el contingut dels fitxers cap a la base de dades.

4.3.5.2 Restriccions en el format de l'entrada dels sistemes externs

Els fitxers d'entrada són tabulats. El primer registre del fitxer conté la capçalera amb el nom de les variables enregistrades.

4.3.6 Factors de qualitat

4.3.6.1 Tolerància a fallides del sistema

El procés d'importació d'informació de regates és tolerant a fallides. Tots els registres erronis dels fitxers importats s'obvien, però el procés no s'atura i continua important la informació que si és correcta.

Els *filtres* implementats són processos interns del sistema, sobre els que l'usuari no té cap control. Són processos que manipulen les dades i dels que es requereix un funcionament sempre correcte. No són processos tolerants a fallides perquè un error d'execució és símptoma de mal funcionament, bé per dades corruptes bé per alguna errada d'implementació.

4.3.6.2 Portabilitat del sistema

Un dels punts forts de *Java* és la seva portabilitat. Qualsevol màquina amb la *Java Virtual Machine* instal·lada serà capaç d'executar el programari desenvolupat.

4.3.7 Temes de seguretat

4.3.7.1 Control d'accés a alguna dada o al propi sistema

L'usuari final no ha realitzat cap petició en aquest aspecte. Tanmateix, el *SGBD* permet el control d'accés a les dades emmagatzemades i el sistema l'utilitza mitjançant un usuari amb permisos específics.

Tot i que el sistema s'ha desenvolupat per a l'execució en local, la possibilitat d'estendre'l per a l'execució en remot (respecte les dades emmagatzemades) seria de cost relativament baix. El treball en remot podria aprofitar-se de manera més àmplia i efectiva del control d'accés.

4.3.8 Entorn de desenvolupament

4.3.8.1 Llenguatge de programació

El llenguatge elegit és *Java*. *L'apèndix C* mostra l'estudi sobre les tecnologies contrastades i els arguments que duen a l'elecció d'aquest llenguatge.

4.3.8.2 Ús de llibreries concretes

Quatre són les llibreries necessàries per donar suport al desenvolupament del sistema:

1. Gestió de calendaris i dates:
    a. S'usa la llibreria *JCalendar* 1.3.3.
2. Connector 5.1.11 per a *MySQL*.
3. Creació de gràfiques:
    a. S'usa la llibreria *JFreeChart* 1.0.12.
4. Creació del manual d'usuari del sistema:
    a. S'usa la llibreria *JavaHelp* 2.0.05.

4.3.8.3 Entorn

L'entorn d'implementació i d'execució del sistema és *Microsoft Windows XP*, però gràcies a que el llenguatge *Java* és multiplataforma tot entorn amb la *JVM* instal·lada serà capaç d'executar-lo.

4.3.8.4 Estàndards de desenvolupament

4.3.8.4.1 Mètodes d'anàlisi/disseny/implementació

La metodologia utilitzada per l'anàlisi i disseny del sistema és *UML*[4].

4.3.8.4.2 Normes de codificació

Totes les classes implementades han d'estar degudament documentades. La documentació inclou una capçalera que especifica l'objectiu de la classe i per a cada procediment una capçalera que el descriurà juntament amb els paràmetres d'entrada i sortida.

4.3.9 Consistència

El sistema ha de respondre a la interacció de l'usuari immediatament oferint informació sempre actualitzada alhora que ha de reflectir totes les accions d'aquest en el model.

[4] *Unified Modeling Language*. En català, Llenguatge unificat de modelat. És el llenguatge de modelat de sistemes de programari més conegut i usat en l'actualitat. És un llenguatge gràfic per visualitzar, especificar, dissenyar, construir i documentar un sistema.

5 Especificació del sistema

Anomenem especificació a la definició formal o semi-formal d'un sistema. Formen part de l'especificació la definició del diagrama de casos d'ús i l'especificació dels diversos casos d'ús del sistema.

Definim un cas d'ús com el document que descriu una seqüència d'esdeveniments que realitza un actor, o agent extern, que utilitza el sistema per dur a terme un procés que té algun valor per a ell. Podem dividir els escenaris d'un cas d'ús en tres categories segons la seva importància: el principal, que descriu els successos en condicions normals. Els alternatius que descriuen els comportaments en situacions concretes en les que no es possible dur a terme el procés normal. I per últim, els fallits que descriuen les situacions anòmales.

Un diagrama de casos d'ús mostra conjuntament els casos d'ús del sistema, els actors i les relacions entre ells.

El capítol detalla cadascun d'aquests elements per donar una visió formal del sistema que estem definint.

El capítol enregistra els requeriments determinats després de la comunicació, mitjançant entrevistes i reunions, amb el client i alhora usuari del sistema.

5.1 Diagrama de casos d'ús

El diagrama de casos d'ús del sistema mostra el següent:

1. L'usuari pot interaccionar amb totes les funcionalitats del sistema, tant de manera directa com indirecta. En el primer cas l'usuari usa casos d'ús com la importació de la informació d'una regata, la inserció d'una vela o la consulta del diari del sistema. La interacció indirecta és produeix quan un cas d'ús utilitza un altre cas d'ús com passa amb els *filtres* del sistema.

2. Només un usuari accedeix al sistema. Sempre amb els mateixos privilegis i restriccions.

Figura 34 : Diagrama de casos d'ús

5.2 Especificació de casos d'ús

5.2.1 Especificació del cas d'ús "Importar regata"

Cas d'ús: Importació d'una nova regata.

Actors: Usuari (iniciador), Sistema.

Propòsit: Emmagatzemar de manera persistent la informació de la nova regata.

Resum: L'usuari ha recol·lectat dades d'una regata realitzada amb un vaixell.
El sistema llegeix la informació, la tracta i l'emmagatzema persistentment.
Al finalitzar les dades estan disponibles per que l'usuari pugui analitzar-les.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari ha recol·lectat les dades d'una regata en un fitxer de *Log* i indica al sistema que desitja importar-lo.	
	2. El sistema mostra la vista d'importació d'una regata a l'usuari.
3. L'usuari informa el nom del vaixell.	
4. L'usuari informa el nom de la regata.	
5. L'usuari informa la vela major utilitzada durant la regata.	
6. L'usuari informa el fitxer de *Log* que conté les dades de la regata.	
	7. El sistema detecta que tots els camps obligatoris s'han introduït correctament i activa els camps opcionals de la vista alhora que activa el botó d'inici de la importació.
8. Si ho desitja, l'usuari informa la qualitat que s'assignarà per defecte a totes les dades de la regata.	
9. Si ho desitja, l'usuari informa un comentari qualsevol relacionat amb la regata.	
10. L'usuari prem el botó *Acceptar* per iniciar el procés d'importació	
	11. Inicialment, el sistema detecta les diferències entre el nombre de camps

existents al fitxer i el nombre de columnes de la base de dades:
   a. **If** noves columnes **initiate** inserir noves columnes
   b. **If** menys columnes que a la base de dades **then** s'insereixen els valors d'aquestes columnes i nuls a les restants

12. El sistema recorre el fitxer de *Log* inserint cadascun dels registres a la base de dades (els camps que estiguin en un format incompatible amb el sistema es formaten).
    Alhora que insereix els registres el sistema actualitza l'estat de la barra de progrés de la importació, visible a la vista.
13. Inserits tots els registres, el sistema informa a l'usuari de que la importació a finalitzat correctament i li pregunta si desitja aplicar el *filtre de qualitat* sobre les dades.
14. L'usuari decideix si vol executar el *filtre de qualitat* sobre les dades:
    a. **If** executar *filtre* **initiate** *Filtre de qualitat*
15. L'usuari està satisfet perquè ha incorporat una nova regata a l'eina i podrà analitzar-la.

Cursos alternatius

Qualsevol línia: Si l'usuari cancel·la la importació de la regata es retrocedeixen les accions realitzades tornant el sistema a l'estat previ al inici del cas d'ús. La base de dades es manté consistent i s'elimina qualsevol referència a la informació tractada durant la importació.

Línia 7: El sistema comprova els camps obligatoris de la vista i detecta que hi ha camps sense informar. Mentre no s'hagin informat tots els camps obligatoris el sistema no permet iniciar la importació de la regata.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del cas d'ús s'avorta el procés i s'escriu al diari del sistema. El sistema comunica a l'usuari que la importació ha finalitzat incorrectament donant-li la possibilitat de consultar els errors en el diari del sistema.

Línia 11: Es detecta un error greu en el format dels registres del fitxer. El sistema avorta la importació i ho comunica a l'usuari. Escriu, també, al diari del sistema l'error que s'ha produït.

Línia 12: Es detecten errors de format en camps d'alguns registres. Els registres no s'insereixen a la base de dades i el sistema escriu els errors al diari del sistema. Finalitzada la importació, el sistema comunica a l'usuari que s'han detectat errors donant-li la possibilitat de consultar-los en el diari del sistema.

Si l'usuari ho desitja, pot editar el fitxer i solucionar els errors descrits al diari del sistema. Si torna a importar el fitxer de la regata el sistema incorporarà només els registres corregits a la base de dades i comunicarà a l'usuari que la importació ha finalitzat correctament preguntant-li finalment si desitja aplicar el *filtre de qualitat* sobre les dades.

5.2.2 Especificació del cas d'ús "Inserir noves columnes"

Cas d'ús: Inserció de noves columnes a la base de dades.

Actors: Usuari (iniciador), Sistema.

Propòsit: Inserció de les noves columnes detectades en el fitxer de *Log* de la regata a la base de dades.

Resum: Si l'algoritme d'importació detecta que el fitxer amb les dades de la regata conté informació d'un nou sensor (que fins ara no estava contemplat a la base de dades del sistema) dóna l'oportunitat d'incorporar les dades del nou sensor com una nova columna de la base de dades.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
	1. El cas d'ús comença quan el sistema comunica a l'usuari que s'han detectat noves columnes i li pregunta si desitja inserir-les a la base de dades.
2. L'usuari decideix si vol inserir les noves columnes: a. **If** inserir columnes **see section** inserir noves columnes b. **If** ignora inserir columnes **see section** no inserir noves columnes	
3. L'usuari està satisfet i continua amb la importació de la regata.	

Section inserir noves columnes

Accions dels actors	Resposta del sistema
1. L'usuari indica al sistema que desitja inserir les noves columnes.	
	2. El sistema mostra a l'usuari la vista d'inserció de noves columnes amb la relació de columnes a definir per afegir-les a la base de dades.
3. L'usuari informa el tipus de dades per cada columna de la relació. La definició del tipus de dades la pot realitzar indicant que ha de ser del mateix tipus que una columna existent a la base de dades, o bé indicant el tipus de dades i definint la dimensió i decimals.	
	4. Quan el sistema detecta que l'usuari ha informat totes les columnes activa el botó d'inserció de les noves columnes.
5. L'usuari prem el botó *Acceptar*	
	6. El sistema insereix les noves columnes a la base de dades. Quan la base de dades està actualitzada es continua amb la importació de les dades de la regata.

Cursos alternatius

Qualsevol línia: Si l'usuari cancel·la la definició dels tipus de dades de les noves columnes es cancel·la la importació de la regata.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del cas d'ús s'avorta el procés i s'escriu l'error al diari del sistema. El sistema comunica a l'usuari que la importació a finalitzat incorrectament donant-li la possibilitat de consultar l'error en el diari del sistema.

Section no inserir noves columnes

Accions dels actors	Resposta del sistema
1. L'usuari indica al sistema que no desitja inserir les noves columnes.	
	2. El sistema continua amb la importació

de les dades descartant les noves columnes en el procés.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del cas d'ús s'avorta el procés i s'escriu l'error al diari del sistema. El sistema comunica a l'usuari que la importació a finalitzat incorrectament donant-li la possibilitat de consultar l'error en el diari del sistema.

5.2.3 Especificació del cas d'ús "Filtre de qualitat"

Cas d'ús: Filtre de qualitat sobre les dades importades d'una regata.

Actors: Usuari (iniciador), Sistema.

Propòsit: Assignar qualitat de les dades *Baixa* als instants que duren les maniobres de l'embarcació durant la regata.

Resum: Analitza la regata cercant les virades i trabujades de l'embarcació. Els instants immediatament anteriors i posteriors a l'execució d'aquestes maniobres es poden descartar en termes d'anàlisi de rendiment.
Així, el sistema determina que les dades associades al temps que dura cada maniobra són de qualitat *Baixa*.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari indica al sistema que desitja executar el *filtre*.	
	2. El sistema executa l'algoritme de cerca de maniobres (canvis de signe en la variable TWA) assignant a les dades dels instants anteriors i posteriors al canvi de signe qualitat *Baixa*. Alhora que es tracten les maniobres implicades el sistema actualitza l'estat de la barra de progrés del *filtre de qualitat*, visible a la vista d'importació d'una regata.
	3. Quan l'algoritme ha tractat totes les maniobres afectades, el sistema comunica a l'usuari que el *filtre* ha finalitzat correctament.

4. L'usuari està satisfet perquè les dades ja han passat el primer *filtre* de l'eina i són més útils per realitzar-ne l'anàlisi.

Cursos alternatius

Qualsevol línia: Si l'usuari cancel·la l'execució del *filtre de qualitat* sobre les dades es retrocedeixen les accions realitzades tornant el sistema a l'estat previ al inici del cas d'ús. La base de dades es manté consistent i s'elimina qualsevol referència a la informació tractada durant l'execució del *filtre*.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del *filtre* el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

5.2.4 Especificació del cas d'ús "Visualitzar regates"

Cas d'ús: Es mostren les regates importades al sistema.

Actors: Usuari (iniciador), Sistema.

Propòsit: Obtenir informació actualitzada de les regates importades al sistema.

Resum: Es mostra una relació amb les regates emmagatzemades a la base de dades del sistema. Es pot consultar per cada regata el seu nom, data d'inici i de fi, el nom del veler amb el que es va realitzar i la data en la que s'importaven les dades al sistema.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja consultar les regates existents.	
	2. El sistema mostra a l'usuari una relació de les regates existents i la informació associada a cadascuna d'elles. Per cada regata es mostren el nom del vaixell, el nom de la regata, les dates d'inici i fi i per últim la data en que s'ha realitzat la importació de la regata.

3. L'usuari està satisfet perquè coneix les regates existents en el sistema.

Cursos fallits

Línia 2: Si es produeix un error intern durant l'obtenció de la relació de regates el sistema ho comunica a l'usuari i escriu l'error al diari del sistema.

5.2.5 Especificació del cas d'ús "Esborrar regates"

Cas d'ús: Esborrat de les regates seleccionades del sistema.

Actors: Usuari (iniciador), Sistema.

Propòsit: Esborrar les regates seleccionades a partir d'una relació amb totes les regates existents en el sistema.

Resum: Es mostra una relació amb les regates emmagatzemades a la base de dades del sistema. L'usuari pot seleccionar una o més regates per esborrar-les del sistema.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja esborrar regates del sistema.	
	2. El sistema mostra a l'usuari una relació de les regates existents i la informació associada a cadascuna d'elles. Per cada regata es mostren el nom del vaixell, el nom de la regata i les seves dates d'inici i fi.
3. L'usuari selecciona les regates que desitja esborrar.	
	4. El sistema comunica a l'usuari que està apunt d'esborrar definitivament les regates seleccionades i que no podrà recuperar la informació.
5. L'usuari decideix si desitja esborrar les regates seleccionades: a. **If** esborrar **see section** esborrar regates b. **If** no esborrar **see section** no esborrar regates	

6. L'usuari està satisfet i ha esborrat les regates que ja no li eren necessàries.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del cas d'ús el sistema l'escriu al diari del sistema i li comunica a l'usuari donant-li la possibilitat de consultar-lo.

Section esborrar regates

Accions dels actors	Resposta del sistema
1. L'usuari indica al sistema que desitja esborrar les regates seleccionades	
	2. El sistema esborra tota la informació que fa referència a les regates seleccionades. Un cop esborrades, comunica a l'usuari que s'han eliminat correctament.

Cursos fallits

Línia 2: Si es produeix un error intern durant la creació del missatge informatiu sobre la correcta eliminació de les regates el sistema ho comunica a l'usuari i escriu l'error al diari del sistema.

Section no esborrar regates

Accions dels actors	Resposta del sistema
1. L'usuari indica al sistema que no desitja esborrar les regates seleccionades	
	2. El sistema no esborra les regates i torna a mostrar a l'usuari una relació de les regates existents i la informació associada a cadascuna d'elles.

5.2.6 Especificació del cas d'ús "Inserir veles"

Cas d'ús: Inserció de veles segons el seu tipus.

Actors: Usuari (iniciador), Sistema.

Propòsit: Inserir noves veles al sistema.

Resum: Permet la inserció de noves veles majors o de proa. Si la taula de regates ja

contenia informació de l'ús de la vela inserida el sistema actualitza el temps d'ús de la vela automàticament.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja inserir una o més veles.	
2. L'usuari informa el nom de la vela.	
3. L'usuari informa el tipus de vela.	
	4. El sistema insereix la vela a la base de dades. Si ja existia informació sobre la vela a la taula de regates (la vela ja s'havia usat anteriorment en el sistema) s'actualitzen les seves estadístiques. Finalment el sistema comunica a l'usuari que la inserció ha finalitzat correctament.
5. L'usuari està satisfet perquè la vela ja forma part del sistema.	

Cursos fallits

Línia 3: Si es produeix un error intern durant la inserció de la vela a la base de dades el sistema ho comunica a l'usuari i escriu l'error al diari del sistema.

5.2.7 Especificació del cas d'ús "Visualitzar veles"

Cas d'ús: Es mostren les veles existents en el sistema.

Actors: Usuari (iniciador), Sistema.

Propòsit: Obtenir informació actualitzada de les veles existents en el sistema.

Resum: Es mostra una relació amb les veles emmagatzemades a la base de dades del sistema. Es pot consultar per cada vela el seu nom, tipus, data en la que es comença a utilitzar i el temps d'ús total.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li	

indica al sistema que desitja consultar les veles existents.	
	2. El sistema mostra a l'usuari una relació de les veles existents i la informació associada a cadascuna d'elles. Per cada vela es mostren el nom, el tipus, i tenint en compte la informació disponible a la base de dades la data en que es comença a utilitzar i el temps d'ús total.
3. L'usuari està satisfet perquè coneix les veles existents en el sistema.	

Cursos fallits

Línia 2: Si es produeix un error intern durant l'obtenció de la relació de veles el sistema ho comunica a l'usuari i escriu l'error al diari del sistema.

5.2.8 Especificació del cas d'ús "Esborrar veles"

Cas d'ús: Esborrat de les veles seleccionades del sistema.

Actors: Usuari (iniciador), Sistema.

Propòsit: Esborrar les veles seleccionades a partir d'una relació amb totes les veles existents en el sistema.

Resum: Es mostra una relació amb les veles emmagatzemades a la base de dades del sistema. L'usuari pot seleccionar una o més veles per esborrar-les del sistema.
Quan una vela s'esborra del sistema s'elimina de la taula de veles, però la informació associada a la l'ús de la vela en la taula de regates es manté.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja esborrar veles del sistema.	
	2. El sistema mostra a l'usuari una relació de les veles existents i la informació associada a cadascuna d'elles. Per cada vela es mostra el nom i el seu tipus.

3. L'usuari selecciona les veles que desitja esborrar.

4. El sistema comunica a l'usuari que està apunt d'esborrar definitivament les veles seleccionades i que no podrà recuperar la informació.

5. L'usuari decideix si desitja esborrar les veles seleccionades:
   a. **If** esborrar **see section** esborrar veles
   b. **If** no esborrar **see section** no esborrar veles
6. L'usuari està satisfet i ha esborrat les veles que ja no li eren necessàries.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del cas d'ús el sistema l'escriu al diari del sistema i li comunica a l'usuari donant-li la possibilitat de consultar-lo.

Section esborrar veles

Accions dels actors	Resposta del sistema
1. L'usuari prem el botó *Acceptar*.	
	2. El sistema esborra la informació que fa referència a les veles seleccionades en la taula de veles, però manté les dades associades en la taula de regates. Un cop esborrades, comunica a l'usuari que s'han eliminat correctament.

Cursos fallits

Línia 2: Si es produeix un error intern durant l'esborrat de les veles seleccionades el sistema escriu l'error al diari del sistema i ho comunica a l'usuari.

Section no esborrar veles

Accions dels actors	Resposta del sistema
1. L'usuari indica al sistema que no desitja esborrar les veles seleccionades	
	2. El sistema no esborra les veles i torna a mostrar a l'usuari una relació de les veles existents i la informació

associada a cadascuna d'elles.

5.2.9 Especificació del cas d'ús "Gràfica TWA vs Time"

Cas d'ús: Generació de la gràfica "TWA vs Time".

Actors: Usuari (iniciador), Sistema.

Propòsit: Creació de la gràfica "TWA vs Time" a partir de les dades importades d'una regata. La gràfica ofereix a l'usuari la possibilitat de seleccionar parts de la regata i actualitzar per cada selecció realitzada els següents paràmetres: tram de la regata, vela major usada, vela de proa usada o qualitat de les dades.

Resum: A partir de la gràfica cronològica de *TWA* de la regata i una representació colorada per a qualsevol dels quatre paràmetres: tram de la regata, vela major usada, vela de proa usada i qualitat de les dades, l'usuari pot modificar la informació realitzant seleccions específiques en funció del temps.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja actualitzar les dades d'una regata a partir de la seva gràfica "TWA vs Time".	
	2. El sistema mostra a l'usuari una vista prèvia a la gràfica per determinar quina regata mostrar. Els camps de la vista prenen per defecte els valors de l'última importació realitzada.
3. L'usuari modifica, si ho desitja, el nom del vaixell.	
4. L'usuari modifica, si ho desitja, la data d'inici de la regata.	
5. L'usuari modifica, si ho desitja, la data de fi de la regata	
6. L'usuari prem el botó *Acceptar* per obtenir la gràfica.	
	7. El sistema genera la gràfica cronològica de *TWA*. Esta formada per dues gràfiques. La primera és la de selecció i actualització de les dades i

correspon a la *TWA vs Datetime*. La segona mostra per a cadascuna de les variables que es poden actualitzar de la regata, quins valors prenen en funció del temps.

8. L'usuari decideix en quin mode realitzar cadascuna de les seleccions de dades de la gràfica:
   a. **If** selecció amb *zoom* **see section** *zoom* de la zona
   b. **If** desfer *zoom* **see section** desfer *zoom*
   c. **If** actualització **see section** actualitzar la zona
9. L'usuari està satisfet perquè ha actualitzat la informació de totes les zones de la gràfica que desitja.

Cursos alternatius

Línies 3, 4, 5: Si l'usuari cancel·la el cas d'ús no es realitza cap acció.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del cas d'ús el sistema l'escriu al diari del sistema i li comunica a l'usuari donant-li la possibilitat de consultar-lo.

Section *zoom* de la zona

Accions dels actors	Resposta del sistema
1. L'usuari selecciona l'opció de *zoom*.	
2. L'usuari selecciona la zona en la primera gràfica.	
	3. El sistema realitza un *zoom* sobre la selecció mostrant amb més detall la informació.

Section desfer *zoom*

Accions dels actors	Resposta del sistema
1. L'usuari realitza el moviment invers al de les seleccions per *zoom*.	
	2. El sistema desfà el *zoom* mostrant de nou la gràfica completa.

Section actualitzar la zona

Accions dels actors	Resposta del sistema
1. L'usuari selecciona, amb modalitat d'actualització, zones de la primera gràfica.	
	2. El sistema mostra la vista d'actualització.
3. Si l'usuari ho desitja informa el camp Tram.	
4. Si l'usuari ho desitja informa el camp Vela major.	
5. Si l'usuari ho desitja informa el camp Vela de proa.	
6. Si l'usuari ho desitja informa el camp Qualitat	
7. L'usuari prem el botó *Acceptar*.	
	8. El sistema actualitza la informació de la base de dades i les dues gràfiques que s'estan mostrant pels camps que hagin canviat de valor.

Cursos alternatius

Línies 3, 4, 5, 6: Si l'usuari prem el botó *Cancel·lar* de la vista no es realitza cap acció d'actualització sobre les dades seleccionades.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del cas d'ús el sistema l'escriu al diari del sistema i li comunica a l'usuari donant-li la possibilitat de consultar-lo.

5.2.10 Especificació del cas d'ús "Gràfiques X vs múltipes Y"

Cas d'ús: Generació de gràfiques d'anàlisi "X vs múltiples Y".

Actors: Usuari (iniciador), Sistema.

Propòsit: Creació de gràfiques d'anàlisi amb un domini i un o més rangs.

Resum: És possible generar gràfiques cronològiques d'una o més variables alhora, gràfiques d'anàlisi d'una variable qualsevol en funció del *TWS* o gràfiques d'una variable qualsevol contra qualsevol altra.
També és possible especificar, si es desitja, distinció en les gràfiques de les

dades de babord i d'estribord.
El sistema permet, en la generació de qualsevol de les gràfiques, afegir criteris de selecció addicionals (tram, vela major, vela de proa i qualitat de les dades). Addicionalment, un i només un d'aquest criteris pot contenir una selecció de múltiples valors, mentre que la resta queden restringits a la selecció individual. El criteri de qualitat de les dades és l'únic que no admet selecció múltiple dels seus valors.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja analitzar certs aspectes de les dades d'una regata a partir d'una gràfica "Gràfica X vs múltiples Y".	
	2. El sistema mostra a l'usuari la vista amb els paràmetres a definir per generar la gràfica desitjada. El sistema informa per defecte el nom del vaixell i les dates d'inici i fi de la gràfica amb les de l'última importació realitzada.
3. L'usuari informa el nom del vaixell.	
4. L'usuari informa si ho desitja la casella de verificació de *globalitat* per indicar que es tinguin en compte totes les dades existents del vaixell.	
5. L'usuari informa la data d'inici, si no ha informat la casella de verificació *global*.	
6. L'usuari informa la data de fi, si no ha informat la casella de verificació *global*.	
7. L'usuari informa la variable que farà de domini de la gràfica. a. **If** gràfica cronològica **then** l'usuari selecciona la variable *Datetime* b. **If** gràfica d'anàlisi en funció del *TWS* **then** l'usuari selecciona la variable *TW_speed* c. **else** l'usuari selecciona qualsevol altra variable disponible per generar una altra gràfica d'anàlisi.	

8. L’usuari informa la/les variables que es mostraran en funció del domini.
9. Si la variable de domini escollida és *TW_speed* o qualsevol altre que no sigui Datetime, l’usuari informa, si ho desitja, la casella de verificació *Contra vent*.
10. L’usuari informa, si ho desitja, la casella de verificació *babord/estribord*.
11. L’usuari informa, si ho desitja, el títol de la gràfica
12. L’usuari selecciona, si ho desitja, un o més d’un tram pels que mostrar les dades.
    Només es pot realitzar la selecció de múltiples trams si no existeix una selecció múltiple prèvia d’un altre dels camps opcionals.
13. L’usuari selecciona, si ho desitja, una o més d’una vela major per les que mostrar les dades.
    Només es pot realitzar la selecció de múltiples veles majors si no existeix una selecció múltiple prèvia d’un altre dels camps opcionals.
14. L’usuari selecciona, si ho desitja, la vela de proa per la que mostrar les dades.
    Només es pot realitzar la selecció de múltiples veles de proa si no existeix una selecció múltiple prèvia d’un altre dels camps opcionals.
15. L’usuari selecciona, si ho desitja, la qualitat de les dades per la que mostrar les dades.
    Aquest camp no admet selecció múltiple.
16. L’usuari prem el botó *Acceptar*.
17. El sistema executa el *filtre* adequat sobre les dades en funció del tipus de gràfica i els paràmetres escollits:
    a. **If** gràfica cronològica **initiate** *filtre genèric*
    b. **If** gràfica cronològica amb babord/estribord **initiate** *filtre*

genèric babord/estribord

c. **If** gràfica *TWS* o una altra que no sigui cronològica **initiate** *filtre TW_speed*
d. **If** gràfica *TWS* amb babord/estribord o una altra amb babord/estribord que no sigui cronològica **initiate** *filtre TW_speed amb babord/estribord*

18. El sistema genera i mostra la gràfica.

19. L'usuari està satisfet perquè ha obtingut la gràfica d'anàlisi que desitjava.

Cursos alternatius

Línies 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15: Si l'usuari prem el botó *Cancel·lar* de la vista no es realitza cap acció d'actualització sobre les dades seleccionades.

Cursos fallits

Línia 18: Si es produeix un error intern durant l'obtenció de les dades per la generació de la gràfica el sistema l'escriu al diari del sistema i ho comunica a l'usuari.

Línia 18: Si es produeix un error intern en la generació de la gràfica el sistema l'escriu al diari del sistema i ho comunica a l'usuari.

5.2.11 Especificació del cas d'ús "Filtre TW_speed"

Cas d'ús: Filtre TW_speed sobre les dades seleccionades.

Actors: Sistema (iniciador).

Propòsit: L'objectiu principal del *filtre* és preparar la informació per poder generar la gràfica d'estudi de la variable escollida en funció de la velocitat real de vent.
Tanmateix, si la variable escollida pel domini no correspon a la velocitat real del vent, s'executen els mateixos càlculs, però sense llegir cap fitxer de *targets*.

Resum: El *filtre* tracta la informació amb un conjunt de càlculs específics per obtenir finalment per cada rang de valors de la variable del domini un valor mig de la variable d'estudi.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema

1. El cas d'ús comença quan el sistema detecta que s'ha d'executar el *filtre* per generar una gràfica d'anàlisi respecte la variable del domini.
2. El sistema obté les dades a tractar.
3. **If** gràfica d'anàlisi *TWS* **then** el sistema obté, si existeixen per la variable d'estudi i vaixell demanat, els valors objectiu donats pel dissenyador.
4. El sistema executa l'algoritme del *filtre* que calcula el valor mig que resulta de la variable d'estudi per cada interval obtingut de la variable del domini.
5. El *filtre* retorna les dades tractades i les dades d'objectiu per comparar-les en la gràfica.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del *filtre* el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

Línia 2, 3: Si es produeix un error intern durant l'obtenció de les dades el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

5.2.12 Especificació del cas d'ús "Filtre TW_speed amb babord/estribord"

Cas d'ús: Filtre de babord/estribord sobre les dades del *filtre TW_speed*.

Actors: Sistema (iniciador).

Propòsit: Executar el *filtre TW_speed* per babord i per estribord.

Resum: Es separen els valors de la variable d'estudi en funció de si són de babord o d'estribord (a partir de la variable *TWA*). Per cadascun dels conjunts s'executa el *filtre TW_speed*.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
	1. El cas d'ús comença quan el sistema detecta que s'ha d'executar el *filtre* per generar una gràfica d'anàlisi respecte

	la variable del domini amb valors per babord i estribord del rang.
	2. El sistema obté les dades a tractar.
	3. **If** gràfica d'anàlisi *TWS* **then** el sistema obté, si existeixen per la variable d'estudi i vaixell demanat, els valors objectiu donats pel dissenyador.
	4. El sistema executa l'algoritme del *filtre* que separa les dades segons siguin de babord o estribord i aplica el *filtre TW_speed* per a cada conjunt.
	5. El *filtre* retorna les dades tractades i les dades d'objectiu per comparar-les en la gràfica.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del *filtre* el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

Línia 2, 3: Si es produeix un error intern durant l'obtenció de les dades el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

5.2.13 Especificació del cas d'ús "Filtre genèric babord/estribord"

Cas d'ús: Filtre X vs Y de babord/estribord sobre un conjunt qualsevol de dades.

Actors: Sistema (iniciador).

Propòsit: Separar les dades segons valors de babord i valors d'estribord.

Resum: Es separen els valors de la variable d'estudi en funció de si són de babord o d'estribord (a partir de la variable *TWA*). Es genera la gràfica demanada per cadascun dels conjunts generats.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
	1. El cas d'ús comença quan el sistema detecta que s'ha d'executar el *filtre* per generar una gràfica amb valors agrupats segons siguin de babord o estribord.

	2. El sistema obté les dades a tractar. 3. El sistema executa l'algoritme del *filtre* que separa les dades segons siguin de babord o estribord. 4. El *filtre* retorna les dades tractades.

Cursos fallits

Qualsevol línia: Si es produeix un error intern durant l'execució del *filtre* el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

Línia 2, 3: Si es produeix un error intern durant l'obtenció de les dades el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

5.2.14 Especificació del cas d'ús "Filtre genèric"

Cas d'ús: Filtre X vs Y sobre un conjunt qualsevol de dades.

Actors: Sistema (iniciador).

Propòsit: Obté les dades a partir dels paràmetres de la vista.

Resum: *Filtre* que obté el conjunt de dades per generar la gràfica on X correspon al domini Y és el conjunt de variables que formen el rang.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
	5. El cas d'ús comença quan el sistema detecta que s'ha d'executar el *filtre* per generar una gràfica cronològica. 6. El sistema obté les dades. 7. Retorna les dades.

Cursos fallits

Línia 6: Si es produeix un error intern durant l'obtenció de les dades el sistema l'escriu al diari del sistema i li comunica a l'usuari que el *filtre* a finalitzat incorrectament donant-li la possibilitat de consultar-lo.

5.2.15 Especificació del cas d'ús "Diari del sistema"

Cas d'ús: Consulta del diari del sistema.

Actors: Usuari (iniciador), Sistema.

Propòsit: Consultar els missatges informatius, d'avís i errors del diari del sistema.

Resum: La consulta del diari del sistema serveix a l'usuari per conèixer tots els missatges que el sistema genera automàticament durant la seva execució. Es mantenen missatges informatius, missatges d'avís i missatges d'error.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja consultar el diari del sistema (bé per decisió pròpia, bé perquè l'eina li ha ofert la possibilitat de consultar errors produïts durant l'execució d'algun cas d'ús previ).	
	2. El sistema mostra a l'usuari la vista del diari del sistema. El sistema informa per defecte la data inicial sobre la que consultar la informació amb la data de l'última arrencada de l'eina.
3. L'usuari informa la data inicial sobre la que consultar la informació si ho desitja.	
	4. El sistema mostra tota la informació a partir de l'última arrencada on està inclosa la data seleccionada.
5. L'usuari està satisfet perquè pot consultar els missatges generats pel sistema a partir de la data demanada.	

5.2.16 Especificació del cas d'ús "Crear taula de regates"

Cas d'ús: Creació de la taula de regates de la base de dades.

Actors: Usuari (iniciador), Sistema.

Propòsit: Creació de la taula de regates.

Resum: El cas d'ús permet crear la taula regates a la base de dades sempre i quan

no existeixi. Implícitament el procés de creació de la taula dóna d'alta també la taula d'importacions (taula interna del sistema).

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja crear la taula de regates.	
	2. El sistema mostra la vista de creació de la taula regates.
3. L'usuari informa el nom del fitxer que conté la definició de la taula regates.	
	4. El sistema crea primer la taula *regates* a la base de dades. Després crea la taula *importacions*. Un cop existeixen les dues taules comunica a l'usuari que el procés de creació a finalitzat correctament.
5. L'usuari està satisfet perquè a creat la taula regates a la base de dades.	

Cursos alternatius

Línia 3: Si alguna de les taules existeix no es creen de nou i el sistema ho comunica a l'usuari.

Cursos fallits

Línia 3: Si el fitxer de definició de la taula regates és incorrecte el sistema ho comunica a l'usuari donant-li la possibilitat de consultar els errors al diari del sistema.

Línia 3: Si es produeix un error intern durant la creació de les taules el sistema l'escriu al diari del sistema i ho comunica a l'usuari.

5.2.17 Especificació del cas d'ús "Esborrar taula de regates"

Cas d'ús: Eliminació de la taula regates de la base de dades.

Actors: Usuari (iniciador), Sistema.

Propòsit: Eliminació de la taula regates.

Resum: El cas d'ús permet l'esborrat de la taula regates de la base de dades sempre i quan aquesta existeixi. Implícitament el procés d'esborrat de la taula elimina la taula d'importacions.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja esborrar la taula de *regates*.	
	2. El sistema informa a l'usuari de que es perdran totes les dades i li pregunta si està segur de que desitja esborrar-la.
3. L'usuari decideix si desitja esborrar la taula regates: a. **if** esborrar **see section** esborrar taula b. **if** no esborrar **see section** no esborrar taula	
4. L'usuari està satisfet perquè, si ho desitjava, ha esborrat la taula regates de la base de dades.	

Section esborrar taula

Accions dels actors	Resposta del sistema
1. L'usuari prem el botó *Acceptar* per esborrar la taula.	
	2. El sistema esborra la taula importacions i després la taula regates de la base de dades.
	3. El sistema comunica a l'usuari que el procés d'esborrat a finalitzat correctament.

Cursos fallits

Línia 2: Si alguna de les taules no existeix s'atura el procés d'esborrat i el sistema ho comunica a l'usuari.

Línia 2: Si es produeix un error intern durant l'esborrat de les taules el sistema l'escriu al diari del sistema i ho comunica a l'usuari.

Section no esborrar taula

Accions dels actors	Resposta del sistema
1. L'usuari prem el botó *Cancel·lar* per no esborrar la taula.	
	2. El sistema no realitza cap acció.

5.2.18 Especificació del cas d'ús "Crear taula de veles"

Cas d'ús: Creació de la taula de veles a la base de dades.

Actors: Usuari (iniciador), Sistema.

Propòsit: Creació de la taula de veles

Resum: El cas d'ús permet crear la taula de veles a la base de dades sempre i quan no existeixi.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja crear la taula de veles.	
	2. El sistema crea la taula i comunica a l'usuari que el procés de creació ha finalitzat correctament.
3. L'usuari està satisfet perquè a creat la taula veles a la base de dades.	

Cursos alternatius

Línia 2: Si la taula existeix no es realitza cap acció i el sistema ho comunica a l'usuari.

Cursos fallits

Línia 2: Si es produeix un error intern durant la creació de la taula el sistema l'escriu al diari del sistema i ho comunica a l'usuari.

5.2.19 Especificació del cas d'ús "Esborrar taula de veles"

Cas d'ús: Eliminació de la taula de veles de la base de dades.

Actors: Usuari (iniciador), Sistema.

Propòsit: Eliminació de la taula de veles

Resum: El cas d'ús permet l'esborrat de la taula de veles de la base de dades sempre i quan existeixi.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja esborrar la taula de veles.	
	2. El sistema informa a l'usuari de que es perdran totes les dades i li pregunta si està segur de que desitja esborrar-la.
3. L'usuari decideix si desitja esborrar la taula veles: c. **if** esborrar **see section** esborrar taula d. **if** no esborrar **see section** no esborrar taula	
4. L'usuari està satisfet perquè, si ho desitjava, ha esborrat la taula veles de la base de dades.	

Section esborrar taula

Accions dels actors	Resposta del sistema
1. L'usuari prem el botó *Acceptar* per esborrar la taula.	
	2. El sistema esborra la taula de veles de la base de dades.
	3. El sistema comunica a l'usuari que el procés d'esborrat a finalitzat correctament.

Cursos alternatius

Línia 2: Si la taula no existeix s'atura el procés d'esborrat i el sistema ho comunica a l'usuari.

Cursos fallits

Línia 2: Si es produeix un error intern durant l'esborrat de la taula el sistema l'escriu al diari del sistema i ho comunica a l'usuari.

Section no esborrar taula

Accions dels actors	Resposta del sistema
1. L'usuari prem el botó *Cancel·lar* per no esborrar la taula.	
	2. El sistema no realitza cap acció.

5.2.20 Especificació del cas d'ús "Manual d'usuari"

Cas d'ús: Consulta del manual d'usuari.

Actors: Usuari (iniciador), Sistema.

Propòsit: Consulta del manual d'usuari per conèixer el funcionament de l'eina.

Resum: Es mostra el manual d'usuari de l'eina per a possibles consultes sobre el seu funcionament, sobre el significat d'elements de les vistes o el funcionament de certs processos imprescindibles.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja consultar el manual d'usuari.	
	2. El sistema mostra la vista amb el manual d'usuari.
3. L'usuari està satisfet perquè ja pot consultar la informació que li sigui d'interès.	

5.2.21 Especificació del cas d'ús "Sobre Regaton"

Cas d'ús: Consulta de la informació sobre *Regaton*.

Actors: Usuari (iniciador), Sistema.

Propòsit: Consultar la informació sobre l'eina.

Resum: Mostra la informació que identifica l'eina, versió, autor i director, etc.

Tipus: Primari essencial.

Curs típic d'esdeveniments

Accions dels actors	Resposta del sistema
1. El cas d'ús comença quan l'usuari li indica al sistema que desitja consultar la informació sobre l'eina.	
	2. El sistema mostra la vista amb la informació sobre l'eina.
3. L'usuari està satisfet perquè ja pot consultar la informació.	

5.3 Diagrama de classes

5.3.1 Diagrama de classes del domini

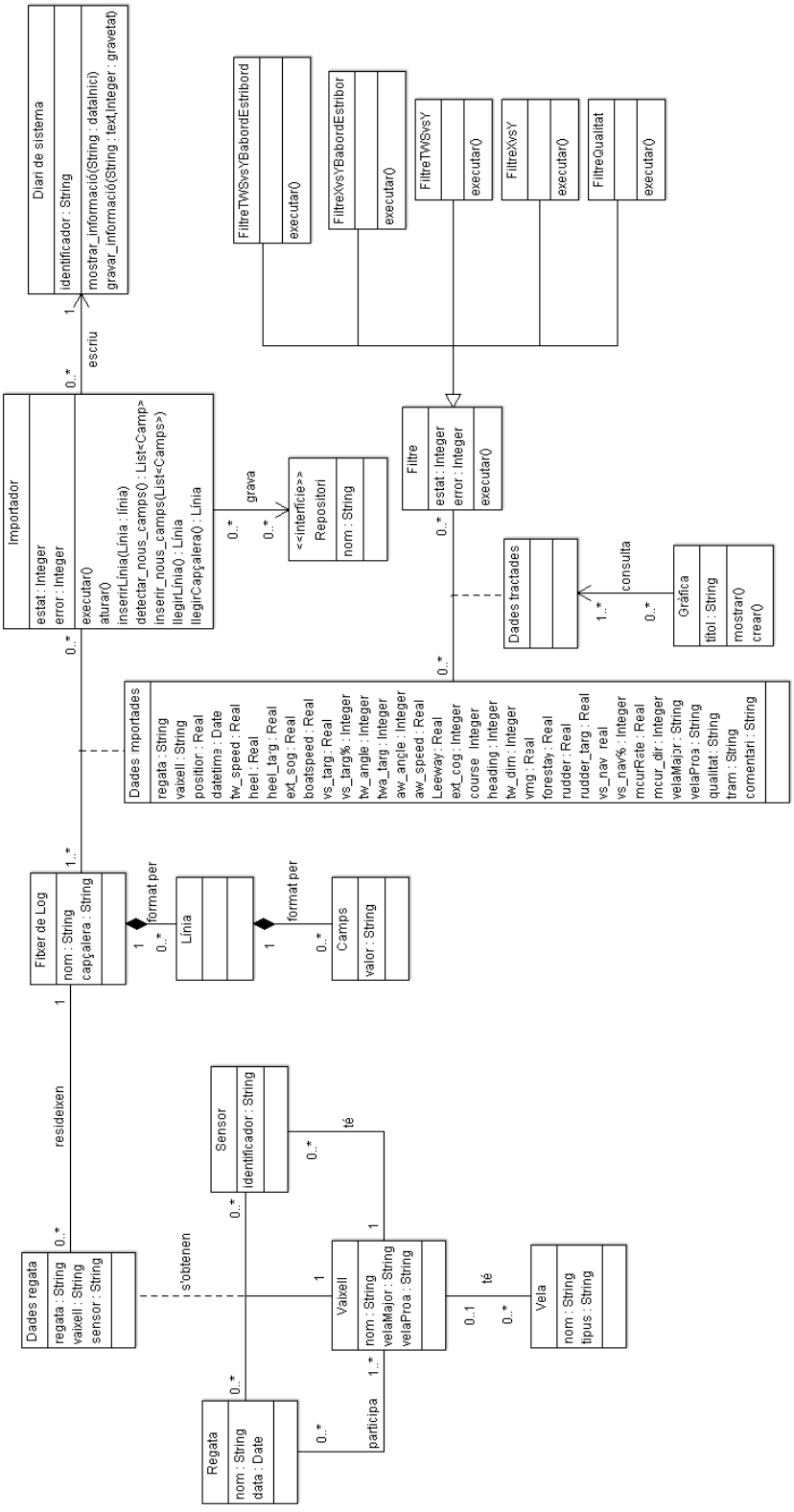


Figura 35: Diagrama de classes del domini

El model conceptual del sistema mostra que a partir dels sensors d'un vaixell que navega en regates, siguin d'entrenament o de competició, s'obté la informació per cada regata.

La informació de cada regata s'emmagatzema en fitxers de *Log* separant les variables d'estudi, que recull cada sensor, en camps de manera estructurada.

Es gràcies al importador que s'insereixen les dades de cada regata dins del sistema.

Una vegada les dades són persistents, és possible generar gràfiques que ajudaran a la tasca d'anàlisi de l'equip de popa.

Segons el tipus de gràfica a mostrar s'usa un *filtre* o un altre. Cada *filtre* tracta el conjunt de dades d'entrada realitzant els càlculs oportuns per obtenir la informació que es representarà dins la gràfica.

6 Disseny

En el capítol anterior, especificació del sistema, s'ha definit el que ha de fer el sistema per complir amb els requeriments demanats. Però, encarà queda pendent respondre a com el sistema ho ha de resoldre, és a dir, cal definir el seu comportament.

El disseny arquitectònic consisteix en dissenyar l'arquitectura del programari del sistema. L'arquitectura del programari estableix l'estructura de tots els components del sistema, així com les seves propietats i les relacions que hi ha entre ells.

Per exemple, un component pot identificar-se com una classe, una taula de la base de dades, o qualsevol element pertanyent al sistema. Podem descriure una relació com les crides a procediments de les classes o les relacions entre taules de la base de dades.

El capítol explica el patrons arquitectònics elegits per desenvolupar el sistema, el disseny de les diferents parts de l'arquitectura, els patrons de disseny utilitzats i el comportament dels processos més importants de l'eina.

6.1 Patrons arquitectònics

En primer lloc determinarem la distribució dels components del sistema, és a dir el patró arquitectònic a utilitzar.

Existeixen diversos patrons arquitectònics, com el patró Pissarra, el patró orientat a serveis, el patró Presentació – Abstracció – Control, el patró Model – Vista – Controlador i altres. Cada patró té els seus avantatges i inconvenients, però la gran majoria d'aplicacions de la categoria del sistema que estem desenvolupant utilitzen el patró arquitectònic més comú que és el Model – Vista – Controlador o MVC.

6.1.1 Patró Model – Vista – Controlador

És un patró de l'arquitectura del programari que separa les dades d'una aplicació, la interfície d'usuari i la lògica de control en tres components diferents.

Definim el *Model* com la representació específica de la informació amb la que treballa el sistema.

La *Vista* és la que presenta el *Model* en un format adequat per interactuar-hi, acostuma a correspondre a la interfície d'usuari.

La resposta als esdeveniments produïts normalment per l'usuari recau sobre el *Controlador* i invoca peticions al *Model*, i potser també a la *Vista*.

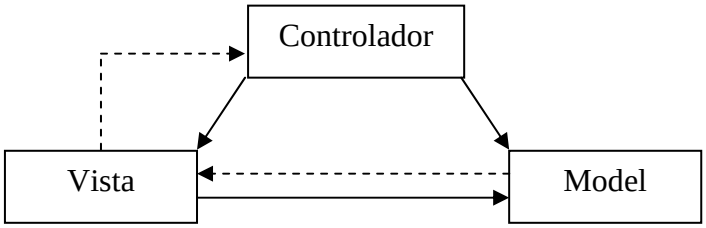


Figura 36: Esquema bàsic del Model – Vista - Controlador

El diagrama anterior mostra la relació entre el *Model*, la *Vista* i el *Controlador*. Les línies sòlides indiquen una associació directa, les puntejades una indirecta.

L'ús d'aquests patró permet separar completament la part visible i d'interacció amb l'usuari (*Vista*) de la part interna (*Controlador*, *Model*). S'aconsegueix un sistema ben organitzat que podrà ser reusable, escalable i mantenible; Totes qualitats d'interès per ajudar a una correcta comprensió i extensió del sistema.

6.1.1.1 Model

El *Model* del sistema conté totes les classes que representen el tractament d'una regata realitzada per un vaixell. Les classes estan formades pels atributs que guarden la seva informació i els mètodes que defineixen com interactuen.

El diagrama de classes del capítol anterior, representat a la pàgina *82*, descriu el nostre *Model* del sistema.

6.1.1.2 Controlador

Els controladors del sistema contenen la lògica encarregada de la gestió d'esdeveniments i comunicació d'errors a l'usuari a través de les vistes.

L'aplicació posseeix un únic controlador encarregat de tota lògica. Només se'n defineix un degut a que el nombre de funcionalitats de l'eina no són molt elevades i no representen una sobrecàrrega pel seu funcionament.

S'obvien els tipus de dades dels paràmetres per aquelles operacions que es farien massa grans fent així la classe llegible.

Figura 37: Classe Controlador

6.1.1.3 Vista

Les vistes s'encarreguen de mostrar les interfícies disponibles a l'usuari per a que interaccioni amb el *Model*.

El següent és un diagrama que mostra les classes que formen part de la interfície d'usuari del sistema. També es pot observar que és la barra de menús la que executa cadascuna de les vistes del sistema.

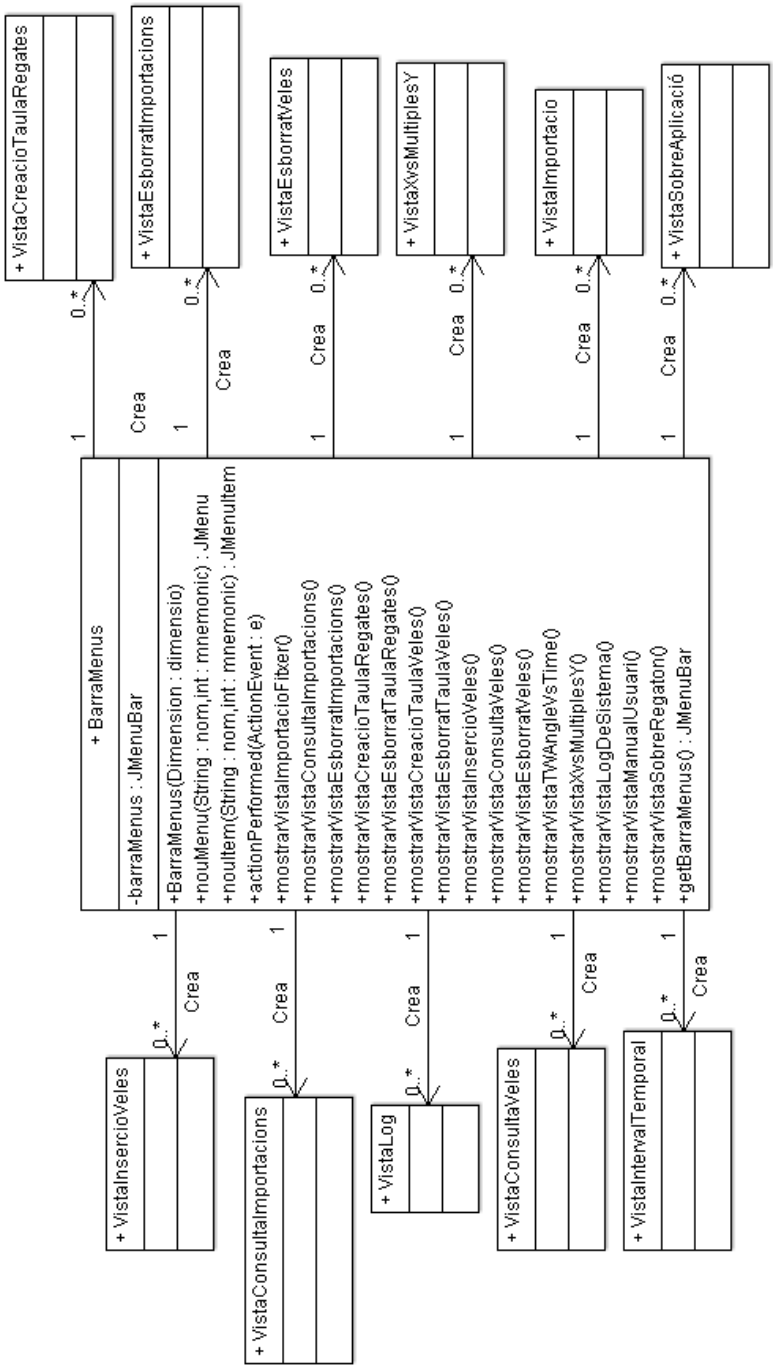


Figura 38: Diagrama de les classes que formen part de la interfície d'usuari

Els següent diagrama mostra la generalització de les vistes de la interfície d'usuari.

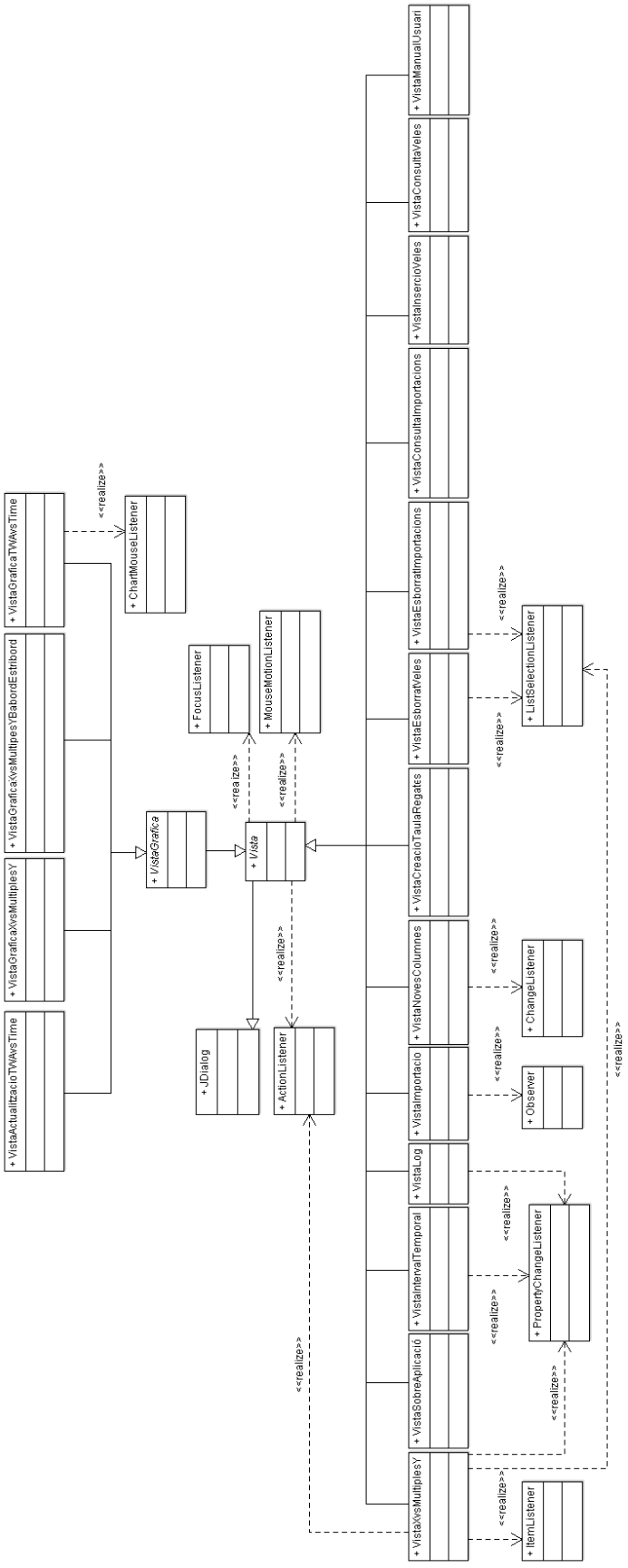


Figura 39: Diagrama amb la generalització de les vistes del sistema

El diagrama que es mostra a continuació correspon a la navegabilitat entre les vistes.

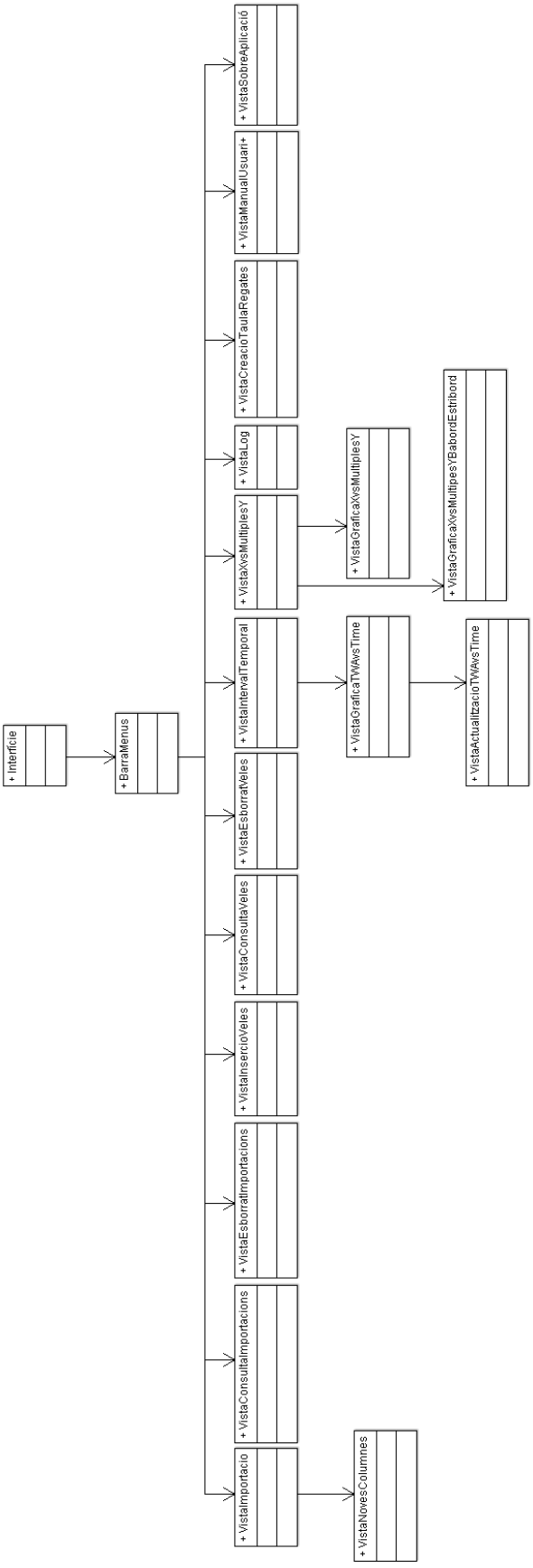


Figura 40: Diagrama de navegabilitat entre les vistes del sistema

Per últim, els següents diagrames mostren el contingut de les vistes més importants:

Figura 41: Diagrama amb les vistes més importants del sistema

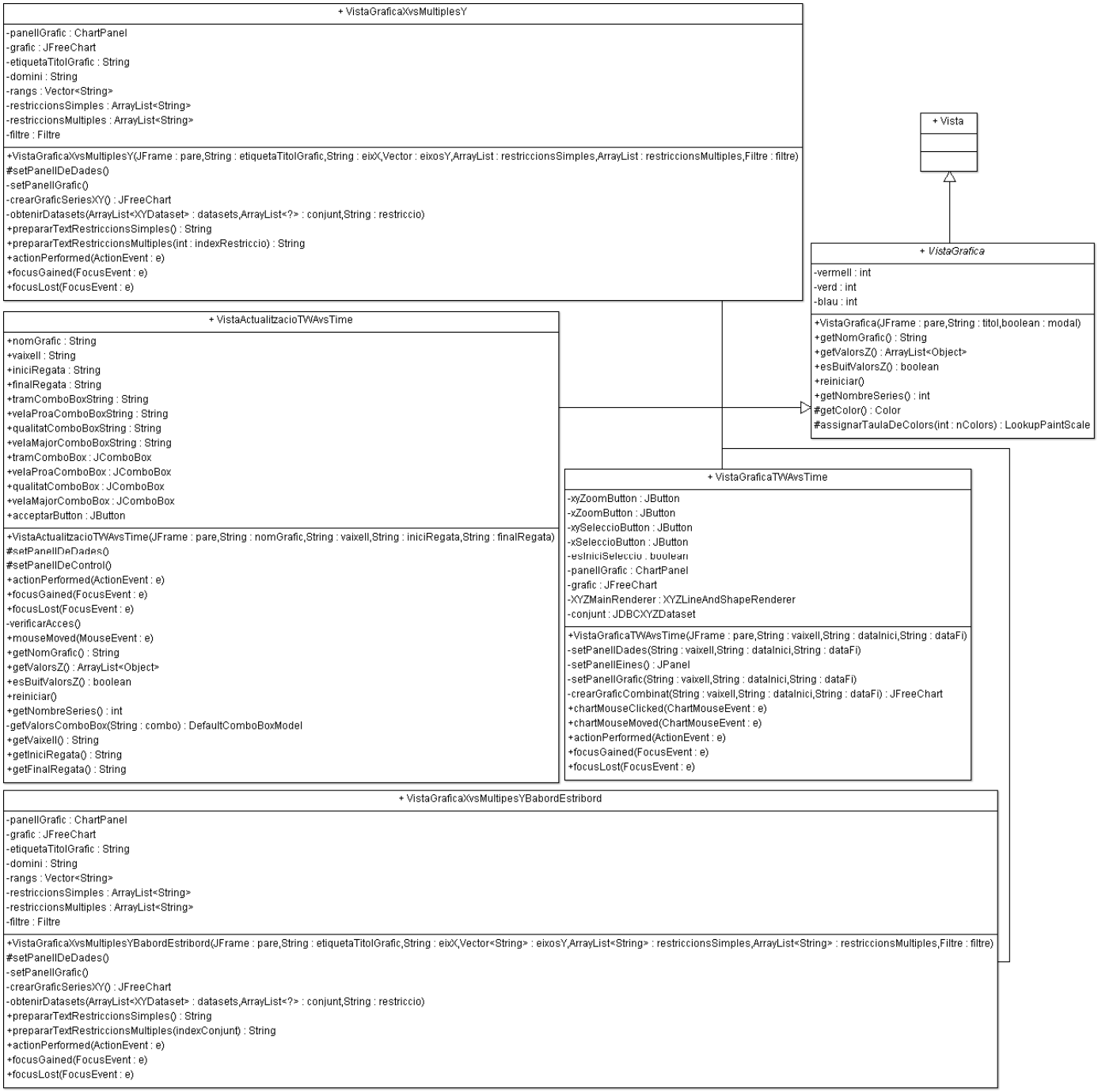


Figura 42: Diagrama amb les vistes gràfiques més importants del sistema

6.1.2 Patró Arquitectura per Capes

L'arquitectura per capes és un dels patrons arquitectònics més estesos. El patró té com a objectiu principal la separació de la lògica dels negocis de la lògica del disseny. L'exemple més clar és la separació de la capa de presentació a l'usuari de la capa de dades.

L'avantatge principal d'aquest estil és que el desenvolupament es pot realitzar en diverses capes, i en el cas que sobrevingui un canvi, només cal modificar la capa afectada sense haver de revisar entre codi barrejat.

El disseny usat en el sistema és l'arquitectura en tres capes, que és el més utilitzat en l'actualitat.

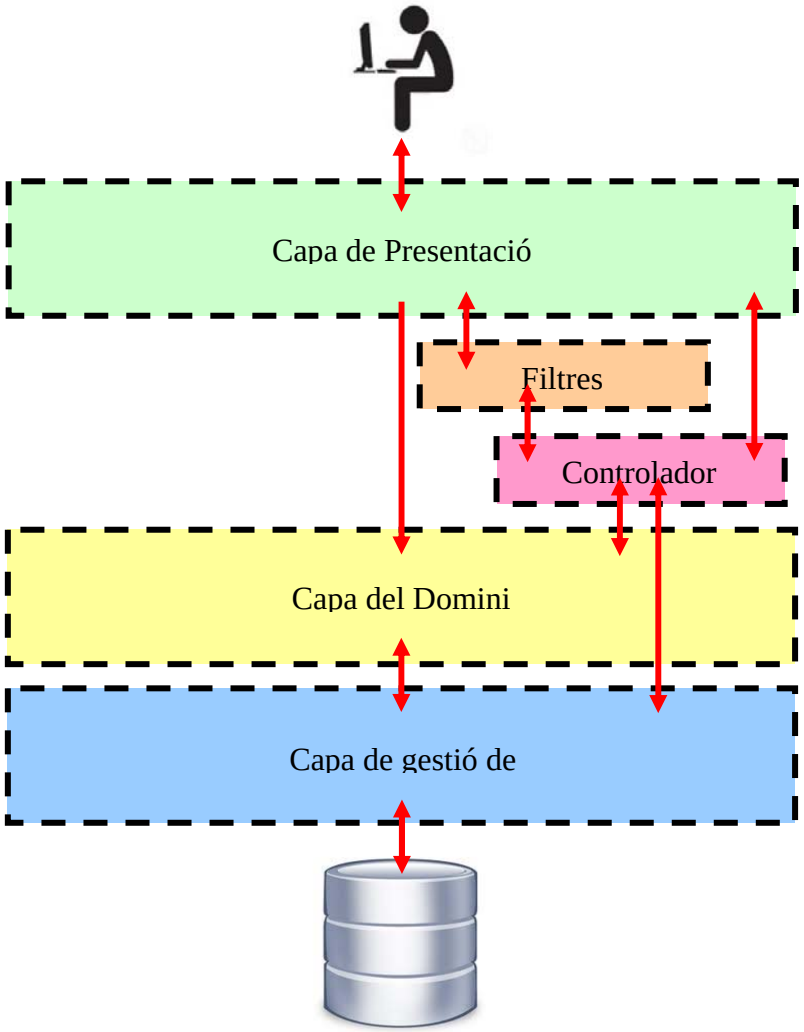


Figura 43: Arquitectura per capes

La figura anterior, mostra les capes del sistema i els camins d'accés de cadascuna.

La primera capa és la de presentació. És la capa amb la que interacciona l'usuari i és coneguda com interfície d'usuari. Aquesta capa només es comunica amb la capa de domini.

La capa de domini és on resideix la lògica que s'executa. És la que rep les peticions de l'usuari i envia una resposta per a cadascuna d'elles. Aquesta és la capa que es comunica amb la de gestió de dades.

L'última capa és la de gestió de dades. És la capa on resideixen les dades i és l'encarregada de l'accés a elles. Està formada per un sistema de gestió de base de dades que rep sol·licituds d'emmagatzematge i recuperació d'informació des de la capa de domini.

Entre la capa de presentació i la de domini existeixen dos elements importants, els *filtres* de tractament i anàlisi de dades i el controlador del sistema.

El primers són objectes cridats per la capa de presentació. Són entitats independents de la capa de domini i la seva execució accedeix directament a la capa de gestió de dades aconseguint un accés més eficient.

Per altra banda, l'ús de l'arquitectura en tres capes combinada amb el Model – Vista - Controlador provoca que el controlador prengui rellevància convertint-lo en l'entitat encarregada de tractar els esdeveniments que es produeixen en el sistema i delegar el tractament de les peticions als processos adequats, comunicant finalment les respostes obtingudes als peticionaris.

6.2 Patrons de disseny

Els patrons de disseny ajuden a resoldre problemes comuns que apareixen als projectes de desenvolupament de programari i ofereixen solucions genèriques a les que es poden adaptar els elements del sistemes.

L'ús de patrons de disseny també aporta que el inici del desenvolupament sigui a partir d'estructures ben dissenyades evitant començar des de zero.

6.2.1 Patró Expert

Aquest patró persegueix que s'assigni una responsabilitat a la classe que té la informació necessària per realitzar-la.

L'ús del patró atorga alta cohesió[1] i manté l'encapsulament[2] (aconseguint baix acoblament[3]).

Les següents classes són exemples del patró aplicats al sistema:

[1] És una mesura del grau de relació i concentració de les diverses responsabilitats d'una classe.
[2] Ocultació de l'estat, és a dir, de les dades d'un objecte de manera que només es pot canviar mitjançant les operacions definides per aquell objecte.
[3] És una mesura del grau de connexió, coneixement i dependència d'una classe respecte d'altres classes.

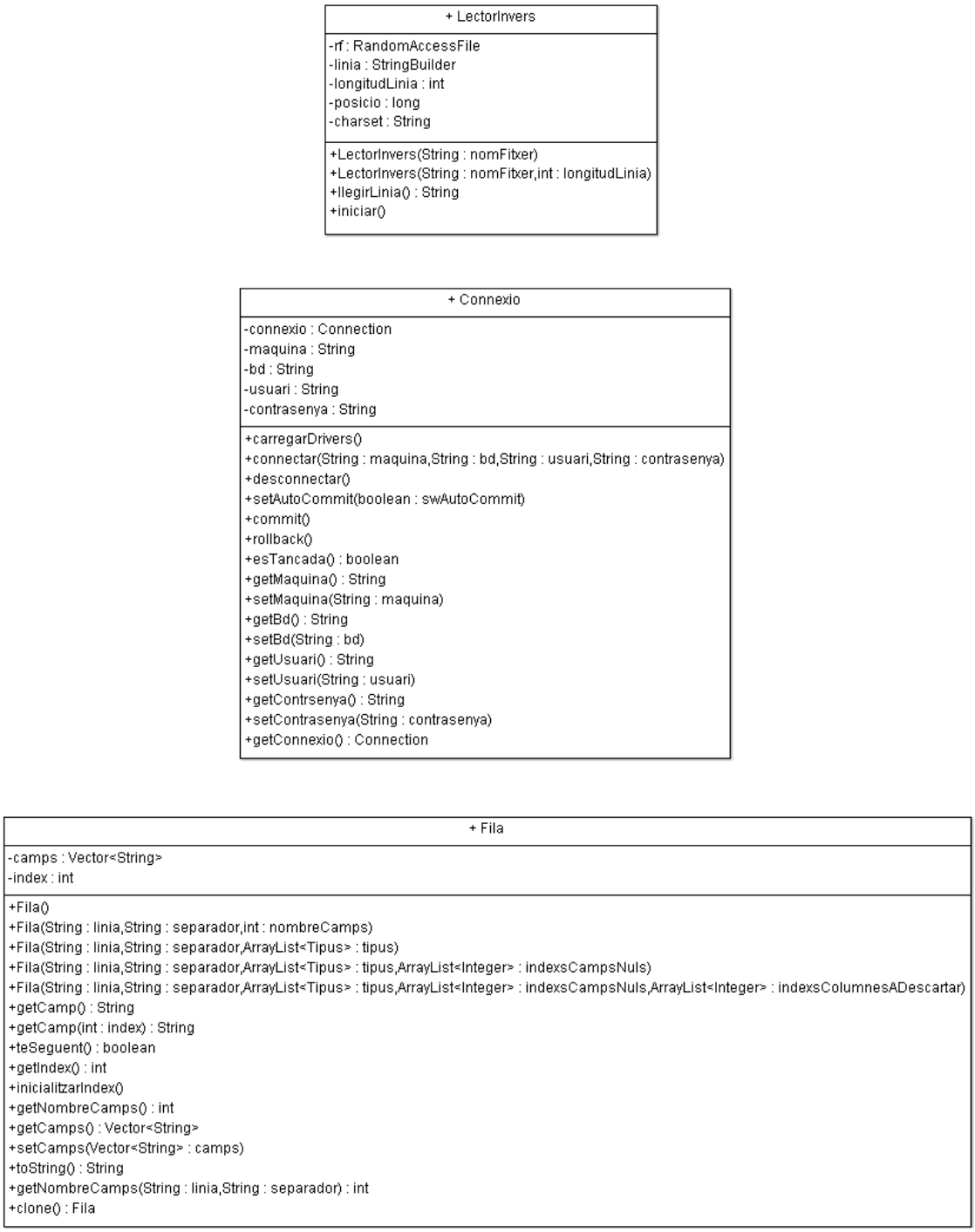


Figura 44: Exemples d'aplicació del Patró Expert

La classe *LectorInvers* només té mètodes relacionats o orientats a realitzar la lectura del fitxer amb les dades del diari del sistema des del final del fitxer al seu inici.

Per altra banda, la classe *Connexio* té tots els mètodes i atributs necessaris per crear i gestionar una connexió amb el *SGBD* del sistema, però cap altre atribut o mètode que no estigui relacionat amb les seves responsabilitats.

Per últim la classe *Fila* conté només els atributs i mètodes específics per a la creació dels seus objectes. Els objectes de la classe *Fila* contenen els camps en els que es divideixen els registres del fitxer de *Log*.

6.2.2 Patró Controlador

Amb aquest patró s'afegeix una nova classe que serà l'encarregada de gestionar els esdeveniments del sistema i delegar les responsabilitats a altres controladors més específics si existeixen.

Aquest patró atorga l'augment de la reusabilitat del codi gràcies a la separació de la capa de presentació de la lògica del sistema. Així el model, o certes parts, es poden tornar a utilitzar dins el mateix sistema o en projectes futurs.

El patró l'hem aplicat implícitament en el model MVC descrit anteriorment en aquest capítol. El sistema posseeix un únic controlador donada la seva dimensió i tractant-se d'una aplicació amb funcionalitats específiques.

6.2.3 Patró Observador

Patró de disseny que s'aplica quan un objecte necessita avisar a altres que ha canviat d'estat, sense necessitat de saber quins objectes ni com. Els objectes dependents, o observadors, s'afegeixen a una llista d'objectes que seran notificats dels canvis produïts en els objectes observats.

La figura següent mostra l'aspecte del patró:

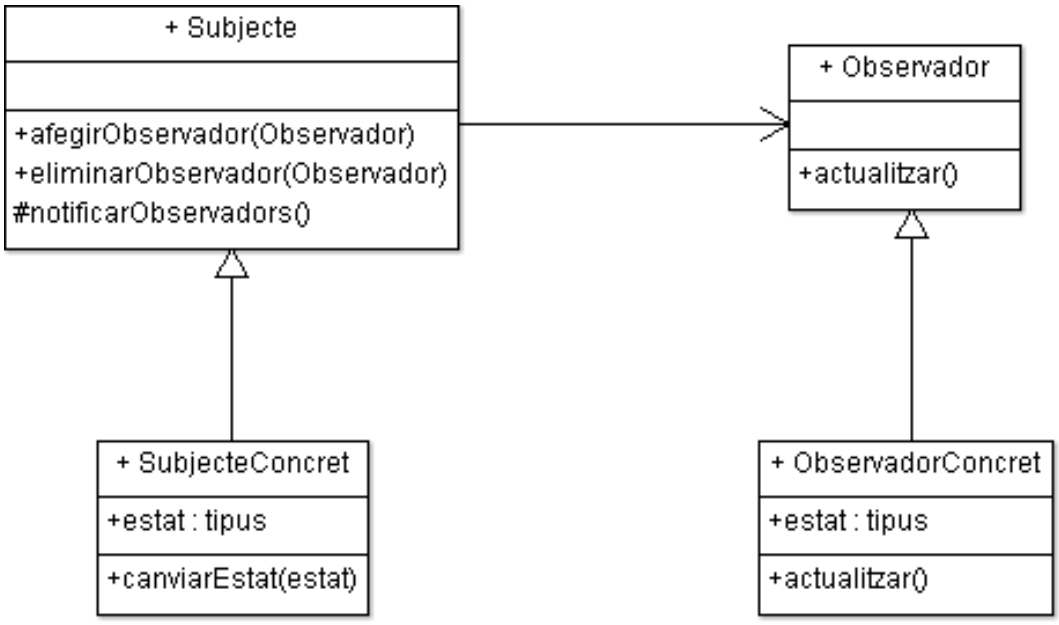


Figura 45: Diagrama del patró Observador

Existeixen dues classes del sistema que utilitzen aquest patró, la classe *Importador* i la classe *FiltreQualitat.*

Figura 46: Diagrama amb les classes del sistema que usen el patró Observador

La classe *Importador* està usant aquest patró per informar del seu estat a la vista d'importació d'una regata. Així, la vista actualitza la barra de progrés de la importació.

La classe *FiltreQualitat* fa ús del patró pel mateix motiu que l'anterior. Informa a la vista d'importació d'una regata de la quantitat de dades tractades pel *filtre* fins l'instant actual. Així, la vista actualitza la barra de progrés del *filtre de qualitat* sobre les dades.

6.3 Realitzacions dels casos d'ús més importants

La realització d'un cas d'ús descriu com es realitza al model de disseny aplicant els patrons escollits i tenint en compte l'arquitectura seleccionada.

És un procés que ajuda a conèixer més detalls de com funcionaran els casos d'ús del sistema, quina seqüència seguiran i quines classes i mètodes necessitaran. Per la realització d'un cas d'ús es descriuen els diagrames de seqüència i els contractes de les operacions més importants.

Un diagrama de seqüència mostra en detall quins actors i classes interactuen en un cas d'ús i la seqüència de les crides que es realitzen a les classes involucrades.

Els contractes de les operacions especifiquen quines condicions s'han de complir abans (precondicions) i després (postcondicions) de la seva l'execució.

6.3.1 Realització del cas d'ús "Importar regata"

Els següents diagrames de seqüència mostren com es realitza el cas d'ús.

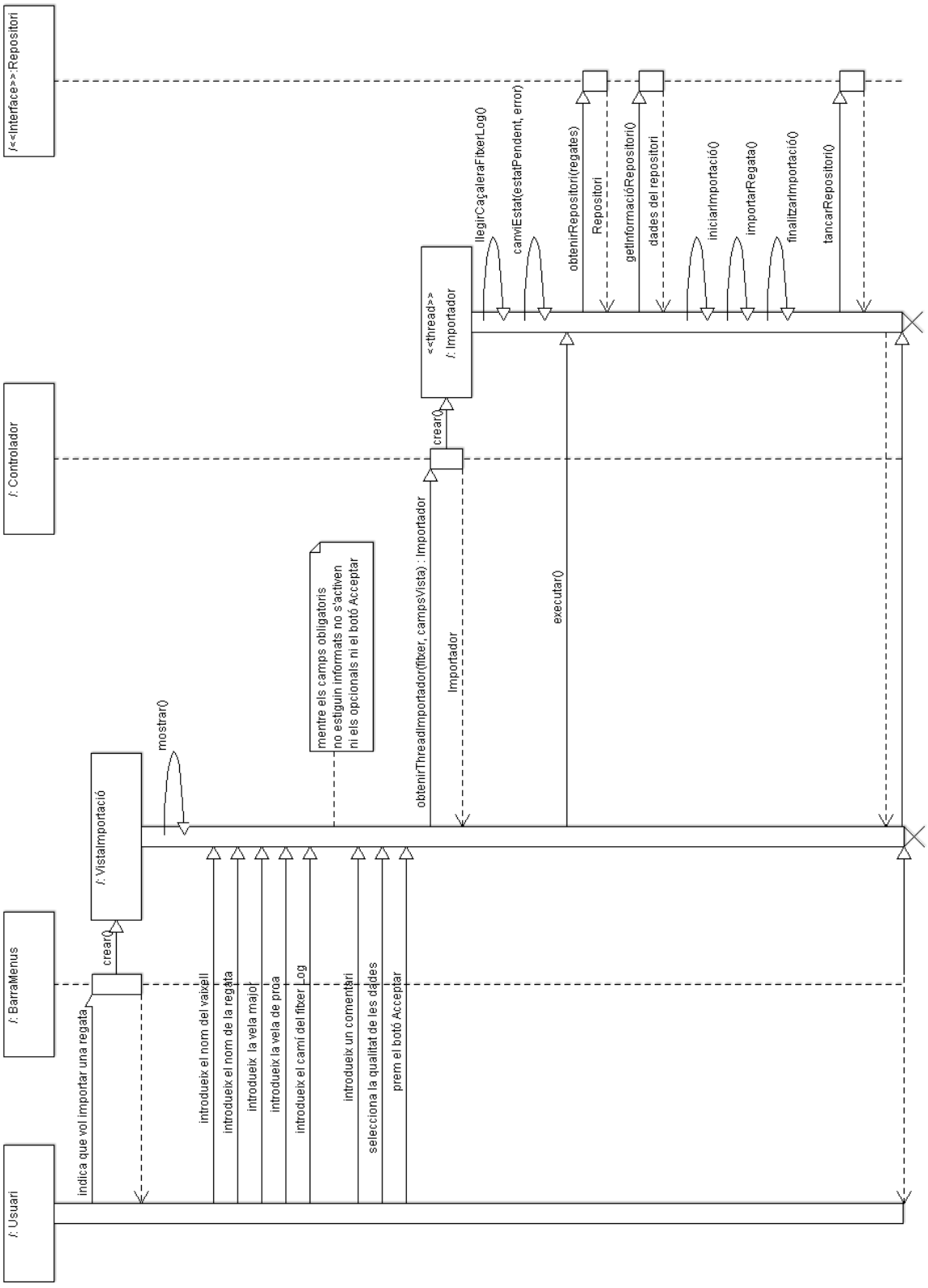


Figura 47: Realització del cas d'us "Importar regata"

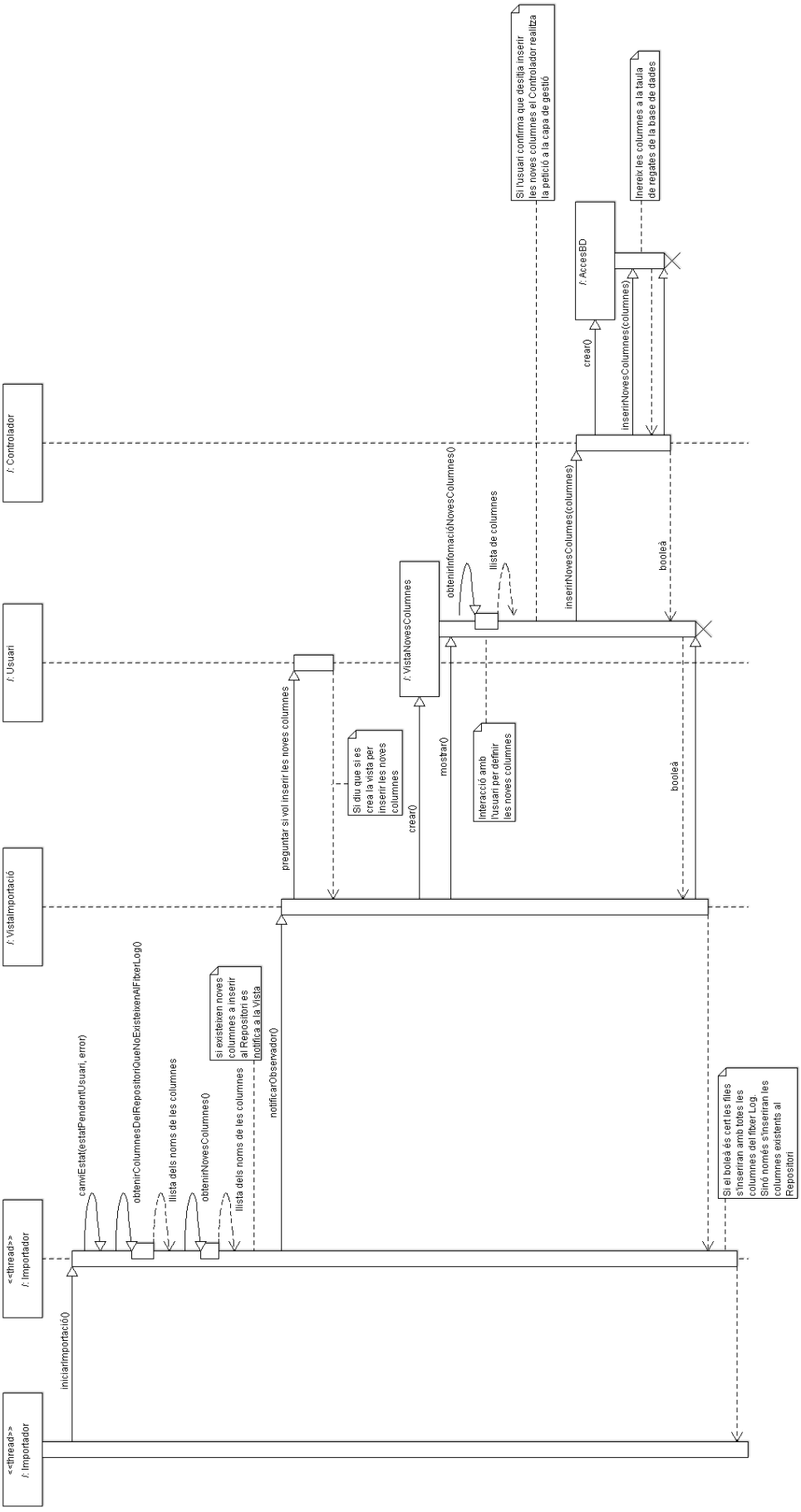


Figura 48: Diagrama de seqüència de l'operació "iniciarImportació"

Name: iniciarImportació()
Responsibilities:
Iniciar el procés d'importació de les dades de la regata.
Exceptions:
Si el fitxer és buit finalitzar la importació.
Si l'usuari cancel·la el procés finalitzar la importació amb estat *Cancel·lat*.
Si es produeix un error intern finalitzar importació amb estat *Avortament*.
Preconditions:
El primer registre del fitxer *Log* és la capçalera.
Postconditions:

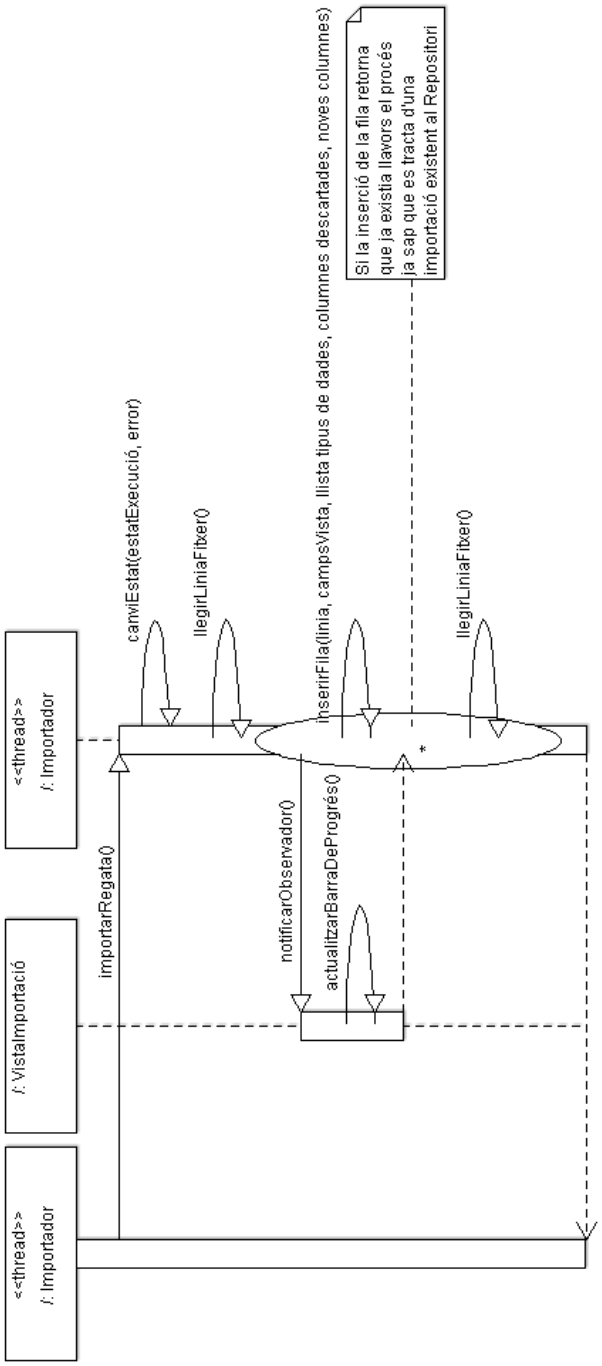


Figura 49: Diagrama de seqüència de l'operació "importarRegata"

Name: importarRegata ()

Responsibilities:

Procés d'importació de les dades del fitxer de *Log* a la base de dades del sistema.

Exceptions:

Si l'usuari cancel·la el procés finalitzar la importació amb estat *Cancel·lat*.

Si es produeix un error intern finalitzar importació amb estat *Avortament*.

Preconditions:

Si s'han detectat noves columnes i l'usuari ha confirmat que s'havien d'inserir el sistema ja s'ha actualitzat amb la nova informació.

Postconditions:

Tots els registres correctes del fitxer s'han inserit a la base de dades.

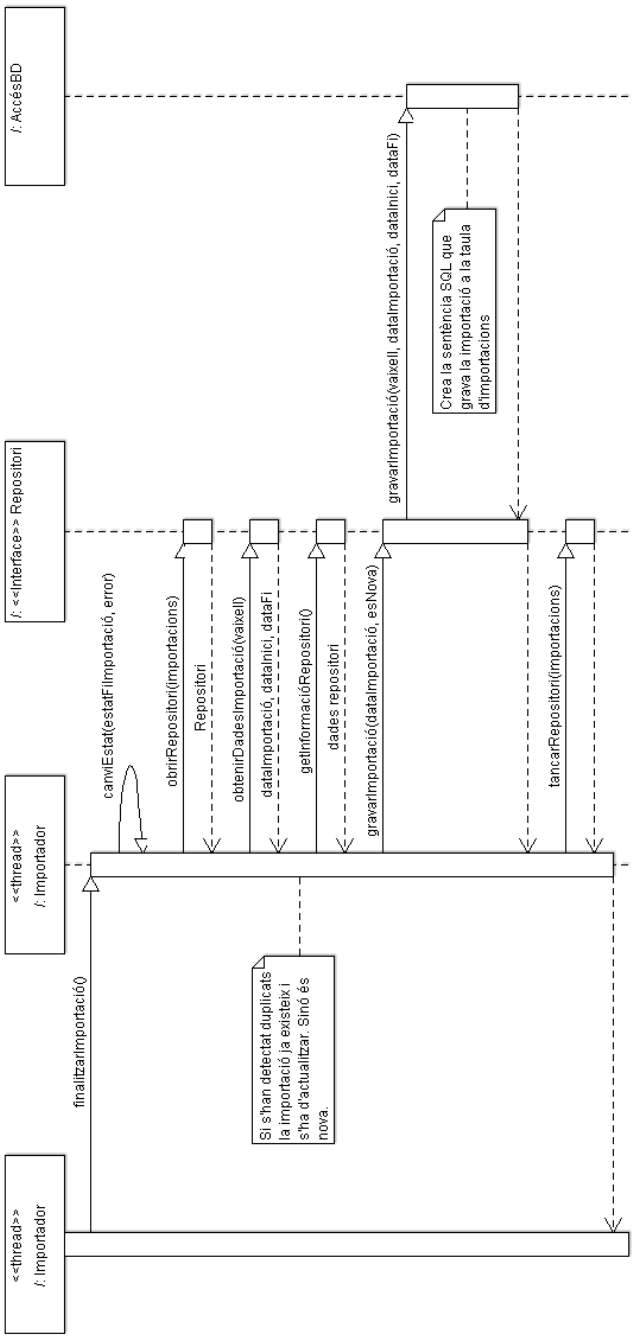


Figura 50: Diagrama de seqüència de l'operació "finalitzarImportació"

Name: finalitzarImportació()

Responsibilities:

Es grava a la taula d'importacions la informació de la regata importada. Si és nova s'insereix una nova fila, si ja existia s'actualitza amb la nova informació.

Exceptions:

Si es produeix un error intern finalitzar la importació amb estat *Avortament*.

Preconditions:

La taula de regates ja conté totes les dades importades.

Postconditions:

Tota la informació referent a la regata importada és persistent a la base de dades del sistema.

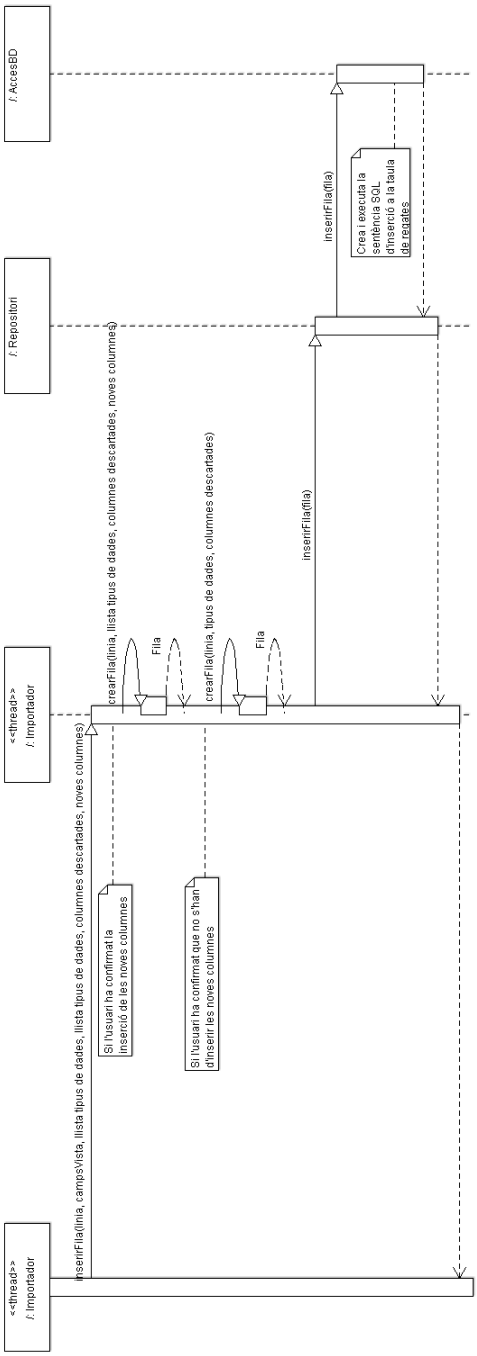


Figura 51: Diagrama de seqüència de l'operació "inserirFila"

Name: inserirFila (línia,
camps de la vista,
llista tipus de dades,
columnes descartades,
noves columnes)

Responsibilities:

Es grava a la taula de regates la nova fila llegida del fitxer de *Log*.

Exceptions:

Si es produeix un error intern finalitzar importació amb estat *Avortament*.

Preconditions:

Postconditions:

El procés d'importació passa pels següents estats duran la seva execució:

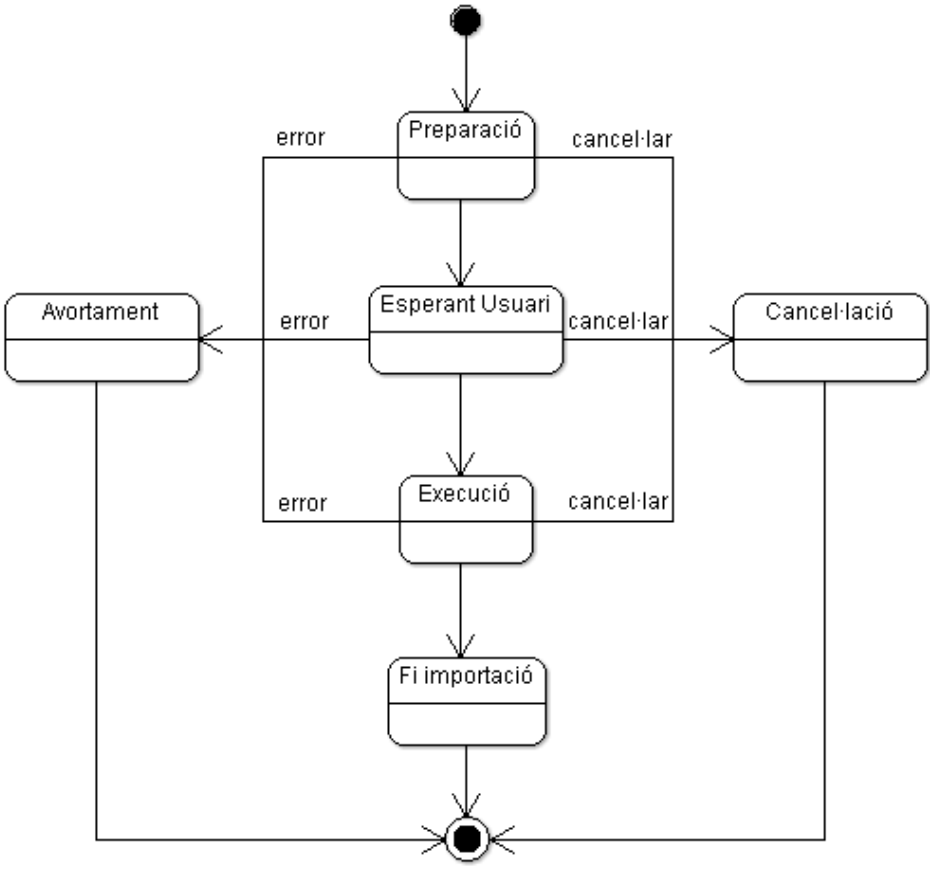


Figura 52: Diagrama d'estats del procés d'importació

A partir d'aquests estats i del patró *Observador* els observadors són informats dels canvis d'estat de l'objecte importador i poden actuar en conseqüència.

El procés s'inicia en estat de *Preparació*. Quan el sistema esbrina si han aparegut camps nous canvia l'estat a *Esperant usuari,* així la vista d'importació pregunta a l'usuari si desitja inserir les noves columnes i es realitza el cas d'ús "Inserir noves columnes". Una vegada inserides les noves columnes, o obviades, el sistema inicia el procés d'importació i l'estat avança a *Execució.* Quan s'arriba al final del fitxer i la gravació de la informació al sistema finalitza el sistema pren l'estat *Fi importació* a partir del qual el procés d'importació finalitza definitivament.

Si durant el procés es produeix algun error intern l'estat passa a *Avortament*, aturant l'execució del procés i desfent totes les accions realitzades.

Per altra banda, si en qualsevol moment l'usuari decideix cancel·lar el procés, el sistema pren l'estat *Cancel·lació* i finalitza el procés desfent totes les accions realitzades.

6.3.2 Realització del cas d'ús "Filtre de qualitat"

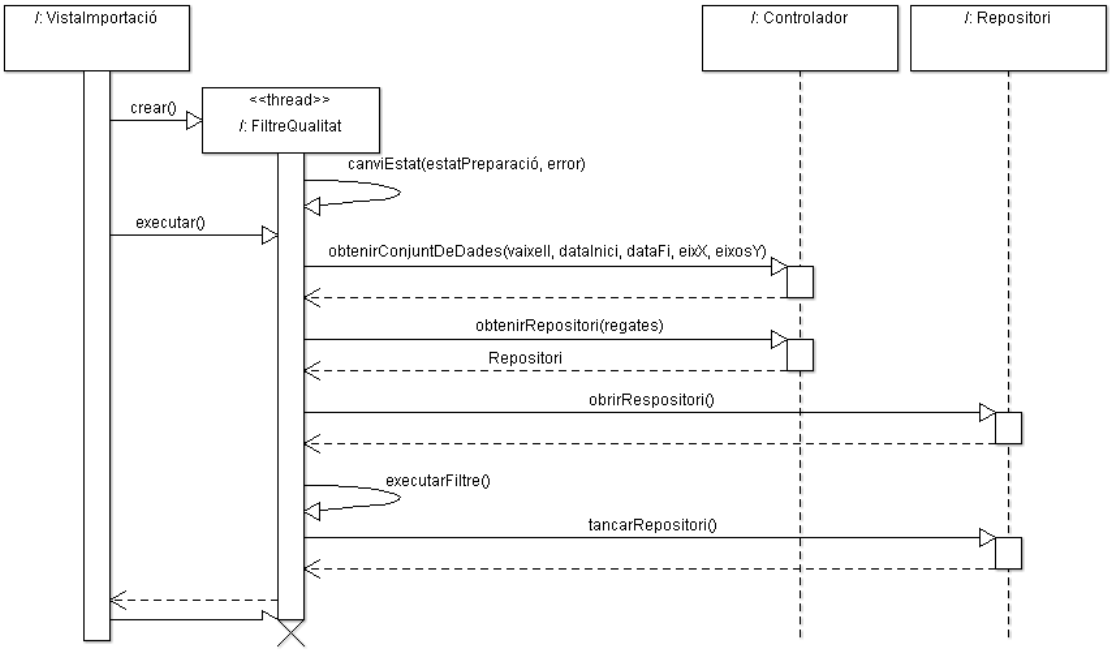


Figura 53: Realització del cas d'ús "Filtre de qualitat"

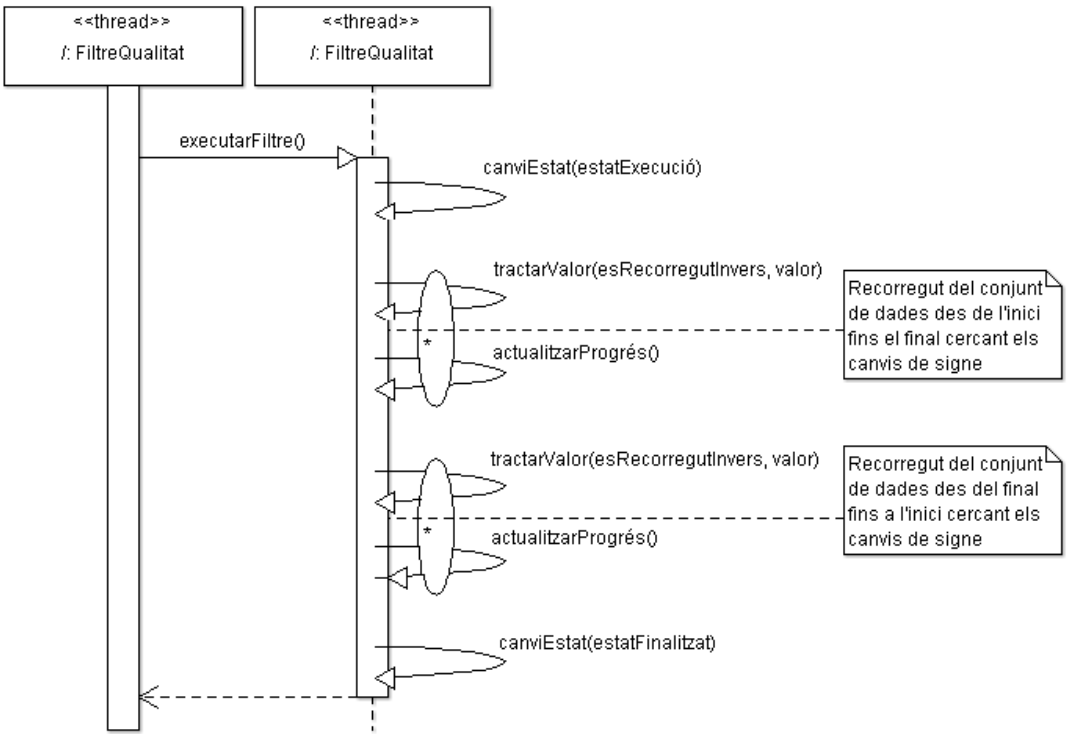


Figura 54: Diagrama de seqüència de l'operació "executarFiltre"

Name: executarFiltre()

Responsibilities:

Es realitzen dos recorreguts sobre el conjunt de dades a tractar. El primer va d'inici a fi del conjunt i tracta els instants previs dels canvis de signe. El segon recorregut comença des de final fins al inici del conjunt i tracta els instants posteriors dels canvis de signe.

Exceptions:

Si es produeix un error intern finalitzar l'execució del *filtre* i retornar l'error.

Preconditions:

S'ha d'haver importat correctament la regata del fitxer de *Log*.

Postconditions:

S'ha actualitzat tota la informació immediatament prèvia i posterior de tots els canvis de signe a qualitat *Baixa*.

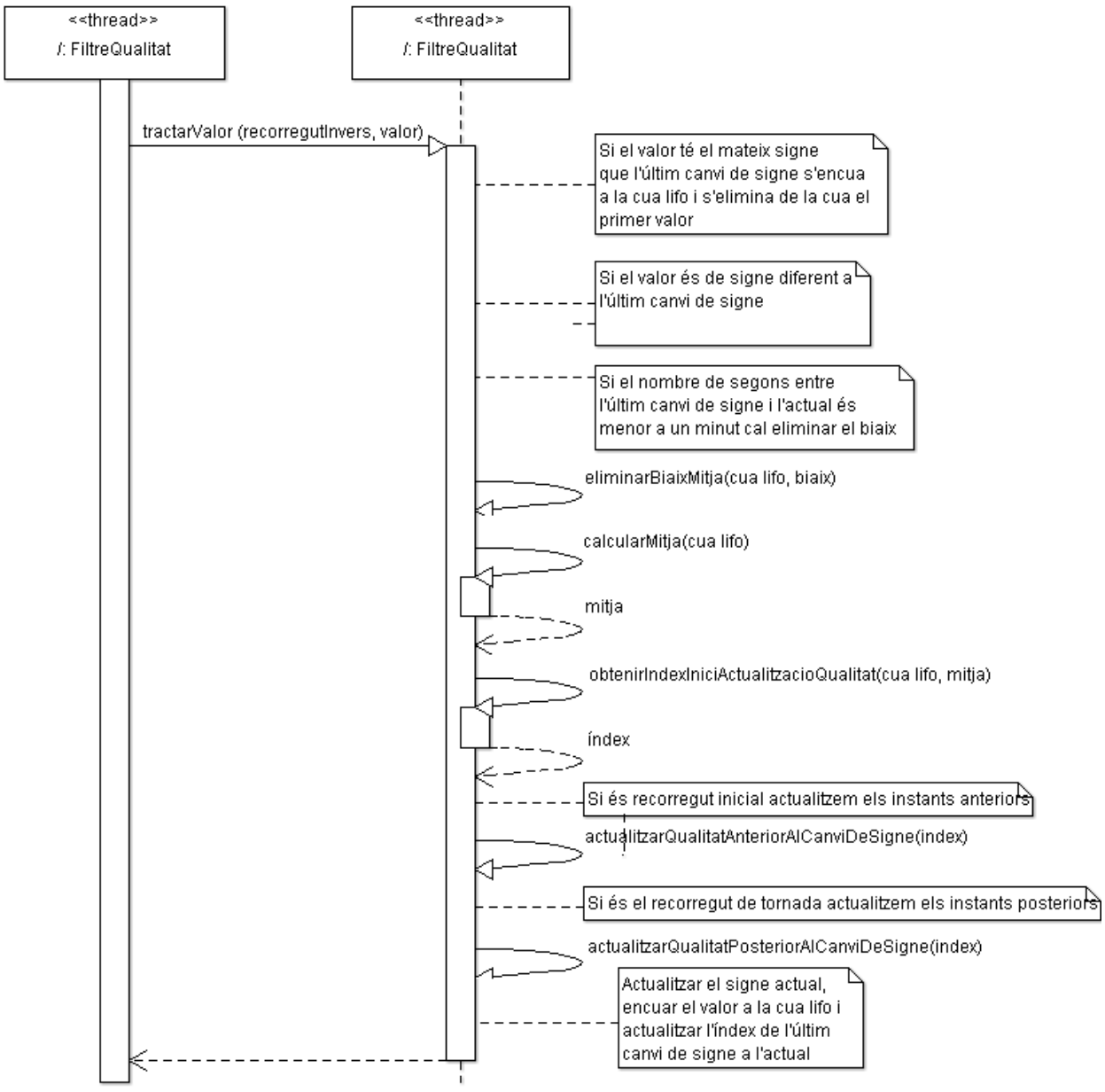


Figura 55: Diagrama de seqüència de l'operació "tractarValor"

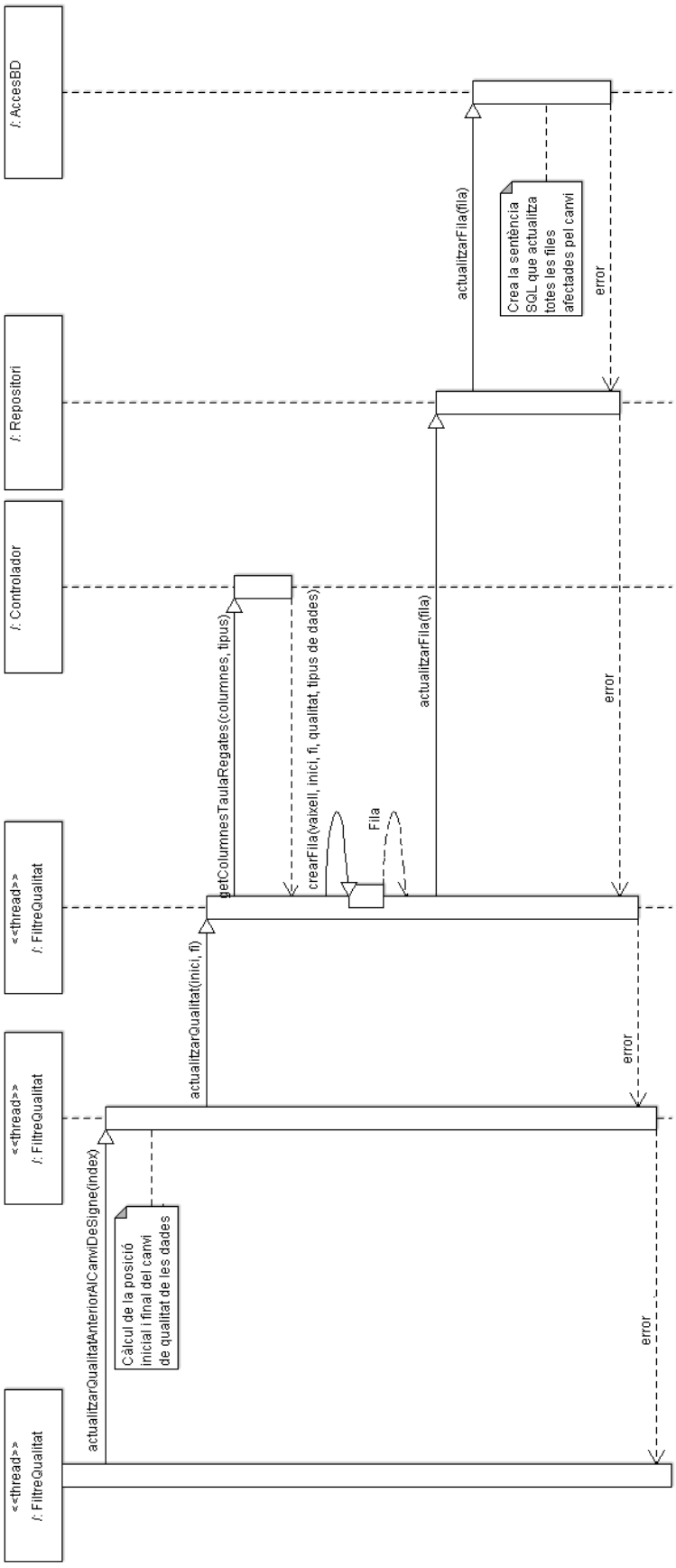


Figura 56: Diagrama de seqüència de l'operació "actualitzarQualitatAnteriorAlCanviDeSigne"

Name: tractarValor (recorregutInvers, valor)
Responsibilities:

Si el valor no representa un canvi de signe respecte l'últim canvi s'encua a la cua de valors i s'elimina el primer valor. En canvi, si es tracta d'un canvi de signe cal realitzar diverses accions:

1. Si no ha passat més d'un minut des de l'últim canvi de signe eliminar el biaix dels valors que hi ha a la cua de valors.
2. Calcular la mitja actual.
3. Obtenir l'índex d'inici d'actualització de les dades a qualitat *Baixa*.
4. Actualitzar la qualitat dels instants afectats.

Exceptions:

Si es produeix un error intern finalitzar l'execució del *filtre* i retornar l'error.

Preconditions:
Postconditions:

Name: actualitzarQualitatAnteriorAlCanviDeSigne(índex): int
Responsibilities:

Actualitza els instants anteriors al canvi de signe actual.

Exceptions:

Si es produeix un error intern finalitzar l'execució del *filtre* i retornar l'error.

Preconditions:
Postconditions:

La qualitat de les dades dels instants previs al canvi de signe s'ha canviat a *Baixa* de manera persistent.

L'operació *actualitzarQualitatPosteriorAlCanviDeSigne* funciona de manera molt similar a l'operació *actualitzarQualitatAnteriorAlCanviDeSigne.*

L'execució del *filtre de qualitat* sobre les dades passa pels següents estats:

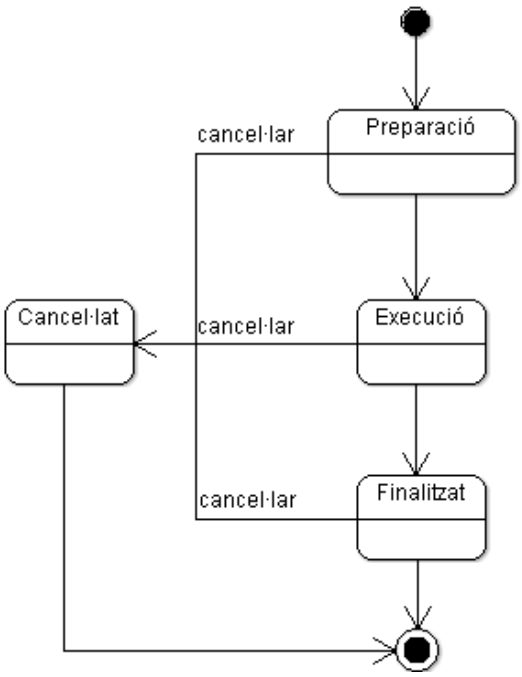


Figura 57: Diagrama d'estats de l'execució del filtre de qualitat

6.3.3 Realització del "Filtre TW_speed"

Els següents diagrames de seqüència mostren la realització del cas d'ús.

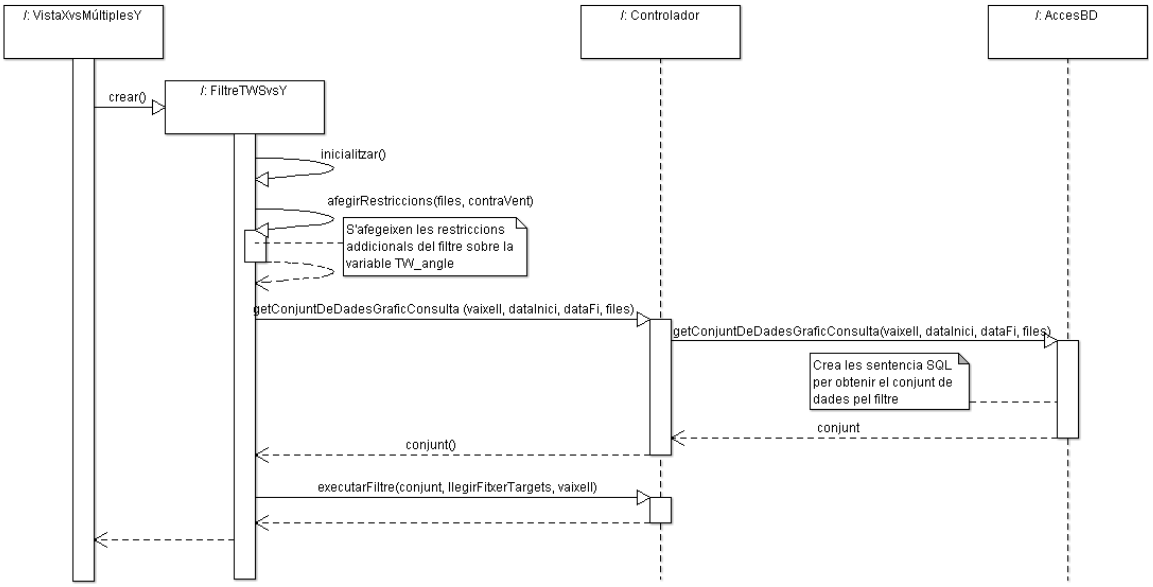


Figura 58: Realització del cas d'ús "Filtre TW_speed"

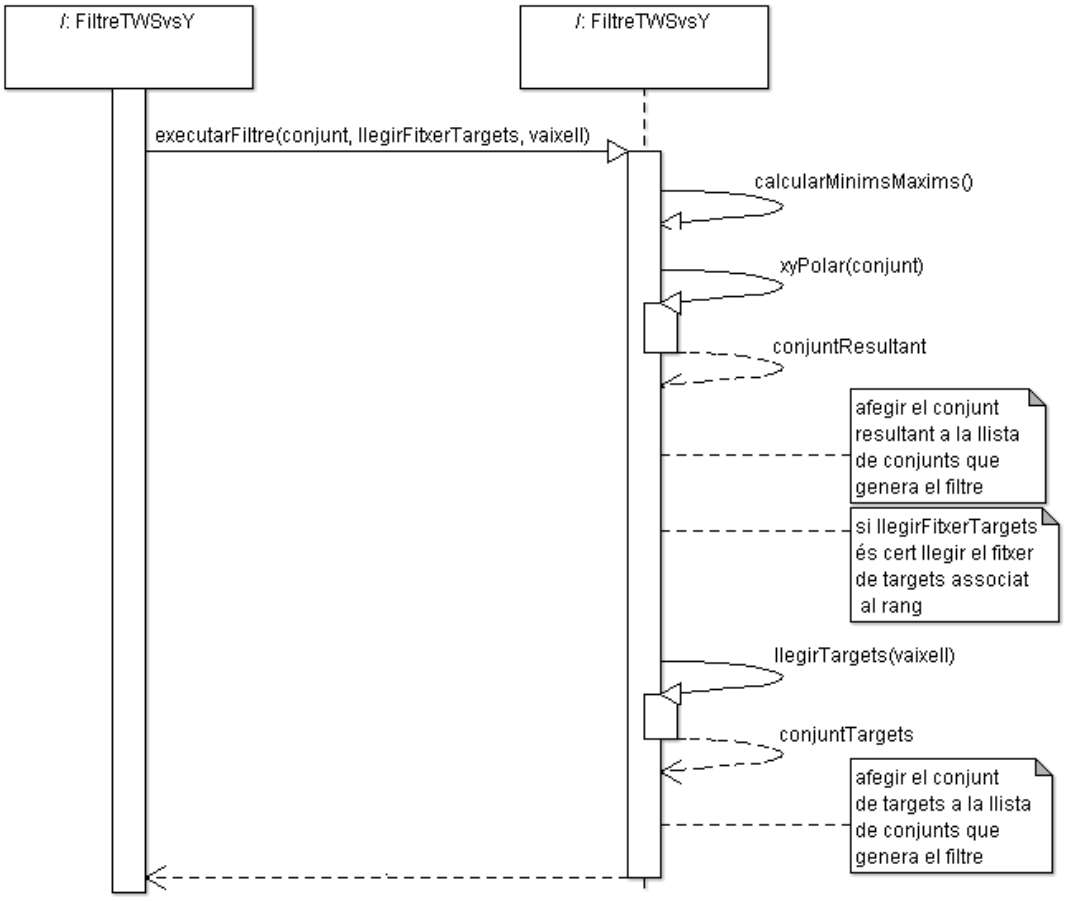


Figura 59: Diagrama de seqüència de l'operació "executarFiltre"

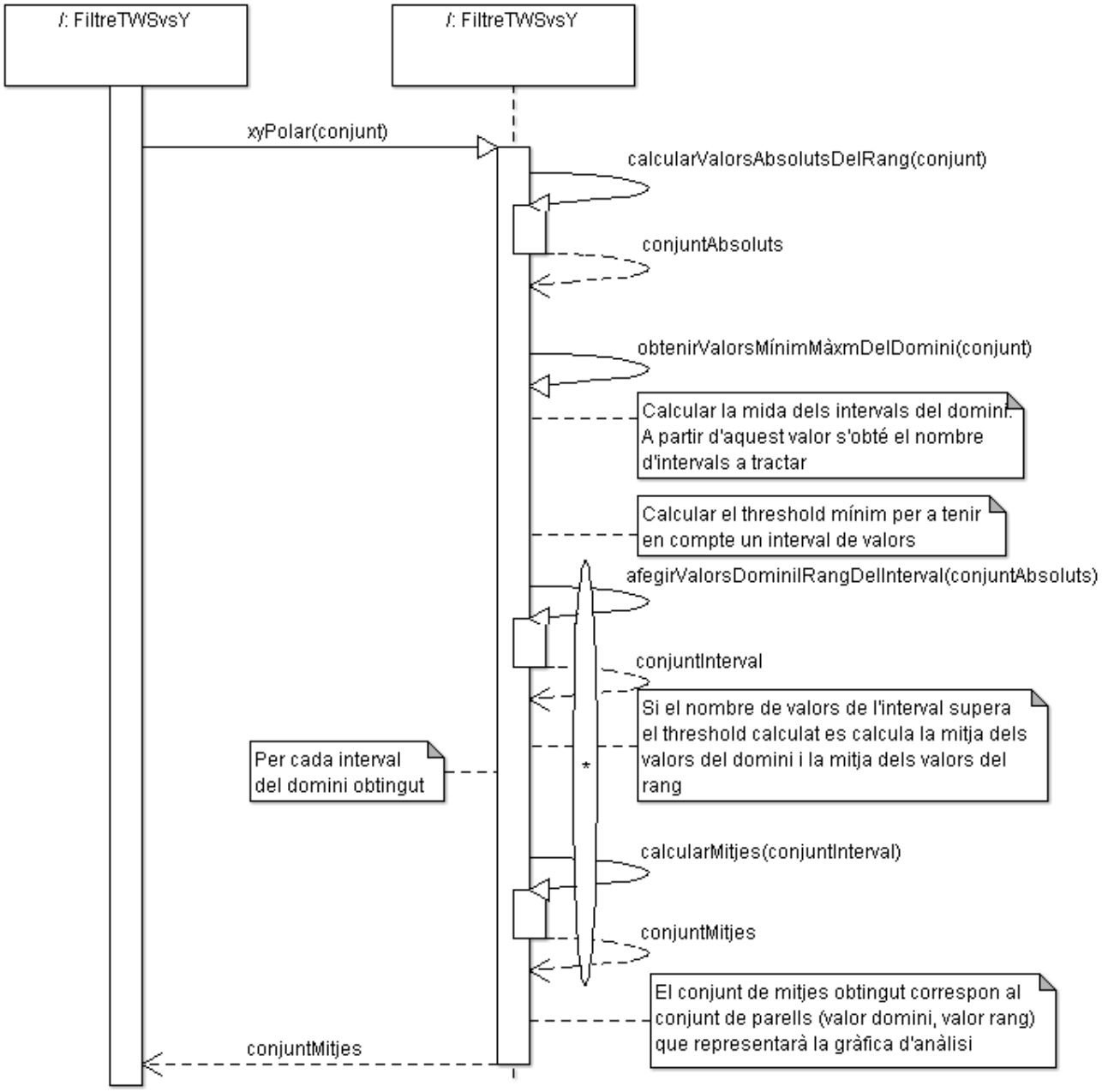


Figura 60: Diagrama de seqüència de l'operació "xyPolar"

Name: executarFiltre(conjunt, llegirFitxerTargets, vaixell)
Responsibilities:

Genera la llista de conjunts de dades que es representaran en la gràfica d'anàlisi demanada per l'usuari.

Exceptions:

Si es produeix un error intern finalitzar l'execució del *filtre* i retornar l'error.

Preconditions:

El *filtre* s'executa si la variable seleccionada pel domini és diferent de *Datetime*.

Postconditions:

Name: xyPolar(conjunt)
Responsibilities:
Calcula el conjunt de dades que retorna el *filtre* i que representarà la gràfica d'anàlisi. Per cada interval seleccionat del domini es calculen la mitja dels valors tant del domini com del rang que hi pertanyen. El conjunt de parells de valors obtinguts és el resultat del procediment.
Exceptions:
Si es produeix un error intern finalitzar l'execució del *filtre* i retornar l'error.
Preconditions:
El conjunt d'entrada no és buit.
Postconditions:

6.4 Disseny de la base de dades

El disseny de la base de dades del sistema consta de tres taules. Una taula s'anomena *regates* i conté tota la informació que prové de la importació de les regates. La segona taula s'anomena *importacions*, és una taula interna del sistema i guarda un registre per cada importació d'una regata amb informació suficient per identificar totes les files que en formen part a la taula *regates*. L'última taula és la de *veles*, aquesta manté la informació sobre les veles que l'usuari introdueix en el sistema.

Així l'usuari és capaç d'interaccionar amb la taula de regates important noves regates o esborrant-ne d'existents en el sistema. També pot treballar amb la taula de veles inserint o esborrant veles en el sistema.

La taula de veles no està relacionada ni amb la taula de regates ni la d'importacions. Així és possible mantenir la informació de les veles usades en les regates en el cas d'eliminar la vela del sistema.

Per altra banda, la taula *importacions* és gestionada internament pel sistema i no és accessible per l'usuari.

La següent figura mostra el model lògic de base de dades amb les tres taules i la informació que contenen:

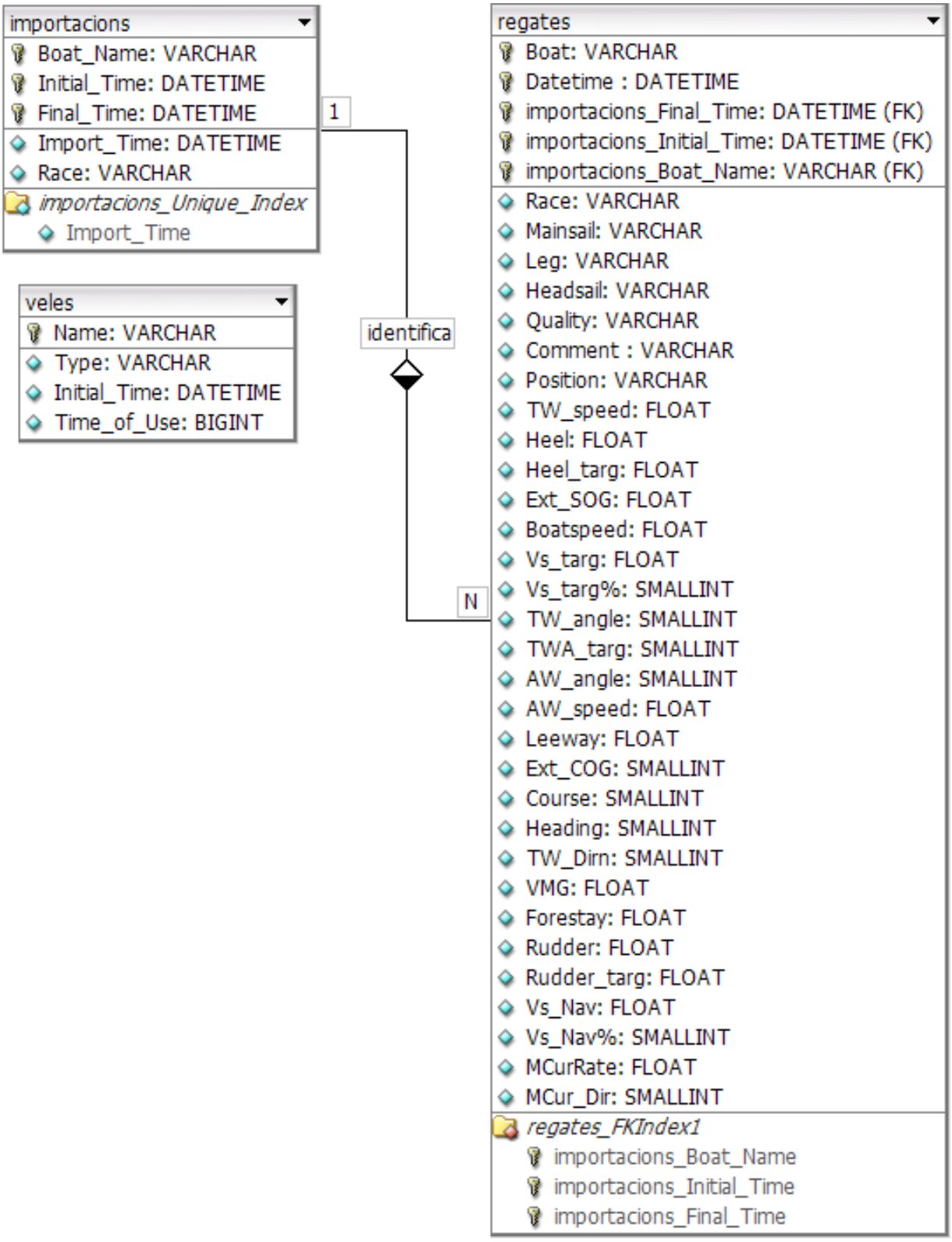


Figura 61: Disseny de la base de dades del sistema

7 Implementació

La implementació és la realització d'una especificació tècnica o algoritmes com un programa, component de programari o un altre sistema de còmput.

Aquesta és una de les etapes en les que es detecten més problemes a l'hora d'aplicar una arquitectura o els patrons de disseny escollits.

En aquest capítol s'aborden aspectes com de quina manera s'ha programat el sistema, els problemes que han sorgit i la seva solució.

També es mostraran alguns diagrames parcials de classes d'implementació per donar una visió global del sistema. Tanmateix, per algunes classes s'obvia l'especificació dels seus atributs i operacions degut a les dimensions que adquiririen.

7.1.1 Diagrama de classes dels components gràfics

El següent diagrama mostra les classes de la llibreria *JFreeChart* afectades per l'ampliació de funcionalitats realitzada per donar suport a les necessitats del sistema.

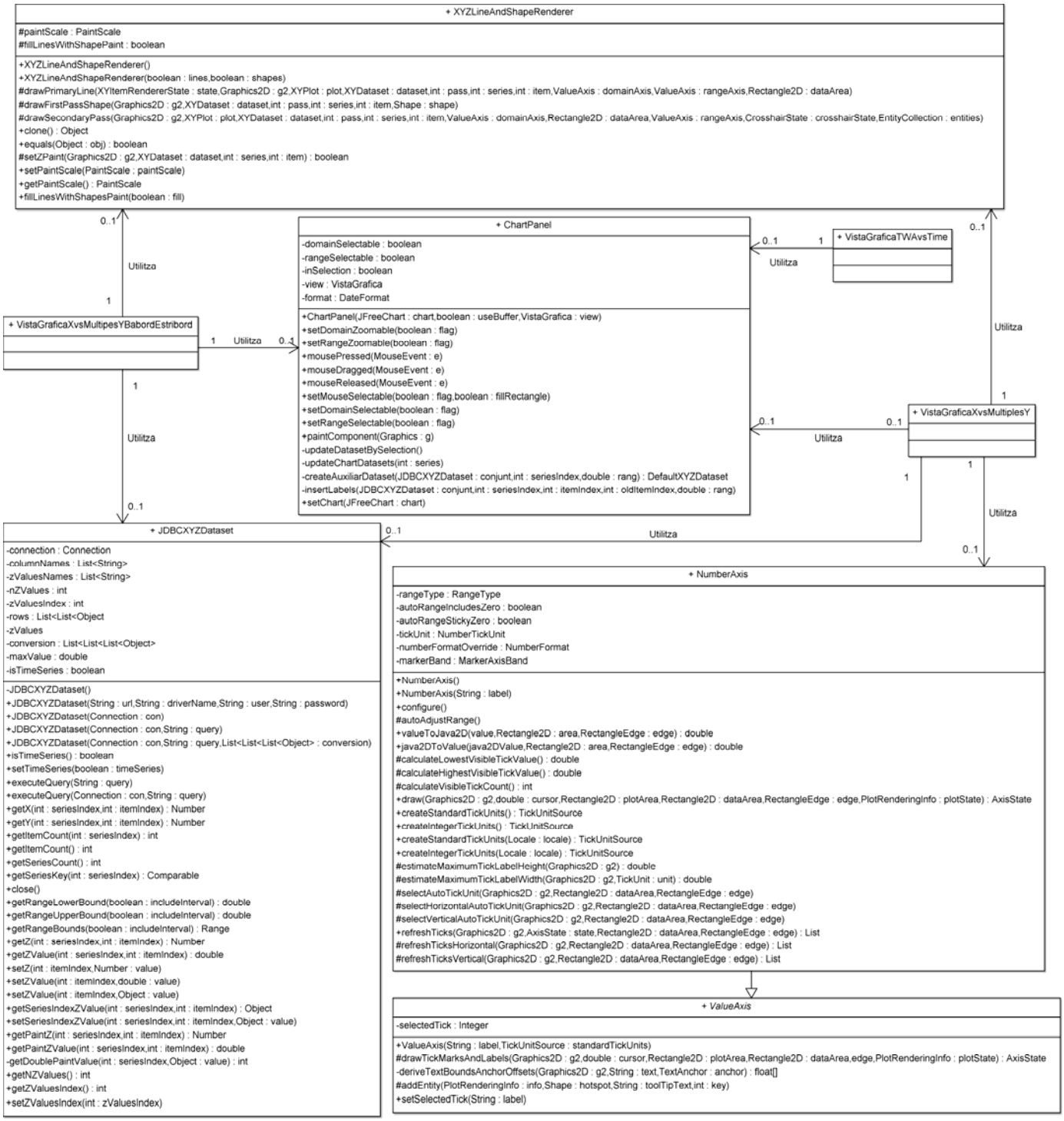


Figura 62: Diagrama d'implementació dels components gràfics

7.1.2 Diagrama de classes dels filtres per components *Java Swing*[1]

Diagrama que mostra la relació de la classe *FiltreDocuments*, encarregada de gestionar els components *Swing* de text, i la classe *FiltreFitxers* que filtra els fitxers d'un directori segons la seva extensió, amb les vistes del sistema.

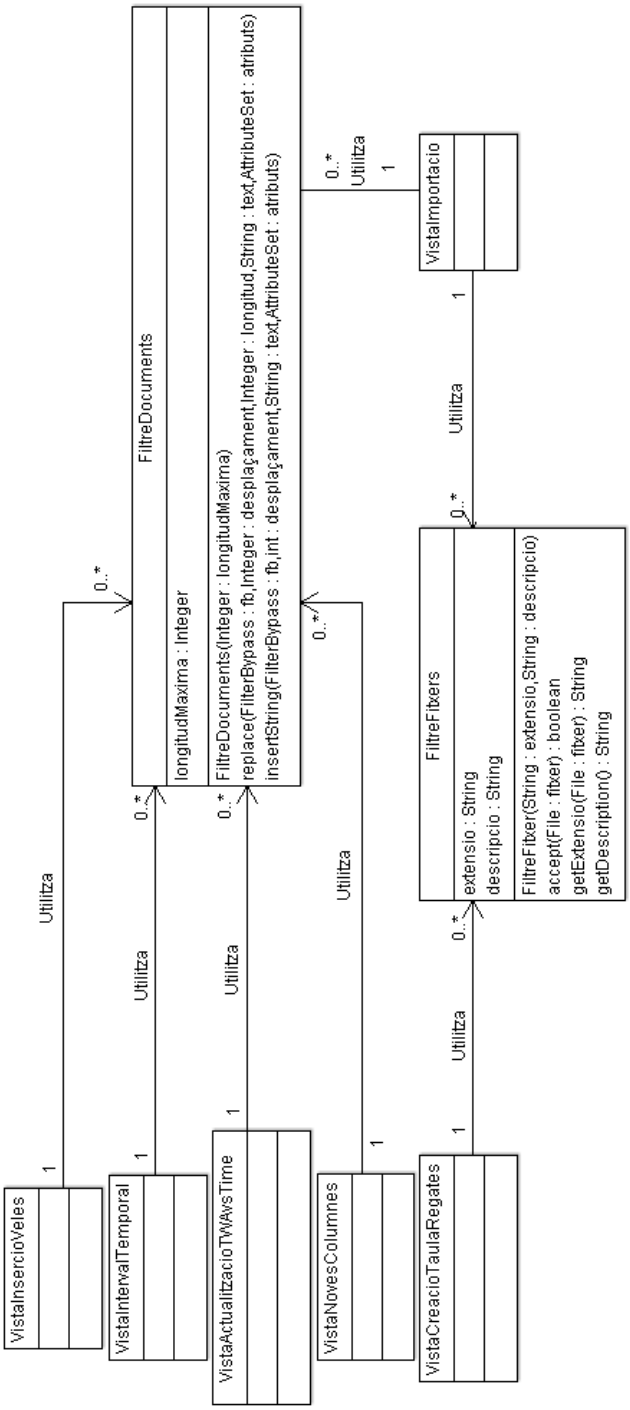


Figura 63: Diagrama d'implementació dels filtres per components Java Swing

[1] Swing és una biblioteca gràfica per a Java. Inclou *widgets* per crear interfícies gràfiques d'usuari com caixes de text, botons, desplegables, llites, taules, etc. Un *widget* és una aplicació petita que té com objectiu donar un accés fàcil a funcions usades freqüentment i proveir d'informació visual.

7.1.3 Diagrama de classes dels filtres del sistema

Diagrama amb la generalització dels *filtres* d'anàlisi de les dades d'una regata.

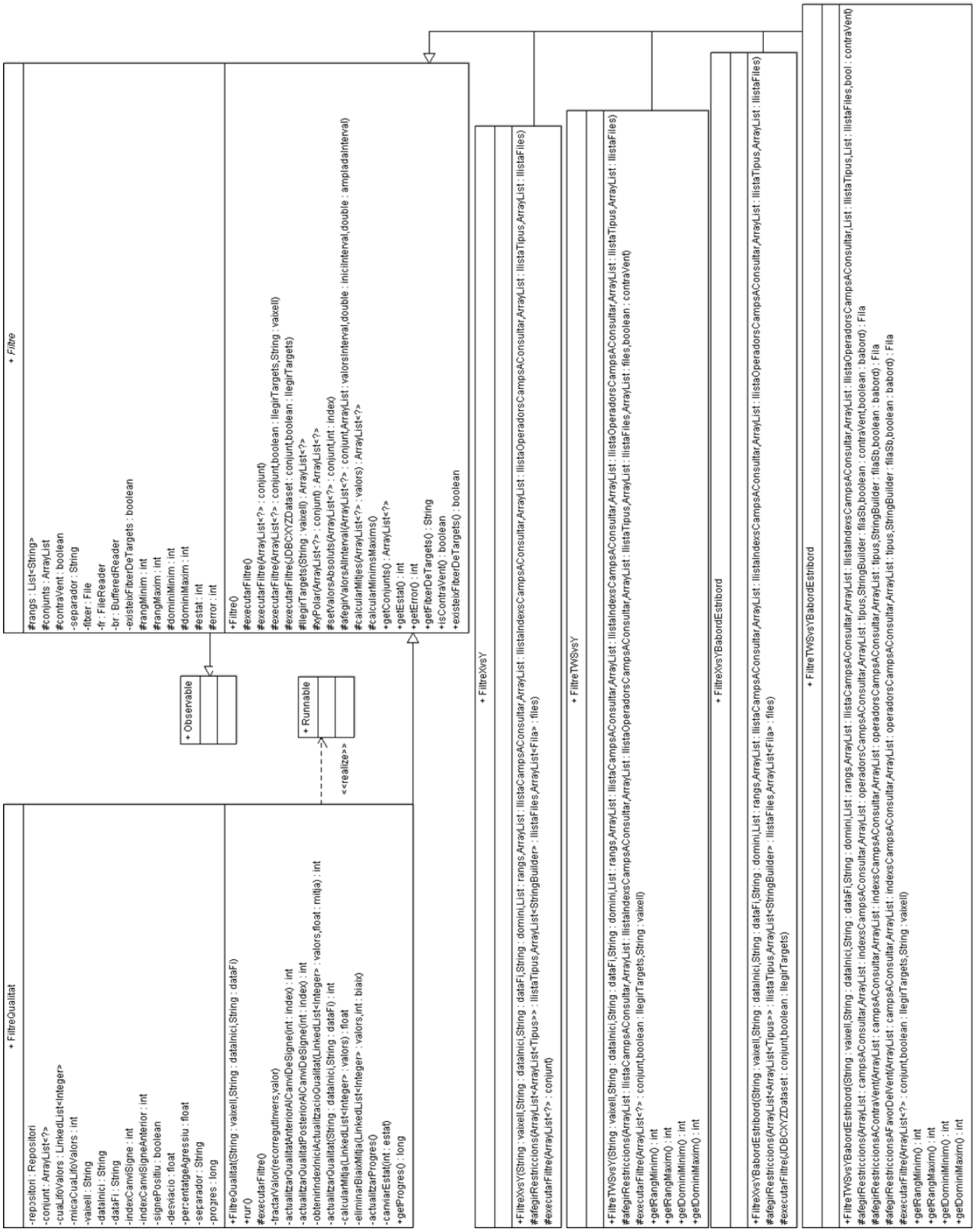


Figura 64: Diagrama d'implementació dels filtres del sistema

El següent diagrama mostra la relació entre les vistes del sistema i els *filtres*. També s'han afegit les relacions entre els *filtres* i les classes que accedeixen a la capa de gestió de dades, com la interfície *Repositori* o la classe *JDBCXYZDataset*.

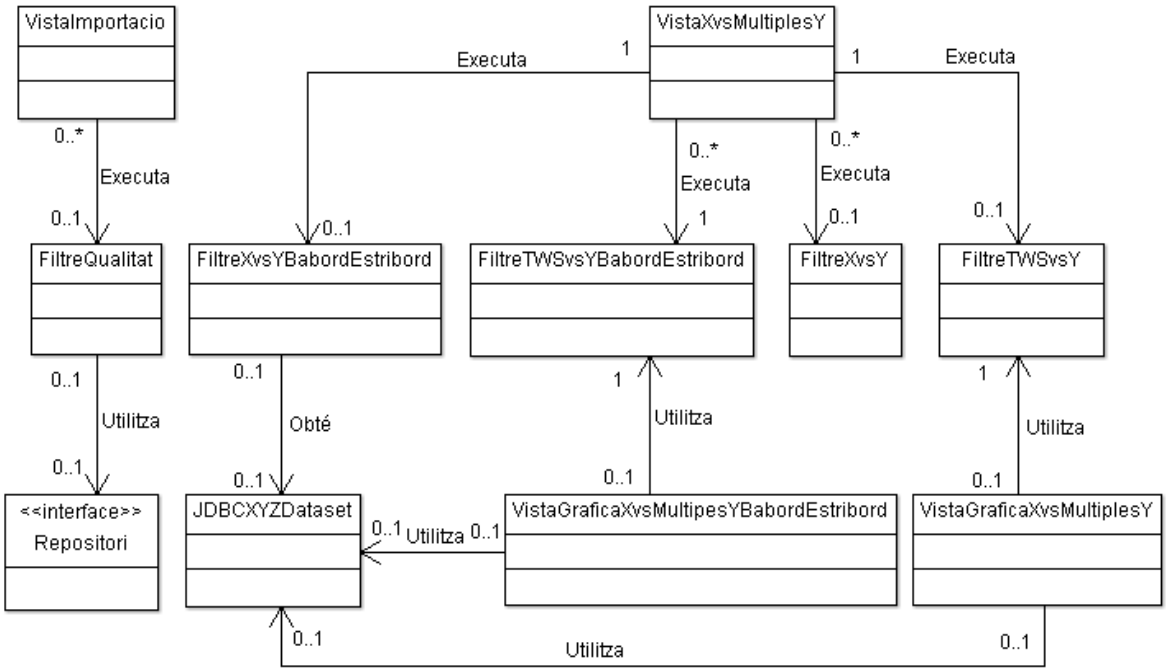


Figura 65: Diagrama d'implementació de relació entre vistes i filtres del sistema

7.1.4 Diagrama de classes de les vistes del sistema

Diagrama de classes que mostra la generalització/especialització de les vistes del sistema.

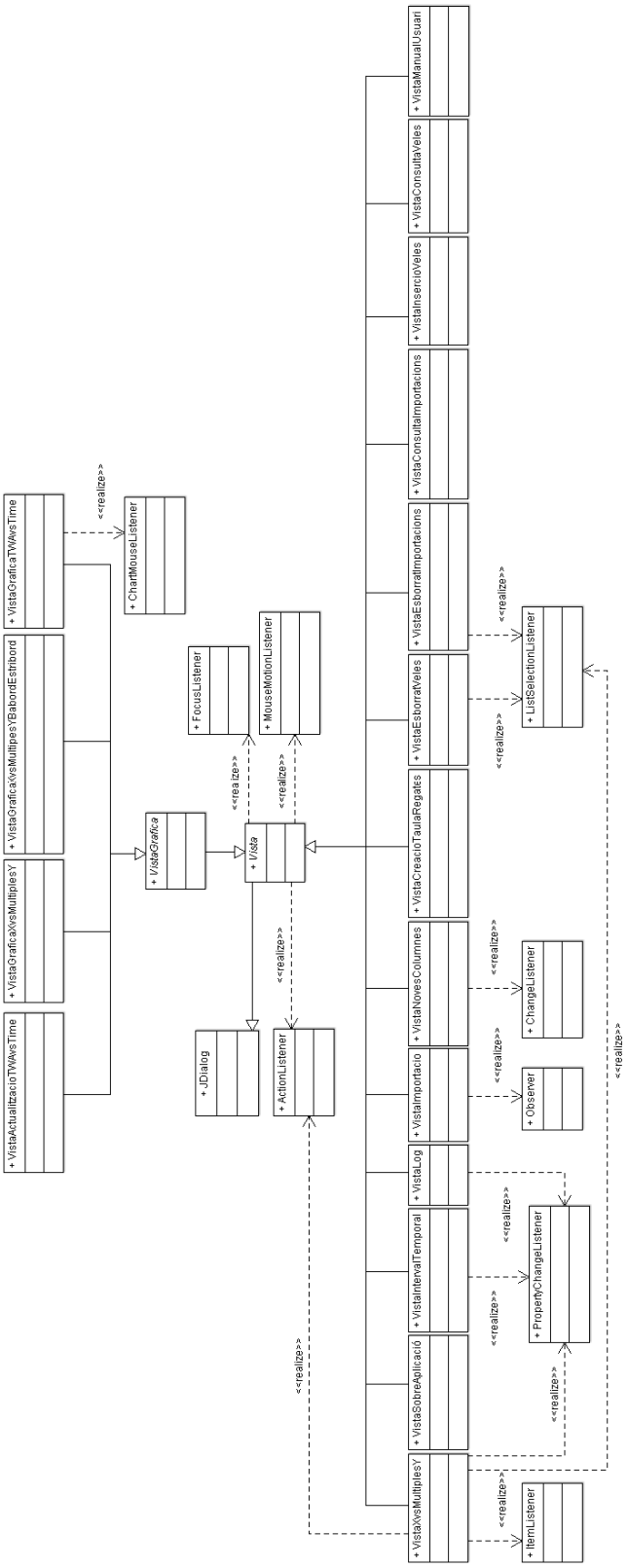


Figura 66: Diagrama d'implementació de la generalització de les vistes del sistema

7.1.5 Diagrama de classes de l'arrencada del sistema

Aquest diagrama mostra el conjunt de classes implicades l'arrencada del sistema i les relacions entre elles.

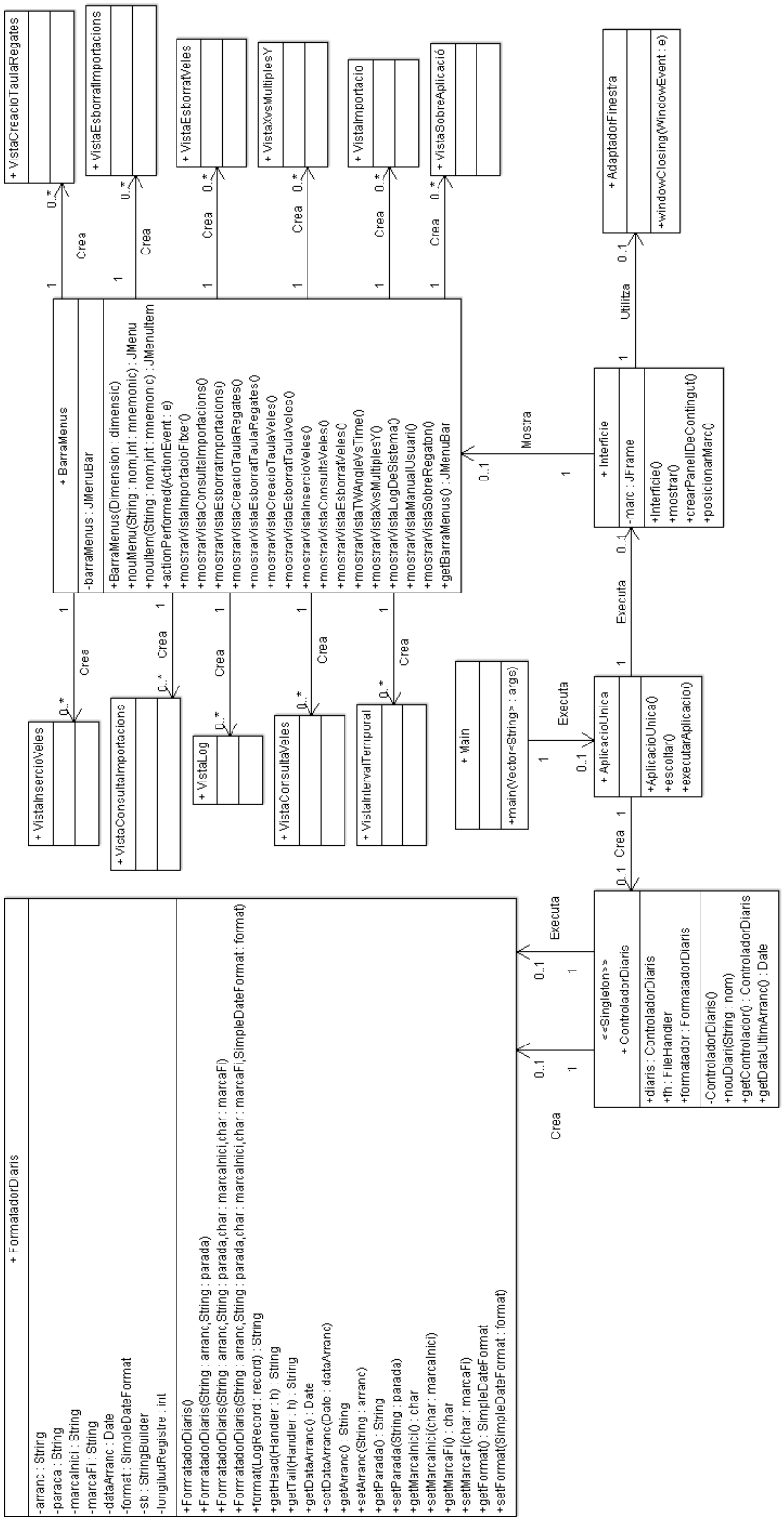


Figura 67: Diagrama d'implementació de l'arrencada del sistema

7.1.6 Diagrama de classes del patró *Observador*

Aquest és el diagrama d'implementació de les classes que utilitzen el patró *Observador*. Les classes *Importador* i *FiltreQualitat* són els subjectes observats i la vista d'importació el seu observador.

Figura 68: Diagrama d'implementació de les classes que usen el patró Observador

7.1.7 Diagrama de la classe *Controlador*

Classe principal del patró Model – Vista – Controlador.

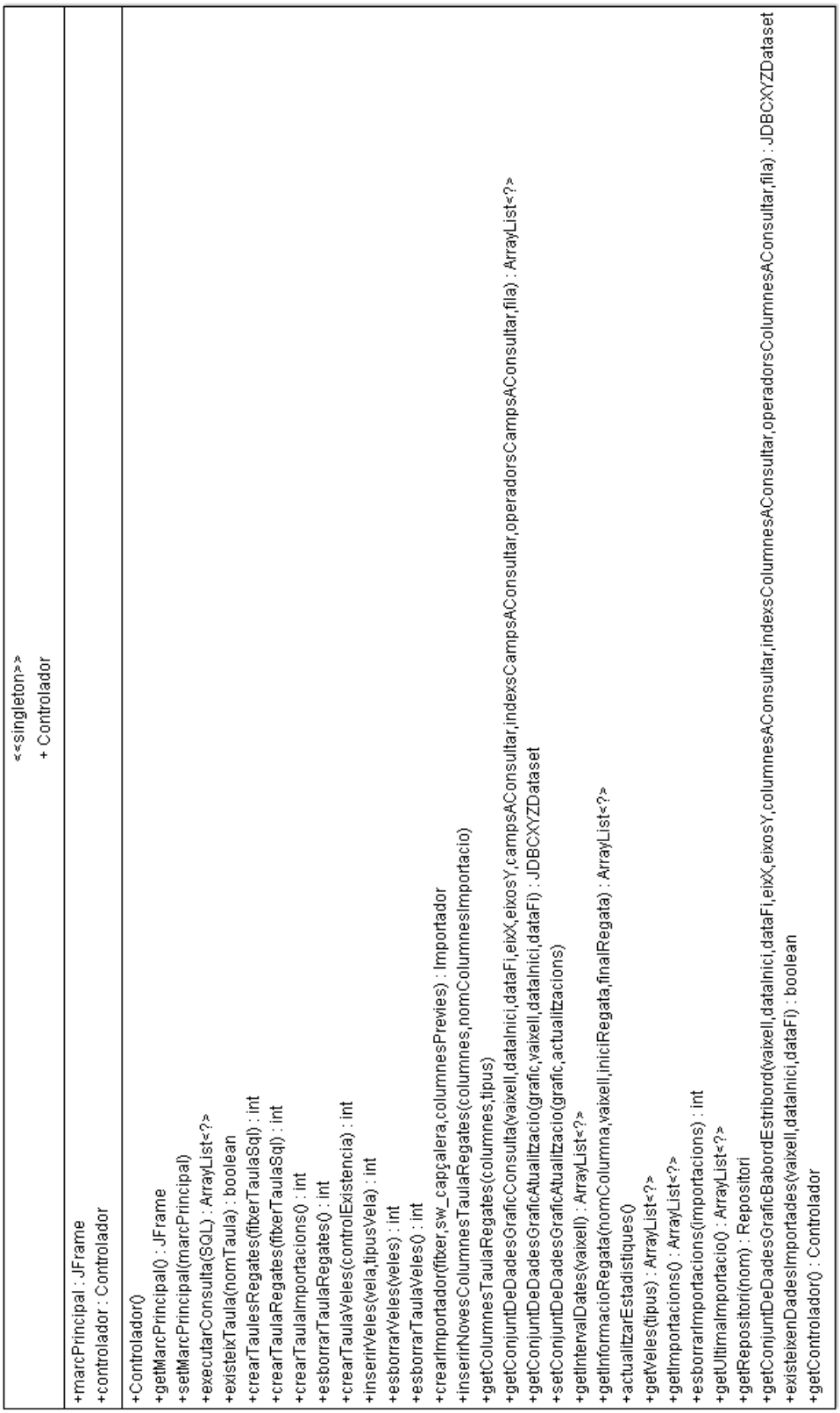


Figura 69: Diagrama d'implementació de la classe Controlador

7.1.8 Diagrama de la classe *Acces*

La classe Acces s'encarrega de construir les consultes al SBGD i retornar els resultats.

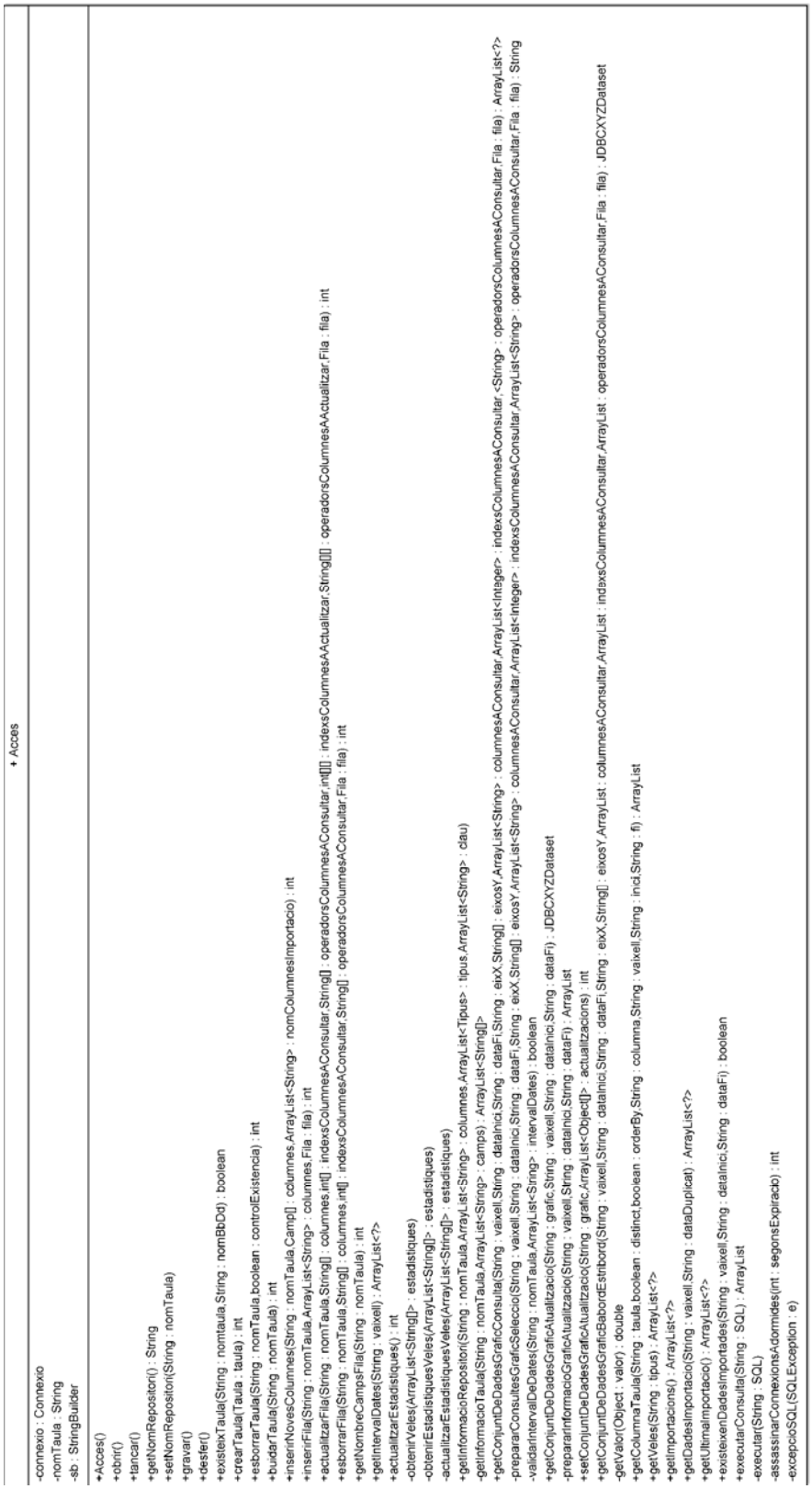


Figura 70 : Diagrama d'implementació de la classe Acces.

7.1.9 Diagrama dels tipus de dades

Diagrama d'implementació de les classes encarregades de convertir la informació dels fitxers de Log al tipus de dades admès pel sistema.

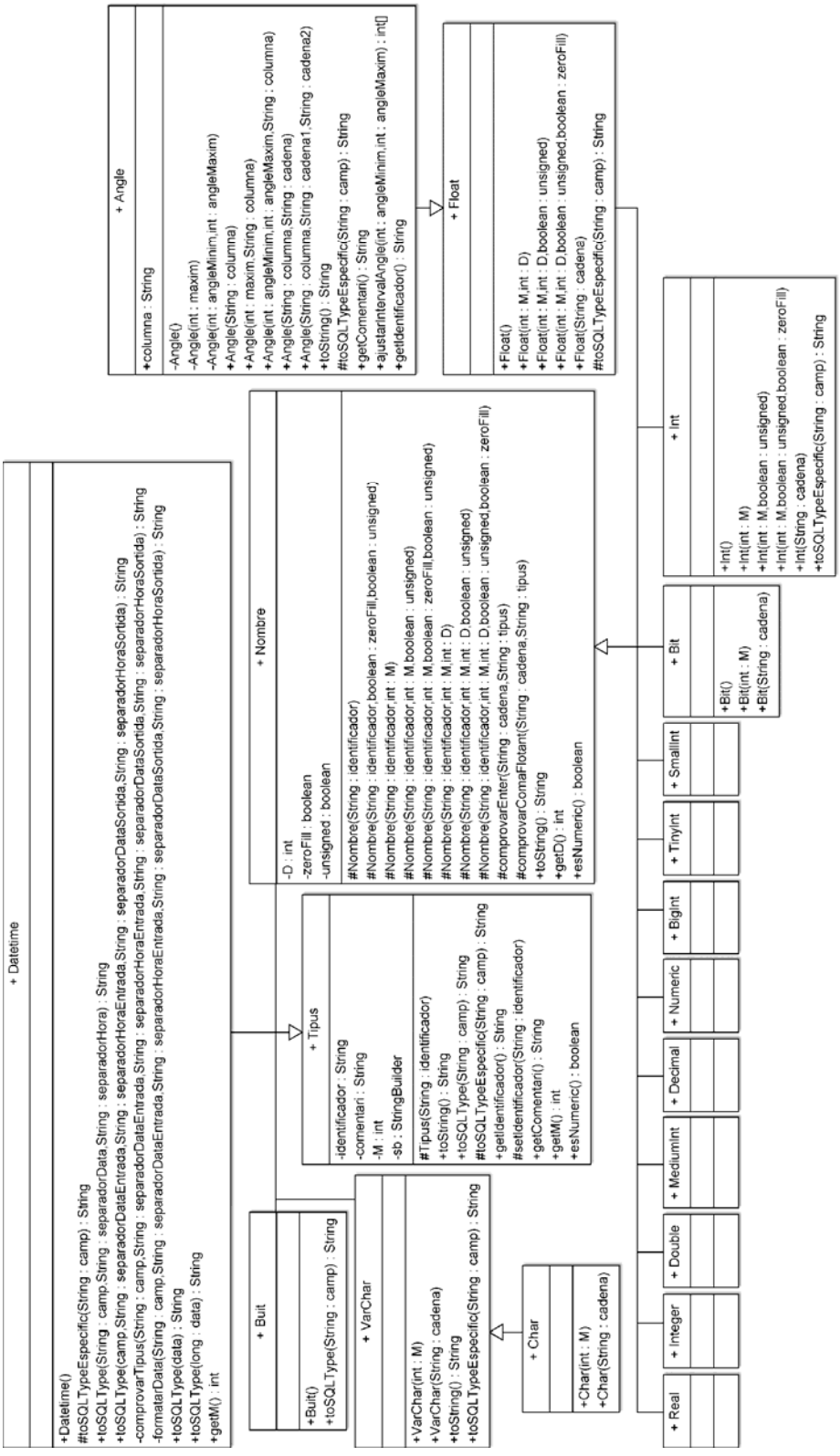


Figura 71 : Diagrama d'implementació de les classes dels tipus de dades

7.1.10 Diagrama de classes amb la resta de classes i les seves relacions

Aquest diagrama mostra la resta de classes usades durant l'execució del sistema.

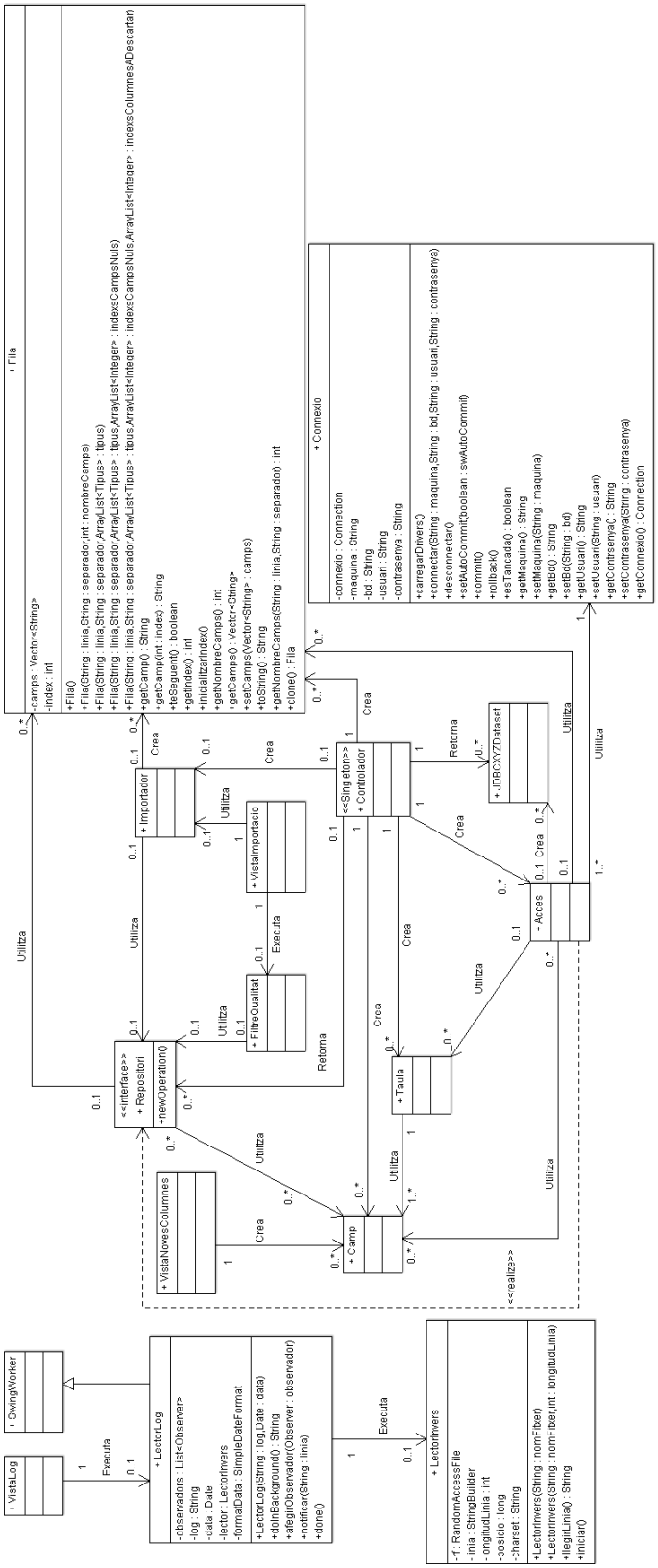


Figura 72: Diagrama d'implementació de la resta de classes del sistema

7.1.11 Problemes detectats durant el desenvolupament

La següent llista documenta les complicacions que han anat apareixent durant la implementació del projecte:

1. Limitacions de les llibreries per generar gràfiques: *JFreeChart* és una llibreria que dóna suport per la programació de gràfiques en *Java*. La versió utilitzada d'aquest producte és la 1.0.12 amb la que s'han produït les següents complicacions:

    a. Està dissenyada per desenvolupar gràfiques de consulta. La necessitat del sistema anava més enllà, i amb les gràfiques *TWA vs Time* era necessària la capacitat d'actualitzar la informació del sistema a partir de seleccions i modificacions sobre la gràfica.

    Per solucionar aquesta situació hem hagut d'ampliar les funcionalitats d'algunes de les classes de la llibreria per a que sigui capaç d'admetre actualitzacions i modificacions, en temps real, a partir de seleccions de l'usuari.

    b. La llibreria no tenia implementat el tractament d'esdeveniments del ratolí sobre les etiquetes dels eixos en gràfiques *XY*.

    S'implementa el tractament d'esdeveniments per aquestes gràfiques de manera molt similar al mecanisme existent per a les gràfiques amb categories.

    c. No existia *Renderitzador*[2] per a gràfiques amb tres eixos, *XYZ*. Dos eixos per a la gràfica bidimensional, i un tercer per la coloració dels punts de la gràfica.

    Es realitzen les modificacions i es creen les classes corresponents per donar suport a aquesta necessitat. Amb això, és possible crear gràfiques amb diversos rangs i colorar els seus punts en funció d'una variable determinada.

2. Limitacions de la llibreria de calendaris: *JCalendar* en la versió 1.3.3 no dóna suport per a seleccions de múltiples dies consecutius ni no consecutius. La solució d'aquest aspecte, ha passat per afegir restriccions addicionals en el codi del projecte controlant les dates seleccionades i comprovant que existeixi informació pels intervals temporals corresponents.

3. Limitacions de *MySQL*: El servidor del *SGBD MySQL* en la versió 5.1.32 realitza l'anàlisi sintàctic de les restriccions *CHECK* però tots els enginys que ofereix l'ignoren. Amb això, tot i que es dóna suport per la definició de

[2] Renderitzat és un terme informàtic usat per referir-se al procés de generar una imatge des d'un model. En termes de visualització a l'ordinador, la renderització és un procés de càlcul complex destinat a generar una imatge 2D a partir d'una escena 3D.

restriccions en la declaració de les columnes d'una taula, no existeix cap implementació que la dugui a terme.
Les solucions que proposa l'equip de suport de *MySQL* per emular el funcionament dels *CHECK* són:

a. L'ús de *triggers*[3] amb *BEFORE*.

b. L'ús de funcions d'usuari que controlin aquesta mancança de l'eina.

La solució adoptada per la implementació del projecte és la segona. Hem implementat una classe per cadascun dels tipus que pot utilitzar el sistema amb els mecanismes necessaris per controlar les restriccions que li corresponen. Per exemple, el sistema ofereix el tipus de dades *Angle*, pel qual són necessàries restriccions sobre l'angle mínim i màxim que admet, o per garantir que un angle no supera els 360º graus.

[3] Un *Trigger* o disparador en una base de dades, és un procediment que s'executa quan es compleix una establerta al realitzar una operació. Depenent de la base de dades, els disparadors poden ser d'inserció, d'actualització o d'esborrat

8 Conclusions

En aquest capítol exposaré tant les conclusions i resultats finals del projecte, argumentant els motius que poden haver influenciat en l'assoliment dels objectius que es van establir en la fase inicial.

Com a punt final, donaré la meva valoració personal sobre la realització del projecte a partir d'una visió global que només a aquestes alçades és possible valorar.

8.1 Conclusions

Cadascun dels capítols i seccions previs del document han detallat i argumentat cadascuna de les etapes del projecte: l'estudi previ de les tecnologies disponibles, l'anàlisi de la competència en el mercat, el pressupost econòmic i les planificacions temporals, l'anàlisi de requeriments, l'etapa d'especificació del sistema, l'etapa de disseny, la implementació i finalment la documentació que inclou tant el present document com el manual d'usuari de l'eina *Regaton*. Cadascun dels temes abordats pretén donar a conèixer amb més detall els aspectes implicats en l'elaboració del projecte.

Ara és el moment d'analitzar els resultats obtinguts i extreure les degudes conclusions. És important mostrar i argumentar si les gràfiques que genera el sistema *Regaton* són fiables i si en realitat l'eina esdevindrà útil per resoldre les necessitats reals del client. Per aconseguir aquesta fita, utilitzaré una comparativa de gràfiques d'anàlisi obtingudes per diverses variables d'estudi a partir de les quals donaré resposta a les qüestions plantejades.

Per obtenir el conjunt de gràfiques en qüestió, el client ha utilitzat les dues eines que té al seu abast. Per una banda, un conjunt de *scripts* programats per executar sobre l'eina *Matlab*. Aquests *srcripts* els va programar ell mateix quan li va aparèixer la necessitat de generar gràfiques d'anàlisi sobre les variables d'estudi recol·lectades de les regates. Per altra banda, el sistema *Regaton* desenvolupat en aquest projecte.

Les dades utilitzades per a aquestes proves corresponen a la regata realitzada per l'equip del que forma part el client a Alacant amb data 15 de Maig de 2009. Del conjunt de dades d'entrada el sistema *Regaton* només ha utilitzat la informació de qualitat mitja i alta, descartant la resta.

A continuació es mostren les gràfiques obtingudes per cadascuna de les eines:

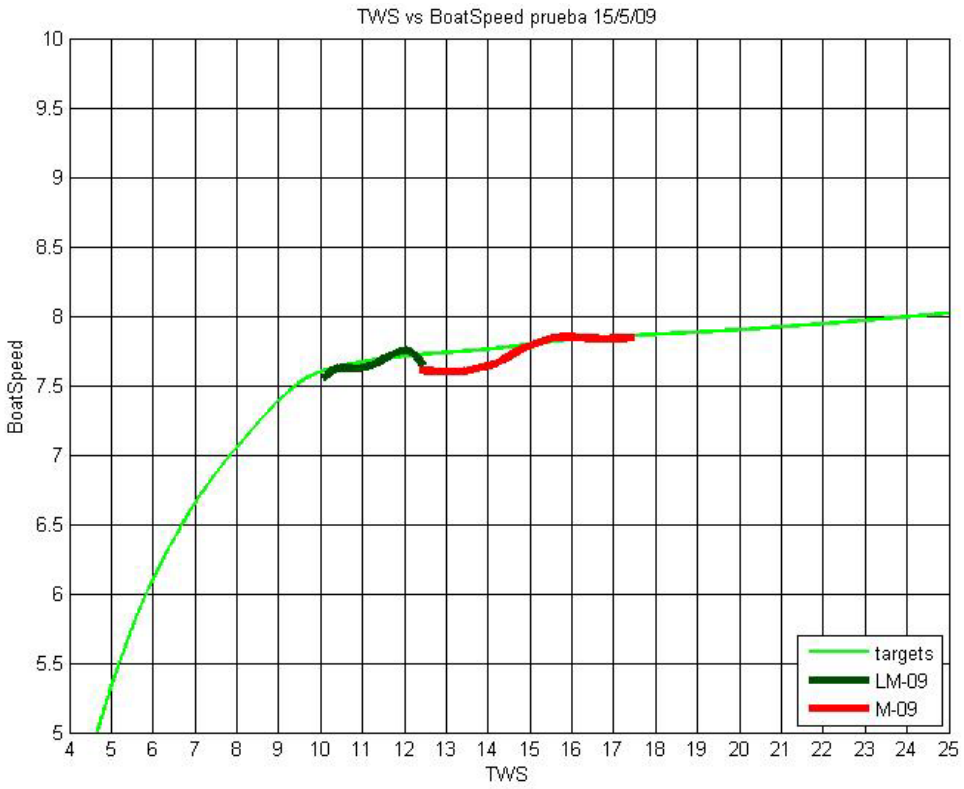


Figura 73: Gràfica d'anàlisi en funció TWS vs Boatspeed, versió scripts

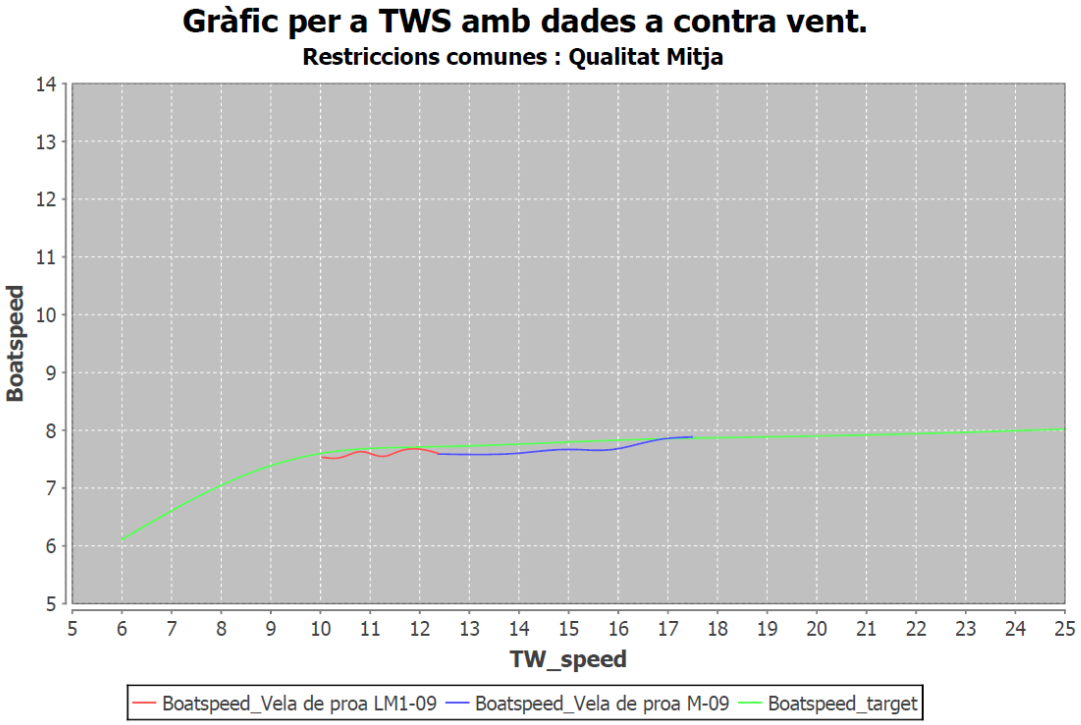


Figura 74: Gràfica d'anàlisi en funció TWS vs Boatspeed, versió Regaton

Ambdues gràfiques mostren la velocitat del vaixell en funció de la velocitat real del vent. En la gràfica del sistema *Regaton* els valors calculats no oscil·len tant com en la

gràfica generada pel *script* de *Matlab*. Tanmateix, podem observar que les dues representacions segueixen la mateixa tendència dins del mateix rang de valors.

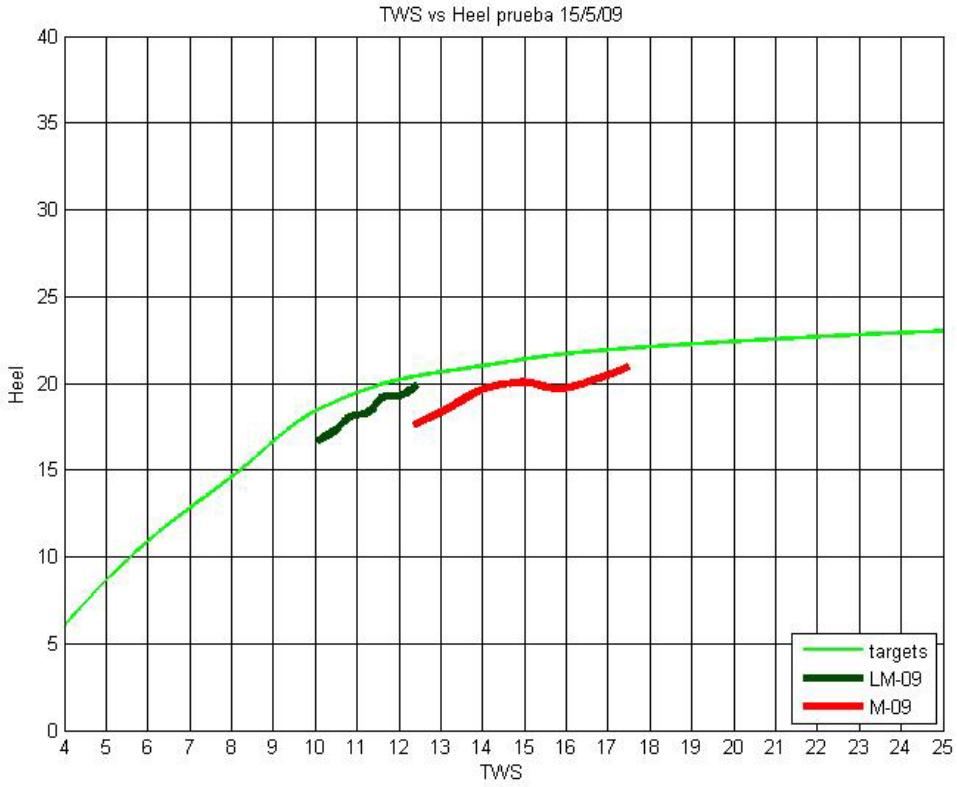


Figura 75: Gràfica d'anàlisi en funció TWS vs Heel, versió scripts

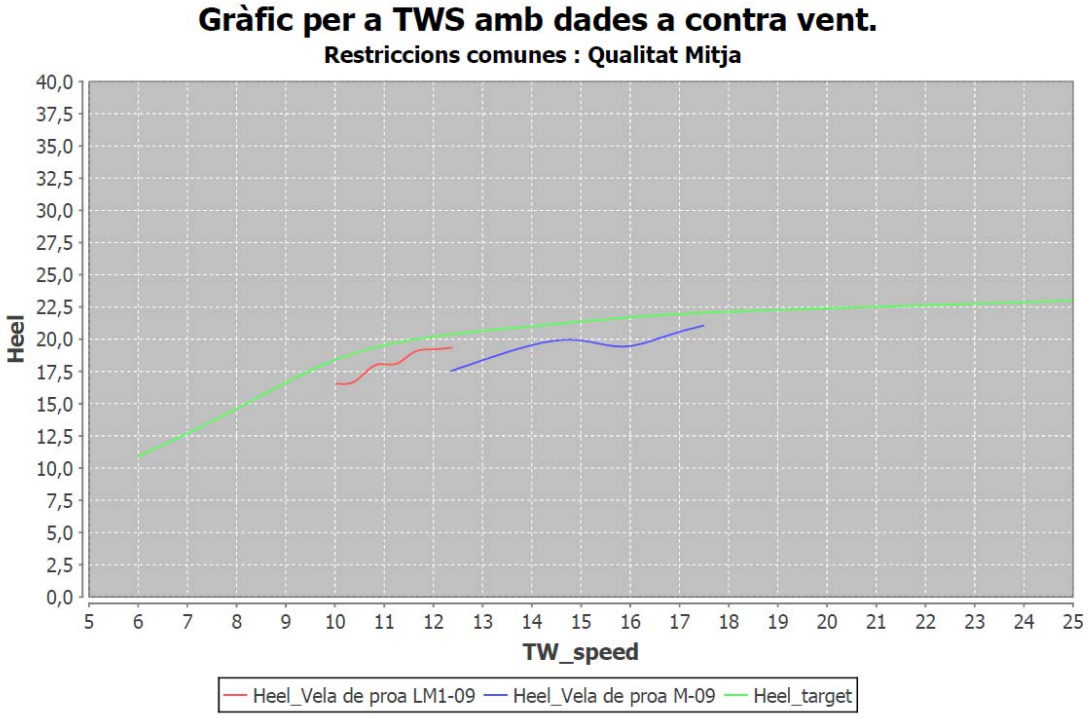


Figura 76: Gràfica d'anàlisi en funció TWS vs Heel, versió Regaton

Aquestes gràfiques mostren l'angle d'escora del vaixell en funció de la velocitat real del vent. En aquest cas les dues representacions són molt similars i les oscil·lacions de la primera gràfica no destaquen tant respecte la segona.

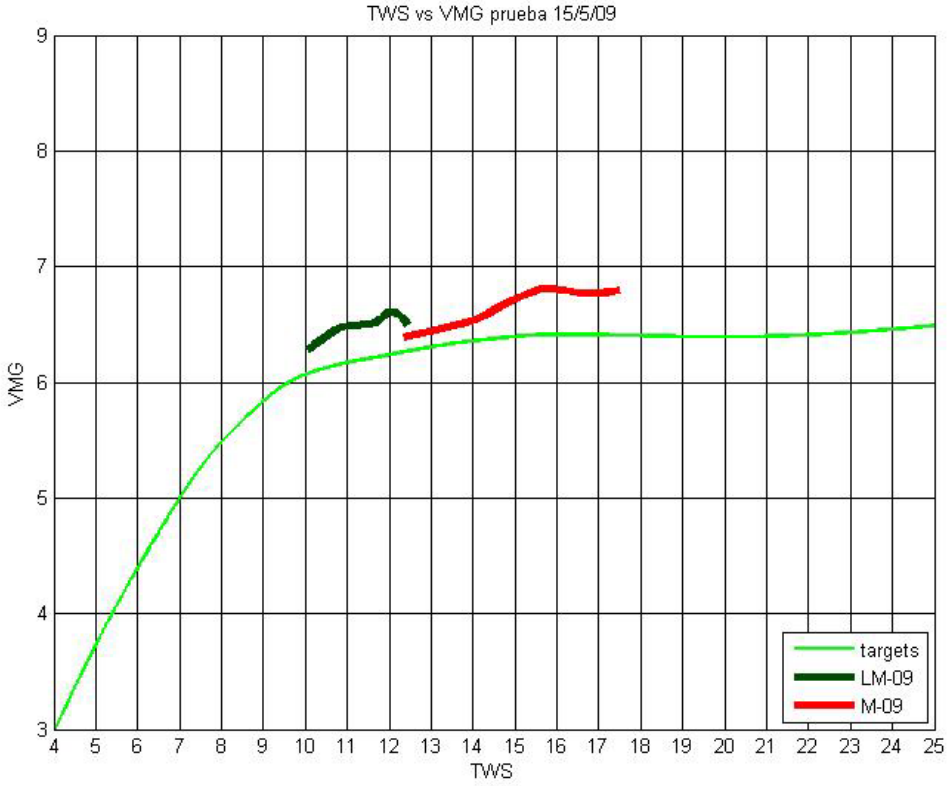


Figura 77: Gràfica d'anàlisi en funció TWS vs VMG, versió scripts

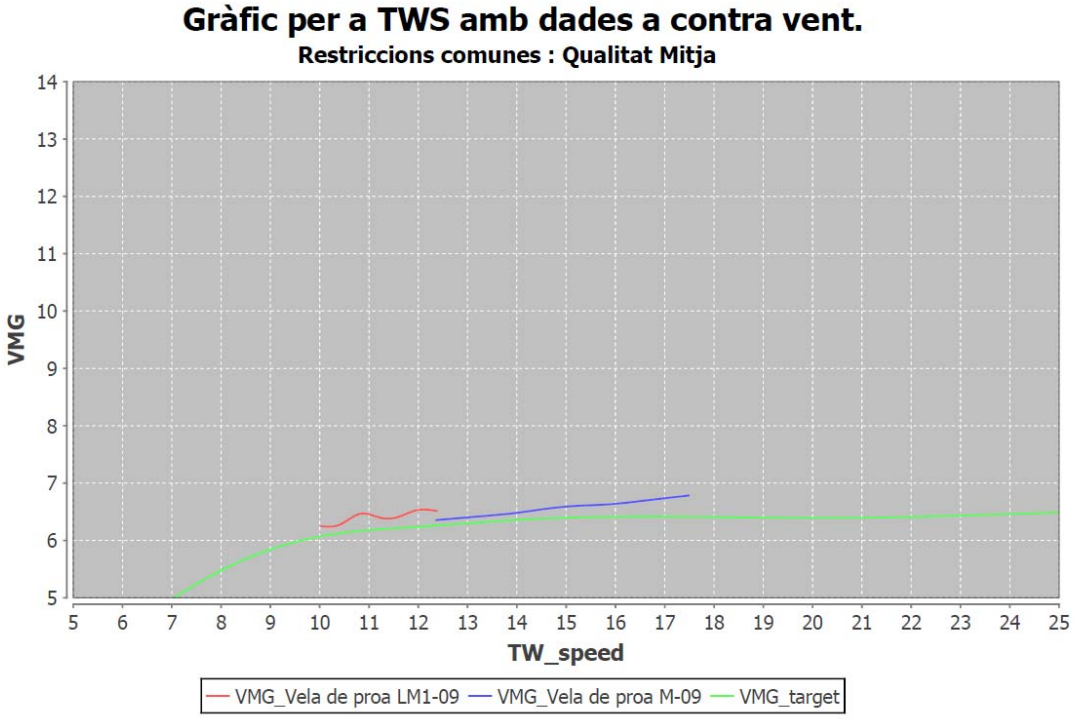


Figura 78: Gràfica d'anàlisi en funció TWS vs VMG, versió Regaton

Aquestes gràfiques mostren la variable *VMG* en funció de la velocitat real del vent. Ara si s'evidencien més diferències, sobretot en el cas de la vela de proa *M-09*, en la que la corba de la primera gràfica és més pronunciada que la del sistema *Regaton*.

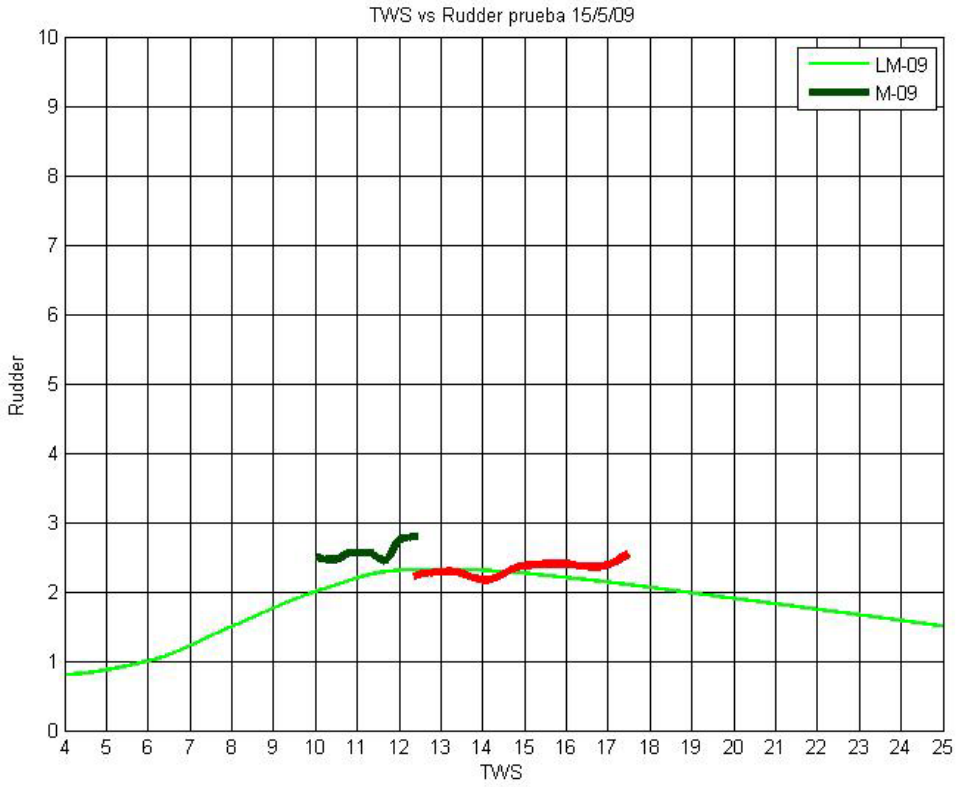


Figura 79: Gràfica d'anàlisi en funció TWS vs Rudder, versió scripts

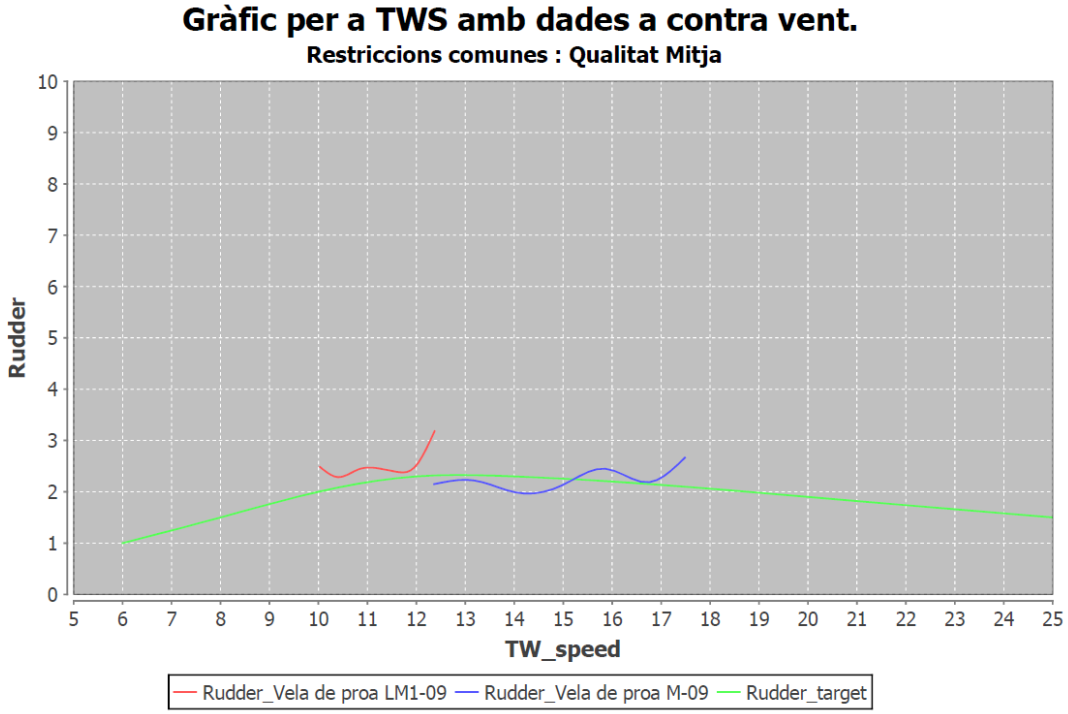


Figura 80: Gràfica d'anàlisi en funció TWS vs Rudder, versió Regaton

Aquestes representacions corresponen a l'angle del timó respecte la proa del vaixell en funció de la velocitat real del vent. Al contrari del que hem observat en les dues primeres gràfiques de la comparativa, ara és la gràfica del sistema *Regaton* la que té oscil·lacions més accentuades. Però, de nou podem veure clarament que les corbes segueixen la mateixa tendència.

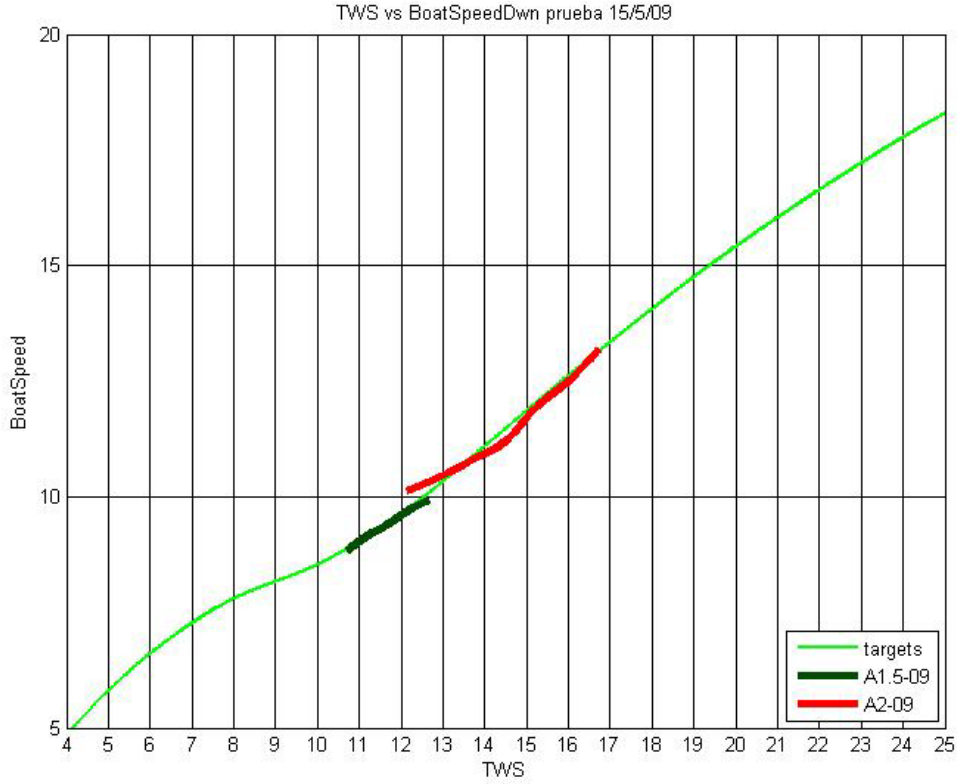


Figura 81: Gràfica d'anàlisi en funció TWS vs Boatspeed Down, versió scripts

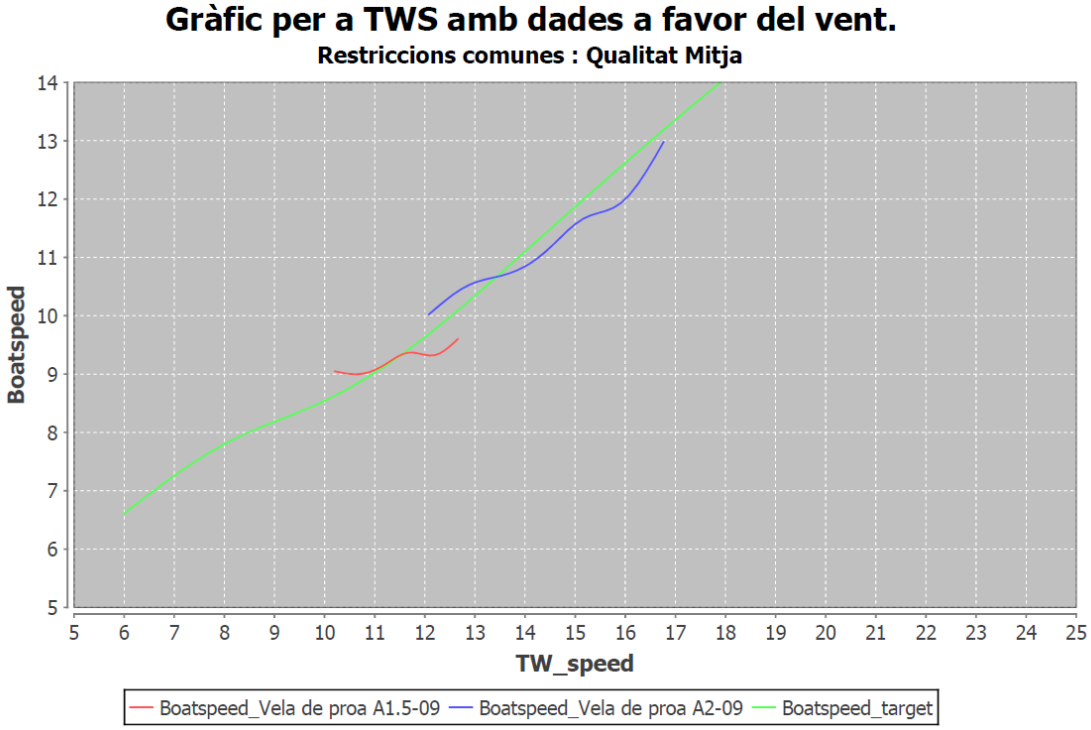


Figura 82: Gràfica d'anàlisi en funció TWS vs Boatspeed Down, versió Regaton

Aquest parell de gràfiques mostra la velocitat del vaixell en funció de la velocitat real del vent. De nou els càlculs del sistema *Regaton* són més sensibles a l'oscil·lació dels valors calculats que en l'altra gràfica.

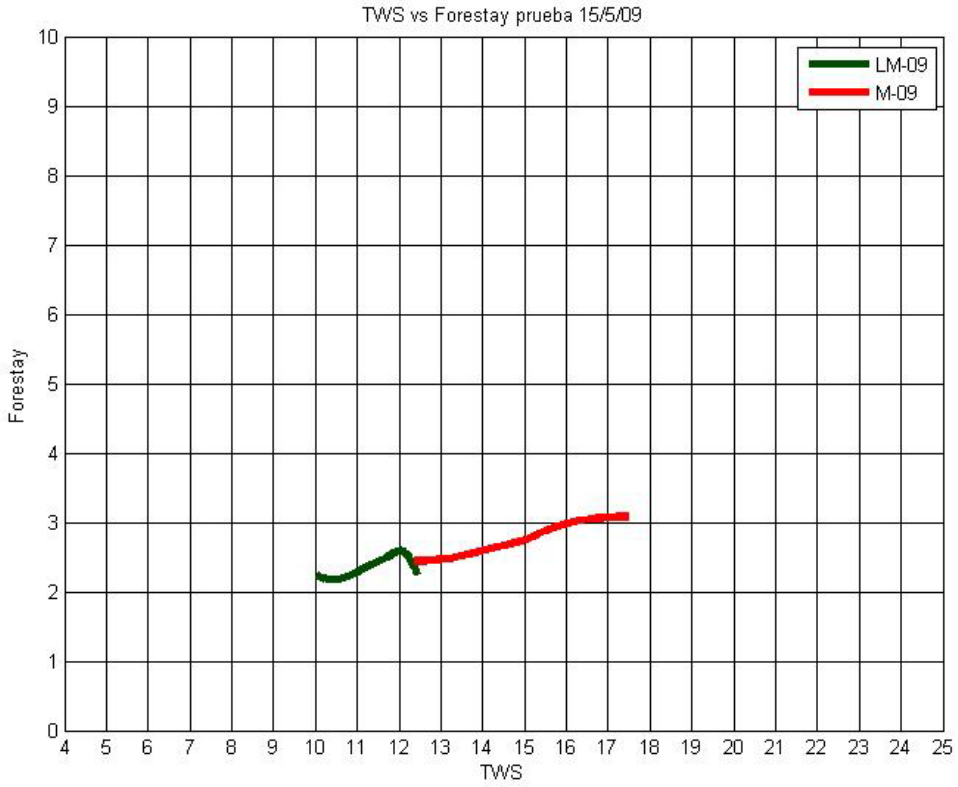


Figura 83: Gràfica d'anàlisi en funció TWS vs Forestay, versió scripts

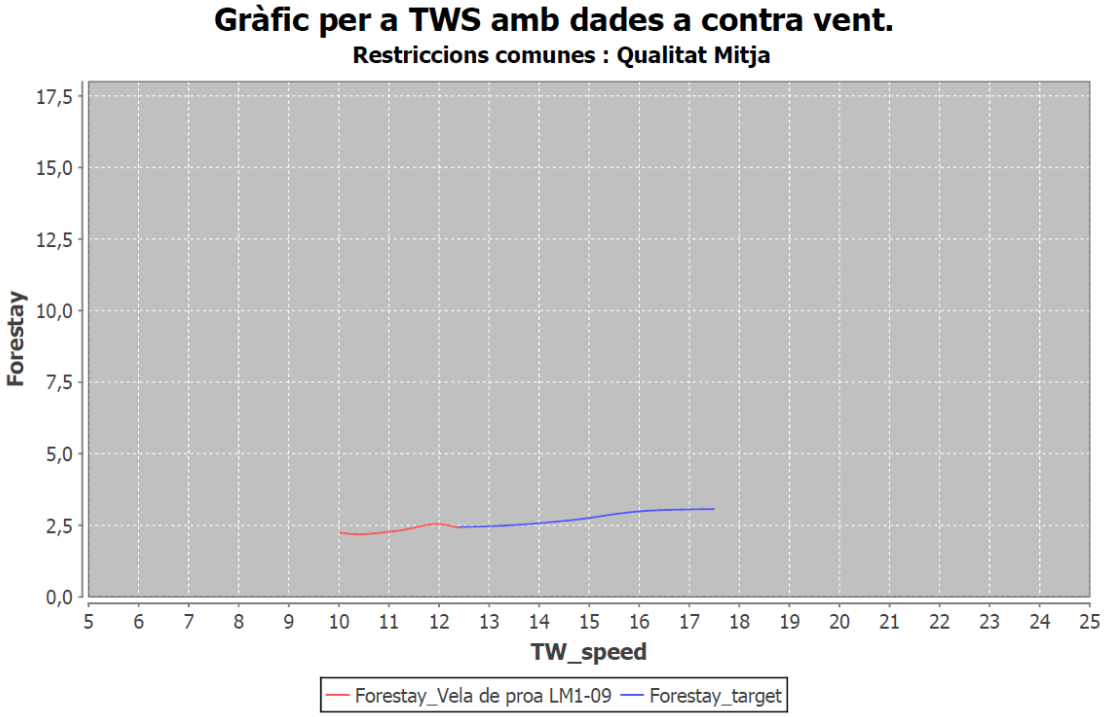


Figura 84: Gràfica d'anàlisi en funció TWS vs Forestay, versió Regaton

Per finalitzar la comparativa, les dues representacions són de la pressió del sensor de l'estai en funció de la velocitat real del vent. Tot i que les corbes tenen la mateixa tendència, en aquest cas la gràfica del *script* de *Matlab* mostra les corbes més pronunciades.

Amb la comparativa finalitzada, podem afirmar que ambdues eines mostren resultats força semblants. Tot i així, és important raonar quins són els factors que influeixen en els càlculs causant les diferències entre unes i altres gràfiques:

1. *Regaton* posseeix el *filtre de qualitat* sobre les dades que assigna qualitat baixa als instants immediatament anteriors i posteriors a cada maniobra realitzada durant les regates. En l'altre cas, aquest algoritme no existeix i el que en el primer sistema és una tasca automàtica es converteix en un procés manual i pesat amb un cost de temps molt més elevat. Amb això podem assegurar que els conjunts de dades d'entrada dels càlculs no són exactament els mateixos.

2. La implementació del *script* de *Matlab* pel càlcul d'aquestes gràfiques es basa en el mateix algoritme que el filtre *TWSvsY* del sistema *Regaton*. Però, ambdues implementacions són diferents. Mentre que el *script* deixa en mans del *Matlab* els càlculs matemàtics, la implementació del *filtre* realitza els càlculs mitjançant funcions pròpies de l'eina.

3. Els tipus de dades d'ambdues eines no són els mateixos. La precisió que utilitza *Matlab* és molt superior a la que posseeix el llenguatge de programació *Java* amb tipus *Double*, que és el tipus de dades usat en els càlculs del *filtre*.

4. L'arrodoniment dels nombres en *Matlab* pot donar resultats diferents al del llenguatge *Java*.

Si valorem els resultats de la comparativa i considerem els arguments que expliquen els factors que potencialment han causat les diferencies entre les corbes representades dels dos sistemes, podem concloure que l'eina *Regaton* és capaç d'oferir resultats amb un grau de fiabilitat elevat.

8.2 Balanç d'assoliment d'objectius

L'objectiu principal era desenvolupar un aplicació capaç d'importar la informació obtinguda a partir dels sensors i el sistema electrònic dels vaixells durant les regates, gestionar-la i emmagatzemar-la persistentment. A partir de la informació emmagatzemada l'usuari havia de poder crear gràfiques d'anàlisi a partir de variables com el *TW_speed* per poder avaluar el rendiment de la resta de variables d'estudi.

Ara que la primera versió de l'eina és una realitat, estic en condicions d'afirmar que hem aconseguit complir tant els objectius obligatoris com els opcionals. S'han implementat totes les funcionalitats i requeriments funcionals demanats pel client. Per altra banda, s'han respectat i seguit en la mesura del possible els requeriments no funcionals establerts.

Hem usat les tecnologies definides al inici del projecte: el llenguatge *Java* amb el *SGBD MySQL*. Tot i trobar-nos amb problemes amb algunes de les eines, com *MySQL* o la llibreria per generar gràfiques *JFreeCart*, els hem solucionat i continuat endavant amb la realització del projecte.

El requeriment d'utilitzar programari lliure no ha estat un impediment per la implementació del sistema. Per altra banda, eines com *Microsoft Word* o *ArgoUML* si tenen certes limitacions que han fet més pesada la redacció de la documentació.

El disseny de la interfície d'usuari s'ha realitzat a partir de les peticions realitzades pel client durant les reunions de seguiment. Gràcies a aquestes reunions l'usuari s'ha pogut familiaritzar amb la interfície i valorar el seu funcionament satisfactòriament.

Hem treballat amb el Model – Vista – Controlador i altres patrons de disseny que han ajudat a que l'eina sigui extensible i més fàcil de mantenir en un futur. Per altra banda totes les classes del projecte estan documentades per facilitar la comprensió del codi i ajudar el màxim possible al manteniment de l'eina.

Des d'un punt de vista més subjectiu i personal, l'esforç i el temps dedicats en el desenvolupament del projecte han valgut la pena. Crec que hem aconseguit una versió inicial de l'eina molt satisfactòria, que ofereix una solució real a les necessitats del client i que podrà créixer i millorar amb futures ampliacions.

8.3 Proposta de millores

Són diversos els aspectes de tot programari, en les seves primeres etapes de desenvolupament , que es poden millorar i ampliar. A nivell conceptual, segur que hi ha elements de l'especificació i el disseny que es poden millorar. Les tecnologies utilitzades, el temps disponible per desenvolupar l'eina o els requeriments també poden influir en la implementació de l'eina causant que s'adoptin a vegades solucions que, tot i que són útils o vàlides, potser no són les més apropiades.

En primer lloc, fora bo mantenir-se informat sobre les noves versions del *SGBD MySQL* per esbrinar si aporten actualitzacions importants com la implementació les restriccions de columna *CHECK*. És aconsellable actualitzar la versió del servidor *MySQL* per aprofitar al màxim les seves funcionalitats si es preveu ampliar el sistema en un futur.

En segon lloc, millorar el tractament dels calendaris que ofereix la llibreria *JCalendar* per permetre a l'usuari seleccionar un conjunts de dies discrets no consecutius.

Es podria modificar el procés actual d'inserció de noves columnes a la base de dades o incorporar un nou procés per a que actualitzi automàticament el fitxer de definició de la taula *SQL regates*. Així no seria necessari que l'usuari estigui pendent de tenir el fitxer actualitzat i en un format correcte.

Una altra millora passaria per ampliar les capacitats de la inserció de noves columnes de manera que es suportés més d'un identificador, o nom de la variable de d'estudi, per una mateixa columna.

No estaria gens malament afegir al sistema una nova funcionalitat de configuració de paràmetres globals de l'eina. Aquesta funcionalitat permetria la configuració de valors que actualment són constants fixades del sistema, com el valor dels separadors entre camps als fitxers de text o la variable *threshold* del *filtre TWSvsY*.

L'ampliació de les funcions genèriques d'inserció, actualització i esborrat de files de la classe *Acces* per admetre sentències *SQL* més complexes és una millora molt interessant i que en un futur pot esdevenir important. Actualment aquests procediments donen suport per a sentències amb operadors *AND*, però no admeten operadors *OR*.

Actualment les gràfiques que genera el sistema només poden desfer el *zoom* retornant l'escala de la gràfica al cent per cent. Permetre que l'usuari pugui desfer el *zoom* fins on ell ho desitgi, envers de mostrar sempre la gràfica al complet, esdevé una millora a tenir en compte si considerem que es tracta d'una eina d'us comú en qualsevol de les gràfiques.

Proporcionar dades numèriques addicionalment a la informació gràfica. Aquesta millora permetria a l'usuari obtenir mesures de certes columnes per conjunts de dades que compleixin determinades restriccions.

De la mateixa manera que seria desitjable que el fitxer de definició de la taula de regates s'actualitzés automàticament també es podria automatitzar l'actualització dels fitxers de *targets* a partir d'un conjunt de valors. Un cas on té molt de sentit aquesta millora és quan un vaixell està navegant, durant els entrenaments o en les regates de competició, per sobre dels valors objectiu donats pel dissenyador.

El llenguatge *Java* no destaca pel seu alt rendiment. Si tenim en compte que la generació de gràfiques pot tractar volums de dades de dimensió considerable, en noves versions es podria revisar el codi font i si és possible optimitzar-lo per millorar el rendiment de les classes que treballen amb molta informació.

8.4 Conclusions personals

Les últimes línies, però no per això les més fàcils.

Des de la primera reunió, en la que vaig conèixer el projecte, no he deixat d'aprendre ni un instant. He après a utilitzar noves eines i ampliat coneixements sobre altres que ja coneixia. He adquirit nous coneixements sobre el llenguatge *Java*, amb el que no havia tingut l'oportunitat de treballar gaire fins ara. He hagut d'ampliar funcionalitats de les llibreries de generació de gràfiques, feina que m'ha portat força mal de caps però que alhora m'ha donat la satisfacció de demostrar-me a mi mateix que podia fer-ho. Però, també he tingut l'oportunitat d'obrir les portes a un món que fins ara m'era aliè, la navegació d'alt nivell.

El meu propòsit sempre ha estat ser a l'alçada del projecte. Assolir els objectius no era la meva única fita. He intentat abordar cada fase del projecte amb cura i oferint el millor de mi. Estic segur que l'eina pot millorar, però em sento molt satisfet. He donat el màxim de mi i m'he esforçat fins el final.

Per concloure, arriba l'hora de tancar aquesta etapa de la meva vida i em complau fer-ho sabent que des de la primera classe magistral que vaig rebre en una de les aules de la facultat no he deixat de lluitar ni un sol moment per arribar on sóc.

9 Glossari

Acoblament: És una mesura del grau de connexió, coneixement i dependència d'una classe respecte d'altres.

Aleta: Cadascuna de les dues parts corbes dels costats d'una embarcació més pròximes a la popa.

Arboradura: Conjunt de pals, botavares, etc., destinat a sostenir les veles.

Babord: Costat esquerre d'una embarcació respecte a la línia de crugia, mirant de popa a proa.

Balisa de desmarcatge: Balisa que es col·loca molt a prop de la balisa de sobrevent per ajudar als velers a virar-les.

Balisa o boia: Element flotant de senyalització nàutica.

Base de dades: Conjunt de dades que pertanyen a un mateix context i emmagatzemades sistemàticament pel seu posterior ús.

Bordada: Camí recorregut per un veler entre dues virades en cenyir.

Botavara: Perxa horitzontal subjectada a un pal per l'extrem de proa i maniobrable per l'extrem de popa.

Buc: Cos d'una embarcació, de fusta o fibra sintètica i desproveït de màquines i aparell, que en determina la flotabilitat i la velocitat.

Cenyida: Tram del recorregut d'una prova durant el qual un veler cenyeix.

Cenyir: Navegar amb el vent de proa situant-se en el menor angle possible respecte a la direcció del vent.

Cohesió: És una mesura del grau de relació i concentració de les diverses responsabilitats d'una classe.

Compàs: Instrument d'orientació que serveix per a saber la direcció del meridià magnètic, amb l'objectiu de prendre'l com a línia de referència per a determinar el rumb.

Consistència: Qualitat d'un sistema que és coherent, estable, robust.

Corredera: Instrument que serveix per mesurar la velocitat d'una embarcació en marxa.

Eixàrcia: Conjunt de caps utilitzats a bord d'un vaixell.

Empopada: Navegació d'un veler que rep el vent per la popa.

Encapsulament: Ocultació de l'estat, és a dir, de les dades d'un objecte de manera que només es puguin canviar mitjançant les operacions definides per l'objecte.

Equip de popa: L'equip de popa d'un vaixell de regates el formen el patró, el tàctic i el navegant. A vegades s'afegeix al grup un estrateg.

Escora: Inclinació lateral d'una embarcació, provocada per l'acció del vent o per un canvi de posició dels tripulants.

Estai: Cable que va de l'extrem del pal major a la proa del vaixell.

Estratègia: Forma de prendre les decisions a mig i llarg termini, com l'elecció de la ruta a seguir o els canvis realitzats depenent dels vents i els corrents que apareguin.

Estribord: Costat dret d'una embarcació respecte a la línia de crugia, mirant de popa a proa.

Extensibilitat: Factor de qualitat del programari que consisteix en la facilitat d'adaptació del programari a nous requisits o canvis en l'especificació.

Feedback: Conjunt de reaccions o dades que l'equip de l'embarcació retorna als dissenyadors per tal que modifiquin els seus vaixells, veles, etc.

GPS: Sistema de posicionament global. Permet determinar la posició d'un objecte en tot el món amb una precisió de centímetres.

Impuls vèlic: La massa d'aire, que es desplaça per una vela corbada, produeix un augment de la pressió en el seu costat convex i una succió en el costat còncau. Com a resultat d'això, en tots i cadascun dels punts de la vela es produeixen petites forces, pràcticament perpendiculars a ella. Aquestes forces es poden considerar en una única força resultant anomenada impuls vèlic.

Interfície: Dispositiu informàtic que permet comunicar dos sistemes que no parlen el mateix llenguatge.

JVM: *Java Virtual Machine.* És la màquina virtual de *Java*.

Layline: Línia recta des de la posició actual del vaixell fins la balisa.

Línia de crugia: Línia imaginària d'una embarcació traçada de proa a popa seguint l'eix longitudinal, que la divideix en dos costats iguals.

Llarg: Cadascun dels trams del recorregut d'un veler coberts navegant per l'aleta.

Logger: Equip electrònic que enregistra la informació dels sensors.

Multiplataforma: Terme utilitzar per referir-se a un sistema que pot funcionar en diverses plataformes.

Nàutica: Relatiu o pertanyent a la navegació.

Pal major: Pal principal d'una embarcació, que és també el més gros.

Perxa: Cadascuna de les barres de fusta, ferro o acer accessòries que formen l'arboradura d'una embarcació, a excepció del pal major.

Popa: Part posterior del buc d'una embarcació.

Porta: Element format per dues boies.

Proa: Part anterior del buc d'una embarcació.

Programari lliure: És el programari que pot ser usat, estudiat i modificat sense restriccions, i que pot ser copiat i redistribuït bé en una versió modificada o sense modificar sense cap restricció, o bé amb unes restriccions mínimes per garantir que els futurs destinataris també tindran aquests drets.

Programari tàtic: Programari dissenyat i construït per ajudar a l'equip de popa a prendre les decisions tàctiques.

Regata: Cursa per a embarcacions.

Vela de creuer: Les embarcacions de creuer són les que tenen, normalment, major eslora, que es mantenen en l'aigua tot el temps, ja que la seva orsa i el seu timó no es poden treure de l'aigua. Poden pertànyer a una classe determinada o ser dissenys únics.
La competició més important de la vela de creuer és la *Copa Amèrica*, seguida d'algunes regates transoceàniques com la *Volvo Ocean Race*.

Vela lleugera: Modalitat de vela practicada en un sector d'aigües interiors o de mar a prop de la costa, que requereix un nombre poc elevat de tripulants.

Renderitzat: Terme informàtic usat per referir-se al procés de generar una imatge des d'un model. En termes de visualització a l'ordinador, la renderització és un procés de càlcul complex destinat a generar una imatge 2D a partir d'una escena 3D.

Reusabilitat: Qualitat d'un sistema del que es poden reutilitzar els components.

Rumb: Direcció que segueix una embarcació.

Rumb absolut: Direcció de desplaçament del vaixell en graus.

Rumb relatiu: Desplaçament del vaixell respecte de la direcció del vent.

Sensor: Components electrònics distribuïts arreu de la nau que permeten que el navegant i el tàctic, en temps real, disposin de mesuraments i representacions gràfiques de certs paràmetres.

SGBD: Sistema de gestió de base de dades. Agrupació de programes que serveixen per definir, construir i manipular una base de dades.

Sobrevent: Costat cap a on va el vent.

Sotavent: Costat d'on ve el vent.

Tàctica: Forma de perseguir que el vaixell navegui el millor possible i superi els demès velers, mitjançant ajustos i canvis en la configuració de les veles.

Taula de polars: Descriu el rendiment d'un vaixell per totes les condicions de velocitat i angle real del vent.

Taula (base de dades): Tipus de modelat de dades, on s'emmagatzemen les dades recollides per un programa.

Telemetria: Tecnologia que permet mesurar de manera remota magnitud físiques.

Timó: Instrument de la popa d'una embarcació dotat d'un moviment giratori a babord i estribord, que serveix per a determinar el rumb de l'embarcació segons la resistència a l'aigua que ofereixi la seva posició.

Tolerància a fallades: Característica que permet que un sistema continuï treballant correctament fins i tot després d'una fallada o algun error dins dels seus components.

Trabujada: Canvi de direcció en el que la popa passa per la direcció d'on ve el vent.

Través: Tram del recorregut d'una prova durant el qual un veler navega de través.

Trimat: Ajustar les veles a les condicions de vent per aconseguir que treballin de manera òptima.

Tripulació: Conjunt de persones que governen una embarcació.

UML: Llenguatge unificat de modelatge. És el llenguatge de modelat de sistemes de programari més conegut i usat en l'actualitat. És un llenguatge gràfic per visualitzar, especificar, dissenyar, construir i documentar un sistema.

Vela de proa: Vela triangular col·locada a proa del pal major i envergada a l'estai.

Vela major: Vela que s'enverga al pal major i a la botavara.

Veler: Embarcació proveïda de buc que utilitza les veles com a mitjà principal de propulsió.

Vent aparent o relatiu: Impuls que s'origina de la combinació del vent real i el moviment propi del vaixell.

Vent real: Impuls real que reben les veles.

Virada: Canvi de direcció en el que la proa passa per la direcció d'on ve el vent.

XML: Llenguatge de marques extensible. No és un llenguatge en particular, sinó una manera de definir llenguatges per diferents necessitats.

10 Bibliografia

Continguts generals

- *Viquipèdia*
  http://ca.wikipedia.org/wiki/Portada

- *Google*
  http://www.google.es/

Continguts de caràcter informàtic

- Apunts de l'assignatura Enginyeria del Software Especificació

- Apunts de l'assignatura Enginyeria del Software Disseny I

- Apunts de l'assignatura Enginyeria del Software Disseny 2

- *Java*
  http://www.oracle.com/technetwork/java/index.html

- *JFreeChart*
  http://www.jfree.org/jfreechart/

- *JFreeChart Forum*
  http://www.jfree.org/phpBB2/viewforum.php?f=3

- *JCalendar*
  http://www.toedter.com/en/jcalendar/

- *JavaHelp*
  http://java.sun.com/javase/technologies/desktop/javahelp/

- *Tutorial de JavaHelp*
  http://www.chuidiang.com/java/herramientas/javahelp/ejemplo-javahelp.php

- *Tutorial de JavaHelp*
  http://chuwiki.chuidiang.org/index.php?title=M%C3%A1s_ficheros_de_JavaHelp

- *MySQL*
  http://www.mysql.com/

- *Eclipse*
  http://www.eclipse.org/

- *DBDesigner*
  http://fabforce.net/dbdesigner4/

- *ArgoUML*
  http://argouml.tigris.org/

- *Gantt Project*
  http://www.ganttproject.biz/

- Pàgines consultades per realitzar l'estudi de llenguatges de programació
  http://www.oracle.com/technetwork/java/index.html
  http://www.cprogramming.com/
  http://www.cplusplus.com/doc/tutorial/
  http://www.webtaller.com/construccion/lenguajes/
  http://struts.apache.org/primer.html
  http://www.programacion.com/java/tutorial/struts/2/
  http://www.adaptivepath.com/ideas/essays/archives/000385.php
  http://en.wikipedia.org/wiki/AJAX
  http://www.php.net/
  http://www.python.org/
  http://www.perl.org/
  http://cgi.resourceindex.com/
  http://www.oracle.com/technetwork/java/javaee/jsp/index.html
  http://www.desarrolloweb.com/articulos/831.php
  http://www.asp.net/
  http://www.desarrolloweb.com/articulos/2380.php
  http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html
  http://www.ruby-lang.org/es/
  http://www.oracle.com/technetwork/java/javaee/servlet/index.html
  http://www.programacion.com/articulo/servlets_y_jsp_82
  http://www.jfree.org/jfreechart/
  http://websphere.sys-con.com/node/46516
  http://www.desarrolloweb.com/articulos/832.php
  http://www.whenpenguinsattack.com/2006/02/07/php-vs-java/
  http://tech.rufy.com/2006/06/ruby-vs-java-matter-of-taste_09.html
  http://www.cmswire.com/cms/industry-news/php-vs-java-vs-ruby-000887.php
  http://www.presionblogosferica.com/2008/02/18/java-vs-ruby/
  http://www.paulgraham.com/pypar.html
  http://www.onjava.com/pub/a/onjava/2005/11/16/ruby-the-rival.html
  http://www.linuxquestions.org/questions/2008-linuxquestions.org-members-choice-awards-83/programming-language-of-the-year-695662/
  http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html
  http://en.wikipedia.org/wiki/Model-view-controller
  http://rubyonrails.org/
  http://www.dmh2000.com/cjpr/
  http://lambda-the-ultimate.org/node/1480
  http://www.objs.com/survey/lang.htm
  https://developer.mozilla.org/en/JavaScript
  http://en.wikipedia.org/wiki/Scripting_language
  http://en.wikipedia.org/wiki/Turing_complete
  http://arizonacommunity.com/articles/java_32001.shtml
  http://www.gamarod.com.ar/articulos/tutorial_de_cgi.asp
  http://www.servlets.com/soapbox/problems-jsp.html
  http://www.linuxjournal.com/article/5915
  http://www.infoq.com/articles/From-Java-to-Ruby--Risk
  http://www.desarrolloweb.com/articulos/1325.php
  http://www.mediacollege.com/internet/perl/perl-vs-php.html

http://www.fathersfate.com.mx/wordpress/2006/11/10/introduccion-a-perl-rollo/
http://wiki.rubyonrails.org/rails/pages/HowtoGenerateJFreeCharts
http://kfahlgren.com/blog/2007/04/13/jruby-jfreechart-sparklines/
http://www.rmchart.com/
http://code.enthought.com/chaco/

- Pàgines consultades per realitzar l'estudi de llibreries de generació de gràfiques
http://www.carlosag.net/Tools/WebChart/
http://jpgraph.net/
http://nubyonrails.com/pages/gruff
http://omnipotent.net/jquery.sparkline/
http://matplotlib.sourceforge.net/
http://search.cpan.org/~mverb/GDGraph-1.43/Graph.pm
http://www.gigasoft.com/charting-software.html
http://www.softintegration.com/products/silib/graphlib/
http://www.int.com/products/2d/carnac/chart_component.htm

Continguts de caràcter nàutic :

- *Termcat* – Centre de terminologia, Àrea d'esports nàutics :
http://www.termcat.cat/ca

- Guia per a instruccions de regata :
http://www.diccionario-nautico.com.ar/rrv/reglamento-de-regatas-a-vela-21-guia-para-las-instrucciones-de-regata.php

- Teoría de regatas
http://www.witblits.eu/regatas/parte-11-rrv-representacion-del-recorrido/

- Para Náuticos
http://www.paranauticos.com/capacitacion/materiales-estudio.htm

- Fondear
http://www.fondear.com/Todo_Regatas/Tecnicas_Regatas/Tactica_Estrategia/Tactica_Estrategia.htm

- Lecciones básicas de navegación a vela
http://www.mailxmail.com/curso-lecciones-basicas-navegacion-vela

- *B&G*
http://www.bandg.com/

- *Landfall*
http://www.landfallnavigation.com/

- *Adrena*
http://www.adrena.fr/

- *Seatrack*
http://www.seatrack.co.uk/Default.htm

- *Evolution*
  http://www.evolution-tactic.com/product/welcome/welcome.html

- *GPS Race Driver*
  http://www.gpsracedriver.com/

- *Copa Amèrica*
  http://www.americascup.com/

- *Volvo Ocean Race*
  http://volvooceanrace.com/

- *Barcelona World Race*
  http://www.barcelonaworldrace.org/ca/index.html

11 Apèndix A: Instal·lació

Aquest apèndix exposa els components necessaris per la correcta execució de l'eina *Regaton*. També s'explica quins passos d'instal·lació i configuració del *SGBD MySQL* s'han seguir per fer possible la comunicació amb el sistema. Per últim, es detalla la instal·lació i configuració del sistema *Regaton*.

11.1 Programari requerit

La versió actual del sistema *Regaton* necessita els següents components pel seu correcte funcionament:

1. Les següents llibreries de *JCalendar* 1.3.3: Tractament de calendaris.

    - jcalendar-1.3.3.jar
    - looks-2.0.1.jar

2. Les següents llibreries de *JFreeChart* 1.0.12: Llibreries de generació de gràfiques.

    - gnujaxp.jar
    - jcommon-1.0.15.jar
    - junit.jar
    - servlet.jar
    - jfreechart-1.0.12.jar

    Atenció: La versió actual del sistema ha requerit la implementació de nous algoritmes i modificació d'algunes de les classes de la llibreria per ampliar les seves funcionalitats i cobrir així les necessitats del projecte. S'aconsella la revisió de les classes afectades en el cas de que es desitgi incorporar una nova versió de la llibreria en futures actualitzacions del sistema.

3. Les següents llibreries de *MySQL* 5.1.11: Connector *JDBC*[1] per a *MySQL*.

    - mysql-connector-java-5.1.11-bin.jar

4. Les següents llibreries de *JavaHelp* 2.0.05: Creació d'ajudes en *Java*.

    - jh.jar
    - jhall.jar
    - jhbasic.jar

[1] *Java Database Connectivity*. És una *API* que permet l'execució d'operacions sobre bases de dades des del llenguatge de programació *Java*, independentment del sistema operatiu des d'on s'executi o de la base de dades a la que s'accedeixi, utilitzant el dialecte *SQL* del model de base de dades que s'utilitzi.
Una *API*, o *Aplication Programming Interface*, no és més que una sèrie de serveis o funcions que el sistema operatiu ofereix al programador.

- jhindexer.jar
- jhsearch.jar
- jsearch.jar
- src.jar

5. Servidor *MySQL* 5.1.32 o posteriors.

6. *JDK 6*[2] o posterior pel futur desenvolupament de l'eina. *JRE*[3] en l'última versió existent només per executar el sistema.

7. *Eclipse* pel futur desenvolupament de l'eina o qualsevol altre *IDE*[4] amb facilitats per la programació amb llenguatge *Java*, per exemple *NetBeans*.

11.2 Instal·lació del *SGBD*

La instal·lació del *SGBD MySQL* es realitza mitjançant l'executable disponible a la seva pàgina web http://dev.mysql.com/downloads/. L'executable adequat és el del sistema operatiu *Microsoft Windows XP*.

L'instal·lador ofereix a l'usuari la possibilitat de crear un servei de *Windows* per a *MySQL*, que s'arrenca automàticament al iniciar el *Windows XP*.

Per altra banda, també permet introduir la contrasenya de l'usuari *root* del *SGBD*. Si tots els passos de l'instal·lador finalitzen correctament *MySQL* ja està instal·lat en la màquina.

Si pel contrari, *MySQL* no aconsegueix crear el compte de l'usuari root amb la nova contrasenya cal seguir els següents passos (descrits al web http://dev.mysql.com/doc/refman/5.0/es/resetting-permissions.html) per restablir-la en un sistema operatiu Windows:

1. Entrar en el sistema com Administrador.

2. Parar el servidor *MySQL* si s'està executant. Si el servidor s'executa com a servei de *Windows* aturar-lo des del *Gestor de Serveis*: *Menú Inici -> Panell de Control -> Eines administratives -> Serveis*
   Si el servidor no s'està executant com un servei, podria ser necessari el *Gestor de Tasques* per forcar la seva aturada.

3. Crear un arxiu de text escrivint la següent comanda en una única línia:
   SET PASSWORD FOR 'root'@'localhost' = PASSWORD('novaContrasenya');

   Guardar el fitxer, per exemple amb el nom *C:\mysql-init.txt*.

[2] *Java Development Kit*. Conjunt d'eines, format per programes i llibreries, que permeten desenvolupar (compilar, executar, generar documentació, etc.) programes en llenguatge *Java*.
[3] *Java Runtime Environment*. És un conjunt d'utilitats que permet l'execució de programes *Java*.
[4] *Integrated Development Environment*. Programari que proporciona diverses facilitats per augmentar la productivitat dels programadors durant el desenvolupament d'aplicacions.

4. Obrir una finestra de comandes per obtenir una consola de comandes *DOS*: *Menú Inici -> Executar -> cmd*

5. Considerem que el servidor està instal·lat a *C:\mysql*. Si no és així, cal modificar les següents comandes amb el *path* adequat.

   A la línia de comandes *DOS*[5], executar la següent ordre:

   C:\> C:\mysql\bin\mysqld-nt --init-file=C:\mysql-init.txt

   El contingut de l'arxiu indicat en l'opció *–init-file* s'executa al inici del servidor, canviant la contrasenya de *root*. Quan el servidor hagi arrencat correctament, cal esborrar l'arxiu *C:\mysql-init.txt*.

   Si s'ha instal·lat el servidor *MySQL* 4.1 o superior mitjançant l'instal·lador de *MySQL*, potser serà necessari especificar l'opció *–defaults-file*:

   C:\> C:\Archivos de Programa\MySQL\MySQL Server 5.0\bin\mysqld-nt.exe --defaults-file="C:\Archivos de Programa\MySQL\MySQL Server 5.0\my.ini" --init-file=C:\mysql-init.txt

   La configuració apropiada de *–defaults-file* es pot trobar usant el Gestor de Serveis: Menú Inici -> Panell de Control -> Eines Administratives -> Serveis

   Cercar el servei a la llista i prémer el boto dret del ratolí, escollint l'opció *Propietats*. El camp *Ruta* a l'executable conté la configuració de *–default-file*.

6. Parar els servidor *MySQL*, i reiniciar-lo en mode normal de nou. Si el servidor s'executa com a servei, cal iniciar-lo des de la finestra de serveis de *Windows*. Si s'executa manualment, cal usar la comanda habitual.

Si els passos anteriors no funcionen provar la següent seqüència:

1. Aturar completament el servidor *MySQL*. Aturant el servei o el procés com s'ha comentat en el passos anteriors.

2. Obrir la consola de comandes *DOS* i navegar fins el directori *\bin* del servidor *MySQL*.

3. Executar la comanda a la línia de comandes: *mysqld.exe –u root –skip-grant-tables*. No tancar aquesta finestra deixant-la en l'estat actual.

4. Obrir una nova consola de comandes DOS i navegar al directori *\bin* del servidor *MySQL*. Escriure la comanda *mysql* i prémer la tecla retorn de carro.

[5] *MicroSoft Disk Operating System*. El sistema operatiu de disc de *Microsoft* és un sistema operatiu per ordinadors basats en *x86*. Va ser el més popular de la família de sistemes operatius *DOS* de *Microsoft*, i el principal sistema per ordinadors personals compatible amb *IBM PC* durant la dècada del 1980 i mitjans de 1990, fins que gradualment la van substituir sistemes operatius que oferien interfície gràfica d'usuari, en particular diverses generacions de *Microsoft Windows*.

5. En aquests moments s'hauria d'haver arrencat la línia de comandes del client *MySQL*. Escriure la comanda *use mysql;* amb la que accedim a la base de dades anomenada *mysql*.

6. Executar la següent comanda: *UPDATE user SET Password = PASSWORD('NEW_PASSWORD') WHERE User = 'root';*

7. Escriure la comanda exit per sortir del client *MySQL*. Tancar la consola de comandes DOS. Tancar la consola de comandes DOS en la que s'està executant el servidor *MySQL*. Al tancar la finestra s'atura el servidor.

8. Reiniciar el servei de Windows del servidor *MySQL* o arrencar manualment el servidor amb la comanda habitual.

Si els passos anteriors tampoc han solucionat el problema desinstal·lar el servidor *MySQL* comprovant que s'han eliminat tots els directoris i fitxers de *MySQL* de l'ordinador i tornar a instal·lar-lo.

Per altra banda, *MySQL* ofereix un paquet d'eines gràfiques per treballar amb el servidor. El paquet conté:

1. *MySQL Administrator*: Administració del servidor *MySQL*, usuaris, catàleg, contingència, seguretat, etc.

2. *MySQL Browser*: Client gràfic de *MySQL* per executar sentències *SQL* contra el servidor.

3. *MySQL Migration Kit*: Permet migrar bases de dades d'altres *SGBD* a bases de dades *MySQL*.

4. *MySQL System Tray Monitor*: Monitorització del servidor *MySQL*.

11.3 Configuració del *SGBD*

A través de l'eina *MySQL Administrator* cal realitzar els següents passos per crear la base de dades anomenada *regates* i l'usuari regates que utilitza l'eina *Regaton* per accedir al servidor.

Passos per crear la base de dades *regates*:

1. Entrar a l'eina *MySQL Administrator* amb l'usuari *root* i la seva contrasenya.

2. Seleccionar l'opció *Catalogs*. Premer el botó dret en el marc *Schemata* on apareixen les bases de dades existents i seleccionar *Create New Schema*.

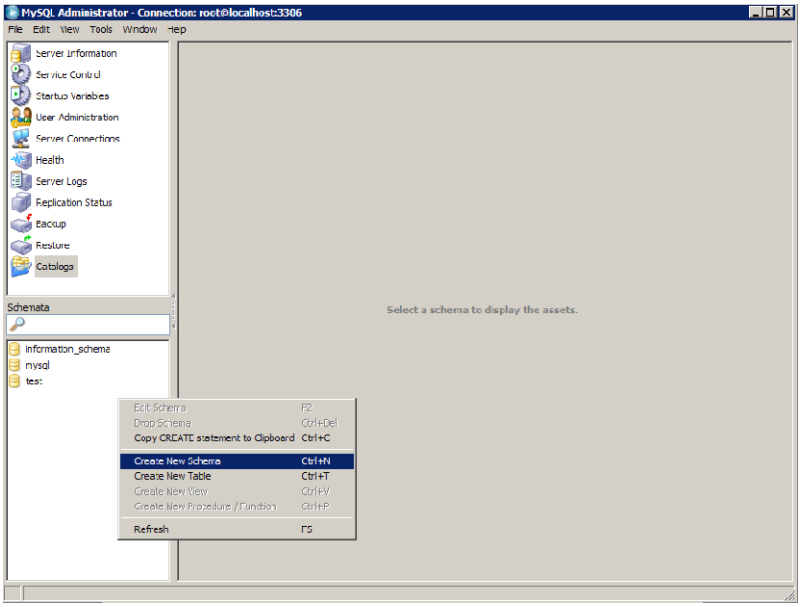


Figura 85: Creació d'una nova base de dades

3. Apareix la següent pantalla:

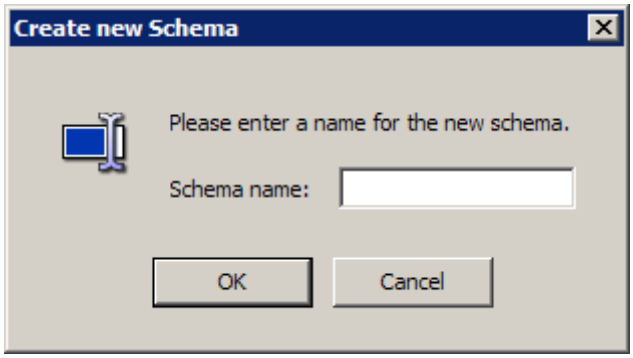


Figura 86: Introducció del nom de la base de dades

4. Introduir el nom de la nova base de dades *regates*.

5. Automàticament apareix la nova base de dades a la llista del marc on apareixen les bases de dades existents al servidor.

Passos per crear l'usuari regates que utilitza l'eina *Regaton* per connectar amb el servidor *MySQL*.

1. Si encara som a l'eina *MySQL Administrator* continuem amb el següent pas, sinó entrem a l'eina usant l'usuari *root* i la seva contrasenya.

2. Seleccionar l'opció *User Administration*. Prémer el botó dret en el marc *User Accounts* on apareixen els usuaris existents al sevidor *MySQL* i seleccionar l'opció *Add New User*.

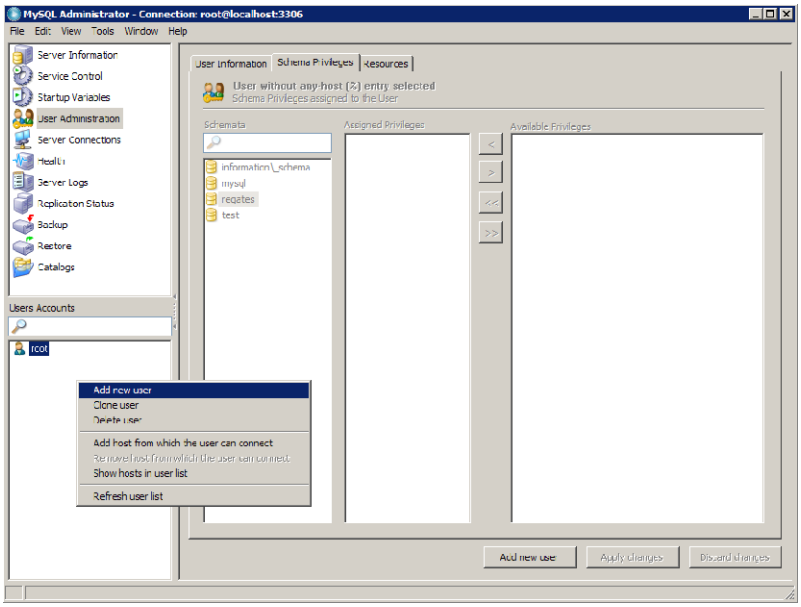


Figura 87: Creació d'un nou usuari de treball

3. L'usuari apareix automàticament a la llista del marc *User Accounts*. Seleccionar el nou usuari i informar els següents camps de la pestanya *User Information*:

    a. *MySQL User* amb el nom *regates*.

    b. *Password* amb contrasenya *regates*.

    c. *Confirm password* amb contrasenya *regates*.

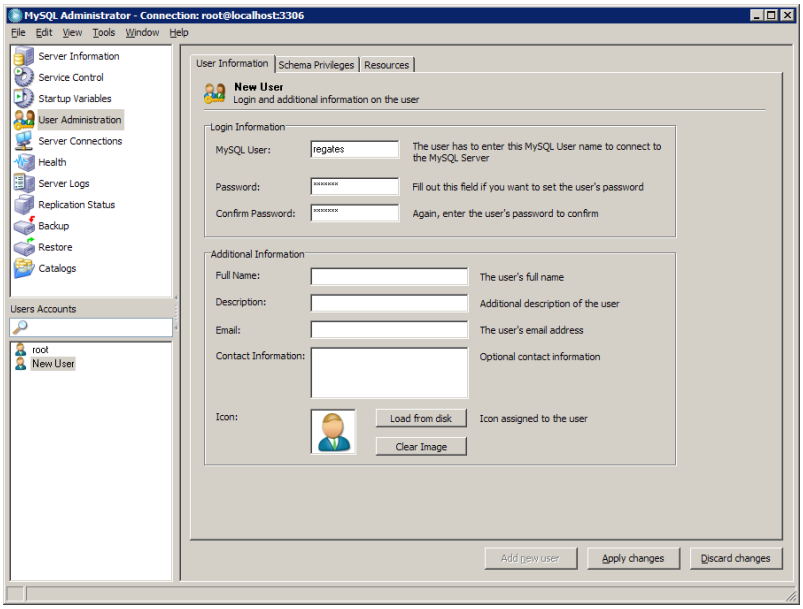


Figura 88: Introducció del nom i contrasenya del nou usuari

4. Prémer el botó *Apply Changes* per aplicar els canvis al servidor.

5. Seleccionar la pestanya *Schema privileges.* Donar a l'usuari regates els següents privilegis sobre la base de dades *regates*:

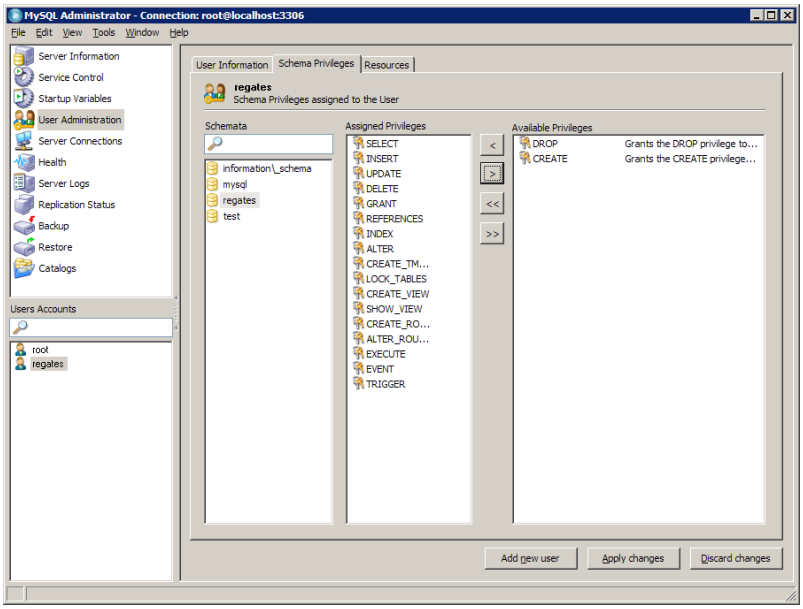


Figura 89: Alta de privilegis del nou usuari

6. Prémer el botó *Apply Changes* per aplicar els canvis al servidor.

Una vegada realitzats tots els passos anteriors el *SGBD MySQL* està preparat per admetre connexions a través de l'usuari regates sobre la base de dades regates.

Serà l'eina *Regaton* la que utilitzarà l'usuari *regates* per accedir a la base de dades *regates*. És la mateixa eina la que gestiona la creació, esborrat i actualització de les taules *regates*, *importacions* i *veles*.

11.4Instal·lació de l'eina *Regaton*

El sistema *Regaton* s'entrega en un fitxer comprimit anomenat *Regaton.rar*. El procés d'instal·lació del sistema és tan simple com descomprimir el fitxer on l'usuari ho desitgi.

Una vegada descomprimit es pot observar els següents arbres de directoris:

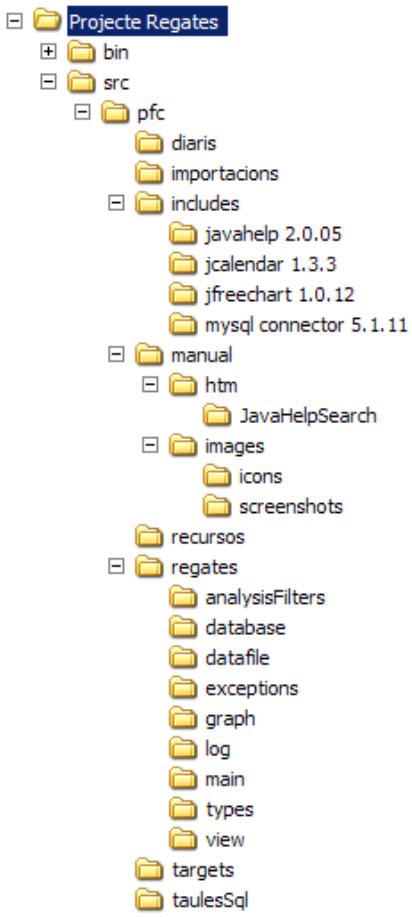


Figura 90: Arbre de directoris del projecte

Descripció dels directoris de *src*, o *source*:

- src.pfc.diaris
  Conté els diaris del sistema.

- src.pfc.importacions
  Directori que el sistema utilitza per defecte per cercar els fitxers de *Log* de les regates.

- src.pfc.includes
  Conté totes les llibreries *JAR*[6] que necessita el sistema.

- src.pfc.manual.htm
  Fitxers **.HTM* amb el contingut del manual d'usuari en format *HTML*[7].

- src.pfc.manual.JavaHelpSearch
  Directori que conté l'índex de cerques del manual d'usuari.

[6] *Java ARchive*. És un tipus d'arxiu que permet executar aplicacions escrites en *Java*. Està basat en el popular format *ZIP* i s'utilitza per agregar diversos fitxer en un. Un fitxer *JAR* és essencialment un fitxer *ZIP* que conté un directori addicional anomenat *META-INF*.

[7] *HyperText Markup Language*, en català Llenguatge de marcat *d'hypertext*. Llenguatge de marcat predominant en pàgines web. S'utilitza per descriure l'estructura i el contingut en format de text, així com per complementar el text amb altres objectes com imatges.

- src.pfc.images.icons
  Icones usades pel manual d'usuari.

- src.pfc.images.screenshots
  Captures de pantalla de l'execució dels processos del sistema.

- src.pfc.recursos
  Conjunt d'imatges i icones que usa l'eina.

- src.pfc.regates
  Directori principal amb els fonts del sistema.

- src.pfc.regates.analysisFilters
  Fonts dels *filtres* del sistema.

- src.pfc.regates.database
  Fonts de les classes de la capa de gestió de dades.

- src.pfc.regates.datafile
  Fonts de les classes de tractament de fitxers.

- src.pfc.regates.exceptions
  Fonts de les excepcions pròpies del sistema.

- src.pfc.regates.graph
  Fonts de les classes modificades de la llibreria gràfica per ampliar les seves funcionalitats.

- src.pfc.regates.log
  Fonts de les classes encarregades de la gestió dels diaris del sistema.

- src.pfc.regates.main
  Fonts de les classes principals.

- src.pfc.regates.types
  Fonts de les classes amb els tipus de dades propis del sistema.

- src.pfc.regates.view
  Fonts de les vistes.

- src.pfc.targets
  Directori que l'eina usa per defecte per llegir els fitxers de *targets*.

- src.pfc.taulesSql
  Directori amb els fitxers de definició de la taula regates.

El directori *bin*, o *binaries*, té la mateixa estructura que el directori *source*. Tanmateix aquest directori conté totes les classes executables del sistema i és el que utilitza l'executable del sistema anomenat *Regaton.exe*.

L'executable *Regaton.exe* està ubicat en el directori principal del projecte. La seva execució arrenca el procés *Batch*[8] anomenat *Regaton.bat* amb el següent contingut:

```
Set CLASSPATH=./bin/;./src/pfc/includes/jcalendar 1.3.3/jcalendar-
1.3.3.jar;./src/pfc/includes/jcalendar 1.3.3/looks-
2.0.1.jar;./src/pfc/includes/jfreechart
1.0.12/gnujaxp.jar;./src/pfc/includes/jfreechart 1.0.12/jcommon-
1.0.15.jar;./src/pfc/includes/jfreechart 1.0.12/jfreechart-
1.0.12.jar;./src/pfc/includes/jfreechart
1.0.12/junit.jar;./src/pfc/includes/jfreechart
1.0.12/servlet.jar;./src/pfc/includes/mysql connector 5.1.11/mysql-connector-
java-5.1.11-bin.jar;./src/pfc/includes/javahelp
2.0.05/jh.jar;./src/pfc/includes/javahelp
2.0.05/jhall.jar;./src/pfc/includes/javahelp
2.0.05/jhbasic.jar;./src/pfc/includes/javahelp
2.0.05/jsearch.jar;./src/pfc/includes/javahelp
2.0.05/src.jar;./src/pfc/includes/javahelp
2.0.05/jhindexer.jar;./src/pfc/includes/javahelp 2.0.05/jhsearch.jar;

java -Xms5m -Xmx512m -Dsun.java2d.noddraw=true -Duser.language=ES -
Duser.region=CA pfc.regates.main.Main
```

Primer es defineix la variable d'entorn *CLASSPATH* afegint el directori bin que conté totes les classes pròpies del sistema i la resta de llibreries *JAR* amb les classes addicionals.

En segon lloc, s'executa l'aplicació *Regaton* mitjançant la *JVM* amb els següents paràmetres:

- *–Xms5m*: indica que l'espai mínim de la memòria del *Heap* de la *JVM* ha de ser 5MB.

- *–Xmx512M*: indica que l'espai màxim de memòria del *Heap* de *JVM* no pot excedir els 512MB.

- *-Dsun.java2d.noddraw=true*[9]: s'utilitza per evitar certs errors del *pipeline* del pintat de *Java* durant la generació de gràfiques.

- *-Duser.language=ES -Duser.region=CA*: indica que el llenguatge usat és l'espanyol amb la variant del català.

[8] Arxiu de processament per lots. Són fitxer de text sense format, guardats amb l'extensió *.bat* que contenen un conjunt de comandes *MS-DOS*. Quan s'executa aquest arxiu els comandaments s'executen en grup, de forma seqüencial, permetent automatitzar diverses tasques.

[9] El web http://java.sun.com/products/java-media/2D/perf_graphics.html documenta específicament els errors que es produeixen en el *pipeline* de pintat de *Java*.

11.5Configuració de l'eina *Regaton*

El sistema *Regaton* adjuntat en el CDROM i que està comprimit en fitxer *Regaton*.rar no necessita cap pas de configuració addicional a part dels passos d'instal·lació i configuració de *MySQL*.

Finalitzada la instal·lació del *SGBD* i la seva configuració basta executar *Regaton.exe* per arrencar el sistema.

Per altra banda, si el que es desitja és continuar amb el desenvolupament de l'eina, es possible que al crear el nou projecte dins de l'eina *IDE*, com per exemple *Eclipse*, sigui necessari configurar el *Build Path* del projecte. Aspectes a configurar:

1. Afegir al projecte les llibreries JAR externes. Estan incloses en el directori *pfc.regates.inclues*.

2. Definir quina *JVM*, de les existents al sistema operatiu, ha d'utilitzar l'eina *IDE*.

3. Afegir els paràmetres descrits en la pàgina anterior a la *JVM* per agilitzar l'execució de l'eina.

12 Apèndix B: Manual d'usuari

Com tot sistema del mercat, *Regaton* posseeix un manual d'usuari per a comprendre l'ús de les seves funcionalitats. El manual està disponible en la mateixa eina, però per a comoditat del lector adjunto el seu contingut en aquest capítol.

El manual d'usuari es divideix en capítols. Cada capítol conté una o més seccions en funció del nombre de temes que tracta.

Totes les dades dels exemples són fictícies i no pertanyen a l'anàlisi real de cap regata. Tanmateix, els exemples i figures usats en el manual mostren el funcionament real de l'eina i el seu aspecte en la versió actual.

12.1 Fitxers de *Log* per importació de regates

Els fitxers de *Log* són fitxers de text pla que contenen la informació recol·lectada d'una o més regates realitzades per un vaixell.

Aquest fitxer han de complir els següents requisits per tal que la versió actual del sistema els tracti correctament:

1) El primer registre del fitxer ha de ser la capçalera amb el nom de cadascuna de les columnes que conté.

2) La capçalera pot estar repetida en qualsevol altra registre. El sistema obvia qualsevol altre registre de capçalera dels fitxers que no sigui el primer.

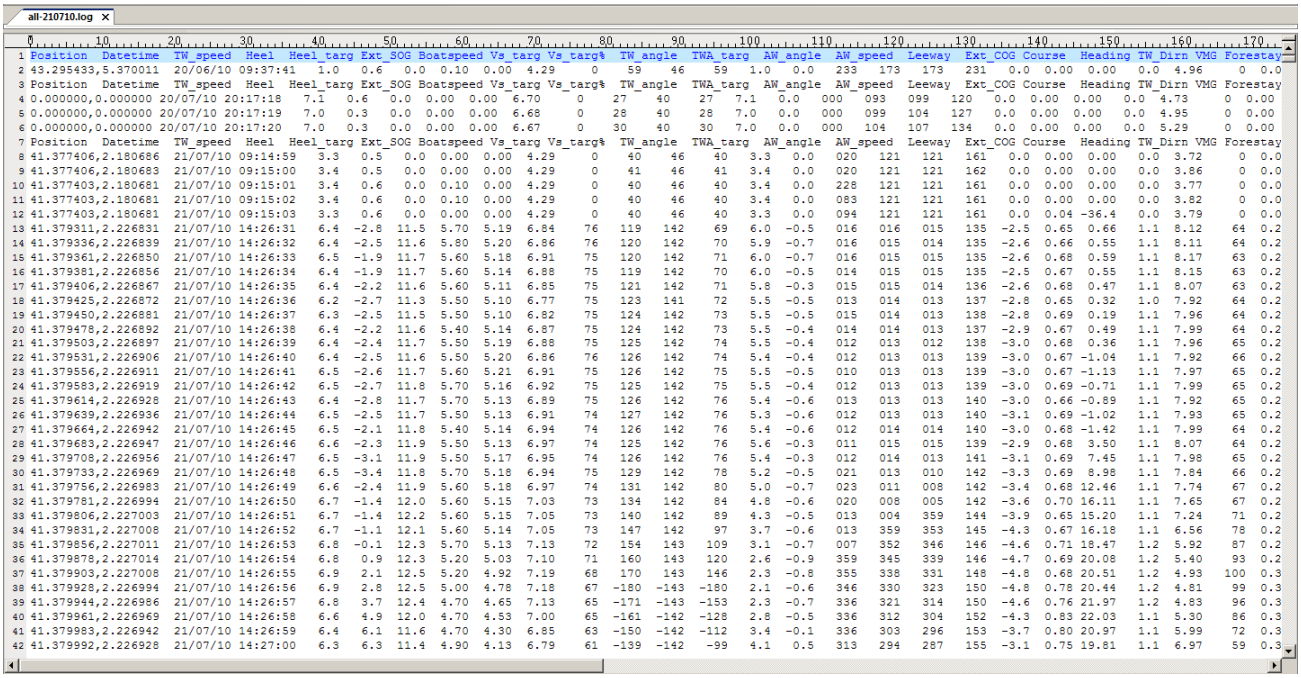
3) El separador entre les columnes és el codi *ASCII*[1] tabulador.

4) El camp dels registres amb l'instant d'enregistrament dels valors de les variables d'estudi, és a dir el camp *Datetime*, és el seu identificador. Això significa que un fitxer de Log no hauria de contenir registres amb instants d'enregistrament repetits, perquè el sistema ho considera un error i ho comunica a l'usuari, avortant la importació.

5) L'eina considera que un fitxer de *Log* posseeix l'extensió *.log*, així que en els quadres de diàleg per seleccionar els fitxers mostra per defecte només aquests tipus de fitxers. Si l'usuari ho desitja li pot indicar al sistema que mostri tots els fitxer del directori.

Un exemple parcial de fitxer de *Log* és el següent:

[1] De l'anglès *American Standard Code for Information Interchange*. És un joc de caràcters que assigna valors numèrics (del 0 al 127, 7 *bits* de longitud) a les lletres, xifres i signes de puntuació. Existeixen codis *ASCII* extensos de 256 caràcters (del 0 al 255, un *byte*), que permeten representar caràcters no anglesos com poden ser accents o la *ç*. Els caràcters de la part superior (127 a 255) varien d'un estàndard a un altre, encara que el més usat és el *ISO Latin-1* o *ISO-8859-1*.



all-210710.log

	Position	Datetime	TW_speed	Heel	Heel_targ	Ext_SOG	Boatspeed	Vs_targ	Vs_targ%	TW_angle	TWA_targ	AW_angle	AW_speed	Leeway	Ext_COG	Course	Heading	TW_Dirn	VMG	Forestay
2	43.295433,5.370011	20/06/10 09:37:41	1.0	0.6	0.0	0.10	0.00	4.29	0	59	46	59	1.0	0.0	233	173	173	231	0.0	0.00	0.00	0.0	4.96	0	0.0
3	Position	Datetime	TW_speed	Heel	Heel_targ	Ext_SOG	Boatspeed	Vs_targ	Vs_targ%	TW_angle	TWA_targ	AW_angle	AW_speed	Leeway	Ext_COG	Course	Heading	TW_Dirn	VMG	Forestay
4	0.000000,0.000000	20/07/10 20:17:18	7.1	0.6	0.0	0.00	0.00	6.70	0	27	40	27	7.1	0.0	000	093	099	120	0.0	0.00	0.00	0.0	4.73	0	0.00
5	0.000000,0.000000	20/07/10 20:17:19	7.0	0.3	0.0	0.00	0.00	6.68	0	28	40	28	7.0	0.0	000	099	104	127	0.0	0.00	0.00	0.0	4.95	0	0.00
6	0.000000,0.000000	20/07/10 20:17:20	7.0	0.3	0.0	0.00	0.00	6.67	0	30	40	30	7.0	0.0	000	104	107	134	0.0	0.00	0.00	0.0	5.29	0	0.00
7	Position	Datetime	TW_speed	Heel	Heel_targ	Ext_SOG	Boatspeed	Vs_targ	Vs_targ%	TW_angle	TWA_targ	AW_angle	AW_speed	Leeway	Ext_COG	Course	Heading	TW_Dirn	VMG	Forestay
8	41.377406,2.180686	21/07/10 09:14:59	3.3	0.5	0.0	0.00	0.00	4.29	0	40	46	40	3.3	0.0	020	121	121	161	0.0	0.00	0.00	0.0	3.72	0	0.0
9	41.377406,2.180683	21/07/10 09:15:00	3.4	0.5	0.0	0.00	0.00	4.29	0	41	46	41	3.4	0.0	020	121	121	162	0.0	0.00	0.00	0.0	3.86	0	0.0
10	41.377403,2.180681	21/07/10 09:15:01	3.4	0.6	0.0	0.10	0.00	4.29	0	40	46	40	3.4	0.0	228	121	121	161	0.0	0.00	0.00	0.0	3.77	0	0.0
11	41.377403,2.180681	21/07/10 09:15:02	3.4	0.6	0.0	0.10	0.00	4.29	0	40	46	40	3.4	0.0	083	121	121	161	0.0	0.00	0.00	0.0	3.82	0	0.0
12	41.377403,2.180681	21/07/10 09:15:03	3.3	0.6	0.0	0.00	0.00	4.29	0	40	46	40	3.3	0.0	094	121	121	161	0.0	0.04	-36.4	0.0	3.79	0	0.0
13	41.379311,2.226831	21/07/10 14:26:31	6.4	-2.8	11.5	5.70	5.19	6.84	76	119	142	69	6.0	-0.5	016	016	015	135	-2.5	0.65	0.66	1.1	8.12	64	0.2
14	41.379336,2.226839	21/07/10 14:26:32	6.4	-2.5	11.6	5.80	5.20	6.86	76	120	142	70	5.9	-0.7	016	015	014	135	-2.6	0.66	0.55	1.1	8.11	64	0.2
15	41.379361,2.226850	21/07/10 14:26:33	6.5	-1.9	11.7	5.60	5.18	6.91	75	120	142	71	6.0	-0.7	016	015	015	135	-2.6	0.68	0.59	1.1	8.17	63	0.2
16	41.379381,2.226856	21/07/10 14:26:34	6.4	-1.9	11.7	5.60	5.14	6.88	75	119	142	70	6.0	-0.5	014	015	015	135	-2.5	0.67	0.55	1.1	8.15	63	0.2
17	41.379406,2.226867	21/07/10 14:26:35	6.4	-2.2	11.6	5.60	5.11	6.85	75	121	142	71	5.8	-0.3	015	015	014	136	-2.6	0.68	0.47	1.1	8.07	63	0.2
18	41.379425,2.226872	21/07/10 14:26:36	6.2	-2.7	11.3	5.50	5.10	6.77	75	123	141	72	5.5	-0.5	013	014	013	137	-2.8	0.65	0.32	1.0	7.92	64	0.2
19	41.379450,2.226881	21/07/10 14:26:37	6.3	-2.5	11.5	5.50	5.10	6.82	75	124	142	73	5.5	-0.5	015	014	013	138	-2.8	0.69	0.19	1.1	7.96	64	0.2
20	41.379478,2.226892	21/07/10 14:26:38	6.4	-2.2	11.6	5.40	5.14	6.87	75	124	142	73	5.5	-0.4	014	014	013	137	-2.9	0.67	0.49	1.1	7.99	64	0.2
21	41.379503,2.226897	21/07/10 14:26:39	6.4	-2.4	11.7	5.50	5.19	6.88	75	125	142	74	5.5	-0.4	012	013	012	138	-3.0	0.68	0.36	1.1	7.96	65	0.2
22	41.379531,2.226906	21/07/10 14:26:40	6.4	-2.5	11.6	5.50	5.20	6.86	76	126	142	74	5.4	-0.4	012	013	013	139	-3.0	0.67	-1.04	1.1	7.92	66	0.2
23	41.379556,2.226911	21/07/10 14:26:41	6.5	-2.6	11.7	5.60	5.21	6.91	75	126	142	75	5.5	-0.5	010	013	013	139	-3.0	0.67	-1.13	1.1	7.97	65	0.2
24	41.379583,2.226919	21/07/10 14:26:42	6.5	-2.7	11.8	5.70	5.16	6.92	75	125	142	75	5.5	-0.4	012	013	013	139	-3.0	0.69	-0.71	1.1	7.99	65	0.2
25	41.379614,2.226928	21/07/10 14:26:43	6.4	-2.8	11.7	5.70	5.13	6.89	75	126	142	76	5.4	-0.6	013	013	013	140	-3.0	0.66	-0.89	1.1	7.92	65	0.2
26	41.379639,2.226936	21/07/10 14:26:44	6.5	-2.5	11.7	5.50	5.13	6.91	74	127	142	76	5.3	-0.6	012	013	013	140	-3.1	0.69	-1.02	1.1	7.93	65	0.2
27	41.379664,2.226942	21/07/10 14:26:45	6.5	-2.1	11.8	5.40	5.14	6.94	74	126	142	76	5.4	-0.6	012	014	014	140	-3.0	0.68	-1.42	1.1	7.99	64	0.2
28	41.379683,2.226947	21/07/10 14:26:46	6.6	-2.3	11.9	5.50	5.13	6.97	74	125	142	76	5.6	-0.3	011	015	015	139	-2.9	0.68	3.50	1.1	8.07	64	0.2
29	41.379708,2.226956	21/07/10 14:26:47	6.5	-3.1	11.9	5.50	5.17	6.95	74	126	142	76	5.4	-0.3	012	014	013	141	-3.1	0.69	7.45	1.1	7.98	65	0.2
30	41.379733,2.226969	21/07/10 14:26:48	6.5	-3.4	11.8	5.70	5.18	6.94	75	129	142	78	5.2	-0.5	021	013	010	142	-3.3	0.69	8.98	1.1	7.84	66	0.2
31	41.379756,2.226983	21/07/10 14:26:49	6.6	-2.4	11.9	5.60	5.18	6.97	74	131	142	80	5.0	-0.7	023	011	008	142	-3.4	0.68	12.46	1.1	7.74	67	0.2
32	41.379781,2.226994	21/07/10 14:26:50	6.7	-1.4	12.0	5.60	5.15	7.03	73	134	142	84	4.8	-0.6	020	008	005	142	-3.6	0.70	16.11	1.1	7.65	67	0.2
33	41.379806,2.227003	21/07/10 14:26:51	6.7	-1.4	12.2	5.60	5.15	7.05	73	140	142	89	4.3	-0.5	013	004	359	144	-3.9	0.65	15.20	1.1	7.24	71	0.2
34	41.379831,2.227008	21/07/10 14:26:52	6.7	-1.1	12.1	5.60	5.14	7.05	73	147	142	97	3.7	-0.6	013	359	353	145	-4.3	0.67	16.18	1.1	6.56	78	0.2
35	41.379856,2.227011	21/07/10 14:26:53	6.8	-0.1	12.3	5.70	5.13	7.13	72	154	143	109	3.1	-0.7	007	352	346	146	-4.6	0.71	18.47	1.2	5.92	87	0.2
36	41.379878,2.227014	21/07/10 14:26:54	6.8	0.9	12.3	5.20	5.03	7.10	71	160	143	120	2.6	-0.9	359	345	339	146	-4.7	0.69	20.08	1.2	5.40	93	0.2
37	41.379903,2.227008	21/07/10 14:26:55	6.9	2.1	12.5	5.20	4.92	7.19	68	170	143	146	2.3	-0.8	355	338	331	148	-4.8	0.68	20.51	1.2	4.93	100	0.3
38	41.379928,2.226994	21/07/10 14:26:56	6.9	2.8	12.5	5.00	4.78	7.18	67	-180	-143	-180	2.1	-0.6	346	330	323	150	-4.8	0.78	20.44	1.2	4.81	99	0.3
39	41.379944,2.226986	21/07/10 14:26:57	6.8	3.7	12.4	4.70	4.65	7.13	65	-171	-143	-153	2.3	-0.7	336	321	314	150	-4.6	0.76	21.97	1.2	4.83	96	0.3
40	41.379961,2.226969	21/07/10 14:26:58	6.6	4.9	12.0	4.70	4.53	7.00	65	-161	-142	-128	2.8	-0.5	336	312	304	152	-4.3	0.83	22.03	1.1	5.30	86	0.3
41	41.379983,2.226942	21/07/10 14:26:59	6.4	6.1	11.6	4.70	4.30	6.85	63	-150	-142	-112	3.4	-0.1	336	303	296	153	-3.7	0.80	20.97	1.1	5.99	72	0.3
42	41.379992,2.226928	21/07/10 14:27:00	6.3	6.3	11.4	4.90	4.13	6.79	61	-139	-142	-99	4.1	0.5	313	294	287	155	-3.1	0.75	19.81	1.1	6.97	59	0.3

Figura 91: Contingut parcial d'un fitxer de Log

12.2Fitxers de Targets

Els fitxers de *targets* són fitxers de text pla que contenen els valors objectius proporcionats pel dissenyador per a una variable d'estudi i un vaixell de regates. Cada registre dels fitxers conté dos valors, el primer correspon a la velocitat real del vent i el segon al valor objectiu del vaixell per aquesta velocitat. El separador dels camps és el codi *ASCII* tabulador.

El nom d'aquests fitxers ha de seguir el format que es mostra a continuació i així el sistema serà capaç d'identificar-los quan sigui necessari.

Format per a trams a contra vent:

NomVaixell_VariableEstudi_targets.txt

Per exemple alguns noms vàlids per aquests fitxers són:

El grito_TW_angle_targets.txt
Els quatre gats_Boatspeed_targets.txt
Fantasma de la ópera_Rudder_targets.txt
Les canons de Navarone_Heel_targets.txt
Platero y yo_VMG_targets.txt

Format per a trams a favor del vent:

NomVaixell_VariableEstudi_targets_down.txt

Per exemple alguns noms vàlids per aquests fitxers són:

El grito_TW_angle_targets_down.txt
Els quatre gats_Boatspeed_targets_down.txt
Fantasma de la ópera_Rudder_targets_down.txt
Les canons de Navarone_Heel_targets_down.txt
Platero y yo_VMG_targets_down.txt

La següent figura mostra un possible contingut d'un fitxer:

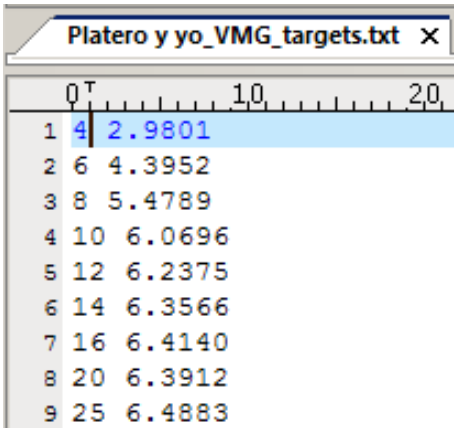


Figura 92: Fitxer de targets de la variable VMG

12.3 Fitxer bàsic de definició de la taula de regates

El sistema *Regaton* ofereix un fitxer de definició de la taula regates bàsic. Aquest és l'aspecte del fitxer en la versió presentada per aquest projecte:

```
#########################################################################
# DEFINICIÓ DE LA TAULA SQL PER A REGATES                               #
# -------------------------------------                                 #
#                                                                       #
# Estructura del fitxer:                                                #
#                                                                       #
#     Nom columna\tTipus columna\t\tParàmetre 1                         #
#                                                                       #
#     Paràmetre 1: PRI si es clau primària                              #
#                                                                       #
# Requisits:                                                            #
#   La taula regates ha de tenir com a clau primària Boat + Datetime.#
#   El separador entre camps és el codi ASCII tabulador!                #
#   El caràcter # indica que el registre conté un comentari.            #
#                                                                       #
#########################################################################
#Vaixell.
Boat  VarChar(50) PRI

#Regata.
Race  VarChar(50)

#Vela major.
```

```
Mainsail    VarChar(50)

#Tram.
Leg   VarChar(50)

#Vela de proa.
HeadSail    VarChar(50)

#Qualitat de les dades.
Quality     VarChar(50)

#Comentari.
Comment     VarChar(500)

#Posició latitud/altitud.
Position    VarChar(30)

#Data, hora, minut, segon.
Datetime    Datetime    PRI

#Velocitat del vent.
#TW_speed   Float(3,1)
TW_speed    Float

#Angle d'escorament.
#Angles mínim i màxim per defecte: -180,180
Heel  Angle

#Angle d'escorament objectiu.
#Angles mínim i màxim per defecte: -180,180
Heel_targ   Angle

#Angle d'escorament proporcionat pel dissenyador.
#Ext_SOG    Float(4,2)
Ext_SOG     Float

#Velocitat del vaixell.
#Boatspeed  Float(4,2)
Boatspeed   Float

#Velocitat objectiu del vaixell proporcionada pel dissenyador.
#Vs_targ    Float(4,2)
Vs_targ     Float

#Percentatge assolit del vaixell comparada amb la del dissenyador.
Vs_targ%    SmallInt

#Angle del vent.
#Angles mínim i màxim per defecte: -180,180
TW_angle    Angle

#Angle del vent objectiu proporcionat pel dissenyador.
#Angles mínim i màxim per defecte: -180,180
TWA_targ    Angle

#Angle aparent.
#Angles mínim i màxim per defecte: -180,180
AW_angle    Angle

#Velocitat del vent aparent.
#AW_speed   Float(3,1)
```

```
AW_speed    Float

#Deriva
#Leeway     Float(3,1)
Leeway      Float

#Rumb del vaixell calculat pel GPS.
Ext_COG     Angle(0,360)

#Rumb del vaixell tenint en compte la deriva.
Course      Angle(0,360)

#Rumb del vaixell segons el sensor de rumb (compàs).
Heading     Angle(0,360)

#Direcció o rumb del vent.
TW_Dirn     Angle(0,360)

#Càlcul vectorial que mesura el rendiment del vaixell en funció de la
#seva velocitat i angle.
#VMG  Float(3,1)
VMG   Float

#Pressió del sensor d'estai.
#Forestay   Float(3,2)
Forestay    Float

#Angle del timó respecte la proa del vaixell.
#Angles mínim i màxim per defecte: -180,180
Rudder      Angle

#Angle objectiu del timó del vaixell.
#Angles mínim i màxim per defecte: -180,180
Rudder_targ Angle

#Velocitat objectiu segons les polars del vaixell.
#Vs_Nav     Float(4,2)
Vs_Nav      Float

#Percentatge de velocitat segons les polars.
Vs_Nav%     SmallInt

#Intensitat del corrent marítim.
#MCurRate   Float(3,2)
MCurRate    Float

#Rumb o direcció del corrent marítim.
MCur_Dir    Angle(0,360)
```

Figura 93: Exemple de fitxer de definició de la taula de regates

Partint d'aquest fitxer inicial el sistema es pot trobar amb tres situacions a l'hora de realitzar una importació des d'un fitxer de *Log*:

6) Importar dades des d'un fitxer que conté el mateix nombre de columnes que les definides actualment. El sistema no precisa accions addicionals per realitza la importació.

7) Importar dades des d'un fitxer que conté menys columnes que les definides actualment. El sistema insereix les columnes del fitxer deixant a nul la resta de columnes.
8) Importar dades des d'un fitxer que conté més columnes que les definides actualment. El sistema les detecta i ofereix a l'usuari la possibilitat d'incorporar-les a la taula SQL de *regates*. Aquest procés no actualitza el fitxer de definició de la taula *regates*.

En la versió actual recau en l'usuari la responsabilitat d'actualitzar la informació del fitxer de definició. Qualsevol modificació sobre el fitxer ha de respectar els requisits descrits en la capçalera del fitxer, en cas contrari el sistema podria no funcionar de la manera esperada.

12.4Manual d'usuari de la versió actual del sistema

A continuació es mostra el contingut del manual d'usuari del sistema *Regaton*. Aquest manual resideix en la mateixa implementació del sistema i es pot consultar sempre que l'usuari ho desitgi.

1 Introducció

El present manual té com objectiu orientar els usuaris del sistema *Regaton*.

El capítol **Aspectes generals** del manual descriu de manera general el sistema, abordant els aspectes globals més importants.

El capítol **Interfície del sistema** del manual descriu el funcionament de les funcionalitats que ofereix el sistema.

El capítol **Filtres del sistema** del manual descriu el funcionament dels filtres que ofereix el sistema per optimitzar l'anàlisi de la informació de les regates.

Per a facilitat de l'usuari s'il·lustra la informació amb figures i gràfiques que mostren les mateixes pantalles que desplega el sistema *Regaton*.

2 Aspectes generals

1. Abast.
2. Usuaris del sistema.
3. Visió global del sistema.

2.1 Abast

El sistema *Regaton* és una solució computacional per l'anàlisi de rendiment en regates d'alt nivell.

Mitjançant el sistema els usuaris poden importar noves regates realitzades, actualitzar la seva informació i analitzar les dades a partir de la generació de gràfiques d'una o diverses variables.

Relació de funcionalitats que ofereix la versió actual del sistema:

- Importació de fitxers de *Log* amb informació d'una o diverses regates.
- Preparació i tractament de les dades importades per analitzar-les posteriorment.
- Gravació persistent de les dades llegides dels fitxers de *Log* a una base de dades de regates.
- Inserció, consulta i esborrat de veles.
- Manteniment del temps d'ús de les veles.
- Consulta i esborrat de regates importades.
- Diari del sistema amb l'històric d'avisos i errors produïts i les seves descripcions.
- Gràfiques d'anàlisi
    - Gràfiques cronològiques.
    - Gràfiques d'anàlisi de variables en funció de la velocitat real del vent (*TWS*).
    - Gràfiques d'anàlisi de variables en funció d'una variable diferent a la velocitat real del vent.
    - Gràfiques babord / estribord, per a qualsevol dels tipus anteriors de gràfiques.
- Filtres
    - Filtre de qualitat de les dades.
    - Filtre per a les gràfiques d'anàlisi de variables en funció de la velocitat real del vent.
    - Filtre babord / estribord per les gràfiques babord / estribord.
- Creació i esborrat de la taula de regates.
- Creació i esborrat de la taula de veles.
- Fitxer bàsic de definició de la taula de regates de la base de dades.

2.2 Usuaris del sistema

Regaton és una eina específica que gestiona dades d'un àmbit molt concret. Els usuaris als que es dirigeix el sistema són persones amb bons coneixements nàutics i del món de les regates professionals.

2.3 Visió global del sistema

La següent figura mostra una visió global del sistema:

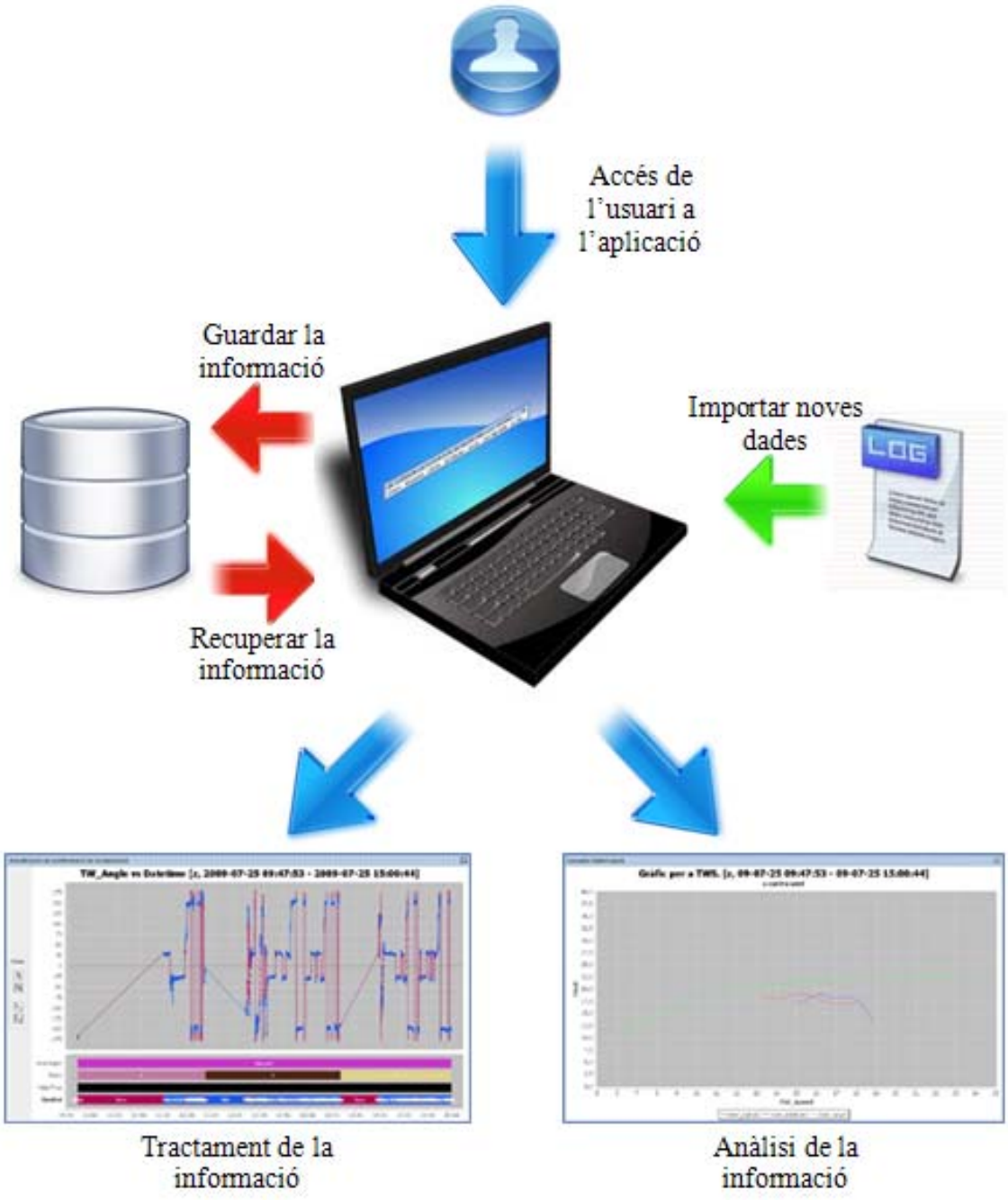


Figura 1: Esquema general del sistema.

Si observem la figura veiem dos traçats principals, un en sentit horitzontal i un altre en vertical. Ambdós traçats mostren les funcionalitats més importants del sistema.

El traçat horitzontal mostra la incorporació de noves dades al sistema, a partir d'un fitxer extern, i com en última instància la informació es fa persistent.

Per altra banda, el vertical mostra les possibilitats que el sistema ofereix a l'usuari per treballar amb les dades: actualització i anàlisi.

Interfície del sistema

1. Vista principal
2. Fitxer
3. Regates
    1. Importar regata
        1. Vista
        2. Procés
        3. Inserir noves columnes
    2. Visualitzar regates
    3. Esborrar regates
4. Veles
    1. Inserir veles
    2. Visualitzar veles
    3. Esborrar veles
5. Gràfiques
    1. TWAngle vs Time
        1. Selecció de la informació a actualitzar
        2. Vista
            1. Barra d'eines
            2. Gràfica TWA vs Time
            3. Gràfiques temporals de les variables d'actualització
        3. Actualització de la informació de la regata
    2. X vs Múltiples Y
        1. Vista
        2. Gràfiques cronològiques
        3. Gràfiques d'anàlisi en funció de la variable *TWS*
            1. Fitxers de *targets*
        4. Altres gràfiques d'anàlisi
        5. Gràfiques babord estribord
    3. Altres funcionalitats comunes
6. Logs
    1. Log del sistema
7. Configuració
    1. Taula de regates
        1. Crear taula de regates
            1. Fitxer de definició de la taula
        2. Esborrar taula de regates
    2. Taula de veles
        1. Crear taula de veles
        2. Esborrar taula de veles
8. Ajuda
    1. Manual d'usuari
    2. Sobre *Regaton*

3.1 Vista principal

La següent figura mostra la interfície principal del sistema:

Figura 2: Interfície principal del sistema.

La interfície principal consta d'una barra de menús amb els següents elements:

- *Fitxer*: Dóna l'opció de sortir del sistema.
- *Regates*: Conjunt de funcionalitats orientades al tractament de la informació recol·lectada durant les regates.
- *Veles*: Conjunt de funcionalitats orientades al tractament de la informació sobre les veles.
- *Gràfiques*: Generació de gràfiques d'anàlisi de la informació de les regates.
- *Logs*: Diaris del sistema. La versió actual només posseeix un diari encarregat d'emmagatzemar els avisos i errors produïts durant l'execució del sistema.
- *Configuració*: Gestió de les taules SQL de l'eina.
- *Ajuda*: Manual d'usuari i informació descriptiva de la versió del sistema.

3.2 Fitxer

La següent figura mostra les funcionalitats generals del sistema:

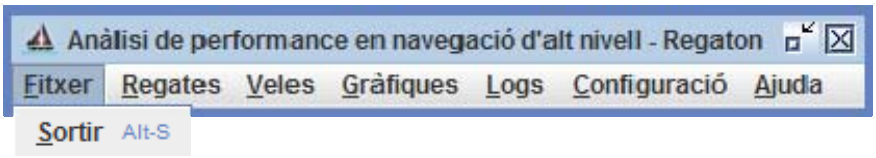


Figura 3: Sortir del sistema.

Funcionalitats del menú:

- *Sortir*: Surt del sistema. També es pot prémer la creu de la interfície principal per tancar el sistema.

3.3 Regates

Funcionalitats del sistema de gestió de regates:

Figura 4: Funcionalitats del sistema de gestió de regates.

Funcionalitats del menú:

- *Importar regata*: Incorporació de les dades d'una nova regata al sistema.
- *Visualitzar regates*: Consulta de les regates existents en el sistema.
- *Esborrar regates*: Eliminació de regates del sistema.

3.3.1 Importar regata

1. Vista
2. Procés
3. Inserir noves columnes

3.3.1.1 Descripció de la vista

La següent figura mostra la vista d'importació d'una regata:

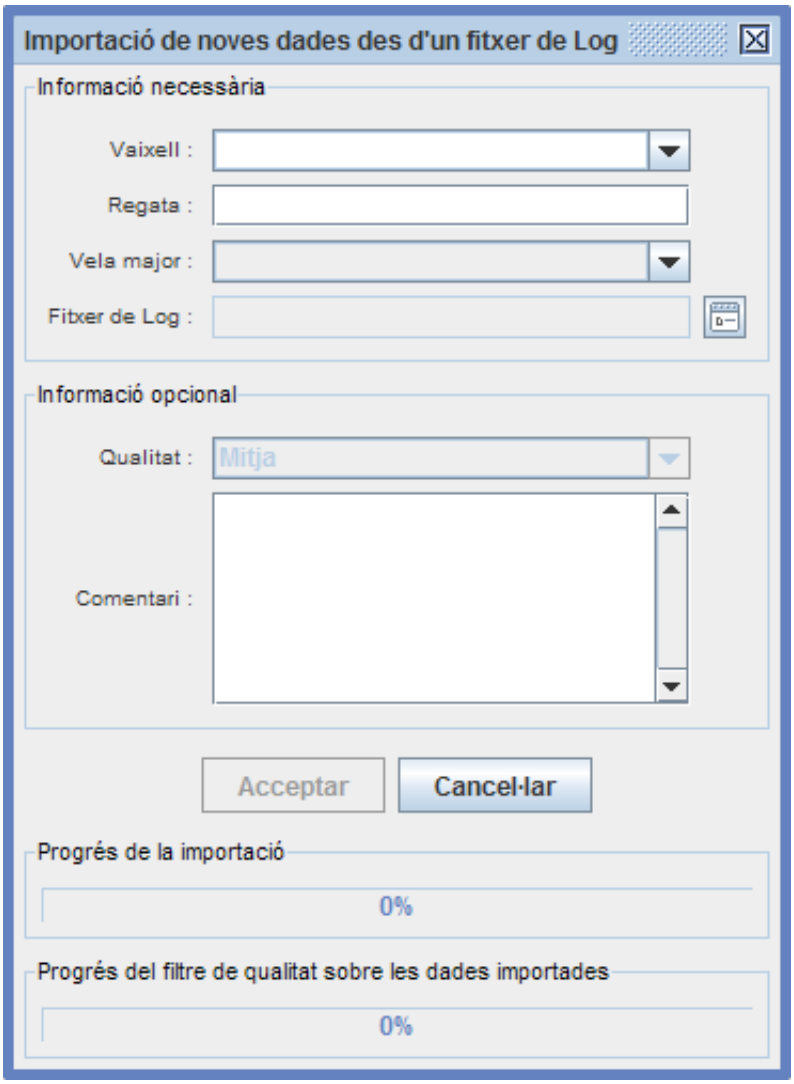


Figura 5: Vista d'importació d'una regata.

Descripció dels camps de la vista

- Camps que s'han d'informar obligatòriament:
  - *Vaixell*: Nom del vaixell sobre el que s'han recol·lectat les dades. És possible escriure el nom d'un vaixell nou o escollir-ne un entre els que existeixen al sistema.
  - *Regata*: Nom de la regata que es vol importar.
  - *Vela major*: Nom de la vela major usada durant la regata. S'ha d'escollir entre les veles majors donades d'alta al sistema.
  - *Fitxer de Log*: Nom del fitxer que conté la informació a importar. Prement el botó de la dreta del camp no editable on es mostra el nom del

fitxer s'obre una nova vista que permet recórrer l'arbre de directoris dels discs de l'ordinador.

- Camps opcionals:
  - *Qualitat*: Qualitat que s'assignarà a les dades importades. La qualitat indica com són de bones les dades que s'estan important. Existeixen tres tipus de qualitat: *Alta*, *Mitja* i *Baixa*. La qualitat *Alta* indica que les dades són rellevants per l'anàlisi de la regata. Les dades amb qualitat *Mitja* també són d'interès per l'anàlisi. En canvi, les dades amb qualitat *Baixa* acostumen a pertànyer a situacions com la realització de maniobres o la navegació a motor i es descarten a l'hora de realitzar els anàlisis.
  - *Comentari*: Qualsevol informació addicional.
- *Barra de progrés de la importació*: Mostra el percentatge de progrés de l'execució del procés d'importació de la informació de la regata.
- *Barra de progrés del filtre sobre la qualitat de les dades*: Mostra el percentatge de progrés de l'execució del filtre de qualitat sobre les dades.

3.3.1.2 Descripció del procés d'importació d'una regata

El procés d'importació d'una regata passa per les següents etapes:

1. Lectura de la capçalera del fitxer de *Log*.
2. Detecció de noves columnes en el fitxer de *Log*.
3. Inserció de les noves columnes al sistema si l'usuari ho desitja.
4. Detecció de columnes existents al sistema que no apareixen al fitxer de *Log*.
5. Inserció de la informació al sistema.
6. Execució del filtre de qualitat sobre les dades si s'escau.

La següent figura mostra un exemple del procés d'importació d'una regata:

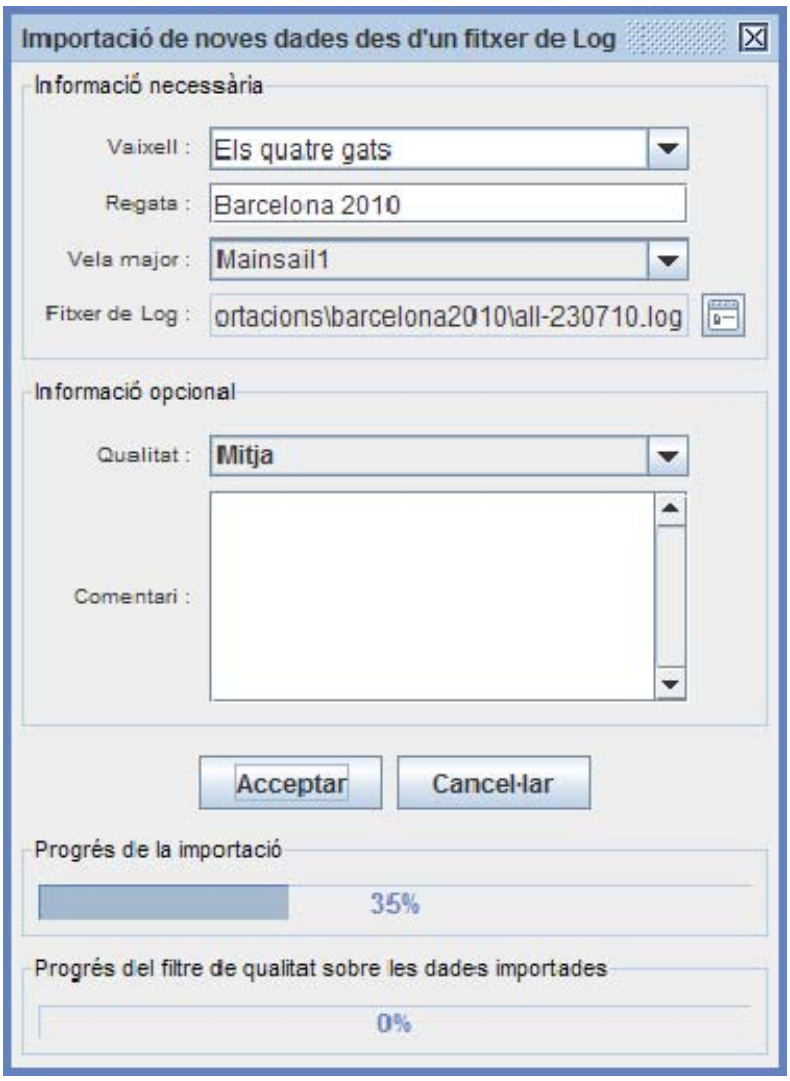


Figura 6: Exemple d'importació d'una regata.

3.3.1.3 Inserir noves columnes al sistema

Quan el procés d'importació detecta que el fitxer de *Log* conté noves columnes, que no existeixen al sistema, pregunta a l'usuari si desitja incorporar-les.

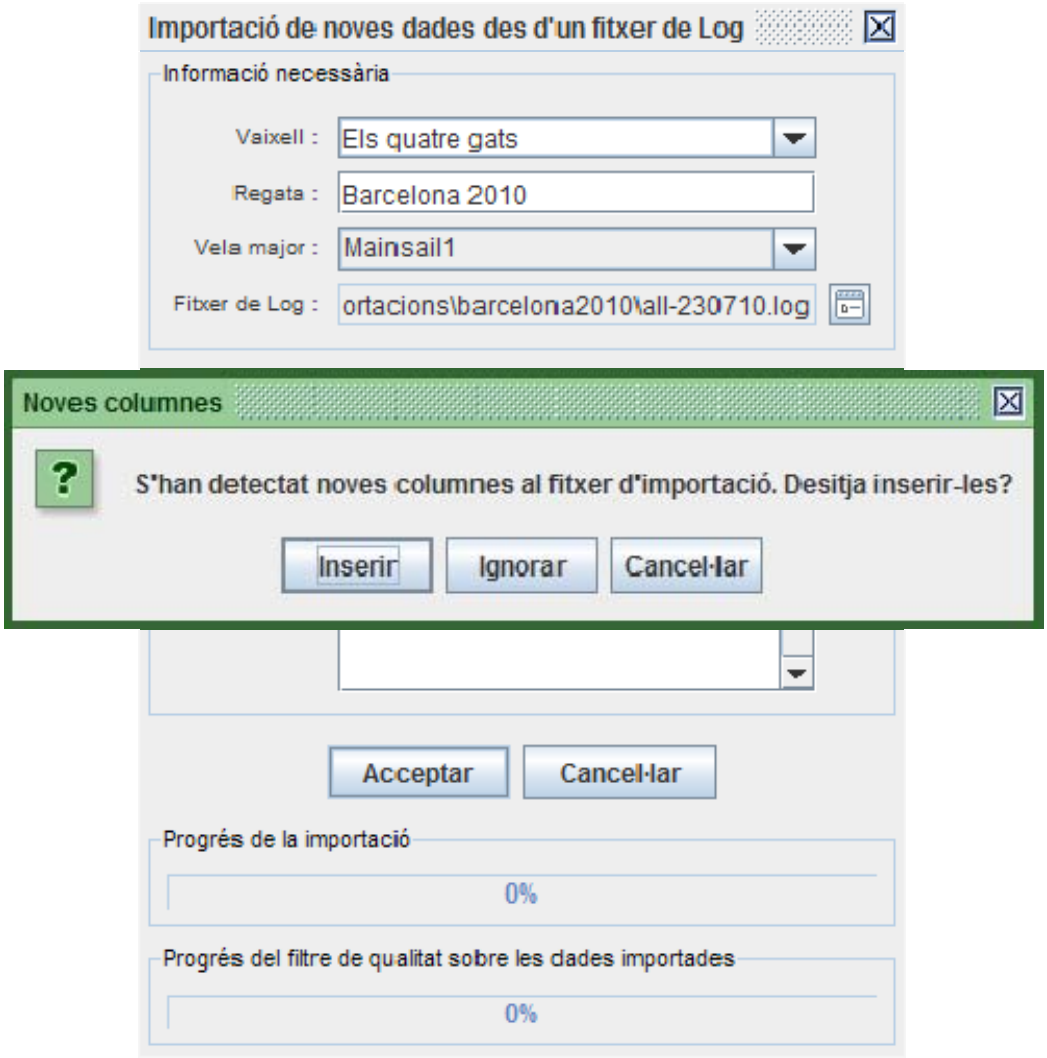


Figura 7: Quadre de diàleg preguntant a l'usuari quin acció realitzar.

Si l'usuari cancel·la la importació no es realitza cap acció i es finalitza la importació.

Si l'usuari ignora la inserció de les noves columnes el sistema importarà només les columnes que actualment existeixen al sistema, obviant les demés.

Si l'usuari decideix inserir les columnes el sistema mostra la vista següent:

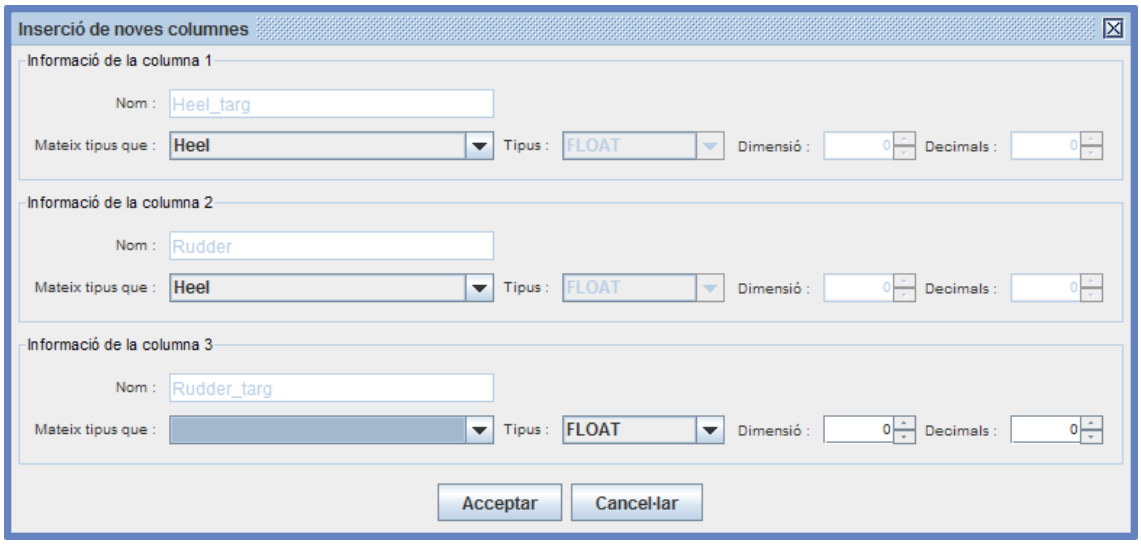


Figura 8: Exemple de vista d'inserció de noves columnes.

Descripció dels camps de la vista

Els camps que es descriuen a continuació són els mateixos per a cadascuna de les noves columnes detectades:

- *Nom*: Nom de la nova columna. El seu valor es pren automàticament de la capçalera del fitxer de *Log*.
- *Mateix tipus que*: Una de les dues maneres possibles d'introduir tots els valors de la nova columna és escollint el tipus de dades de la columna prenent com a referent una de les columnes del sistema. L'altra manera és omplir els valors a mà. Existeix la possibilitat d'escollir en primer lloc una columna entre les existents i després deixar el camp a blanc de nou; amb això el sistema manté el tipus de dades de l'última columna seleccionada i dóna a l'usuari el control sobre els camps per si desitja modificar-los.
- *Tipus*: Tipus de dades de la columna.
- *Dimensió/Mínim*: Mida del tipus de dades. Si el tipus de dades seleccionat és Angle, el camp serveix per indicar l'angle mínim que pot prendre.
- *Decimals/Màxim*: Nombre de decimals del tipus de dades. Només s'activa amb dos tipus numèrics. Si definim un coma flotant podem introduir el nombre de decimals que contindrà. Si el tipus de dades seleccionat és Angle, el camp serveix per indicar l'angle màxim que pot prendre.

IMPORTANT!: La definició d'un tipus de dades amb valor zero a la dimensió i al nombre de decimals equival a definir el tipus de dades genèric. El sistema generarà el tipus de dades obviant dimensions i decimals. És a dir, si a la vista definíssim un tipus de dades FLOAT amb dimensió 2 i decimals 2 el sistema crearia el tipus de dades FLOAT(2,2) admeten valors sota aquest format, però si definíssim un FLOAT amb dimensió i decimals amb valor zero el sistema crearia un FLOAT, que admet valors sense restriccions.

3.3.2 Descripció de la vista Visualitzar regates

La següent figura mostra la vista de visualització de les regates existents al sistema:

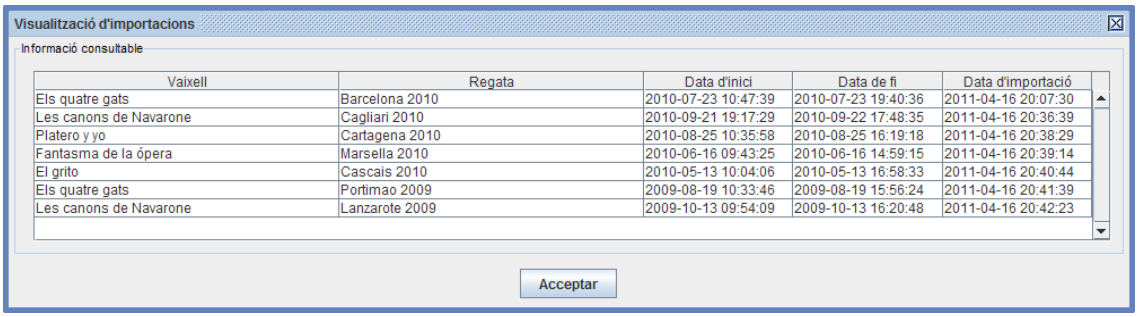


Figura 9: Vista de visualització de les regates existents.

Descripció dels camps de la vista

- *Vaixell*: Nom del vaixell sobre el que s'han recol·lectat les dades.
- *Regata*: Nom de la regata que s'està mostrant.
- *Data d'inici*: Data en la que s'inicia la regata.
- *Data de fi*: Data en la que es finalitza la regata.
- *Data d'importació*: Data en la que s'importa la regata al sistema.

3.3.3 Descripció de la vista Esborrar regates

La vista suporta la selecció múltiple de regates per esborrar-les alhora.

La següent figura mostra la vista d'esborrat de regates del sistema:

Figura 10: Vista d'esborrat de regates.

Descripció dels camps de la vista

- *Vaixell*: Nom del vaixell sobre el que s'han recol·lectat les dades.
- *Regata*: Nom de la regata que s'està mostrant.
- *Data d'inici*: Data en la que s'inicia la regata.
- *Data de fi*: Data en la que es finalitza la regata.

3.4 Veles

Funcionalitats del sistema de gestió de veles:

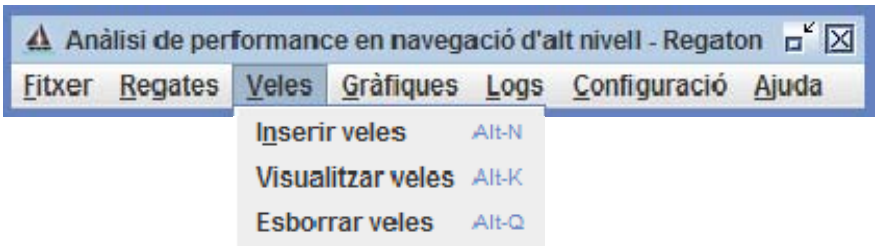


Figura 11: Funcionalitats del sistema de gestió de veles.

Funcionalitats del menú:

- *Inserir veles*: Incorporació de noves veles al sistema.
- *Visualitzar veles*: Consulta de les veles existents en el sistema.
- *Esborrar veles*: Eliminació de veles del sistema.

Les veles inserides en el sistema es poden usar tant en la *importació de regates* com en la *gràfica TWAngle vs Time* per actualitzar la informació de les mateixes regates.

Quan l'usuari esborra una vela del sistema aquesta s'elimina de la taula de veles. Tanmateix, si la vela s'ha utilitzat prèviament en alguna regata la informació associada es manté en el sistema. Si en algun moment es torna a inserir la vela, el sistema comprovarà si existeix informació associada en les regates existents i actualitzarà les seves estadístiques.

3.4.1 Descripció de la vista Inserir veles

L'usuari pot inserir veles mentre ho desitgi introduint el seu nom i el tipus. Cada vegada que premi el botó *Inserir* els camps es buiden permetent la inserció d'una nova vela.

La següent figura mostra la vista d'inserció de veles al sistema:

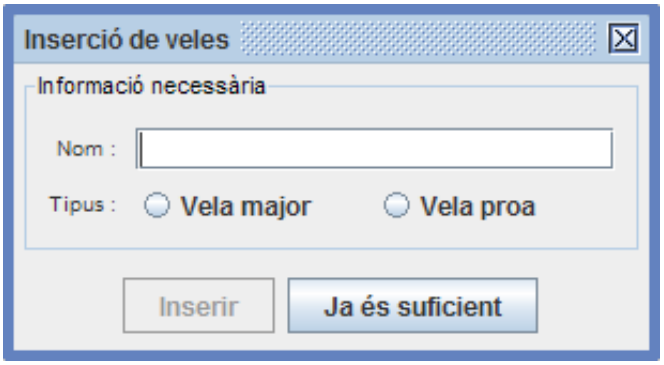


Figura 12: Vista d'inserció de veles.

Descripció dels camps de la vista

- *Nom*: Nom de la vela.
- *Tipus*: Tipus de la vela. El sistema només ofereix la possibilitat de donar d'alta veles majors o de proa.

3.4.2 Descripció de la vista Visualitzar veles

La següent figura mostra la vista de visualització de les veles existents al sistema:

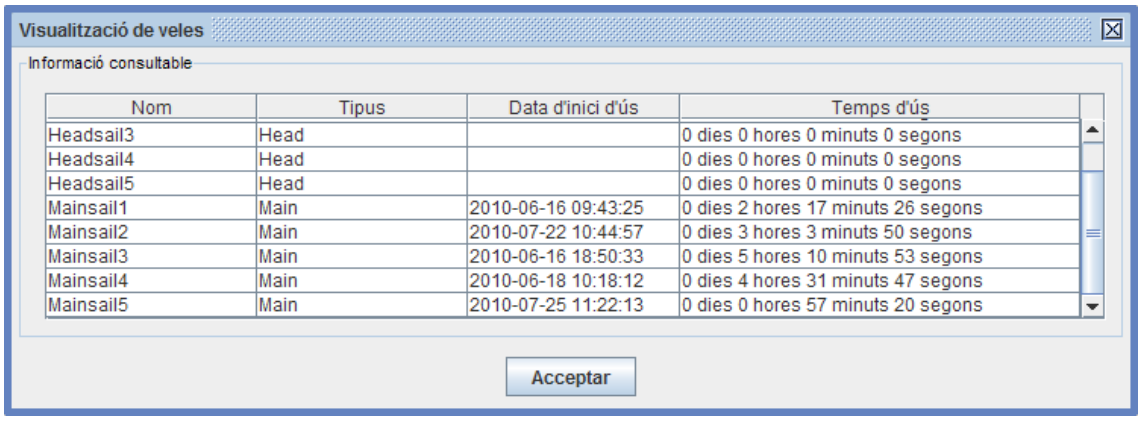


Figura 13: Vista de visualització de les veles existents.

Descripció dels camps de la vista

- *Nom*: Nom de la vela.
- *Tipus*: Tipus de la vela.
- *Data d'inici d'ús*: Data inicial de la regata del sistema en que s'utilitza la vela per primer cop.
- *Temps d'ús*: Temps d'us total de la vela tenint en compte totes les regates existents al sistema.

3.4.3 Descripció de la vista Esborrar veles

La vista suporta la selecció múltiple de veles per esborrar-les alhora.

La següent figura mostra la vista d'esborrat de veles del sistema:

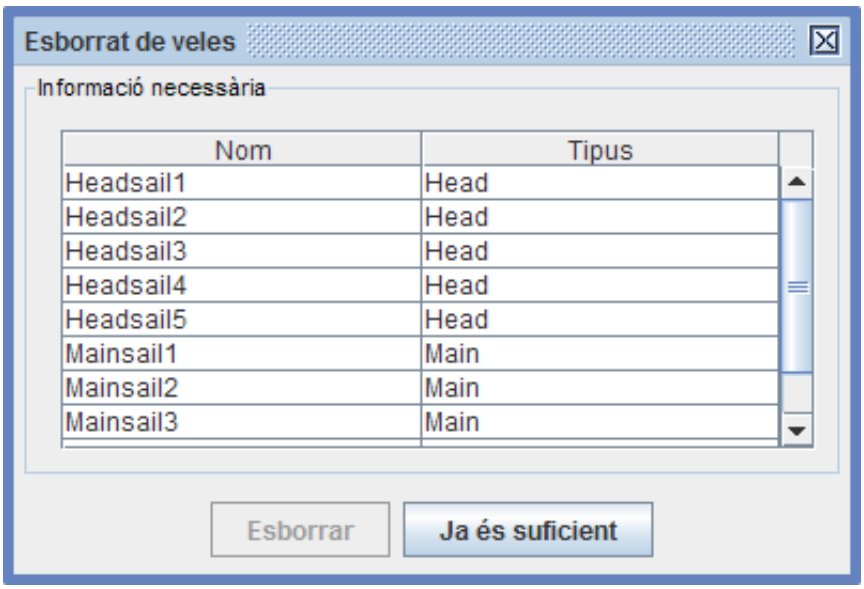


Figura 14: Vista d'esborrat de veles.

Descripció dels camps de la vista

- *Nom*: Nom de la vela.
- *Tipus*: Tipus de la vela.

3.5 Gràfiques

La següent figura mostra les funcionalitats de generació de gràfiques del sistema:

Figura 15: Funcionalitats del sistema per generar gràfiques.

Funcionalitats del menú:

- *TWAngle vs Time*: Actualització de les dades d'una regata.
- *X vs Múltiples Y*: Generació de gràfiques d'anàlisi sobre les dades d'una regata.

3.5.1 TWAngle vs Time

L'objectiu de la vista és permetre a l'usuari que actualitzi certa informació de les regates importades a partir d'una gràfica de *l'angle real entre el rumb de l'embarcació i el rumb del vent* en funció del temps.

L'usuari pot modificar el valor de qualsevol de les següents variables en qualsevol instant de la regata seleccionada:

- *Tram de la regata*
- *Vela major*
- *Vela de proa*
- *Qualitat de les dades*

Les modificacions realitzades per l'usuari sobre la gràfica es propaguen i fan persistents en temps real. Això significa que variables com el temps d'ús de les veles que depenen directament del que l'usuari introdueix es s'actualitzen immediatament després de confirmar els canvis realitzats. Així, el sistema ofereix sempre informació real i útil a l'usuari.

3.5.1.1 Selecció de la informació a actualitzar

La següent figura mostra la vista de selecció prèvia a la *TWAngle vs Time*:

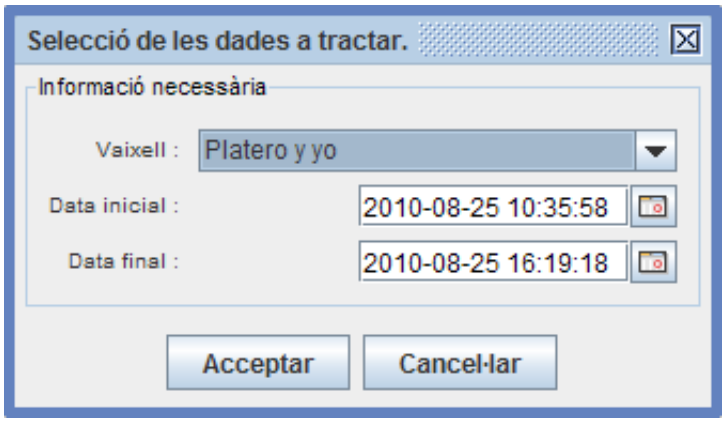


Figura 16: Selecció de la informació a actualitzar.

Descripció dels camps de la vista

- *Vaixell*: Nom del vaixell del que es volen actualitzar les dades.
- *Data d'inici*: Data inicial a partir de la que s'obtindrà la informació a actualitzar.
- *Data de fi*: Data final fins la que s'obtindrà la informació a actualitzar.

3.5.1.2 Descripció de la vista

La vista es divideix en tres components principals:

1. *Barra d'eines*
2. *Gràfica TWAngle vs Time*
3. *Gràfiques temporals de les variables d'actualització:*
    - Vela major
    - Tram de la regata
    - Vela de proa
    - Qualitat de les dades

La següent figura mostra cadascun dels components de la vista:

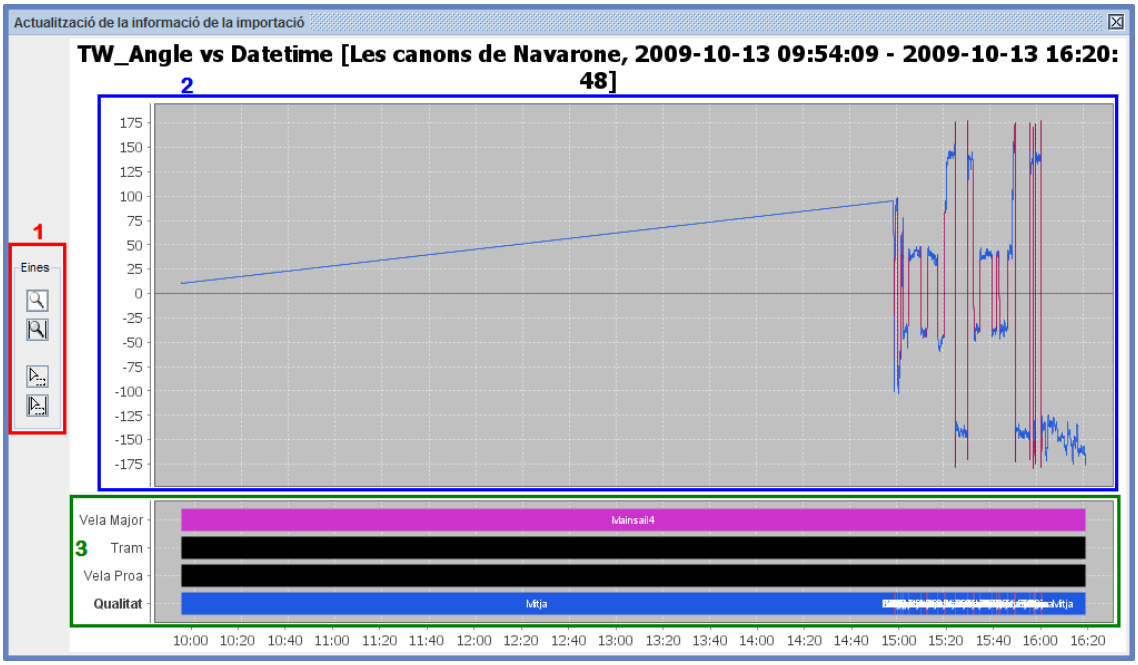


Figura 17: Components principals de la vista.

3.5.1.2.1 Descripció de la barra d'eines

La barra d'eines està formada per un grup de quatre botons. Els dos primers botons serveixen per realitzar *zoom* sobre les gràfiques. Els dos últims s'utilitzen per seleccionar les zones de la *gràfica TWAngle vs Time* que es volen actualitzar.

La realització de *zoom* i la selecció d'àrees a actualitzar són excloents entre si i no es poden realitzar alhora.

Quan s'utilitza un dels botons de selecció el sistema automàticament mostra els punts de la gràfica, així l'usuari pot observar exactament quins instants han quedat afectats per l'actualització.

Els *zoom* i les seleccions es duen a terme arrossegant el cursor (amb el botó esquerre del ratolí premut) d'esquerra a dreta i sentit descendent. Mentre s'arrossega el cursor el sistema mostra l'àrea que contindrà els punts afectats.

Per desfer tan els *zoom* com les seleccions només cal arrossegar el cursor (amb el botó esquerre del ratolí premut) en sentit invers. Per tant, de dreta a esquerra i sentit ascendent.

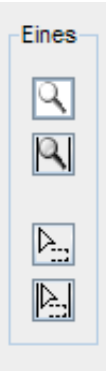


Figura 18: Barra d'eines.

A continuació s'explica la funcionalitat de cadascun dels botons de la barra:

Zoom de l'àrea delimitada pel cursor. A continuació es pot observar un exemple:

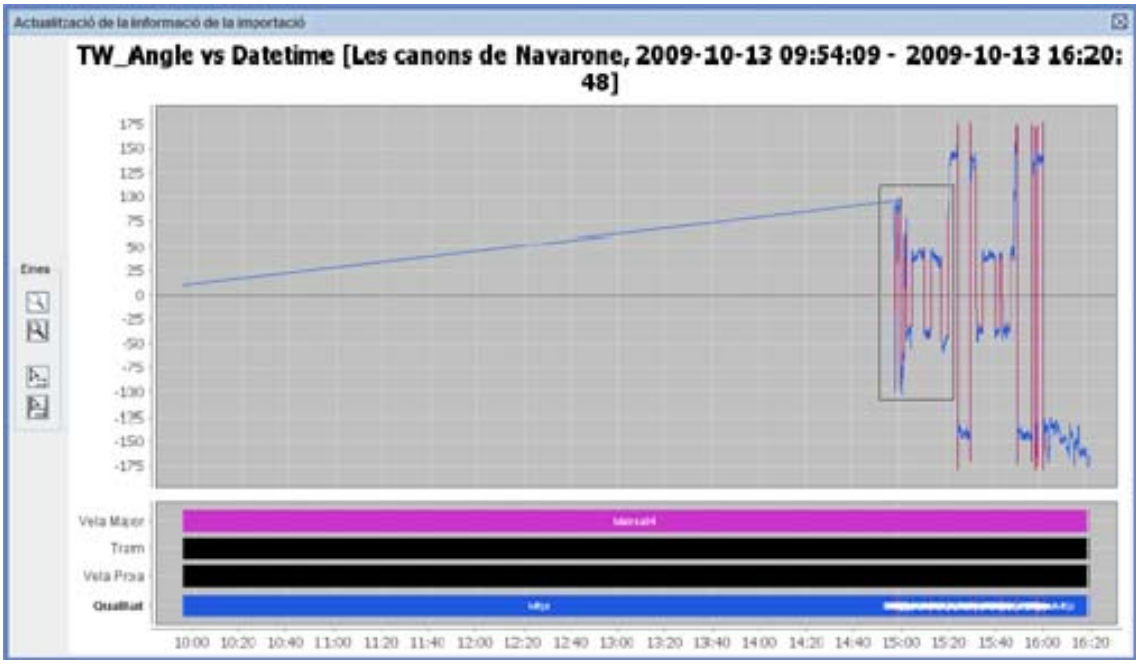


Figura 19: Petició de *zoom* sobre una àrea.

Figura 20: Àrea amplificada.

Zoom en domini. Es realitza *zoom* de l'àrea delimitada entre dos instants de la regata. A continuació es pot observar un exemple:

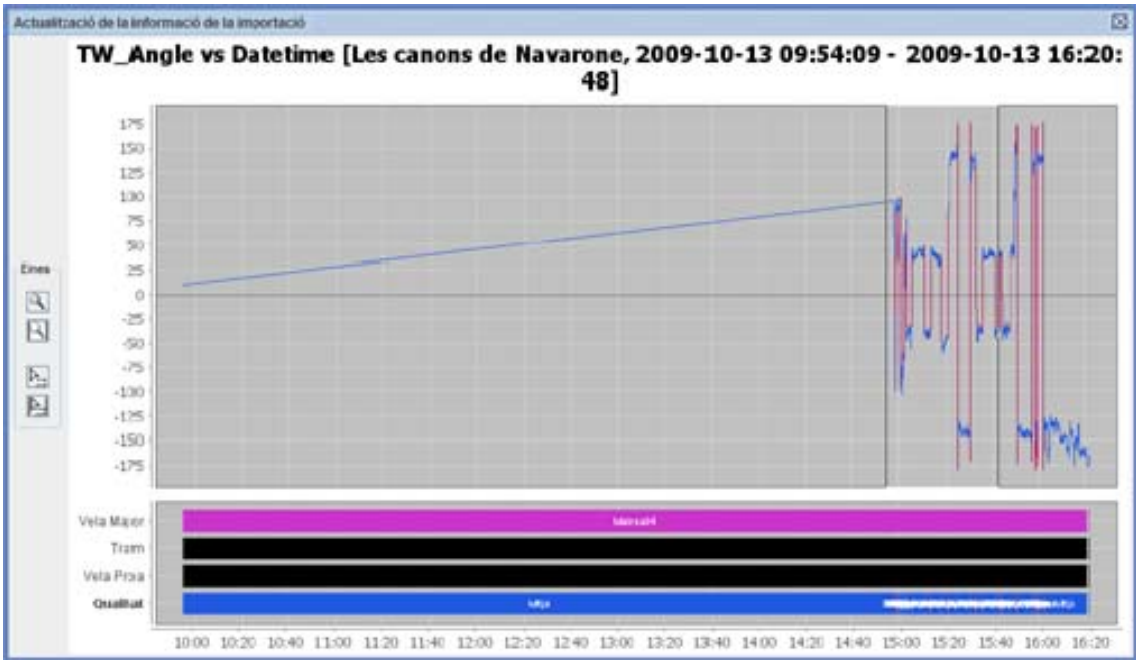


Figura 21: Petició de *zoom* en domini.

Figura 22: Àrea amplificada.

Selecció de l'àrea delimitada pel cursor amb l'objectiu d'actualitzar la seva informació.

A continuació es pot observar un exemple:

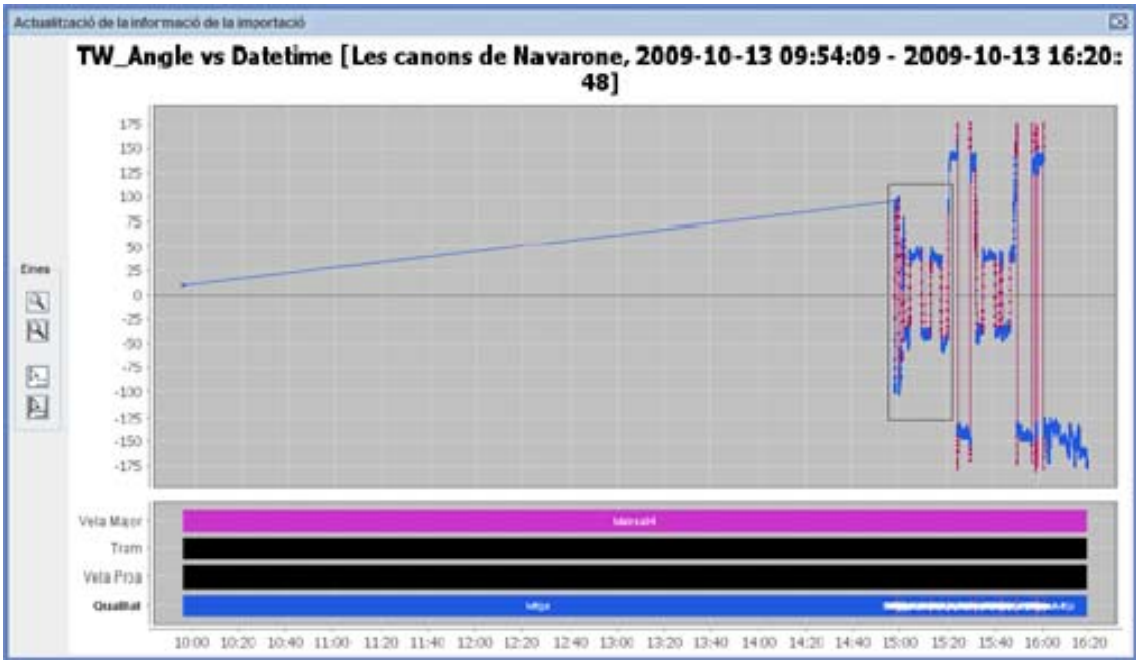


Figura 23: Petició de selecció sobre una àrea.

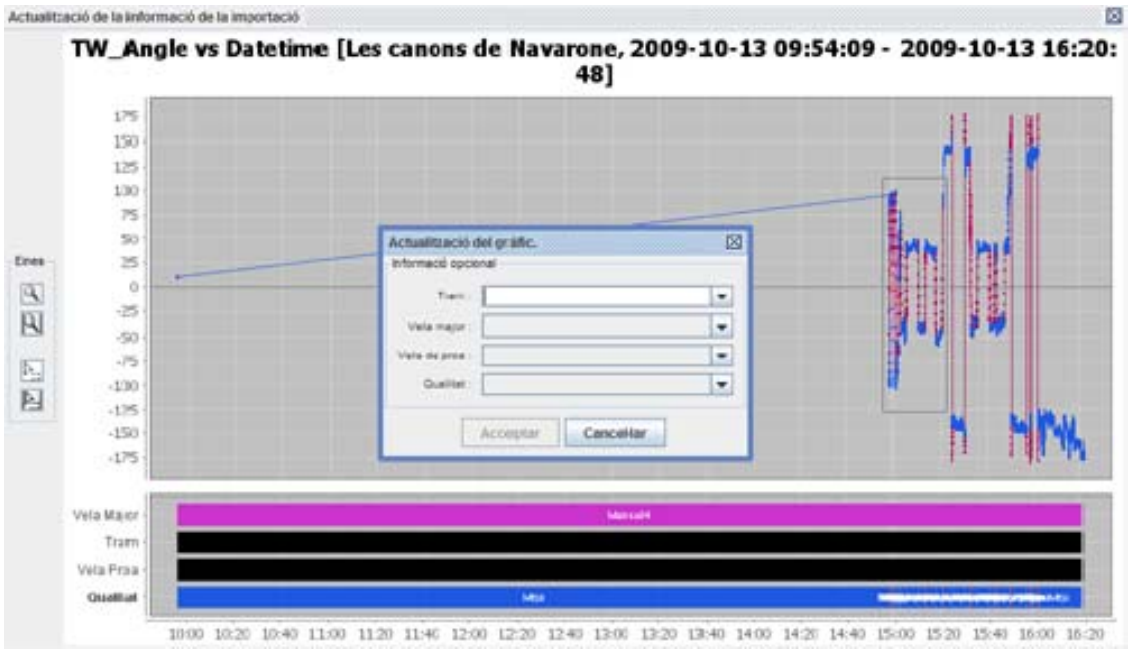


Figura 24: Àrea amplificada pendent d'actualitzar.

Selecció en domini. Es selecciona l'àrea delimitada entre dos instants de la regata. L'àrea seleccionada queda pendent de ser actualitzada. A continuació es pot observar un exemple:

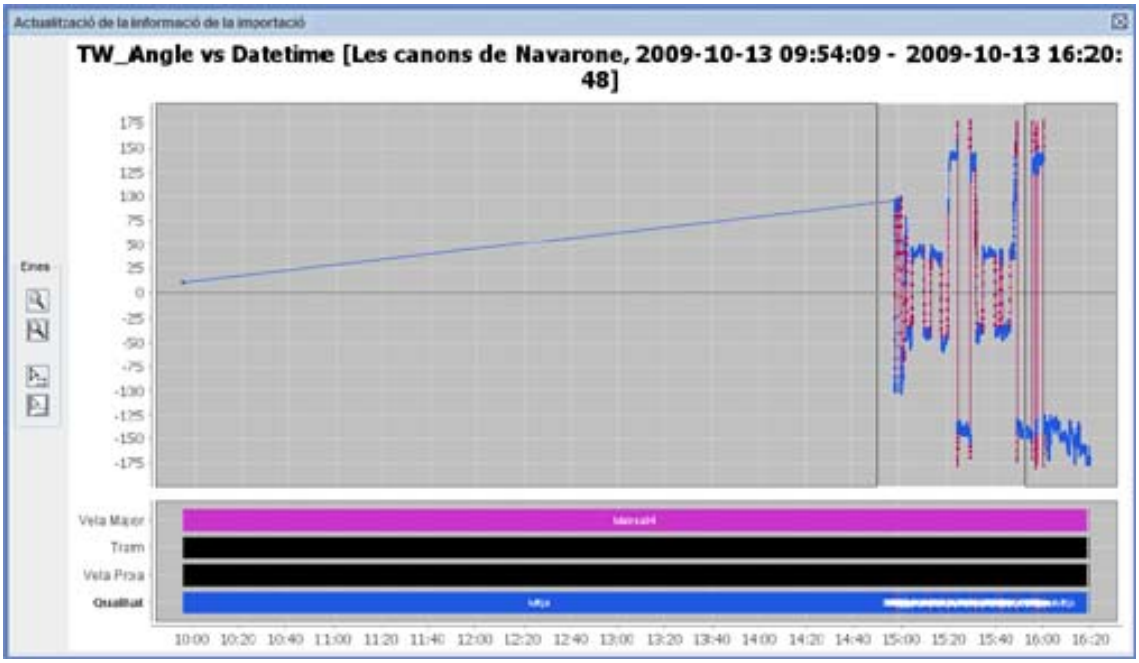


Figura 25: Petició de selecció en domini.

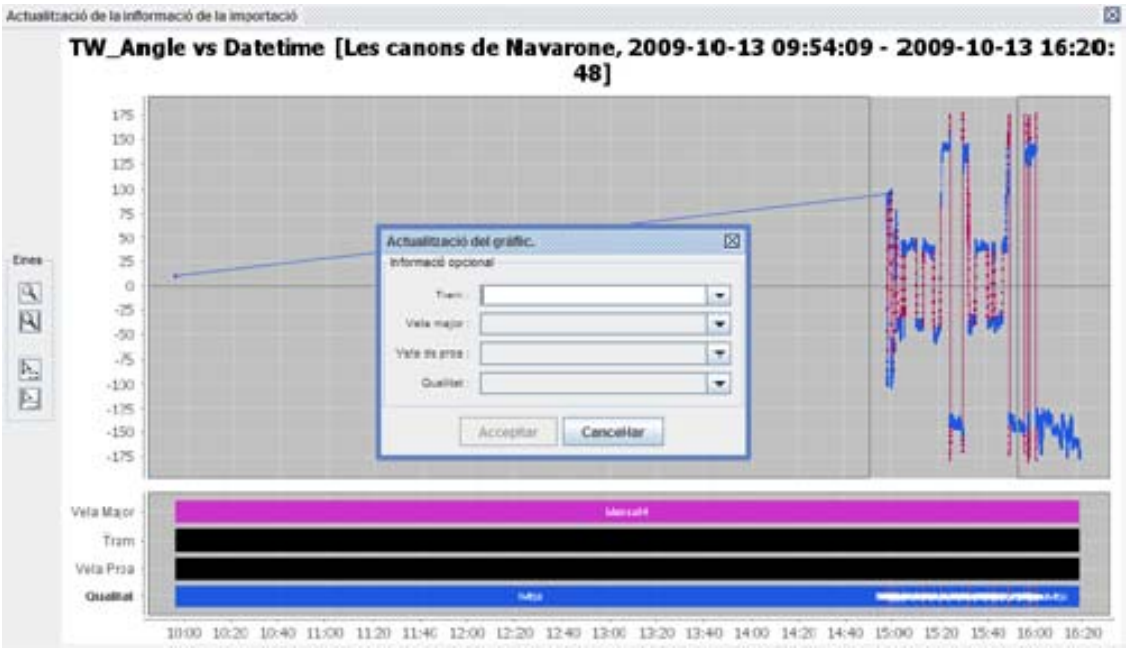


Figura 26: Àrea amplificada pendent d'actualitzar.

3.5.1.2.2 Gràfica TWA vs Time

Com el seu nom indica mostra els valors de *l'angle real entre el rumb de l'embarcació i el rumb del vent* en funció del temps.

L'usuari pot realitzar tant accions de *zoom* com de selecció per actualitzar àrees concretes sobre aquesta gràfica. Cada vegada que es selecciona una àrea per actualitzar-la el sistema mostra la *vista d'actualització de la informació* afectada. Totes i cadascuna de les actualitzacions realitzades sobre aquesta gràfica es reflecteixen immediatament en les *gràfiques temporals de les variables d'actualització* afectades.

Les accions a realitzar per dur a terme una actualització de les dades seleccionades d'una regata són les següents:

1. Gràfica en l'estat original abans de realitzar el nou canvi.

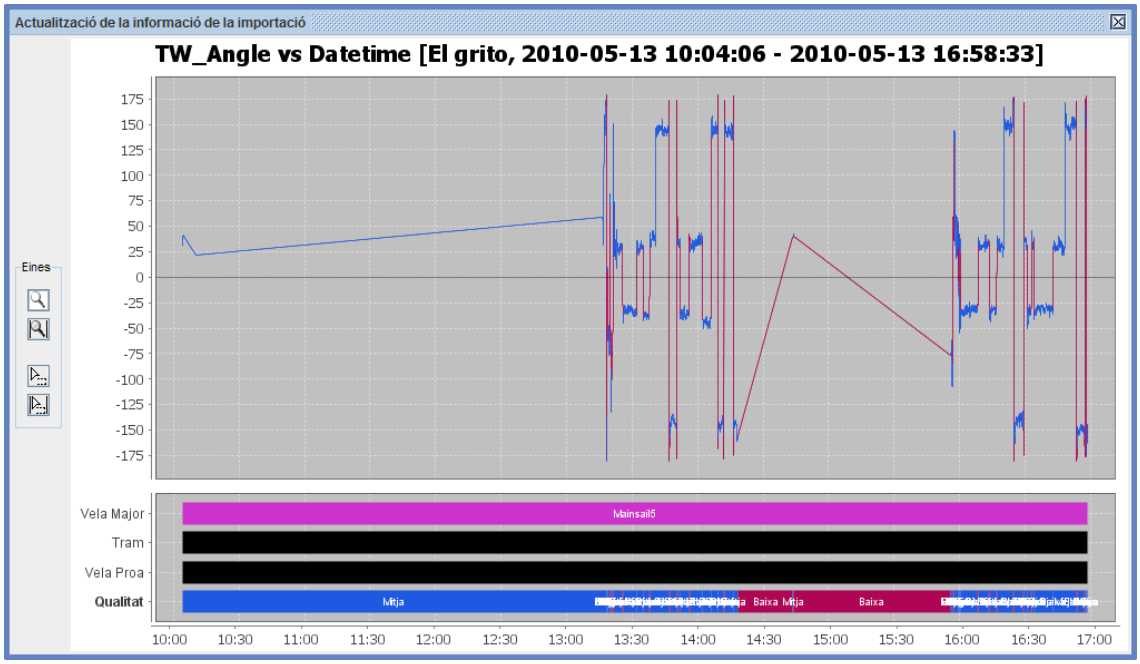


Figura 27: Gràfica amb les dades abans del canvi.

2. Realitzar tants *zoom* com siguin necessaris per visualitzar l'àrea que es desitja actualitzar correctament.

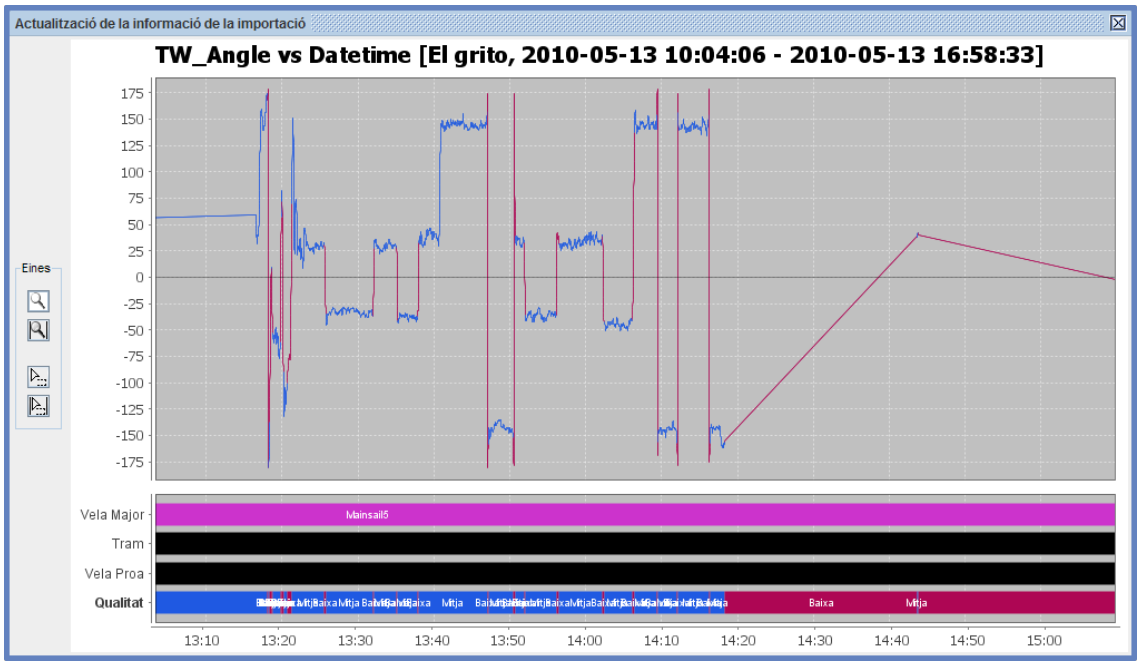


Figura 28: Gràfica amb el *zoom* desitjat.

3. Quan l'àrea desitjada es pot observar correctament activar un dels botons de selecció. El sistema mostra els punts de la gràfica, així l'usuari serà capaç de comprovar l'afectació real sobre les dades.

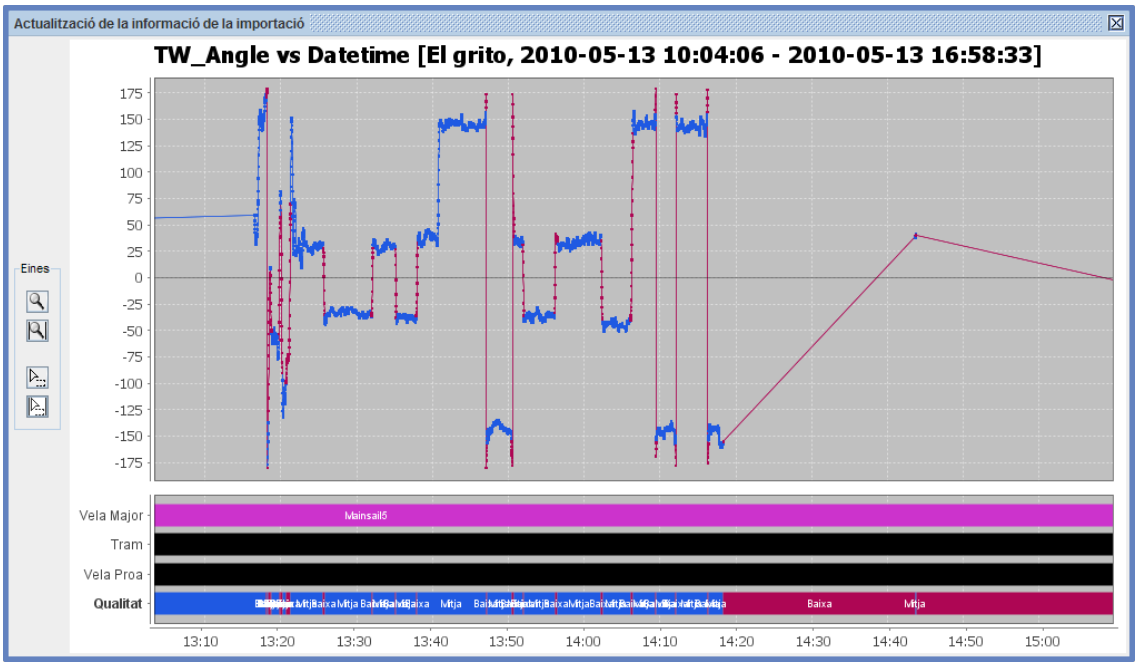


Figura 29: Gràfica en mode selecció.

4. Seleccionar l'àrea.

Figura 30: Selecció de l'àrea a actualitzar.

5. El sistema mostra la vista d'actualització de les dades.

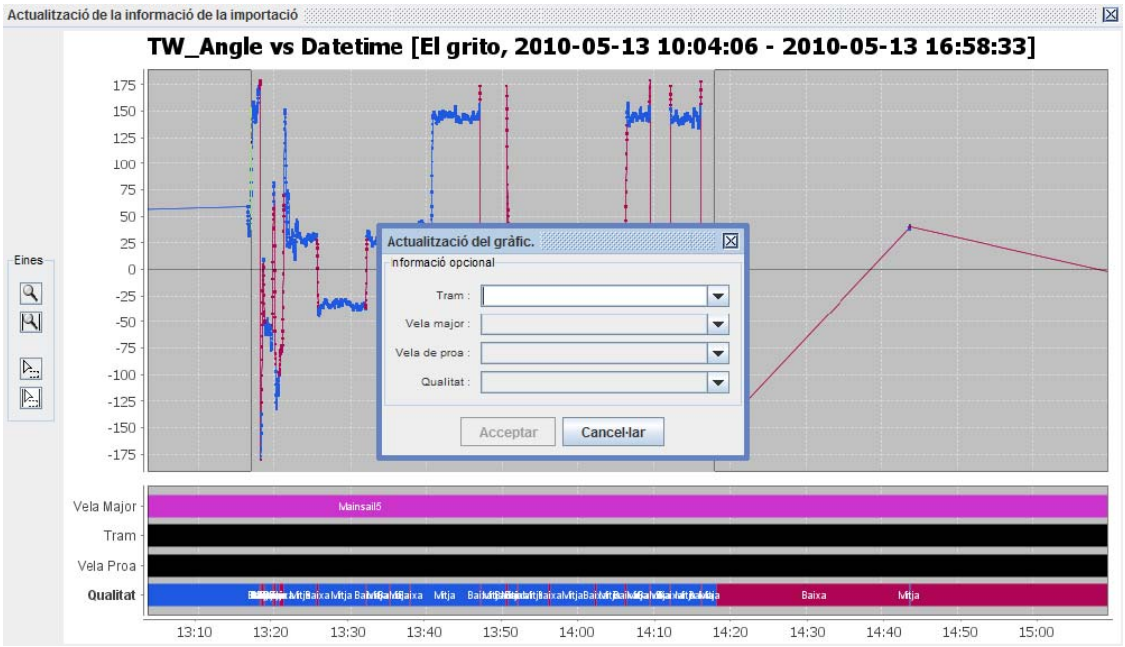


Figura 31: Vista d'actualització de les dades seleccionades de la regata.

6. Introduir els valors desitjats i confirmar l'actualització.

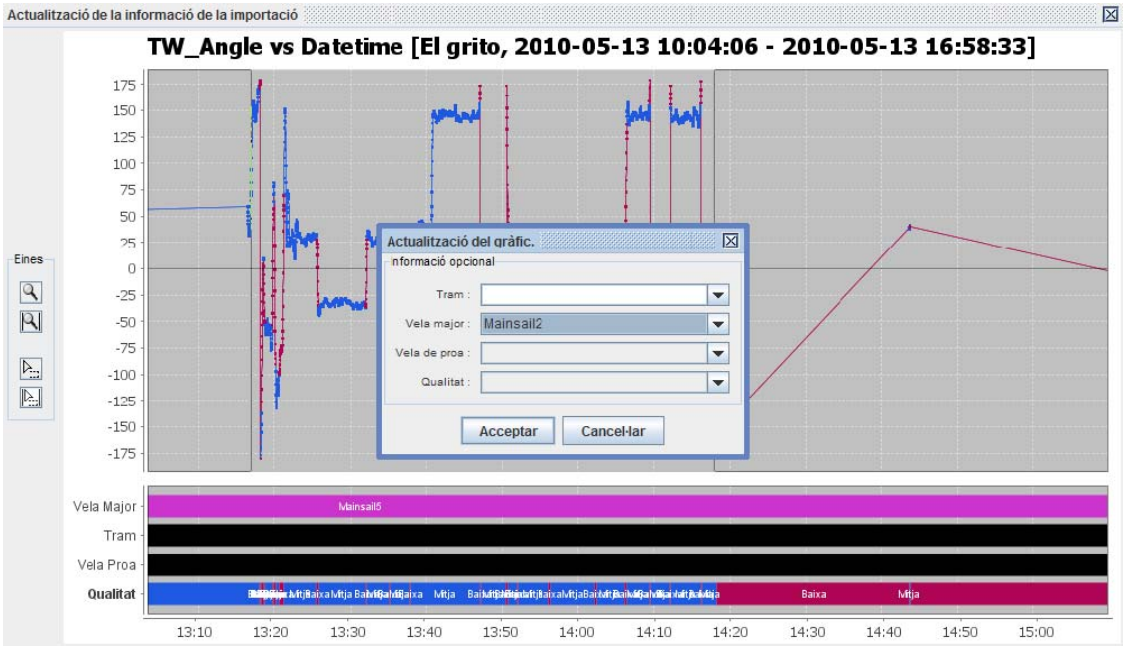


Figura 32: Canvis introduïts i confirmació.

7. El sistema actualitza la informació i mostra l'àrea afectada amb els canvis realitzats a l'usuari.

Figura 33: Dades actualitzades.

8. Desfer el *zoom* si s'escau.

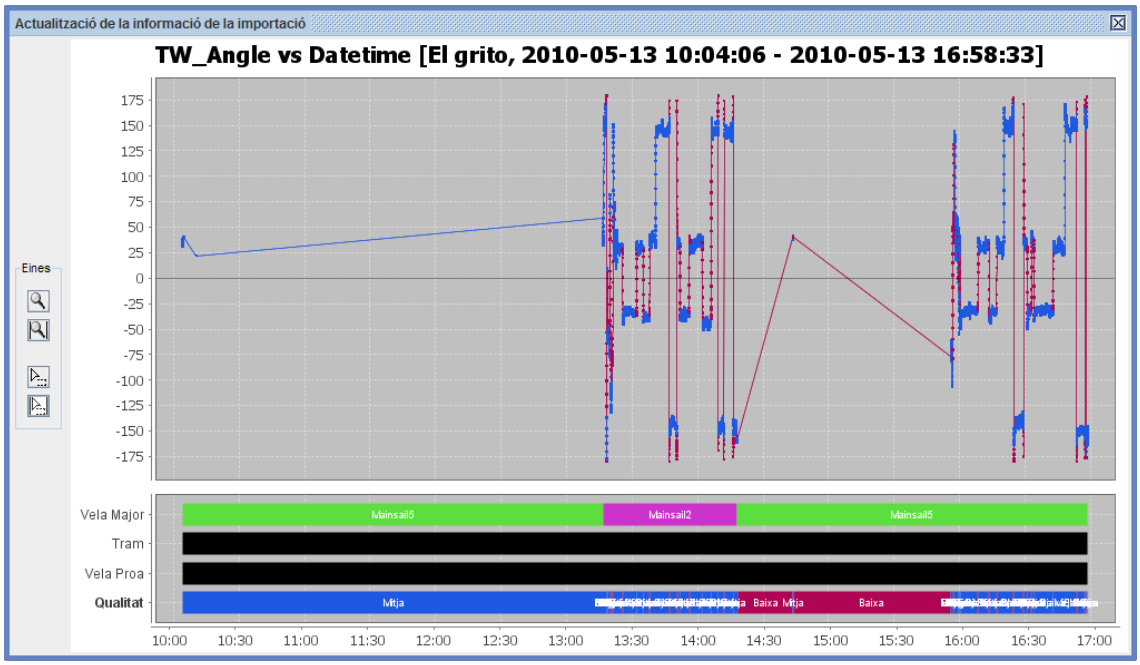


Figura 34: Gràfica amb les dades després del canvi.

3.5.1.2.2.1 Descripció de la vista

La següent figura mostra la vista d'actualització d'informació de la regata:

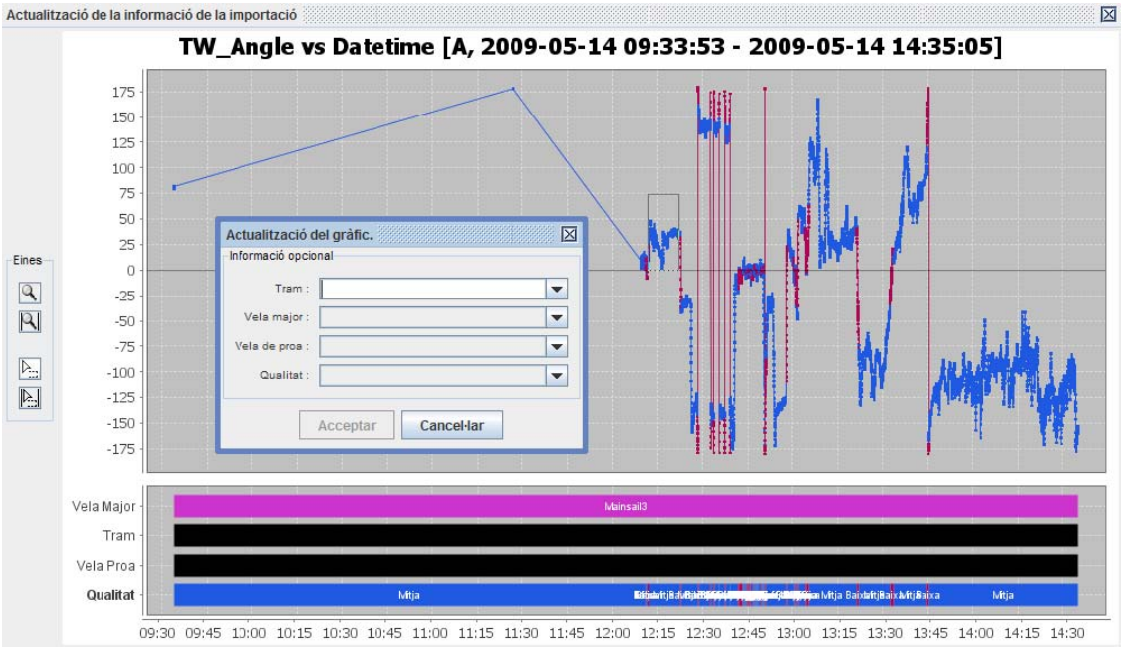


Figura 35: Vista d'actualització d'informació d'una regata.

Descripció dels camps de la vista

- *Tram*: Nom del tram que s'assignarà a totes les dades incloses dins l'àrea seleccionada. Aquests camp permet a l'usuari seleccionar un tram entre els existents o introduir-ne un de nou.
- *Vela major*: Nom de la vela major que s'assignarà a totes les dades incloses dins l'àrea seleccionada. El camp només permet la selecció d'una vela major entre les existents.
- *Vela de proa*: Nom de la vela de proa que s'assignarà a totes les dades incloses dins l'àrea seleccionada. El camp només permet la selecció d'una vela de proa entre les existents.
- *Qualitat*: Qualitat de les dades que s'assignarà a totes les dades incloses dins l'àrea seleccionada. El camp només permet la selecció d'una qualitat entre les existents.

Cap dels camps de la vista és obligatori. L'usuari pot actualitzar les dades seleccionades pels camps de la vista que desitgi. Quan es confirmen els canvis el sistema actualitza la informació amb els valors dels camps que no són buits.

3.5.1.2.3 Gràfiques temporals de les variables d'actualització

Cada gràfica reflexa el valor actual de les dades per a la variable que representen.

Tot i que aquestes gràfiques no estan orientades a l'ús del mode selecció per actualitzar les dades de la regata, suporten la selecció en domini. Qualsevol actualització realitzada sobre aquestes gràfiques es reflexa immediatament en la *gràfica TWA vs Datetime*.

Després d'importar una regata, variables com el *Tram* d'una regata o la *Vela de proa* usada encara no han pres cap valor i es mostren en negre. Mentre no s'inicialitzin els valors per aquestes variables la seva representació es mantindrà en aquest color.

La *gràfica TWA vs Datetime* es mostra per defecte amb la representació dels valors que pren la variable *Qualitat* en funció del temps. Així ho indica la vista, doncs si observem l'etiqueta de la gràfica de la variable *Qualitat* està ressaltada en negreta.

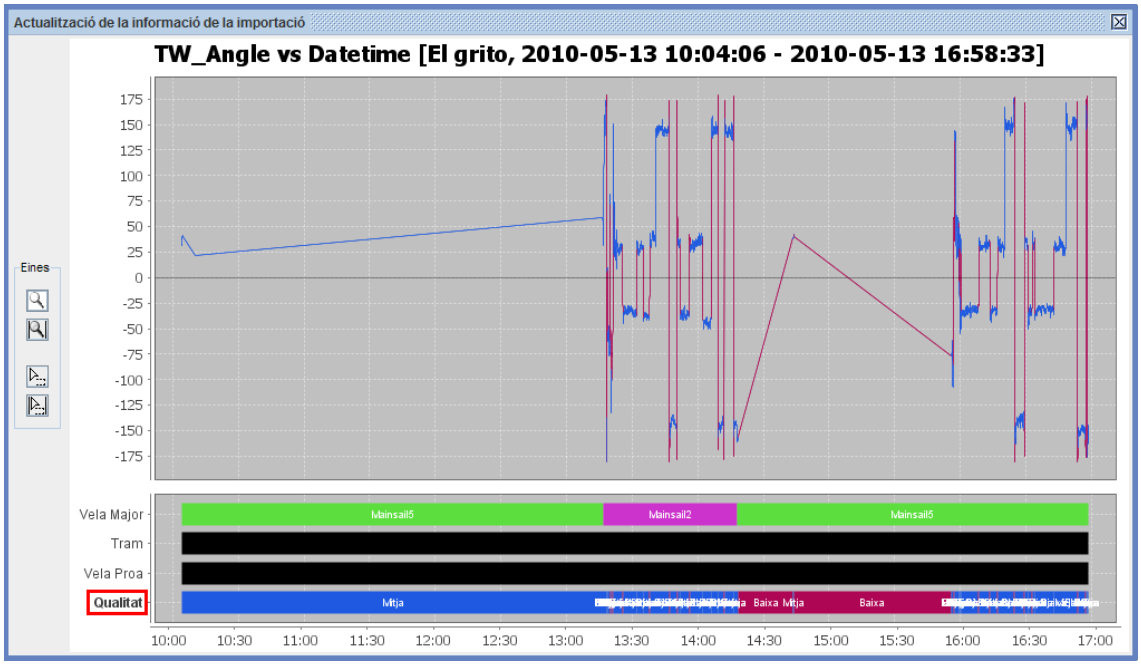


Figura 36: Es mostra la variable *Qualitat* per defecte a la *gràfica TWA vs Datetime*.

És possible canviar la variable temporal que s'està mostrant actualment a la gràfica principal, només cal prémer el botó esquerre del ratolí sobre l'etiqueta de la variable desitjada i el sistema actualitza la representació. Es pot confirmar que la variable és la que s'està mostrant perquè la seva etiqueta és la que ara està en negreta, a més que els colors coincideixen amb els de la representació de la *gràfica TWA vs Datetime*.

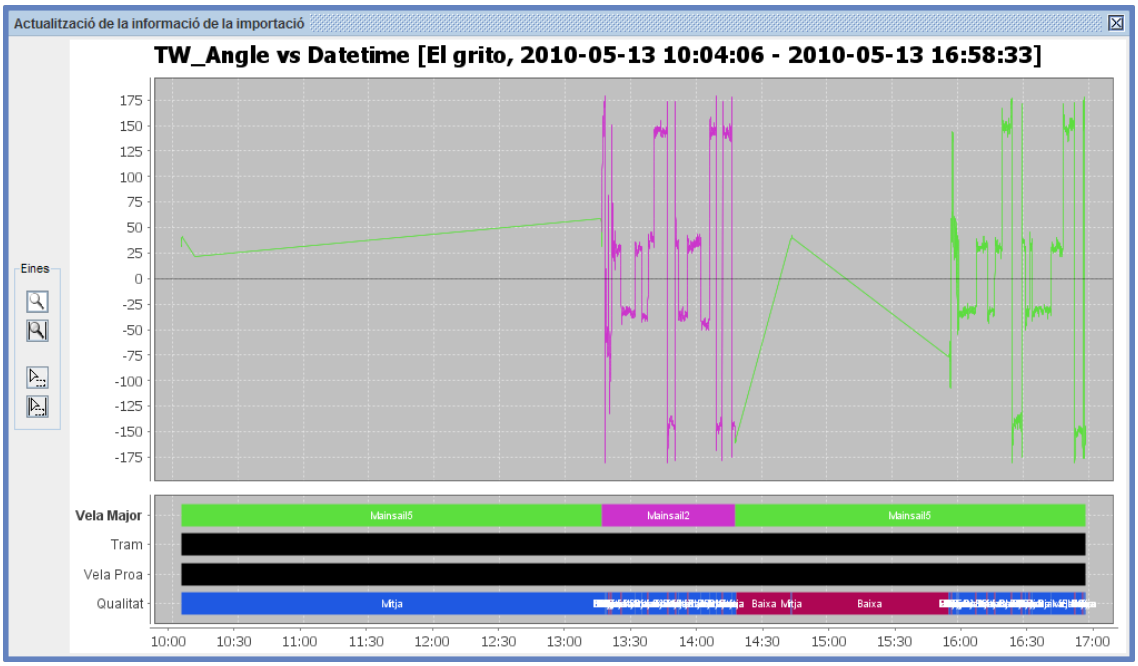


Figura 37: Representació de la variable *Vela major*.

3.5.2 X vs Múltiples Y

L'objectiu de la vista és permetre a l'usuari que generi gràfiques diverses a partir de les dades importades d'una regata.

El sistema permet la generació de diferents tipus de gràfiques d'anàlisi:

- *Gràfiques cronològiques*
- *Gràfiques d'anàlisi en funció de la variable TWS*
- *Altres gràfiques d'anàlisi*
- *Qualsevol de les gràfiques anteriors amb informació per babord i estribord*

En funció dels valors que esculli l'usuari en la vista obtindrà una gràfica d'un tipus o un altre.

3.5.2.1 Descripció de la vista

La vista és divideix en dos grups de camps. Els primers són obligatoris, l'usuari ha de seleccionar un valor entre els disponibles per poder generar qualsevol gràfica d'anàlisi. El segon grup està format pels camps opcionals, dels que l'usuari pot utilitzar els que desitgi, i serveixen per acotar la informació que el sistema ha de considerar al generar les gràfiques.

La següent figura mostra cadascun dels components de la vista:

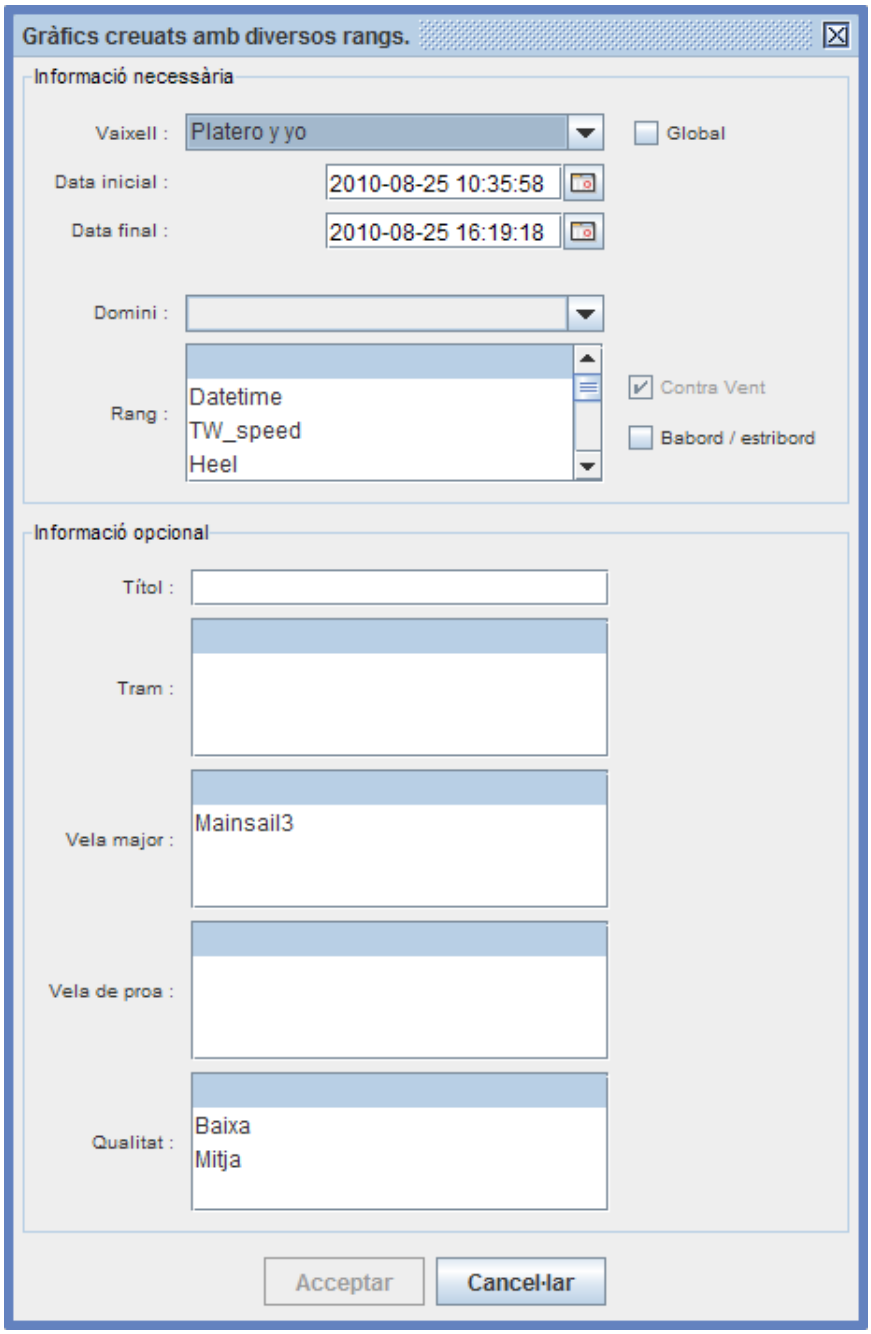


Figura 38: Vista X vs Múltiples Y.

A continuació es descriuen els camps obligatoris:

- *Vaixell*: Nom del vaixell sobre el que es desitja generar la gràfica d'anàlisi.
- *Global*: Si el camp està desactivat es pren com a conjunt de dades a analitzar tota la informació inclosa entre les dates d'inici i fi i el vaixell seleccionats. En canvi, si s'activa el camp el sistema utilitza tota la informació existent sobre el vaixell per generar la gràfica, descartant les dates d'inici i fi.
- *Data inicial*: Data d'inici a partir de la qual s'ha d'incloure en la gràfica la informació del vaixell. El camp només es té en compte si el camp *Global* està desactivat.
- *Data final*: Data de fi fins la que s'inclou en la gràfica la informació del vaixell. El camp només es té en compte si el camp *Global* està desactivat.
- *Domini*: Variable sobre la que es basen els valors de l'eix X.
- *Rang*: Variable sobre la que es basen els valors de l'eix Y. Les *gràfiques cronològiques* admeten múltiples rangs.
- *Contra vent*: Indica si les dades que s'han de tenir en compte en la gràfica són a favor o en contra del vent. Les *gràfiques cronològiques* són les úniques que no utilitzen el camp.
-
- *Babord / estribord*: Indica si es desitja que la gràfica consideri per separat la informació de babord de la d'estribord.

A continuació es descriuen els camps opcionals:

- *Títol*: Títol de la gràfica.
- *Tram*: La informació que s'utilitzarà per generar la gràfica només correspondrà al tram seleccionat.
- *Vela major*: La informació que s'utilitzarà per generar la gràfica només correspondrà a la vela major seleccionada.
- *Vela de proa*: La informació que s'utilitzarà per generar la gràfica només correspondrà a la vela de proa seleccionada.
- *Qualitat*: Si es selecciona la qualitat *Alta* només es considerarà la informació amb aquesta qualitat. Si es selecciona la qualitat *Mitja* es considerarà la informació tant d'aquesta qualitat com la de qualitat *Alta*. Si es selecciona la qualitat *Baixa* s'utilitzaran totes les dades sense distingir entre qualitats.

Existeix la possibilitat de seleccionar múltiples valors per un i només un dels camps opcionals. Una vegada s'ha escollit un dels camps per a una selecció múltiple, la resta de camps només admetran selecció individual. L'excepció és el camp *Qualitat* que no admet la selecció múltiple.

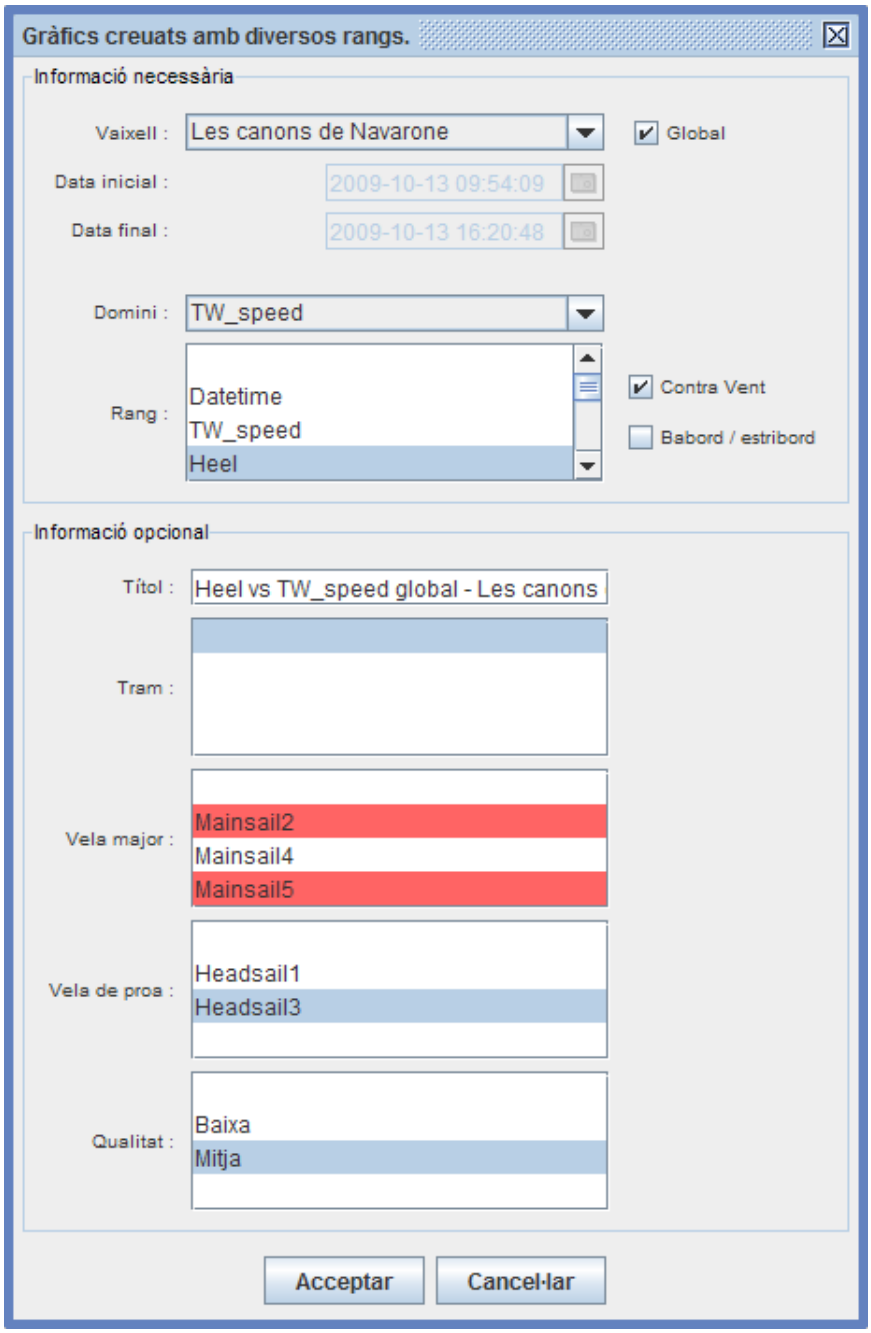


Figura 39: Selecció de múltiples valors del camp *Vela major*.

La figura anterior mostra una selecció de múltiples valors pel camp *Vela major*. Tota selecció múltiple es distingeix perquè els elements elegits es representen en vermell, mentre que les seleccions individuals es coloren en blau.

La gràfica que mostraria la figura correspondria a la de la variable *Heel* en funció del *TW_speed* usant la informació a contra vent que pertanyi a les veles majors *Mainsail 2* i *Mainsail 5*, a la vela de proa *Headsail 3* i que sigui de qualitat *Alta* i *Mitja* a partir de les dades existents al sistema del vaixell *Les canons de Navarone*. Addicionalment, la gràfica tindria per títol *Heel vs TW_speed global - Les canons de Navarone*.

Descripció de les gràfiques

Totes les gràfiques es divideixen en tres components principals a banda de la informació representada:

1. *Títol de la gràfica*: Les gràfiques tenen un títol per defecte que conté el tipus de gràfica generada, el vaixell i les dates de la regata. L'usuari també pot afegir informació addicional a través del camp *Títol* de la vista.
2. *Restriccions comunes*: Les restriccions comunes són les que comparteixen totes les variables representades en la gràfica. Aquestes restriccions corresponen a les seleccions individuals en els camps opcionals, o paràmetres generals de la vista com el camp *Contra Vent*.
3. *Llegenda amb les variables representades i les seves restriccions específiques*: Llegenda que conté les variables que es representen a la gràfica i per cadascuna d'elles les restriccions específiques aplicades. Les restriccions específiques corresponen a la selecció múltiple en un dels camps opcionals, o paràmetres generals de la vista com el camp *Babord / estribord*. En *gràfiques d'anàlisi en funció del TWS*, si la variable representada té *fitxer de targets* pel vaixell seleccionat també es mostraran els valors objectiu.

La següent figura mostra cadascun dels components d'una gràfica:

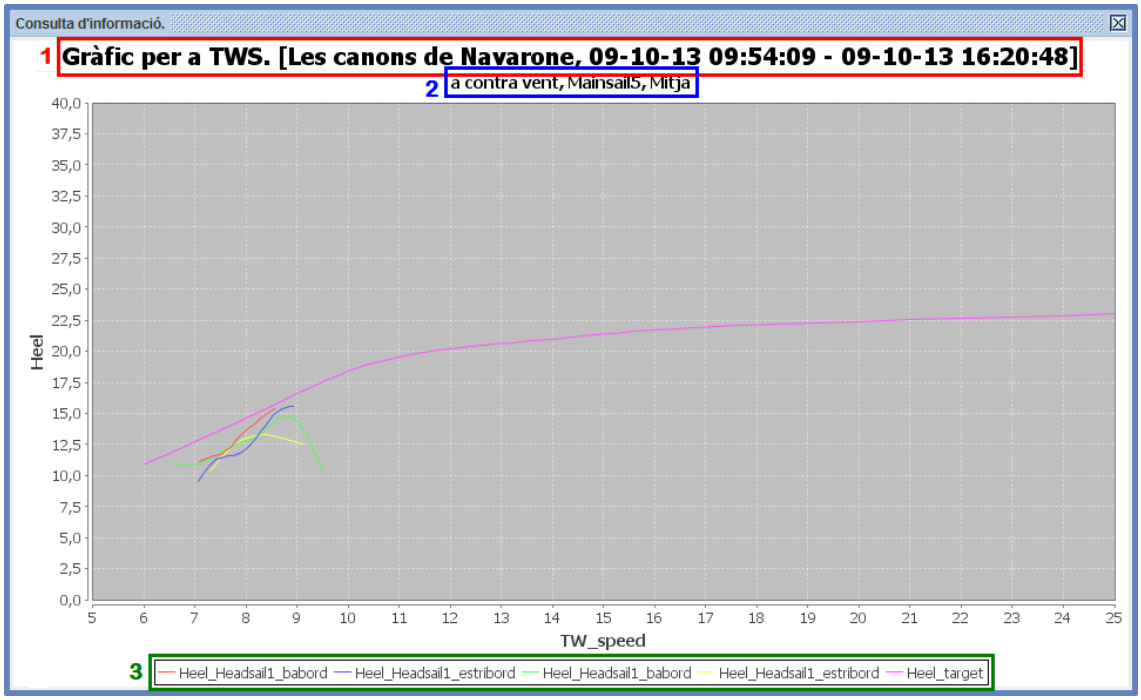


Figura 40: Components principals d'una gràfica.

3.5.2.2 Gràfiques cronològiques

Les gràfiques cronològiques mostren l'evolució d'una o més variables d'estudi en funció del temps. Aquestes gràfiques executen el *filtre XvsY*.

Per crear una gràfica cronològica cal escollir la variable *Datetime* com a domini i tantes variables com es desitgin consultar en el rang. Per exemple:

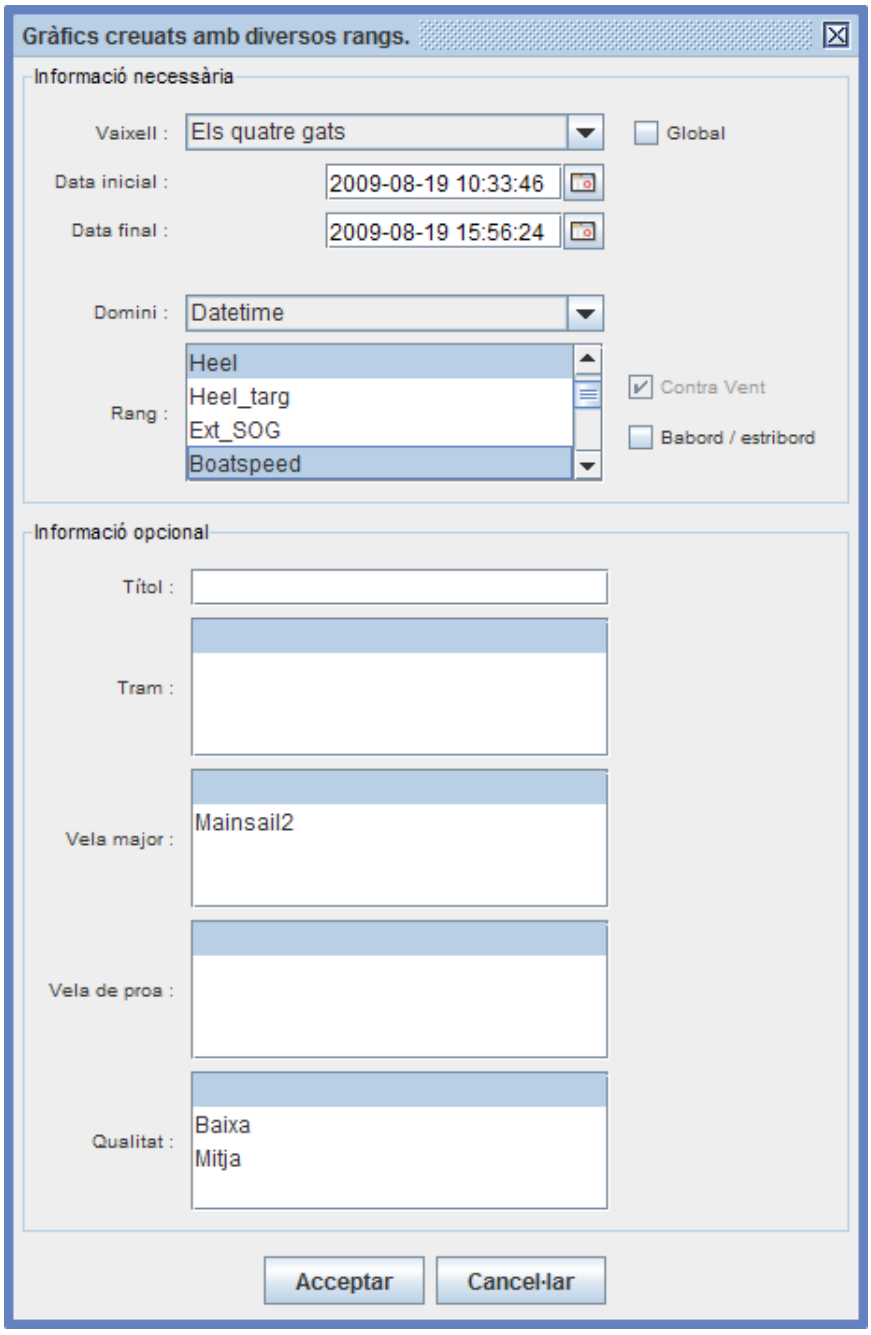


Figura 41: Petició d'una gràfica cronològica per les variables *Heel* i *Boatspeed*.

La següent figura mostra la gràfica generada a partir dels paràmetres anteriors:

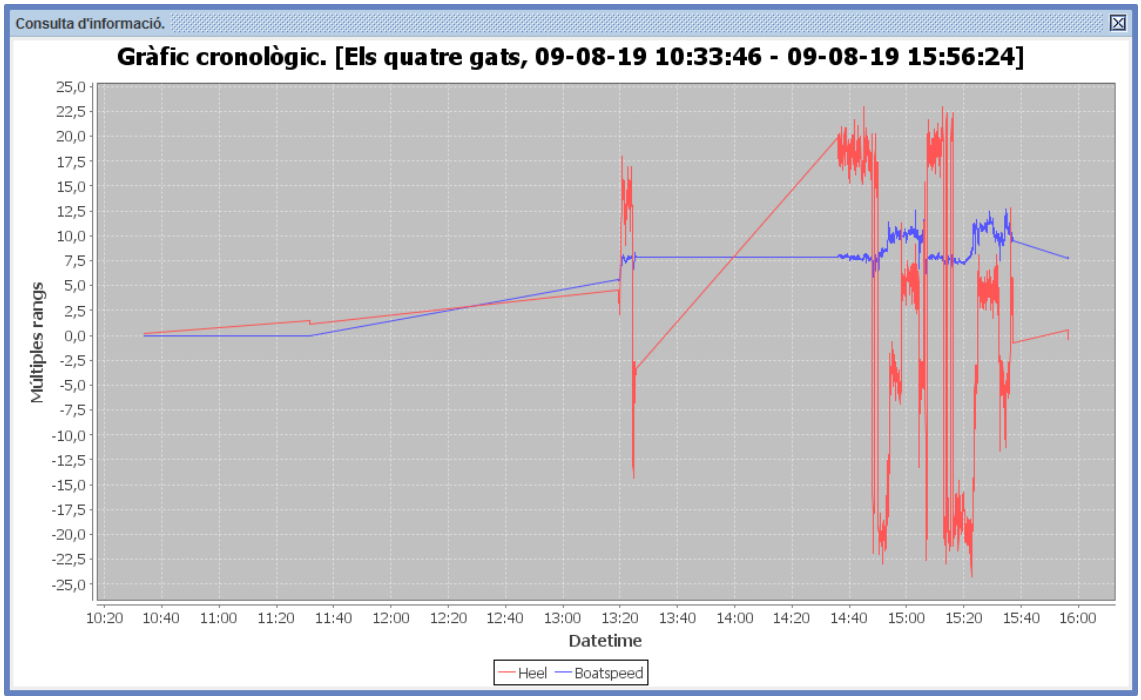


Figura 42: Gràfica cronològica.

3.5.2.3 Gràfiques d'anàlisi en funció del *TWS*

Les gràfiques d'anàlisi en funció del *TWS* mostren l'evolució d'una i només una variable d'estudi en funció de la velocitat real del vent. Aquestes gràfiques executen el *filtre TWSvsY*.

Per crear una gràfica d'anàlisi en funció del *TWS* cal escollir la variable *TW_speed* com a domini i la variable que es desitgi analitzar en el rang. Per exemple:

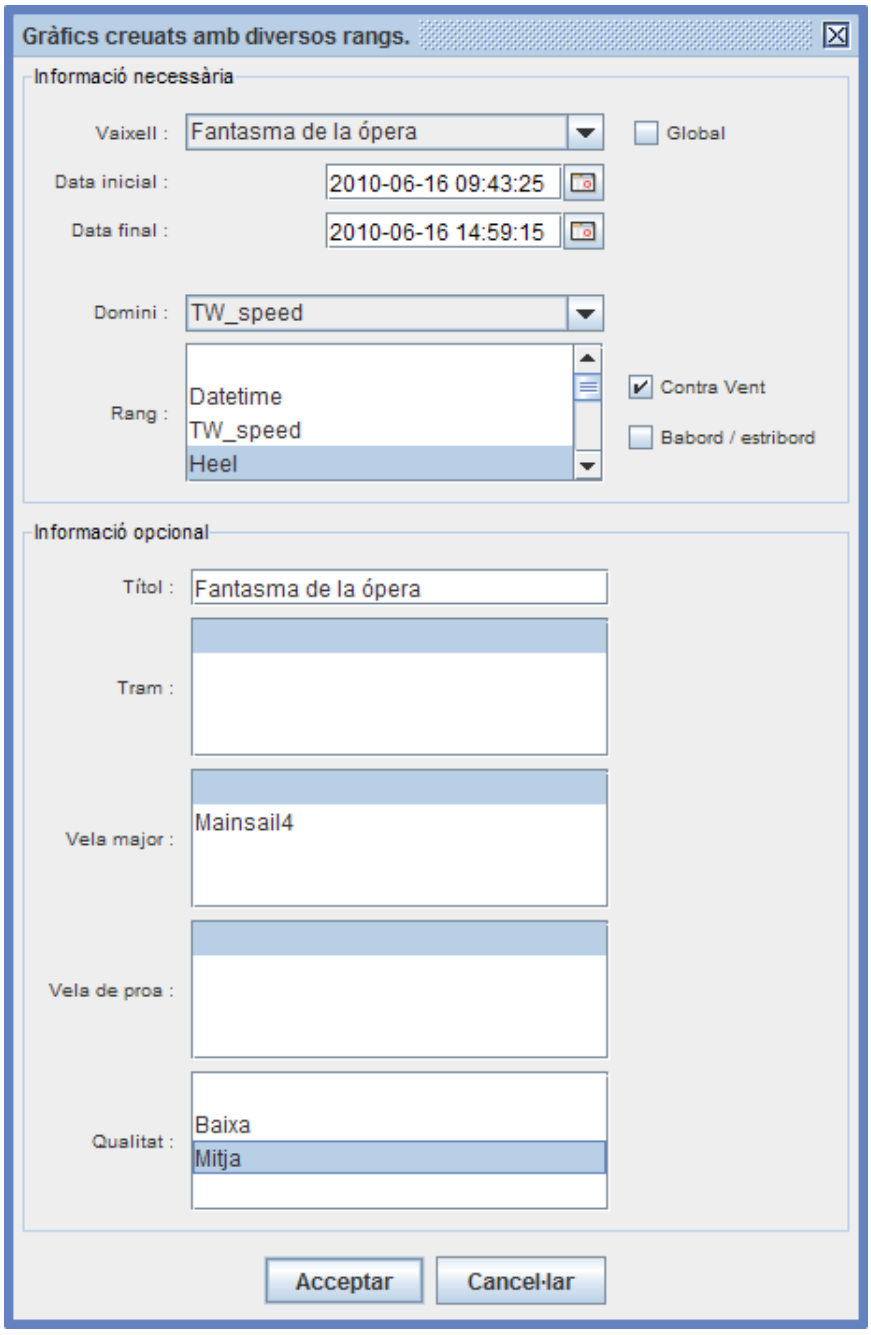


Figura 43: Petició d'una gràfica d'anàlisi del Heel en funció del *TWS*.

La següent figura mostra la gràfica generada a partir dels paràmetres anteriors:

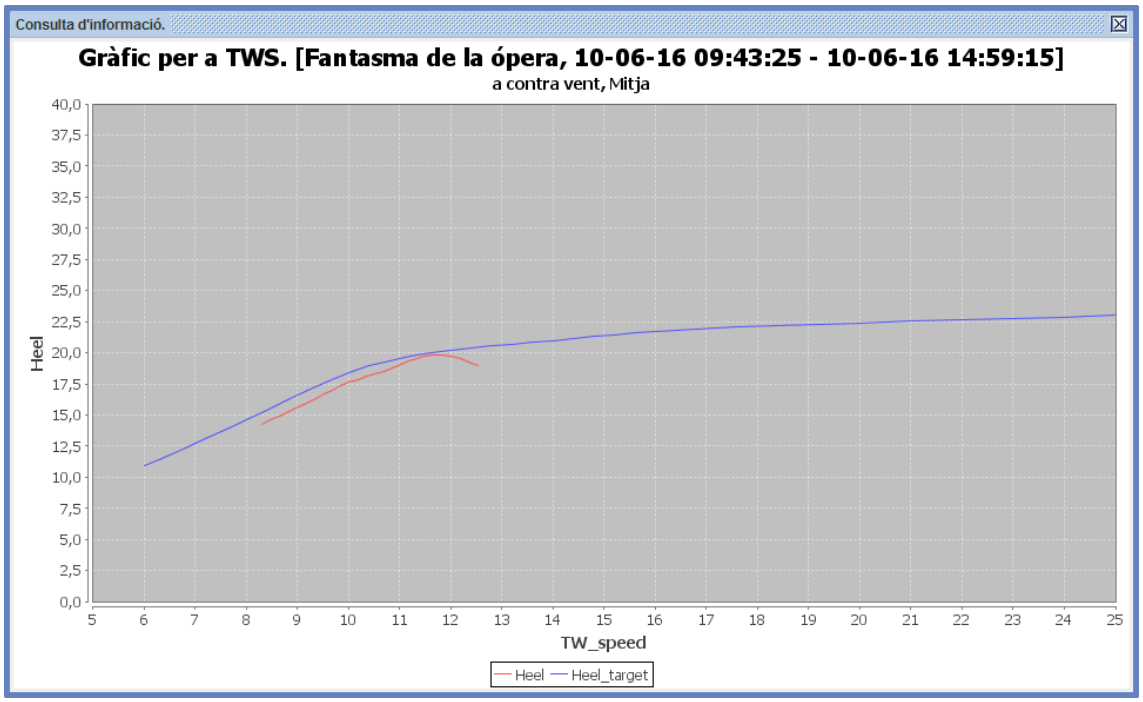


Figura 44: Gràfica d'anàlisi del Heel en funció del *TWS*.

Sempre que existeixi un *fitxer de targets* per la variable d'estudi i el vaixell seleccionats el sistema el mostrarà juntament amb les representacions demanades.

3.5.2.3.1 Fitxers de targets

Els *fitxers de targets* són fitxers de text pla que contenen els valors objectius proporcionats pel fabricant per a una variable d'estudi i un vaixell de regates. Cada registre dels fitxers conté dos valors, el primer correspon a la velocitat real del vent i el segon al valor objectiu del vaixell per aquesta velocitat. El separador dels camps correspon al caràcter tabulador.

El directori de la instal·lació del sistema on han de residir els fitxer de targets és el següent:

Projecte Regates\bin\targets

El nom d'aquests fitxers ha de seguir el format que es mostra a continuació i així el sistema serà capaç d'identificar-los quan sigui necessari.

Format per a trams a contra vent:

*NomVaixell_VariableEstudi*_targets.txt

Per exemple alguns noms vàlids per aquests fitxers són:

- El grito_TW_angle_targets.txt
- Els quatre gats_Boatspeed_targets.txt
- Fantasma de la ópera_Rudder_targets.txt
- Les canons de Navarone_Heel_targets.txt
- Platero y yo_VMG_targets.txt

Format per a trams a favor del vent:

*NomVaixell_VariableEstudi*_targets_down.txt

Per exemple alguns noms vàlids per aquests fitxers són:

- El grito_TW_angle_targets_down.txt
- Els quatre gats_Boatspeed_targets_down.txt
- Fantasma de la ópera_Rudder_targets_down.txt

- Les canons de Navarone_Heel_targets_down.txt
- Platero y yo_VMG_targets_down.txt

La següent figura mostra un possible contingut d'un fitxer:

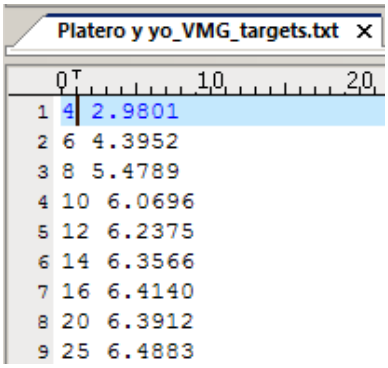


Figura 45: Valors d'un *fitxer de targets* per la variable *VMG* del vaixell *Platero y yo*.

3.5.2.4 Altres gràfiques d'anàlisi

És possible generar altres gràfiques d'anàlisi, a part de les *gràfiques d'anàlisi en funció de la variable TWS*. Només permeten representar una i només una variable en el rang a partir de qualsevol variable diferent al *TWS* i el *Datetime*. Aquestes gràfiques també executen el *filtre TWSvsY*.

En la versió actual el sistema no dóna suport per la lectura de *fitxers de targets* en aquest tipus de gràfiques.

Per exemple, a partir dels següents paràmetres:

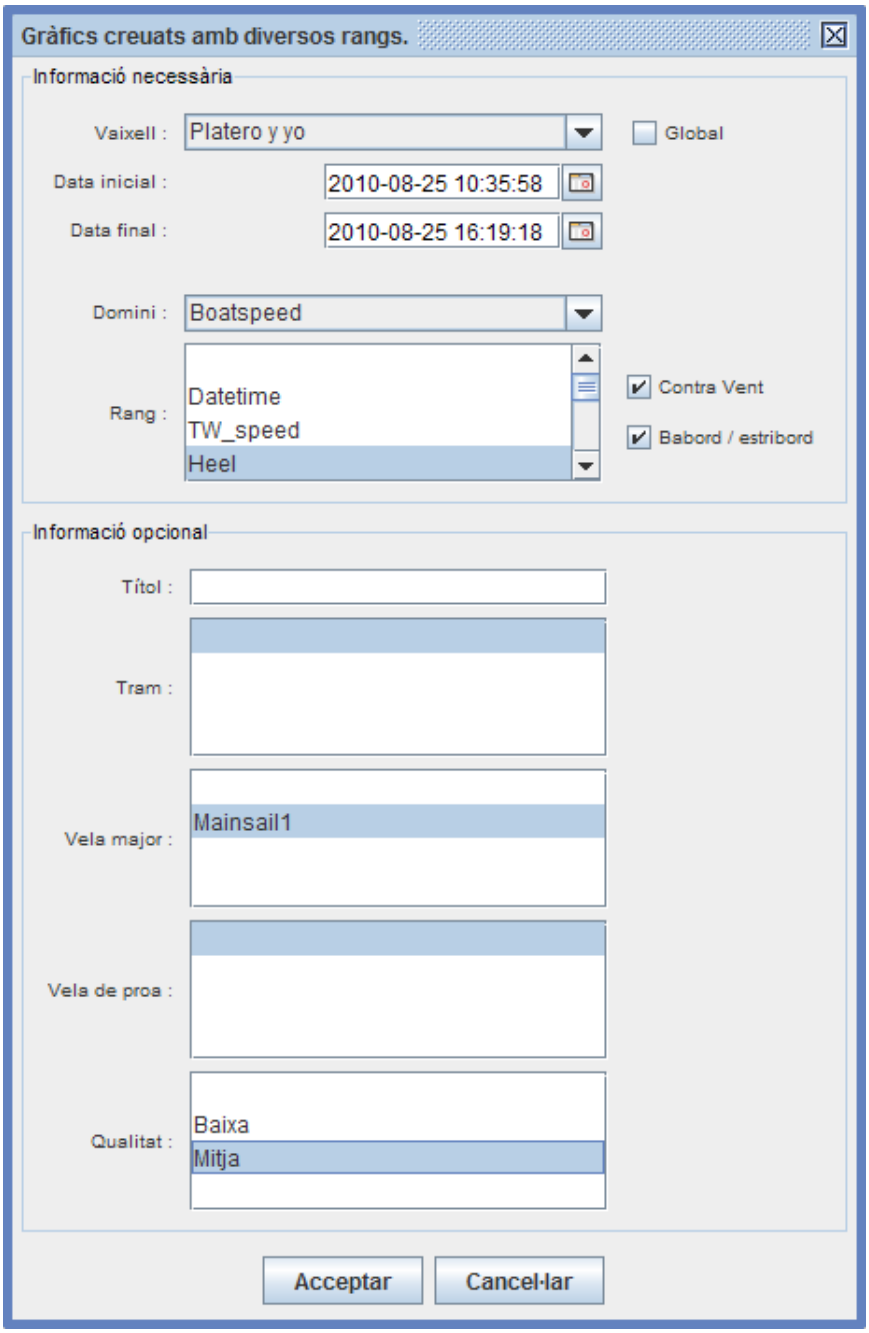


Figura 46: Petició d'una gràfica d'anàlisi en funció del TW_angle.

S'obté la següent gràfica:

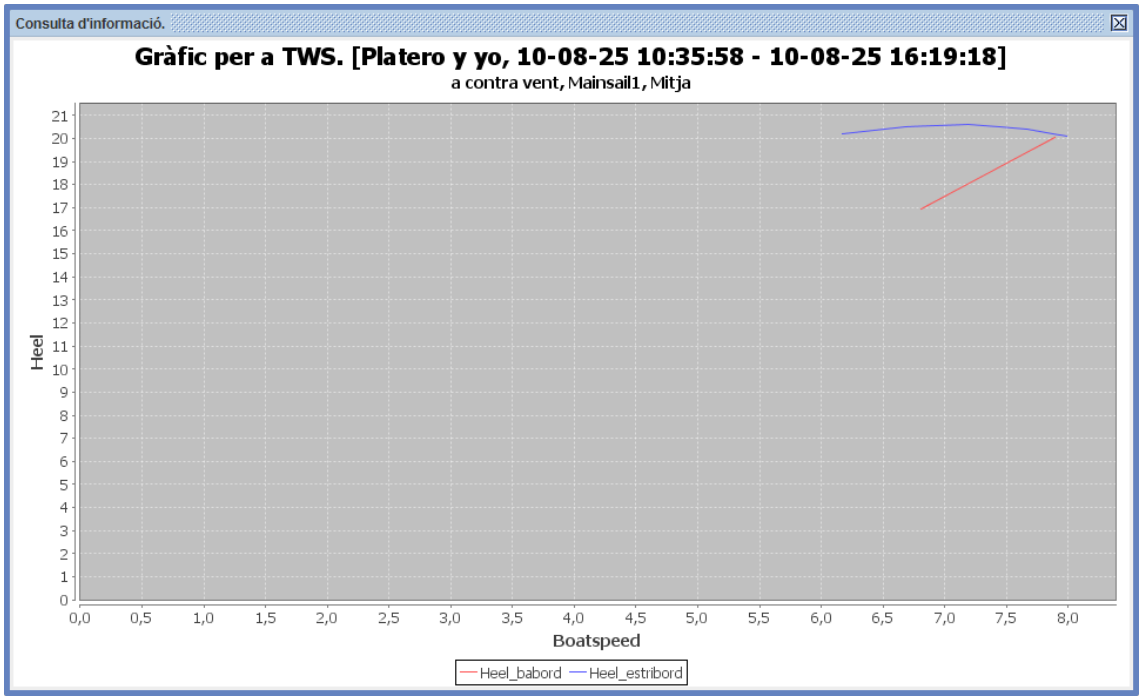


Figura 47: Gràfica d'anàlisi generada.

3.5.2.5 Gràfiques Babord / Estribord

Les gràfiques Babord / estribord permeten representar les variables del rang mostrant per una banda la informació que correspon a babord i per l'altra la informació que pertany a estribord. Aquestes gràfiques executen el *filtre TWSvsYBabordEstribord* per gràfiques d'anàlisi en funció del *TWS*, i el filtre *filtre XvsYBabordEstribord* per la resta.

Per exemple, la generació d'una gràfica cronològica indicant que es desitgen dades per babord i estribord:

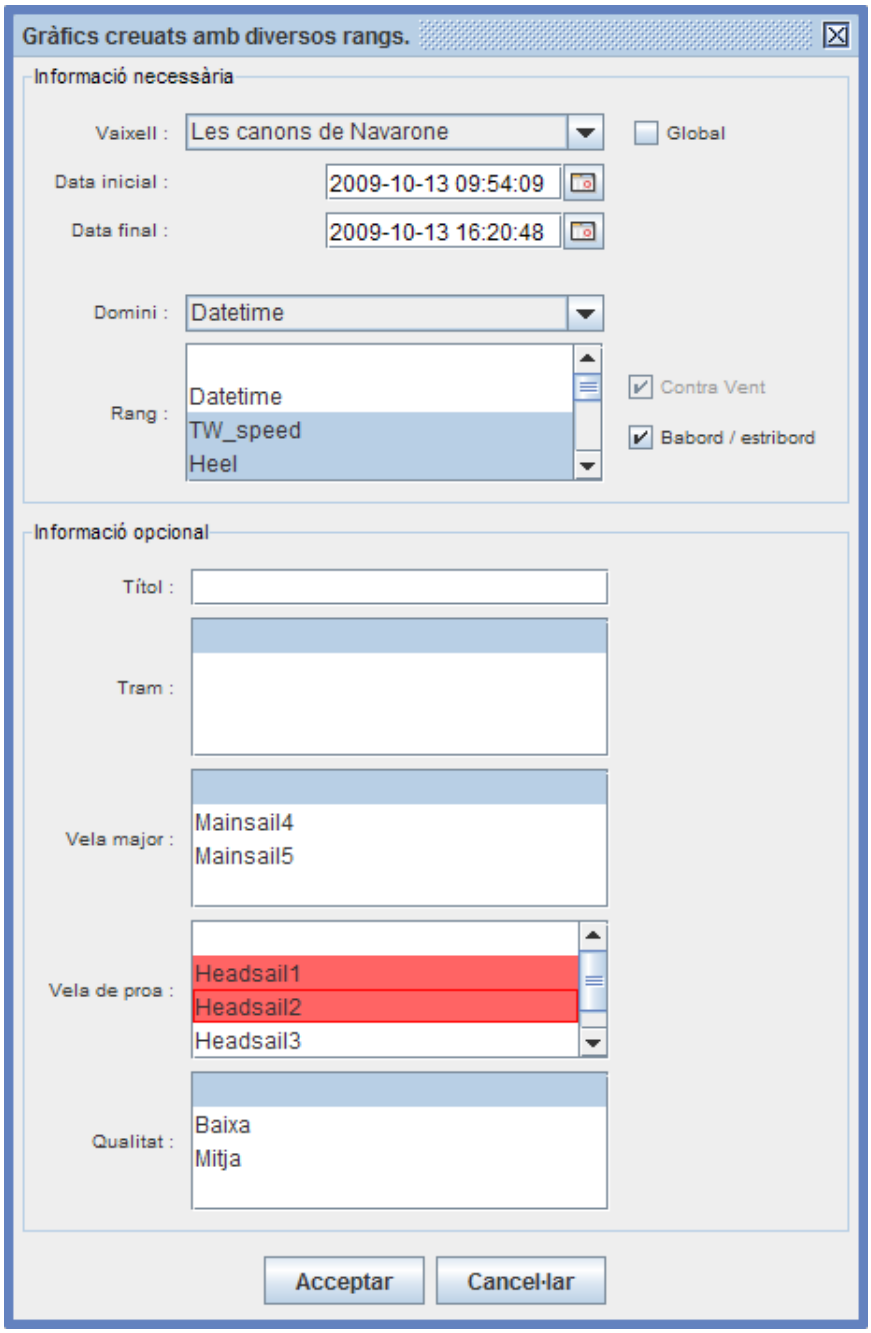


Figura 48: Petició d'una gràfica cronològica amb dades per babord i estribord.

S'obté la següent gràfica:

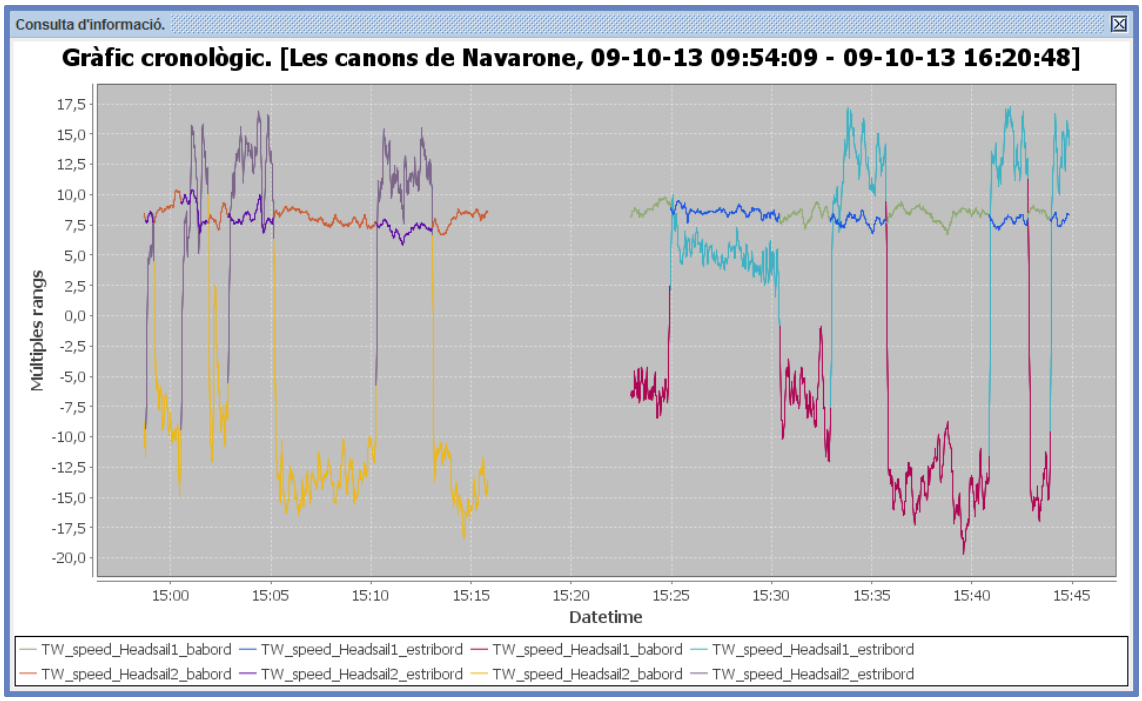


Figura 49: Gràfica cronològica amb dades per babord i estribord.

Un exemple diferent, la generació d'una gràfica d'anàlisi en funció del *TWS* indicant que es desitgen dades per babord i estribord:

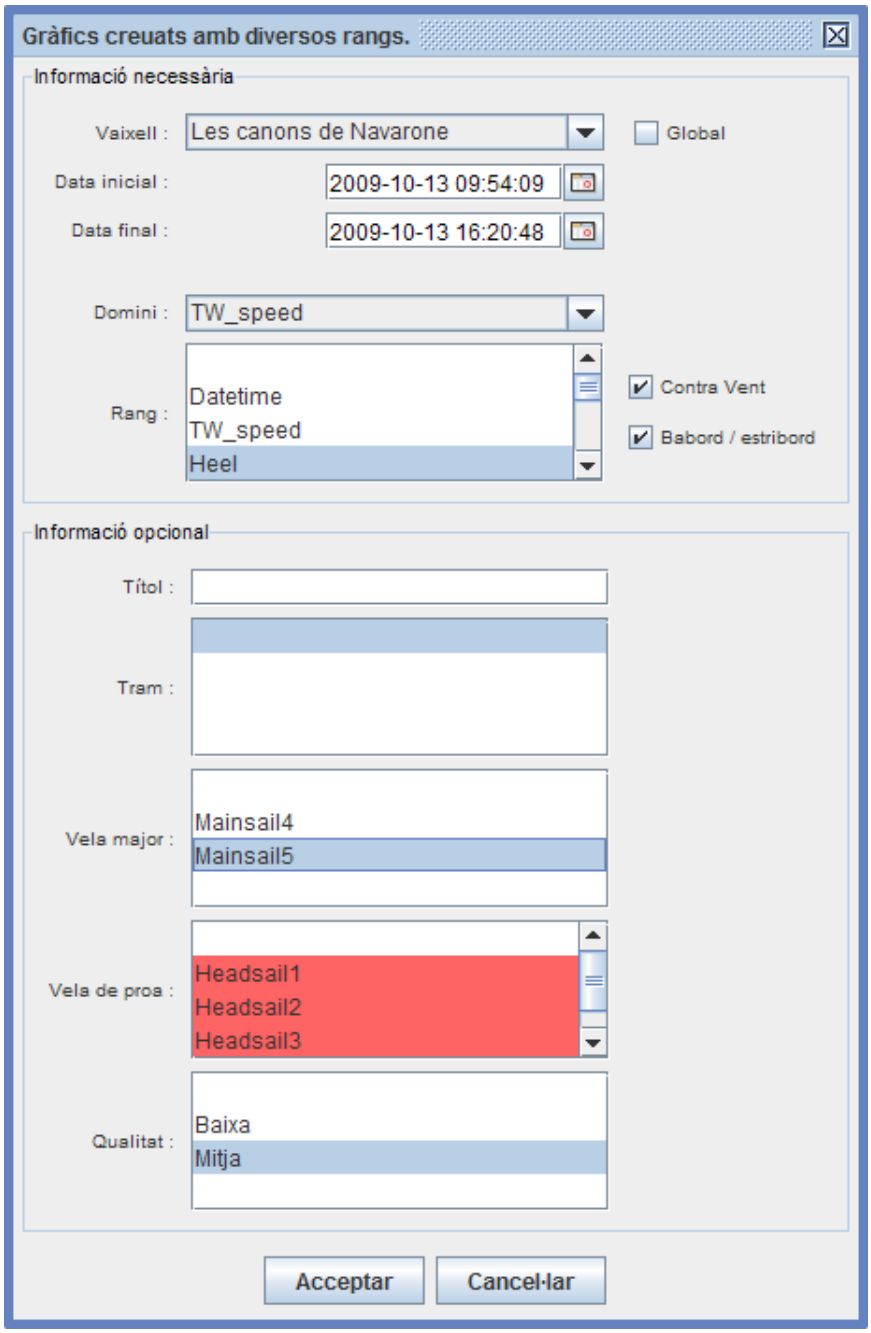


Figura 50: Petició d'una gràfica d'anàlisi en funció del *TWS* amb dades per babord i estribord.

S'obté la següent gràfica:

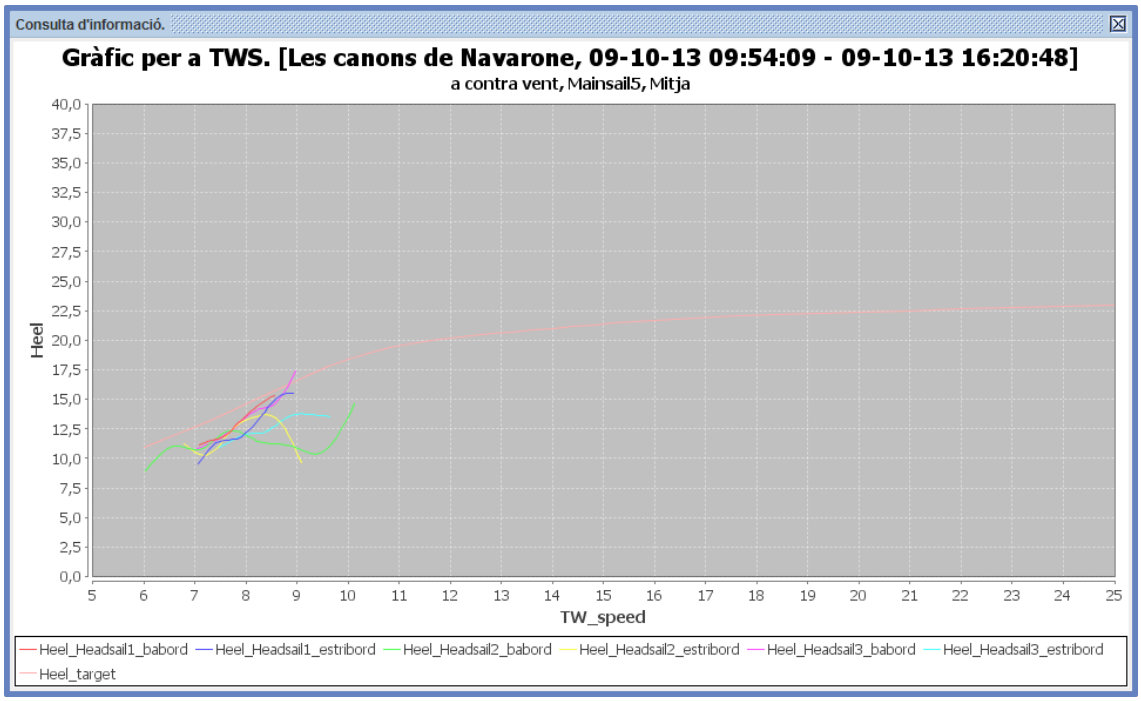


Figura 51: Gràfica d'anàlisi en funció del *TWS* amb dades per babord i estribord.

3.5.3 Altres funcionalitats comunes

La llibreria *JFreeChart* 1.0.12 ofereix funcionalitats addicionals a qualsevol de les gràfiques que generi el sistema. Prement el botó dret del ratolí el sistema mostra les següents funcionalitats:

1. *Propiedades*: Obre un nou panell que permet, modificar les propietats generals de la gràfica: títol, font, color de fons, aparença dels eixos, etc.
2. *Grabar como*: Permet gravar la gràfica com una imatge en format PNG.
3. *Imprimir*: Obre el panell d'impressió.
4. *Acercar*: Ofereix la possibilitat de realitzar un *zoom* a partir dels dos eixos de la gràfica, *zoom* en domini o *zoom* en rang.
5. *Alejar*: Ofereix la possibilitat d'allunyar el *zoom* actual a partir dels dos eixos de la gràfica, *zoom* en domini o *zoom* en rang.
6. *Escala automàtica*: Desfà el *zoom* actual completament bé per a ambdós eixos, només el domini o només el rang.

La següent figura mostra un exemple del pop up que apareix un cop premut el botó dret del ratolí sobre la gràfica:

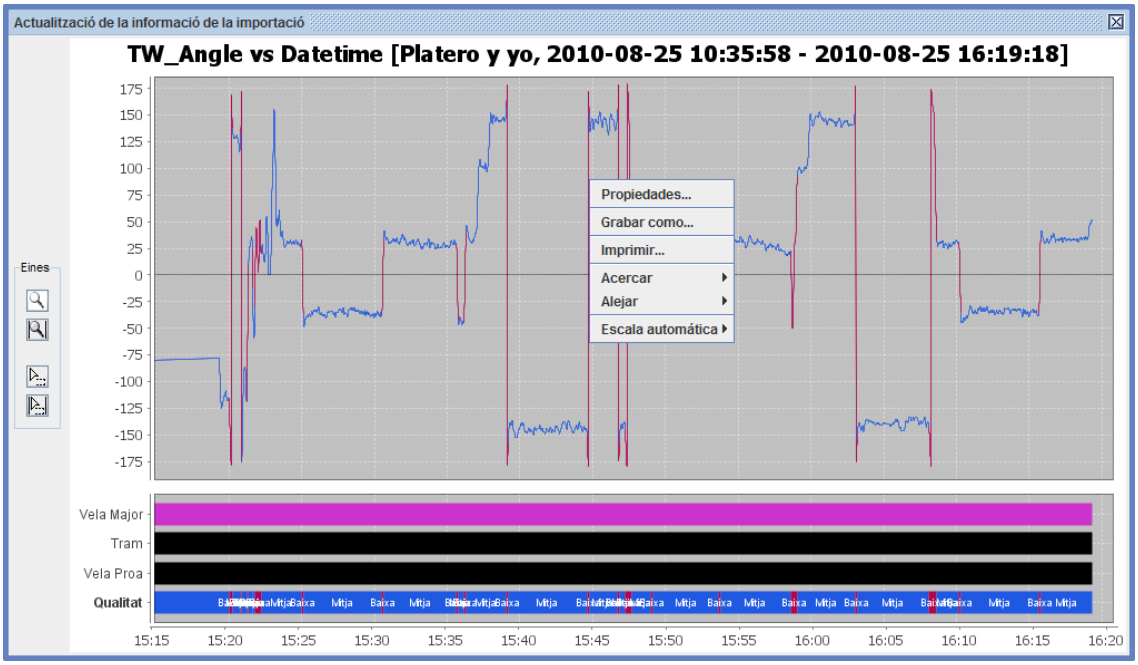


Figura 52: Altres funcionalitats.

3.6 Logs

La següent figura mostra els diaris existents al sistema:

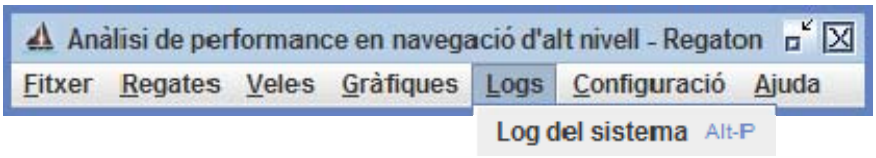


Figura 53: Diaris del sistema.

Funcionalitats del menú:

- *Log del sistema*: Diari del sistema que gestiona els avisos i errors produïts durant l'execució del sistema.

3.6.1 Diari del sistema

L'usuari pot accedir al diari del sistema a través de la vista principal sempre que ho desitgi. Per altra banda, els processos més complexes del sistema, com la importació de regates o l'execució de filtres també ofereixen a l'usuari la possibilitat de consultar al diari del sistema els errors produïts durant la seva execució.

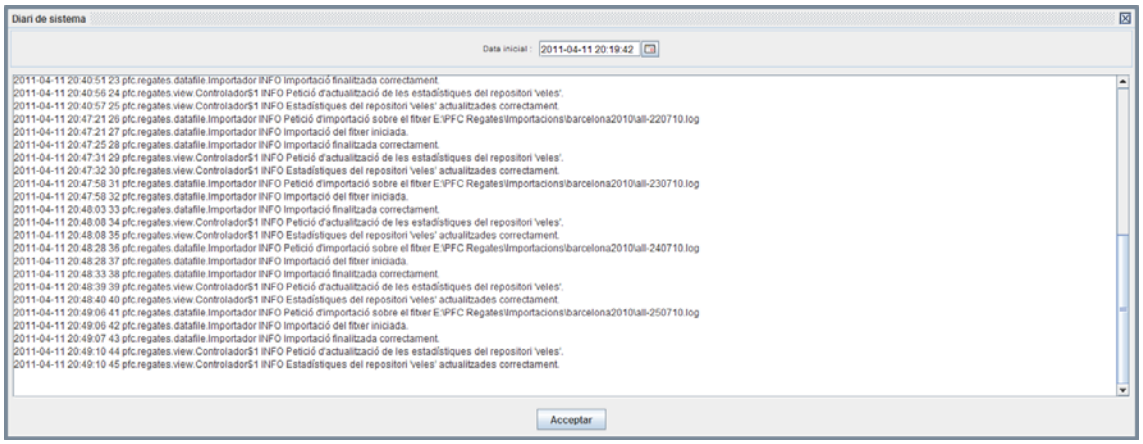


Figura 54: Diari del sistema.

Descripció dels camps de la vista

- *Data inicial*: Data a partir de la que es desitja consultar la informació del diari del sistema. L'usuari pot escriure la data al camp editable o bé prémer el botó de la seva dreta i introduir la data mitjançant el calendari.

3.7 Configuració

La següent figura mostra les funcionalitats de configuració del sistema:

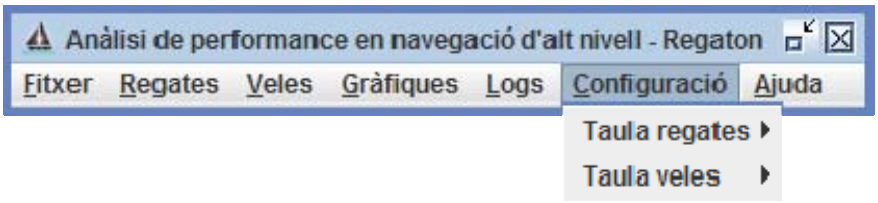


Figura 55: Configuració del sistema.

Funcionalitats del menú:

- *Taula de regates*: Funcionalitats de gestió de la taula SQL de regates.
- *Taula de veles*: Funcionalitats de gestió de la taula SQL de veles.

3.7.1 Taula de regates

Funcionalitats del sistema de gestió de la taula SQL de regates:

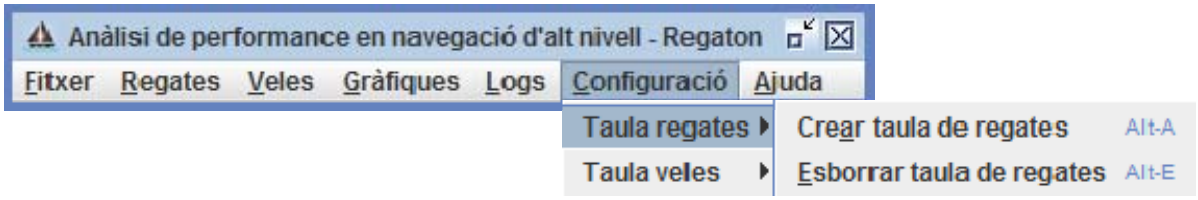


Figura 56: Funcionalitats del sistema de gestió de la taula de regates.

Funcionalitats del menú:

- *Crear taula de regates*: Creació de la taula SQL de regates.
- *Esborrar taula de regates*: Eliminació de la taula SQL de regates.

3.7.1.1 Crear la taula SQL regates

Funcionalitat que permet a l'usuari crear la taula SQL *regates* al sistema sempre que no existeixi.

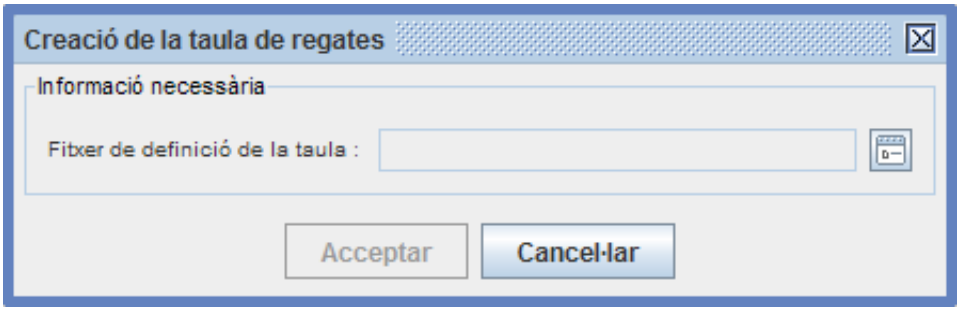


Figura 57: Diari del sistema.

Descripció dels camps de la vista

- *Fitxer de definició de la taula*: Fitxer de text pla del sistema on es defineix l'estructura bàsica de la taula SQL *regates*. Prement el botó de la dreta del camp no editable on es mostra el nom del fitxer s'obre una nova vista que permet recórrer l'arbre de directoris dels discs de l'ordinador.

3.7.1.1.1 Fitxer de definició de la taula regates

Fitxer de text pla del sistema on es defineix l'estructura bàsica de la taula SQL *regates*. Cada registre del fitxer defineix un dels camps de la taula.

El directori de la instal·lació del sistema on han de residir els fitxers de definició és el següent:

Projecte Regates\bin\taulesSQL

La figura següent mostra el fitxer bàsic de definició de la taula que ofereix el sistema:

```
################################################################################
# DEFINICIÓ DE LA TAULA SQL PER A REGATES                                      #
# ---------------------------------------                                      #
#                                                                              #
# Estructura del fitxer :                                                      #
#                                                                              #
#   Nom columna\tTipus columna\t\tParàmetre 1                                  #
#                                                                              #
#   Paràmetre 1 : PRI si es clau primària                                      #
#                                                                              #
# Requisits :                                                                  #
#   La taula regates ha de tenir com a clau primària Boat + Datetime.          #
#   Els separadors entre camps són tabuladors!                                 #
#   El caràcter # indica que el registre conté un comentari.                   #
#                                                                              #
################################################################################
#Vaixell.
Boat    VarChar(50) PRI

#Regata.
Race    VarChar(50)

#Vela major.
Mainsail    VarChar(50)

#Tram.
Leg VarChar(50)

#Vela de proa.
HeadSail    VarChar(50)

#Qualitat de les dades.
Quality VarChar(50)

#Comentari.
Comment VarChar(500)

#Posició latitud/altitud.
Position    VarChar(30)

#Data, hora, minut, segon.
Datetime    Datetime    PRI

#Velocitat del vent.
#TW_speed   Float(3,1)
TW_speed    Float

#Angle d'escorament.
#Heel   Float(3,1)
Heel    Float

#Angle d'escorament proporcionat pel fabricant.
#Ext_SOG    Float(4,2)
Ext_SOG Float

#Velocitat del vaixell.
#Boatspeed  Float(4,2)
Boatspeed   Float

#Velocitat del vaixell del fabricant.
#Vs_targ    Float(4,2)
Vs_targ Float

#Percentatge assolit del vaixell comparada amb la del fabricant.
Vs_targ%    SmallInt

#Angle del vent.
#Angles mínim i màxim per defecte : -180,180
TW_angle    Angle
```

Figura 58: Part del fitxer bàsic de definició del sistema.

3.7.1.2 Esborrar la taula SQL regates

Funcionalitat que permet a l'usuari eliminar la taula SQL *regates* i tota la seva informació del sistema. Una vegada esborrada la taula ja no és possible recuperar la seva informació.

3.7.2 Taula de veles

Funcionalitats del sistema de gestió de la taula SQL de veles:

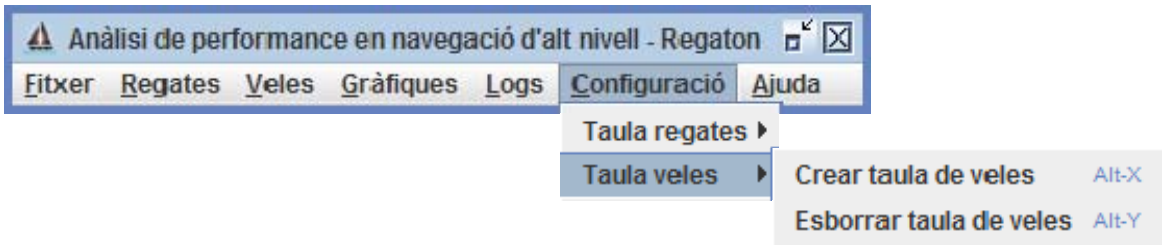


Figura 59: Funcionalitats del sistema de gestió de la taula de veles.

Funcionalitats del menú:

- *Crear taula de veles*: Creació de la taula SQL de veles.
- *Esborrar taula de veles*: Eliminació de la taula SQL de veles.

3.7.2.1 Crear la taula SQL veles

Funcionalitat que permet a l'usuari crear la taula SQL *veles* al sistema sempre que no existeixi.

3.7.2.2 Esborrar la taula SQL veles

Funcionalitat que permet a l'usuari eliminar la taula SQL *veles* i tota la seva informació del sistema. Una vegada esborrada la taula ja no és possible recuperar la seva informació.

3.8 Ajuda

La següent figura mostra la ajuda del sistema:

Figura 60: Ajuda del sistema.

L'ajuda del sistema conté els següents elements:

- *Manual d'usuari*: Manual de funcionament del sistema *Regaton*.
- *Sobre Regaton*: Descripció sobre la versió del sistema.

3.8.1 Manual d'usuari

El manual d'usuari consta d'una barra d'eines amb els següents botons:

- *Pàgina anterior*: Anar a la pàgina consultada prèviament a l'actual.
- *Pàgina següent*: Anar a la pàgina consultada posteriorment a l'actual.
- *Imprimir*: Panell d'impressió.
- *Configurar la pàgina d'impressió*: Panell de configuració de la pàgina d'impressió.

El marc esquerre ofereix l'arbre de temes del manual i la cerca de continguts.

El marc dret mostra la informació del tema seleccionat.

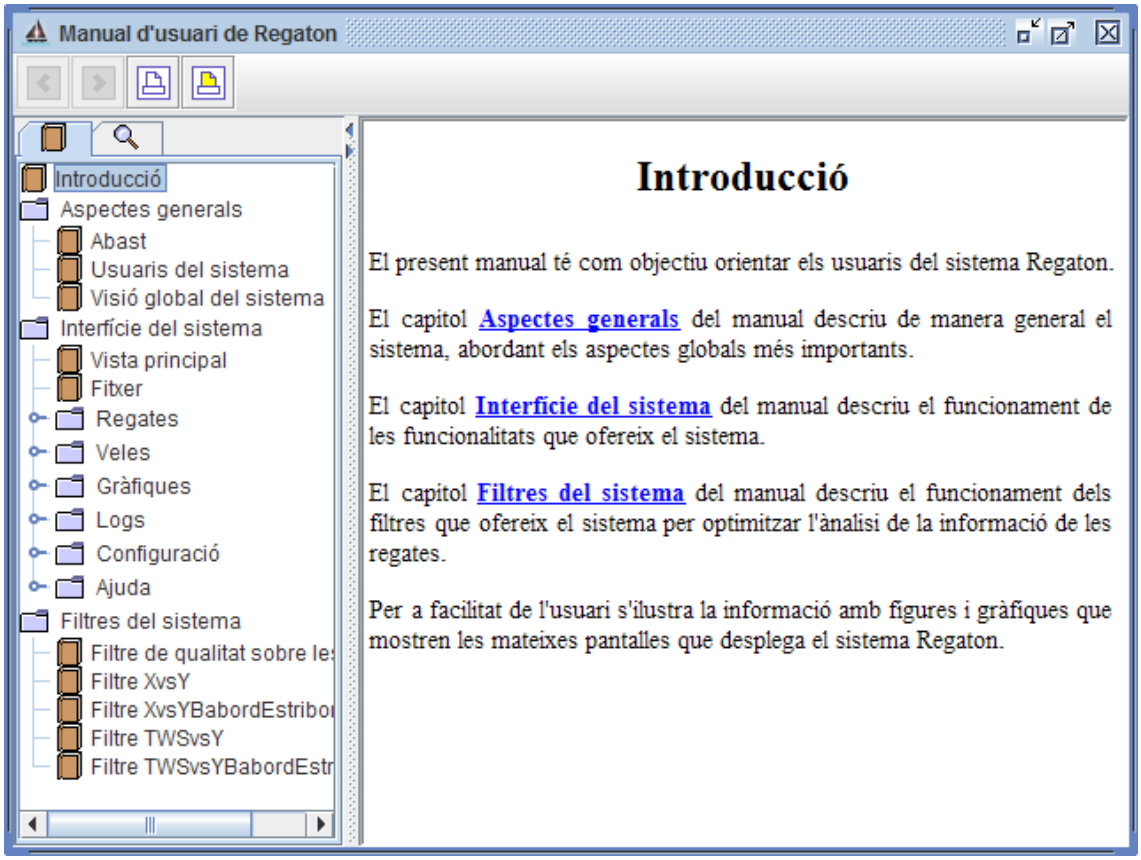


Figura 61: Manual d'usuari.

3.8.2 Sobre Regaton

Informació sobre la versió del sistema, data de creació i autors.

4 Filtres del sistema

1. Filtre de qualitat
2. Filtre XvsY
3. Filtre XvsYBarbordEstribord
4. Filtre TWSvsY
5. Filtre TWSvsYBarbordEstribord

4.1 Filtre de qualitat sobre les dades

Una vegada s'ha importat la informació de la regata el sistema pregunta a l'usuari si desitja executar el filtre de qualitat sobre les dades.

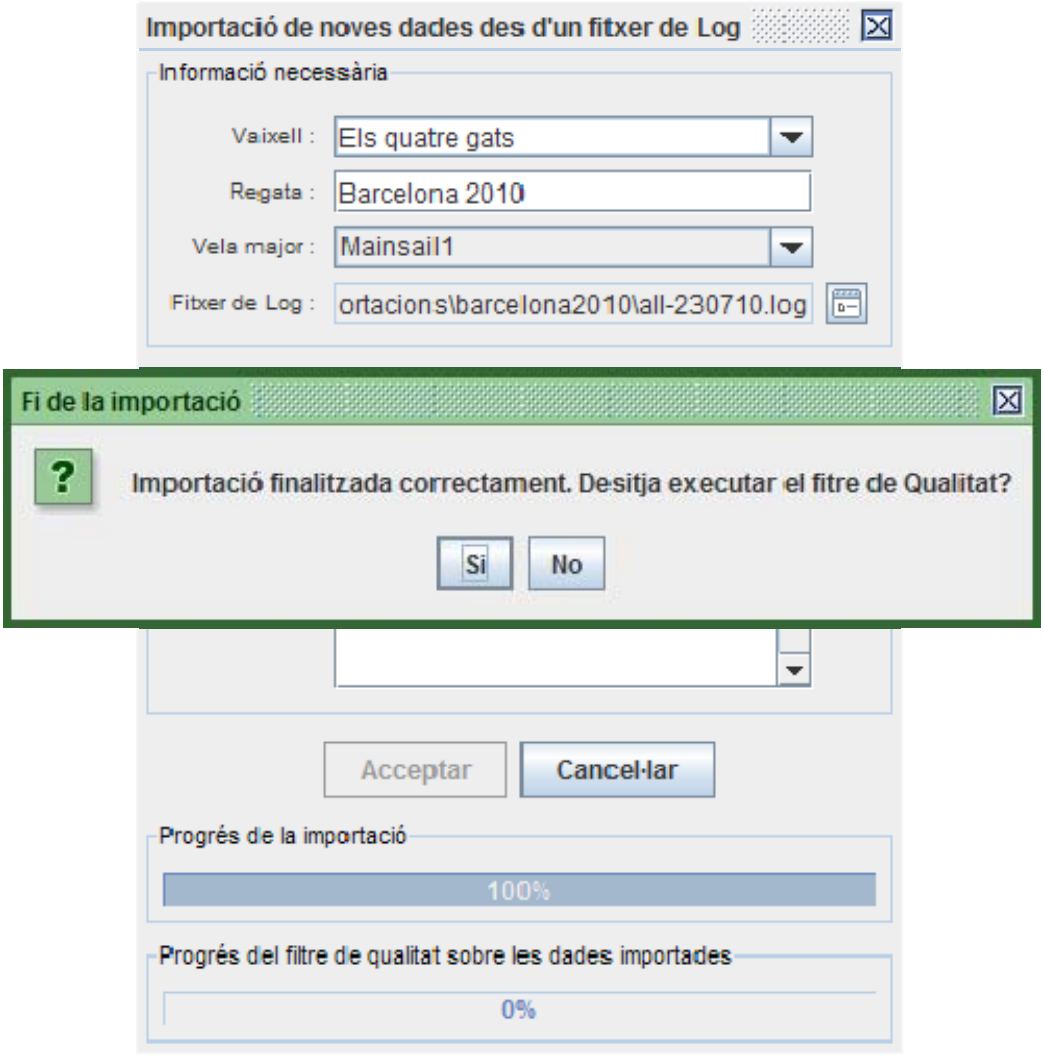


Figura 62: Quadre de diàleg preguntant a l'usuari si executar el filtre de qualitat.

Si l'usuari cancel·la l'execució del filtre es dóna la importació per finalitzada. L'aspecte de les dades importades de la regata, amb qualitat mitja, seria el següent:

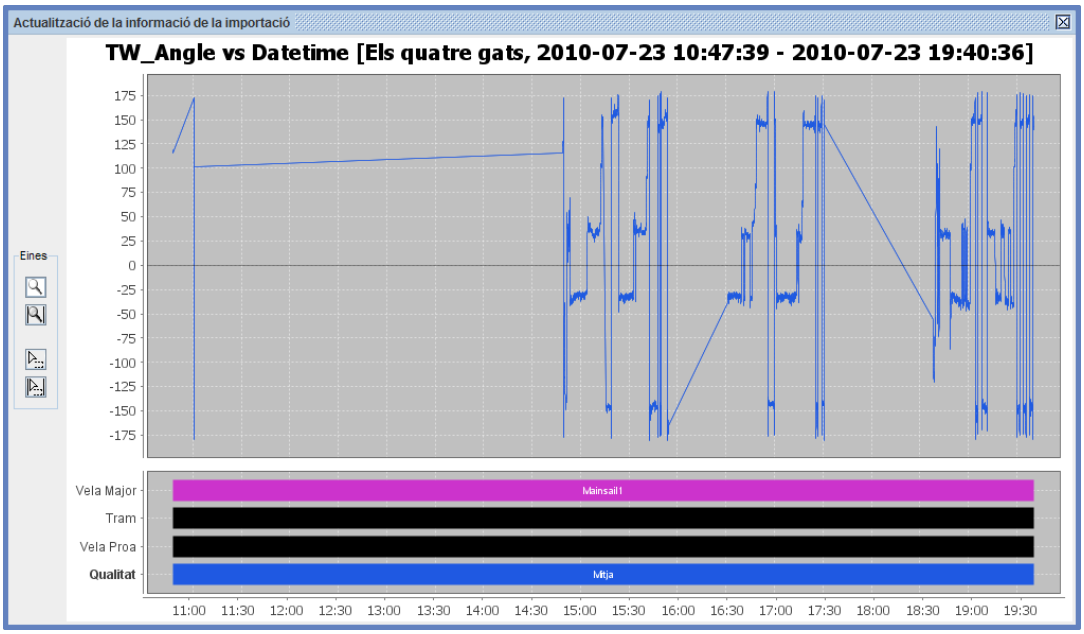


Figura 63: Vista TWAngle vs Time sense l'execució del filtre de qualitat.

Si l'usuari decideix executar el filtre el sistema inicia el procés.

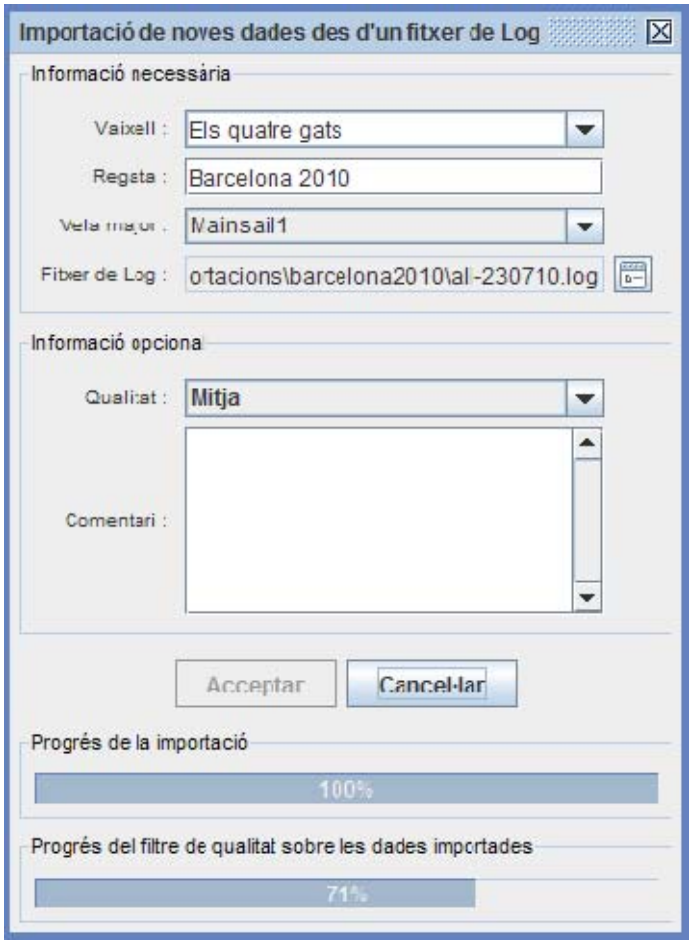


Figura 64: Exemple d'execució del filtre de qualitat.

Finalitzada l'execució del filtre, l'aspecte de les dades importades de la regata, amb qualitat mitja, seria el següent:

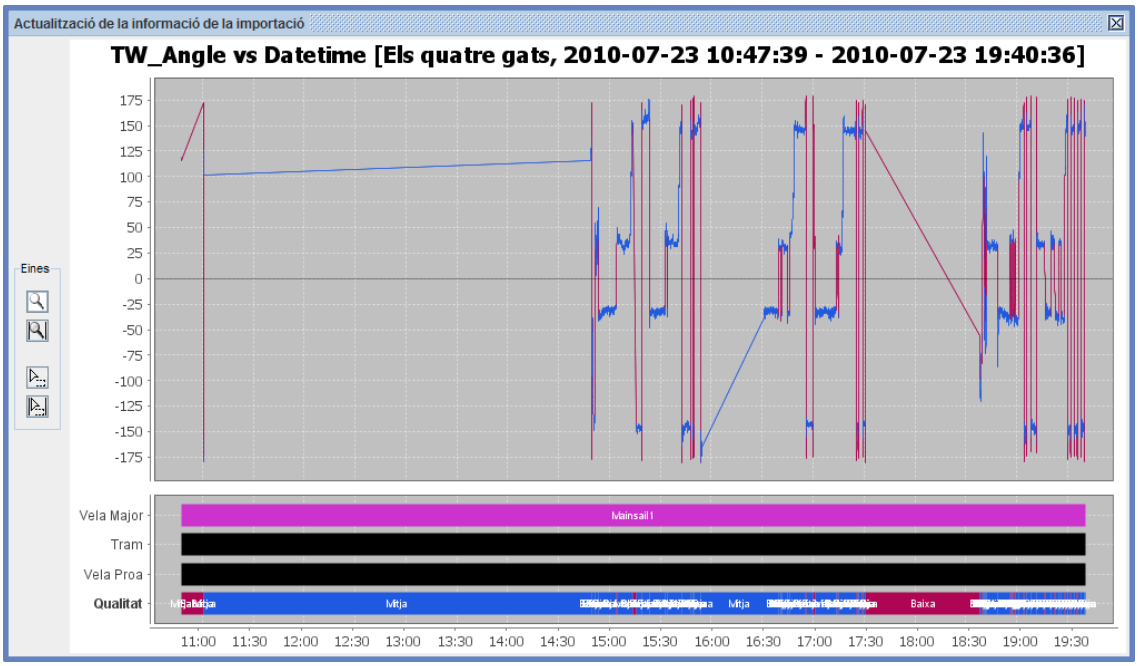


Figura 65: Vista TWAngle vs Time finalitzada l'execució del filtre de qualitat.

Descripció del procés del filtre de qualitat sobre les dades

Per un bon anàlisi de les dades d'una regata és necessari descartar les dades de baixa qualitat. Els instants que duren les maniobres de virada i trabujada dels vaixells de regata es consideren de mala qualitat degut a que són instants en que el vaixell perd considerablement la velocitat.

L'objectiu del filtre de qualitat sobre les dades és assignar qualitat *Baixa* als instants immediatament anteriors i posteriors a cada maniobra del vaixell. Això s'aconsegueix detectant els canvis de signe en la variable *TW_angle*.

El procés del filtre passa per les següents etapes:

1. Recorregut inicial de les dades actualitzant la qualitat dels instants anteriors a cada canvi de signe de la variable *TW_angle*.
2. Recorregut de les dades actualitzant la qualitat dels instants posteriors a cada canvi del signe de la variable *TW_angle*.

4.2 Filtre X vs Y

Filtre del sistema que obté el conjunt de dades a mostrar a partir del domini i dels rangs seleccionats per l'usuari. Aquest filtre no realitza cap acció addicional sobre el conjunt de dades que finalment mostra la gràfica.

4.3 Filtre X vs Y per babord i estribord

Filtre del sistema que obté el conjunt de dades a mostrar a partir del domini i dels rangs seleccionats per l'usuari. La sentència SQL que construeix el filtre indica al sistema que ha de colorar les dades obtingudes amb colors diferents per les dades de babord i les d'estribord.

4.4 Filtre TWS vs Y

Filtre del sistema que obté el conjunt de dades a mostrar utilitzant la variable TW_speed com a domini i qualsevol altra variable d'estudi com a rang.

L'objectiu del filtre és mostrar el valor mig que la variable del rang pren per cada valor de referència de la variable TW_speed.

Passos del filtre, previs a la representació de la informació en la gràfica:

1. Obtenció del conjunt de dades a tractar. Només s'obtenen les dades *a contra vent* o *a favor del vent* en funció del que hagi seleccionat l'usuari.
2. Obtenció del interval de valors representables del rang en funció de la variable seleccionada.
3. Obtenció dels valors absoluts del rang. Tots els càlculs posteriors es realitzen a partir dels valors absoluts.
4. Obtenció del valor mínim i màxim del domini.
5. Es divideix el domini en N intervals. La versió actual crea 10 intervals.
6. Càlcul del nombre mínim de valors del rang que han de caure dins d'un interval per tenir-lo en compte en els càlculs. Per tenir en compte un interval, la versió actual requereix que almenys un 3% dels valors del rang hi pertanyin.
7. Càlcul de la mitja dels valors del rang que pertanyen a cadascun dels intervals. Els intervals dels que el nombre de valors del rang no superi el valor calculat en el pas anterior són descartats del càlcul.
8. Càlcul de la mitja dels valors del domini que pertanyen a cadascun dels intervals.
9. Les parelles de mitges *(mitja domini, mitja rang)* calculades per cada interval són els punts que formen el conjunt de dades a representar en la gràfica.
10. Si existeixen les dades *target* del fabricant s'afegeixen a la gràfica.

IMPORTANT!: El filtre admet qualsevol variable en el domini, no només el *TWS*; Però en la versió actual, per la resta de variables el procés obvia la lectura de *fitxers de targets*.

4.5 Filtre TWS vs Y per babord i estribord

Filtre del sistema que obté el conjunt de dades a mostrar utilitzant la variable del domini i qualsevol altra variable d'estudi com a rang.

El filtre s'encarrega d'executar el *filtre TWSvsY* per les dades de babord per una banda i per les dades d'estribord per l'altra.

13 Apèndix C: Estudi de les tecnologies disponibles

L'estudi de les tecnologies disponibles ens ajuda a conèixer quines s'adapten millor als requeriments del sistema. Avaluant els pros i els contres, i realitzant una comparativa dels llenguatges de programació més reconeguts i actuals, tot i que no és exhaustiva, tindrem més informació per decidir com desenvolupar el sistema.

13.1 Llenguatges de programació

Considerem l'existència de dues categories generals de llenguatges de programació, els llenguatges *Turing-complets* i els llenguatges dinàmics.

Un llenguatge Turing-complet és un sistema capaç de computar tota funció *Turing-computable*[1]. Mentre que els sistemes Turing-complets són físicament impossibles, degut a que necessiten emmagatzematge infinit, el concepte de Turing-complet acostuma a relaxar-se per referir-se a màquines físiques o llenguatges de programació que podrien ser universals si aconseguissin emmagatzematge infinit.

Llenguatges de propòsit general:

1. Procedurals: *Pascal*, etc.
2. Orientats a objectes: *Java*, etc.
3. Multiparadigma: *Ada*, *C++*, etc.

Llenguatges de paradigmes menys comuns:

1. Funcionals: *Lisp*, *Haskell*, etc.
2. Lògics: *Prolog*, etc.
3. Declaratius: X*SLT*, etc.

Els llenguatges dinàmics o de *Scripting* són programes que permeten el control d'una o certes aplicacions.

Els *scripts* es tracten de forma diferent als programes. S'executen independentment de qualsevol altra aplicació. Acostumen a ser diferents del nucli de l'aplicació que acostuma a estar programada en un llenguatge diferent. Quan són accessibles a l'usuari permeten tractar el comportament de l'aplicació adaptant-lo a les seves necessitats.

1. Llenguatges de control d'eines (*JCL*) i shells: *JCL*, *Bash*, etc.
2. Llenguatges específics per a una aplicació: *QuakeC*, etc.

[1] Una funció Turing-computable si el seu valor pot ser computat per una màquina de Turing. Una màquina de Turing és una màquina imaginària inventada per Alan Turing per descriure el significat de computar alguna cosa. La descripció física d'una màquina de Turing és una caixa amb una cinta i amb un capçal. La cinta consta d'un nombre infinit de cel·les en les dues direccions, amb el capçal posicionat sempre en exactament una d'aquestes cel·les. Cada cel·la conté un dels símbols d'una relació finita escrit. La màquina té un conjunt d'estats finit i amb cada moviment canvia el seu estat, canvia el símbol escrit en la cel·la actual i desplaça una posició cap a la dreta o l'esquerra el capçal. La màquina s'atura quan assoleix una combinació de símbol i estat per la que no s'ha definit un moviment. Existeix un estat inicial en el que la màquina inicia la seva computació. Per més informació accedir a http://ca.wikipedia.org/wiki/Màquina_de_Turing

3. Navegadors Web: *Javascript*, *VBScript*, etc.
4. Servidors Web: *PHP*, *ASP*, *JSP*, *ColdFusion*, *Ruby*, *Python*, *CGI*, *Servlets*, etc.
5. Processament de text: *AWK*, *grep*, etc.
6. Llenguatges dinàmics de propòsit general: *Perl*, etc.

La següent taula mostra les diferències entre llenguatges Turing-complets i llenguatges dinàmics.

Turing- complets	Dinàmics
Tipificats estàticament	Tipificats dinàmicament
Compilats (normalment)	Interpretats (normalment)
Execució més ràpida	Acostumen a ser més lents
Més difícils de programar. Codi possiblement més difícil de comprendre.	Codi més entenedor i senzill. Millor productivitat.
Detecció d'errors en temps de compilació	Els errors apareixen en temps d'execució

Taula 6: Diferències entre llenguatges Turing - complets i dinàmics

13.1.1 Característiques dels llenguatges de programació actuals més comuns

A continuació és mostra una relació dels llenguatges més comuns en l'actualitat explicant els seus avantatges i inconvenients.

13.1.1.1 Llenguatge C

Avantatges

1. Llenguatge molt eficient gràcies a que es poden usar les seves característiques de baix nivell per realitzar implementacions òptimes.
2. Tot i el seu baix nivell, és el llenguatge més portat dels existents, amb compiladors per quasi tots els sistemes actuals.
3. Proporciona facilitats per realitzar programes modulars i/o utilitzar codi o llibreries al seu abast.
4. Les aplicacions finalitzades utilitzen millor els recursos hardware.
5. Compilació nativa. Aplicacions més ràpides.

Inconvenients

1. Velocitat de desenvolupament lenta.
2. Manteniment difícil i costós.
3. Manca de controls de gestió de memòria i entrada de dades.

13.1.1.2 Llenguatge C++

Avantatges

1. Compilació nativa. Aplicacions més ràpides.
2. És una extensió del *C*.
3. Permet control de la memòria i programació en baix nivell no possible en *Java*.
4. És robust.

Inconvenients

1. No és multiplataforma.
2. No presenta una arquitectura estàndard pel desenvolupament orientat a Internet.
3. És una extensió del *C*. Això provoca que certs dogmes de l'orientació a objectes es sacrifiquin en favor de les característiques pròpies del *C*.
4. És difícil d'aprendre.

13.1.1.3 Llenguatge Java

Avantatges

1. La compilació genera codi objecte.
2. És interpretat. Necessita un intèrpret (*JVM, Java Virtual Machine*).
3. Portable a tots els sistemes que disposin del seu intèrpret.
4. Més simple que el *C++*.
5. Usa assignació automàtica de memòria i gestor de brossa.
6. Orientat a objectes.
7. L'orientació a objectes aporta més flexibilitat, disseny modular i reutilització.
8. És distribuït.
9. Considera la seguretat com part del seu disseny.
10. És robust.
11. Permet múltiples fluxos.

Inconvenients

1. És interpretat. Les aplicacions desenvolupades en *Java* són més lentes que les generades amb llenguatges que compilen codi natiu a la plataforma.

13.1.1.4 CGI (Common Gateway Interface, o programes d'interfície comú de passarel·la)

Fins fa alguns anys els programes *CGI* eren els únics que permetien executar codi en els servidors web i capturar informació enviada des dels seus clients.

Avantatges

1. 100% portables.

2. Si s'escriuen en *C, C++* consumeixen menys recursos.
3. Es poden programar en diversos llenguatges de programació com Perl, C++, C, etc.

Inconvenients

1. Es carreguen a memòria en cada petició realitzada per l'usuari.
2. La comunicació entre dos aplicacions *CGI* s'ha de realitzar a través de fitxers de E/S, convertint-se potencialment en colls d'ampolla.
3. Si una aplicació *CGI* accedeix a un *SGBD*, aquest també es carrega cada vegada que es crida l'aplicació *CGI*.

13.1.1.5 Servlets

Els servlets substitueixen als *CGI*. Un servlet és una aplicació *Java* que s'emmagatzema i executa en un servidor *J2EE* compatible que pot connectar-se amb un servidor diferent.

Avantatges

1. La *JVM* es manté arrencada, i permet que cada petició es tracti en un flux (*thread*) i no en un procés com en els *CGI*. Per N peticions simultànies d'un *CGI*, es carregaran N processos mentre que amb *Java* només s'usaran N fluxos, i només existirà un còpia del *servlet*.
2. Suport important al tractament de capçaleres *HTTP* i tractament de dades de formularis.
3. Poden compartir dades entre ells.
4. Poden mantenir informació entre diverses sol·licituds.
5. Són portables.
6. Els clients que realitzen peticions no necessiten tenir instal·lat el *J2SE* ni el *JRE* ja que la resposta acostuma a codificar-se en *HTML*.

Inconvenients

1. Es programen en *Java* i seguint l'especificació dels servlets definida per Sun.
2. Els *servlets* sempre es mantenen en memòria (tot i que per cadascun només n'hi hagi una còpia). Si no s'utilitzen representen una despesa de recursos innecessària.
3. El servidor web necessita la *JVM* per poder executar *servlets*.
4. El procés de proves pels *servlets* no és tan senzill com en els *CGI*.

13.1.1.6 JSP (Java Server Pages)

Avantatges

1. Permeten separar la codificació del format del web (*HTML*) mantenint el contingut en els *servlets*.
2. Preparades per treballar amb *Beans*.

Inconvenients

1. La depuració és difícil, doncs els programes *JSP*, es converteixen en classes *Java* (que són alhora *servlets*) i després es compilen. Els errors que puguin aparèixer apunten al codi de les classes *Java* i no als programes *JSP*.
2. Existeixen motors per a *servlets* que ofereixen plantilles per una millor programació, però escollir el més adequat no és senzill.
3. Utilitzen aproximadament el doble d'espai en disc. Degut a la compilació dels programes *JSP*, el servidor ha d'emmagatzemar conjuntament aquests amb les classes compilades.
4. Cada compilació dels programes *JSP* requereix l'aturada del servidor web per introduir les noves classes.

13.1.1.7 PHP (Personal Home Pages)

Avantatges

1. Portable.
2. Expansible, doncs existeixen un nombre elevat de mòduls.
3. Interfícies diferents per a cada tipus de servidor (*Apache*, *IIS*, ...).
4. Gran quantitat de mòduls disponibles per als programadors (tractament de gràfiques, *PDF*, *XML*, *Flash*, ...).
5. Escrit completament en *C*. Això i el fet que sigui un mòdul de l'*Apache* en converteixen en un llenguatge ràpid i de poc ús de memòria.
6. Actualment dóna suport a l'orientació a objectes.
7. Té tractament d'excepcions.
8. No obliga a seguir cap metodologia de programació.
9. Té suport natiu per part de *MySQL* fins la versió 4.X.

Inconvenients

1. El tractament d'excepcions no és tan sofisticat com el d'*ASP*.
2. No obliga a seguir cap metodologia de programació. Pot derivar en codi desordenat i de difícil manteniment.
3. Difícil d'optimitzar.
4. Disseny del llenguatge pobre, doncs en cada versió apareixen canvis en les especificacions.
5. El suport a l'orientació a objectes s'ha de millorar.
6. No ajuda a la programació modular.

13.1.1.8 ASP (Active Server Pages)

Avantatges

1. Les pàgines s'executen dins del servidor web, és a dir, que no cerquen recursos fora del servidor com fan els *CGI*.
2. És el sistema més optimitzat per a l'ús de *IIS* i *Windows*.

3. Llenguatge de ràpida programació similar al *Visual Basic*.
4. Tecnologia actualitzada de forma regular.

Inconvenients

1. No és lliure.
2. No és portable (només funcionen sobre *IIS* de *Windows*).
3. Metodologia poc robusta.

13.1.1.9 Ruby

Avantatges

1. Orientació a objectes.
2. Minimització del nombre de línies de codi. Orientat a obtenir la màxima productivitat.
3. Ús important de força convencions en la programació per evitar detalls complicats de configuració, etc.
4. Desenvolupament més senzill que en *Java*.

Inconvenients

1. Llenguatge emergent amb pocs anys d'experiència.
2. Possible manca de documentació.
3. La seva escalabilitat per a grans projectes no està demostrada.
4. Nombre reduït de llibreries de suport.
5. Els fluxos no són natius.

13.1.1.10 Python

Avantatges

1. És de propòsit general.
2. Sintaxis molt visual, amb notació indentada obligatòria.
3. Empreses de gran renom l'utilitzen *Yahoo, Google, Red Hat, ...*
4. Escala millor que el *Perl*.
5. Tractament d'excepcions.
6. Portable.
7. Fàcil de programar i mantenir.

Inconvenients

1. Documentació limitada.

13.1.1.11 Perl (Porcessing Estraction Report Language)

Avantatges

1. És de propòsit general.
2. Permet treballar amb aplicacions pròpies del sistema operatiu.
3. S'usa en *scripts* de tipus *CGI*.
4. És robust.
5. Molts anys d'experiència.
6. Més complicat d'aprendre que la resta de llenguatges dinàmics descrits. Però alhora té moltes més funcionalitats.
7. És un llenguatge més potent que *PHP*.
8. No obliga a seguir cap metodologia de programació.

Inconvenients

1. La seva plataforma natural és *Unix*.
2. En la programació web s'usa en *scripts* de tipus *CGI*.
3. En aplicacions de baix nivell és lent, tot i que pot interactuar amb C per agilitzar l'execució.
4. No obliga a seguir cap metodologia de programació. Pot derivar en codi desordenat i de difícil manteniment.

13.1.2 Taules comparatives dels llenguatges de programació estudiats

	C	C++	Java
Empresa	AT&T	AT&T	Sun Microsystems
Any d'aparició	1969	1985	1995
Lliure	✓	✓	✗ GPL
Codi obert	✗	✗	✗
Interpretat	✗	✗	✓
OOP	✗ Procedural imperatiu	✓	✓
Compatibilitat amb SGBDs	⇑⇑	⇑⇑	⇑⇑
Suport de MySQL	✓	✓	✓
Facilitat d'aprendre	↓	↓	↓
Reusabilitat	↓	✓	✓
Portabilitat	✓ Existeix per quasi totes les arquitectures	✓	✓ Sempre que la plataforma posseeixi la MV
Extensibilitat	↓	⇑⇑	⇑⇑
Manteniment	↓	⇑	⇑
Flexibilitat	✓	✓	✓
Eficiència	⇑⇑	⇑	↓
Tipificació	Estàtica	Estàtica	Estàtica
Llibreries lliures amb suport per a gràfiques	GDChart ProEssentials de Gigasoft	ProEssentials de Gigasoft SoftIntegration Graphical Library Chart Toolkit de INT	JFreeChart

Taula 7: Comparativa de llenguatges de programació Turing - complets

	ASP	PHP	JSP/Java	Ruby (on Rails)	Python	JavaScript	Perl
Empresa	Microsoft	PHP Group	Sun Microsystems	Yukihiro Matsumoto	Python Software Foundation	Netscape/Sun	Larry Wall
Any d'aparició		1995	1995	1995	1991	1995	1987
Lliure	Sota llicència propietària	Llicència PHP	GPL	Llicència Ruby/GPL	Python Software Fundation License		GPL (General Public License)
Codi obert							
Llenguatge que utilitza	VBScript o Jscript (llenguatges d'script)	PHP (llenguatge d'script procedural, orientat a objectes)	Java (més potent que els llenguatges d'script) Les JSP es compilen en Servlets que permeten usar tota la potència de Java	Ruby (llenguatge d'script orientat a objectes)	Python (llenguatge d'script orientat a objectes, estructurat i funcional interpretat i interactiu)	Javascript (llenguatge d'script orientat a objectes, funcional i procedural)	Perl (multiparadigma)
Interpretat	En el servidor	Cada vegada que es carrega En el servidor	Els servlets es carreguen a memòria una sola vegada En el servidor (compilació prèvia manual)	Cada vegada que es carrega En el servidor	Cada vegada que es carrega En el servidor	Cada vegada que es carrega En el client	Cada vegada que es carrega En el servidor
OOP	Funciona amb COM						
Independència servidor	Només de Microsoft (IIS i Personal Web Server)	Tots els més populars	Tots els més populars	Apache	Apache, lighttpd		
Compatibilitat amb SGBDs							
Suport de MySQL							
Facilitat d'aprendre							
Tags	Combina tags i scripts		Combina tags i scripts i permet crear nous tags				
Reusabilitat							
Portabilitat		Sempre que la plataforma posseeixi la MV	Sempre que la plataforma posseeixi la MV	Sempre que la plataforma posseeixi la MV	Sempre que la plataforma posseeixi la MV		
Extensibilitat		(sense oo)					(sense oo)

		(amb oo)					(amb oo)
Manteniment							
Flexibilitat			És estructurat				
Tipificació	Dinàmica	Dinàmica	Estàtica	Dinàmica	Dinàmica	Dinàmica	Dinàmica
Frameworks (MVC)	ASP Xtreme Evolution (AXE)	Frameworks FUSE, CakePHP, altres	Frameworks Struts, JSF, altres	Ruby on Rails, altres	Django, Pylon, altres	SproutCore, Woozoo, altres	Catalyst, altres
Llibreries lliures amb suport per a gràfiques	WebChart	JpGraph	JFreeChart	Gruff o Sparklines però tenen certes limitacions. JFreeChart utilitant JRuby	Matplotlib. JFreeChart utilitzant JPython	JFreeChart	GDGraph

Taula 8: Comparativa de llenguatges dinàmics

13.1.3 Conclusions de l'estudi

En funció de si el sistema ha de ser executable o es pot executar en altres aplicacions externes com navegadors ens decidirem per llenguatges complets o dinàmics respectivament.

Entre el primer grup de llenguatges *C* i *C++* es presenten com la millor opció per desenvolupar aplicacions on el rendiment i la velocitat d'execució són primordials. Per altra banda, *Java* és el que millor s'adapta dels tres a les noves tecnologies relacionades amb Internet.

Tant *C++* com *Java* són orientats a objectes, paradigma que aporta extensibilitat i reusabilitat mentre el llenguatge *C* és procedural i de manteniment més difícil.

Java permet construir aplicacions multiplataforma, gràcies a que és la *JVM* qui la interpreta.

El segon grup de llenguatges és més indicat per a petits i mitjans projectes on la velocitat no és un requisit i la programació no és complexa. També són indicats per a projectes orientats al web. *Perl*, és l'únic llenguatge que es desmarcaria del conjunt, perquè també s'ha utilitzat moltíssim en tasques d'administració.

Per concloure, si tenim en compte els requeriments del client com la creació d'un sistema executable que sigui extensible, que el rendiment no és una necessitat de primer ordre i que la claredat, manteniment, la reusabilitat i la portabilitat siguin elements destacats podríem dir que el llenguatge *Java* és una bona opció.