



Disseny i implementació d'un sistema d'intel·ligència de negoci per parcs eòlics amb boies meteorològiques

Santi Béjar Latonda
Màster Universitari En Enginyeria Informàtica
Business Intelligence

David Amorós Alcaraz
María Isabel Guitart Hormigo

01/2017



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Disseny i implementació d'un sistema d'intel·ligència de negoci per parcs eòlics amb boies meteorològiques</i>
Nom de l'autor:	<i>Santi Béjar Latonda</i>
Nom del consultor/a:	<i>David Amorós Alcaraz</i>
Nom del PRA:	<i>María Isabel Guitart Hormigo</i>
Data de lliurament (mm/aaaa):	<i>01/2017</i>
Titulació o programa:	<i>Màster Universitari En Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Business Intelligence</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Business Intelligence JasperReports ETL</i>

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

La *intel·ligència de negoci* (BI) tracta de millorar la presa de decisions dels usuaris de les organitzacions mitjançant l'anàlisi efectiu i ràpid de les dades disponibles, que cada cop n'hi ha més. Per això la seva importància creix any darrera any.

La finalitat del present treball consisteix en el disseny i implementació d'una solució de *Business Intelligence* (BI) per tal d'analitzar les dades d'una sèrie de parcs eòlics *offshore*. En particular, 1) dissenyar i implementar un magatzem de dades (DWH) adequat a les dades originals i les anàlisis que es volen realitzar 2) dissenyar i implementar els processos ETL per transformar i guardar les dades al DWH 3) comparar i implementar diferents solucions de BI de font oberta 4) dissenyar i implementar el cub OLAP eficient per fer les consultes i 5) realitzar les anàlisis sobre el cub OLAP.

S'ha seguit una metodologia tradicional *bottom-top* per la realització de projecte de BI a on es parteix de les dades disponibles i les preguntes a respondre per dissenyar tota la solució.

Els principals objectius s'han completat amb èxit ja que després del projecte es disposa d'una solució de BI per fer les anàlisis de les diferents variables ens els eixos i la granularitat desitjada.

Abstract (in English, 250 words or less):

The Business Intelligence (BI) improves the decision making of users of organizations through the effective and rapid analysis of the increasing available data. Its importance is growing year after year.

The purpose of this work is to design and implement a Business Intelligence (BI) solution that analyzes data from a number of offshore wind farms. In particular, 1) design and implement a data warehouse (DWH) capable of handling the original data and performing the desired analysis 2) design and implement ETL processes that transform and save the data to DWH 3) compare and deploy various open source BI solutions 4) design and implement an efficient OLAP cube to perform queries and 5) perform analysis on the OLAP cube.

We have followed the traditional methodology of bottom-top for the BI project, one based on the available data and questions to design the entire solution.

The main objectives have been successfully completed since *it has a BI* solution for analysis of different variables with the desired axis and granularity.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball.....	2
1.3 Enfocament i mètode seguit.....	3
1.4 Planificació del Treball.....	4
1.5 Breu sumari de productes obtinguts.....	6
1.6 Breu descripció dels altres capítols de la memòria.....	6
2. Comparació Solucions de BI i eines utilitzades.....	8
2.1 Solucions de <i>Business Intelligence</i>	8
2.2 Solucions de Sistemes de gestió de bases de dades.....	12
3. Disseny i implementació de Data Warehouse i el ETL.....	13
3.1 Data Warehouse.....	13
3.2 Indicadors clau.....	14
3.3 Model conceptual.....	14
3.4 Processos d'extracció, transformació i càrrega (ETL).....	18
3.5 Implementació del DWH.....	24
3.6 OLAP (On-line Analytical Processing) i MDX.....	26
4. Anàlisis.....	31
4.1 Vistes OLAP amb JasperReports.....	31
4.2 Gràfiques i informes amb JaspersoftStudio.....	33
4.3 Anàlisi dels parc eòlics més productius.....	36
4.4 Anàlisi de les dades meteorològiques i la producció.....	38
4.5 Anàlisi de les zones amb millor relació vent/potència.....	42
4.6 Anàlisi d'alarmes.....	43
4.7 Anàlisi de la disponibilitat i les empreses de manteniment.....	47
5. Conclusions.....	51
6. Glossari.....	53
7. Bibliografia.....	55
8. Annexos.....	56
Annex 1 Guió SQL per la creació del DWH.....	56
Annex 2 Schema OLAP del OSWF.....	59
Annex 3 JasperReport amb gràfics de dispersió per cada tipus d'alarma.....	61

Llista de figures

Il·lustració 1: Quadrant Gartner 2016.....	9
Il·lustració 2: Model conceptual en forma de floquet de neu.....	17
Il·lustració 3: Model físic en forma d'estrella.....	18
Il·lustració 4: Taules auxiliars per el procés ETL.....	20
Il·lustració 5: Procés ETL per la la dimensió "Temps".....	20
Il·lustració 6: Procés ETL per la dimensió "parc eòlic".....	21
Il·lustració 7: Procès ETL per les taules de fets.....	22
Il·lustració 8: Detail del pel procés ETL de les taules de fets.....	24
Il·lustració 9: Contingut de la taula de la dimensió "Tipus d'alarma".....	24
Il·lustració 10: Contingut de la taula de la dimensió "Temps".....	25
Il·lustració 11: Contingut de la taula de la dimensió "Parcs eòlics".....	25
Il·lustració 12: Contingut de la taula de fets "Alarmes".....	25
Il·lustració 13: Contingut de la taula de fets "Mesures Parc".....	25
Il·lustració 14: Definició del model a SpagoBI Meta.....	27
Il·lustració 15: Exemple de consulta MDX utilitzant Jaspersoft OLAP Schema Workbench.....	29
Il·lustració 16: Dades de connexió a la base de dades del DWH.....	30
Il·lustració 17: Elements necessaris per crear una vista OLAP amb Jaspersoft Server.....	31
Il·lustració 18: Paràmetres de JasperReport(JFreeChart) per crear un gràfic XY.....	34
Il·lustració 19: Gràfica de la producció energètica dels parcs per mes.....	36
Il·lustració 20: Gràfica de la eficiència de la producció energètica dels parcs per mes.....	37
Il·lustració 21: Gràfic de la potència produïda en front de l'alçada de les ones	39
Il·lustració 22: Gràfic de la potència produïda en front de la velocitat del vent.	40
Il·lustració 23: Gràfic de la potència produïda en front de la temperatura.....	41
Il·lustració 24: Gràfic de la relació potència/velocitat del vent i potència/velocitat del vent pels diferents parcs.....	42
Il·lustració 25: Gràfics del nombre d'alarmes per turbina en front de la temperatura per les diferents tipus d'alarmes.....	44
Il·lustració 26: Gràfics del nombre d'alarmes per turbina en front de la velocitat del vent per les diferents tipus d'alarmes.....	45
Il·lustració 27: Gràfics del nombre d'alarmes per turbina en front de l'alçada de les ones per les diferents tipus d'alarmes.....	46

1. Introducció

1.1 Context i justificació del Treball

El Treball de Final de Màster en la present memòria consisteix en el disseny i implementació d'una solució de *Business Intelligence* (BI) per tal d'analitzar les dades d'una sèrie de parcs eòlics *offshore*.

Avui en dia, la energia Eòlica és la més avançada tècnicament. A més, de totes les energies renovables, és també la més rentable. Segons alguns experts, només aquesta energia podria arribar algun dia a abastir Europa sense problemes, sent el seu potencial, pel 2020, 3 vegades superior a la demanda prevista.

Un aerogenerador és un dispositiu capaç de transformar l'energia eòlica en energia elèctrica. Les parts de l'aerogenerador que permeten realitzar aquesta transformació són les pales, la multiplicadora (caixa d'engranatges) i el generador, encarregat de produir l'energia elèctrica.

Els aerogeneradors que actualment es construeixen, estan altament sensoritzats. L'elevat nombre de dades que es recullen, pot donar molta informació sobre l'estat actual dels diferents sistemes que els componen. Indicadors com la potencia, la temperatura, les vibracions o el vent ens poden ajudar en gran mesura a anticipar aturades i evitar fallades que poden arribar a ser molt costoses.

El rendiment d'un aerogenerador depèn, entre d'altres, de la seva localització geogràfica. No obstant, un manteniment adequat permet treure més profit quan les condicions atmosfèriques són les adequades.

L'energia eòlica té un futur prometedor encara que els millors terrenys han estat ja ocupats o estan en tràmits d'autorització i cada vegada és més difícil trobar zones amb altes velocitats de vent sense explotar; com a solució a aquest problema apareixen una nova possibilitat, l'energia eòlica *offshore*.

A qualsevol parc eòlic el factor decisiu és la velocitat del vent, en el mar es calcula una velocitat en 1 m/s per sobre de les zones costaneres properes a

causa que en el mar no existeixen obstacles i la rugositat del sòl és moltíssim menor, això significa que en un parc eòlic offshore la producció d'electricitat al llarg de l'any és de l'ordre d'un 20% més que en terra.

El fet que els parcs eòlics offshore se situen mar endins i el seu manteniment és costós (monitorització, actuacions, recanvis, etc), han començat a aparèixer solucions que poden millorar aquestes activitats. Una d'elles és la col·locació d'una Boia meteorològica dins el parc que permeti obtenir informació sobre les condicions del parc en tot moment. Això permetrà conèixer l'origen de determinades incidències al parc i preveure possibles alarmes o avaries.

Per tot l'anterior sorgeix la necessitat de tractar totes aquestes dades d'una manera global i, per tant, centralitzar-ho en un magatzem de dades i disposar d'un sistema d'intel·ligència de negoci per poder prendre decisions en base als anàlisis realitzats.

1.2 Objectius del Treball

L'objectiu d'aquest treball es el disseny i implementació d'un sistema de Business Intelligence que faciliti l'adquisició, l'emmagatzemament i l'explotació de dades provinents de diferents parcs eòlics on tenim instal·lades Boies Meteorològiques. Això permetrà poder disposar de informació addicional sobre les condicions meteorològiques i tenir més capacitat d'anàlisi.

Els objectius concrets del projecte són:

1. Dissenyar una magatzem de dades (*Data Warehouse*) que permeti emmagatzemar la informació adquirida des dels parcs eòlics tant a nivell de producció com meteorològic.
2. Implementar aquest magatzem de dades i programar els processos ETL (extracció, transformació i càrrega) que permetin alimentar el Data Warehouse a partir dels fitxers base facilitats.
3. Analitzar les diferents plataformes BI Open Source disponibles al mercat que ens permetin explotar la informació emmagatzemada.
4. Triar i implantar una d'aquestes eines Open Source de tal forma que es disposi d'una capa de programari per l'anàlisi de la informació.

Les preguntes que es volen resoldre són les següents:

- Quins són els parcs eòlics més productius. No necessàriament les zones amb constants de vent més altes són les millors, de vegades

la col·locació dels aerogeneradors o la tecnologia (marca, model) poden fer variar la potència generada en cada cas.

- Fer una anàlisi profunda sobre les dades meteorològiques de cada parc i la incidència sobre la producció.
- Zones amb millor relació vent/potència. Donat el cas que es vulgui ampliar un parc o crear-ne un de nou, aquesta dada ens pot ser molt útil per valorar la ubicació definitiva.
- Anàlisi d'alarmes. Veure si hi ha alguna relació entre les alarmes que es produeixen (que acostumen a aturar el aerogeneradors) i les variacions meteorològiques. Això ens pot ajudar a analitzar quins són els problemes dels parcs.
- Efectivitat de les empreses de manteniment a partir de la dada de disponibilitat. És desitjable que una empresa de manteniment minimitzi les alarmes i maximitzi la disponibilitat. Aquesta informació és vital per la establir els criteris contractuals o renovar-los.
- Relacions entre variables de producció i meteorològiques. La temperatura externa en un parc pot ser un indicador de futures avaries, de la mateixa forma que ho pot ser ratxes de vent extremes molt continuades o l'alçada de les onades.

1.3 Enfocament i mètode seguit

Dintre del context de BI existeixen dos grans enfocaments: el top-bottom i el bottom-top. El primer model parteix de les necessitats i objectius de negoci mentre que el model tradicional bottom-top analitzar les dades originals i les seves agrupacions per arribar a la informació de gestió requerida.

La metodologia seguida en aquest TFM ha estat la d'analitzar els requeriments del client a partir de l'enunciat que juntament amb les dades originals i les seves agrupacions permetin extreure la informació necessària. Per tant, s'ha optat per l'enfocament més tradicional bottom-top.

1.4 Planificació del Treball

1.4.1 Fases del projecte

El treball es pot descompondre en les següents fases:

1. Dissenyar una magatzem de dades (Data Warehouse) que permeti emmagatzemar la informació adquirida des dels parcs eòlics tant a nivell de producció com meteorològic.
2. Implementar aquest magatzem de dades i programar els processos ETL (extracció, transformació i càrrega) que permetin alimentar el Data Warehouse a partir dels fitxers base facilitats.
3. Analitzar les diferents plataformes BI Open Source disponibles al mercat que ens permetin explotar la informació emmagatzemada.
4. Triar i implantar una d'aquestes eines Open Source de tal forma que es disposi d'una capa de programari per l'anàlisi de la informació.

1.4.2 Planificació

Les diferents tasques i planificació seran inicialment les següents:

Nom	Durada	Inici	Final
PAC2	35 dies	04/10/16	07/11/16
Anàlisi de requisits	4 dies	04/10/16	07/10/16
Model de dades	6 dies	08/10/16	13/10/16
Model de fets, dimensions i mètriques	12 dies	14/10/16	25/10/16
Anàlisi solucions BI	4 dies	26/10/16	29/10/16
Elecció solució BI	1 dies	30/10/16	30/10/16
Disseny Arquitectura Datawarehouse	5 dies	31/10/16	04/11/16
Implementació Datawarehouse	3 dies	05/11/16	07/11/16
PAC3	36 dies	08/11/16	13/12/16
Implementació Datawarehouse	2 dies	08/11/16	09/11/16
Disseny ETL	4 dies	10/11/16	13/11/16
Implementació ETL	10 dies	14/11/16	23/11/16
Proves i correccions ETL	6 dies	24/11/16	29/11/16
Disseny OLAP	4 dies	30/11/16	03/12/16
Implementació OLAP	8 dies	04/12/16	11/12/16
PAC4	28 dies	13/12/16	09/01/17
Implementació OLAP	4 dies	13/12/16	16/12/16
Proves i correccions OLAP	6 dies	17/12/16	22/12/16
Redacció i correcció	18 dies	23/12/16	09/01/17
Memòria	98 dies	04/10/16	09/01/17
Redacció i correcció	98 dies	04/10/16	09/01/17
Debat Virtual		23/01/17	25/01/17

Amb la següent descripció:

- Anàlisi de requeriments: anàlisi detallada dels requeriments del sistema
- Model de dades: confecció del model de dades resultant de les dades originals.
- Model de fets, dimensions i mètriques: confecció del model de fets, dimensions i mètriques a partir del model de dades i les necessitats d'anàlisis.
- Disseny Arquitectura Datawarehouse: elecció de l'arquitectura del magatzem de dades, com s'introduiran les dades i com accediran els usuaris.
- Implementació Datawarehouse: instal·lació i configuració d'una base de dades segons el disseny anterior
- Anàlisi solucions BI: Comparativa de les diferents solucions Open Source en BI, tant les seves característiques com les necessitats pròpies del projecte
- Elecció solució BI: a partir de la comparativa anterior escollir la solució BI que millor s'adapta al projecte actual

- Disseny ETL: dissenyar tot el procés de recollida de les dades originals, les transformacions necessàries i la seva càrrega a la base de dades. Pot ser necessari una comparació de les diferents opcions: des de una eina en la solució de BI anterior o una independent.
- Implementació ETL: Implementar amb la eina més adient el procés ETL.
- Proves i correccions ETL: Crear un conjunt de dades de prova per carregar al sistema BI i corregir els errors trobats.
- Disseny OLAP: dissenyar un conjunt de consultes i/o informes que permetin respondre a les preguntes formulades als objectius.
- Implementació OLAP: implementar les consultes i informes anteriors
- Proves i correccions OLAP: Crear un conjunt de consultes i informes de prova per comprovar la seva utilitat i corregir els errors trobats.
- Redacció i correcció: Redacció de l'informe final del TFM. Aquest es realitzarà en paral·lel a totes les tasques anteriors i de forma exclusiva al final del projecte.

Tal com es pot veure la planificació és lineal excepte la redacció i correcció que es faran paral·lelament i exclusivament al final, degut principalment que només una persona estarà dedicada al projecte.

1.5 Breu sumari de productes obtinguts

1.6 Breu descripció dels altres capítols de la memòria

- Capítol 2. Comparació Solucions de BI i eines utilitzades

Al capítol 2 es mostra una petita comparació de les solucions de BI disponibles, principalment *Open Source*. A continuació es presentaran les eines utilitzades i una justificació sobre la seva elecció.

- Capítol 3. Disseny i implementació de *Data Warehouse* i el *ETL*

Al capítol 3 es mostra el disseny del DWH necessari per emmagatzemar les dades requerides i la seva implementació sobre la base de dades escollida. La segona part del capítol conté el disseny i implementació del processos necessaris per importar les dades des de la seva ubicació original al DWH.

- Capítol 4. Anàlisi

Al capítol 4 es mostra com utilitzar l'eina de BI escollida que, juntament amb les dades introduïdes a partir del ETL al DWH, permet realitzar les anàlisis, taules i gràfiques amb la finalitat de respondre a les preguntes inicials.

2. Comparació Solucions de BI i eines utilitzades

El primer pas per implementar una solució de BI adequat és realitzar una comparació de les diferents solucions disponibles actualment. A continuació es realitzarà un petit estudi comparatiu de les diferents eines necessàries per aquest estudi.

En aquest TFM s'han d'escollir principalment dos eines o plataformes:

- Solució de *Business Intelligence* (**BI**): eina o plataforma amb la qual es realitzaran tots els processos des de les dades originals fins l'anàlisi. Pot ser una plataforma integrada o un conjunt d'eines que treballen coordinadament.
- Solució de Base de dades (**BBDD**): base de dades relacional que emmagatzemarà les dades del DWH.

2.1 Solucions de Business Intelligence

La elecció de la solució de BI és una de les més importants ja que d'ella dependrà en gran mesura l'èxit del projecte. La solució escollida tindrà impacte en el model de dades, la càrrega de dades (ETL) i les anàlisis.

Hi ha una multitud de solucions disponibles al mercat i fer una comparació de tots està fora de l'abast del projecte. Tot i així (Garner 2016) publica anualment una comparació extensa *Magic Quadrant for Business Intelligence and Analytics Platforms*. Les solucions que apareixen han de complir de 9 de les 14 funcionalitats següents:

- Infraestructura
 - Administració de la plataforma BI
 - Cloud BI
 - Administració d'usuaris i seguretat
 - Connectivitat amb fonts de dades
- Gestió de dades

- Gestió de metadades
- ETL i emmagatzematge de dades
- Preparació de dades per l'usuari
- Creació d'anàlisis i contingut
 - Quadre de comandaments analítics
 - Exploració visual interactiva
 - Exploració i autoria mòbil
- Compartició de resultats
 - Incrustació de contingut analític
 - Publicació de contingut analític
 - BI col·laboració i social



Il·lustració 1: Quadrant Gartner 2016

Tal com es pot observar al quadrant, Gartner valora les diferents solucions segons l'habilitat d'excussió i la completitud de la visió, i els classifica en quatre categories: líders, visionaris, "challengers" i solucions de nínxol.

En el cas d'estudi els requeriments sobre les solucions de Business Intelligence són:

- Solució que cobreixi tots els passos necessaris des de la pressa de dades fins l'anàlisi per part dels usuaris finals i generació dels informes
 - Pot ser una solució integral o un conjunt d'utilitats que treballen conjuntament.
- Solució preferiblement de font oberta (open source)
- Anàlisis:
 - Relacions entre diferents indicadors
 - Evolucions temporals de les dades
 - Diferents nivells de detalls i jerarquies

Per una breu comparació de les diferents solucions en Business Intelligence es pot trobar a la bibliografia.

Les solucions comparades i les seves característiques són les següents:

Jasper Reports:

- Pàgina principal: <http://www.jaspersoft.com/>
- Llicència: Community and comercial editions
- Components destacats:
 - Jasper Server: "robust static and interactive reporting, report server, and data analysis capabilities"
 - Creació d'informes ad-hoc a través de la web
 - Gestió d'informes: ordenar, canvi de format, introducció de paràmetres,...
 - Creació de quadres de comandaments
 - Jasper Studio: "Design interactive pixel perfect and/or ad hoc based reports for the web, the printer or mobile device"
 - Informes interactius
 - Informes Pixel-Perfects

Pentaho: "A Comprehensive Data Integration and Business Analytics Platform"

- Pàgina principal: <http://www.pentaho.com/>
- Llicència: Community and comercial editions
- Components destacats:
 - Business Analytics Platform: "Pentaho's modern, simplified and interactive approach empowers business users to access, discover and blend all types and sizes of data"
 - Anàlisis avançades: des de informes bàsics fins predictius
 - Anàlisis de múltiples dimensions
 - Dissenyat per usuaris mòbils

- Data Integration: "delivers powerful extraction, transformation, and loading (ETL) capabilities"
- Report Designer: "graphical tool that generates reports from data streamed through the Data Integration engine without the need for any intermediate staging tables"

BIRT: "An open source technology platform used to create data visualizations and reports that can be embedded into rich client and web applications."

- Llicència: Eclipse Public License
- Pàgina principal: <http://www.eclipse.org/birt/>

SpagoBI: "100% Open Source Business Intelligence and Big Data analytics"

- Llicència: free/open source software
- Components destacats:
 - Reporting: generació d'informes estructurats
 - OLAP: anàlisi multidimensional
 - KPI: eines per crear, gestionar i visualitzar KPI
 - ETL: Integra la solució TOS (*Talend Open Studio*)

Amb tot l'esmentat anteriorment, juntament amb la documentació d'introducció de cadascun, la solució adoptada inicialment en aquest treball és *SpagoBI*, ja que compleix tots els requeriments i és totalment open source, en contraposició de *Jaspersoft* i *Pentaho* que són *Open Core*. Cal esmentar en aquest punt que tots compleixen els requeriments i que són igual de vàlids en el context d'aquest treball.

Al final la solució adoptada ha estat utilitzar *Talend Open Solution for Data Integration* juntament amb *Jaspersoft*. Aquest canvi va ser degut a diferents problemes i dificultats per treballar amb *SpagoBI*. Entre aquests podem destacar:

- Falta de documentació, en particular un manual actualitzat d'instal·lació i una guia d'usuari
- Plataforma molt potent i completa però que alhora fa que tingui una interfície complexa i amb una corba d'aprenentatge alta.

La elecció de *Talend Open Source for Data Integration* es basa en que és una solució molt completa i potent, tot i que amb una corba d'aprenentatge alta, i estar integrada a *SpagoBI* i *Jaspersoft*. Per altra banda la elecció de *Jaspersoft* rau en la documentació disponible, tant d'instal·lació com la guia d'usuari i la documentació específica sobre OLAP.

2.2 Solucions de Sistemes de gestió de bases de dades

Per fer l'elecció del SGBD hem de tenir en consideració el suport d'aquest en la solució de BI escollida, però també les seves característiques. En comptes de fer una comparació de les diferents solucions SGBD (que està fora de l'abast del projecte actual) escollirem *MySQL* pels següents motius:

- Compatibilitat amb gairebé totes les solucions de BI i sistemes operatius
- Solució àmpliament utilitzada i coneguda tant en entorns personals com empresarials
- Solució altament flexible i escalable

3. Disseny i implementació de Data Warehouse i el ETL

3.1 Data Warehouse

Un magatzem de dades, o *data warehouse* en anglès, (DWH) es pot definir com:

"Un data warehouse és una col·lecció de informació creada per donar suport a les aplicacions per la presa de decisions"

Hugh J. Watson

Van aparèixer per donar resposta a la necessitat dels usuaris per disposar d'informació consistent, integrada, històrica i preparada per prendre decisions. Les característiques que ha d'acomplir són:

- **Orientat a temes:** s'organitzen les dades al voltant d'una àrea per donar solucions a problemes de negoci.
- **Integrat:** s'integren les dades de diferents sistemes operacionals i, per tant, s'ha posat en comú els diferents conceptes.
- **Indexat en el temps:** es guarden canvis en les dades per poder analitzar variacions d'aquestes en el temps.
- **No volàtil:** els usuaris finals no mantenen aquestes dades, sinó que només les consulten per la presa de decisions.

Per tant, podem dir que un Data Warehouse guarda les dades provinents dels sistemes operacionals de la organització d'una manera integrada i que periòdicament s'actualitza amb noves dades.

Relacionat amb els Data Warehouse hi ha els *Data Marts*, que contenen una part del DWH per tal de facilitar el seu manteniment i consulta i reduir alhora els requeriments de programari i maquinari. Normalment aquests es creen al voltant d'una àrea específica.

Un altre component important del DWH és *Metadata*. El metadata és el repositori que recull la informació sobre la informació continguda al DWH. És a dir, dóna un significat a la informació continguda al DWH. Aquest component és imprescindible en entorns grans de BI per unificar i entendre la informació continguda.

3.2 Indicadors clau

Abans de definir el model conceptual és interessant trobar els indicadors claus per incorporar-los en el model del DWH. Els indicadors claus són les mesures (ja siguin en cru o calculades a partir d'altres) que donen resposta a les preguntes plantejades i/o en l'ajuda de les presses de decisions.

Per tant, a partir dels objectius plantejats, podem definir els indicadors claus:

- **Producció del parc eòlic:** energia i potència mitjana generada a un període particular.
- **Dades meteorològiques de cada parc eòlic:** velocitat del vent, alçada de les ones i temperatura
- **Dades de producció de cada parc eòlic:** energia produïda, disponibilitat i temps de reparació
- **Nombre d'alarmes de cada tipus a cada parc eòlic:**
- **Eficiència de la producció:** Percentatge de la energia produïda en front de la producció nominal del parc.
 - Fórmula: "potència mitjana" / "potència nominal"
- **Ràtio velocitat del vent i potència.**
 - Fórmula: "potència" / "velocitat del vent"
- **Nombre d'alarmes per turbina:** Nombre mitjà d'alarmes per turbina.
 - Fórmula: "nombre d'alarmes" / "nombre de turbines"

3.3 Model conceptual

Tot DWH es basa en un model conceptual que defineix el seu contingut així com les seves relacions i la seva relació amb les estructures utilitzades dintre de la solució de Business Intelligence, com ara les dimensions i mesures. Aquí presentem els diferents conceptes de l'usuari final necessaris i com es traslladen a conceptes de la solució de BI.

- **Dimensions:** aquests defineixen les perspectives amb les quals es pot analitzar les dades. Per exemple podem tenir la dimensió "Temps" o la dimensió "Tenda". Les dimensions s'organitzen en una o més jerarquies.
- **Jerarquia:** elements de la jerarquia que contenen un seguit de nivells organitzats de forma jeràrquica.
- **Nivells:** cadascun dels elements d'una jerarquia i defineix el seu nivell d'agrupació. Per exemple, podem tenir a la dimensió "Temps" la jerarquia "TempsTrimestres" amb els nivells "Any:Trimestre:Mes:Dia" i la jerarquia "Temps Quatrimestres" amb els nivells "Any:Quatrimestre:Mes:Dia".
- **Indicadors o mètriques:** dades que es volen analitzar, normalment són valors numèrics. Per exemple: nombre de productes produïts.
- **Fets:** Conjunt d'indicadors associat a un membre d'una dimensió i contingut en una taula de fets.
- **Taula de fets:** Conjunt de fets que segueixen el mateix esquema.

A continuació aplicarem aquests conceptes al tema tractat en aquest TFM:

Dimensions

- **Temps:** Dimensió per associar el moment de presa de les dades. Es vol guardar les dades al DWH amb una granularitat mínima de dies.
 - Atributs:
 - Any
 - Mes
 - Dia
 - Jerarquia:
 - Nivell 0: Any
 - Nivell 1: Mes
 - Nivell 2: Dia
- **Parc eòlic:** Dimensió per associar el parc eòlic.
 - Atributs:
 - País
 - Mantenedor
 - Operador

- Jerarquia: Parc
 - Nivell 0: Parc
- Jerarquia: País
 - Nivell 0: País
 - Nivell 1: Parc
- Jerarquia: Operator
 - Nivell 0: Operator
 - Nivell 1: Parc
- Jerarquía: Mantenedor
 - Nivell 0: Mantenedor
 - Nivell 1: Parc
- **Tipus d'alarma:** Dimensió per associar el tipus d'alarma
 - Atributs:
 - Nom
 - Jerarquia:
 - Nivell 0: Tipus d'alarma

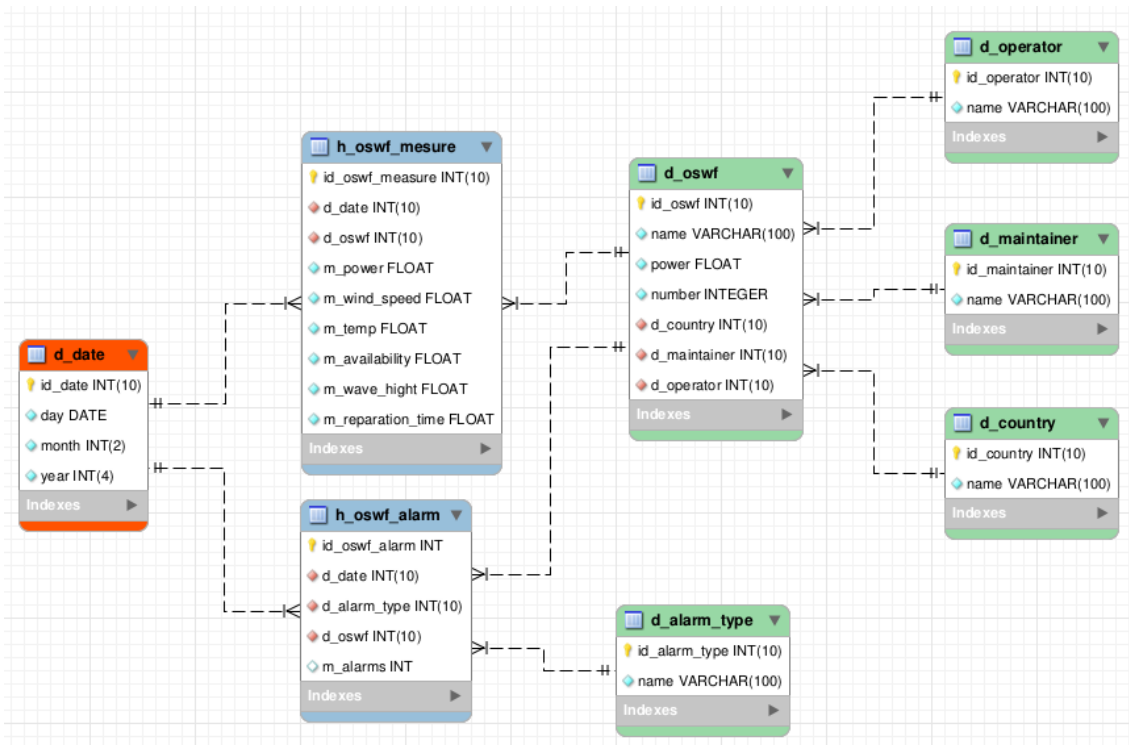
Taula de Fets

- Mesures parc: fets que descriuen les mesures meteorològiques i de producció d'un parc en un moment determinat.
 - Dimensions:
 - Parc
 - Tipus alarma
 - Temps
 - Indicadors:
 - Velocitat del vent (m)
 - Alçada de les ones (m)
 - Temperatura (m)
 - Energia produïda (+)
 - Disponibilitat (Temps → m)
 - Temps de reparació (+)
- Alarmes: fets que descriuen la quantitat d'alarmes en un període determinat en un parc determinat.
 - Dimensions:
 - Parc
 - Tipus d'alarma
 - Data
 - Indicadors:
 - Quantitat d'alarmes (+)
- Mesures parc: fets que descriuen atributs dels parc necessaris per calcular els indicadors clau

- Dimensions:
 - Parc
- Indicators:
 - Potència nominal total del parc (+)
 - Nombre de turbines (+)

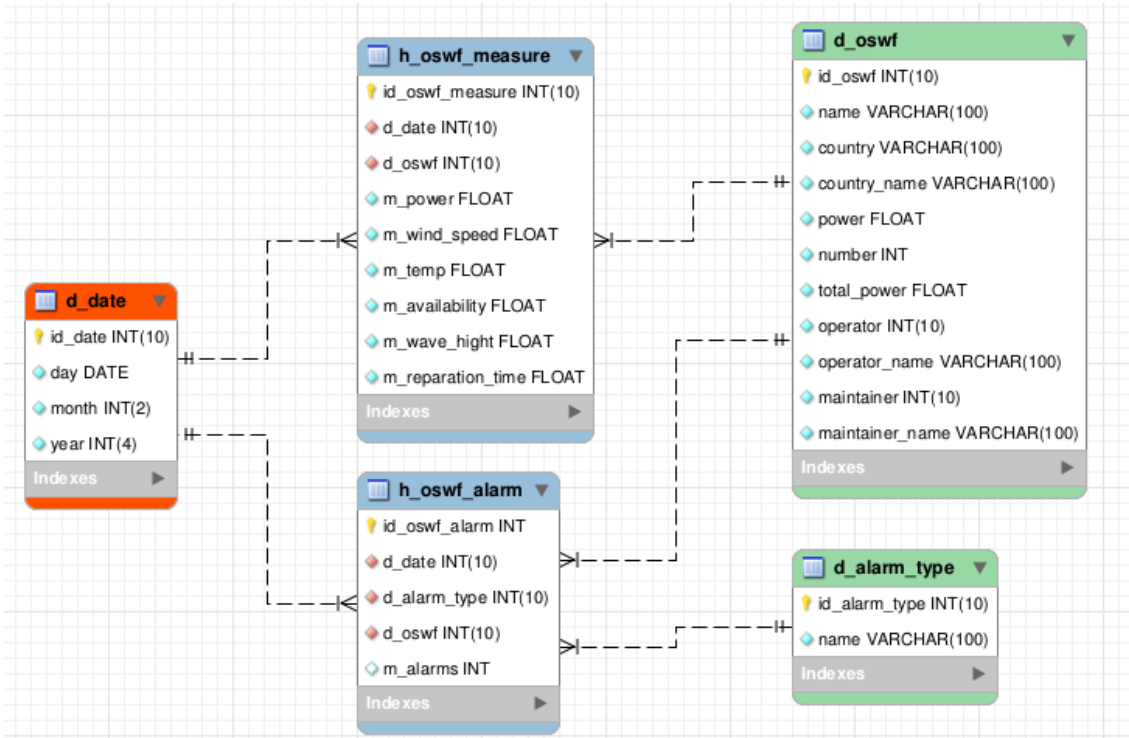
a on (m) representa els indicadors amb una funció de resum "mitjana" i (+) "suma". El cas de la disponibilitat (Temps → m) és diferent, ja que es tracta d'una mesura que té una funció d'agregació "mitjana" en el cas del temps, però no es pot fer agregació en la dimensió "parc eòlic". En l'últim cas s'hauria de fer una mitjana ponderada utilitzant el nombre de turbines com a pes. Una possibilitat seria descomposar la disponibilitat en el "nombre efectiu de turbines disponibles" i el "nombre efectiu de turbines total" i calcular la disponibilitat com la seva ràtio. En aquest cas el "nombre efectiu de turbines disponibles" seria un atribut amb la funció d'agregació "suma".

A continuació es pot veure el disseny en forma de flocc de neu:



Il·lustració 2: Model conceptual en forma de floquet de neu

Amb tot el disseny físic de la solució proposada és en forma d'estrella per les seves avantatges a l'hora de tractar les dades posteriorment:



II-lustració 3: Model físic en forma d'estrella

A continuació descriu diferents decisions preses a l'hora de dissenyar el model de DWH:

- La taula parc eòlic (d_oswf) conté a la vegada la dimensió de parc eòlics (amb atributs de país, mantenidor i operador) i la taula de fets sobre cada parc (nombre de turbines i potència nominal).
- Utilitzar un model conceptual en forma d'estrella per simplificar la definició del cub OLAP i evitar l'ús d'unions (joins) de taules de la base de dades

3.4 Processos d'extracció, transformació i càrrega (ETL)

El procés d'extracció, transformació i càrrega (ETL, per les seves sigles en anglès) tracta de recuperar les dades de les fonts originals d'informació, transformar-les per adaptar-les a les necessitats del DWH i guardar-les a la DWH per la seva anàlisi posterior.

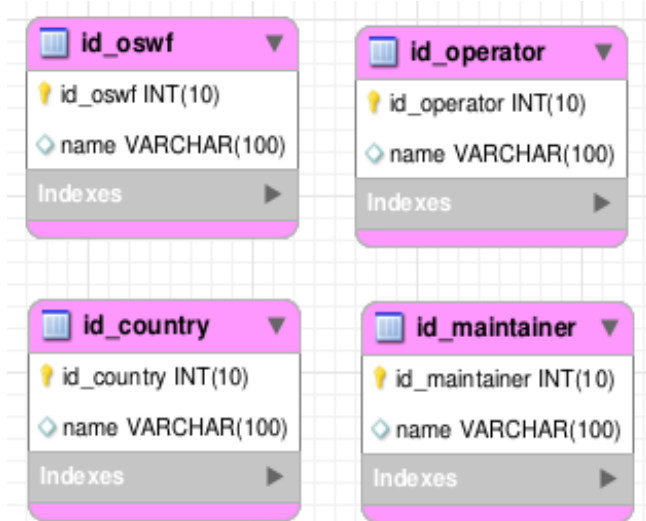
Els processos ETL consumeixen una part molt significativa dels recursos, en especial el temps, en el projectes de BI i requereix coneixements específics de tecnologia.

El procès ETL es divideix en 5 subprocessos:

- **Extracció:** Recuperació de les dades de les fonts originals d'informació. En aquest moment obtindrem les dades en cru.
- **Neteja:** Comprovació la qualitat, elimina els duplicats i corregeix, quan es possible, els valors erronis o que falten. En aquest moment obtindrem les dades netes i d'alta qualitat.
- **Transformació:** Recupera les dades anteriors i els estructura i resumeix segons els diferents models d'anàlisis. En aquest moment obtindrem les dades netes, consistents, resumides i útils.
- **Integració:** Validació segons les definicions i formats del DWH i integració en els diferents models de les diferents àrees.
- **Actualització:** Addició de noves dades al DWH

Per el disseny i implementació del ETL s'ha optat per l'eina open Sourou *Talend Open Studio* ja que ofereix una solució completa open Sourou de ETL.

Consta de dos parts ben diferenciades, la introducció de les dimensions i la taula de fets. Per la introducció de les dimensions "Temps" i "Parc eòlic" s'ha utilitzat el propi programa *Talend*. Per tal de disposar d'identificadors únics pels parcs, els països, els mantenidors i els operadors s'ha fet servir taules auxiliars externes tal com es mostra a la figura següent:

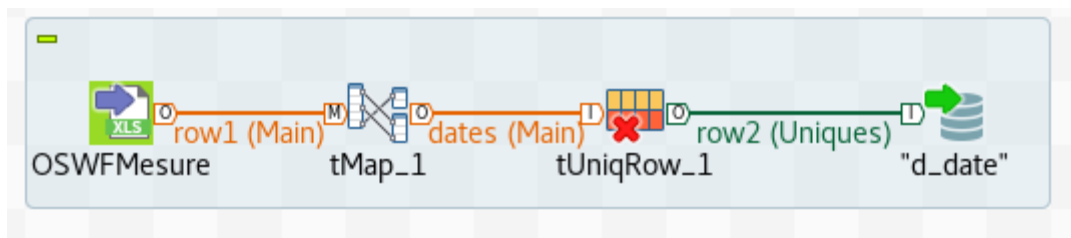


Il·lustració 4: Taules auxiliars per el procés ETL

Les dades originals es llegeixen directament dels arxius xls, ja que *Talend* permet llegir les dades des de múltiples orígens, entre ells arxius xls. S'han definit dos fonts de dades:

- WindFarm: font de dades que llegeix les dades sobre els parc eòlics, en concret la pestanya "OFFSHORE WIND FARM" del fitxer "DATAOFFSHOREvs20.xlsx".
- OSWFMesures: font de dades que llegeix de cadascuna de les pestanyes amb les mesures de producció i meteorològiques, en concret les pestanyes "NAMPER", "RIAS BAIXAS", "GUARACHICO", "KIRSKEN", "POLVARS", "COUNSCOT", "GREENBLUE", "NORTHENCAP". Això es possible perquè totes les pestanyes segueixen la mateixa estructura i format.

A continuació es poder observar el disseny del procés ETL per la dimensió "Temps":



Il·lustració 5: Procés ETL per la la dimensió "Temps"

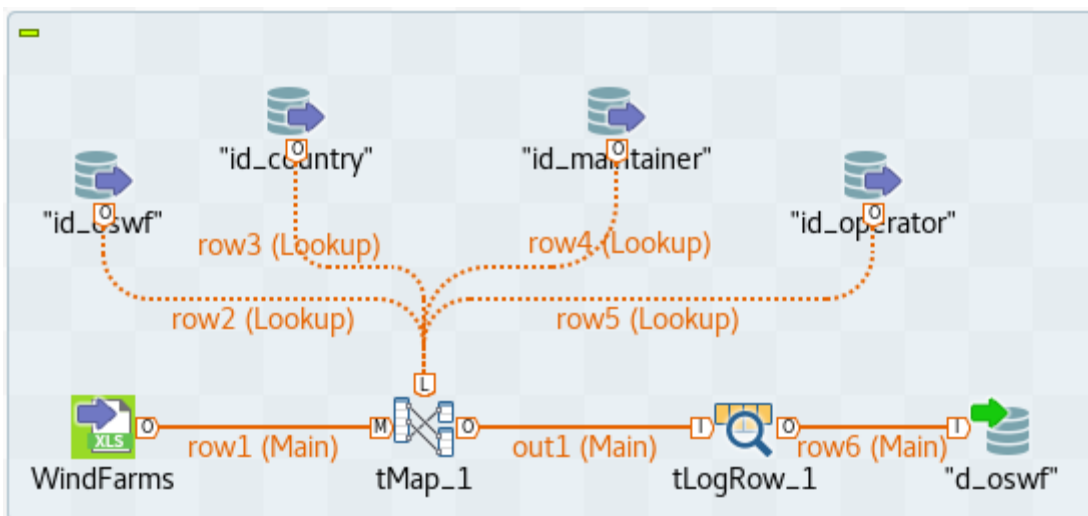
A on els passos realitzats són:

- OSWFMesures:

- Llegir les dades de les mesures
- tMap_1:
 - Seleccionar només la data
 - Calcular la clau primària:
 - `(int)TalendDate.diffDate(row1.DATE,TalendDate.parseDate("yyyy-MM-dd", "1970-01-01"), "dd")`
 - Calcular columnes extres com "any" i "mes"
 - `year=TalendDate.getPartOfDate("YEAR",row1.DATE)`
 - `month=TalendDate.getPartOfDate("MONTH",row1.DATE)+1`
- tUniqRow_1
 - Suprimir els duplicats
- d_date
 - Guardar a la DWH

Per la clau primaria de la taula "d_date" s'ha realitzat una simplificació i s'ha calculat com el nombre de dies des del 1 de gener del 1970. Així en els pròxims processos ETL es pot també calcular la clau primària de "d_date" sense haver de fer un tractament de les dates d'origen.

A continuació es poder observar el disseny del procés ETL per la dimensió "Parc eòlic":

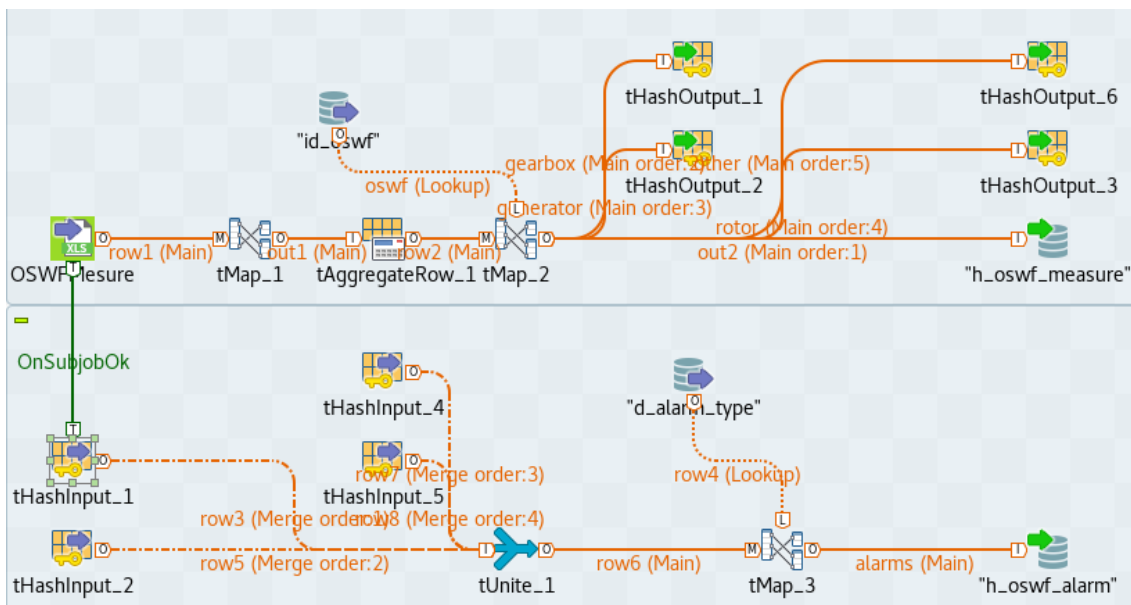


Il·lustració 6: Procés ETL per la dimensió "parc eòlic"

A on els passos realitzats són:

- WindFarms:
 - Llegir les dades dels parc eòlics
- tMap_1:
 - Afegir els claus foranes de: parc eòlic, país, mantenidor i operador
- d_oswf:
 - Guardar al DWH

Per finalitzar, el procès ETL dels fets és:



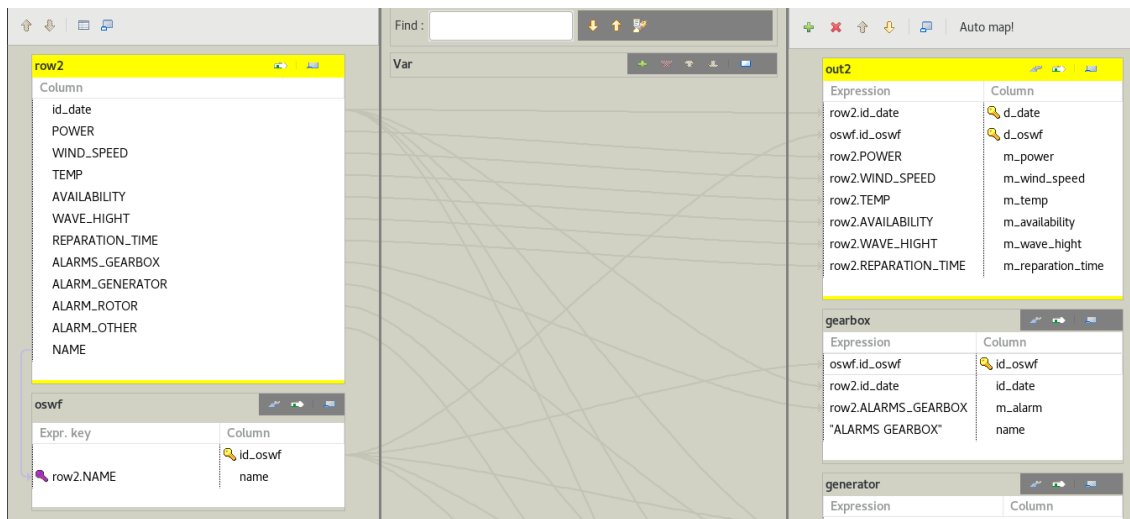
Il·lustració 7: Procès ETL per les taules de fets

A on els passos realitzats són:

- OSWFMesure:
 - Càrrega del fitxer
- tMap_1:
 - Calcular la clau primaria de la data
 - Afegir atribut "Name" amb el nom de la pestanya del fitxer d'origen:
 - ((String)globalMap.get("tFileInputExcel_1_CURRENT_SHEET"))
- tAggregateRow_1:
 - Agregar les mesures per parc (atribut "Name") i dia (clau foranea de la data)

- tMap_2:
 - Afegir els claus forànees de: parc eòlic
 - Separar cada registre en diferents registres: 1 per les dades meteorològiques i de producció i un per cada tipus d'alarma (guardant el nom del tipus d'alarma).
- tHashOutput_n:
 - Guardar els registres sobre les alarmes a memòria, ja que no es poden fer cercles (*loops* en anglès) a *Talend*
- h_oswf_measure:
 - Guardar les mesures meteorològiques i de producció a la seva taula de fets
- OnSubjobOk:
 - La subtasca de sota del diagrama s'executarà quan finalitza la tasca principal de forma correcta.
- tHashInput_n:
 - Carregar les dades guardades al tHashOutput_n corresponent
- tUnite_1:
 - Combinar totes les dades sobre les diferents alarmes ara que ja tenen un atribut que identifica el tipus d'alarma
- tMap_3:
 - Afegir la clau forànea del tipus d'alarma
- h_oswf_alarm
 - Guardar les mesures de les alarmes a la seva taula corresponent

A continuació es pot observar més en detall el procès "tMap_2":



Il·lustració 8: Detail del pel procés ETL de les taules de fets

3.5 Implementació del DWH

Tal com ja s'ha comentat anteriorment la BBDD del DWH escollit es *MySQL*. Per al disseny i creació de la base de dades s'ha utilitzat el *MySQL Workbench* que permet crear models de les bases de dades i crea un guió SQL que es pot executar al servidor *MySQL* per tal de crear totes les taules, atributs, relacions,... Aquest guió SQL es pot consultar al Annex 1 al final d'aquesta memòria.

Un cop executat l'guió SQL per la creació del DWH i els processos ETL ja tenim disponibles les dades en un format adequat per les consultes, que és l'objectiu principal d'aquest projecte. A continuació es mostra un extracte del contingut d'aquestes taules.

id_alarm_type	name
1	ALARMS GEARBOX
2	ALARM GENERATOR
3	ALARM ROTOR
4	ALARM OTHER

Il·lustració 9: Contingut de la taula de la dimensió "Tipus d'alarma"

id_date	day	month	year
16861	2016-03-01	3	2016
16862	2016-03-02	3	2016
16863	2016-03-03	3	2016
16864	2016-03-04	3	2016
16865	2016-03-05	3	2016
16866	2016-03-06	3	2016

Il·lustració 10: Contingut de la taula de la dimensió "Temps"

id_oswf	name	country	country_name	power	number	total_power	operator	operator_name	maintainer	maintainer_name
1	NAMPER	1	DENMARK	2	15	30	1	EUROWIND	1	GDEP
2	RIAS BAIXAS	2	SPAIN	2.5	25	62.5	2	ACTIONAK	1	GDEP
3	GUARACHICO	2	SPAIN	2	30	60	3	EOLIAGER	2	CUPRA
4	KIRSKEN	3	GERMANY	2.5	25	62.5	3	EOLIAGER	3	MANWIN
5	POLVARS	4	POLSKA	2.5	40	100	1	EUROWIND	3	MANWIN
6	COUNSCOT	5	UK	2	29	58	3	EOLIAGER	1	GDEP

Il·lustració 11: Contingut de la taula de la dimensió "Parcs eòlics"

id_oswf_alarm	d_date	d_alarm_type	d_oswf	m_alarms
667	16861	1	1	49
1403	16861	2	1	0
2139	16861	3	1	2
2875	16861	4	1	1
280	16861	1	2	107
1016	16861	2	2	8

Il·lustració 12: Contingut de la taula de fets "Alarmes"

id_oswf_measure	d_date	d_oswf	m_power	m_wind_speed	m_temp	m_availability	m_wave_high	m_reparation_time
1	16874	1	68448.4	7.6475	7.1875	99.5827	6.125	34.4
2	16940	1	115384	7.45486	7.82639	99.6053	5.98611	56.6
3	16907	1	124607	7.57361	7.51389	99.6649	5.64583	51.24
4	16870	1	122684	7.74514	7.3125	99.6607	5.88194	46.41
5	16936	1	117899	7.57266	7.47482	99.5931	6.1223	54.16
6	16903	1	130796	7.62639	7.09028	99.6482	5.9375	54.27

Il·lustració 13: Contingut de la taula de fets "Mesures Parc"

3.6 OLAP (On-line Analytical Processing) i MDX

OLAP és una tecnologia o solució que permet fer consultes a una base de dades d'una forma ràpida i eficaç. Es basa en la construcció d'unes estructures multidimensionals anomenades Cubs OLAP que defineixen les diferents consultes que es poden realitzar. Això facilita el processament de les consultes ja que aquestes es poden calcular amb antelació.

Un cub OLAP conté els següents elements:

- Dimensions associades al cub. Poden ser compartides entre cubs o privades.
- Mesures del cub. Mesura de la taula de fets o calculades a partir de les altres mesures.

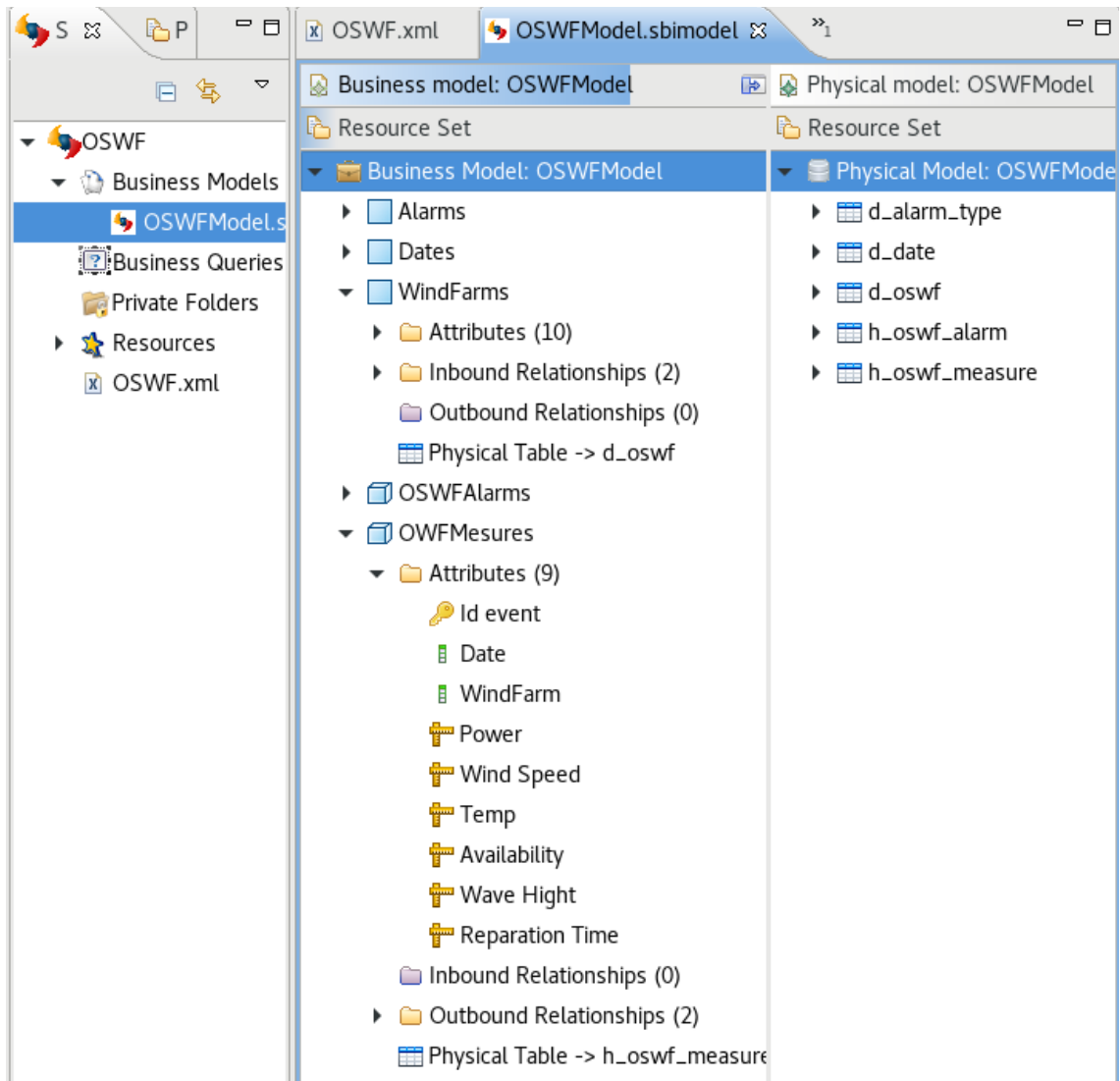
Un cub OLAP també pot ser un cub virtual que conté mesures i dimensions de diferents cubs.

Poden visualitzar els cubs OLAP com cubs multidimensionals a on cada eix és una dimensió i cada cel·la conté el valor agregat de la mètrica.

Per realitzar consultes sobre un cub OLAP s'utilitza un llenguatge anomenat MDX, similar al SQL, però més adequat a les característiques dels cubs OLAP. Per exemple el suport per la dimensió temporal (per tal de fer consultes en relació al període anterior) o el fet que les dades en un cub OLAP tenen un ordre (fet que no passa a les entrades d'una base de dades relacional).

La definició d'un cub OLAP es concreta en un fitxer xml seguint l'esquema Mondrian. En aquest esquema es defineixen els elements del cub necessaris per crear consultes MDX.

Per crear l'esquema del cub OLAP es poden fer servir diferents eines, com ara *SpagoBIMeta* i *Jaspersoft OLAP Schema Workbench*. En el cas de *SpagoBIMeta* la creació del cub OLAP forma part de la definició del model, que conté les metadades i els cubs. Un cop creat el model es pot crear un Cub OLAP separat. A la figura següent es pot veure la definició del model.



Il·lustració 14: Definició del model a SpagoBI Meta

El procés es guiat:

1. Importar les taules i atributs rellevants pel model
2. El programa crea elements del model de negoci corresponents
3. Configuració de cada taula com taula de dimensió o taula de fet
4. Configuració els atributs com tipus "mesura" i la seva funció d'agregació

En el nostre cas això no ha sigut suficient ja que necessitem crear un cub virtual per poder tractar les mesures dels dos cubs unificadament. Per aquesta raó s'ha optat per crear l'esquema Mondrian directament.

A continuació presento un extracte de l'esquema OLAP amb les parts més rellevants. A l'Annex 2 es pot trobar la definició sencera.

```
<?xml version="1.0"?>
<Schema name="OSWF">
```

```

<Dimension name="WindFarm">
  <Hierarchy hasAll="true" primaryKey="id_oswf">
    <Table name="d_oswf"/>
    <Level name="WindFarm Name" column="id_oswf" nameColumn="name"
uniqueMembers="true"/>
  </Hierarchy>
  <Hierarchy name="Country" allMemberName="All Countries" hasAll="true"
primaryKey="id_oswf">
    <Table name="d_oswf"/>
    <Level name="WindFarm Country" column="country" nameColumn="country_name"
uniqueMembers="true"/>
    <Level name="WindFarm Name" column="id_oswf" nameColumn="name"
uniqueMembers="true"/>
  </Hierarchy>
...
<Cube name="OSWFAlarms">
...
</Cube>
<Cube name="OSWFMeasures">
  <Table name="h_oswf_measure"/>
  <DimensionUsage name="WindFarm" source="WindFarm" foreignKey="d_oswf"/>
  <DimensionUsage name="Date" source="Date" foreignKey="d_date"/>
  <Measure name="Energy" column="m_power" aggregator="sum"
formatString="#.###"/>
  <Measure name="EnergyAvg" column="m_power" aggregator="avg" visible="false"
formatString="#.###"/>
...
  <CalculatedMember name="Power" dimension="Measures">
    <Formula>[Measures].[EnergyAvg] / 24 / 1000 </Formula>
  </CalculatedMember>
</Cube>
<VirtualCube name="Alarms and Measures">
  <CubeUsages>
    <CubeUsage cubeName="OSWFWindFarms" ignoreUnrelatedDimensions="true"/>
    <CubeUsage cubeName="OSWFMeasures" ignoreUnrelatedDimensions="true"/>
    <CubeUsage cubeName="OSWFAlarms"/>
  </CubeUsages>
  <VirtualCubeDimension name="WindFarm"/>
  <VirtualCubeDimension name="Date"/>
  <VirtualCubeDimension name="Alarm"/>
  <VirtualCubeMeasure cubeName="OSWFWindFarms" name="[Measures].[Nominal
Power]"/>
...
  <VirtualCubeMeasure cubeName="OSWFAlarms" name="[Measures].[Alarms]"/>
...
  <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].[Power]"/>
...
</VirtualCube>
</Schema>

```

Hi ha una sèrie de comentaris a fer respecte sobre la definició final del cub OLAP.

Les jerarquies s'ha mantingut tal com es van definir al capítol anterior, però les mesures requereixen d'una explicació extra.

La mesura "Power" a les dades inicials en realitat era una mesura de l'energia produïda, per això la mesura "Energy" al cub OLAP és la suma de la

mesura "m_power" del DWH. En canvi defineixo una mesura "Power" com la mitjana de l'energia produïda en MW. La divisió per 24 es degut a que l'energia produïda tal com es guarda al DWH és respecte d'un període de 24 hores, i divisió per 1000 per passar de KW a MW.

S'ha creat un cub virtual per combinar totes les mesures del cub d'alarmes i el cub de producció i meteorològic. Com que hi ha dimensions que no són comunes, en particular el tipus d'alarma, he hagut d'afegir un paràmetre "ignoreUnrelatedDimensions=true" per tal que les consultes MDX considerin les dimensions no relacionades com si no existissin. Això, juntament amb amb la funció ValidMeasure() és el que permet calcular indicadors amb diferents dimensions. Per exemple, per calcular l'eficiència d'un parc necessitaríem calcular la següent fórmula: "Potència del parc" / "Potència nominal del parc". En el cas que es vulgui calcular a nivell de mesos el cub "Parc eòlic" no conté la dimensió temps i per tant no tindria sentit la formula. Per tant el cub "Parc eòlic" ha d'ignorar aquesta dimensió i la fórmula ha saber que la mesura "Potència nominal del parc" és valida igualment. Això s'aconsegueix definint l'eficiència com:

[Measures].[Power] / ValidMeasure([Measures].[Nominal Power])

Aquest també va ser el motiu d'afegir la taula de fets "Parc eòlic", per poder calcular mesures que depenien d'atributs del parc.

Per fer consultes MDX podem fer servir el Jaspersoft OLAP Schema Workbench tal com es mostra a continuació:

The screenshot shows the JasperAnalysis Schema Workbench interface. At the top, there is a menu bar (File, View, Tools, Windows, Help) and a toolbar with icons for file operations and MDX. Below the menu is a toolbar for the MDX query editor, including a 'Schema' dropdown set to '1 OSWF.xml' and a 'Connect' button. The main area contains the following MDX query:

```
WITH MEMBER [Measures].[Efficiency] AS '[Measures].[Power]/ValidMeasure([Measures].[Nominal Power])',
    FORMAT_STRING = '#.##0'
select Crossjoin({[WindFarm].Children}, {[Measures].[Efficiency]}) ON COLUMNS,
    {Descendants([Date], [Month])} ON ROWS
from [Alarms and Measures]
```

Below the query is an 'Execute' button. The results pane shows the following data table:

	NAMPER	RIAS BAIXAS	GUARACHICO	KIRSKEN	POLVARS	COUNSCOT	GREENBLUE	NORTHENCAP
	Efficiency	Efficiency	Efficiency	Efficiency	Efficiency	Efficiency	Efficiency	Efficiency
2016	3	.163	.075	.089	.365	.503	.364	.241
	4	.164	.075	.091	.374	.498	.365	.243
	5	.167	.078	.093	.379	.509	.376	.252

Il·lustració 15: Exemple de consulta MDX utilitzant Jaspersoft OLAP Schema Workbench

A on es pot veure el resultat d'una consulta en forma de taula, segons l'esquema "OSWF.xml" (al desplegable de la barra superior) i utilitzant la connexió OSWF (connexió definida anteriorment al DWH utilitzant el paràmetres i el controladors per la base de dades "oswf" al MySQL, tal com es mostra a la següent figura:

The screenshot shows a 'Database Connection' dialog box with the following fields and values:

- Connection Name: OSWF
- Default Connection:
- Database Type: MySQL
- Driver Class Name: mysql.jdbc.Driver
- Host: localhost
- Port: (empty)
- Database Name: oswf
- Connection URL: jdbc:mysql://localhost/oswf
- User name: oswf
- Password:
- Schemas: (empty)
- Require Schema Attributes:

Buttons at the bottom: Delete, Clear, Test, Save.

Il·lustració 16: Dades de connexió a la base de dades del DWH

4. Anàlisi

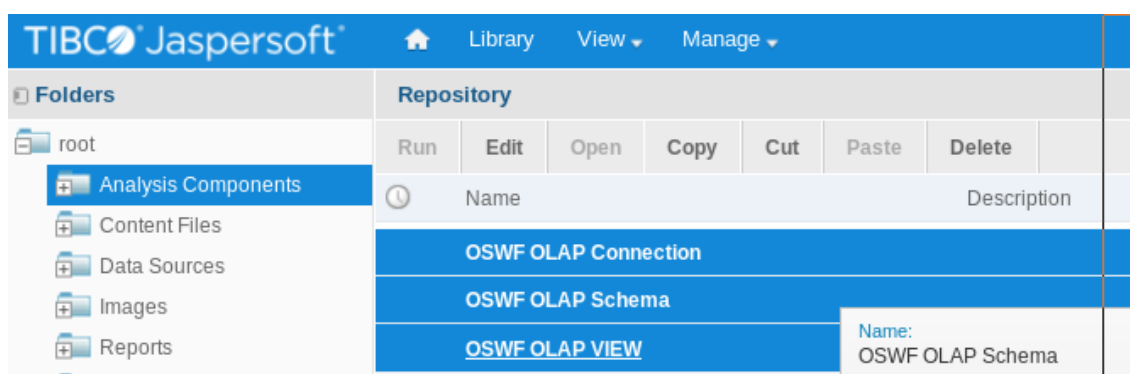
4.1 Vistes OLAP amb JasperReports

Per poder visualitzar i analitzar les dades del cub OLAP dels parcs eòlics utilitzarem, tal com ja hem avançat, el programa *Jaspersoft Server Community Edition*, més concretament utilitzarem les vistes OLAP (OLAP View) que són un component basat en el projecte *Mondrian* de *Pentaho*.

Un cop tenim instal·lat el servidor¹ de *Jaspersoft Server* necessitem uns quants ingredients per crear una vista OLAP (OLAP View):

- La font de dades: la base de dades del DWH
- Un esquema OLAP: l'esquema Mondrian definit al capítol anterior
- Un connexió Mondrian
- Una consulta MDX

La forma més senzilla es crear una vista des del *Jaspersoft Server* i el programa permet anar creant els altres elements. El resultat són un seguit d'elements al servidor tal com es pot veure a la figura següent:



Il·lustració 17: Elements necessaris per crear una vista OLAP amb JasperSoft Server

El elements estan definits per arxius xml, que es poden consultar al fer una exportació, tal com es pot veure a continuació:

¹ La instal·lació de JasperReports Server es realitza d'una forma guiada i llest per utilitzar.

Font de les dades

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbcDataSource exportedWithPermissions="false">
  <folder>/datasources</folder>
  <name>OSWF_DWH</name>
  <version>0</version>
  <label>OSWF DWH</label>
  <creationDate>2016-12-31T08:24:28.041+01:00</creationDate>
  <updateDate>2016-12-31T08:24:28.041+01:00</updateDate>
  <driver>org.mariadb.jdbc.Driver</driver>
  <connectionUrl>jdbc:mysql://localhost:3306/oswf?
tinyInt1isBit=false</connectionUrl>
  <connectionUser>oswf</connectionUser>
  <connectionPassword>XXXXXXXXXX</connectionPassword>
  <timezone></timezone>
</jdbcDataSource>
```

Connexió Mondrian

```
<?xml version="1.0" encoding="UTF-8"?>
<mondrianConnection exportedWithPermissions="false">
  <folder>/analysis</folder>
  <name>OSWF_OLAP_Connection</name>
  <version>4</version>
  <label>OSWF OLAP Connection</label>
  <description></description>
  <creationDate>2017-01-02T17:40:16.694+01:00</creationDate>
  <updateDate>2017-01-03T17:24:47.270+01:00</updateDate>
  <schema>
    <uri>/analysis/OSWF_OLAP_Schema2</uri>
  </schema>
  <dataSource>
    <uri>/datasources/OSWF_DWH</uri>
  </dataSource>
</mondrianConnection>
```

Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<fileResource exportedWithPermissions="false"
dataFile="OSWF_OLAP_Schema2.data">
  <folder>/analysis</folder>
  <name>OSWF_OLAP_Schema2</name>
  <version>1</version>
  <label>OSWF OLAP Schema</label>
  <description></description>
  <creationDate>2017-01-03T16:31:08.696+01:00</creationDate>
  <updateDate>2017-01-05T14:01:58.107+01:00</updateDate>
  <fileType>olapMondrianSchema</fileType>
</fileResource>
```

L'esquema Mondrian està contingut a **OSWF_OLAP_Schema2.data**

Vista OLAP

```
<?xml version="1.0" encoding="UTF-8"?>
<olapUnit exportedWithPermissions="false">
  <folder>/analysis</folder>
  <name>OSWF_OLAP_VIEW</name>
  <version>1</version>
  <label>OSWF OLAP VIEW</label>
  <description></description>
  <creationDate>2017-01-02T17:40:16.739+01:00</creationDate>
  <updateDate>2017-01-03T09:59:59.180+01:00</updateDate>
  <olapClientConnection>
    <uri>/analysis/OSWF_OLAP_Connection</uri>
  </olapClientConnection>
  <mdxQuery>WITH MEMBER [Measures].[Efficiency] AS '[Measures].
[Power]/ValidMeasure([Measures].[Nominal Power])',&#xd;
  FORMAT_STRING = '#.##0'&#xd;
select Crossjoin({[WindFarm].Children}, {[Measures].[Efficiency]}) ON
COLUMNS,&#xd;
  {[Date].[2016].Children} ON ROWS&#xd;
from [Alarms and Measures]&#xd;
</mdxQuery>
</olapUnit>
```

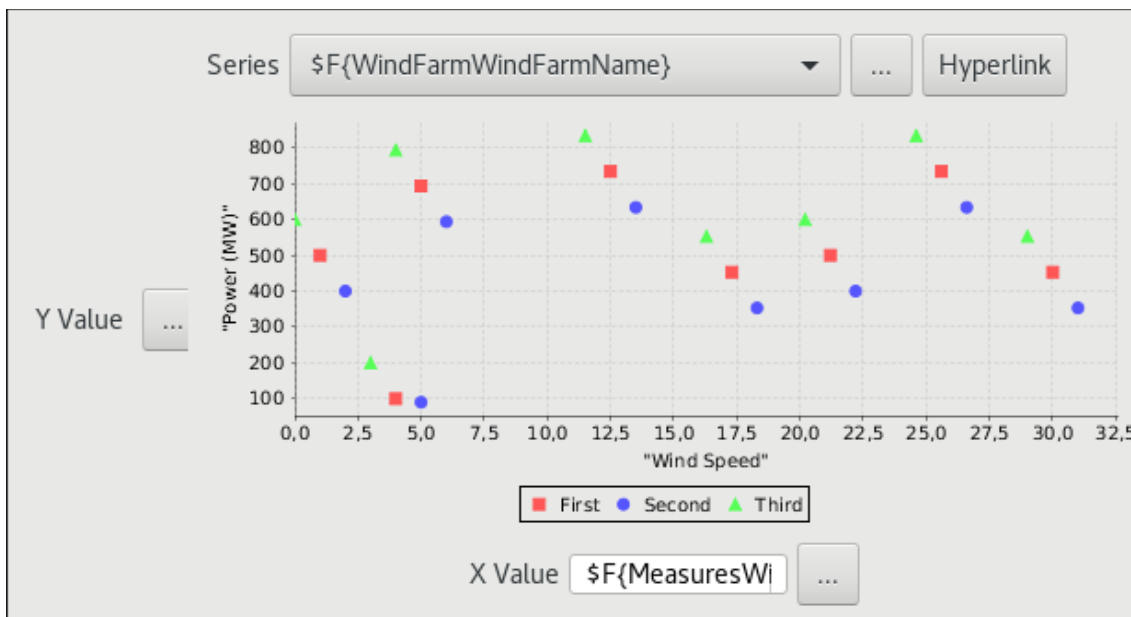
4.2 Gràfiques i informes amb JaspersoftStudio

Per la realització d'altres gràfics a part del gràfics de barres que suporta les vistes OLAP utilitzarem els informes de *Jaspersoft Studio*, que utilitza la llibreria *JFreeChart*.

Per exemple per la realització dels gràfics XY necessaris per veure la relació entre variables haurem de fer els següents passos:

- Crear un nou informe (*Report*).
- Seleccionar el cub OLAP com a font de dades, prèviament introduït.
- Definir la consulta MDX amb les mesures a les columnes i les sèries i punts com a fileres. Per exemple, la "velocitat del vent" i l'"energia produïda" a les columnes i els diferents parcs i el dia a les fileres.
- Seleccionar els camps de la consulta anterior que serviran com a variables de l'informe o gràfic.

- Afegir un gràfic tipus XY a la banda (o secció) "summary" seleccionant quines de les variables seran el que defineix la sèrie, el valor X i el valor Y. Es pot veure un exemple a la figura següent:



Il·lustració 18: Paràmetres de JasperReport(JFreeChart) per crear un gràfic XY

Amb tot el resultat en xml quedaria així (el resultat es pot consultar a la següent secció):

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Jaspersoft Studio v7.3.1.final using
JasperReports Library version 6.3.1 -->
<jasperReport
xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="Blank_A4-2" pageWidth="595" pageHeight="842" columnWidth="555"
leftMargin="20" rightMargin="20" topMargin="20" bottomMargin="20"
uuid="5b6a054f-090d-45c6-af1d-1bf544e9afa5">
  <property name="com.jaspersoft.studio.data.defaultdataadapter"
value="OSWF OLAP Connection"/>
  <queryString language="mdx">
    <![CDATA[select NON EMPTY {[Measures].[Wind Speed], [Measures].[Wave
Hight], [Measures].[Temperature], [Measures].[Power]} on columns,
NON EMPTY Crossjoin({[WindFarm].[All WindFarms].Children},
{Descendants([Date], [Day])}) ON rows from [Alarms and Measures]]]>
  </queryString>
  <field name="WindFarmWindFarmName" class="java.lang.String">
    <fieldDescription><![CDATA[Rows[WindFarm][WindFarm
Name]]]></fieldDescription>
  </field>
  <field name="MeasuresWaveHight" class="java.lang.Number">
    <fieldDescription><![CDATA[Data([Measures].[Wave
Hight],?)]]></fieldDescription>
  </field>
```



```

<field name="MeasuresTemperature" class="java.lang.Number">
  <fieldDescription><![CDATA[Data([Measures].
[Temperature],?)]]></fieldDescription>
</field>
<field name="MeasuresWindSpeed" class="java.lang.Number">
  <fieldDescription><![CDATA[Data([Measures].[Wind
Speed],?)]]></fieldDescription>
</field>
<field name="MeasuresPower" class="java.lang.Number">
  <fieldDescription><![CDATA[Data([Measures].
[Power],?)]]></fieldDescription>
</field>
<background>
  <band splitType="Stretch"/>
</background>
<summary>
  <band height="561" splitType="Stretch">
    <scatterChart>
      <chart evaluationTime="Report">
        <reportElement positionType="Float" x="0" y="10" width="550"
height="470" uuid="be93bec5-01af-4321-bd80-682d4e45dac1"/>
        <chartTitle/>
        <chartSubtitle/>
        <chartLegend/>
      </chart>
      <xyDataset>
        <xySeries autoSort="true">
          <seriesExpression><![
CDATA[ ${F{WindFarmWindFarmName}} ]]></seriesExpression>
          <xValueExpression><![
CDATA[ ${F{MeasuresWindSpeed}} ]]></xValueExpression>
          <yValueExpression><![
CDATA[ ${F{MeasuresPower}} ]]></yValueExpression>
        </xySeries>
      </xyDataset>
      <scatterPlot isShowLines="false">
        <plot/>
        <xAxisLabelExpression><![CDATA[ "Wind
Speed" ]]></xAxisLabelExpression>
        <xAxisFormat>
          <axisFormat/>
        </xAxisFormat>
        <yAxisLabelExpression><![CDATA[ "Power
(MW) " ]]></yAxisLabelExpression>
        <yAxisFormat>
          <axisFormat/>
        </yAxisFormat>
      </scatterPlot>
    </scatterChart>
  </band>
</summary>
</jasperReport>

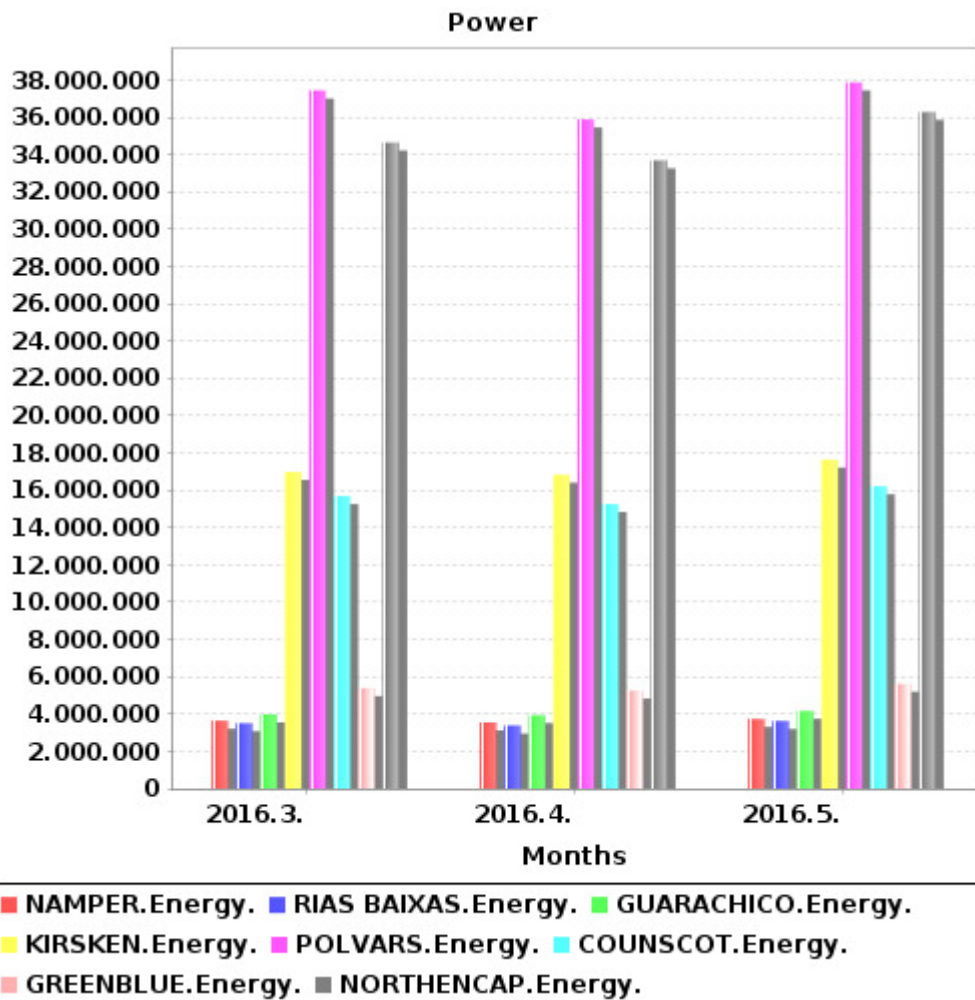
```

I de forma equivalent amb altres variables i camps. Per això als gràfics que mostraré a continuació només especificaré la `queryString`, `seriesExpression`, `xExpression` i `yExpression`.

4.3 Anàlisi dels parc eòlics més productius

Amb aquest anàlisi es vol trobar els parc eòlics més productius. Per això generarem una gràfica i una taula de l'energia produïda. Però per fer-ho més interessant afegirem el mes de l'any per veure si hi variacions temporals.

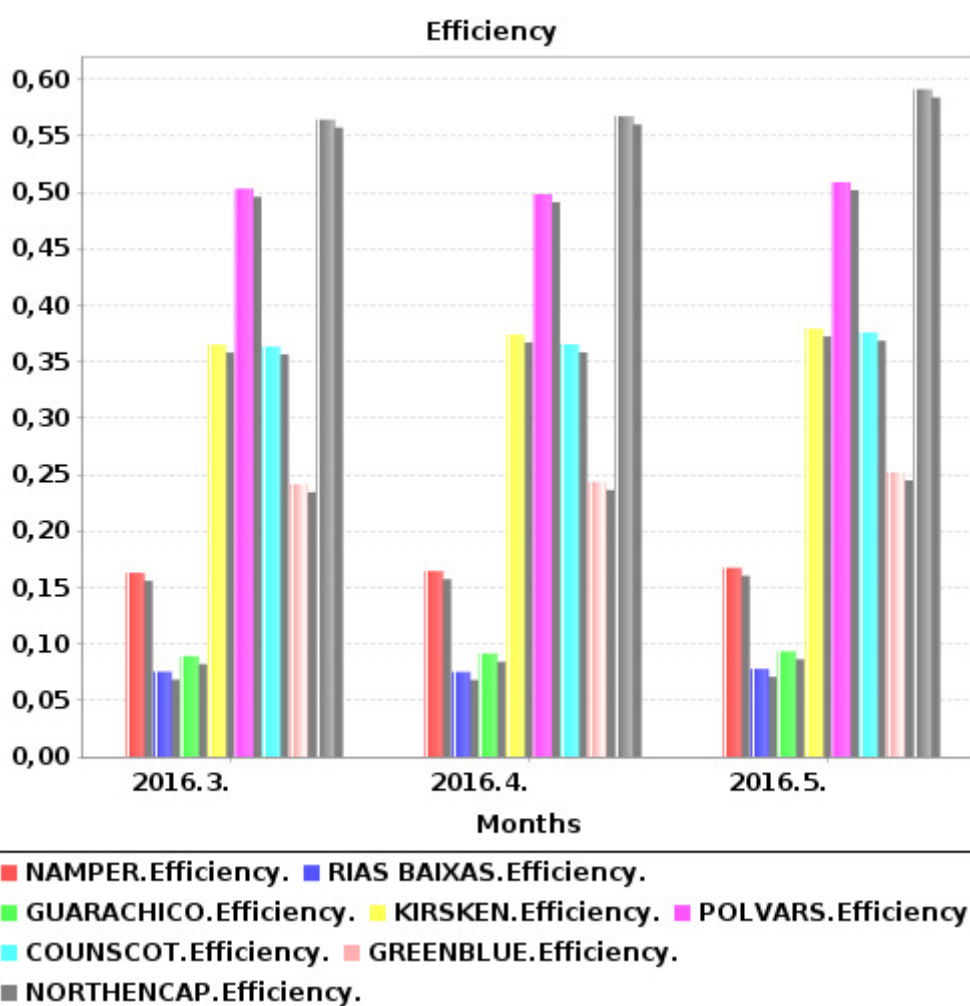
Dimensions			WindFarm							
			All WindFarms							
			NAMPER	IAS BAIXA	UARACHIC	KIRSKEN	POLVARS	COUNSCOT	GREENBLU	ORTHENCA
Date	Year	Month	Measures	Measures	Measures	Measures	Measures	Measures	Measures	Measures
			Energy	Energy	Energy	Energy	Energy	Energy	Energy	Energy
All Dates	2016	3	3,64E+09	3,51E+09	3,98E+09	16,98E+09	37,44E+09	15,69E+09	5,39E+09	34,65E+09
		4	3,55E+09	3,38E+09	3,94E+09	16,83E+09	35,89E+09	15,26E+09	5,26E+09	33,70E+06
		5	3,74E+09	3,62E+09	4,17E+09	17,65E+09	37,88E+09	16,21E+09	5,62E+09	36,29E+09



Il·lustració 19: Gràfica de la producció energètica dels parcs per mes

. A més a més de la producció total de cada parc he trobat interessant comparar també l'eficiència de la producció respecte la producció nominal, que presento a continuació.

Dimensions			WindFarm							
			All WindFarms							
			NAMPER	IAS BAIXA	JARACHIC	KIRSKEN	POLVARS	COUNSCOT	REENBLU	ORTHENC
Date	Year	Month	Measures	Measures	Measures	Measures	Measures	Measures	Measures	
			Efficiency	Efficiency	Efficiency	Efficiency	Efficiency	Efficiency	Efficiency	
All Dates	2016	3	.163	.075	.089	.365	.503	.364	.241	.565
		4	.164	.075	.091	.374	.498	.365	.243	.567
		5	.167	.078	.093	.379	.509	.376	.252	.591



Il·lustració 20: Gràfica de la eficiència de la producció energètica dels parcs per mes

Com es pot observar a les figures i taules anteriors, no hi ha diferències destacables al llarg dels mesos, però sí que hi ha diferències entre els parc com també entre la producció i l'eficiència.

El parc més productiu és el POLVARS, seguit de NORTHENCAP. Però si mirem l'eficiència es canvien els torns, amb NORTHENCAP amb una eficiència entre el 55% i el 60% mentre que POLVARS es queda al voltant del 50%. Per tant, si es volgués ampliar el parc seria més adequat ampliar el parc NORTHENCAP (a igualtat de cost dels aerogeneradors i potència nominal).

Les consultes MDX de les anàlisis anterior són:

Consulta de l'energia

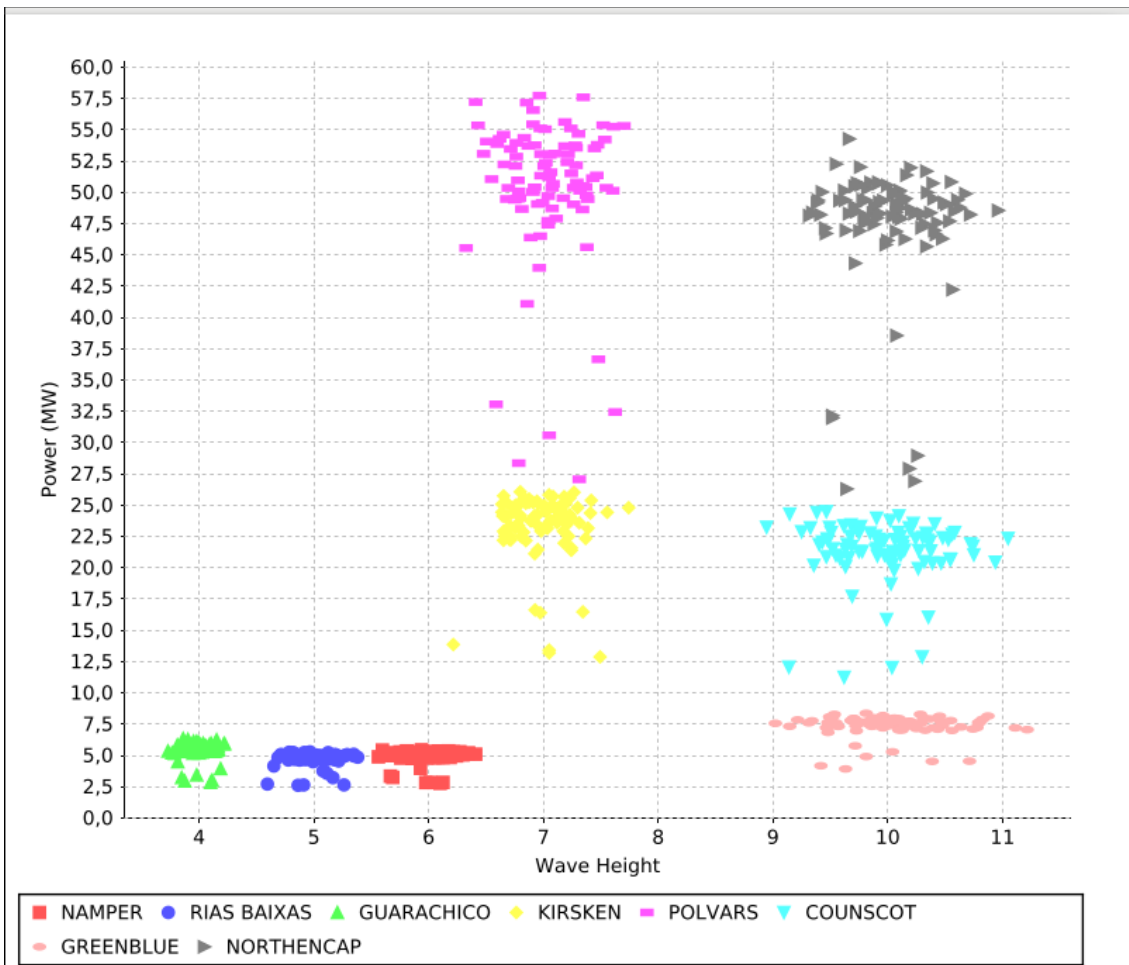
```
select
  Crossjoin({[WindFarm].Children}, {[Measures].[Energy]}) ON COLUMNS,
  {Descendants([Date], [Month])} ON ROWS
from [Alarms and Measures]
```

Consulta de l'eficiència

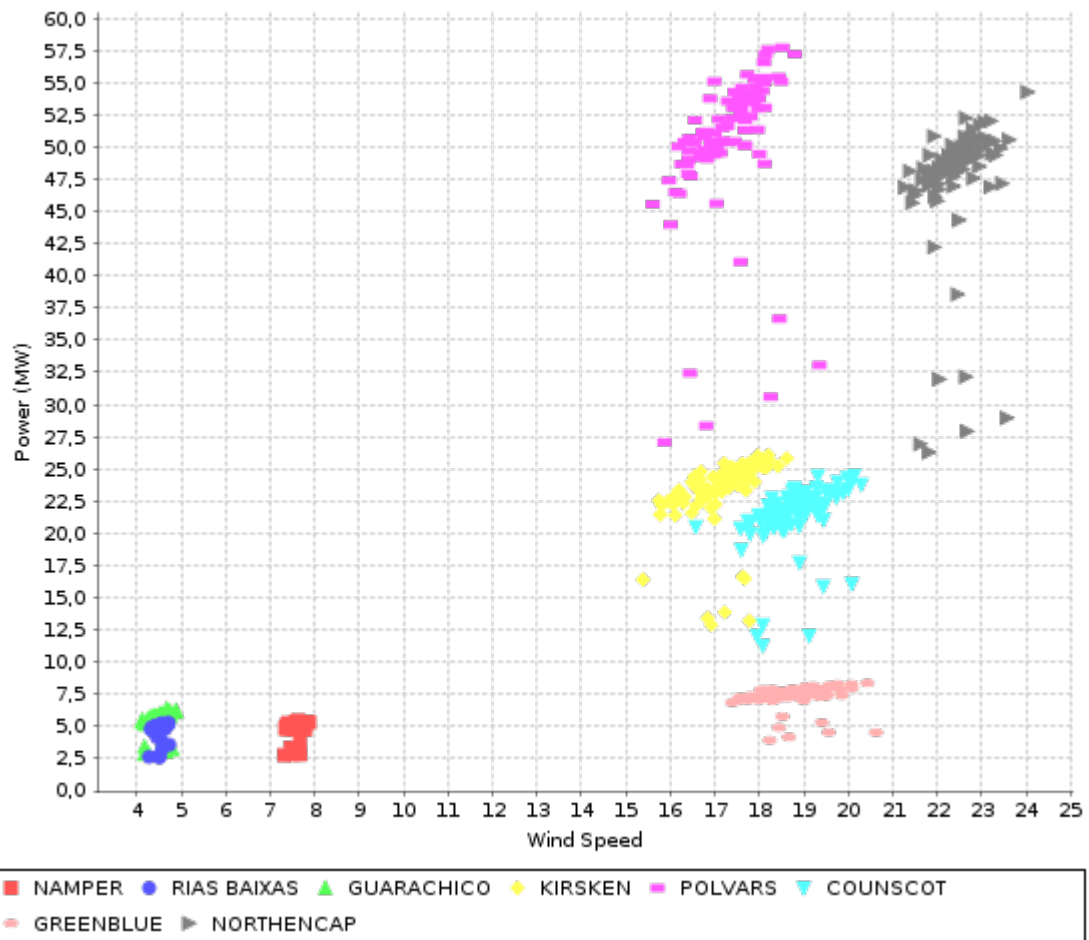
```
WITH
  MEMBER [Measures].[Efficiency]
  AS '[Measures].[Power]/ValidMeasure([Measures].[Nominal Power])',
  FORMAT_STRING = '#.##0'
select
  Crossjoin({[WindFarm].Children}, {[Measures].[Efficiency]}) ON COLUMNS,
  {Descendants([Date], [Month])} ON ROWS
from [Alarms and Measures]
```

4.4 Anàlisi de les dades meteorològiques i la producció

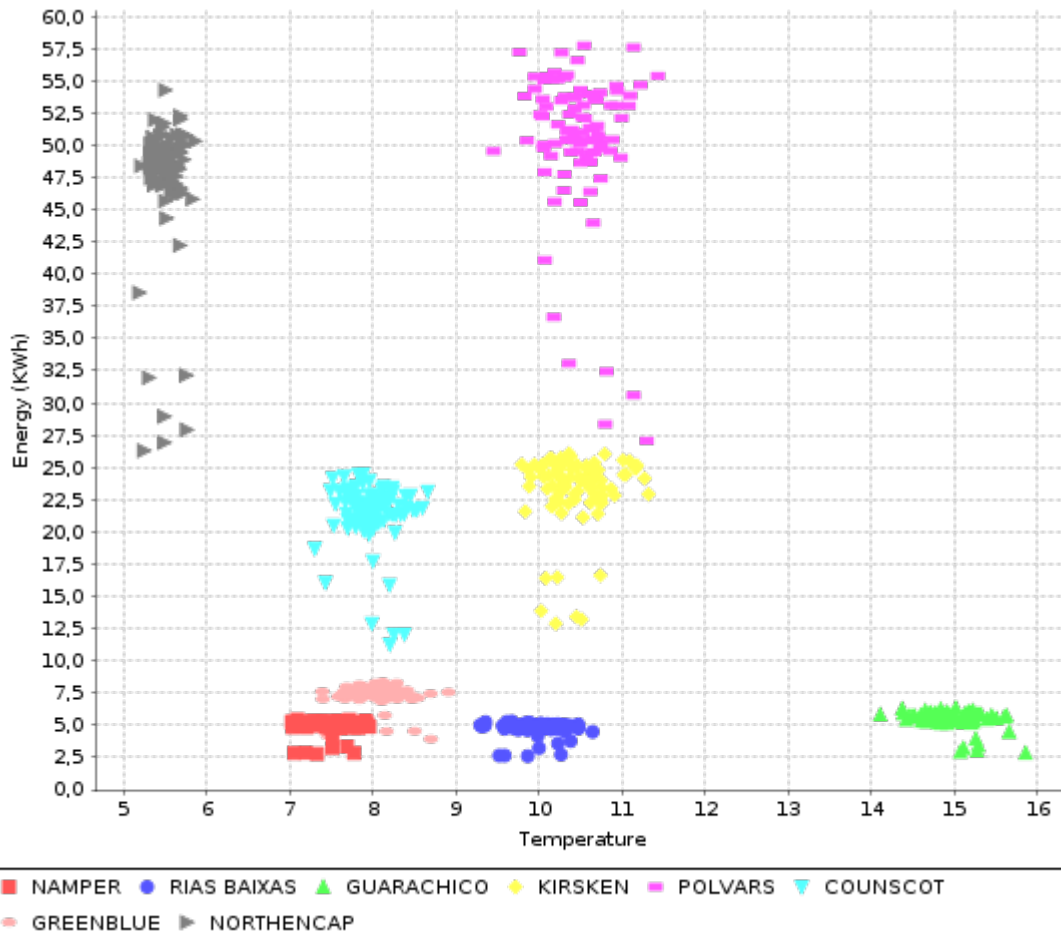
Per l'anàlisi de les dades meteorològiques i la producció he optat per fer gràfics de dispersió entre la producció d'energia i cadascuna de les variables meteorològiques de cada dia.



Il·lustració 21: Gràfic de la potència produïda en front de l'alçada de les ones



Il·lustració 22: Gràfic de la potència produïda en front de la velocitat del vent



Il·lustració 23: Gràfic de la potència produïda en front de la temperatura

Tal com es pot veure a les gràfiques anteriors la potència produïda té una relació aproximadament lineal en front de la velocitat del vent, tal com hom esperaria. Fins i tot es pot observar que la relació es estrictament proporcional, sense terme independent, o sigui que per una velocitat del vent zero la producció és zero, òbviament.

Per altra banda, es s'observa a les gràfiques de l'alçada de les ones i la temperatura que no hi ha relació entre elles i la producció d'energia.

Fora interessant estudiar amb més detall la possible causa de la dispersió de la potència, si aquesta es pot relacionar amb la disponibilitat del parc o el nombre d'alarmes. Però per això hauríem de fer una anàlisi multivariable que està fora de l'abast del projecte.

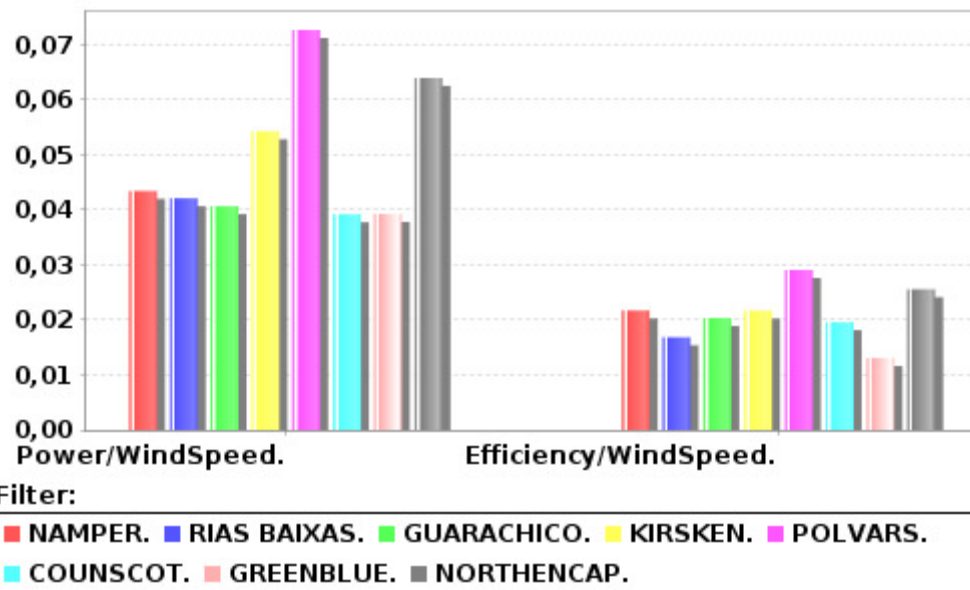
La consulta MDX de l'anàlisi anterior és:

```
select
  {[Measures].[Wind Speed], [Measures].[Wave Hight], [Measures].
    [Temperature], [Measures].[Power]} on columns,
  Crossjoin({[WindFarm].[All WindFarms].Children},
    {Descendants([Date], [Day])}) ON rows
from [Alarms and Measures]
```

4.5 Anàlisi de les zones amb millor relació vent/potència

A continuació es poden consultar l'anàlissobre la relació velocitat del vent i la potència generada. Les diferents zones les interpreto com la zona a on està situat el parc eòlic, però també es podria fer pels diferents països.

Dimensions		Measures	
WindFarm	WindFarm Name	Power/WindSpeed	Efficiency/WindSpeed
	NAMPER	.043	.022
	RIAS BAIXAS	.042	.017
	GUARACHICO	.041	.02
	KIRSKEN	.054	.022
	POLVARS	.073	.029
	COUNSCOT	.039	.02
	GREENBLUE	.039	.013
All WindFarms	NORTHENCAP	.064	.026



Il·lustració 24: Gràfic de la relació potència/velocitat del vent i potència/velocitat del vent pels diferents parcs

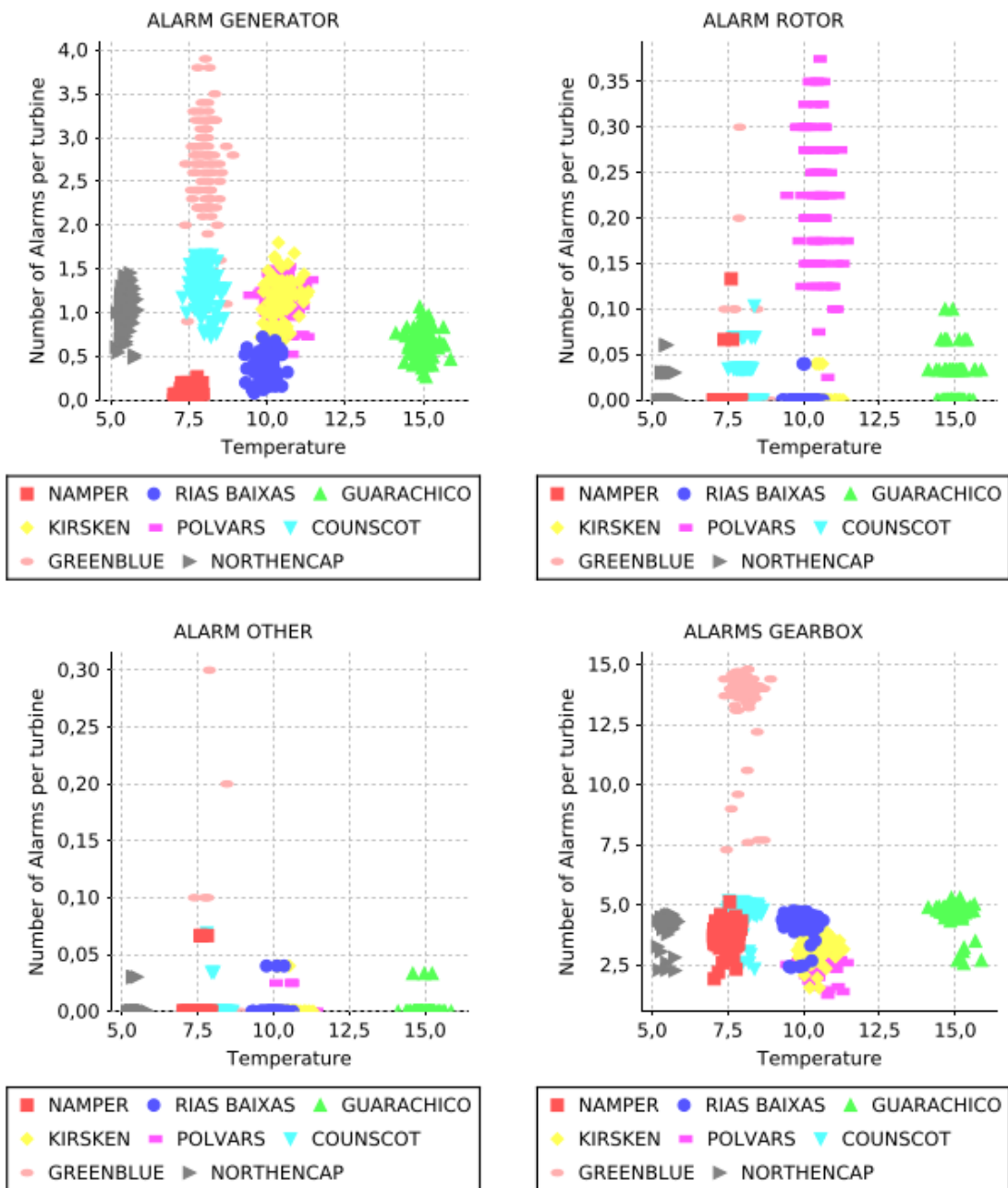
Igual que en el primer anàlisi, he considerat interessant fer l'estudi sobre l'eficiència a més a més de fer-ho sobre la producció. Aquí també es veu diferències entre una variable i l'altre. Els dos parcs amb millor relació potencia/vent (GREENBLUE i NORTHENCAP) també ho són en eficiència/vent, però les diferències amb els altres no són tan grans en el cas de mirar l'eficiència/vent. La resta de parc tenen una relació potència/vent similar exceptuant KIRSKEN, però en eficiència/vent la posicions son bastant diferents.

La consulta MDX de l'anàlisi anterior és:

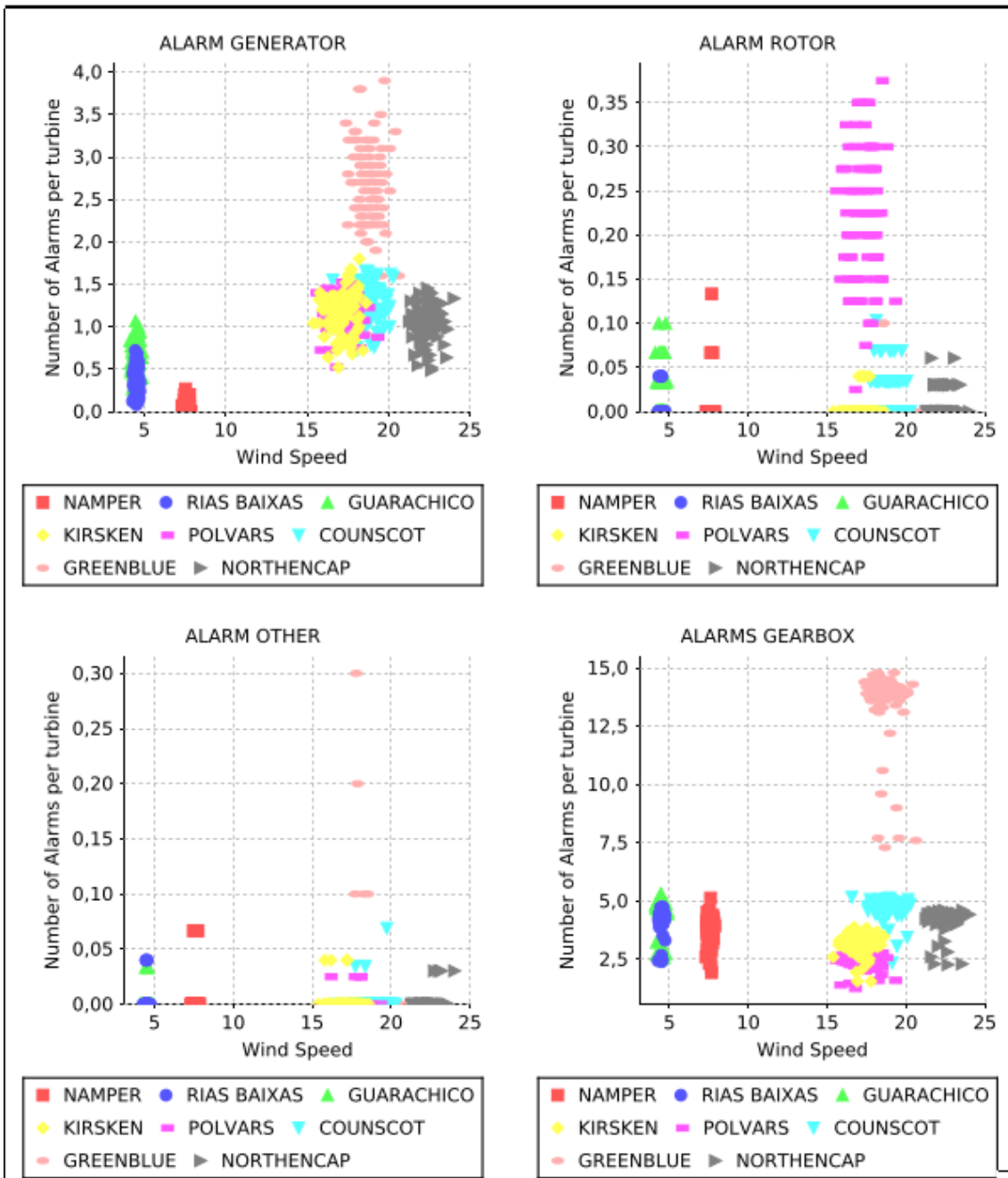
```
with member [Measures].[Power/WindSpeed] as '((([Measures].[Power] /  
ValidMeasure([Measures].[Number of Wind Turbines])) / [Measures].[Wind  
Speed])', FORMAT_STRING = "#.##0"  
    member [Measures].[Efficiency/WindSpeed] as '((([Measures].[Power] /  
ValidMeasure([Measures].[Nominal Power])) / [Measures].[Wind Speed])',  
FORMAT_STRING = "#.##0"  
select {[WindFarm].Children} ON COLUMNS,  
    {[Measures].[Power/WindSpeed], [Measures].[Efficiency/WindSpeed]} ON  
ROWS  
from [Alarms and Measures]
```

4.6 Anàlisi d'alarmes

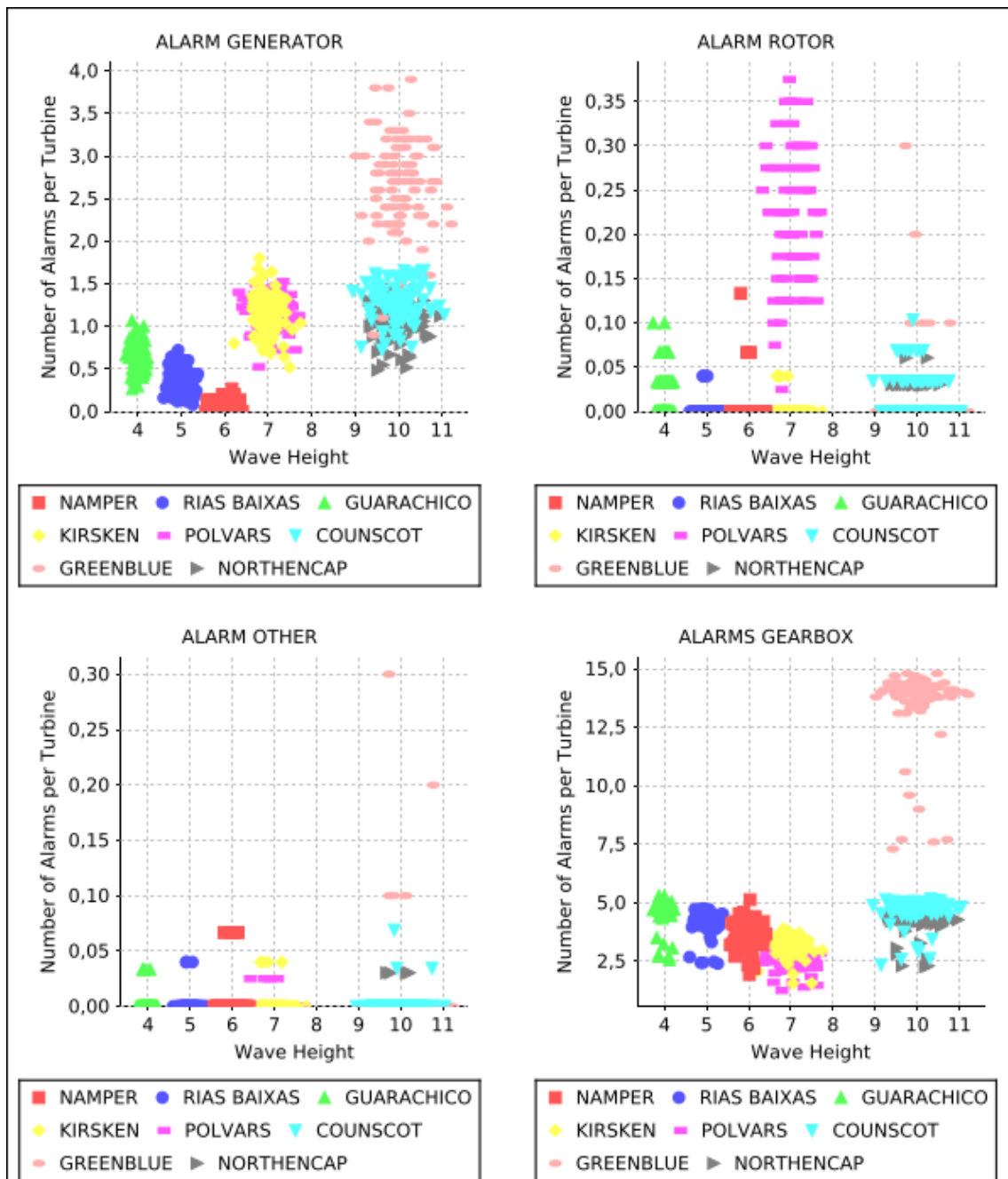
Per l'anàlisi de les alarmes i la seva dependència amb les condicions meteorològiques s'han realitzat un conjunt de gràfics de dispersió, del nombre d'alarmes ocorregudes al dia i la variable meteorològica (velocitat del vent, temperatura i alçada de les ones) per cada tipus d'alarma. Com que hi ha 4 tipus d'alarma i 3 variables meteorològica en total han sigut 12 gràfics de dispersió. A l'annex 3 es pot consultar la definició de l'informe de JasperReport amb els gràfics de dispersió del nombre d'alarmes per turbina en front de l'alçada de les ones, un gràfic per cada tipus d'alarma (utilitzant els grups de JasperReport).



Il·lustració 25: Gràfics del nombre d'alarmes per turbina en front de la temperatura per les diferents tipus d'alarmes



Il·lustració 26: Gràfics del nombre d'alarmes per turbina en front de la velocitat del vent per les diferents tipus d'alarmes



Il·lustració 27: Gràfics del nombre d'alarmes per turbina en front de l'alçada de les ones per les diferents tipus d'alarmes

Tal com es pot veure als gràfics no hi ha cap dependència entre el nombre d'alarmes i cap de les variables meteorològiques enlloc.

La consulta MDX de l'anàlisi anterior és:

```
with
member [Measures].[WH] AS 'ValidMeasure([Measures].[Wave Height])'
member [Measures].[WS] AS 'ValidMeasure([Measures].[Wind Speed])'
member [Measures].[T] AS 'ValidMeasure([Measures].[Temperature])'
member [Measures].[A/T] AS 'ValidMeasure([Measures].[
[Alarms])/ValidMeasure([Measures].[Number of Wind Turbines])'
```

```

select NON EMPTY
  {[Measures].[WH], [Measures].[WS], [Measures].[T], [Measures].
  [A/T]} on columns,
NON EMPTY
  Crossjoin({[Alarm].[All Alarms].Children},
    {Crossjoin({[WindFarm].[All WindFarms].Children},
      {Descendants([Date], [Day]))})}) ON rows
from [Alarms and Measures]

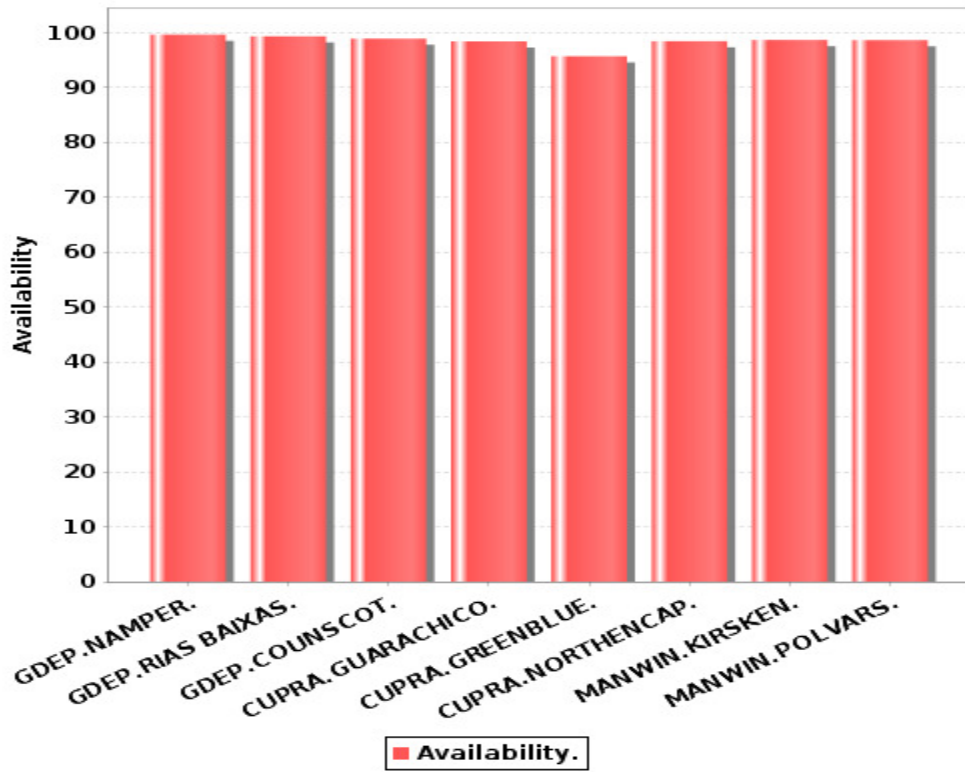
```

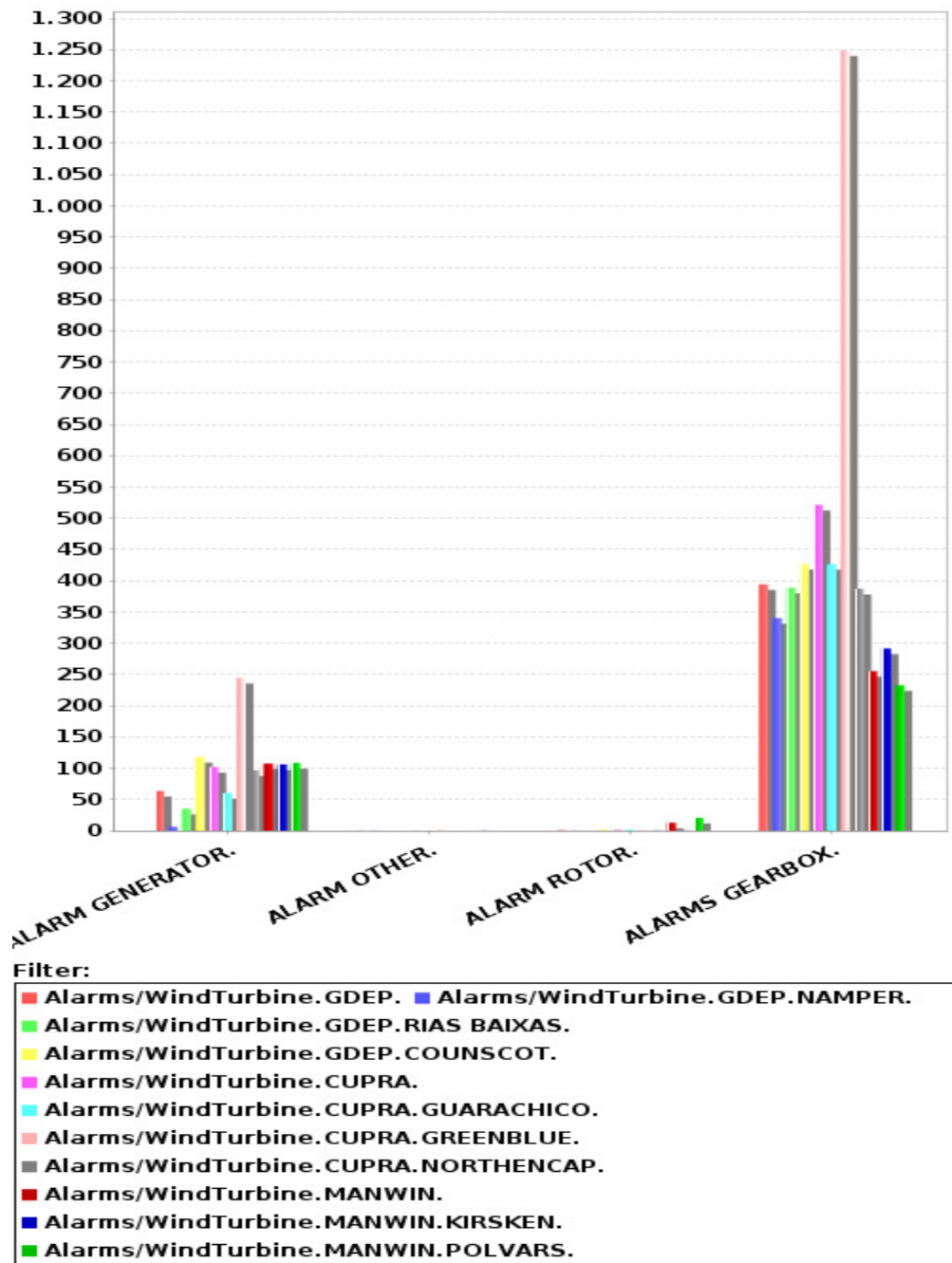
4.7 Anàlisi de la disponibilitat i les empreses de manteniment

Per aquest anàlisi s'ha optat per un simple gràfic de barres amb la disponibilitat i el nombre d'alarmes. Un aspecte a tenir en compte és que, com es va comentar al definir el cub, la mitja de la disponibilitat no es pot fer per diferents parcs ja que hauria de ser una mitja ponderada. Per aquesta raó, en el cas de la disponibilitat, no mostraré la seva agregació a nivell d'empresa de manteniment, sinó que simplement agruparé els parcs per empresa de manteniment.

WindFarm Maintainer	WindFarm Name	Availability
GDEP	NAMPER	99,64%
	RIAS BAIXAS	99,35%
	COUNSCOT	98,93%
CUPRA	GUARACHICO	98,41%
	GREENBLUE	95,71%
	NORTHENCAP	98,47%
MANWIN	KIRSKEN	98,66%
	POLVARS	98,64%

WindFarm		Alarms/WindTurbine			
Maintainer	Name	Generator	Other	Rotor	Gearbox
GDEP		63,58	0,16	0,80	394,28
GDEP	NAMPER	6,07	0,27	0,27	340,27
	RIAS BAIXAS	35,16	0,12	0,12	388,84
	COUNSCOT	117,83	0,14	1,66	426,90
CUPRA		101,84	0,21	1,01	521,34
CUPRA	GUARACHICO	60,17	0,10	1,43	426,57
	GREENBLUE	244,40	0,80	0,90	1248,90
	NORTHENCAP	96,52	0,12	0,67	387,03
MANWIN		107,48	0,12	12,72	255,42
MANWIN	KIRSKEN	105,92	0,16	0,16	291,56
	POLVARS	108,45	0,10	20,58	232,83





Tal com es pot veure a les taules i gràfics anteriors el mantenidor amb menys alarmes per turbina és MANWIN per les "Gearbox" i GDEP per les "Generator". També comentar que les variacions entre parcs són bastant importants.

En el cas de la disponibilitat tenim la que té millors disponibilitats és "GDEP" amb dos parcs amb disponibilitats més altes del 99% i l'altre a prop. El segon és MANWIN amb dos parcs al voltant de 98,6% de disponibilitat i per finalitzar CUPRA amb dos parcs al voltant de 98,4% i l'altre amb 95,7%.

Les consultes MDX de les anàlisis anteriors són:

Consulta de les alarmes per turbina

```

with member [Measures].[Alarms/WindTurbine] as
    '([Measures].[Alarms] /
        ValidMeasure([Measures].[Number of Wind Turbines]))',
    FORMAT_STRING = "#.##0"
select Crossjoin({[Measures].[Alarms/WindTurbine]},
    {[WindFarm.Maintainer].Children}) ON COLUMNS,
    {[Alarm].Children} ON ROWS
from [Alarms and Measures]

```

Consulta de les disponibilitat

```

select {[Measures].[Availability]} ON COLUMNS,
    {[WindFarm.Maintainer].[GDEP].Children,
        [WindFarm.Maintainer].[CUPRA].Children,
        [WindFarm.Maintainer].[MANWIN].Children} ON ROWS
from [Alarms and Measures]

```


5. Conclusions

El present treball de final de màster ha sigut una oportunitat per explorar l'àrea de *Business Intelligence*, un àrea cada cop més present i important. Així que les lliçons apreses durant la realització del mateix han sigut:

- Conceptes del *Business Intelligence*, com ara els DWH, els cubs OLAP i el processos ETL.
- Disposar de les dades operatives d'una organització no és suficient per prendre decisions, sinó que s'ha d'extreure la informació (les dades no són informació).
- Hi ha una sèrie de punts en tot projecte *Business Intelligence* de vital importància, entre ells el disseny del DWH, els cubs i el processos ETL, en especial poder obtenir dades d'alta qualitat i integrades.
- Encara que no s'ha tractat en aquest projecte també disposar de la *metadata* del DWH facilita i homogeneïtza l'ús de la solució de BI per tots el usuaris.
- Les solucions de BI disponibles al mercat són variades i madures. Es pot trobar la solució adequada per, podríem dir, tots els casos d'us. En el cas de la solució utilitzada en aquest projecte es tracta d'una solució que, amb la configuració adequada, poden cobrir la majoria de les necessitats de *Business Intelligence*, encara que en les versions utilitzades he trobat a faltar eines més interactiu i fàcils d'utilitzar.

El objectius del present treball s'han complert amb èxit:

1. S'ha dissenyat un magatzem de dades (*Data Warehouse*) que permet emmagatzemar la informació adquirida des dels parcs eòlics tant a nivell de producció com meteorològic.
2. S'ha implementat aquest magatzem de dades utilitzant *MySQL* i s'han programat els processos ETL (extracció, transformació i càrrega) el *Talend Open Source for Data Integration* que extreu les dades del fitxer xls i els guarda al DWH.
3. S'han comparat les diferents plataformes BI Open Source disponibles al mercat que permet explotar la informació emmagatzemada.
4. S'ha implementat la solució de *Business Intelligence Jaspersoft Server* i *Jaspersoft Studio* per la generació d'informes i taules per fer les anàlisis de la informació.

Per manca de temps ha sigut impossible realitzar totes les anàlisis del Pla de Treball, en particular la relació entre les variables de producció i meteorològiques. Aquest anàlisi és similar a altres anàlisis realitzades anteriorment però amb variables diferents.

Per altra banda, part de la raó de la manca de temps ha sigut el temps extra que he hagut d'utilitzar per diferents problemes durant la implantació, entre ells:

- *Talend Open Source for Data Integration* té problemes al llegir fitxers xls generats per *LibreOffice* (necessari per aplicar els canvis en l'enunciat sobre el fitxer de dades d'origen).
- Definició dels cubs amb *SpagoBIMeta*, la creació de l'esquema OLAP és incorrecte si les dimensions contenen el membre "Tots els membres". Falta d'una eina específica de definició de l'esquema OLAP i comprovar el seu funcionament correcte amb consultes MDX.
- Els errors retornats per *SpagoBI Server* al realitzar vistes OLAP amb Mondrian són difícils d'interpretar, com ara el següent missatge:
 - «Can't find resource for bundle org.eigenbase.resgen.ShadowResourceBundle\$MyPropertyResourceBundle, key CannotImplicitlyConvertDimensionToHierarchy»
- El canvi de solució escollida de *SpagoBI* a *Jaspersoft*.

La planificació del treball, sobretot en la seva part final, no s'ha complert. Inicialment s'havia planificat tenir llesta la implementació i els informes dos setmanes abans de la data límit. Degut als problemes anteriors i la manca de temps he hagut d'utilitzar aquestes últimes setmanes per la redacció en paral·lel de la memòria així com la generació dels informes. També considero que a la planificació faltava dedicar temps a familiaritzar-se amb els programes i llegir la documentació.

Aquest treball mostra els primers passos en un projecte de *Business Intelligence*, que es podria continuar disposant d'un quadre de comandament, solucions d'anàlisi més interactives, afegir una capa de metadades i integrar-ho amb altres àrees. També fora interessant explorar la escalabilitat de la solució, començant pels processos ETL fins a la generació dels gràfics e informes, així com la seva possible inclusió en altres sistemes.

6. Glossari

BBDD. Acrònim de base de dades.

BI. Business Intelligence. Defineix la intel·ligència de negoci com un un conjunt d'estratègies encarats a l'administració i creació de coneixement sobre l'àmbit d'una em presa o organització, a partir de l'anàlisi de les dades existents en aquesta.

OLAP. Online Analytical Processing. OLAP és una tecnologia o solució que permet fer consultes a una base de dades d'una forma ràpida i eficaç. Es basa en la construcció d'unes estructures multidimensionals anomenades Cubs OLAP que defineixen les diferents consultes que es poden realitzar

Data Warehouse. Magatzem de dades, Un data warehouse és una col·lecció de informació creada per donar suport a les aplicacions per la presa de decisions

ETL. Extract, Transform, Load. Procès amb el qual s'extrau, es transforma i es carreguen, típicament, a una base de dades.

Jaspersoft Schema Workbench. És una eina de Jaspersoft que permet editar cubs OLAP.

Jpivot. És una llibreria de JSP renderitza taules OLAP i permet als usuaris realitzar cerques i navegar a través de cubs i estructures OLAP.

MDX. Multidimensional expressions. És un llenguatge de consultes per a bases de dades multidimensionals sobre cubs OLAP.

MySQL. És un sistema de gestió de bases de dades (SGBD) relacional, multifil i multiusuari d'Oracle amb llicència dual.

Jaspersoft Server. Part servidor de la solució de *Business Intelligence* de Jaspersoft.

Jaspersoft Studio. Aplicació client de la solució de BI de Jaspersoft per la generació d'informes.

SGBD. Sistema de Gestió de Bases de Dades. Software que serveix per crear, consultar, gestionar i modificar bases de dades.

XML. Extensible Markup Language. És un llenguatge de marques utilitzat per emmagatzemar dades en forma llegible.

7. Bibliografia

C. Ballard, *et al.* *Data Modeling Techniques for Data Warehouse*. International Technical Support Organization, 1998.

<http://www.redbooks.ibm.com/redbooks/pdfs/sg242238.pdf>

A. Slaughter. *OLAP*. <http://web.mit.edu/profit/PDFS/SlaughterA.pdf>

J. Cano. *Business Intelligence: Competir con información*. ESADE. 2008.

J. Parenteau, *et al.* Magic Quadrant for Business Intelligence and Analytics Platforms <https://www.gartner.com/doc/reprints?id=1-2XXET8P&ct=160204>

5 Open Source Business Intelligence Tools.

<http://www.enterpriseappstoday.com/business-intelligence/slideshows/5-open-source-business-intelligence-tools.html> Data 21/8/2015 [Visitat: 29/3/2016]

Top 3 open source business intelligence and reporting tools

<https://opensource.com/business/14/6/three-open-source-business-tools>
Data 23/6/2014 [Visitat: 29/3/2016]

Mondrian Schema

<http://mondrian.pentaho.com/documentation/schema.php> [Visitat 23/12/2016]

Jaspersoft OLAP Schema Workbench

<http://community.jaspersoft.com/wiki/jaspersoft-olap-schema-workbench>
[Visitat 28/12/2016]

Jasperreports Server <http://community.jaspersoft.com/project/jasperreports-server> [Visitat 28/12/2016]

Mondrian

<http://community.pentaho.com/projects/mondrian/> [Visitat 23/12/2016]

JFreeChart

<http://www.jfree.org/jfreechart/> [Visitat 28/12/2016]

SpagoBI

<http://www.spagobi.org/> [Visitat 29/3/2016]

8. Annexos

Annex 1 Guió SQL per la creació del DWH

```
-- MySQL Script generated by MySQL Workbench
-- Model: Offshow Wind Farm   Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-----
-- Schema oswf
-----

-----
-- Schema oswf
-----

CREATE SCHEMA IF NOT EXISTS `oswf` DEFAULT CHARACTER SET utf8 ;
USE `oswf` ;

-----
-- Table `oswf`.`d_date`
-----

CREATE TABLE IF NOT EXISTS `oswf`.`d_date` (
  `id_date` INT(10) NOT NULL,
  `day` DATE NOT NULL,
  `month` INT(2) NOT NULL,
  `year` INT(4) NOT NULL,
  PRIMARY KEY (`id_date`));

-----
-- Table `oswf`.`d_oswf`
-----

CREATE TABLE IF NOT EXISTS `oswf`.`d_oswf` (
  `id_oswf` INT(10) NOT NULL,
  `name` VARCHAR(100) NOT NULL,
```

```

`country` VARCHAR(100) NOT NULL,
`country_name` VARCHAR(100) NOT NULL,
`power` FLOAT NOT NULL,
`number` INT NOT NULL,
`total_power` FLOAT NOT NULL,
`operator` INT(10) NOT NULL,
`operator_name` VARCHAR(100) NOT NULL,
`maintainer` INT(10) NOT NULL,
`maintainer_name` VARCHAR(100) NOT NULL,
PRIMARY KEY (`id_oswf`),
UNIQUE INDEX `name_UNIQUE` (`name` ASC));

```

```

-----
-- Table `oswf`.`h_oswf_measure`
-----

```

```

CREATE TABLE IF NOT EXISTS `oswf`.`h_oswf_measure` (
  `id_oswf_measure` INT(10) NOT NULL AUTO_INCREMENT,
  `d_date` INT(10) NOT NULL,
  `d_oswf` INT(10) NOT NULL,
  `m_power` FLOAT NOT NULL,
  `m_wind_speed` FLOAT NOT NULL,
  `m_temp` FLOAT NOT NULL,
  `m_availability` FLOAT NOT NULL,
  `m_wave_hight` FLOAT NOT NULL,
  `m_reparation_time` FLOAT NOT NULL,
  PRIMARY KEY (`id_oswf_measure`),
  INDEX `fk_861b21fc-03c3-11e6-8e97-0800278a5045` (`d_date` ASC),
  INDEX `fk_861b42c2-03c3-11e6-8e97-0800278a5045` (`d_oswf` ASC),
  CONSTRAINT `fk_861b21fc-03c3-11e6-8e97-0800278a5045`
    FOREIGN KEY (`d_date`)
    REFERENCES `oswf`.`d_date` (`id_date`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
  CONSTRAINT `fk_861b42c2-03c3-11e6-8e97-0800278a5045`
    FOREIGN KEY (`d_oswf`)
    REFERENCES `oswf`.`d_oswf` (`id_oswf`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT)
ENGINE = InnoDB;

```

```

-----
-- Table `oswf`.`d_alarm_type`
-----

```

```

CREATE TABLE IF NOT EXISTS `oswf`.`d_alarm_type` (
  `id_alarm_type` INT(10) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id_alarm_type`))
ENGINE = InnoDB;

```

```
-----  
-- Table `oswf`.`h_oswf_alarm`  
-----
```

```
CREATE TABLE IF NOT EXISTS `oswf`.`h_oswf_alarm` (  
  `id_oswf_alarm` INT NOT NULL AUTO_INCREMENT,  
  `d_date` INT(10) NOT NULL,  
  `d_alarm_type` INT(10) NOT NULL,  
  `d_oswf` INT(10) NOT NULL,  
  `m_alarms` INT NULL,  
  PRIMARY KEY (`id_oswf_alarm`),  
  INDEX `fk_h_oswf_alarm_d_date1_idx` (`d_date` ASC),  
  INDEX `fk_h_oswf_alarm_d_alarm1_idx` (`d_alarm_type` ASC),  
  INDEX `fk_h_oswf_alarm_d_oswf1_idx` (`d_oswf` ASC),  
  CONSTRAINT `fk_h_oswf_alarm_d_date1`  
    FOREIGN KEY (`d_date`)  
    REFERENCES `oswf`.`d_date` (`id_date`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_h_oswf_alarm_d_alarm1`  
    FOREIGN KEY (`d_alarm_type`)  
    REFERENCES `oswf`.`d_alarm_type` (`id_alarm_type`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_h_oswf_alarm_d_oswf1`  
    FOREIGN KEY (`d_oswf`)  
    REFERENCES `oswf`.`d_oswf` (`id_oswf`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `oswf`.`id_oswf`  
-----
```

```
CREATE TABLE IF NOT EXISTS `oswf`.`id_oswf` (  
  `id_oswf` INT(10) NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(100) NULL,  
  PRIMARY KEY (`id_oswf`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `oswf`.`id_country`  
-----
```

```
CREATE TABLE IF NOT EXISTS `oswf`.`id_country` (  
  `id_country` INT(10) NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(100) NULL,  
  PRIMARY KEY (`id_country`))
```



```
ENGINE = InnoDB;
```

```
-----  
-- Table `oswf`.`id_operator`  
-----
```

```
CREATE TABLE IF NOT EXISTS `oswf`.`id_operator` (  
  `id_operator` INT(10) NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(100) NULL,  
  PRIMARY KEY (`id_operator`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `oswf`.`id_maintainer`  
-----
```

```
CREATE TABLE IF NOT EXISTS `oswf`.`id_maintainer` (  
  `id_maintainer` INT(10) NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(100) NULL,  
  PRIMARY KEY (`id_maintainer`))  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Annex 2 Schema OLAP del OSWF

```
<?xml version="1.0"?>  
<Schema name="OSWF">  
  <Dimension name="WindFarm">  
    <Hierarchy hasAll="true" primaryKey="id_oswf">  
      <Table name="d_oswf"/>  
      <Level name="WindFarm Name" column="id_oswf" nameColumn="name"  
uniqueMembers="true"/>  
    </Hierarchy>  
    <Hierarchy name="Country" allMemberName="All Countries"  
hasAll="true" primaryKey="id_oswf">  
      <Table name="d_oswf"/>  
      <Level name="WindFarm Country" column="country"  
nameColumn="country_name" uniqueMembers="true"/>  
      <Level name="WindFarm Name" column="id_oswf" nameColumn="name"  
uniqueMembers="true"/>  
    </Hierarchy>  
    <Hierarchy name="Operator" allMemberName="All Operators"  
hasAll="true" primaryKey="id_oswf">  
      <Table name="d_oswf"/>  
    </Hierarchy>  
  </Dimension>  
</Schema>
```

```

    <Level name="WindFarm Operator" column="operator"
nameColumn="operator_name" uniqueMembers="true"/>
    <Level name="WindFarm Name" column="id_oswf" nameColumn="name"
uniqueMembers="true"/>
  </Hierarchy>
  <Hierarchy name="Maintainer" allMemberName="All Maintainer"
hasAll="true" primaryKey="id_oswf">
    <Table name="d_oswf"/>
    <Level name="WindFarm Maintainer" column="maintainer"
nameColumn="maintainer_name" uniqueMembers="true"/>
    <Level name="WindFarm Name" column="id_oswf" nameColumn="name"
uniqueMembers="true"/>
  </Hierarchy>
</Dimension>
<Dimension name="Date">
  <Hierarchy hasAll="true" primaryKey="id_date">
    <Table name="d_date"/>
    <Level name="Year" column="year" uniqueMembers="false"/>
    <Level name="Month" column="month" uniqueMembers="false"/>
    <Level name="Day" column="day" uniqueMembers="true"/>
  </Hierarchy>
</Dimension>
<Dimension name="Alarm">
  <Hierarchy hasAll="true" primaryKey="id_alarm_type">
    <Table name="d_alarm_type"/>
    <Level name="Name" column="name" uniqueMembers="true"/>
  </Hierarchy>
</Dimension>
<Cube name="OSWFWindFarms">
  <Table name="d_oswf"/>
  <DimensionUsage name="WindFarm" source="WindFarm"/>
  <Measure name="Nominal Power" column="total_power"
aggregator="sum" formatString="#.###"/>
  <Measure name="Number of Wind Turbines" column="number"
aggregator="sum" formatString="#.###"/>
</Cube>
<Cube name="OSWFAlarms">
  <Table name="h_oswf_alarm"/>
  <DimensionUsage name="WindFarm" source="WindFarm"
foreignKey="d_oswf"/>
  <DimensionUsage name="Date" source="Date" foreignKey="d_date"/>
  <DimensionUsage name="Alarm" source="Alarm"
foreignKey="d_alarm_type"/>
  <Measure name="Alarms" column="m_alarms" aggregator="sum"
formatString="#.###"/>
</Cube>
<Cube name="OSWFMeasures">
  <Table name="h_oswf_measure"/>
  <DimensionUsage name="WindFarm" source="WindFarm"
foreignKey="d_oswf"/>
  <DimensionUsage name="Date" source="Date" foreignKey="d_date"/>
  <Measure name="Energy" column="m_power" aggregator="sum"
formatString="#.###"/>
  <Measure name="EnergyAvg" column="m_power" aggregator="avg"
visible="false" formatString="#.###"/>
  <Measure name="Wind Speed" column="m_wind_speed" aggregator="avg"
formatString="#.###"/>
  <Measure name="Temperature" column="m_temp" aggregator="avg"
formatString="#.###"/>

```

```

    <Measure name="Availability" column="m_availability"
aggregator="avg" formatString="#.###"/>
    <Measure name="Wave Hight" column="m_wave_high" aggregator="avg"
formatString="#.###"/>
    <Measure name="Reparation Time" column="m_reparation_time"
aggregator="sum" formatString="#.###"/>
    <CalculatedMember name="Power" dimension="Measures">
    <Formula>[Measures].[EnergyAvg] / 24 / 1000 </Formula>
    </CalculatedMember>
</Cube>
<VirtualCube name="Alarms and Measures">
    <CubeUsages>
    <CubeUsage cubeName="OSWFWindFarms"
ignoreUnrelatedDimensions="true"/>
    <CubeUsage cubeName="OSWFMeasures"
ignoreUnrelatedDimensions="true"/>
    <CubeUsage cubeName="OSWFAlarms"/>
    </CubeUsages>
    <VirtualCubeDimension name="WindFarm"/>
    <VirtualCubeDimension name="Date"/>
    <VirtualCubeDimension name="Alarm"/>
    <VirtualCubeMeasure cubeName="OSWFWindFarms" name="[Measures].
[Nominal Power]"/>
    <VirtualCubeMeasure cubeName="OSWFWindFarms" name="[Measures].
[Number of Wind Turbines]"/>
    <VirtualCubeMeasure cubeName="OSWFAlarms" name="[Measures].
[Alarms]"/>
    <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].
[Energy]"/>
    <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].
[Power]"/>
    <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].[Wind
Speed]"/>
    <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].
[Temperature]"/>
    <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].
[Availability]"/>
    <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].[Wave
Hight]"/>
    <VirtualCubeMeasure cubeName="OSWFMeasures" name="[Measures].
[Reparation Time]"/>
    </VirtualCube>
</Schema>

```

Annex 3 JasperReport amb gràfics de dispersió per cada tipus d'alarma

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Jaspersoft Studio version 6.3.1.final using
JasperReports Library version 6.3.1 -->
<jasperReport
xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="R4_relation_alarms_wave_height" columnCount="2" pageWidth="595"
pageHeight="842" columnWidth="277" leftMargin="20" rightMargin="20"
topMargin="20" bottomMargin="20" uuid="d1992e01-4053-4b5f-a155-
be01e09c3ec5">
  <property name="com.jaspersoft.studio.data.defaultdataadapter"
value="OSWF OLAP Connection"/>
  <queryString language="mdx">
    <![CDATA[with member [Measures].[WH] AS 'ValidMeasure([Measures].
[Wave Hight])'
member [Measures].[WS] AS 'ValidMeasure([Measures].[Wind Speed])'
member [Measures].[T] AS 'ValidMeasure([Measures].[Temperature])'
member [Measures].[A/T] AS 'ValidMeasure([Measures].
[Alarms])/ValidMeasure([Measures].[Number of Wind Turbines])'
select NON EMPTY {[Measures].[WH], [Measures].[WS], [Measures].[T],
[Measures].[A/T]} on columns,
NON EMPTY Crossjoin({[Alarm].[All Alarms].Children},
{Crossjoin({[WindFarm].[All WindFarms].Children}, {Descendants([Date],
[Day]))})}) ON rows
from [Alarms and Measures]]]>
  </queryString>
  <field name="MeasuresWS" class="java.lang.Number">
    <fieldDescription><![CDATA[Data([Measures].
[WS],?)]]></fieldDescription>
  </field>
  <field name="AlarmName" class="java.lang.String">
    <fieldDescription><![CDATA[Rows[Alarm][Name]]]></fieldDescription>
  </field>
  <field name="MeasuresWH" class="java.lang.Number">
    <fieldDescription><![CDATA[Data([Measures].
[WH],?)]]></fieldDescription>
  </field>
  <field name="MeasuresT" class="java.lang.Number">
    <fieldDescription><![CDATA[Data([Measures].
[T],?)]]></fieldDescription>
  </field>
  <field name="WindFarmWindFarmName" class="java.lang.String">
    <fieldDescription><![CDATA[Rows[WindFarm][WindFarm
Name]]]></fieldDescription>
  </field>
  <field name="MeasuresAlarms" class="java.lang.Number">
    <fieldDescription><![CDATA[Data([Measures].
[A/T],?)]]></fieldDescription>
  </field>
  <group name="GroupAlarmName">
    <groupExpression><![CDATA[$F{AlarmName}]]></groupExpression>
    <groupHeader>
      <band height="319">
        <scatterChart>
          <chart evaluationTime="Group" evaluationGroup="GroupAlarmName">
            <reportElement x="10" y="10" width="250" height="300"
uuid="da92ae2b-28cb-4c91-87ce-7358c26390f7"/>
            <chartTitle>
              <titleExpression><![CDATA[$F{AlarmName}]]></titleExpression>
            </chartTitle>
            <chartSubtitle/>
            <chartLegend/>
          </chart>
        </scatterChart>
      </band>
    </groupHeader>
  </group>

```

```

    <dataset resetType="Group" resetGroup="GroupAlarmName"/>
    <xySeries autoSort="true">
      <seriesExpression><!
[CDATA[ ${F{WindFarmWindFarmName}} ]></seriesExpression>
      <xValueExpression><![CDATA[ ${F{MeasuresWH}} ]></xValueExpression>
      <yValueExpression><![
CDATA[ ${F{MeasuresAlarms}} ]></yValueExpression>
    </xySeries>
  </xyDataset>
  <scatterPlot isShowLines="false">
    <plot/>
    <xAxisLabelExpression><![CDATA[ "Wave
Height" ]></xAxisLabelExpression>
    <xAxisFormat>
      <axisFormat/>
    </xAxisFormat>
    <yAxisLabelExpression><![CDATA[ "Number of Alarms per
Turbine" ]></yAxisLabelExpression>
    <yAxisFormat>
      <axisFormat/>
    </yAxisFormat>
  </scatterPlot>
</scatterChart>
</band>
</groupHeader>
</group>
</jasperReport>

```