UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

TECNOLOGÍA ESPECÍFICA DE

INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE GRADO

“Uso de tecnología CEP para la detección de desgaste en aerogeneradores”

Enrique Brazález Segovia

Junio de 2018

UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

TECNOLOGÍA ESPECÍFICA DE

INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE GRADO

Uso de tecnología CEP para la detección de desgaste en aerogeneradores

Autor: Enrique Brazález Segovia.

Directores: Gregorio Diáz Descalzo.

Junio 2018

A todos aquellos que me han ayudado y formado parte de estos 4 años de carrera.

Gracias.

Declaración de Autoría

Yo, Enrique Brazález Segovia, con DNI 74519608E, declaro que soy el único autor del Trabajo Fin de Grado titulado “Uso de tecnología CEP para la detección de desgaste en aerogeneradores” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está propiadamente atribuido a sus legítimos autores.

Albacete, a martes, 3 de abril de 2018.

Fdo.: Enrique Brazález Segovia.

Resumen

Internet of Things es un concepto tecnológico que está revolucionando la forma de concibir nuestro día a día estos últimos años. La integración de sistemas informáticos en nuestros hogares, en nuestros puestos de trabajo, en la industria es muy notable, y en algunos casos, vital. Son muchas las tecnologías que toman participación en este concepto, y una de ellas es “Complex Event Processing” o “Procesamiento de Eventos Complejos” (CEP).

CEP, es una nueva forma de analizar datos. Una aplicación de Fast Data, que dota a los sistemas que lo soportan de una capacidad de sintésis, y análisis de datos abrumadora. Como toda tecnología tiene sus ventajas y desventajas, pero sin duda alguna, hemos de destacar que el procesamiento de eventos complejos dota de una de las capacidades de análisis más rápidas del mercado, permitiendo la toma de decisiones a tiempo real.

El escenario en el que nos encontramos en este trabajo es el de las energías renovables, en concreto, la energía eólica. A rasgos generales, el funcionamiento de un molino de viento es muy sencillo, a través de la fuerza que el viento ejerce sobre sus palas se generá energía mecánica sobre el rotor, y este a su vez electricidad a través de un generador. Lo que intentaremos en este trabajo de fin de grado es aplicar CEP, para la optimización de obtención de energía eléctrica a través aerogeneradores.

En este trabajo de fin de grado se desarrolla un sistema de predicción del desgaste en aerogeneradores en base a los datos obtenidos por su sistema de monitorización. El objetivo del sistema a desarrollar es, utilizando el procesamiento de eventos complejos y tecnologías de código abierto, implementar un sistema que nos permita predecir cuando el comportamiento los aerogeneradores dentro de un parque eólico es anómalo. Por tanto, aquel aerogenerador que tenga un comportamiento poco común tendrá mas probabilidades de tener alguna avería, actualmente, o a corto-medio plazo.

Abstract

Internet of Things is a technological concept that is revolutionizing the way we conceive our day to day in recent years. The integration of computer systems in our homes, in our jobs, in the industry is very remarkable, and in some cases, vital. There are many technologies that participate in this concept, and one of them is the "Processing of Complex Events" or "Procesamiento de Eventos Complejos" (CEP).

CEP, is a new way to analyze data. An application of Fast Data, which endows the systems that support it with syntax capability, and overwhelming data analysis. As each technology has its advantages and disadvantages, but without any doubt, we must emphasize that complex event processing provides one of the fastest analysis capabilities of the market, allowing real-time decision making.

The scenario in which we find ourselves in this work is renewable energy, specifically, wind energy. In general terms, the operation of a windmill is very simple, through the force applied by the wind in its blades, mechanical energy is generated in the rotor, generating electricity through a generator. What we will try in this end-of-degree project is to apply CEP, for the optimization of obtaining electricity through wind turbines.

In this end-of-course project, a wind weathering prediction system is developed based on the data obtained by its monitoring system. The objective of the system is to develop, using the processing of complex events and open source technologies, implement a system that allows us to predict when the behavior of wind turbines within a wind farm is anomalous. Therefore, the wind turbine that has an unusual behavior is more likely to be damaged, currently, or a short-medium term.

Agradecimientos

Me gustaría dar las gracias en primer lugar a mis padres, por brindarme la oportunidad de poder estudiar la carrera que quería desde un primer momento. Por apoyarme día sí, día también, desde que empecé mi camino como ingeniero informático, y hacerme ver que no todo en la vida es ser bueno en tu profesión, sino también hay que ser buena persona, buen hijo y buen amigo.

A mi director de trabajo de fin de grado, Gregorio Diáz Descalzo y a Juan Boubeta Puig, por ayudarme en el desarrollo de este proyecto e introducirme en el mundo del Procesamiento de Eventos Complejos.

A Ingeteam S.A por su colaboración tanto ocupando el rol de clientes, como de facilitarme formación en energías renovables.

A mis hermanos y a mi novia, que han sabido reconducirme y tranquilizarme en todo momento, aguantando mis nervios y mi mal humor por el estrés.

Y por último a mis amigos por su incondicional y gran apoyo.

Índice de Contenido

Índice de Ilustraciones

Índice de Tablas

CAPÍTULO 1. INTRODUCCIÓN

"Internet of Things" (IoT), o "Internet de las cosas" en español, es el tema que ha inspirado este proyecto. Cómo poder comunicar todos los elementos que influyen en nuestro día a día puede parecer poco viable, debido a la gran inversión y trabajo que se necesita. No obstante, cada día que pasa dicho esfuerzo es menor [1]. Así, la evolución del IoT cambiará nuestro futuro. Nuestra forma de percibir la realidad cambiará totalmente y no sólo en nuestra vida cotidiana. si no que irá mucho más allá afectando a ámbitos como la educación, la comunicación, las empresas, la ciencia, y el gobierno. En este trabajo de fin de grado nos centraremos en el ámbito científico e industrial, en concreto el que afecta a las energías renovables.

1.1. MOTIVACIÓN

Este proyecto nace de la necesidad de optimizar la obtención de energía a un bajo coste y teniendo en cuenta la preservación del medio ambiente. Las fuentes de energía no renovables son aquellas que se encuentran de forma limitada en nuestro planeta dado que su consumo es mayor que su "regeneración" [2].

La integración de los sistemas de información en el sector industrial está al orden del día, es decir, la industria está vitalmente unida a los sistemas software en su enorme mayoría. Si la tecnología decae, cualquier actividad profesional relacionada está destinada al fracaso como es el caso de la generación de energía [3]. Existen multitud de aplicaciones software para controlar las herramientas de extracción de energía por medios renovables. Por ejemplo, en placas solares móviles podemos observar un sistema para ajustar la orientación con respecto al sol y optimizar su producción y, por otro lado, en el sistema de control de capacidad en un embalse podemos decidir cuándo nos conviene aprovechar la energía producida por su caudal.

A lo que energía renovable se refiere, hemos de destacar la energía eólica y la implicación que tienen los aerogeneradores en la misma. La integración del software y

sistemas electrónicos en todos sus componentes tanto a nivel individual, así como, a nivel de parque eólico es abrumador. Un ejemplo destacable sería el flujo de información que se produce entre los aerogeneradores y las estaciones centrales. Sin embargo, todo ese volumen de información no se aprovecha nada más que para labores muy sencillas de mantenimiento y monitorización.

El objetivo de este proyecto es ir un paso más allá y en base a este gran volumen de información predecir el desgaste de un aerogenerador. Como dijo Bill Gates: "La información es poder". Llevando esta frase a la práctica. Si tenemos toda la infraestructura de sensorización, ¿por qué no aprovechar esos grandes volúmenes de datos no solamente para labores de control y monitorización sino también para realizar predicciones?

1.2. OBJETIVOS

Fruto de la motivación para la elaboración de este proyecto surgen los siguientes objetivos, aunque hemos de destacar en concreto el principal: "Implementar un sistema experto que nos permita predecir los eventos relacionados con el desgaste de un aerogenerador en base a los datos obtenidos por sus sistemas de monitorización".

Para considerar como satisfecho este objetivo hemos de alcanzar las siguientes metas a, que serían:

a. Estudio de energía eólica y familiarizarnos con el entorno de los aerogeneradores, tanto desde un punto de vista técnico, como desde uno más general.
b. Controlar el "Enterprise Service Bus" de Mule, para poder analizar el flujo de datos.
c. Familiarizarnos con el lenguaje para implementar los patrones en Esper EPL.
d. Detectar qué eventos son de interés en las medidas dadas en un aerogenerador para implementar los patrones adecuados.
e. Desarrollo de una aplicación gráfica que nos permita visualizar todos los eventos simples, y complejos fruto de los patrones implementados.
f. Generar alertas en base a los patrones implementados.

1.3. ESTRUCTURA DE LA MEMORIA

El esqueleto interno de esta memoria por capítulos es el siguiente:

CAPÍTULO 1. Introducción

Se trata del capítulo donde nos encontramos, y en el que se han desarrollado temas que sirven como introducción para el correcto entendimiento de los conceptos de este proyecto.

CAPÍTULO 2. Antecedentes y estado de la cuestión

En esta sección veremos reflejados todos y cada uno de los fundamentos teóricos en los que se basa “Eolic Event Consumer”. Se definirá el funcionamiento del Procesamiento de Eventos Complejos y la energía eólica. Terminando con el estado del arte, donde se analizan las tecnologías existentes, revelando sus ventajas y desventajas, en busca de la mejor o la más viable para la solución del problema.

CAPÍTULO 3. ENTORNO DE TRABAJO

En este capítulo veremos qué elementos hardware y software hemos empleado, para el desarrollo de la aplicación, “Eolic Event Consumer”.

CAPÍTULO 4. Metodología.

Una vez presentado el CAPÍTULO 3, donde se ponía de manifiesto una discusión de las metodologías existentes. En este punto describiremos con detalle la elegida, esquematizando los artefactos a entregar, quién ocupa qué roles, así como un Diagrama de Gantt del primer planteamiento del proyecto.

CAPÍTULO 5. Desarrollo

En este capítulo veremos reflejado el trabajo realizado sprint a sprint, con toda la información detallada, desde varios diagramas UML hasta lo recogido en las reuniones con el cliente y el equipo.

CAPÍTULO 6. Experimentos y resultados

En este capítulo quedan reflejadas las pruebas llevadas a cabo en el sistema, siendo contrastadas con el modelo de calidad ISO-IEC 9126. Describiendo algunos de los aspectos visuales y técnicos de la aplicación gráfica, y de la aplicación CEP que trabaja por debajo.

CAPÍTULO 7. Conclusiones y propuestas

Por último, se expondrán una serie de conclusiones, culminando con el planteamiento de futuros trabajos que pueden realizarse utilizando como punto de partida “Eolic Event Consumer”.

CAPÍTULO 2. ANTECEDENTES Y ESTADO DE LA CUESTIÓN

Hoy en día la incorporación del análisis de datos en los sistemas informáticos es un elemento indispensable para su éxito. Con el transcurso del tiempo, el anhelo de conocimiento de ingenieros y tecnólogos produce que la evolución en la industria crezca cada día mas. Ultimamente, estos avances están relacionados con el desarrollo de sistemas que pretenden optimizar la productividad y eficiencia de cualquier actividad, ya bien a sea en nuestros hogares, a través de domótica, o a nivel comercial, a través del análisis de la tendencia en el mercado utilizando técnicas de Machine Learning.

Ese incremento en la productividad y eficiencia, con respecto la competencia, puede significar que nos posicionemos como pioneros en nuestra actividad profesional. Y esto también tiene su lugar en el contexto que nos encontramos, la energía eólica. Optimizar la producción energía frente al resto de multinacionales puede significar que nos coloquemos como cabeza en la lucha por ser líderes de producción energética renovable.

La incorporación de Internet of Things en los parques eólicos puede significar un cambio en el modo de producir energía eléctrica. Los sistemas de monitorización de los aerogeneradores producen una ingesta cantidad de información, de ahí que actualmente se intente utilizar toda esa cantidad de datos, para hacer análisis predictivo y optimizar la producción de energía.

Cada vez son más las productoras de energía renovables como Endesa y Acciona, que integran en sus aerogeneradores estos sistemas predictivos, para poder optimizar su actividad. Cabe destacar en este campo a uno de los fabricantes más famosos de aerogeneradores, Gamesa, que el 29 de octubre de 2008[4], que consolidó una tecnología propia para el mantenimiento predictivo mediante el análisis de vibraciones a través del uso de acelerómetros para detectar el desgaste en los elementos mecánicos del aerogenerador. De ahí que más adelante estudiemos la implantación del procesamiento de los eventos complejos, en una arquitectura orientada a servicios, pero dirigida por eventos.

Volviendo al hilo principal, dividimos este capítulo en dos. Por un lado, tenemos la definición y desarrollo de los fundamentos teóricos de los que nace "Eolic Event Consumer", como herramienta para el mantenimiento predictivo, y, por otro lado, un análisis comparativo de las posibles soluciones al problema planteado.

2.1. FUNDAMENTOS TEÓRICOS.

En este punto veremos todos y cada de los puntos teóricos que rodean este trabajo de fin de grado, tanto a nivel de arquitectura, como de software, como a nivel de contexto, puesto que nos encontramos dentro del escenario de la energía eólica.

2.1.1. ARQUITECTURA ORIENTADA A SERVICIOS (SOA)

La arquitectura orientada a servicios es un estándar basado en componentes que relacionan entre si diferentes funcionalidades de un nodo, estas funcionalidades se denominan "servicios"[5], por medio de tecnologías e interfaces que facilitan la interacción con el usuario.

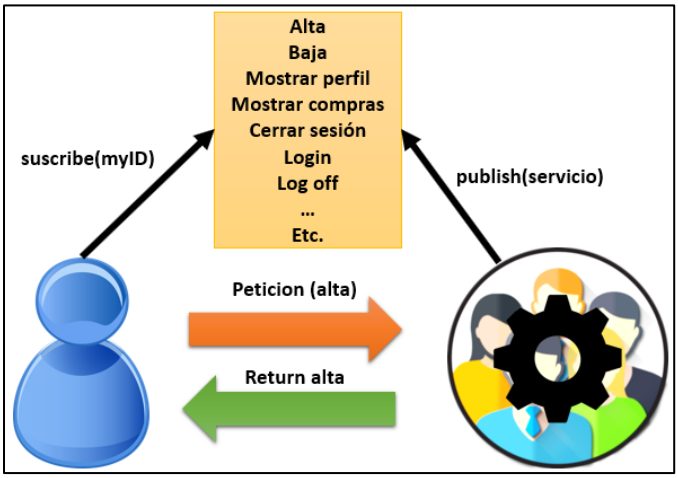

Ilustración 1. Funcionamiento de SOA a través de publish/suscribe.

Un servicio es un elemento reutilizable que puede ser parte o todo de un proceso de negocio, a mayor o menor escala. Un servicio puede ser una simple petición GET a un servidor, para obtener el saldo que nos queda en nuestro banco. Además, puede servir como parte en un sistema completo de logeo en una página web. Hemos de añadir que a lo mejor ese componente del sistema de logeo, por sí solo no aporta ninguna funcionalidad, pero en conjunto con más servicios sí que lo hace. Tal y como vemos en la Ilustración 1, normalmente las arquitecturas orientadas a servicios se implantan a través de protocolos publish/suscribe, donde el usuario se suscribe al proveedor de servicios para solicitar esas funcionalidades.

Una parte imprescindible para el buen funcionamiento de una arquitectura orientada a

servicios es la interfaz, ha de definirse de forma neutral, independiente a lo que trabaja por debajo y proporciona el servicio. Esto desencadena que estas funcionalidades puedan interactuar correctamente con el usuario de forma fácil, sencilla, uniforme y universal sea cual sea su perfil.

La arquitectura orientada a servicios [6] permite que los componentes estén débilmente acoplados, reutilizando los servicios entre sí, optimizando su funcionamiento y compartiendo los componentes internos de la arquitectura con todos los usuarios. Las características que facilitan este débil acoplamiento son:

- Sigue estándares abiertos que permiten la fácil compatibilidad.
- Aseguran la compatibilidad con multitud de tecnologías.
- Separa el código de la interacción del usuario a través de interfaces.
- Modular. Puede ejecutarse como parte o todo de un proceso de negocio, y pueden ser remplazables.
- Sin estado. El servicio no mantiene el estado entre invocaciones, bajo la petición de un servicio se toma el parámetro introducido y se invoca aisladamente la función correspondiente.
- Comunicación. Permite la comunicación entre nodos distintos que proveen de servicios externos.
- Ubicación. El usuario descnoce la ubicación real de la máquina que ofrece el servicio.

2.1.2. ARQUITECTURA DIRIGIDA POR EVENTOS (EDA).

Según dos de los pioneros en lo que al procesamiento de eventos complejos se refiere, David Luckham y W. Roy Schulte, un evento es un suceso que está ha llamado a suceder, [7] y que tiene una relevancia en un contexto concreto. Por ejemplo, la notificación cada diez minutos de un sensor de temperatura, o la llegada de un correo electrónico a nuestra bandeja de entrada. Por tanto, si aplicamos su definición de evento para describir EDA, podemos decir se trata de un estilo de arquitectura cuyos componentes son los eventos, dirigidos y comunicados por ellos mismos.

Este patrón arquitectural permite la producción de eventos, monitorización, así como

su visualización y análisis [8]. Tenemos cuatro elementos que son claves para el funcionamiento normal de este tipo de arquitectura:

- **Productor de eventos.** Es aquel componente que tiene como misión generar los eventos a raíz de lo que ocurre en su entorno. Si lo aplicamos a "Eolic Event Consumer", diremos que los productores de eventos son los aerogeneradores, mandando cada diez minutos un registro con todo lo que ha captado su sistema de monitorización.
- **Consumidores de eventos.** Un componente que consume la información generada por el productor para efectuar la toma de decisiones. En el caso de este trabajo de fin de grado, podríamos decir que es el motor inteligente que procesará toda esa información utilizando los patrones que nosotros como desarrolladores elaboraremos.
- **Canal.** Es el medio por el que se transmiten los eventos.
- **Acciones.** Se trata de las acciones reales que se llevan a cabo, fruto del análisis ofrecido por el consumidor de eventos.

Cabe destacar que EDA siempre tendrá estos actores, en caso de no tener estos actores diremos por tanto que no sigue este tipo de arquitectura y se trata de una arquitectura derivada

2.1.3. ARQUITECTURA ORIENTADA A SERVICIOS DIRIGIDO POR EVENTOS (SOA 2.0)

El procesamiento de eventos complejos tiene lugar en una arquitectura orientada a servicios, pero dirigido por eventos. Esta surge como la evolución de SOA utilizando el mecanismo de EDA [9], y contiene tanto las características de uno como las del otro. En el punto 2.2, explicaremos con detenimiento el concepto del Procesamiento de Eventos Complejos.

El aspecto más relavante de este tipo de arquitectura, es que los servicios son distribuidos en base al flujo de eventos que circulan, y no se queda como en SOA, en una sencilla arquitectura cliente-servidor. A través de este tipo de arquitecturas, se permite la construcción de apliaciones con una gran capacidad de respuesta, puesto que son más

adaptables a entornos impredecibles [10]. Para el correcto funcionamiento de la integración de SOA y EDA, hace falta una capa homogénea que permita la comunicación entre los productores y los consumidores de eventos. Se denomina, “Enterprise Service Bus” (ESB), una capa de integración para poder encaminar el flujo de eventos y mensajes entre los diferentes actores de la arquitectura. En otras palabras, podríamos decir que se trata de una capa de middleware, que permite la comunicación y la administración de datos en todas las aplicaciones distribuidas del sistema.

En la Ilustración 2, podemos ver la arquitectura básica de una arquitectura SOA 2.0 donde, la vía común para todos los productores para comunicarse con los consumidores es a través del ESB. Los productores de eventos pueden ser desde el sistema de monitorización de un molino de viento o de una casa, hasta mediciones muy simples de velocidad del viento y temperatura, siendo muy sencillo el evento producido. Un ejemplo de aplicación de CEP en SOA 2.0, podría ser controlar la temperatura en todas las habitaciones de nuestra casa. ¿Cómo controlar esa temperatura en las diferentes habitaciones? A través de la elaboración de patrones que detecten esos eventos simples, produciendo una alerta en forma de evento complejo. El enunciado de este patrón podría ser, suponiendo que hubiera un termómetro en cada habitación, que cuando todos hayan mandado un evento simple con temperatura mayor a 20 grados centígrados, se produzca ese evento complejo, cuya acción asociada haga que la calefacción se pare.

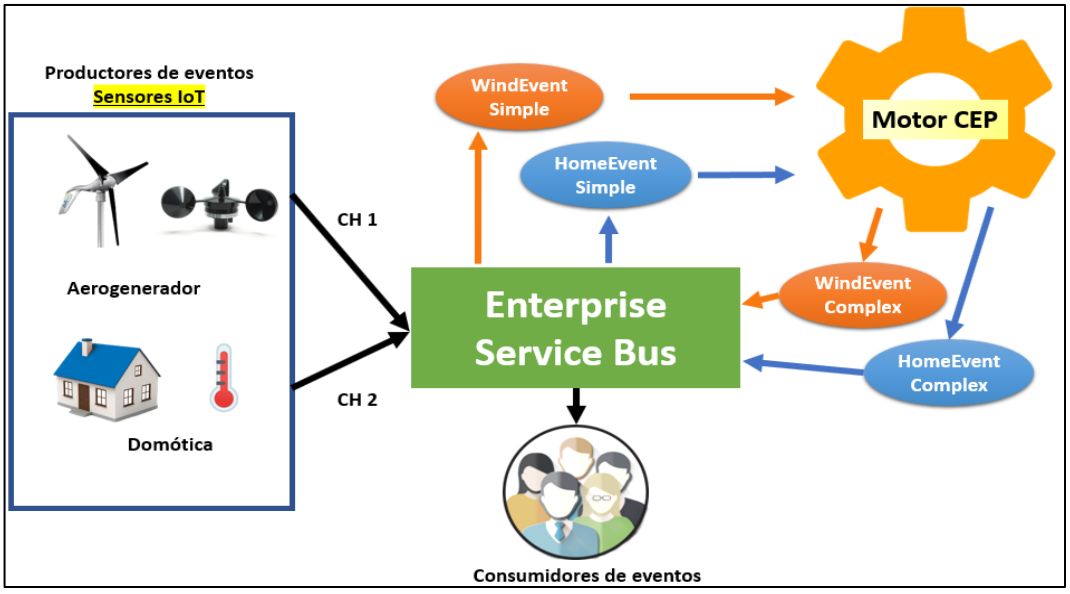

Ilustración 2. Funcionamiento ideal SOA 2.0.

Volviendo al hilo principal, aunque hayamos adelantado la noción de patrón, hemos de tener muy claro, que la consecución de eventos de un tipo no tiene como consecuencia sólo la llamada a una función en un servidor para llevar a cabo una acción, si no que va más allá, y puede generarse a su vez, otro evento que, junto a otros, sí que puede ser de interés. El objetivo de este tipo de arquitecturas es optimizar y agilizar la toma de decisiones, en base a la información obtenida a tiempo real y en tiempo de ejecución.

2.2. PROCESAMIENTO DE EVENTOS COMPLEJOS (CEP)

Dando respuesta a la pregunta lanzada en la motivación: "¿por qué no aprovechar esos grandes volúmenes de datos no solamente para labores de control y monitorización sino también para realizar predicciones?

Surge como posible solución el Procesamiento de Eventos Complejos (tecnología CEP) como aplicación de "Fast Data". La mejor forma de tomar decisiones en estos ámbitos es hacerlo cuanto antes, de ahí que se examine toda la información a tiempo real para de esta forma dirigir satisfactoriamente la toma de decisiones o acciones correctivas en el ámbito industrial. Hay que saber distinguir que se habla de "Fast Data" y no de "Big Data" [11]. ¿Por qué? Fast Data tiene una ventaja, y es que no almacena ningún tipo de información. Conforme los datos son generados, una vez procesados por primera vez, se desechan, es decir, se analizan conforme van siendo generados para después desecharlos, lo que lo hace muy rentable. Si se almacenara hablaríamos de "Big Data". Por lo tanto, como se genera, se procesa, y se desecha sin almacenarla hablamos de "Fast Data", evitándonos el gran coste que llevaría salvaguardar tanta información.

Partiendo de que un evento complejo es un elemento que resume, representa o denota un conjunto de otros eventos diremos que CEP, tal y como Fast Data determina, es que puede dominar cualquier volumen de datos, a una velocidad en tiempo real y de orígenes muy diversos. CEP, es una tecnología emergente capaz de monitorizar múltiples flujos de eventos, procesarlos mediante la determinación de patrones implementados en Esper Event Processing Language (EPL) [12], y así tomar decisiones tanto predictivas como correctivas en base a la información que el contexto nos aporta.

Como conclusión a la problemática generada por la necesidad de optimizar la

producción en un parque eólico, este trabajo de fin de grado tiene como meta la predicción del desgaste de aerogeneradores a través de un sistema SOA 2.0. SOA 2.0, es una arquitectura orientada a servicios, aunque dirigida por eventos [13], en la sección 2.1.3 se ha explicado su funcionamiento.

Cabe destacar que CEP [14] tiene varios aspectos comúnes con una arquitectura basada en servicios, donde el usuario sabe cual es el servicio para solicitar, sin embargo, en el contexto de CEP, son los productores de eventos los que manda las peticiones a los consumidores (usuarios) para que procesen toda esa información. Podemos decir que la arquitectura donde se implantan se implanta la tecnología basada en el procesamiento de eventos complejos es reactiva (bajo la ocurrencia de situaciones de interés) y desacoplada (al poder haber más de un único productor de eventos).

Debido a las pequeñas diferencias y similitudes con SOA, podemos concluir que SOA 2.0 nace como fruto de la combinación con una arquitectura dirigida por eventos, EDA, ajustándonos a ese funcionamiento reactivo y desacoplado.

Si analizamos el funcionamiento de CEP [15], podemos enumerar diferentes características como son:

- **Comunicación múltiple**. Por lo general los productores de eventos envían la información que es de interés a cada uno de los consumidores, que se han suscrito a ese productor.
- **Inmediatez.** La comunicación de los eventos es a tiempo real en el momento en el que se detecta alguna situación de interés.
- **Asincronismo.** El sistema que publica un evento no espera a que los sistemas que lo reciben lo procesen. Publica y continúa.
- **Eventos de grano fino.** Todo el análisis surge a raíz de eventos simples con datos de grano fino, pudiendo así, simplificar el flujo de información y facilitar en análisis y la toma de decisiones en base a la información obtenida. Cuanto más se simplifique el flujo de eventos, mejor
- **Nomenclatura y categorización.** Se puede presentar la jerarquización de los eventos permitiendo la discretización de la información, o la clasificación de eventos en base

a su relevancia o lecturas.

- **Procesamiento de Eventos Complejos.** El sistema interpretará y monitorizará las relaciones entre los eventos. Por ejemplo, la agregación de eventos (un patrón de eventos implica un evento de mayor nivel) o relación de causalidad (un evento es originado por un evento previo).

2.2.1. TIPOS DE PROCESAMIENTO

Por lo general hay tres tipos de procesamiento de eventos[16], todos y cada uno de ellos son complementarios entre sí, son los siguientes:

a. **Procesamiento de eventos simple.** El sistema se nutre de los eventos de más bajo nivel, iniciando acciones de muy baja transcendencia. Normalmente se utiliza para impulsar en tiempo real la información, disminuyendo el tiempo de demora a la hora de iniciar procesos paralelos.
b. **Procesamiento de eventos en streaming**. Suceden eventos comunes y notables. Los eventos ordinarios (pedidos, transmisiones de RFID) se analizan en busca de notoriedad y se transmiten a los consumidores de eventos. Se usa comúnmente para impulsar el flujo de información en tiempo real dentro y alrededor de la empresa, permitiendo la toma de decisiones a tiempo.
c. **Procesamiento de eventos complejos.** Este es el tipo de procesamiento que trataremos en este trabajo de fin de grado. Se procesan esos eventos durante un tiempo determinando, pudiéndose almacenar temporalmente según su orden de temporal normalmente, aunque la correlación de eventos puede ser casual, o espacial. CEP se usa comúnmente para detectar y responder a anomalías, amenazas y oportunidades comerciales.

2.2.2. DIAGRAMA DE FUNCIONAMIENTO

El funcionamiento del motor CEP, es sencillo. Una vez que se hayan implementado los patrones. Los añadiremos a la cola de patrones desplegados dentro del motor CEP, tal y como vemos en la Ilustración 3. Una vez arrancado nuestra aplicación, los productores de eventos irán mandando lo captado a través del ESB, para que el motor CEP analice los datos. Podemos decir que, hay tres etapas principales dentro de la situación normal de una

aplicación de CEP en SOA 2.0, son las siguientes:

- **Producción.** Que es toda esa trama de medición del entorno real, así como la normalización de los datos para que sea material legible para el motor CEP y su envío al ESB.
- **Detección y análisis.** La información transformada (si fuera necesario) llega al entramado inteligente de la aplicación de tal forma que se coteja toda la recibida con los patrones de eventos desplegados. Es importante destacar, que normalmente en tiempo de ejecución, el motor CEP es capaz de desplegar nuevos patrones para la detección de eventos complejos.
- **Respuesta y despliegue.** Una vez detectados los eventos complejos, se comienza la toma de decisiones, y se emprenden las acciones estipuladas.

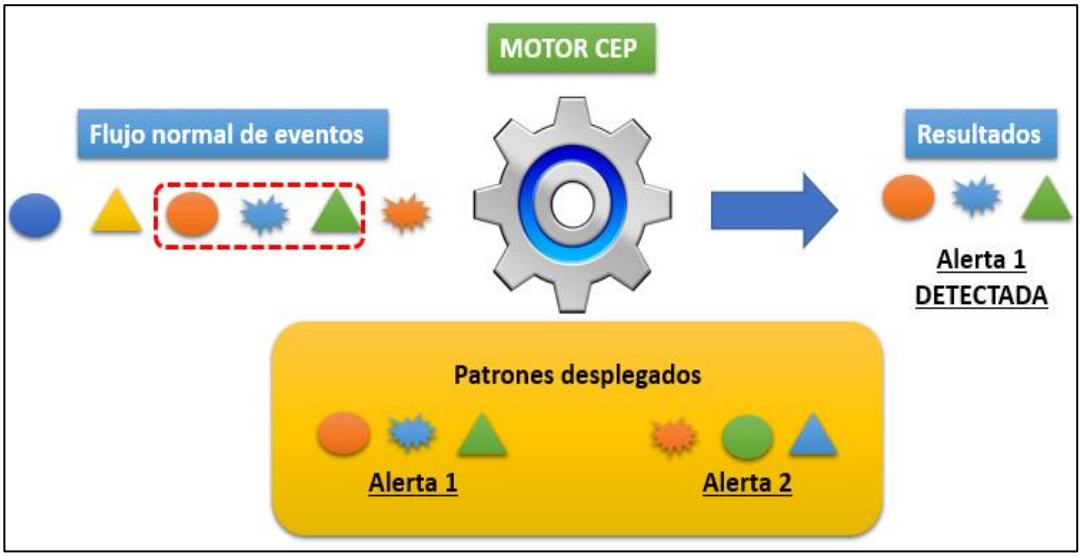

Ilustración 3. Funcionamiento básico del motor CEP.

Habiendo visto, cómo funcionan los motores CEP, y cómo se integra SOA 2.0 con el procesamiento de eventos complejos, podemos adelantar un primer acercamiento en la Ilustración 4, a la solución que "Eolic Event Consumer" proporciona al problema.

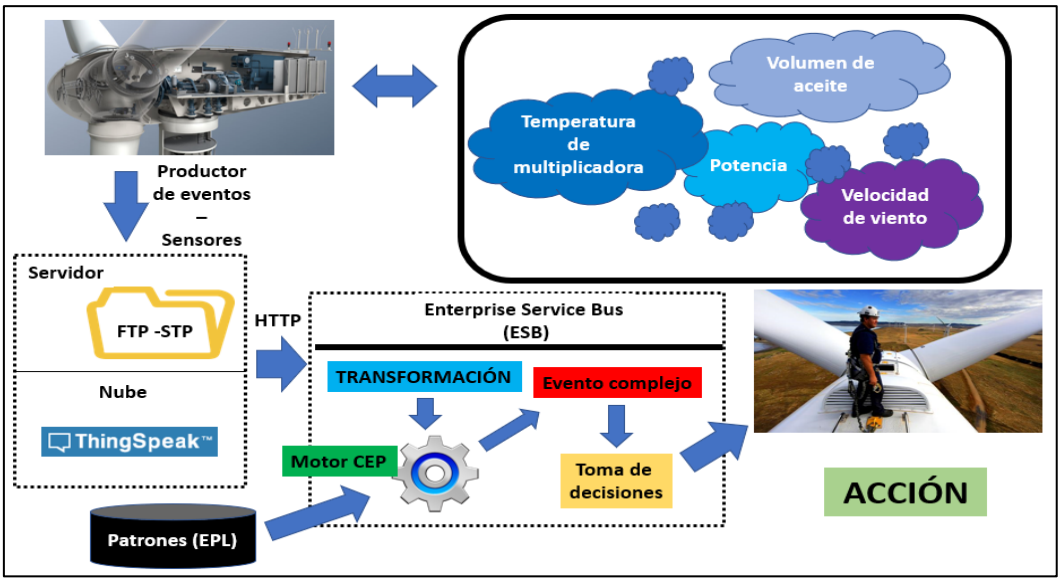

Ilustración 4. Esquema de "Eolic Event Consumer".

Como vemos, el origen de los datos proviene de los múltiples sensores fluyendo vía HTTP para transmitirlos al motor CEP. El motor CEP, procesa la información utilizando los patrones implementados. De esta forma, desplegaremos los eventos complejos en una aplicación gráfica aplicando medidas correctivas y preventivas dependiendo del evento complejo que se detecte en el flujo de datos.

2.3. ESPER EPL. LENGUAJE DE PROGRAMACIÓN PARA CEP.

Para poder implementar esos patrones para la detección de eventos complejos usaremos Esper EPL. EPL, es el acrónimo de “Event Processing Language”, es un leguaje de programación declarativo con gran escalabilidad, con memoria eficiente, computación en tiempo de ejecución, SQL estándar, mínima latencia, transmisión en tiempo real con capacidad de motor de grandes volúmenes de datos y de gran variedad, permitiendo además el análisis de históricos. Un patrón de eventos, ciñéndonos a lo publicado por David Luckham y W. Roy Schulte [7], es un esquema de operadores relacionales y variables, intentando reflejar una situación real mediante su morfologia. Un patrón de evento puede hacer coincidir conjuntos de eventos relacionados al reemplazar variables con valores.

Esper ofrece un lenguaje específico de dominio (DSL) para el procesamiento de eventos. El lenguaje de procesamiento de eventos (EPL) es un lenguaje declarativo para tratar datos de eventos basados en el tiempo de alta frecuencia. EPL cumple con el estándar

SQL-92 y se amplía para analizar series de eventos y con respecto al tiempo [17].

Además, Esper proporciona en su página web, una aplicación en línea para probar EPL, la URL es esta: http://esper-epl-tryout.appspot.com/epltryout/index.html

Esper ofrece un lenguaje de patrones de eventos para especificar coincidencias de patrones de eventos basados en expresiones. Detrás del motor de coincidencia de patrones está la implementación de una máquina de estado. Este método de procesamiento de eventos coincide con las secuencias esperadas de presencia, ausencia, o combinaciones de eventos.

Esper también ofrece consultas de series de eventos que abordan los requisitos de análisis de series de eventos de las aplicaciones de CEP. Las consultas de series de eventos proporcionan las funciones de ventanas, agregación, unión y análisis para su uso con flujos de eventos. Estas consultas siguen la sintaxis de EPL. EPL ha sido diseñado para similitud con el lenguaje de consulta SQL, pero difiere de SQL en el uso de vistas en lugar de tablas. Las vistas representan las diferentes operaciones necesarias para estructurar datos en una serie de eventos y derivar datos de una serie de eventos.

2.3.1. SINTAXIS

Además, hemos de destacar que Esper EPL[18], sigue el esquema sintáctico de Ilustración 5, sin embargo, tiene incluido las librerías de java.lang, java.math, java.text, y java. util, que hacen a este lenguaje SQL aún mucho más interesante para el análisis de datos.

```
[@Name (“Pattern’s Name”)]
[expression_declarations]
[context contextname]
[into table table_name]
[insert into insert_into_def]
select select_list
from stream_def [as name] [, stream_def [as name]] [,...]
[where search_conditions]
[group by grouping_expression_list]
[having grouping_search_conditions]
[output output_specification]
[order by order_by_expression_list]
[limit num_rows]
```

Ilustración 5. Estructura de patrón en Esper EPL.

Cada una de las sentencias de la Ilustración 5, tiene una función, y son las siguientes:

- **`@Name ("XXX").`** Se añade el nombre del patrón a implementar, es opcional, sin embargo, es esencial para legibilidad del código.
- **`INSERT INTO.`** La creación de un flujo de eventos complejos, identificados por el nombre introducido en esta parte de la sentencia SQL.
- **`SELECT`**. Recoge las propiedades de los eventos simples/complejos del flujo capturado.
- **`FROM`**. Flujo de evento seleccionado.
- **`FROM PATTERN.`** Lugar donde se añaden las condiciones del patrón de los eventos del flujo seleccionado.
- **`WHERE.`** Clausula que se encarga de hacer un filtrado al flujo de eventos seleccionado en la clausula FROM.
- **`GROUP BY`**. Agrupaciones.
- **`HAVING.`** Clausula para el filtrado de los elementos agrupados.
- **`ORDER BY.`** Ordenación de los eventos por el criterio que se seleccione.

Los elementos por los que Esper EPL se distingue de un lenguaje SQL estándar, son los operadores de patrón, y el uso de ventanas temporales, que permiten capturar el tráfico de eventos en un flujo durante un tiempo determinado para su análisis. Partiendo de aquí, definiremos cuales son los elementos más útiles, y de los que nacen la mayoría de los códigos en Esper EPL. Por un lado, tenemos las ventanas de datos, que retienen los eventos entrantes hasta que una política de caducidad indique que se liberarán los eventos. Por lo tanto, las ventanas de datos son un medio de indicar qué subconjunto de eventos a analizar. Un ejemplo de las más representativas son las siguientes:

a. **`WIN:TIME_BATCH(N SECONDS)`**. Ventana por lotes que obtiene todos los elementos retenidos el tiempo especificado.
b. **`WIN:LENGTH_BATCH(N)`**. Ventana que extrae el número de eventos especificado como parámetro.
c. **`WIN:TIME(N SECONDS)`**. Se declara una ventana deslizante de un período de tiempo especificado.
d. **`WIN:LENGTH(N)`**. Ventana deslizante vinculada al número de eventos.

Como podemos ver, hay dos tipos de ventanas temporales. Las ventanas que van por lotes, y las deslizantes. Las ventanas deslizantes capturan tantos eventos, como haya en X período de tiempo desde que se encuentra el primero, y por otro lado, tenemos las ventanas que van por lotes, que separan el tiempo en intervalos de X período de tiempo capturando todos los que nazcan en ese tiempo.

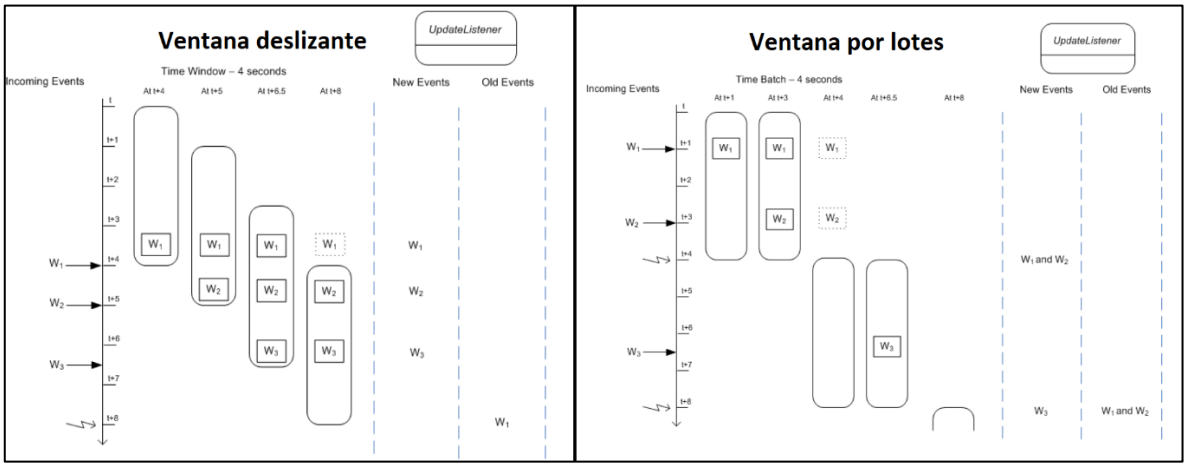

Ilustración 6. Ventanas deslizantes y por lotes.

Para verlo de una forma más gráfica, podemos ver en la Ilustración 6, como la ventana deslizante, captura desde que encuentra el primero todos los que hay en un período de tiempo X, y la que va por lotes, captura todos los eventos que surgen en intervalos de 4 segundos. Lo "negativo" de las ventanas por lotes, es que podemos hacer una de 4 segundos, y puede que del segundo 8 al 12 no surja ningún evento, de manera que se devolvería una ventana vacía.

Finalmente, tenemos los operadores de patrón, elementos que estudian a través de operadores lógicos y aritméticos el flujo de eventos de eventos capturados, y los temporizadores, para relacionarlos con el momento con el que fueron desplegados. Los elementos más destacables de los operadores de patrón son:

a. **`EVERY`**. Selecciona todos los eventos del tipo que se especifique.
b. **`EVERY-DISTINCT`**. Es exactamente igual, sin embargo, elimina las instancias repetidas.
c. **`FOLLOW BY ("->")`**. Para especificar, el patrón que esté a la izquierda debe ser seguido por un segundo.

d. **`UNTIL`**. Se comprueba un patrón hasta que se cumpla.
e. **`WHILE`**. Se comprueba un patrón mientras se cumpla.

Para entender los operadores de patrón correctamente, Esper, sugiere varios ejemplos que pueden ser de interés, por ejemplo, suponiendo el siguiente flujo de eventos:

```
A1, B1, C1, B2, A2, D1, A3, B3, E1, A4, F1, B4
```

Si hacemos un filtrado utilizando los operadores de patrón, “every” y “follow by” podemos obtener los siguientes resulados según el orden en el que los escribamos:

I. `Every (A -> B)`

Resultado I. {A1, B1}, {A2, B3}, {A4, B4}

II. `Every A -> B`

Resultado II. {A1, B1}, {A2, B3}, {A3, B3}, {A4, B4}

III. `A -> every B`

Reusltado III. {A1, B1}, {A1, B2} {A1, B3} {A1, B4}

IV. `Every A -> every B`

Resultado IV. {A1, B1}, {A1, B2}, {A1, B3}, {A2, B3}, {A3, B3}, {A1, B4}, {A2, B4}, {A3, B4}, {A4, B4}

Como hemos comentado anteriormente, el último elemento que hemos de destacar es el temporizador, para vincular los operadores de patrón con el tiempo en el que nacieron.

- **`TIMER:INTERVAL(N SECONDS)`**. Espera el tiempo especificado antes de que el valor se evalue a verdadero.
- **`TIMER:AT(*,*,*,*,*,*)`**. Retornará verdadero cuando llegue el momento indicado.
- **`TIMER:WITHIN (N SECONDS)`**. Se da falso cuando la condición del patron no

se ha cumplido en un tiempo estipulado.

2.4. ESTADO DEL ARTE

En este punto veremos unas discusiones de lo que actualmente hay en el mercado, y donde justificaremos porqué hemos escogido CEP, en base a qué criterios hemos elegido el ESB de MuleSoft, así como una pequeña comparación de las diferentes metodologías para el desarrollo software, y el porqué de la elección de utilizar una metodología ágil.

2.5. DISCUSIÓN TECNOLÓGICA

En este punto veremos dos comparativas tecnológicas. Por un lado, la discusión de los lenguajes para el procesamiento de eventos complejos existentes y el porqué elegimos Esper EPL, y, por otro lado, tenemos la comparativa de los diferentes Enterprise Service Bus del mercado.

2.5.1. DISCUSIÓN DE LENGUAJES CEP.

Algunos lenguajes que manejan el procesamiento de eventos complejos son Esper EPL, Continuos Query Language (CQL) de Oracle, StreamSQL y Continous Computation Languaje (CLL).

En el caso de CQL de Oracle[19], podemos decir que se tata de un lenguaje de consulta basado en SQL para el procesamiento de eventos complejos sólo compatible con el software Oracle CEP. Oracle CEP es un servidor de Java para desarrollar aplicaciones dirigidas por eventos. Proporciona un entorno abundante y declarativo basado en Oracle SQL, con todas y cada una de las librerías puestas a disposición en su SQL tradicional, además de algunas otras de java, al igual que Esper EPL. Oracle CEP admite rendimiento ultra alto y latencia de microsegundos utilizando JRockit Real Time y proporciona Oracle CEP Visualizer y Oracle CEP IDE para las herramientas de desarrollador de Eclipse para una plataforma completa de desarrollo de arquitectura basada en eventos (EDA) Java en tiempo real. Sin embargo, no nos hemos decantado por esta opción debido a dos factores:

- Nuestra experiencia con los últimos IDEs de Oracle como SQLDeveloper, podemos asegurar que los entornos se quedan congelados por fallos en su

implementación, no pudiendo así explotar al 100% su EPL.

- Coste. Al contrario que Esper EPL que es gratuito, para poder utilizarlo hemos de pagar una licencia que cuesta XXXX.

Stream SQL es el EPL que es acoplado en SQLstream s-Server desarrollado por SQLStream[20]. Está basado en SQL y su servidor está diseñado para satisfacer las necesidades de baja latencia, alto volumen y rápida integración de las empresas actuales en tiempo real. SQLstream s-Server procesa transacciones de diversas fuentes de forma continua, proporcionando análisis de transmisión y salida de datos a múltiples destinos. Las transformaciones y análisis complejos y sensibles al tiempo son simples de configurar, y se ejecutan de forma continua en múltiples orígenes de datos de entrada y escriben en múltiples destinos. Este software es gratuito, pero a partir del día 60, a partir de allí solo podrá procesar un 1 GB/ día. El coste [21] de esta herramienta en máquinas de Azure, Amazon, Docker, y Linux oscila desde unos 80 € como mínimo, a como máximo unos 1465 € mensuales. También existe la posibilidad de hacerlo por consumo, es decir pagando 6.60 $ la hora. Esta opción queda descartada por los costes y por lo costoso que es utilizar su documentación.

Al igual que Stream SQL, Esper EPL y CQL, el lenguaje también se basa en SQL. Continuous Computation Language (CCL), es un lenguaje que sólo integrable en Sybase CEP, está desarrollado por SAP. Proporciona las mismas utilidades que el resto. Cabe destacar su precio, [22] que es de $ 775.48, y que no hay versión gratuita ni ninguna trial.

Elegimos Esper EPL por las siguientes razones:

- Totalmente gratuito, una solución de Open Source.
- Esper EPL proporciona un servicio web para probar de forma fácil y sencilla los patrones implementados, simulando el funcionamiento normal del motor CEP.
- Todo el lenguaje queda recogido como un simple “jar”, lo que lo hace muy manejable y se puede amoldar a cualquier ESB.
- El soporte de Esper es muy bueno, la documentación técnica de la sintáxis es muy didáctica y fácil de aprender, todo con muchos ejemplos permitiendo la fácil comprensión del funcionamiento tanto del lenguaje como de la tecnología basada en el procesamiento de eventos complejos.
- He recibido un curso por Juan Boubeta Puig, profesor titular en la Universidad de

Cádiz, donde tuvimos un primer acercamiento a CEP. Al utilizar Esper EPL en el curso, y aprender de una forma tutorizada me siento más seguro utilizando esta opción.

2.5.2. DISCUSIÓN DE ENTERPRISE SERVICE BUS

En este punto veremos veremos los ESB más representativos, y finalmente haremos un análisis comparativo de las soluciones de Open Source y Closed Source.

https://www.chakray.com/comparativa-esb-open-source/

Comparo los diferentes ESB que hay en el mercado y decido cual es el mejor para este problema.

AQUÍ ATENCION UTILIZA LA FUENTE [23]

Cccc

https://dzone.com/articles/esb-comparison

Discusión de porque utilizamos CEP contra otro tipo de tecnologías.

https://bbvaopen4u.com/es/actualidad/tres-alternativas-solidas-de-big-data-en-tiempo-real-spark-storm-y-datatorrent-rts

	WSO2 ESB and SOA Platform	Mule ESB	FuseSource ESB	Adroit Logic UltraESB	JBoss ESB and SOA Platform	Tibco ActiveMatrix
Supports Enterprise Integration Patterns	Y	Y	Y	Y	Y	Y
Delivers all required ESB features (i.e. web services, message transformation, protocol mediation, content routing)	Y	Y	Y	Y	Y	Y
Offers a complete and cohesive SOA Platform (i.e. ESB, Message Broker, Governance Registry, Business Process Server, Data Services Server, Application Server)	Y	N	N	N	Y	Y
SOA Governance	Y	N	N	N	N	Y
Graphical ESB Development Workbench	Y	Y	Y	N	Y	Y
Based on a composable architecture	Y	N	N	N	N	N
Cloud integration platform offering (iPaaS)	Y	Y	N	N	N	Y
Cloud Connectors and Legacy Adapters	Y	Y	N	N	Y	Y
Performance	High	Moderate	Moderate	High	Moderate	High

Security and Identity Management	Y	Limited	Limited	Limited	Limited	Limited
Open Business Model	Y	Y	Y	Y	N	N

2.6. METODOLOGÍAS EXISTENTES

La eterna batalla entre las metodologías ágiles y las tradicionales tiene cabida en este trabajo de fin de grado también. La elección de la metodología para desarrollar un sistema software no es una decisión sencilla, debemos de barajar todos y cada uno de los factores que influyen directa e indirectamente en el proyecto. Antes de elegir una, debemos de definir cuales son las bases tanto de las metodologías tradicionales y las metodologías ágiles.

Piattini en 1996 [24], definía el concepto de metodología, como el "conjunto de pasos y procedimientos que deben seguirse para el desarrollo de software". Aunque el concepto no está claro del todo, es evidente que una metodología de desarrollo garantiza la sistematización del desarrollo. Otro de los pioneros en ingeniería del software, Maddison, en 1983, definía metodología, como el conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladres. De manera que una metodología debe dictaminar:

- La estructura en etapas del proyecto.
- Tareas de cada etapa.
- Artefactos que tienen lugar tras la realización de las tareas.
- Roles y deberes.
- Qué restricciones existen.
- Cuáles son las mejores herramientas que se pueden aplicar, y cuales son las que nosotros podemos aplicar.
- Cómo gestionar y controlar un proyecto.

Dejado esto claro, enfrentamos dos tipos de metodologías:

a. Metodología tradicional.

Las metodologías tradicionales se centran en el buen trabajo del proceso de desarrollo del softare, con el objetivo de obtener un software eficiente [25]. Para esto se hace hincapié en la planificación completa del trabajo que se ha llevar a cabo, comenzando así el ciclo de

desarrollo del software. Primero planificamos todas las tareas, y a continuación comenzamos a desarrollar. Se centra como he dicho al comienzo de este punto en el control del proceso, fijando desde un primer momento, los roles, tareas, artefactos, etapas y modelado. Algunos ejemplos de metodología tradicional pueden ser RUP, MSF, Win-Win Spiral Model e Iconix.

b. Metodología ágil.

Los procesos ágiles son una buena elección cuando se trabaja con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero. En estas situaciones, un proceso adaptativo será mucho más efectivo que un proceso predictivo. Por otra parte, los procesos de desarrollo adaptativos también facilitan la generación rápida de prototipos y de versiones previos a la entrega final, lo cual agradará al cliente. Algunos ejemplos de metodología son XP, Scrum, Crystal Clear, DSDM, FDD, XBreed, y Extreme Modeling.

Si comparamos estas dos metodologías podemos sacar las conclusiones extraídas en la Tabla 1 [26].

Tabla 1. Comparativa entre metodología ágil y metodología tradicional.

Metodologías ágiles	Metodologías tradicionales
Desarrollos cortos (menos tiempo)	Desarrollos medio-largos (más tiempo)
Impuestas internamente, las decisiones parten del equipo.	Impuestas externamente, dependen más del cliente.
Proceso menos controlado debido a que se rige por menos principios.	El proceso es controlado por muchos sectores, con multitud de normas y directrices.
No existe un contrato tradicional, en principio.	Existe un contrato prefijado.
Muy flexible a la hora de incluir cambios.	No tan flexible, ya que incluir un cambio es más lento. Resistencia al cambio.
El cliente es parte del equipo de desarrollo, continuamente en contacto con él.	El cliente interactúa con el equipo de desarrollo únicamente a través de

	reuniones.
Grupos pequeños, no más de una decena de personas, trabajando cerca geográficamente.	Grupos grandes con posibilidades de que se encuentren lejos entre sí.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos hincapié en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.
Desarrollo incremental e integración continua.	Modelo de desarrollo en cascada.
Menos caro.	Más caro.

Hecha esta comparación entre las dos modalidades de metodologías existentes, hemos elegido una metodología ágil, debido a:

- Poco tiempo para desarollar el sistema.
- El cliente debe involucrarse activamente con el desarrollo del proyecto debido al desconocimiento del Team en materia de energías renovables.
- El criterior de aceptación debe ser flexible, ya que además de las preferencias del cliente también tiene que ser regido y evaluado por un tribunal académico.
- Desde un primer momento se sabe que se quiere hacer una aplicación CEP, sin embargo, en ningún momento se sabe qué patrones se van a implementar ni qué va a encuadrar el sistema, por tanto, el resultado final no puede regirse por ningún contrato.
- Debe ser flexible al cambio.
- No hay tiempo para hacer toda la documentación que una metodología tradicional requeriría.
- No hay ningún tipo de fondo para desarrollar el sistema.

Una vez hecha la elección del tipo de metodología a utilizar, pasamos a ver cuáles son las metodologías ágiles que existen para poder seleccionar la que mejor se adapta a nuestras necesidades. Las metodologías ágiles que estamos planteándonos utilizar, en base a las que más se usan comúnmente y más implantadas están son las siguientes:

a. SCRUM.

SCRUM [27] fue propuesto a principios de los 90 por Jeff Sutherland y Ken Schwaber. La base principal del éxito de SCRUM es la descentralización de los equipos, puesto que son autoorganizados. Los requisitos funcionales que se describían en las metodologías tradicionales son en las ágiles, descompuestas en historias de usuario y estas en tareas para llevarlas a cabo.

Dividen las funcionalidades en tareas, y estas tareas deberán de estar resueltas al terminar un período de tiempo determinado, denominado "sprint". El ciclo de vida del proyecto será dividido en sprints, y esos sprints constarán de diferentes historias de usuario. Los roles dentro de una metodología SCRUM son:

- Scrum Master. Jefe de proyecto.
- Product Owner. Cliente.
- Team. Equipo de desarrollo.

b. Kanban.

Kanban [28] surgió como uno de los componentes claves en el Sistema de Producción Toyota, origen a su vez de Lean and Just In Time.

En Kanban todos los requisitos funcionales estarían planteados como tareas en un flujo de trabajo: el trabajo se divide en artículos (tareas) que jugarían en un tablero, el tabler tiene varias columnas que representan las etapas de este flujo de trabajo. Un ejemplo del tablero la podemos ver en Ilustración 7.

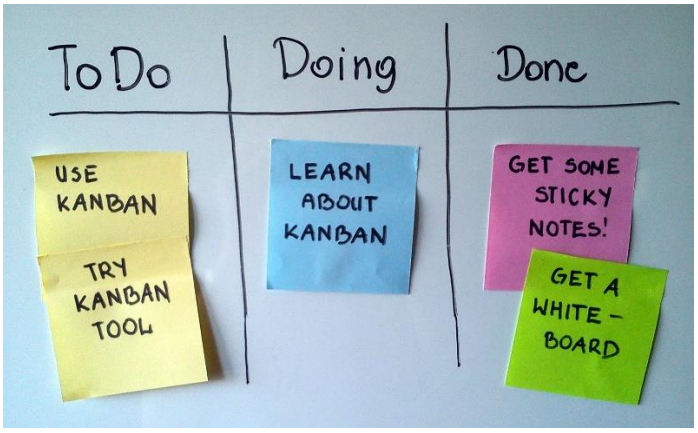

Ilustración 7. Tablero Kanban.

Es importante destacar que el número de artículos están en progreso son finitos, depende del número de personas en el equipo. El tiempo promedio que se tarda en completar un artículo, también conocido como tiempo de ciclo, es analizado y optimizado para que el proceso sea eficiente y eficaz. La eliminación de artículos considerados como basura, o desechos es primordial (elementos que no aportan ni valor, ni funcionalidad).

c. Extreme Programming.

eXtreme Programming [29] (XP), ante todo busca ofrecer la máxima calidad posible en el software desarrollado a través de técnicas de análisis, desarrollo y testeo adecuadas. Cada miembro del equipo de XP participa activamente para la planificación de la iteración, al contrario que el resto de las metodologías, donde sí que hay una persona encargada de coordinar al equipo, aquí no. Las iteraciones son muy cortas bajo demanda del cliente, a partir de varias prácticas que son:

- Test Driven Development. Desarrollo dirigido a los tests.
- Customer Testing. Las pruebas funcionales con el cliente son básicas para el éxito del desarrollo del sistema.
- Integración continua.
- Pequeños lanzamientos.
- Programación de pares. Se programa en grupos de dos personas, ambas personas se nutren como programadores, aportan, se ayudan y optimizan la solución.

2.7. ELECCIÓN DE METODOLOGÍA ÁGIL.

Hecha una pequeña definición de tres de las más importantes metodologías ágiles hemos de decantarnos por la ideal que se ajusta a nuestras necesidades. Para ello [30], existen Frameworks que en base a nuestras necesidades nos permiten elegir la metodología que más se ajusta a nuestro caso, un ejemplo de ello es el creado por Adrian Iacovelli, autor de “Framework for Agile Methods Classification”. Se establece una clasificación basada en cuatro principios, definidos en la Ilustración 8.

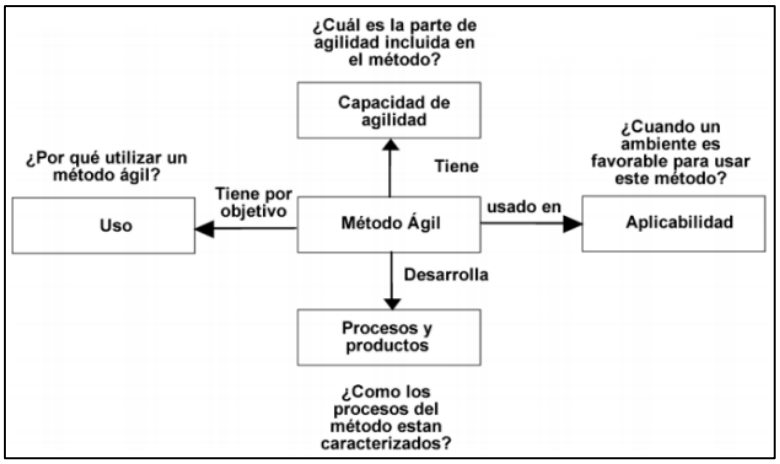

Ilustración 8. Principios de Iacovelli.

En la Tabla 2, quedan reflejadas las diferencias y similitudes entre las metodologías ágiles definidas anteriormente. También hay que destacar, que quedan señaladas cuáles son los aspectos que tienen cabida en nuestro proyecto en color verde, y en base a su resultado elegiremos una metodología u otra.

Tabla 2. Criterios de elección de metodologías ágiles de Iacovelli.

		Orientada al desarrollo	**Orientada a la gestión de proyectos**	
		XP	***SCRUM***	***KANBAN***
USO	Respeto de las fechas de entrega.	F	V	F
	Cumplimiento de requisitos.	V	V	V
	Respeto al nivel de calidad.	F	F	F
	Satisfacción del usuario final.	F	V	F
	Satisfacción del usuario final.	V	V	V
	Entornos turbulentos.	F	V	F
	Aumento de la productividad.	V	V	V
CAPACIDAD DE AGILIDAD	Iteraciones cortas.	V	V	V
	Colaboración.	V	V	V
	Centrado en las personas.	V	V	V
	Refactoring político.	V	F	F
	Prueba política.	V	V	F
	Integración de los cambios.	V	V	V
	De peso ligero.	V	V	V

	Lo RF pueden cambiar.	V	V	V
	Los NO RF pueden cambiar.	F	F	V
	El plan de trabajo puede cambiar	V	F	V
	Los recursos humanos pueden cambiar.	V	F	V
	Cambiar los indicadores	V	F	F
	Intercambio de conocimientos.	ALTO	BAJO	BAJO
APLICABILIDAD	Tamaño del proyecto.	PEQUEÑO	PEQUEÑO GRANDE	PEQUEÑO
	La complejidad del proyecto.	BAJA	ALTA	BAJA
	Los riesgos del proyecto.	BAJA	ALTO	BAJO
	El tamaño del equipo.	PEQUEÑO	PEQUEÑO	PEQUEÑO
	Grado de interacción con el cliente.	ALTA	ALTA	BAJA
	Grado de interacción con los usuarios.	BAJO	ALTA	BAJO
	Grado de interacción entre el equipo.	ALTA	ALTA	BAJA
	Grado de integración de la novedad.	ALTA	ALTA	BAJA
	La organización del equipo.	AUTO	AUTO	AUTO
PROCESOS Y PRODUCTOS	**Nivel de abstracción de las normas y directrices**			
	Gestión de proyectos.	F	V	F
	Normas y orientaciones concretas sobre las actividades y productos.	V	F	F
	Descripción de procesos.	V	F	F
	Las actividades cubiertas por el método ágil			
	Puesta en marcha del proyecto.	F	F	F
	Definición de requisitos.	V	V	F
	Modelado.	V	V	V
	Código.	V	V	V
	Pruebas unitarias.	V	V	V
	Pruebas de integración.	V	V	V
	Prueba del sistema	V	V	V
	Prueba de aceptación	F	F	F
	Control de calidad	F	F	F
	Sistema de uso	F	F	F
	Productos de las actividades del método ágil.			
	Modelos de diseño.	F	V	F
	Comentario del código fuente.	V	V	V
	Ejecutable.	V	V	V
	Pruebas unitarias.	V	V	V

	Pruebas de integración.	V	V	V
	Pruebas de sistema.	V	F	V
	Pruebas de aceptación.	F	F	F
	Informes de calidad.	F	F	F
	Documentación de usuario.	F	F	F

El resumen de Tabla 2, queda reflejado en Tabla 3. Viendo los resultados podemos decir que la mejor metodología a utilizar para este proyecto según Iacovelli, es SCRUM. Hecho este estudio acerca de las metodologías existentes, coincidimos con Iacovelli y emprenderemos SCRUM con unas pequeñas modificaciones que las explicaremos más adelante en el CAPÍTULO 4.

Tabla 3. Resumen de la elección de metodologías ágiles por Iacovelli.

Metodologías	Positivos	Negativos
XP	34	17
SCRUM	49	2
Kanban	36	15

Con esto, concluimos el capítulo relacionado con el estado del arte, y a continuación pasamos a concretar aspectos más relacionados con el caso práctico en el que nos encontramos.

CAPÍTULO 3. ENTORNO DE TRABAJO

En este apartado hablaremos de todos los medios tanto software como hardware tenemos pensado utilizar para llevar a cabo el proyecto en primera instancia.

3.1. MEDIOS HARDWARE.

Los medios hardware que utilizaremos para llevar a cabo el proyecto serán los siguientes:

- **Ordenador personal.** Se trata de un ordenador ASUS bastante común, utilizado durante el transcurso de la carrera. Procesador i7-5610QM CPU de 2,30 GHz y memoria RAM de 6 GB, con un tipo de sistema de 64 bits y con un sistema operativo Windows 10.
- **PLC de aerogenerador.** Se trata del PLC que controla todas las mediciones de las centralitas PT100 que nutren el sistema de monitorización del aerogenerador. Esta será la fuente de datos del sistema, aunque en primer lugar trataremos un conjunto de datos sintético en formato Excell o .db con el que simularemos el flujo de información, debido a que no tenemos acceso directo a ese PLC.

3.2. MEDIOS SOFTWARE.

En esta sección veremos cuáles son los elementos software que tomarán participación en nuestro entorno de trabajo. Desde los programas, IDEs y demás, hasta las librerías particulares que hacen singular este proyecto.

- **Eclipse.** Eclipse es un entorno de desarrollo integrado de Java (IDE), con gran cantidad de plugins de Open Source con lenguajes integrados como C / C++ y PHP. Puede combinar fácilmente el soporte de idiomas y otras funciones en cualquiera de los paquetes que la nube de Eclipse facilita en los paquetes predeterminados, y Eclipse Marketplace permite una personalización y extensión prácticamente ilimitadas. Hemos elegido Eclipse y no otro entorno porque el ESB está

implementado sobre Eclipse, de esta forma podemos unificar paralelamente todo el código, tanto el del simulador como el del Enterprise Service Bus [31].

- **EGit**. EGit es un proveedor de Eclipse Team para el sistema de control de versiones de Git. Git es un SCM distribuido, lo que significa que cada desarrollador tiene una copia completa de todo el historial de cada revisión del código, lo que hace que las consultas contra el historial sean muy rápidas y versátiles. El proyecto EGit está implementando herramientas Eclipse además de la implementación JGit Java de Git [32].
- **Team Foundation Service.** Se trata de un producto Software que nos permitirá llevar a cabo la gestión del trabajo. En concreto la enumeración de historias de usuario, de tareas, y generación de artefactos como sería el Sprint Burndown.
- **Anypoint Studio, ESB de Mulesoft**. Es el entorno de desarrollo de integración basado en Eclipse de MuleSoft para diseñar y probar aplicaciones Mule. Puede implementar la aplicación y ejecutarla en su servidor Mule. El mismo editor también le permite editar archivos de definición de API (en RAML y WSDL) y crear dominios que definan recursos compartidos. Anypoint Studio ofrece dos pestañas paralelas que puede utilizar para diseñar y crear sus aplicaciones: Editor visual; y un editor XML [33]**.**
- **Librería poi-3.11 para el tratamiento de documentos Excel.** Se trata de una librería de Apache para la manipulación de documentos Excel que en un principio se generarían por el PLC del aerogenerador y nutrirían al motor CEP.
- **Librería de Esper EPL.** Esper ofrece un lenguaje específico de dominio (DSL) para el procesamiento de eventos. El lenguaje de procesamiento de eventos (EPL) es un lenguaje declarativo para tratar datos de eventos basados en el tiempo de alta frecuencia. EPL cumple con el estándar SQL-92 y se amplía para analizar series de eventos y con respecto al tiempo [17].
- **Medit4CEP.** Es una herramienta para la generación de código Esper EPL, a través de un editor gráfico. Esta aplicación sirve es una solución para acercar la tecnología de procesamiento de eventos complejos a cualquier usuario independientemente del nivel de programación que posea, para que, de esta forma, se facilite la toma de decisiones [34].

CAPÍTULO 4. METODOLOGÍA.

En el comienzo de este capítulo veremos la organización del proyecto, cuáles son los roles de los participantes en el proyecto, el diagrama de Gantt orientativo del comienzo del proyecto, así como una breve descripción de este, y a continuación el documento de gestión de riesgos que se ha generado al inicio del proyecto. Por último, en este capítulo explicaremos que peculiaridades tiene la elección de la metodología elegida, así como la descripción de los hitos y artefactos que se han generado.

4.1. INTRODUCCIÓN

En un primer lugar el desarrollo del proyecto desde el mes de enero cuenta con 23 semanas de desarrollo. En cuanto a la metodología de trabajo hemos de destacar que se usará una metodología ágil, en concreto será SCRUM, combinándolo con prototipado.

Se basará en una metodología SCRUM donde los roles están bastante bien identificados, sin embargo, el Team, solo tiene un integrante. Por lo tanto, no podremos decir que es un SCRUM "puro" como tal. Los integrantes del grupo ajustando los roles dentro del equipo son:

- **Jefe de proyecto.** Gregorio Diaz Descalzo. Profesor dentro de la Politécnica de Albacete, en la Escuela de Ingenieros Informáticos. Coordinador entre Team y Product Owner para que haya buena comunicación y no haya ningún problema a la hora de establecer los requisitos y compromisos para y con el proyecto tanto a nivel de organización como a nivel técnico.
- **Product Owner.** Francisco José Polo Sánchez. Integrante de la empresa Ingeteam. Cliente en cuestión dentro del desarrollo del proyecto, solicita el sistema y proveerá de infraestructura de trabajo al Team. Una vez hecha la recogida y elicitación de requisitos, será el encargado en aprobarlos. En caso de haber algún error, se hallará un acuerdo y se tomarán las medidas pertinentes para su aprobación.
- **Team.** Enrique Brazález Segovia. Estudiante de 4º de Ingeniería Informática,

responsable de defender el trabajo de fin de grado.

Tal y como hemos comentado anteriormente, la metodología va a estar entrelazada con el prototipado, de forma que tras acabar cada iteración en el proyecto tendremos un producto operativo y funcionando. La organización del trabajo tal y como queda esquematizado en el diagrama de Gantt de la Ilustración 9 es la siguiente:

- **Inicio del proyecto**. En esta iteración se planteará el proyecto y se interiorizarán todos los aspectos técnicos acerca del ESB, la energía eólica, etc. Además, plantearemos la planificación del proyecto, realizaremos una primera versión de los requisitos a alcanzar, y se elaborará un primer prototipo del flujo de datos usando un intermediario vía Web como es "Thingspeak". Esta aplicación externa servirá de intermediario entre el generador de eventos y el motor CEP.
- **Adaptación al caso real.** A través del ESB generaremos otro prototipo que no dependa de" Thingspeak" (a priori) y empezaremos a implementar de forma real todos y cada uno de los patrones ajustándonos al modelo de aerogenerador concreto. Insertaremos las alertas obtenidas de los eventos complejos en una base de datos MySQL.
- **Desarrollo de la plataforma.** Pasaremos a la elaboración de la aplicación gráfica que servirá como pantalla de visualización de los eventos detectados a través del framework CakePHP.
- **Tiempo de margen para la memoria y flecos sueltos.** Es una pequeña iteración que servirá como tiempo de reserva por si surge algún imprevisto que retrase el proyecto.
- **Presentación y defensa del trabajo de fin de grado.**

Ilustración 9. Diagrama de Gantt de planificación de tiempo.

4.2. ANÁLISIS DE RIESGOS

Partiendo de las nociones en la quinta edición del PMBOK [35],un riesgo es un problema potencial que puede o no ocurrir, y suele afectar a futuros acontecimientos. Los riesgos implican cambios, y debemos estudiar pese a que mi rol no sea el de "Scrum Manager", qué implicaciones tiene sobre nuestro trabajo la aplicación de dichos cambios. Además, los riesgos nos hacen enfrentarnos a diversas elecciones, como escoger las herramientas a emplear, la estructuración de nuestro código, modificaciones en el tipo de licencias que manejamos o a importancia que debemos de darle a la calidad de sistemas software en dicho instante.

Por todo lo anterior aparece este Plan de Gestión de Riesgos, para actuar en caso de que estos aparezcan en nuestro proyecto. Este plan consiste en una serie de pasos que ayudan al equpo de software a comprender y gestionar la incertidumbre en la toma de decisiones tanto a nivel organizativo com de desarrollo.

4.3. POSIBLES RIESGOS

Hay multitud de riesgos que pueden darse lugar, sin embargo, he querido destacar solo los que he visto más relevantes:

a. **Problemas con la comprensión de EPL.** Podría ocurrir, que, una vez hecho el estudio del contexto, y analizados cuáles son los fenómenos eólicos que se pueden medir a través de patrones, no sepamo implementarnos. Que la sintáxis nos de muchos problemas y que no entendamos de primeras bien el paradigma de la programación orientada por eventos.
b. **Problemas con la programación con CakePHP.** Podría ocurrir, que, debido a que no tenemos ninguna noción de PHP, y ni mucho menos del framework de CakePHP, la implementación de la aplicación gráfica se retrase o que ni si quiera se lance. Es un riesgo que se ha de afrontar debido a que el sistema interno del cliente está implementado de esta forma.
c. **El cliente no tenga detectados, aunque sea en lenguaje formal los patrones.** Debido a la complejidad de hallar patrones, puede caber la posibilidad de que el cliente no tenga asociados las mediciones de los sensores anómalas, a un tipo de avería concreto. En esta situación seremos nosotros los que nos encarguemos de

enunciar los patrones, y que estos sean validados por el cliente.

d. **El cliente no se interese por el proyecto.** Puede que la empresa no tenga conocimiento del funcionamiento de las metodologías ágiles, y no se implique en el proyecto tanto como debiera. Puede caber la posibilidad de que nuestro coordinador con la empresa cliente, se desvincule y perdamos ese primer contacto con Ingeteam.
e. **Enfermedad del Team.** Podría ocurrir, que, el Team enferme, por lo que irremediablemente se retrase la entrega del proyecto a Julio, o incluso a septiembre.
f. **Cambiar de una base de datos relacional a otra que no lo es.** Podría ocurrir, que, debido a la cantidad ingente de datos con las que trabajaremos, el modelo de dato cambie. Por tanto, no podremos cuadrar las alarmas en una base de datos tipo MySQL, y nos tendremos que pasar a MongoDB.

4.4. PONDERACIÓN

En la Tabla 4, podemos observar en las filas, la Probabilidad de que ocurra ese riesgo, y en las columnas las consecuencias que suponen para el proyecto:

Tabla 4. Métrica del PMBok de la Gestión de Riesgos.

Gestión de riesgos en "EolicEventConsumer"	INSIGNIFICANTE (1)	MENOR (2)	MODERADA (3)	MAYOR (4)	CATASTRÓFICA (5)
RARO (1)	BAJO	BAJO	MODERADO	ALTO	ALTO
IMPROBABLE (2)	BAJO	BAJO	MODERADO	ALTO	EXTREMO
POSIBLE (3)	BAJO	MODERADO	ALTO	EXTREMO	EXTREMO
PROBABLE (4)	MODERADO	ALTO	ALTO	EXTREMO	EXTREMO
CASI SEGURO (5)	ALTO	ALTO	EXTREMO	EXTREMO	EXTREMO

- **Bajo.** Los riesgos bajos deben ser objeto de seguimiento por parte únicamente de un miembro del Team.
- **Moderado.** Los riesgos moderados deben ser objeto de seguimiento adecuado por parte tanto del Enrique, como de Fernando.
- **Alto.** Los riesgos altos requieren la atención del Scrum Manager y del Team.
- **Extremo.** Los riesgos extremos han de comunicarse al cliente directamente después de a ver sido planteados por el Scrum Manager y el Team.

En el caso de “Eolic Event Consumer”, podemos clasificar estos riesgos, según la siguiente categoría:

g. **Problemas con la comprensión de EPL.** Tiene una probabilidad probable (4) de que ocurra, ya que enfrentarnos a un lenguaje que trabaja con una lógica distinta a lo que tenemos estructurado tradicionalmente puede suponer problemas. Las consecuencias de dicho riesgo podemos clasificarlas como menores (3), puesto que pese a que desconocemos el lenguaje se nos impartirá un curso de iniciación, y tendremos contacto con profesores doctorados que conocen el lenguaje, dispuestos a ayudarnos.

h. **Problemas con la programación con CakePHP.** Tiene una probabilidad casi segura (5), puesto que el Team no ha tocado nunca PHP, ni mucho menos un framework tan avanzado como es CakePHP. Tampoco está familiarizado con el servidor Apache, y MySQL, tecnologías imprescindibles para instaurar un servicio web que se amolde al de nuestro cliente. Las consecuencias de este riesgo podemos verlas como catastróficas (5) puesto que, si este riesgo se convierte en realidad, retrasaremos el despliegue mucho tiempo e incluso se tome la decisión de instaurar otra tecnología directamente.

i. **El cliente no tenga detectados, aunque sea en lenguaje formal los patrones.** Este riesgo podemos categorizarlo con probabilidad posible (3), ya que por ahora de lo hablado con el cliente tienen muy claras cuales pueden ser las señales que despiertan un comportamiento anómalo del aerogenerador, sin embargo, pensamos todos los miembros del proyecto involucrados en el desarrollo y en la dirección que no los tienen redactados como tal. Las consecuencias de este riesgo son moderadas (3), ya que, pese a que no tienen su redacción, podemos ayudarles a definir los patrones con lenguaje formal ya que tienen muy claro cuáles son las deficiencias que pueden hacernos pensar que ese comportamiento anómalo es debido al desgaste de alguna de sus partes.

j. **El cliente no se interese por el proyecto.** Este riesgo tiene una probabilidad posible (3), debido a que no hay ninguna inversión económica. Este factor puede hacer que el cliente no le de ningún tipo de prioridad a este proyecto. Sin embargo, las consecuencias que conllevaría la desvinculación con el cliente serían mucho mayores (4), debido a que, sin el enunciado de los patrones, ni con expertos de energía eólica

no podremos implantar un sistema que detecte esas máquinas con comportamiento irregular.

k. **Enfermedad del Team.** Se trata de un riesgo que puede ser clasificado con una probabilidad posible (3), debido a que cualquier persona del equipo puede padecer una enfermedad. Debido a que es un trabajo de fin de grado en el que estaremos a jornada completa trabajando en él, en un período de tiempo muy corto las consecuencias de este riesgo son moderadas (3), pudiendo llegar a ser castróficas.

l. **Cambiar de una base de datos relacional a otra que no lo es.** Si clasificamos este riesgo diremos que tiene una probabilidad posible (3) de que suceda, ya que la escalabilidad en un sistema de Big Data es primordial, y puede que al trabajar con distintos modelos de datos se tenga que utilizar una base de datos no relacional tipo MongoDB. Las consecuencias de este riesgo son menores (2), significaría una modificación a nivel de estructura del código, por tanto, un incremento del número de tareas, y a su vez del tiempo invertido.

Si resumimos este análisis de riesgos obtendríamos la Tabla 5, donde se refleja de forma gráfica que nos encontramos ante riesgos muy peligrosos, más de la mitad de ellos son justificados porque se quiere desarrollar este sistema en menos de 5 meses, de ahí que se declaren la mayor parte de ellos como extremos y altos.

Tabla 5. Ponderación de riesgos.

RIESGO	PONDERACIÓN
Problemas con la comprensión de EPL.	EXTREMO
Problemas con la programación con CakePHP.	EXTREMO
El cliente no tenga detectados, aunque sea en lenguaje formal los patrones.	ALTO
El cliente no se interese por el proyecto.	EXTREMO
Enfermedad del Team.	ALTO
Cambiar de una base de datos relacional a otra que no lo es.	MODERADO

4.5. METODOLOGÍA SCRUM CON PROTOTIPADO

Aquí explico que voy a utilizar SCRUM con prototipado.

CAPÍTULO 5. DESARROLLO

En este capítulo veremos todos los artefactos generados a lo largo del desarrollo del proyecto partiendo de la premisa que estamos vinculados a una metodología ágil, en concreto, SCRUM con prototipado, tal y como hemos comentado en el CAPÍTULO 4.

5.1. SPRINT 1. ACERCAMIENTO A CEP.

En este primero no se dieron muchas cosas, pero sirvió como introducción a todo.

Hay una historia de usuario:

- Introducción a Tecnología CEP. Se trata de aprender los conocimientos de CEP tanto por mi cuenta como lo impartido en las clases del máster. Sirve como una pequeña introducción a lo que se desarrollará en el segundo cuatrimestre para tener en mente una primera infraestructura y poder plantear diferentes soluciones bien sean erróneas o no al supuesto que se nos propone.
    - Acudir al seminario del máster. Acudir al seminario de CEP impartido por Juan Boubeta Puig, profesor de la universidad de Cádiz. 9 horas.
    - Instalación del software para hacerse a la idea del modelo prueba. Se trata de instalar las herramientas propuestas por el seminario que son las siguientes:
        - Medit4CEP. Editor gráfico para facilitar la implementación de los patrones.
        - AnyPointStudio. Herramienta para establecer el ESB donde se configurará el motor CEP.
        - Eclipse Oxygen. Entorno de programación para arrancar el simulador.
    - Realizar ejercicios de CEP.
        - Realizar ciertos supuestos prácticos propuestos en el seminario

como por cuanta propia para ir cogiendo soltura en el lenguaje y las nociones de CEP.

5.2. SPRINT 2. INICIO DEL PROYECTO

En esta iteración se empezó

5.2.1. REUNIÓN 1. “CONOCEMOS AL CLIENTE, PRIMERA TOMA CON EXPERTO”

5.2.1.1 INFORMACIÓN

En esta reunión tomaremos un primer contacto oficial con el cliente, “Ingeteam”.

- Dirección: Parque Científico y Tecnológico, Paseo de la Innovación,3, 02006 Albacete. Oficinas de Ingeteam S.A.
- Hora y fecha de la reunión. A las 8:15 el día 8 de enero de 2018.

Los asistentes de la reunión son:

- Vicente Requena Montejano. Ingeniero Industrial. Especializado en aerogeneradores.
- Antonio Fernández Diez. Ingeniero Industrial. Especializado en placas fotovoltaicas.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.

5.2.1.2 ASUNTOS QUE TRATAR

Debido a que ninguno de los asistentes es Salazar, enlace directo con la empresa, tendremos que explicar cuáles son los objetivos de nuestro proyecto, cuál es la tecnología que utilizaremos y la capacidad de análisis que tiene el procesamiento de eventos complejos. Los temas que se tomaron en la reunión fueron:

- Presentación y recepción de contactos.
- Situación actual del contexto.

- Clase introductoria de energía eólica.
- ¿Qué tipo de aerogenerador vamos a analizar?

5.2.1.3 INFORMACIÓN RECOGIDA

La situación con el cliente en un primer momento es buena. Los asistentes conversan y tratan los problemas sin ningún problema. A priori no parece que vayan a haber conflictos a lo largo del ciclo de vida del proyecto. En primer lugar, el contexto del proyecto se ha concretado en el estudio de un aerogenerador estándar de 1500 KW. El acceso a los datos obtenidos por los PLCs de control del sistema de monitorización es algo complicado debido a que los clientes no quieren en un primer momento proporcionar sus datos. Sin embargo, en Burgos hay un parque cuyo servicio es dado por Ingeteam, y sus datos sí que son accesibles. A continuación, se da una clase teórica para adentrarnos en el mundo de la energía eólica, los datos más representativos son los recogidos en Tabla 6.

Tabla 6.Conclusiones de curso introductorio de energía eólica.

Nombre	***Descripción***	***Trababjamos con***
Potencia nominal	Se trata de la potencia sacada por el aerogenerador en su rendimiento más alto	1500 W
Rango de viento		4 -25 m/s
Tensión	La tensión que sale del generador para la estación distribuidora	690 V – 12000 V
Giro del rotor	El giro del rotor	10 – 30 rpm
Palas	Son 3 palas que van acopladas al eje central, dos posiciones extremas. Atención los grados son respecto el eje central de la junta de las palas "CONO". A más palas más coste, 3 es lo óptimo en cuanto a relación dinero/optimización	2 posiciones 0º Barlovento (máximo rendimiento) 90 º Bandera (mínimo rendimiento)
Nacelle	Góndola	-
Rotor	Cono y palas	-
Pitch	Giro de las aspas sobre el cono para cambiar su grado (de 0 a 90)	Es un pitch eléctrico
Multiplicadora	Al ser un aerogenerador con dipolo la velocidad del rotor es muchísimo menor a la que le hace falta al generador para producir energía eléctrica de forma que necesitamos una multiplicadora para adecuar	Sí, trabajamos con un aerogenerador con multiplicadora eléctrica.

	ambas velocidades a la de sincronismo. 1 etapa planetaria y 2 paralelas	
Etapa planetaria	Comunica directamente la parte acoplada al rotor y transmite la velocidad de este a la fase paralela	ATENCIÓN LA LUBRICACIÓN ES IMPORTANTE, debido a la generación de partículas que producen desgaste.
Etapa paralela	Conjunto de engranajes para derivar una velocidad baja transmitida por la fase planetaria al generador con la velocidad de sincronismo adecuada.	ATENCIÓN LA LUBRICACIÓN ES IMPORTANTE, debido a la generación de partículas que producen desgaste.
Lubricación y refrigeración en multiplicadora	Es importante debido a que SIEMPRE ha de estar a una temperatura y nivel de aceite adecuado. La temperatura es controlable, pero ES INEVITABLE el cambio de aceite debido a su gran uso	NO SABEMOS RANGOS ¿Filtros se desgastan? ¿Resistencia calefactora peta? Prestar atención en: • Rodamientos. • Engranajes. • Resistencia calefactora. • PT100 – Sensor térmico del sistema de lubricación.
Junta rotativa	Se trata de una junta rotativa que comunica la multiplicadora con el generador ya que uno al estar estático y otro que es móvil entonces debemos de incorporar para impedir que se líen cables	Puede haber más de una junta rotativa
Generador doble alimentado	Se trata del que hace la conversión y genera la energía en watios que se mandará a la estación eléctrica. Incorpora otro PT100. Circuito de producción para la red (el normal), otro circuito de alimentación que alimenta al generador para producir energía en caso de haber poco viento de esta forma siempre se podrá generar energía puesto que se ayuda al	Con una tensión que se va a la estación de 690 V de forma que hay que subirla para que no haya pérdidas 1500 rpm 10 % deslizamiento

	rotor a girar.	2 – 4 pares de polos. Tener en cuenta: - PT100. - Cuerpo de anillos. - ¿Refrigeración aire o agua?
Polipasto	Grúa de ayuda para llevar herramienta pesada en pequeñas reparaciones.	
Sistema del YAW	El YAW es el giro de la góndola en 360º con respecto el eje vertical del aerogenerador.	Es un sistema de YAW eléctrico.
Freno de fricción	Frena el movimiento de YAW en algún momento.	¿Calor? ¿Número de veces utilizado?
Dispositivo cuentavueltas	Cuentavueltas de góndola para evitar enrollamiento de los cables de potencia.	Máximo dos vueltas, se tiende a desenrollar tras las dos o para optimizar a desenrollar cuando no tiene un buen rendimiento el aerogenerador.
Grupo hidráulico	Controla toda la hidráulica del rotor y las palas	Control de aceite. Acumulador de nitrógeno. Preostatos. PT100.
Armario TOP	Donde se ubica toda la lógica software del sistema, desde el movimiento del YAW, los controles de las temperaturas y sistemas hidráulicos hasta las rotaciones del PITCH.	
Armario GROUND	Controla toda la potencia y las funcionalidades de mantenimiento y demás de los aerogeneradores.	
Transformador	Elevación de la tensión de 690 V obtenidos por el generador hasta elevarlo al voltaje requerido para que no haya perdida de energía.	
Celda de maniobra	Para actividades de maniobra o corte de tensión del aerogenerador.	

5.2.1.4 PRÓXIMOS PASOS

En esta reunión se han acordado las siguientes tareas:

- Petición de datos sintéticos al parque eólico de Arroyal, Burgos. El encargado de recoger esta información es Vicente, para después proporcionar los datos al desarrollador, Enrique.
- Elaboración de modelo de datos a partir de los datos sintéticos obtenidos por Vicente, para comenzar con el primer prototipo de la iteración primera del proyecto y estudio de los conocimientos aportados por la clase introductoria de energía eólica.
- Antonio deberá de hablar con Pedro para coordinar al desarrollador, y fijar el encargado de controlar el progreso de las reuniones.

5.2.2. REUNIÓN 2. “INICIO DEL PROYECTO DE TRABAJO DE FIN DE GRADO”

5.2.2.1 INFORMACIÓN. “CONOCEMOS AL CLIENTE, PRIMERA TOMA CON EXPERTO”

En esta reunión comenzaremos la iteración 1 “Inicio del proyecto”.

- Dirección. Edificio Infante Don Juan Manuel. Avda. de España s/n. Albacete. Despacho 0. B.8.
- Hora y fecha de la reunión. A las 18:00 el día 11 de enero de 2018.

Los asistentes de la reunión son:

- Gregorio Diaz Descalzo. Jefe de proyecto de este trabajo de fin de grado. Titular de la Universidad de Castilla-La Mancha.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.
- Fernando Luján Martínez. Desarrollador de trabajo de fin de grado paralelo, su tutor también es Gregorio.

5.2.2.2 ASUNTOS QUE TRATAR

En resumen, esta reunión es para atender aspectos relacionados con la elaboración del trabajo de fin de grado de cara a la universidad. Es decir, de cara a la presentación del anteproyecto y los criterios de la convocatoria. Los temas que se trataron fueron:

- Criterios de evaluación de la convocatoria.
- Aclaración de las fechas de las que disponemos.
- Explicación de cómo hacer el anteproyecto.
- Definir planificación del proyecto.
- ¿Qué metodología vamos a utilizar? ¿Qué iteraciones nos vamos a marcar?
- Explicación de sistema de citas.

5.2.2.3 INFORMACIÓN RECOGIDA

Se repasaron con atención todos y cada uno de los criterios para la evaluación del trabajo de fin de grado, desde su evaluación como a los requisitos mínimos que debería de tener. En concreto se repasaron todos y cada uno de los capítulos de la memoria del trabajo de fin de grado para que no haya dudas en su redacción.

Hablando de las fechas nos imponemos como fecha límite para tenerlo todo terminado, el día 17 de junio de 2018, a efectos de la presentación para la defensa que va desde el día 25 al 29 de junio de 2018. Hay que destacar que el día 18 de junio se ha de hacer el depósito, de ahí que tengamos que tener todo preparado para el día 17. Utilizaremos SCRUM combinándolo con prototipado, de forma que tendremos 4 iteraciones y al final de cada iteración tendremos un prototipo listo y funcionando.

En cuanto al anteproyecto, hemos discutido sobre todo los aspectos referentes a la justificación de las competencias. En concreto hemos tomado las siguientes notas, tanto Fernando como yo:

- **Primera competencia.** Aportar que vamos a usar una metodología ágil. El uso de la tecnología CEP de forma eficiente y con pruebas de la evaluación (pruebas de carga y estrés). Aportar que es barato puesto que no necesitamos ningún tipo de base de datos para guardar la información.
- **Segunda competencia.** Se van a recoger todos y cada uno de los requisitos necesarios para conseguir que el proyecto sea aceptable y cumpla con todo lo impuesto por el cliente.
- **Tercera competencia.** Descripción de las metodologías y herramientas que vamos a utilizar.
- **Cuarta competencia.** Discusión y debate acerca qué tecnologías utilizar y porqué CEP es la más viable.
- **Quinta competencia.** Realizar un plan de gestión de riesgos tal y como hemos visto en Gestión de Proyectos Software.
- **Sexta competencia.** Control de recursos humanos, fallos en las herramientas, contacto continuo con el cliente.

Por último, en la última parte de la reunión se hace muchísimo hincapié en la importancia de la inclusión de bibliografía debido a que es una de las partes clave del trabajo de fin de grado. De esta forma, no tendremos problemas en aspectos referentes a la propiedad intelectual.

5.2.2.4 PRÓXIMOS PASOS

En esta reunión se han acordado las siguientes tareas:

- El anteproyecto ha de estar hecho a primeros de febrero.
- Establecer un primer flujo de datos en esta primera iteración que coja los datos a través de "Thingspeak" con un pequeño muestreo de datos. No hace falta con el modelo de datos real, aunque sí se puede mejor.
- Utilizar algún control de versiones como podría ser GitHub. Como utilizaremos Eclipse, se toma la decisión de acoplarlo todo utilizando EGit, que es la distribución de Git para el IDE de Eclipse.
- Utilizar algún programa tipo "Kunagi" para llevar la gestión de las tareas e historias de usuario de los sprints.
- Hacer una planificación inicial con todos los hitos que tendría el proyecto, a nivel de reuniones, fechas de fin de iteración, y demás.

5.2.3. REUNIÓN 3. "DUDAS Y PROBLEMAS"

5.2.3.1 INFORMACIÓN

En esta reunión se resolvieron dudas del anteproyecto y ciertos problemas de colaboración por parte del cliente.

- Dirección. Edificio Infante Don Juan Manuel. Avda. de España s/n. Albacete. Despacho 0. B.8.
- Hora y fecha de la reunión. A las 16:00 el día 23 de enero de 2018.

Los asistentes de la reunión son:

- Gregorio Diaz Descalzo. Jefe de proyecto de este trabajo de fin de grado.

Titular de la Universidad de Castilla-La Mancha.

- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.
- Fernando Luján Martínez. Desarrollador de trabajo de fin de grado paralelo, su tutor también es Gregorio.

5.2.3.2 ASUNTOS QUE TRATAR

En resumen, esta reunión sirve como recursos para atender ciertos problemas a la hora de hacer el anteproyecto, y aspectos referentes a la coordinación Product Owner-Team en el proyecto. Los temas que se trataron fueron:

- Dudas a la hora de realizar el anteproyecto, posibles restricciones, aspectos positivos, aspectos negativos.
- ¿Dónde está Pedro?
- No sabemos de qué información bebe el motor de datos, hay que decidir cómo se nutre al motor que procesa los eventos.
- Recogida de requisitos.

5.2.3.3 INFORMACIÓN RECOGIDA

En esta reunión se vieron las dudas que hubo a la hora de llevar a cabo el anteproyecto, sirvió para que Gregorio diera el visto bueno ya que el enunciado de las competencias es muy confuso. Tanto Fernando como yo, necesitábamos realizar esta prueba de aceptación para cerciorarnos de que íbamos por el buen camino.

Pedro fue nuestro primer enlace con Ingeteam para llevar a cabo este proyecto, sin embargo, por circunstancias que desconocemos se fue de la empresa sin comentarnos la situación. Un error por su parte al irse habiéndose comprometido con el proyecto. Sin embargo, se acordó establecer una reunión con Antonio y Vicente la semana próxima para solucionar todos los problemas de coordinación y enmarcar los límites de nuestro proyecto. De esta forma, fijaríamos qué debe de incluir nuestro proyecto y no desarrollar ni de menos ni de más.

La problemática de la entrada de datos era muy grande, debido a que no teníamos

acceso directo al PLC. Por lo tanto, tomamos la decisión de que se simularía a través de un servicio FTP, donde se colocarían archivos de un formato X (.db, csv, o Excel) en un directorio y el Enterprise Service Bus se encargaría de procesar para detectar los eventos. Sin embargo, debido a una falta de tener un coordinador dentro de la empresa, no sabíamos en qué formato tomar los datos. En base a la facilidad de procesamiento, utilizaríamos archivos en formato csv.

Finalmente, decidimos tomar los requisitos en formato de storyboards, y en lenguaje natural aquellos aspectos que no se pudieran llevar a cabo con esta técnica. Deberemos de enunciar todos los patrones implementados en EPL, a través de la herramienta Medit4CEP, ya que es mucho más visual que el propio código. Es importante que sepamos si el cliente tiene alguna preferencia en cuanto al lenguaje utilizado para desarrollar la aplicación gráfica.

5.2.3.4 PRÓXIMOS PASOS

En esta reunión se han acordado las siguientes tareas:

- Enrique debe de contactar con la empresa y hacer que alguien, por ejemplo, Vicente o Antonio se haga cargo de la coordinación del proyecto dentro de la empresa. Para ello a la semana siguiente a esta reunión se ha de concertar una reunión para hablar estos aspectos.
- Terminar el anteproyecto para repasarlo con Gregorio, y dar últimas guías para la redacción tanto del mismo como el de la memoria final del trabajo de fin de grado.
- Preparar storyboards.
- Preparar el plan de gestión de riesgos.
- Documentar todos los fallos que vayamos teniendo a lo largo de todo el proyecto.

5.2.4. REUNIÓN 4. “COORDINACIÓN Y ANÁLISIS DE LA SITUACIÓN”

5.2.4.1 INFORMACIÓN

En esta reunión sirvió en su mayor parte para dar solución al problema de coordinación con el cliente, y volver explicar la mecánica de la tecnología CEP:

- Dirección: Parque Científico y Tecnológico, Paseo de la Innovación, 3, 02006 Albacete. Oficinas de Ingeteam S.A.
- Hora y fecha de la reunión. A las 16:30, el día 1 de febrero de 2018.

Los asistentes de la reunión son:

- Vicente Requena Montejano. Ingeniero Industrial. Especializado en aerogeneradores.
- Francisco José Polo Sánchez. Jefe de departamento de I+D+i de Ingeteam Service.
- Antonio Fernández Diez. Ingeniero Industrial. Especializado en placas fotovoltaicas.
- Gregorio Diaz Descalzo. Jefe de proyecto de este trabajo de fin de grado. Titular de la Universidad de Castilla-La Mancha.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.
- Fernando Luján Martínez. Desarrollador de trabajo de fin de grado paralelo, su tutor también es Gregorio.

5.2.4.2 ASUNTOS QUE TRATAR

En resumen, esta reunión sirve como recursos para volver a explicar la mecánica de la tecnología CEP debido a que Pedro se fue, y hacer hincapié donde tendría cabida la aplicación, bien dentro del sistema de Ingeteam o a parte. Los temas que se trataron fueron:

- ¿Quién es el encargado de coordinar el proyecto como Pedro se fue?
- Explicar otra vez cómo es la mecánica de la tecnología CEP.
- ¿Dónde vamos a integrar la aplicación?
- ¿Cómo tenemos que mostrar la ocurrencia de los eventos complejos?
- ¿Cuál es la nueva forma de trabajar después de irse Pedro?
- Ingeboards y SCADA.

5.2.4.3 INFORMACIÓN RECOGIDA

El encargado debido a la ausencia de Pedro en la empresa será Francisco José Polo. Se explicó cómo funcionaba la tecnología CEP haciendo mucho hincapié en que los desarrolladores no iban a hacer labores de estadísticos para descubrir por ellos mismos cuales son las causas por las que se desgastan los aerogeneradores. Los patrones han de ser implementados en base el enunciado de estos por el cliente en el formato: "Quiero detectar X, afectan Y, W, …, t variables y cada Z de tiempo."

Se introdujeron dos conceptos nuevos que por parte de los desarrolladores y el jefe de proyecto eran desconocidos. Hablamos de Ingeboards y SCADA. Tal y como vemos en la Ilustración 10, observamos dos sistemas o estructuras, que colaboran entre sí. Por un lado, tenemos SCADA, que es el sistema que tiene almacenados todos los datos recogidos por los sistemas de monitorización de los aerogeneradores, además, de los logs de averías, y, por otro lado, tenemos Ingeboards, plataforma donde se visualiza la información dada por SCADA de una forma más legible. A través de Ingeboards, los técnicos encargados de las reparaciones de los aerogeneradores mandarán al sistema los informes técnicos que han llevado a cabo una vez terminadas sus reparaciones. Gracias a Ingeboards se puede determinar el rendimiento de los técnicos.

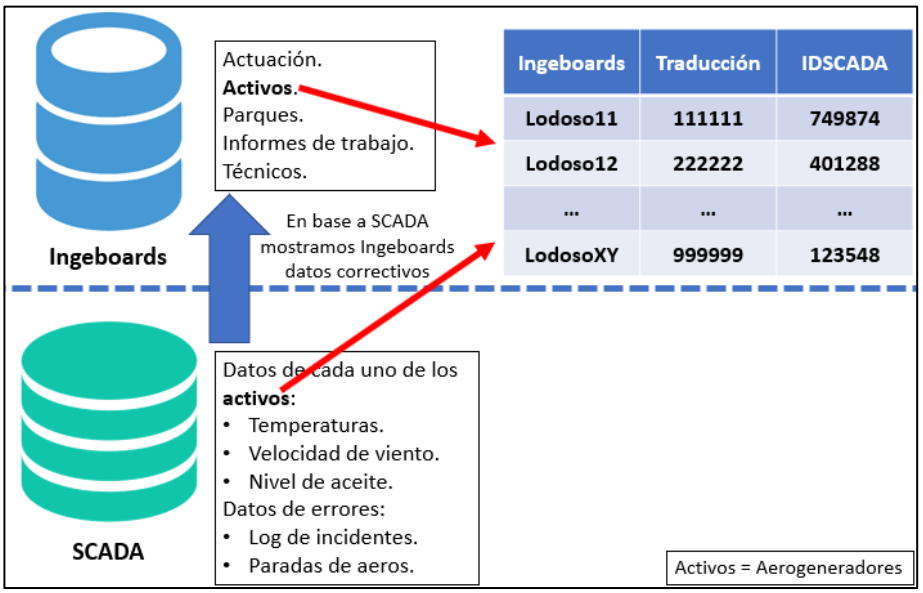

Ilustración 10. Infraestructura del entorno de Eolic Event Consumer.

Nuestra aplicación tiene que interactuar con ambos sistemas tal y como vemos en la

Ilustración 11. Como “Eolic Event Consumer” predice en base a los datos obtenidos de los activos, nuestra aplicación trabajará debajo de “Ingeboards”, y paralelamente con “SCADA”. De esta forma, tras procesar los datos que son enviados desde “SCADA” y detectar un evento de interés, se insertaría una tupla a una tabla de la base de datos de “Ingeboards”, desplegando la información en forma de alarma.

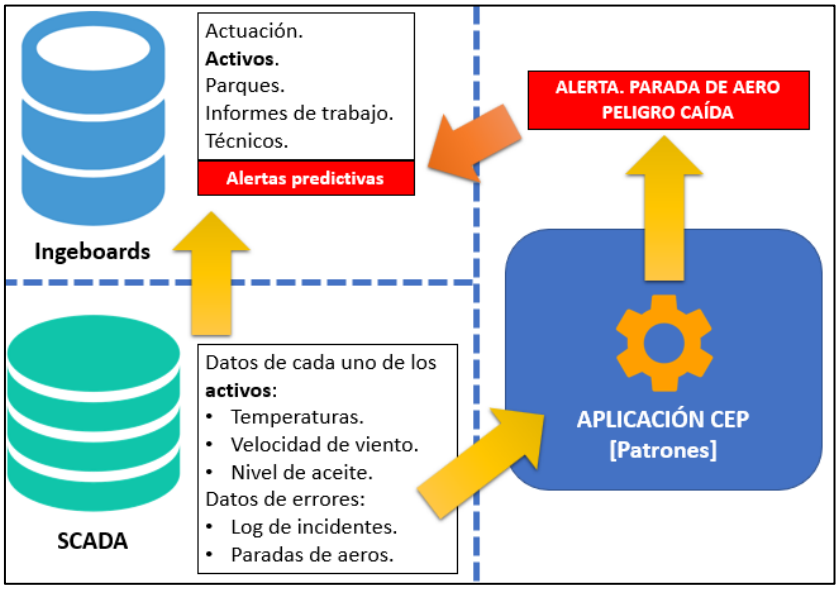

Ilustración 11. Incorporación de CEP al sistema actual.

Habiendo visto esto, como punto de partida, diremos que “NO TODOS” los campos captados por el sistema de monitorización son representativos para analizar. Solo tendremos en cuenta los factores más significativos, estos son la temperatura, la presión, la velocidad del viento y la potencia producida. Todos y cada uno de estos factores son los más importantes si examinamos el rendimiento de un aerogenerador. En concreto, haremos mucho hincapié en examinar la curva de potencia, que representa la potencia producida frente a la velocidad del viento.

No obstante, esto será nuestro punto de partida, en base a las conclusiones sacadas tras analizar estos cuatro factores iremos teniendo en cuenta el resto.

A la hora de analizar los datos deberemos de agruparlos por aerogenerador ya que las mediciones que se nos proporcionan son de un parque eólico. Por lo tanto, también sería interesante ver el rendimiento frente a otros parques para ver cuál es óptimo. Cada parque

eólico y sus medidas se han de contextualizar debido a que la situación geográfica es distinta. Tendremos que filtrar los datos puesto que hay mediciones que no son significativas y son despreciables de forma que estas pueden afectar negativamente a nuestras predicciones. Ambos sistemas tanto SCADA como Ingeboards trabajan sobre bases de datos MySQL, de forma que a la hora de detectar un evento de interés bebiendo de los datos de SCADA podemos insertar una tupla en la tabla "Alarmas" de Ingeboards.

Todo el sistema de Ingeboards está implementado con PHP 5 o posterior (no se sabía exactamente), de aquí podemos concluir en qué lenguaje deberemos de implementar la aplicación gráfica.

5.2.4.4 PRÓXIMOS PASOS

En esta reunión se han acordado las siguientes tareas:

- Integrar la lectura de datos en formato .db.
- Filtrar los datos leídos debido a la lectura de valores nulos y registros diez minútales duplicados, porque afecta muchísimo al análisis de los datos.
- Establecer un flujo de datos donde se suelten .db en un directorio y se procese toda la información recogida.
- Mandar anteproyecto e información acerca el procesamiento de eventos complejos a Francisco José Polo.
- Dar información en diez minútales del último año.
- Antonio explicará a Fernando, todo lo referente acerca placas fotovoltaicas.

5.2.5. REUNIÓN 5. CLASE TUTORIZADA DE ENERGÍA EÓLICA DE FRANCISCO JOSÉ POLO

5.2.5.1 INFORMACIÓN

Se trata de una pequeña reunión que sirvió para conocer qué factores son claves para analizar las mediciones de un aerogenerador.

- Dirección: Parque Científico y Tecnológico, Paseo de la Innovación, 3, 02006 Albacete. Oficinas de Ingeteam S.A.

- Hora y fecha de la reunión. A las 8:15, el día 2 de febrero de 2018.

Los asistentes de la reunión son:

- Francisco José Polo Sánchez. Jefe de departamento de I+D+i de Ingeteam Service.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.

5.2.5.2 ASUNTOS QUE TRATAR

Esta reunión es una clase para recoger toda la información que puede ser clave a la hora de tener que hacer el análisis. Los puntos de la reunión fueron:

- Explicación del modelo de datos utilizado.
- ¿Qué es el "SystemNumber"?
- Nociones de energía eólica para tener en cuenta.

5.2.5.3 INFORMACIÓN RECOGIDA

Se dieron diferentes nociones, que son las siguientes:

- **Formaciones en parques eólicos.** Tal y como vemos en la Ilustración 12 podemos observar que no es trivial la formación que hacen los aerogeneradores en un parque eólico. Es un balance de costes y rendimiento. En un primer momento podríamos pensar que cuantos mas aerogeneradores en un mismo espacio pues más viento aprovecharíamos. Sin embargo, esto no es así, ya que una vez que pasa el viento a través de los aerogeneradores el viento genera turbulencias de forma que el viento pierde fuerza, y no es igual de aprovechable que al principio. Es un balance costes y rendimiento, ya que a mayor número de aerogeneradores más costes, a más distancia entre activos más costes debido al uso de cableado, de forma que en cada situación geográfica la formación óptima es distinta.

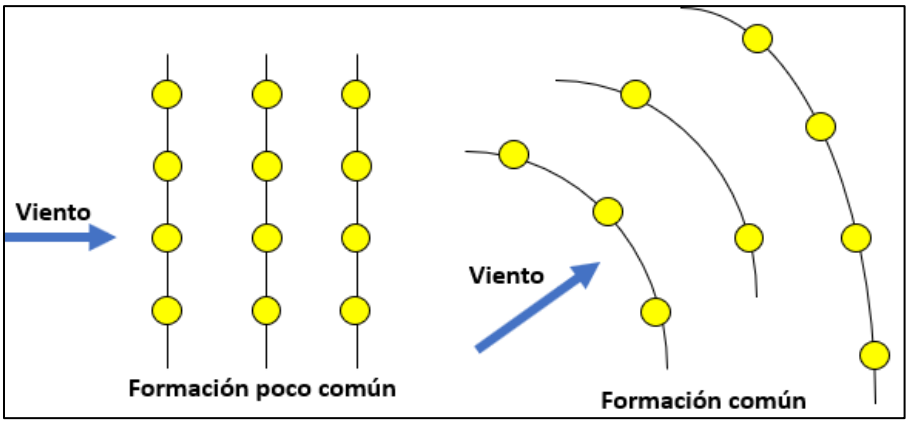

Ilustración 12. Formaciones comunes y no comunes de un parque eólico.

No hay una formación ideal, debido a que la dirección del viento predominante no es siempre igual y en cada lugar es distinto. Por esta razón los aerogeneradores se ubican en lo alto de los cerros. El aire que fluye a través de la montaña es impulsado hacia la parte de arriba del terreno por lo que aprovechamos el doble de su fuerza reduciendo el espacio por donde el viento fluye.

- **Traducción del nombre de los activos.** Cada uno de los aerogeneradores están identificados por un número, este número es el "SystemNumber". Este identificador es el que tiene dentro del sistema de SCADA. Sin embargo, en "Ingeboards" tiene otro identificador. Por ejemplo, en SCADA podríamos denominar a un aerogenerador con el id "92874621", y en Ingeboards podríamos nombrar a ese mismo con el alias "LodosoA1". Esto sucede porque el uso de la información en un sistema y otro es distinto, Ingeboards es utilizado por técnicos y SCADA es más para un uso más estadístico.

- **Factores involucrados.** A la hora de analizar los datos proporcionados por SCADA hemos de tener en cuenta una pequeña jerarquía de factores. Esta jerarquía es la vista en la Ilustración 13. A más viento mayor es la potencia producida. A más temperatura del aire, menor densidad tiene el viento, de forma que, al tener menos fuerza, la potencia producida es menor. Es decir, cuanto más frío sea el viento más fuerza produce, y mayor es la potencia producida. Por otro lado, la presión también es importante, aunque en menor medida. Cuanta más presión hay, mayor es la fuerza que produce el viento sobre las palas del aerogenerador, por lo que la potencia producida es mayor.

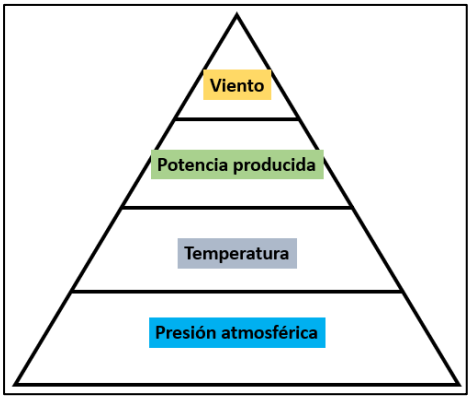

Ilustración 13. Jerarquía de factores.

- **Curva de potencia.** Todo el análisis de datos girará en torno a la curva de potencia. Esta curva esquematiza el rendimiento del aerogenerador, enfrentando el viento en ese momento contra la potencia producida tal y como vemos en la Ilustración 14. Podemos ver que hay tres fases dentro de la curva de potencia, el arranque, la situación normal y la etapa en la que está a pleno rendimiento.

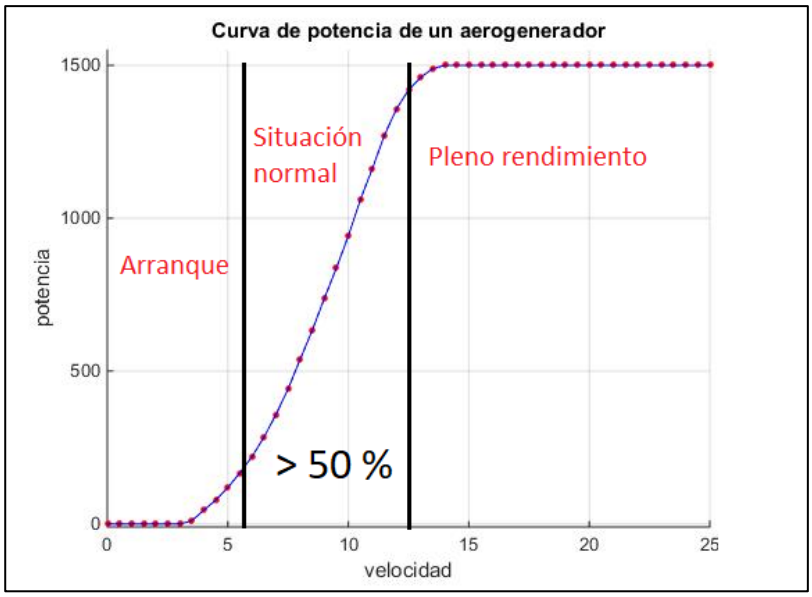

Ilustración 14. Ejemplo de curva de potencia.

Cabe destacar dos aspectos muy relevantes en cuanto a las etapas de la curva de potencia. Por un lado, diremos que más de la mitad de las mediciones de un aerogenerador se encuentran en la fase de "situación normal", por lo que los activos están diseñados para trabajar en esas condiciones al encontrarse la mayor parte del tiempo ahí. Por otro lado, tenemos la etapa de pleno rendimiento que es cuando la velocidad de viento excede de los 15 m/s, ya que la potencia es constante. Esto se debe a que los aerogeneradores sufren un mayor desgaste a velocidades muy altas, por ello se limitan para producir lo mismo en estas situaciones.

- **Defectos en la curva de potencia.** La curva de potencia se dibuja a través de una nube puntos dependiendo de las mediciones reales de un aerogenerador, sin embargo, estas mediciones tienen ruido. Este ruido podría ser por ejemplo 7 puntos con un viento de 20 m/s, pero con potencia reducida igual a 0. Esto podría ser porque el aerogenerador en cuestión está siendo reparado y se haya parado para ejercer labores de mantenimiento. Por lo que deberemos de quitar ese ruido a la hora de analizar los datos. Los defectos que podemos predecir examinando la curva de potencia son:
  - Error referenciado de pitch 0º en PLC > 0º en físico.
  - Suciedad en palas.

  - Errores en tren mecánico.
  - Defectos en las palas.
  - Defectos eléctricos generados.
  - Quitan diez minútales con error asociado.
- **Mediciones de los fabricantes para vender aeros.** No podemos comparar nuestras conclusiones con los datos que los fabricantes otorgan, ya que se obvian muchos detalles que pueden ser muy relevantes.

5.2.5.4 PRÓXIMOS PASOS

En esta reunión se han acordado las siguientes tareas:

- Estudio de la información aportada.
- Sacar conclusiones para analizar los datos.
- Limpiar de ruido la información proporcionada.
- Tener en cuenta que se ha de agrupar las mediciones por aero, y hacer comparaciones a nivel individual (por activo) o bien a nivel de grupo (por parque).

5.2.6. REUNIÓN 6. REUNIÓN DE REVISIÓN DEL SPRINT Y RETROSPECTIVA DEL SPRINT METER AQUÍ AL PRODUCT OWNER

5.2.6.1 INFORMACIÓN

Es la reunión para revisar que es lo que se ha realizado a lo largo de esta primera iteración, ver qué hemos hecho correctamente y qué problemas hemos visto, para de esta forma hallar cómo resolverlos para que no vuelvan a afectar negativamente en el proyecto.

- Dirección. Edificio Infante Don Juan Manuel. Avda. de España s/n. Albacete. Despacho 0. B.8.
- Hora y fecha de la reunión. A las 16:30 el día 14 de febrero de 2018.

Los asistentes de la reunión son:

- Gregorio Diaz Descalzo. Jefe de proyecto de este trabajo de fin de grado. Titular de la Universidad de Castilla-La Mancha.

- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.
- Fernando Luján Martínez. Desarrollador de trabajo de fin de grado paralelo, su tutor también es Gregorio.
- Francisco José Polo Sánchez. Jefe de departamento de I+D+i de Ingeteam Service.

5.2.6.2 ASUNTOS QUE TRATAR

Esta reunión sirve como revisión de lo generado a lo largo de lo que se lleva de proyecto, ver qué aspectos positivos y negativos hemos visto, y en qué podemos mejorar. Los puntos de la reunión son:

- Revisión del prototipo.
- Revisión de la documentación.
- Revisión de como generar los requisitos.
- Como genero los storyboards.
- ¿Cómo insertar la documentación generada en las reuniones?

5.2.6.3 INFORMACIÓN RECOGIDA

En cuanto a la revisión del prototipo. Hemos acordado que hemos cumplido con lo propuesto e incluido algún avance más como el ajuste de los datos, ya que el flujo de datos contiene ruido que influye negativamente en las predicciones. Queda validado, tanto el formato de datos en formato “csv” como la detección de eventos complejos y alarmas. En teoría el formato de entrada iba a ser en formato “.db”, sin embargo, debido a las dificultades que conlleva su transformación.

De momento el primer prototipo, recoge todos los datos obtenidos en “.csv”, y registro a registro alimenta al motor CEP. Los registros son diez minútales, cosa que deberemos de tener en cuenta, puesto que no son datos a tiempo real exactos, si no que están ponderados por la media de lo medido en esos diez minutos. En resumen, el prototipo generado en este primer sprint cumple las expectativas impuestas con creces, ya que no solo tenemos un incremento en el que manda información a ThingSpeak y lo retorna al ESB, sino que,

además, nos saltamos la dependencia de ThingSpeak y lo alimentamos directamente a través de ficheros ".csv". De esta forma, minimizamos dependencias y, por si fuera poco, incrementamos la lectura de los datos notablemente. En la revisión del prototipo se detectaron ciertas deficiencias en cuanto al modelo establecido, no obstante, eran simples erratas en el código que daban conflictos sintácticos muy sencillos de solucionar.

Si nos centramos en la documentación generada, se revisaron los siguientes artefactos:

- **Anteproyecto**. En cuanto a lo que el anteproyecto se refiere, he de destacar que había multitud de erratas a lo largo de todo el texto, fruto de mi inexperiencia a la hora de redactar textos formales. Se revisó cuidadosamente la justificación de las competencias, y se pidió expresamente que se incorporara alguna referencia al modelo de calidad actual. El jefe de proyecto hizo hincapié en justificar correctamente, sin aportar más información de la necesaria. Un anteproyecto corto y conciso, es mejor que uno largo y redundante. Motivación, descripción del proyecto y método, fases de desarrollo y objetivos correctos.

  En cuanto a los medios a utilizar, no sabemos con exactitud todo lo que nos hará falta, ya que hay decisiones que no hemos tomado todavía. Por ejemplo, no hemos decidido aún la tecnología que utilizaremos para hacer la aplicación gráfica, si usar node.js, o usar algo más tradicional a través de "Apache" y "Php". Personalmente, mi postura es utilizar "Php" puesto que puede servirme como reto personal ya que nunca había utilizado "Php", y "node.js" ya lo había visto en asignaturas como "Procesos de Ingeniería del Software" y "Web Engineering and Services".
- **Storyboards para captura de requisitos de patrones.** Los storyboards preparados para capturar los requisitos de los patrones eran correctos, y útiles a la hora de implementar todos y cada uno de los patrones. Echando un vistazo a las viñetas, teniendo nociones muy básicas de energía eólica, cualquier persona puede entender lo que pretenden los patrones implementar. Un ejemplo de storyboard puede ser el de la Ilustración 15.

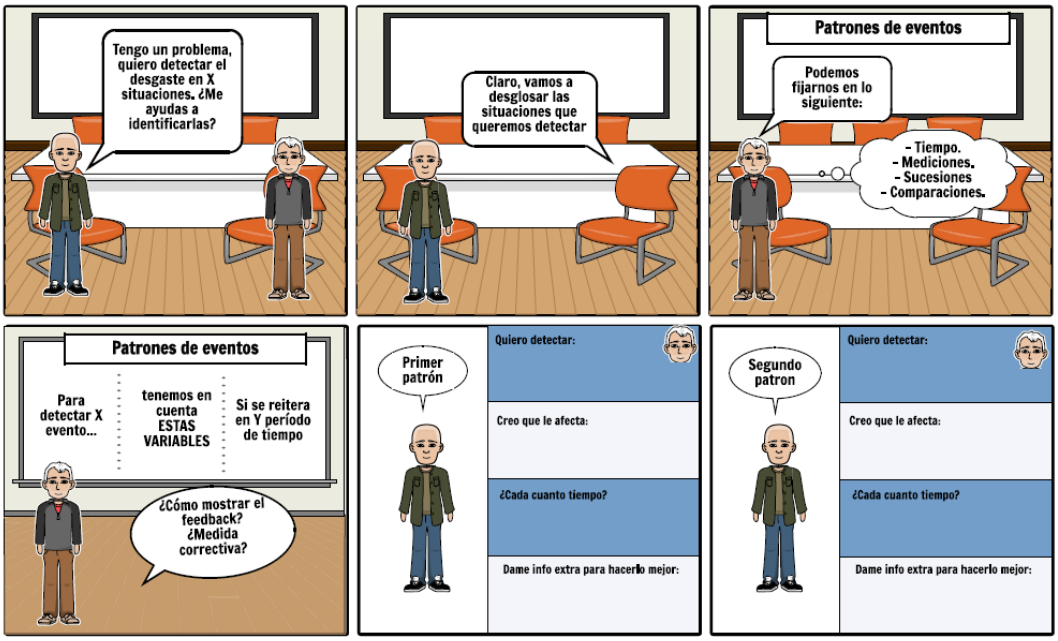

Ilustración 15. Plantilla de enunciado de patrones.

- **Actas de reuniones.** Una de las dudas del Team, es donde iban a tener cabida toda la documentación generada en cada una de las reuniones. Debido a que esta documentación es la correspondiente a la generada en una metodología SCRUM, la plasmaremos en el capítulo 4 de esta memoria, donde se verán todas las actividades desarrolladas en cada uno de los Sprints.
- **Arquetipos.** Este artefacto se ha añadido como extra por parte propia. Donde se esquematizan los integrantes que toman parte en el proyecto.
- **Información técnica clave.** Había dudas de cómo plasmar en la memoria del trabajo de fin de grado toda la información técnica acerca de la energía eólica que se recogerían en el segundo sprint. Se decidió que tendría cabida en el capítulo del estado del arte.

5.2.6.4 PRÓXIMOS PASOS

Se han acordado la incorporación de las siguientes historias de usuario para el siguiente sprint:

- **Estudio de factores externos distintos de la potencia producida y el viento.** En

concreto, el estudio de lo que influye la temperatura y la presión, en la productividad y rendimiento.

- **Estudio de la curva de potencia.** Deberemos de hacer un análisis individual a nivel de aerogenerador y otra a nivel de parque eólico. Buscar fuentes de información de otras universidades, o artículos referentes a la energía eólica en busca de posibles comportamientos anómalos en los aerogeneradores. Sacar conclusiones y enunciar con ayuda del Product Owner, el enunciado de los patrones.
- **Implementar patrones.** Implementar una primera versión de los patrones a raíz de las conclusiones sacadas con el análisis.
- **Documentación**. Para adelantar y no dejar para el último momento la redacción de la memoria, se pone como tarea escribir el primer y segundo capítulo. Cuidar el estilo de la memoria, así como incorporar en el capítulo pertinente toda la información correspondiente al primer Sprint.

5.2.7. RETROSPECTIVA DEL SPRINT

Una vez hecha la reunión de revisión del Sprint, el Product Owner abandonó la sala, para que el Team, junto con el Scrum Master, tuvieran la retrospectiva. El objetivo de esta reunión fue responder a las siguientes preguntas:

¿Qué salió bien en la iteración?

- Se hizo un buen prototipo, que tenía más funcionalidades de las esperadas.
- Dio tiempo a hacer parte de la documentación referente a todas y cada una de las reuniones.
- Se solucionó la coordinación con el cliente, pese al incidente de la ida de Pedro.
- La elección del ESB de MuleSoft, puesto que tiene multitud de funcionalidades.
- Hay muy buena comunicación entre el Team, y el Scrum Master. No hay ningún tipo de problema a la hora de reunirnos.
- La documentación está muy bien detallada.
- Los Storyboards plasman de forma correcta todo.

¿Qué no salió bien en la iteración?

- Debido a la mala configuración de TFS, el Sprint Burndown no refleja correctamente lo introducido en el servicio de Microsoft. Suponemos que es debido a que las tareas se redactaron conforme avanzó el Sprint.
- La conversión de los ficheros de ".db" a ".csv". No se pudo llevar a cabo de forma automática debido a que el único lector de .db que permitía procesos "batch" que los transformaba era "paradox-reader", sin embargo, no funciona correctamente puesto que la tarea se queda colgada y permite la conversión de dos ficheros seguidos.

¿Qué mejoras vamos a implementar en la próxima iteración?

- La implementación de los patrones.
- Amoldamiento de los patrones al ESB.
- Instauración de la base de datos MySQL donde introduciremos las alarmas de interés.
- Si sobra tiempo, integrar de forma automática la conversión de ficheros ".db".
- Empezar la memoria del trabajo de Fin de Grado.

5.3. SPRINT 2. ADAPTACIÓN AL CASO REAL

A continuación, tendremos la explicación de lo que se desarrolló en el Sprint 2.

5.3.1. REUNIÓN 7. REUNIÓN DEL COMIENZO DEL SEGUNDO

5.3.1.1 INFORMACIÓN

En resumen, esta reunión sirvió como punto de inicio de este sprint. En ella se establecieron los objetivos a conseguir y se estipuló la funcionalidad que debería de cumplir este prototipo en la segunda iteración.

- Dirección. Edificio Infante Don Juan Manuel. Avda. de España s/n. Albacete. Despacho 0. B.8.
- Hora y fecha de la reunión. A las 17:00 el día 24 de febrero de 2018.

Los asistentes de la reunión son:

- Gregorio Diaz Descalzo. Jefe de proyecto de este trabajo de fin de grado. Titular de la Universidad de Castilla-La Mancha.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.
- Fernando Luján Martínez. Desarrollador de trabajo de fin de grado paralelo, su tutor también es Gregorio.

5.3.1.2 ASUNTOS QUE TRATAR

Esta reunión sirve como revisión de lo generado a lo largo de lo que se lleva de proyecto, ver qué aspectos positivos y negativos hemos visto, y en qué podemos mejorar. Los puntos de la reunión son:

- ¿Qué va a tener nuestro prototipo?
- ¿Qué patrones he detectar? ¿Qué va a ser objeto de estudio?
- ¿En qué plataforma hacemos la aplicación gráfica? ¿Node.js (sin complicación, ya sabía en su día utilizarlo) o usar PHP (un reto, no sabía utilizarlo)?
- ¿Algún problema con el cliente?

5.3.1.3 INFORMACIÓN RECOGIDA

Lo más representativo de esta reunión fue el planteamiento de objetivos de este sprint. Los objetivos que se deberían de cumplir serían los siguientes:

- **El prototipo deberá de desplegar los patrones que se implementen dentro del ESB.** Lo más relevante de este sprint es el desarrollo de estos patrones, deberemos de centrarnos en el estudio de la curva de potencia tanto a nivel de parque eólico como a nivel de aerogenerador individual.
- **Los eventos complejos de interés deberán de insertarse en una base de datos MySQL.** El Scrum Master sugirió utilizar un todo en uno para incorporar tanto PHP como algún sistema gestor de base de datos como es MySQL, en concreto se tomó la decisión de utilizar XAMPP. Se eligió XAMPP porque es gratuito, y era el

que más documentación tenía para poder consultar dudas.

- Deberemos de traducir los patrones una vez implementados e integrados en el ESB, a través de Medit4CEP, ya que es mucho más visual su representación que el propio código. Sirve como apoyo adicional al código para su comprensión.

Otro punto de la reunión fue el estipular que patrones deberíamos de implementar. En su día en una de las reuniones con el cliente, nos requirió examinar con escrupulosidad la curva de potencia. Según las reuniones con el cliente se hizo mucho hincapié en estudiar el comportamiento de los aerogeneradores en los distintos bines de viento, es decir, el rendimiento que tienen en intervalos de viento concretos. También es interesante estudiar productividad, factor que enfrenta potencia producida y tiempo. Tendremos que normalizar el patrón de alguna forma puesto que el factor geográfico es determinante a la hora de que un aerogenerador es más productivo que otro. También se sugiere contrastar las conclusiones que saquemos con el log de errores para así comprobar que la alerta que el motor CEP ha sacado es correcta, esto último se llevaría a cabo solo en caso de terminar el proyecto antes de tiempo. Por el momento, solo abarcaremos todo lo que tenga que ver con rendimiento en los diferentes bines de viento, y productividad.

A continuación, se trató el tema de la aplicación gráfica que se iba a implementar. Se discutieron varios tipos de tecnologías, todas y cada una de ellas, servicios web. Sobre todo, hemos de destacar la discusión de usar Node.js y un sistema tradicional a través del uso de PHP. Por un lado Node.js, nos daba la ventaja de que ya lo habíamos utilizado, cosa que no nos suponía ningún problema, sin embargo, el cliente utilizaba PHP en su sistema, y su equipo técnico no tenía ninguna noción acerca de Node.js, cosa que dificultaba mucho su integración en el sistema. Por esta razón, y por mis ganas de aprender con el desarrollo de este trabajo de fin de grado, se decidió utilizar PHP, tal y como el servicio técnico de informáticos del cliente utilizaba. Esta decisión facilitó la integración de nuestro trabajo, con el sistema de “Ingeboards” que el cliente estaba desarrollando y me ayudó a adquirir más conocimientos acerca de una tecnología que hasta el momento me era desconocida.

Debido al problema de coordinación se dedicó unos minutos para solucionar este altercado. Exceptuando el problema de coordinación que se solucionó gracias a la intervención de Francisco José Polo, no hubo ningún otro problema que destacar.

5.3.1.4 PRÓXIMOS PASOS

Como consecuencia de esta primera reunión queda fijado llevar a cabo los objetivos nombrados en el punto 3 de esta reunión.

5.3.2. REUNIÓN 8. REUNIÓN PARA EXPLICACIÓN DE

5.3.2.1 INFORMACIÓN

En resumen, esta reunión sirvió como punto de inicio de este sprint. En ella se establecieron los objetivos a conseguir y se estipuló la funcionalidad que debería de cumplir este prototipo en la segunda iteración.

- Dirección. Edificio Infante Don Juan Manuel. Avda. de España s/n. Albacete. Despacho 0. B.8.
- Hora y fecha de la reunión. A las 17:00 el día 22 de febrero de 2018.

Los asistentes de la reunión son:

- Gregorio Diaz Descalzo. Jefe de proyecto de este trabajo de fin de grado. Titular de la Universidad de Castilla-La Mancha.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.
- Fernando Luján Martínez. Desarrollador de trabajo de fin de grado paralelo, su tutor también es Gregorio.

5.3.2.2 ASUNTOS QUE TRATAR

En esta reunión se propusieron las primeras versiones de los patrones, para que el Scrum Master

- Fundamentos teóricos y explicación de patrones.
- Aceptación o denegación de los patrones presentados.

5.3.2.3 INFORMACIÓN RECOGIDA

Tras unos días estudiando las nociones teóricas de la energía eólica junto con algunos expertos procedentes de la empresa cliente, hemos sacado varias conclusiones y un primer acercamiento a los patrones a implementar. Los patrones que he pensado por el momento miden productividad y rendimiento. La gran diferencia entre ellos la podemos ver en la Tabla 7.

Tabla 7. Factores enfrentados.

Rendimiento	Productividad
Viento VS Potencia producida	**Tiempo VS Potencia producida**

Como podemos ver, por un lado, uno enfrenta viento y potencia producida al intervalo de tiempo que deseemos. Por otro lado, tenemos otro que enfrenta la potencia producida diariamente. A continuación, veremos la explicación de los mismos.

a. Patrón de Rendimiento

Para medir el rendimiento nos centraremos en el estudio de la curva de potencia de los aerogeneradores. Sin embargo, hay ciertos problemas a la hora de comparar "el buen rendimiento", ya que no todos los molinos, reciben en un instante X, una velocidad de viento Y. El viento en una región tan amplia como un parque eólico es variable, ya bien sea por la disposición de los aerogeneradores (en matriz, en filas irregulares, etc.) o bien por la disposición geógrafica en la que se encuentran (uno encima de un cerro, otro en mitad de una llanura, etc.). Por tanto, este patrón lo que intenta es ver qué aerogeneradores son los mejores en distintos intervalos de viento. Los intervalos de análisis son unitarios, es decir, de 0 a 1, de 1 a 2, y así hasta 18, puesto que a partir de 18 se da la potencia nominal del aerogenerador que estamos estudiando. Por ejemplo, en la Tabla 8, vemos el estudio del rendimiento en el intervalo que va de 6 a 7 y de 7 a 8, en el día 1 de enero.

Estas mediciones son comparables entre sí, y sí que se podrían analizar por algún tipo de criterio sin tener en cuenta la disposición geográfica. Por ejemplo, se podrían ver las desviaciones en tanto por ciento con respecto la media, o entre ellos. A raíz de ahí, veremos que aerogeneradores son los "peores". En teoría, todos los aeros son iguales, sin embargo, si

hay una diferencia significativa puede que haya un fallo que el sistema de monitorización no haya detectado. Además, gracias a este control podremos prevenir este tipo de deficiencias. Para discretar este análisis y verlo más visual e intuitivo podemos hacer un análisis por ranking de rendimiento en cada intervalo de tiempo.

Tabla 8. Mediciones rendimiento el día 1 de enero.

ID de aero	Intervalo m/s	Media del intervalo	Sigma	Media del aero	Predicción	Posición ranking
1	[6,7)	427.80 W	75.61 W	423 W	A determinar	3º
2	[6,7)	427.80 W	75.61 W	425 W	A determinar	2º
3	[6,7)	427.80 W	75.61 W	450 W	A determinar	1º
1	[7,8)	658.27W	90.12 W	650 W	A determinar	1º
2	[7,8)	658.27W	90.12 W	610 W	A determinar	3º
3	[7,8)	658.27W	90.12 W	636 W	A determinar	2º

Teniendo en cuenta la tabla anterior hemos visto que en el intervalo de tiempo [6,7), que las posiciones en función de la media obtenida por aerogenerador es en primera posición el aero 3, en segunda el aero 2, y en tercero el aero 1.

Si lo representamos en un evento,

$$Evento(x, y, z)\ tal\ que\ x\ día\ del\ año,$$

$$y\ \{media\ de\ aeros\},$$

$$z\ es\ un\ intervalo\ [a, a+1)\ tal\ que\ a\ \in [0{,}18)\ \subseteq \mathbb{R}$$

tendría la siguiente forma:

$$Ranking(1\ de\ enero, \{450{,}425{,}423\}, [6{,}7)) = \{aero\ 3, aero\ 2, aero\ 1\}$$

Veremos pues que los elementos del evento quedan ordenados, según los datos obtenidos. Bien, supongamos que el día 2 de enero se obtiene, el siguiente evento:

$$Ranking(2\ de\ enero, \{463{,}450{,}428\}, [6{,}7)) = \{aero\ 2, aero\ 3, aero\ 1\}$$

Como podemos observar dentro de este ranking, de un día para otro, el aero 3 ha bajado una posición. Esa posición a nivel mecánico puede significar algún tipo de desperfecto en la multiplicadora, una subida de temperatura que no debiera de darse, etc. Esto puede ser un evento de interés a ser analizado. La idea es que el algoritmo revele estos desplazamientos dentro del ranking para hacer predicciones a corto-medio plazo.

b. Patron de Productividad

En cuanto a lo que productividad se refiere, enfrentamos dos factores muy importantes que son la potencia producida y el tiempo. Esta vez, como hemos comentado anteriormente, el factor geográfico y la posición de los aerogeneradores es muy relevante para encontrar un criterio que nos permita decir cuál es más productivo que otro. En la Ilustración 16, vemos lo que producen 6 aerogeneradores a lo largo de un día. Como podemos observar sus líneas de tendencia lineales indican cual es más productivo el uno con respecto el otro. Sin embargo, estas mediciones son dependientes del viento. Por tanto, el que reciba más viento, es el que más produce.

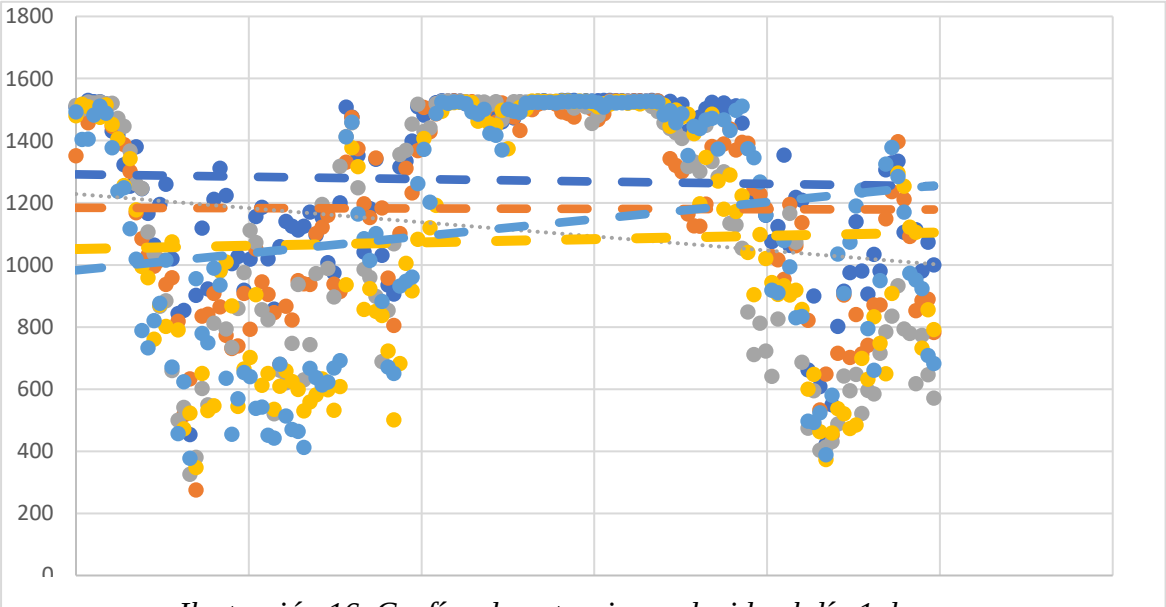

Ilustración 16. Grafíco de potencia producida el día 1 de enero.

Dicho lo anterior, hemos de incorporar el estudio del contexto de cada generador, tanto el factor geográfico, como su lugar dentro de la formación del parque. Debido a estos dos criterios, en condiciones normales, todos y cada uno de los aeros están destinados a ser uno más productivos que otros. Por lo tanto, no podemos hacer un análisis cuantitativo únicamente fijándonos en los valores numéricos de su producción, obviando la situación dentro del parque del aerogenerador.

Por tanto, lo que haremos es analizar los desplazamientos dentro de ese orden al que están destinados a producir cada uno de los aerogeneradores. En la **¡Error! No se encuentra el origen de la referencia.**, vemos un ejemplo de las mediciones de los días 1 y 2 de enero.

Tabla 9. Medidas de productividad el día 1 de enero.

ID de aero	Día	Potencia producida (en día)	Nº Eventos representativos	Media del aero	Posición ranking
1	1 de enero	4200 W	3	1400 W	2º
2	1 de enero	3000 W	2	1500 W	1º
3	1 de enero	3000 W	3	1000 W	3º
1	2 de enero	4100 W	3	1366 W	1º
2	2 de enero	3200 W	3	1066 W	3º
3	2 de enero	4000 W	3	1333 W	2º

Supongamos que el orden de producción es el conjunto ordenado descendientemente por productividad $\mathrm{R1(ideal)} = \{\mathrm{aero2}, \mathrm{aero1}, \mathrm{aero3}\}$. Veremos que el día uno de enero no cumple con el orden ideal, podremos decir que el parque ha producido lo que se esperaba el día 1 de enero, puesto que $R2(día\ 1) = \{aero2, aero1, aero3\}$. Sin embargo, si analizamos el día dos, vemos que $R3(día\ 2) = \{aero1, aero3, aero2\}$. Si analizamos R2 y R3, vemos que la productividad del aero 2 ha bajado muchísimo, esto no debería de pasar, puesto que el aero 2 en condiciones normales, debería de estar de los primeros. Esta bajada, puede ser señal de que hay algún fallo mecánico, alguna reparación, un evento inesperado etc. En base a las traslaciones de esta especie de ranking podremos implantar medidas preventivas cuando más nos convenga.

ATENCIÓN. “Que el aerogenerador dibuje una curva de potencia más elevada con respecto el resto, no implica que produzca más (sea el más productivo)”

Una vez expuestos los patrones a implementar, el Scrum Master, dio el visto bueno a los patrones a implementar, era una buena solución y con una dificultad adecuada. Sin embargo, quedó pendiente que el cliente lo verificará y viera el visto bueno o las apreciaciones necesarias para ajustarse a sus necesidades.

5.3.2.4 PRÓXIMOS PASOS

Como consecuencia de esta reunión y antes de empezar a implementar los patrones, deberemos de comunicarnos con el cliente y cerciorarnos de que las ideas planteadas son correctas y solventa sus necesidades. Por tanto, Enrique se deberá de reunir a solas con Francisco José Polo para confirmar el enunciado de los patrones, así como estructurarlos y definir de una vez por todas qué patron se va a implementar, y cuál es el límite de nuestro proyecto.

5.3.3. REUNIÓN 9. REUNIÓN PARA CONCRETAR Y VALIDAR LOS PATRONES CON EL CLIENTE

5.3.3.1 INFORMACIÓN

En esta reunión se procedió a la validación de los fundamentos funcionales que tienen los patrones ideados, así como posibles retoques que mejorarán la exactitud de los eventos de interés a detectar.

- Dirección: Parque Científico y Tecnológico, Paseo de la Innovación, 3, 02006 Albacete. Oficinas de Ingeteam S.A.
- Hora y fecha de la reunión. A las 10:00 el día 23 de febrero de 2018.

Los asistentes de la reunión son:

- Francisco José Polo Sánchez. Jefe de departamento de I+D+i de Ingeteam Service.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo

de fin de grado.

5.3.3.2 ASUNTOS QUE TRATAR

Los puntos que se desarrollarán en esta reunión son los siguientes:

- Fundamentos teóricos y funcionamiento de los patrones.
- Aceptación o denegación de los patrones ideados.
- Posibles mejoras en los patrones.
- Apreciaciones que se podrían hacer extras utilizando el log de errores.

5.3.3.3 INFORMACIÓN RECOGIDA

En primer lugar, Enrique expusó todos los puntos que definen los patrones ideados, que intentan medir productividad y rendimiento. Una vez hecho esto Francisco José dio el visto bueno a los patrones. Sin embargo, corregió algunos aspectos que eran incorrectos, mejorando así el funcionamiento del algoritmo CEP.

Dio la aprobación completa al patrón para esquematizar la productividad, le pareció interesante. Destacó que analizar cualititativamente los aerogeneradores como si fueran corredores en una carrera, normalizando las condiciones geográficas, mediante los desplazamientos en un ranking, era una buena idea.

Una de las apreciaciones más importantes a la hora de llevar a cabo el análisis, es que a la hora de estudiar el rendimiento que los intervalos de viento no fueran unitarios, si no que fueran por bines de viento. Es decir, que se analicen intervalos de un medio de dominio, tal que si un evento era:

$$Evento(x, y, z)\ tal\ que\ x\ día\ del\ año,$$

$$y\ \{media\ de\ aeros\},$$

$$z\ es\ un\ intervalo\ [a, a+1)\ tal\ que\ a\ \in [0{,}18)\ \subseteq \mathbb{R}\ ,$$

Después de la intervención del cliente estipulamos que un evento queda representado como:

$$Evento(x, y, z)\ tal\ que\ x\ día\ del\ año,$$

$$y\ \{media\ de\ aeros\},$$

$$z\ es\ un\ intervalo\ [a, a + 0.5)\ tal\ que\ a\ \in [0{,}18)\ \subseteq \mathbb{R}$$

Las variaciones de potencia producida en los intervalos centrales de la curva son demasiado grandes, de tal forma que al minimizar el intervalo obtendremos más precisión en el análisis, ya que reducimos las desviaciones en los diferentes dominios de viento. Además, sugirió separar el análisis a través de tres patrones distintos pero complementarios entre sí, son los siguientes:

- **Análisis de desviaciones.** Cada X tiempo se recogerían los datos de los distintos aerogeneradores, de tal forma que haríamos un análisis de sus desviaciones típicas y consideraremos que aquel que tenga una desviación mayor, es más irregular, y, por tanto, peor con respecto el resto.
- **Análisis de puntos fuera de los intervalos de confianza en los bines de viento**. Este patrón nace a raíz del anterior, ya que, a través de la desviación típica obtenida de cada aerogenerador en un bin de viento, podremos obtener el intervalo de confianza de sus mediciones. Contaremos cuantos puntos se quedan fueran de ese intervalo, de tal forma que aislaremos qué puntos no son normales dentro de los datos obtenidos en ese tiempo. Estos puntos pueden ser el indicio de que pueda haber alguna avería mecánica a corto-medio plazo. Por lo general, podremos tener la hipótesis de que el aerogenerador que más puntos tenga fuera de su intervalo de confianza será el peor. También se sugirió hacer un ranking con este criterio.
- **Análisis de media y bajadas en tanto por ciento del unos con respecto otros**. A través de las medias obtenidas por cada aerogenerador en los diferentes bines de viento, analizaremos las bajadas de rendimiento, con respecto sus propias medias o de las del resto de aerogeneradores. Es interesante analizar para un bin de viento, porqué un aerogenerador ha bajado su rendimiento en un tanto por ciento con respecto lo obtenido en el día anterior.

5.3.3.4 PRÓXIMOS PASOS

En esta reunión se confirmaron todos y cada uno de los patrones a implementar, hasta este momento no se acotó la amplitud del proyecto, ni las funcionalidades que iba a tener el prototipo de esta iteración. Hecho esto, aquí empezó la implementación de los patrones. Las tareas fruto de esta reunión son:

- Implementar el patrón para medir productividad.

- Implementar el patrón para medir rendimiento.
- Cambiar el filtrado de datos para que los intervalos sean los correspondientes a los bines de viento, y no en intervalos unitarios.
- Cambiar el flujo de envío de alertas del ESB, para que se añada una tupla a la base de datos MySQL correspondiente al evento complejo detectado.

5.3.4. REUNIÓN 10. REUNIÓN PARA REVISIÓN DE CÓDIGO Y COMUNICACIÓN DE PATRONES DEFINITIVOS

5.3.4.1 INFORMACIÓN

En resumen, esta reunión sirvió como punto de inicio de este sprint. En ella se establecieron los objetivos a conseguir y se estipuló la funcionalidad que debería de cumplir este prototipo en la segunda iteración.

- Dirección. Edificio Infante Don Juan Manuel. Avda. de España s/n. Albacete. Despacho 0. B.8.
- Hora y fecha de la reunión. A las 17:00 el día 1 de marzo de 2018.

Los asistentes de la reunión son:

- Gregorio Diaz Descalzo. Jefe de proyecto de este trabajo de fin de grado. Titular de la Universidad de Castilla-La Mancha.
- Enrique Brazález Segovia. Desarrollador y encargado de defender este trabajo de fin de grado.
- Fernando Luján Martínez. Desarrollador de trabajo de fin de grado paralelo, su tutor también es Gregorio.

5.3.4.2 ASUNTOS QUE TRATAR

Esta reunión sirvió como recurso para comunicar los patrones definitivos y para labores de coordinación dentro del trabajo de fin de grado. Los puntos de la reunión fueron:

- Comunicación de los patrones definitivos.

- Exposición de dudas en un patrón, puesto que no se sabía en su día como plantearlo.
- Coordinación del trabajo de fin de grado.

5.3.4.3 INFORMACIÓN RECOGIDA

Se puso en conocimiento al Scrum Master de cuáles iban a ser los patrones definitivos a implementar, ya dado el visto bueno por parte del cliente. El Scrum Master vio correcto el enunciado de los patrones, y que eran posibles de implementar.

A continuación, ya se había empezado a implementar el patrón que tenía que ver con la productividad. Sin embargo, la solución propuesta no estaba completa ya que a la hora de hacer un SELECT con una ventana temporal de tipo BATCH, no se conseguía añadir el ROW NUMBER para poder añadir una columna más que indique la posición de los aeros dentro del ranking. Sin embargo, se pensó a hacer de otra forma, capturando tantos eventos seguidos como el número de aerogeneradores multiplicado por dos. De esta forma, capturaremos el ranking obtenido en dos días consecutivos y compararemos los unos con respecto los otros para ver si han cambiado las posiciones de un día para otro.

Solucionado este problema a nivel de implementación, pasamos a la coordinación del trabajo de fin de grado. Debido a que solo quedarían tres meses para la entrega del mismo, el Scrum Master mandó a ambos alumnos empezar a elaborar la memoria en la medida de lo posible, en concreto, con los capítulos del estado del arte y a documentar el primer sprint. Además, se estipuló que iba a ver una reunión semanal a partir de la semana del 19 de marzo debido a que había tener un mayor control de lo que se iba haciendo, y así evitar posibles atascos en las tareas del proyecto.

5.3.4.3.1 PRÓXIMOS PASOS

Las tareas que deberemos desarrollar después de esta reunión son principalmente terminar de elaborar los patrones, empezar la memoria de trabajo de fin de grado, y continuar las tareas impuestas en el backlog.

CAPÍTULO 6. EXPERIMENTOS Y RESULTADOS

En este apartado metemos todas las pruebas funcionales unitarios y de integración llevadas a cabo en este proyecto. Así como pruebas de calidad.

CAPÍTULO 7. CONCLUSIONES Y PROPUESTAS

7.1. CONCLUSIONES

7.2. TRABAJO FUTURO

BIBLIOGRAFÍA

En esta sección veremos todas las referencias bibliográficas de las que ha nacido este trabajo de fin de grado a través del estándar IEE.

[1] X. Zhao, H. Fan, H. Zhu, Z. Fu, and H. Fu, "The Design of the Internet of Things Solution for Food Supply Chain," *Proc. 2015 Int. Conf. Educ. Manag. Inf. Med.*, vol. 49, no. Emim, pp. 314–318, 2015.

[2] M. Antonio and R. Borja Díaz, "Energía eólica," vol. 1, pp. 54–57, 2013.

[3] IBM, "Business Rules Management System | IBM Digital Business Automation." [Online]. Available: https://www.ibm.com/cloud/automation-software/business-decision. [Accessed: 29-Jan-2018].

[4] R. E. SIEMENS Gamesa, "Gamesa consolida una tecnología propia para el mantenimiento predictivo de sus aerogeneradores," *29-10-2008*, 2018. [Online]. Available: http://www.siemensgamesa.com/es/comunicacion/noticias/gamesa-consolida-una-tecnologia-propia-para-el-mantenimiento-predictivo-de-sus-aerogeneradores.html?idCategoria=0&fechaDesde=&especifica=0&texto=&fechaHasta=. [Accessed: 14-Mar-2018].

[5] Microsoft Corporation, "La Arquitectura Orientada a Servicios (SOA) aplicada al mundo real," 2006.

[6] S. O. Tortosa, A. O. Baíllo, and R. B. Plata, "Arquitectura Orientada a Servicios Para La Distribuidos," pp. 3–10, 2006.

[7] D. Luckham and R. Schulte, "Event Processing Glossary - Version 2.0," *Processing*, no. July, pp. 1–19, 2011.

[8] S. Moreno Saiz, "Estudio de Arquitecturas Software para Servicios de Internet de las Cosas," 2015.

[9] Exforsys Inc, “Arquitectura impulsada por eventos SOA 2.0 | Capacitación y consultoría en TI - Exforsys,” *7-7-2007*, 2007. [Online]. Available: http://www.exforsys.com/tutorials/soa/soa-event-driven-architecture.html. [Accessed: 16-Mar-2018].

[10] Chris Taylor, “El papel del CEP en un entorno de Arquitectura Orientada a Servicios,” *12-10-2017*, 2017. [Online]. Available: https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/el-papel-del-cep-en-un-entorno-de-arquitectura-orientada-a-servicios. [Accessed: 16-Mar-2018].

[11] P. Russom, “Big data analytics,” *TDWI Best Pract. Rep.*, pp. 1–35, 2011.

[12] M. Döhring *et al.*, “The convergence of workflows, business rules and complex events: Defining a reference architecture and approaching realization challenges,” *ICEIS 2010 - Proc. 12th Int. Conf. Enterp. Inf. Syst.*, vol. 3 ISAS, no. January 2010, pp. 338–343, 2010.

[13] J. Boubeta, “Model-Driven Development of Domain-Specific Interfaces for Complex Event Processing in Service-Oriented Architectures.,” p. 223, 2014.

[14] J. B. Puig, G. O. Bellot, and I. M. Bulo, “Procesamiento de Eventos Complejos en Entornos SOA: Caso de Estudio para la Detecci{ó}n Temprana de Epidemias,” *VII Jornadas Cienc. En Ing. Serv.*, pp. 63–76, 2011.

[15] Alberto del Barrio de Pablos, “Introducción al procesamiento de Eventos Complejos I - Decide Soluciones,” *2013-09-15*, 2018. [Online]. Available: http://decidesoluciones.es/introduccion-al-procesamiento-de-eventos-complejos-i/. [Accessed: 15-Mar-2018].

[16] B. B. M. Michelson and E. Links, “Event-Driven Architecture Overview 2011,” *Architecture*, 2011.

[17] EsperTech, “Esper - EsperTech.” [Online]. Available: http://www.espertech.com/esper/. [Accessed: 29-Jan-2018].

[18] Esper, “Esper Reference,” *Esper Core Engine*, vol. 2014, p. 2014, 2012.

[19] Oracle, “Introduction to Oracle CQL,” 2009. [Online]. Available: https://docs.oracle.com/cd/E16764_01/doc.1111/e12048/intro.htm. [Accessed: 28-Mar-2018].

[20] C. G. V, “SQLstream s-Server,” 2018.

[21] SQLStream, “SQLStream downloads and pricing.” [Online]. Available: http://sqlstream.com/download/. [Accessed: 28-Mar-2018].

[22] SAP Company., “SYBASE Price List as of 2 Feb 2018,” *2/2/2018*, 2018. [Online]. Available: http://www.kernelsoftware.com/products/catalog/sybase.html. [Accessed: 28-Mar-2018].

[23] J. Maréchaux, “Combining service-oriented architecture and event-driven architecture using an enterprise service bus,” *IBM Dev. Work.*, no. April, pp. 1–8, 2006.

[24] Z. Cataldi, “La ingeniería del software,” *Una Metodol. para el diseño, Desarro. y evaluación Softw. Educ.*, pp. 33–57, 2000.

[25] Norberto Figuerola, “Desarrollo de Software – ¿Metodología Tradicional o Agil ? | PMQuality,” 2011. [Online]. Available: https://pmqlinkedin.wordpress.com/about/metodologia-tradicional-o-agil/. [Accessed: 19-Mar-2018].

[26] Maria Eugenia Arevalo Lizardo, “Diferencias entre Metodologías Tradicionales y Ágiles #MetodologiasAgiles,” *15-10-2011*, 2011. [Online]. Available: https://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/. [Accessed: 19-Mar-2018].

[27] M. Trigas Gallego and A. C. Domingo Troncho, “Gestión de Proyectos Informáticos. Metodología Scrum.,” *Openaccess.Uoc.Edu*, p. 56, 2012.

[28] Kanbantool, “Metodología Kanban.” [Online]. Available:

https://kanbantool.com/es/metodologia-kanban. [Accessed: 27-Mar-2018].

[29] J. Joskowicz, “Reglas y prácticas en eXtreme Programming,” *Univ. Vigo. España*, pp. 1–22, 2008.

[30] M. J. Pérez Pérez, “Guía Comparativa de Metodologías Ágiles,” *Univ. Valladolid*, pp. 3–117, 2012.

[31] Eclipse, “Eclipse - The Eclipse Foundation open source community website.” [Online]. Available: https://www.eclipse.org/. [Accessed: 29-Jan-2018].

[32] Eclipse Egit, “EGit.” [Online]. Available: http://www.eclipse.org/egit/. [Accessed: 29-Jan-2018].

[33] MuleSoft, “Anypoint Studio // Documentación de MuleSoft.” [Online]. Available: https://docs.mulesoft.com/anypoint-studio/v/6/. [Accessed: 29-Jan-2018].

[34] J. Boubeta-Puig, G. Ortiz, and I. Medina-Bulo, “MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0,” *Knowledge-Based Syst.*, vol. 89, pp. 97–112, Nov. 2015.

[35] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (guía del PMBOK®)*. 2013.

ANEXOS

A.1. EJEMPLO DE USO DE LA HERRAMIENTA

adflñkajf qelrkj qer lqewrj hqlkrj qlhr lqjr lkqrk ckzfjasdlfh qenrl jqelrkj qleh hnlqwerj qw lajfa lnfladsjf asdfn ladfj aldfladjf ladjflñaeyrqehrn lqwerj oqewrh nqer

A.2. MANUAL DE USUARIO

lkadfjla jdflqjer qertkj qer'ijqtej qoier hnqert
kllñkalñkdfg laesrfj lqñwer hnladf fjowqehtn lkwrjeoyhk lñkjje wlkjwenr ñljwer k