



Gestión integrada de los procesos y máquinas para la mejora del mantenimiento y flexibilización de la producción

Nº Expediente: IMDECA/2015/82

PROGRAMA: PROYECTOS DE I+D EN COLABORACIÓN

ACTUACIÓN: IMDECA-Proyectos de I+D en colaboración

Fecha de concesión: 23 de octubre de 2015

Entregable E4.1 (E4.1)

Diseño del sistema de optimización

Pertenece al paquete de trabajo: E2

Participante responsable: ITI

Mes estimado de entrega: Mes 11

Tabla de Contenidos

1. Glosario de términos	4
2. Introducción	5
2.1. <i>Objetivos del Paquete de Trabajo 4</i>	5
2.2. <i>Objetivo del presente documento</i>	5
3. Diseño del Sistema de Optimización	5
3.1. <i>Infraestructura Big Data</i>	6
3.1.1 Descripción de la arquitectura y servicios	7
3.2. <i>Motor de correlación de datos</i>	12
3.2.1 Paquete de Análisis	12
3.2.2 Componente Experto	18
3.3. <i>Visualización de datos</i>	19
3.3.1 Mapa web	20
3.3.2 Mockups	21
4. Metodología de desarrollo	23
4.1. <i>Proceso de desarrollo</i>	24
5. Referencias bibliográficas	25

Resumen

OPTIMAN es un proyecto financiado con el Instituto Valenciano de Competitividad Empresarial (IVACE) y la Unión Europea a través del Fondo Europeo de Desarrollo Regional (FEDER).

El presente documento tiene como objetivo recoger las *especificaciones de diseño del Sistema de Optimización* de la Fábrica del Futuro. Para ello, se hará un recorrido que partirá desde la descomposición funcional y no funcional resultante de la fase de análisis de requerimientos, hasta llegar a las especificaciones de construcción del sistema desde tres perspectivas: la (1) *Infraestructura Big Data* que soporte la aplicación de técnicas de Machine Learning mediante un (2) *Motor de Correlación de Datos* que alimente a una herramienta de (3) *Visualización de Datos* para la toma de decisiones de optimización y mejora de la eficiencia.

Abstract

OPTIMAN is a project funded by the Valencian Institute for Business Competitiveness (IVACE) and the European Union through the European Regional Development Fund (FEDER).

This document aims at defining the design specifications of the Optimization System of the Factory of the Future. To this extent, first we will analyze the functional and nonfunctional decomposition resulting from the phase of requirements analysis, and second the specifications of construction of the system from three different perspectives: (1) Big Data Infrastructure that supports the Techniques of Machine Learning, (2) Data Correlation Engine and (3) Visualization Data Tool for the decision making of optimization and improvement of the efficiency.



Este documento está bajo una [Licencia Creative Commons Atribución-NoComercial-SinDerivar 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/). Se permite libremente copiar, distribuir y comunicar públicamente esta obra siempre y cuando se reconozca la autoría y no se use para fines comerciales. No se puede alterar, transformar o generar una obra derivada a partir de esta obra.

Los derechos de autor de todas las marcas, nombres comerciales, marcas registradas, logos e imágenes pertenecen a sus respectivos propietarios.

1. Glosario de términos

En el presente documento se utilizará algunos conceptos que deben ser definidos con anterioridad a su lectura para aclarar ambigüedades y facilitar el entendimiento.

- **Fabrica del Futuro (FoF):** Fábrica inteligente capaz de adaptar el proceso productivo a las necesidades de producción haciendo uso de las nuevas tecnologías a fin de aumentar la eficiencia, la calidad y reducir el impacto medioambiental.
- **Big data:** Conjunto de tecnologías que permiten el manejo de grandes volúmenes de datos que no pueden ser tratados con las tecnologías convencionales.
- **Machine Learning:** Conjunto de técnicas que permiten crear algoritmos capaces de generalizar comportamientos a partir de información no estructurada.
- **Business Intelligence:** Conjunto de técnicas y herramientas que sirven de apoyo a la toma de decisiones a partir de datos de diferentes fuentes: generados como parte de la actividad empresarial, obtenidos del contexto, diferentes procesos, etc.
- **Aplicación:** programa informático diseñado para ayudar a los usuarios finales para resolver actividades específicas. En OPTIMAN, las aplicaciones se construyen de forma independiente, pero facilitando la integración con otras aplicaciones mediante el uso de servicios e interfaces desacopladas.
- **Operación:** programa informático cuyo objetivo se centra en la gestión de la información de elementos industriales externos al sistema. En OPTIMAN, estos elementos serán principalmente sensores conectados a las operaciones y estaciones industriales.
- **Servicio (web):** tecnología utiliza un conjunto de protocolos que permite intercambiar datos entre aplicaciones. Un conjunto de servicios puede ser orquestados con el fin de proporcionar servicios web más complejo. Un servicio es generalmente diseñado para proporcionar la independencia de la plataforma de usuario final, y la posibilidad de estar compuesto con los demás, de modo que puedan ser integrados directamente en aplicaciones de terceros.
- **Paquete:** representa la agrupación lógica de servicios, operaciones y/o aplicaciones que cooperan entre sí para producir un resultado. Es importante destacar que, en línea con los principios que rigen el enfoque arquitectónico en OPTIMAN, los servicios procedentes de diferentes paquetes son lo suficientemente flexibles como para ser combinados en diferentes aplicaciones.
- **Componente:** agrupación de paquetes, aplicaciones y servicios que cooperan entre sí para implementar un conjunto de funcionalidades comunes y que representan un objetivo superior a ellos mismos.
- **Sistema:** contempla el total de componentes, paquetes, servicios y aplicaciones que permiten alcanzar los objetivos generales del proyecto.

2. Introducción

2.1. Objetivos del Paquete de Trabajo 4

Este paquete de trabajo *tiene el objetivo de diseñar y desarrollar un Sistema de Optimización que sea capaz de mejorar y optimizar el sistema productivo de la Fábrica del Futuro (FoF)*. Para ello, el sistema deberá contemplar:

- Tecnologías en el dominio de *Big Data* para proveer de un **almacén de grandes volúmenes de datos** provenientes del sistema de sensorización resultante del paquete de trabajo 3.
- Técnicas estadísticas de **procesamiento y análisis de datos** para la modelización del sistema de producción que permita la *predicción* y *detección* de: (1) anomalías de funcionamiento, (2) calidad de producción y (3) optimización energética
- Técnicas de **visualización de datos** que permitan interpretar grandes volúmenes de datos para la toma de decisiones relativas a la mejora y optimización a través de la calibración, acciones correctivas, etc. Como por ejemplo, visualización de datos a través de la clustering, gráficos de control, árboles de decisión, gráficos de comparación, etc.

2.2. Objetivo del presente documento

El principal objetivo del entregable E4.1 es el de recoger las **especificaciones de diseño del Sistema de Optimización** de la Fábrica del Futuro. Para ello, se hará un recorrido que partirá desde la descomposición funcional y no funcional resultante de la fase de análisis de requerimientos, hasta llegar a las especificaciones de construcción del sistema desde tres perspectivas: la (1) *Infraestructura Big Data* que soporte la aplicación de técnicas de Machine Learning mediante un (2) *Motor de Correlación de Datos* que alimente a una herramienta de (3) *Visualización de Datos* para la toma de decisiones de optimización y mejora de la eficiencia.

Se pondrá especial atención a los requerimientos relacionados con el procesamiento y acceso eficiente a grandes volúmenes de datos no estructurados, necesarios para alimentar a las técnicas capaces de calcular modelos estadísticos para la predicción y detección de anomalías de funcionamiento, optimización de la calidad de producción y eficiencia energética.

3. Diseño del Sistema de Optimización

El diseño de sistema de optimización pasa por elaborar, a partir de los requerimientos vistos con anterioridad, un conjunto de especificaciones de diseño que permitan su desarrollo y puesta en marcha.



Figura 1: Arquitectura del Sistema de Optimización

Dichas especificaciones estarán divididas en los tres bloques representados en la figura:

1. A partir de los requerimientos no-funcionales, las especificaciones de la **Infraestructura Big Data** que soportará el funcionamiento y comunicación del conjunto total de elementos del sistema de optimización. Se pondrá especial atención a las especificaciones relacionadas con el procesamiento y acceso eficiente a grandes volúmenes de datos.
2. Los requerimientos funcionales permitirá especificar los objetivos y necesidades **Motor de Correlación de Datos** necesario para dotar al sistema del componente de predicción y detección automática. Para ello, se estudiarán y especificarán las diferentes técnicas estadísticas y Machine Learning.
3. Los casos de uso identificados nos permitirán especificar un conjunto de interfaces de **Visualización de Datos** que faciliten a los usuarios el acceso a herramientas de toma de decisiones y de mejora en los procesos de fabricación.

3.1. Infraestructura Big Data

Aprovechando el potencial de las técnicas de análisis de *Big Data*, que proporcionan los recursos para el análisis de grandes volúmenes de datos, se estudiarán valores medidos en grandes secuencias temporales de producción, con el objetivo de:

- Realizar el **análisis experimental u observacional** que incluya las variables relativas a la energía, la emisión de contaminantes y la calidad final como parámetros a cuantificar.
- Realizar el **análisis y establecimiento de modelos estadísticos** que permitan caracterizar el proceso productivo.
- **Detectar anomalías** respecto al proceso habitual de producción, con el fin último de detectar tendencias anómalas y predecir posibles fallos de producción antes de que éstos se produzcan.
- **Facilitar el mantenimiento predictivo** de las máquinas implicadas en el proceso de fabricación.

3.1.1 Descripción de la arquitectura y servicios

Este módulo se encargara de proporcionar la infraestructura de cómputo distribuida y tratamiento de datos al resto de componentes del sistema, consistirá en un clúster de computadores (Conjunto de máquinas que actúan como una) centrándose en tres ejes:

- **Volumen:** capacidad de gestionar grandes volúmenes de datos. La Fábrica del Futuro generara cantidades inmensas de información que ha de ser correctamente almacenada ofreciendo ciertas garantías de seguridad ante pérdida.
- **Velocidad:** velocidad en la que se gestionan los datos. En algunos entornos no sólo es importante gestionar grandes volúmenes de datos sino que, además, esta se realice en un tiempo acotado (tiempo real).
- **Variedad:** capacidad de integrar y relacionar datos de múltiples fuentes y tipos. Hoy en día la información significativa para un contexto concreto puede generarse desde distintos lugares y en distintos formatos.

Arquitectura Lambda

La arquitectura de este módulo seguirá el patrón lambda descrita en la figura. Este patrón proporciona una solución al acceso a datos masivos en tiempos razonables apoyándose en tres capas con funciones concretas que cito a continuación:

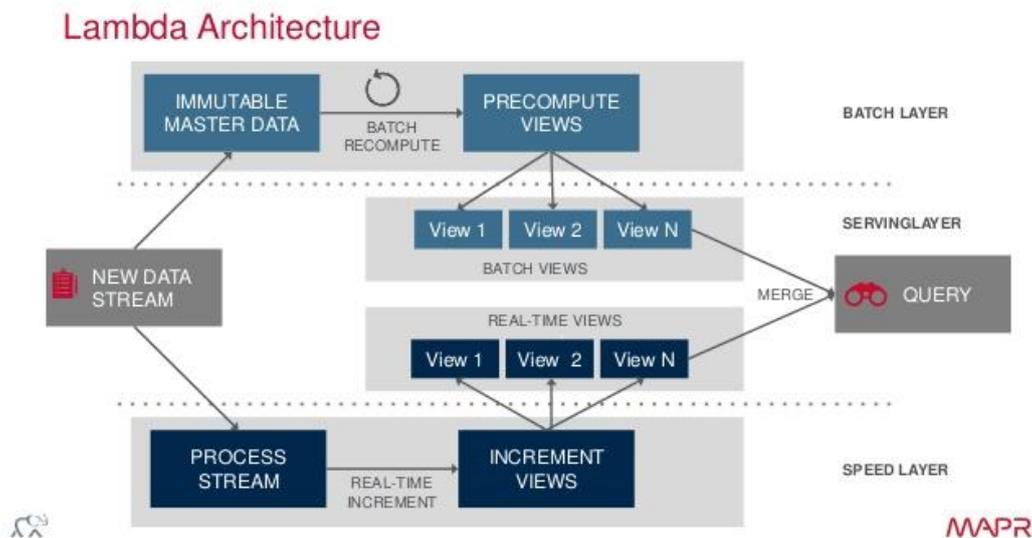


Figura 2: Esquema de arquitectura lambda

La **Capa Batch** almacena todo el Dataset y pre computa una serie de vistas, almacenadas normalmente en una base de datos de solo lectura, estas vistas generadas a partir de todo el Dataset, suelen estar normalizadas y se les ha aplicado un proceso de cleanup para que no contengan datos anómalos, de este modo simplificamos el acceso a los datos para otras capas.

La **Capa de Velocidad** da servicio a datos en tiempo real, sacrifica la consistencia de la capa Batch a fin de proporcionar acceso a los datos con una latencia mínima.

La **Capa Servicio** proporciona el acceso a los datos de las capas anteriores, se podrán consultar los datos procesados por la capa Batch o acceder a los flujos de datos en tiempo real.

Para la implementación de esta arquitectura hemos seleccionado la solución para Big data de **Cloudera**, que es una distribución de Apache Hadoop que incluye una pila de servicios que con la que implementamos la arquitectura lambda, en el siguiente diagrama se muestran los distintos servicios que se incluyen en la distribución.

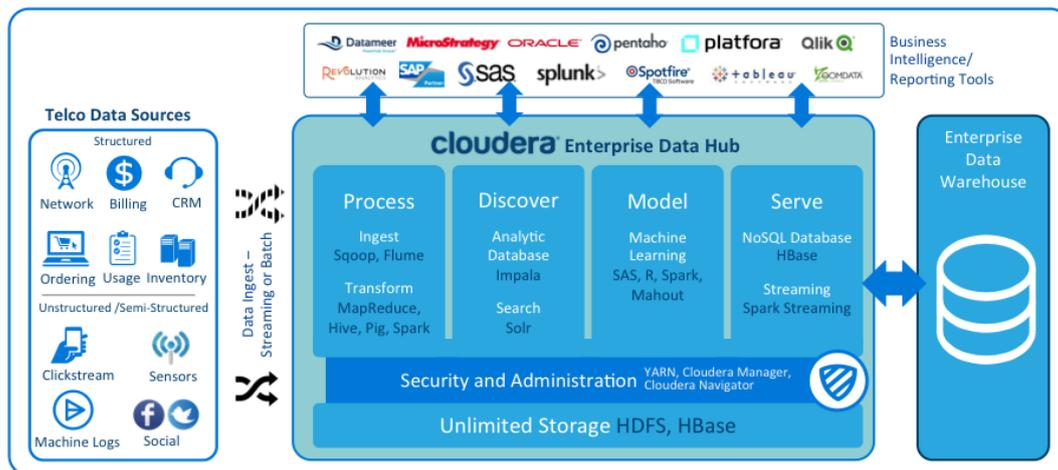


Figura 3: Arquitectura software de Cloudera

Servicios de gestión del clúster

La administración y gestión del clúster se realizara con el servicio **Cloudera manager**, este componente permite añadir dinámicamente nodos al clúster y la instalación de servicios sobre el mismo. Cloudera es un software de gestión de clúster de computadoras (nodos) basados en tecnología **Hadoop** y que provee de un ecosistema de componentes de código abierto que cambia la forma en que las empresas almacenar, procesar y analizar datos.

La gestión de recursos del clúster la gestiona el servicio **YARN**, este servicio se encarga de la gestión de los recursos del sistema, estableciendo unos pools de CPU y memoria que se asignan por usuarios o grupos para la *ejecución de trabajos* en el clúster, este servicio establece una forma común de acceso a los recursos del sistema compartida por otros servicios permitiendo de este modo una mayor interoperabilidad.

Servicios de procesamiento de datos

Para el procesamiento de la información en modo distribuido y en modo Batch usaremos el servicio **HADOOP**. Este servicio se encarga de proporcionar un api de programación JAVA distribuida basado en el paradigma map/reduce. Permite el procesamiento de los fragmentos de información distribuida en Batch sobre hdfs aprovechando el principio de localidad, esto quiere decir que cada nodo procesa la información localizada en sí mismo, reduciendo de este modo el tráfico de red cuello de botella de otros sistemas de procesamiento distribuido.

Para el procesamiento de la información en modo distribuido y en memoria usaremos el servicio **SPARK** que consta de varios Apis con los que podemos cargar y manipular datos y procesarlos con una serie de librerías para machine learning. Apache Spark es framework de código abierto

para el procesado de datos masivos desarrollado en torno a la velocidad, facilidad de uso y análisis sofisticados.

Servicios de almacenamiento de datos

El servicio **HDFS** proporcionara la capa de persistencia distribuida que da soporte al resto de servicios, cada fichero que se añade al clúster es repartido por los distintos nodos con un factor de replicación predefinido (normalmente de tres partes), esta distribución replicada de la información nos da tolerancia a fallos ya que en caso de que un fragmento de información se corrompa, de forma transparente es reparado con los fragmentos replicados. Ha sido desarrollado específicamente para el procesamiento de datos a gran escala, donde la escalabilidad, flexibilidad y rendimiento son críticos, HDFS acepta datos en cualquier formato, independientemente del esquema, esta optimizado para el procesado de flujos de datos de gran tamaño y es capaz de gestionar inmensas cantidades de datos (las escalas de las implementaciones probadas llegan a más de 100PB).

Para el almacenamiento distribuido de tablas disponemos del servicio **Hive**, que es un Dataware House distribuido que permite el acceso a la información mediante queries SQL estándar, internamente esas consultas se transforman en procesos distribuidos que obtienen los datos pedidos. Es una infraestructura de almacenamiento de datos construida sobre Hadoop para proporcionar agregación de datos, consulta y análisis. Proporciona un lenguaje similar a SQL llamado HiveQL con metadatos que transforma de forma transparente las consultas a procesos map/reduce.

Servicios de ingesta de datos

Para la ingesta de datos de tipo relacional (Bases de datos SQL) al clúster utilizaremos el servicio **SQOOP** que permite la importación de tablas a HDFS o Hive en paralelo. Sqoop es una aplicación con interfaz de línea de comandos que permite la transferencia de datos entre las bases de datos relacionales y Hadoop (HDFS, Hive, etc.) Es compatible con cargas incrementales de una sola tabla o una forma libre vía consultas SQL, así como trabajos guardados que se pueden ejecutar varias veces para importar actualizaciones realizadas en una base de datos desde la última importación.

Para la ingesta de datos de tipo fichero de log al clúster utilizaremos el servicio **Flume** que permite la importación de los mismos a HDFS o Hive en paralelo. Apache Flume proporciona un api mediante el cual se pueden definir una serie de agentes (programas residentes en una maquina cliente) que se encargan de monitorizar uno o varios ficheros de log, en caso de haber cambios se envían los mismos al clúster, de este modo centralizamos ficheros de log procedentes de varios sistemas en un mismo punto para su posterior análisis.

Implementación

Para la implementación de la arquitectura lambda propuesta se ha optado por aprovisionar los nodos del clúster sobre una infraestructura disponible en OpenStack, sobre dicho clúster se han proporcionado un conjunto de recursos virtualizados consistentes en 64 cores virtuales y 262 gigas de RAM, asi como una topología de red también virtual y un número limitado de instancias (imágenes de sistemas operativos) variables en recursos sobre el total disponible, en nuestro caso hemos optado por siete instancias con las siguientes características en función del rol.

- Rol master: 16 cores y 32 gigas de RAM sobre un Ubuntu virtualizado.
- Rol worker: 8 cores y 32 gigas de RAM sobre un Ubuntu virtualizado.

El rol master ejerce el control y reparto de trabajos entre las instancias con rol worker, las instancias con rol worker se encargan del cómputo de los trabajos. En la siguiente figura se muestra el listado de instancias creadas y la asignación de los recursos virtuales.

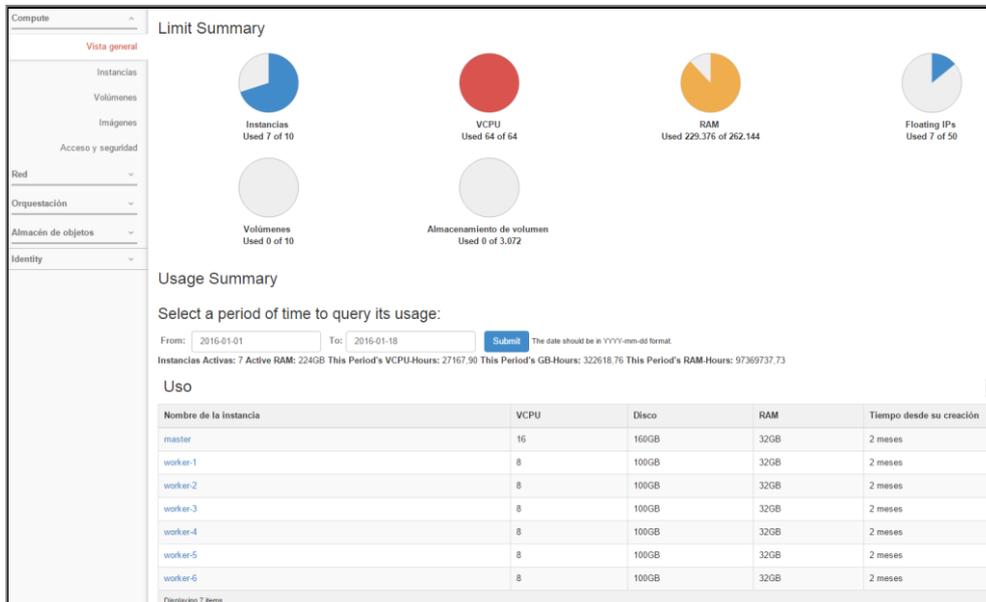


Figura 4: Listado de instancias en OpenStack

Topología de red

Cada una de las instancias pertenecen a una red virtual privada (“red_BigData”) en el rango 10.0.0.x, dentro de esta red cada uno de los nodos del clúster tienen visibilidad entre sí, por otro lado se les proporciona acceso desde y hacia internet mediante un router (“router_BigData”) que habilita el acceso al clúster desde el exterior para la ingesta de datos y el acceso a los servicios del mismo. La figura siguiente muestra un diagrama con la topología implementada.

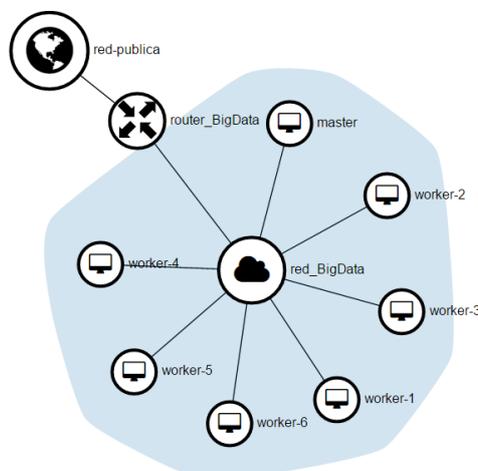


Figura 5: Topología de red en OpenStack

Gestor del clúster y servicios

Como gestor del clúster se ha optado por Cloudera manager, ya que es la distribución Hadoop de uso más generalizado, la instancia master asume el rol del gestor del clúster, a tal fin se le ha instalado el agente de control proporcionado por Cloudera a partir del cual se han descubierto el resto de nodos mediante un patrón por nombre de maquina (“worker-x”), mediante el asistente de gestión de Cloudera se ha procedido a instalar los agentes de control sobre cada uno de los nodos de computo, tras el proceso de instalación se ha procedido a la instalación de los servicios que se muestran en la siguiente figura, quedando de este modo los servicios comentados anteriormente disponibles para su uso.



Figura 6: Servicios instalados en Cloudera

La siguiente figura muestra el conjunto de nodos controlados por Cloudera, mostrando un resumen de sus características como nombre, ip, numero de roles (por servicios), ultimo latido, promedio de carga, uso de disco y uso de memoria RAM, el conjunto de recursos de computo sobrantes tras la instalación de los servicios quedan disponibles para tareas de computo distribuido, del mismo modo es posible la ampliación o decomisado de nodos para ampliar o reducir las capacidades de computo del clúster.

Estado	Nombre	IP	Roles	Último latido	Promedio de carga	Uso del disco	Memoria física
●	master	10.5.0.10	19 Role(s)	hace 5.29s	0,93 0,77 0,79	63.3 GiB / 192.1 GiB	9.3 GiB / 31.4 GiB
●	worker-1	10.5.0.11	12 Role(s)	hace 14.85s	0,24 0,16 0,15	25.6 GiB / 133.1 GiB	3.2 GiB / 31.4 GiB
●	worker-2	10.5.0.13	6 Role(s)	hace 9.13s	0,01 0,03 0,05	24.5 GiB / 133.1 GiB	2.1 GiB / 31.4 GiB
●	worker-3	10.5.0.12	6 Role(s)	hace 9.61s	0,01 0,06 0,05	24.3 GiB / 133.1 GiB	2.1 GiB / 31.4 GiB
●	worker-4	10.5.0.15	5 Role(s)	hace 3.06s	0,00 0,01 0,05	24.3 GiB / 133.1 GiB	2 GiB / 31.4 GiB
●	worker-5	10.5.0.14	5 Role(s)	hace 2.49s	0,04 0,04 0,05	24.2 GiB / 133.1 GiB	1.9 GiB / 31.4 GiB
●	worker-6	10.5.0.16	5 Role(s)	hace 8.26s	0,01 0,03 0,05	24.4 GiB / 133.1 GiB	2 GiB / 31.4 GiB

Figura 7: Instancias en Cloudera

3.2. Motor de correlación de datos

Basándonos en medidas de precisión tomadas en tiempo real mediante distintos sensores instalados en máquina, se desarrollarán algoritmos de aprendizaje para inferir los parámetros de configuración óptimos para un determinado par producto/máquina. El objetivo es conseguir un sistema parametrizable y adaptable a distintos entornos que ayude a maximizar la calidad final de la pieza minimizando el consumo de energía que requiere dicho proceso de fabricación.

La función de este módulo será procesar, analizar e interpretar información del proceso productivo en la Fábrica del Futuro a través de:

- El **análisis experimental u observacional** que incluya las variables relativas a la energía, la emisión de contaminantes y la calidad final como parámetros a cuantificar.
- El **análisis y establecimiento de modelos estadísticos** que permitan caracterizar el proceso productivo.
- La **detección de anomalías** respecto al proceso habitual de producción, con el fin último de detectar tendencias anómalas y predecir posibles fallos de producción antes de que éstos se produzcan.

Para ello, será necesaria la construcción de dos paquetes funcionales que permitan abordar el conjunto total de necesidades en materia de optimización:

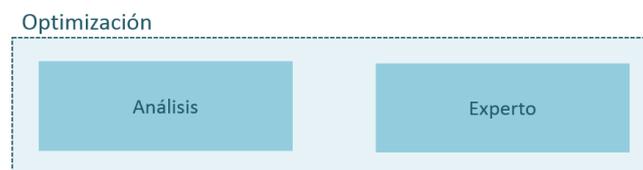


Figura 8: Paquetes del Motor de Correlación de Datos

3.2.1 Paquete de Análisis

Este módulo tiene como finalidad el análisis estadístico de las medidas recibidas desde los sensores de la fábrica, así como otra información relevante del funcionamiento y operativa diaria. Esto incluye acciones de mantenimiento realizadas por los operarios, modelo que se está fabricando, datos de consumo energético, calidad de la pieza fabricada.

Estos datos servirán para construir y entrenar toda una serie de modelos estadísticos que, una vez generados, serán alimentados en tiempo real con datos diarios capturados por el módulo de sensorización. Una vez puestos en producción, estos modelos generarán eventos de forma periódica en función de los datos leídos en tiempo real y de los históricos. Estos eventos alimentarán el módulo experto.

Este módulo constará de una serie de modelos estadísticos implementados con la tecnología requerida según el caso, contemplando actualmente el uso de R y Python como lenguajes de referencia.

El uso de esta arquitectura nos proporciona libertad para usar la tecnología adecuada a cada modelo estadístico que se quiera utilizar. Al ser una arquitectura modular cada componente de análisis podrá estar implementado de una manera tradicional (R, Python) o integrado con Apache Spark.

Cada modelo estadístico constará de cuatro módulos con funciones estancas que se describen a continuación:

Módulo de Precarga

Se encarga de adaptar los datos en crudo a las tablas normalizadas para ese modelo estadístico. Éste componente es personalizable por origen de datos y será de ejecución periódica controlado por un sistema de control planificado tipo cron. La implementación de la precarga será realizada mediante scripts R/Python (con posibilidad de usar Apache Spark) o mediante procesos ETL diseñados a tal efecto. La importación masiva de datos al componente de Big data se contempla en la descripción de dicho modulo.

La operativa de los modelos descriptivos o predictivos será individual por cada componente de fabricación o elemento de la fábrica. Estos datos almacenados por el módulo de sensorización podrán obtenerse del módulo de persistencia. Los datos necesitan ser alineados temporalmente para conformar los individuos multidimensionales en estudio. Para dicho alineamiento se ha de considerar el hecho las frecuencias diferentes en la toma de muestras en los diferentes tipos de sensores. Este hecho conllevará la adaptación de los sensores con frecuencias más bajas a los de frecuencias más elevadas lo acarrea la necesidad de aplicar técnicas de interpolación y todo ello con la aplicación de técnicas de Big Data pues las frecuencias de algunos sensores pueden llegar a ser muy elevadas lo que implica la ingesta de una cantidad elevada de datos. Tras este alineamiento será posible el estudio de las relaciones entre variables a la hora de conseguir el modelado de los objetivos perseguidos.

Adicionalmente, es posible la agregación de los datos mediante diversos estadísticos con objeto de mejorar los modelos. Además los datos también han de ser limpiados para eliminar aquellas medidas anómalas que puedan alterar negativamente el proceso de entrenamiento de los modelos. Al mismo tiempo que se detectarán datos anómalos se valorará en qué porcentaje aparecen, si superan cierto umbral se generará un aviso para revisar el proceso de captura de datos.

La limpieza de datos para entrenamiento se llevará a cabo de manera automática y no supervisada. La limpieza se llevará a cabo mediante técnicas de clustering que nos permitirán obtener una representación mediante clústeres de cuáles son los distintos modos de operación habituales. Cualquier nueva medida que no encaje en ninguno de los clústeres conocidos será considerada un dato anómalo y será comunicado al módulo experto como evento de pronosis para su tratamiento según el sistema de reglas.

Aglutina los distintos scripts de precarga bajo una estructura común de directorios separando los mismos por tipo de modelo y donde se aplica, por ejemplo:

Módulo de Entrenamiento

Se encarga de entrenar el modelo estadístico a partir del histórico almacenado en las tablas normalizadas para ese modelo y los parámetros de configuración almacenados. Este

componente es estándar por modelo estadístico y puede requerirse o no su ejecución periódica dependiendo del modelo en cuestión. En ambos casos se controlará por cron.

Aglutina los distintos scripts de entrenamiento bajo una estructura común de directorios separando los mismos por tipo de modelo y donde se aplica.

Como ya hemos comentado estos scripts que implementan los modelos estadísticos podrán hacer uso o no de la infraestructura del componente Big Data si el volumen de los datos a procesar así lo requiere.

El resultado del entrenamiento será un fichero que contendrá el modelo, preparado para ser utilizado por el componente de prognosis que se almacenara por modelo en una estructura de directorios definida.

Módulo de Prognosis

Se encarga de generar eventos en función de los datos diarios proporcionados por los sensores, basándose en la evaluación del modelo estadístico entrenado y según los parámetros configurados. Este componente es estándar por modelo y será de ejecución periódica.

Para la implementación del componente se va a hacer uso de diferentes modelos de predicción con objeto de estimar los modelos que ayuden a predecir o clasificar el estado de la Fábrica del Futuro. Entre estos modelos cabe destacar:

1. **Modelos de predicción:** En este tipo de modelos se busca, por una parte, con un número de variables independientes que pueden ser de naturaleza cualitativa, cuantitativa discreta o cuantitativa continua explicar una o variables dependientes por medio de las variables independientes anteriormente mencionadas. Destacamos, entre todos los modelos posibles, los siguientes modelos:
 - *Modelos de regresión:* Muy útil a la hora de ofrecer interpretaciones a través de las variables independientes.
 - *Análisis de Series Temporales:* Estos modelos serán de gran utilidad cuando los valores en determinados instantes anteriores, de la propia variable a modelar, además de otras variables explicativas, son útiles como variables explicativas de la variable dependiente final en un momento dado.
 - *Redes Neuronales:* Se trata de modelos con un número de capas ocultas que permiten una gran versatilidad pero que dificultan la interpretación de las variables independientes a la hora de explicar a la variable dependiente además de requerir con frecuencia, elevados costes de entrenamiento.
 - *Máquinas de Soporte Vectorial:* Son modelos muy robustos a los datos anómalos, pero suelen requerir de elevados costes de entrenamiento.
 - *Algoritmos Genéticos:* Se trata de algoritmos en los que la búsqueda de la solución a un problema se trata aplicando procesos evolutivos en los que la mejor solución sobrevive al resto.
2. **Clasificadores:** Cuando existe la necesidad de clasificar individuos en un número finito de clases es necesario estimar modelos que permitan dicha clasificación. Estos

modelos se basan en algoritmos de aprendizaje automático o minería de datos que son capaces de entrenarse a partir de un conjunto de muestras etiquetadas de manera cualitativa con la clase a la que cada muestra de entrenamiento pertenece. Los diferentes algoritmos de aprendizaje que pueden ser aplicados a la hora de obtener dichos clasificadores dependen del tipo de datos con los que nos encontramos o el objetivo final del modelo. Como ejemplo de esto, es posible hacer uso de los mismos modelos algoritmos para el aprendizaje de modelos predictores pero con pequeñas diferencias como por ejemplo la introducción de una etapa logística final que permite discretizar la variable continua ofrecida como resultado por el modelo. También es posible la aplicación de modelos directamente pensados para tareas de clasificación como KNN, regresión multinomial o regresión logística. A modo de ejemplo destacamos:

- *Regresión Logística*: Este es un tipo de modelo de regresión a utilizar cuando se desee crear un clasificador para dos clases diferentes. El resultado final consiste en una función matemática por lo que no requiere de recursos de espacio para almacenar los datos de entrenamiento. Además al tratarse de un modelo que consiste en una expresión matemática final es factible la interpretación de los diferentes parámetros que la forman.
 - *Regresión Multinomial*: Es un modelo de regresión que permite distinguir entre más de dos clases diferentes. El resultado final es una expresión matemática por lo que no requiere de recursos espaciales para el almacenamiento de las muestras de entrenamiento. Además al tratarse de una expresión matemática la interpretación de es factible la interpretación de los diferentes parámetros que la forman.
 - *KNN*: Este modelo se basa en la búsqueda de los k vecinos más cercanos de n observaciones. La clase que se asignará finalmente como resultado de la clasificación será la más frecuente tras extraer los k vecinos más cercanos a un ejemplo dado. El modelo final requiere almacenar todas las muestras de entrenamiento junto con la clase a la que pertenecen en una estructura que permite una búsqueda posterior rápida, es por ello por lo que dicho modelo, a diferencia de los anteriormente mencionados, tiene el problema de requerimientos de espacio físico final.
3. **Modelos de Clustering**: El tipo de problemas que se pretenden modelar con este tipo de técnicas conlleva la búsqueda de agrupamientos de los individuos según su similitud con otros.
4. **Otros modelos**: a considerar en función del éxito de los anteriores son los siguientes:
- Modelos de vecindarios tipo K-NN.
 - Relevance Vector Machines.
 - Modelos de clústering (BIRCH, OPTICS, DBSCAN,
 - Modelos avanzados de aprendizaje supervisado (random forest, percentron)
 - Modelos de reducción de dimensiones y posterior clasificación (PCA, PLS-Regression, PLS-DA)

El **tipo de modelos** a aplicar depende de diversos factores como: tiempo de estimación requerido, necesidad o no de interpretación posterior de los parámetros asociados a las variables independientes, precisión, existencia de estructura temporal en la variable a ser modelada, autocorrelación en la variable dependientes, multicorrelación entre las variables independientes, elevada dimensionalidad, datos dispersos, etc.

Así pues, por ejemplo, al final es posible que se tenga un gran número de variables independientes para modelar una variable dependiente y esto puede conllevar problemas de elevada dimensionalidad para obtener un buen modelo cuando se da esta situación se requiere de un número considerable de muestras supervisadas. Puesto que la obtención de esta gran cantidad de muestras supervisadas es frecuentemente imposible, aparecen dos aproximaciones diferentes que pueden ser aplicadas en la resolución de este tipo de **problemas**:

1. *Selección de características*: Selecciona la combinación de variables que mejor explican a la variable dependiente.
2. *Análisis Multidimensional*: Transforma las variables originales que se encuentran en un espacio n -dimensional en un nuevo tipo de variables que son proyectadas en un nuevo espacio k -dimensional donde k es más pequeño que n .

Los **motores de predicción** tendrán cuatro modos principales de funcionamiento. Cada uno de estos modos de funcionamiento será lanzado a través de una llamada al script concreto que realice el funcionamiento en cuestión. Estos scripts podrán ser lanzados manualmente a través de una interfaz o podrá ser programado a través de un proceso automático que se lance periódicamente o cuando se cumpla algún criterio preestablecido tipo cron. Los modos de funcionamiento principales anteriormente mencionados son:

1. *Forecast*: Es el modo habitual de un modelo en producción dedicado a tomar entradas y realizar predicciones. Los scripts relativos a este modo se integran en el componente de pronosis bajo una estructura común tipo directorio en el sistema de ficheros.
2. *Stop*: Dejar a un modelo en este estado indicará que dejará de dar predicciones. Los scripts relativos a este modo se integran en el componente de pronosis bajo una estructura común tipo directorio en el sistema de ficheros.
3. *Build*: Estado en el que el modelo se está creando a partir de un proceso de aprendizaje con las muestras etiquetadas a tal efecto. Los scripts relativos a este modo se integran en el componente de precarga bajo una estructura común tipo directorio en el sistema de ficheros.
4. *Optimize*: Algunos modelos están gobernados por una serie de parámetros que han de ser optimizados para que sus resultados sean óptimos. Los scripts relativos a este modo se integran en el componente de entrenamiento y reentrenamiento bajo una estructura común tipo directorio en el sistema de ficheros.

Para la **detección de anomalías** es posible aplicar también técnicas de análisis multidimensional, tal y como se hace en la estimación de control estadístico de calidad multidimensional. Mediante el uso de estas técnicas se miden todas las variables que se cree que están implicadas en el proceso con la intención de detectar una anomalía posterior en una o varias variables dependientes que sean de interés. Con frecuencia, el valor que toma una variable es consecuencia de una serie de factores que podrían haber ocurrido antes en el

tiempo. A través de la aplicación de estas técnicas es posible la anticipación a problemas más serios. En el marco del control estadístico de calidad, existen dos problemas principales para ser resueltos, estos problemas están relacionados con la necesidad de obtener una baja probabilidad de falsos positivos en combinación con una baja probabilidad de bajos negativos, también conocido en el campo de la estadística como errores de tipo I y II.

La **fiabilidad** es también un campo de interés para la fábrica del futuro. Como ejemplo, podemos imaginar situaciones en las cuales es posible estimar la fiabilidad de diferentes dispositivos que se encuentran en una fábrica, como bombillas, bombas de presión, dispositivos de control, nodos wifi, sensores de temperatura, etc. La estimación de las distribuciones de probabilidad de los tiempos de vida de cada uno de los dispositivos pueden llegar a ser de gran interés debido a que el conocimiento de dicha distribución de probabilidad posibilita prever el fallo y realizar su reposición o mantenimiento en el momento adecuado consiguiendo así un funcionamiento adecuado y continuado del sistema concreto. Además, el conocer esta información permite realizar estimación sobre los costes de mantenimiento futuro, pudiendo introducir dichas previsiones de costes como variables dentro de un sistema de optimización para minimizarlos. El tipo de modelos que se pueden utilizar a la hora de resolver este tipo de problemas son modelos predictivos o clasificadores, análisis de supervivencia y distribuciones de probabilidad para los tiempos de vida.

Los modelos en estado Forecast ofrecerán sus predicciones con una frecuencia establecida en la configuración sobre una tabla de la BBDD y los sistemas de visualización accederán a dichas predicciones para ofrecer los resultados por pantalla a modo de logs, alertas y gráficas según la necesidad establecida en cada momento. Dichas predicciones serán introducidas en una tabla de Eventos que contendrá los campos que se detallan a continuación:

id_sensor	Identificador del sensor
event_time	Instante de la predicción
clasificacion	Clasificación ofrecida por el modelo clasificador (null en caso de tratarse del modelo de estimación del tiempo)
error_clasificacion	Valor de error del clasificador durante la estimación de éste
prediccion	Predicción del tiempo hasta el fallo
error_prediccion	Valor de la estimación del error obtenido a partir de la clasificación de éste
confianza	Valor de la confianza de la predicción correspondiente (deberá de valorarse como estimar esto).
procesado	A valor false, pues es un indicador que servirá posteriormente al proceso subsiguiente de reglas

La introducción de un modelo supone pasar por diferentes **fases** antes de su inclusión en el sistema. Estas fases son:

1. Inicialización:

- Obtención, alineamiento y limpieza de los datos de entrenamiento.
- Análisis descriptivo de los datos.
- División excluyente de los datos de entrenamiento en tres tipos diferentes:
 - Datos de entrenamiento.
 - Datos de validación.
 - Datos de test.

2. *Para cada tipo de modelo propuesto hacer:*
 - Entrenamiento del modelo.
 - Optimización de parámetros.
 - Evaluación y validación del modelo.
3. *Selección del modelo más adecuado para el problema en cuestión a tratar:*
Reentrenamiento del modelo con todos los datos originales
4. *Puesta en producción.*

Módulo de reentrenamiento

Se encarga de decidir si es necesario el reentrenamiento del modelo estadístico, en función del margen de error detectado durante la generación de los eventos. Este componente es estándar por modelo y será de ejecución periódica por cron.

En caso de requerir reentrenamiento se invocara al componente de entrenamiento descrito anteriormente.

3.2.2 Componente Experto

Este módulo se encarga de recibir los eventos del módulo de análisis y en función de unas reglas predefinidas y una serie de parámetros de configuración externos, decidirá si el evento definido se considera una alerta o no.

Entendemos por alerta cualquier desviación sobre el comportamiento normal basado en el histórico de datos, si por ejemplo el evento proviene de un modelo estadístico que se aplica sobre datos de consumo la alerta podría indicar un problema de mantenimiento.

Si un dato o una serie de datos constituyen una alarma es algo que debe interpretarse desde un sistema experto, que lea las salidas de los modelos a través de la tabla de eventos y pueda interpretarlas para extraer conclusiones en formato de alarmas. Para realizar esta tarea el enfoque más apropiado parece el uso de software que nos permita definir reglas de Complex Event Processing. Este tipo de sistemas nos permite gestionar eventos en el tiempo, estableciendo reglas complejas como:

- Que un evento se repita N veces durante un intervalo de tiempo determinado.
- Que dos eventos concretos se produzcan en el mismo intervalo de tiempo.
- Que se alternen valores superiores o inferiores a un rango con una frecuencia superior a la de un umbral.
- Que haya N valores seguidos que superen repetidamente un umbral, etc.

Las alertas pueden tener distinto nivel de criticidad y urgencia (warning, rotura inminente...). El módulo experto permitirá al usuario definir esos niveles y asignarlos a distintos umbrales e intervalos temporales.

Si se genera una alerta del, ésta será inmediatamente enviada al módulo de comunicaciones, que a su vez notificará al resto de componentes del sistema si así lo requieren.

Para la implementación de este módulo se ha seleccionado Drools, permitiendo aislar la lógica de reglas de los otros módulos y del propio sistema. Drools incluye un motor de razonamiento de gran potencia que permite crear y desplegar sobre él sistemas de Gestión de Eventos Complejos (Complex Event Processing) y configurarlos con gran versatilidad.

Éste módulo consistirá en una aplicación que estará a la espera de la llegada de nuevos eventos generados desde el módulo de análisis. La aplicación se encargará de detectar esos nuevos eventos, transformarlos a hechos y proceder a su inclusión en la base de hechos (asertarlos). Cada nueva aserción en la base de hechos lanza una reevaluación automática de las reglas predefinidas en el motor de razonamiento. Estas reglas son parametrizables según una configuración disponible además en el módulo de persistencia.

En función de las reglas definidas, de los umbrales de sus variables y de los intervalos temporales definidos, se decidirá si el evento pronosticado se considera una alerta o no y cuál es su gravedad. Cada regla disparada en el motor de razonamiento provocará dos acciones: (i) su persistencia en la base de datos y (ii) su transmisión al módulo de comunicaciones, a fin de ser notificada al resto del sistema para su posterior tratamiento.

Se definirán dos tipos de reglas:

- **Reglas estáticas:** Definidas en función de unos máximos y mínimos definidos a priori en la configuración. Un ejemplo de regla de este tipo sería la especificación de unos márgenes de alarma en caso de una desviación del comportamiento histórico de un sensor. Estos valores podrán ajustarse más adelante.
- **Reglas dinámicas:** Definidas en función del tiempo, por ejemplo lanzar una alerta si un sensor se sale del histórico durante un intervalo de tiempo predefinido.

Estas reglas se pueden combinar entre ellas a fin de configurar nuestro sistema de razonamiento experto.

3.3. Visualización de datos

El sistema de optimización debe incluir mecanismos de representación de datos que permitan la creación de un interfaz de consulta orientada a la explotación de la información resultante de los procesos de análisis de datos para la planificación y optimización.

En este sentido, la aplicación de técnicas informáticas convencionales en el front-end no son suficientes, sino que es necesario la aplicación de técnicas ligadas a tecnologías Big Data para representación visual grandes cantidades de datos será el objetivo principal de este componente. Aumentar las capacidades en la toma de decisiones pasa por aplicar técnicas para representar de forma ágil grandes volúmenes de datos y proveer de grafismos que faciliten su comprensión para la extracción de conclusiones y nuevo conocimiento.

Para poder llevar a cabo la representación visual de la información, es necesario realizar un análisis predictivo basado en la aplicación de funciones de agregación (promedio, cuenta, máximo, mínimo, sumatorio, desviación típica, varianza, etc.) sobre los datos con el objetivo de generar información resumida sobre las distintas dimensiones o puntos de vista posibles.

En definitiva, aplicar un conjunto de técnicas relativas al dominio de la minería de datos permite encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

A partir de estos patrones es posible realizar análisis adicionales como son la clasificación de datos, predicción de tendencias, identificación de correlaciones, búsqueda de clústeres, simulación de escenarios, construcción de árboles de decisión, análisis de dependencias, etc.

De esta manera podemos convertir grandes cantidades de datos en información fácil de representar de forma gráfica, pero en este caso lo que se permite es de una manera sencilla prever el futuro en base a criterios objetivos y, en definitiva, tomar decisiones basadas en el conocimiento.

3.3.1 Mapa web

El mapa web contiene el conjunto de páginas que formaran parte del interfaz culpable de representar al componente de visualización de datos. Su diseño nos permitirá ofrecer una vista general de la solución y nos ayudará a navegar por el interfaz para acceder a cada uno de los interfaces desarrollados en los mockups.

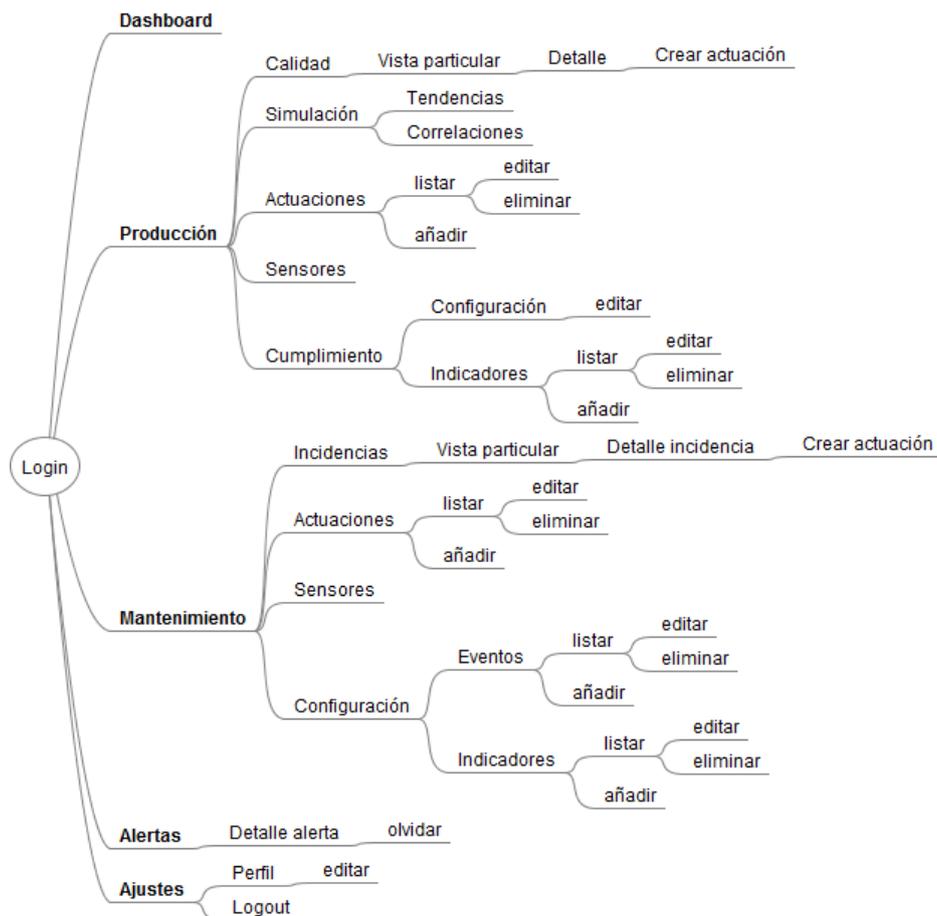


Figura 9: Mapa web

El componente de visualización de datos estará representado por una herramienta web en la que los usuarios podrán identificarse para acceder a tres bloques principales:

1. **Dashboard:** panel que recoge un conjunto de widgets que resumen el estado de sistema en materia de optimización de la calidad de producción y mantenimiento predictivo.
2. **Producción:** conjunto de funcionalidades que permiten controlar la calidad de producción en función a unos parámetros de configurables que establecen una relación entre la energía consumida y la calidad del producto. Permitirá realizar simulaciones de las variables que nos permitan estimar cuáles serán los parámetros de nuestro sistema de forma ideal, así como planificar actuaciones de mantenimiento o consultar los datos de los sensores.
3. **Mantenimiento:** conjunto de funcionalidades que permiten predecir anomalías de funcionamiento para detectar fallos o averías, realizar y planificar actuaciones que permitan corregir o prevenir averías, analizar los datos de los sensores, configurar los eventos que nos permitirán crear alertas o avisos, así como definir indicadores que nos digan cómo de bien está funcionando nuestro sistema.

3.3.2 Mockups

Los mockups o wireframes, son los prototipos de interfaces que se diseñan para especificar en términos generales cómo se implementará la herramienta web. A este nivel del proyecto, resulta muy complejo definir todos los mockups posibles del componente de visualización porque dependerá también de los retos resueltos durante el proceso de investigación y desarrollo.

A continuación se muestran los mockups que se han considerado como los más importantes a este nivel de desarrollo del proyecto:



Figura 10: Layout

La figura representa en layout general de la aplicación con un menú lateral que puede ser compactado, un navbar de usuario, alertas e idioma, así como una miga de pan con un contenido a continuación.



Figura 11: Dashboard

Dashboard: muestra tres widgets (aunque en el futuro puedan ser más) que recogen el comportamiento general de la Fábrica del Futuro en materia de mantenimiento predictivo y optimización del proceso de fabricación. En concreto, muestra un widget que permite conocer el consumo energético a lo largo del tiempo, un listado de las alertas de averías y las acciones correctivas llevadas a cabo, así como un conjunto de indicadores que permiten conocer el estatus de calidad del proceso productivo.

Control de calidad de la producción: Este mockup viene a representar el estado de la calidad de producción en relación a la eficiencia energética del proceso de fabricación. En concreto, representa tres widgets en los que se recoge:

1. La energía consumida de forma global en relación a un periodo concreto y un conjunto de indicadores de la calidad de fabricación.
2. Los datos de los sensores en un periodo seleccionado para determinar si el proceso de captura de datos ha sido correcto durante el continuo proceso de fabricación.
3. Un conjunto de alertas y acciones correctivas relacionadas con ese proceso de fabricación en el periodo seleccionado.

Histórico de acciones correctivas: permite al usuario acceder al conjunto total de acciones correctivas, al contrario de los vistos anteriormente. Permitirá ordenar, buscar y filtrar las alertas por cualquiera de sus parámetros: estado, prioridad, fecha, descripción, persona que solucionó el problema, el coste, grado de cumplimiento, etc. Además, permitirá no sólo el seguimiento de las acciones, sino su modificación de alguno de los parámetros descritos.

Creación de una nueva acción correctiva: Ante una alerta de malfuncionamiento o de consumo excesivo o decremento de la calidad, seremos capaces de crear acciones correctivas asociadas. Este mockup representa el formulario en el que se parametrizará cada uno de los elementos necesarios para el sistema en materia de acciones: solución propuesta, título, descripción, prioridad, coste en euros y horas, persona que lo resolverá, descripción del proceso de resolución de la incidencia, etc.

Simulación de tendencias de consumo y calidad: el usuario será capaz de simular en el futuro cuál será el consumo de energía para su fabricación en función de un conjunto de parámetros de configuración del proceso de fabricación. El modelo estadístico será capaz de recoger los datos de los sensores y realizar una predicción de cuál será el consumo necesario en el futuro, dentro de un margen de error determinado. También permitirá la creación de un reporte o informe.

Simulación de tendencias de consumo y calidad: De una forma similar al anterior, esta herramienta permitirá comparar datos de los sensores para encontrar correlaciones entre ellas. Seleccionado un producto que hemos fabricado, podemos comparar el consumo y la temperatura externa para encontrar si existe alguna relación de causalidad. El resultado puede generar un reporte.

Grado de cumplimiento de los objetivos: permite al usuario acceder a los indicadores que nos permitan conocer si se ha conseguido alcanzar los objetivos de cumplimiento en materia de eficiencia energética, averías, calidad de producción, etc.

Mediante estos dos widgets, seremos capaces de seleccionar los objetivos establecidos y conocer su evolución a lo largo de un tiempo determinado. Esto nos permitirá saber qué grado de cumplimiento hemos alcanzado con nuestras políticas de optimización.

4. Metodología de desarrollo

Cada uno de los componentes, paquetes, servicios y aplicaciones identificados en la arquitectura de referencia, pueden ser desarrollados de forma independiente gracias a la incorporación de las filosofías de integración **Mashup** y **RESTFUL**.

En particular, dicha filosofía permitirá ver cada elemento del sistema como un consumidor de contenidos que deben ser alimentados por la capa servicio de la infraestructura big data. De este modo, la arquitectura propuesta permitirá al equipo de desarrollo realizar un enfoque en el que:

- **El interfaz web será un elemento que invocará a cada uno de los servicios o componentes** que se desarrollen para orquestarlos dentro de un reto funcional superior a ellos mismos.
- **Cada servicio o aplicación podrá ser desarrollado como un elemento independiente**, interconectado con el resto de elementos a través de un API REST. Esto permitirá disponer de un entorno tecnológico específico en cada caso, como se ha presentado en la sección anterior, en el que será necesaria la generación de un conjunto de servicios web que permitan acceder y proveer contenidos.
- Podrán construirse **soluciones funcionales híbridas** en el que distintos servicios y aplicaciones de otras capas realicen las tareas de proveer contenidos, datos, etc.
- Permitirá la construcción de un **sistema escalable** en la que cada componente podrá ser desplegado en un nodo diferente.
- Permitirá la construcción de **aplicaciones y servicios** clave para soportar parte de la ejecución de procesos de negocio necesarios para la justificación del prototipo.

4.1. Proceso de desarrollo

Esta organización de las tecnologías implicará el uso de una metodología de desarrollo, basada en un modelo evolutivo y de integración continua, en el que se irán generando prototipos funcionales de forma incremental.

Este proceso de desarrollo permitirá incorporar el conocimiento generado por todos los participantes del proyecto para lograr un mejor grado de integración de los componentes. El objetivo que se persigue con esta metodología es disminuir los riesgos de dependencia funcionales entre subsistemas y componentes.

Para ello, cada componente será tratado como una **entrega o release** compuesta de un conjunto de tareas relacionadas con el diseño, desarrollo, e integración de las funcionalidades individuales comprendidas por dicha entrega.

Algunas de estas tareas serán desarrolladas de forma individual y otras, como la definición del alcance de la entrega, la validación y la integración, deberán hacerse en colaboración con la organización encargada de la validación de los requerimientos.

En la siguiente figura, se muestra una propuesta de las tareas que podrían realizarse para construir cada una de las entregas:



Figura 12: Metodología de desarrollo e integración

5. Referencias bibliográficas

[Mashup] [https://es.wikipedia.org/wiki/Mashup_\(aplicaciónn_web_híbrida\)](https://es.wikipedia.org/wiki/Mashup_(aplicaciónn_web_híbrida))

[REST] https://en.wikipedia.org/wiki/Representational_state_transfer

[OpenStack] <http://www.openstack.org/>

[Oozie] https://oozie.apache.org/docs/4.2.0/DG_Overview.html

[ZeroMq] <http://zguide.zeromq.org/page:all>

[Spark] <http://spark.apache.org/>

[Cloudera] <http://www.cloudera.com/content/www/en-us.html>

[Hadoop y HDFS] <https://hadoop.apache.org/>

[Cassandra] <http://cassandra.apache.org/>

[Flume] <https://flume.apache.org/>

[Hive] <https://hive.apache.org/>

Sqoop: <http://sqoop.apache.org/>

[Yarn] <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>

[Modelos de regresión] https://es.wikipedia.org/wiki/Regresión_lineal

[Análisis de Series Temporales] https://es.wikipedia.org/wiki/Serie_temporal

[Redes Neuronales] https://es.wikipedia.org/wiki/Red_neuronal_artificial

[Máquinas de Soporte Vectorial] https://en.wikipedia.org/wiki/Support_vector_machine

[Algoritmos Genéticos] https://es.wikipedia.org/wiki/Algoritmo_genético

[Regresión Logística] https://es.wikipedia.org/wiki/Regresión_lineal

[Regresión Multinomial] https://es.wikipedia.org/wiki/Regresión_logística_multinomial

[KNN] https://es.wikipedia.org/wiki/K-vecinos_más_cercanos

[Relevance Vector Machines] https://en.wikipedia.org/wiki/Relevance_vector_machine

[Modelos de clustering] https://en.wikipedia.org/wiki/Cluster_analysis

[Modelos avanzados de aprendizaje supervisado]
https://en.wikipedia.org/wiki/Supervised_learning

[Modelos de reducción de dimensiones y posterior clasificación]
https://en.wikipedia.org/wiki/Dimensionality_reduction