



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 E4.2

# Informe de resultados del desarrollo del sistema de optimización

Pertenece al paquete de trabajo: E4

Participante responsable: ITI

Mes estimado de entrega: 24

## Tabla de Contenidos

<b>1. Glosario de términos .....</b>	<b>3</b>
<b>2. Introducción .....</b>	<b>3</b>
<i>Objetivos del Paquete de Trabajo 4 .....</i>	<i>3</i>
<i>Objetivo del presente documento .....</i>	<i>4</i>
<b>3. Sistema de Optimización .....</b>	<b>4</b>
<i>Infraestructura Big Data .....</i>	<i>5</i>
3.1.1 Clúster de cómputo .....	5
3.1.1 Clúster de persistencia distribuida .....	7
<b>3.2 Motor de Correlación de Datos .....</b>	<b>10</b>
3.2.1 Detección de Anomalías .....	10
3.2.1.1 SPC Unidimensional .....	11
3.2.1.2 SPC Multidimensional .....	13
3.2.2 Predicción de valores futuros .....	16
3.2.3 Optimización de parámetros y objetivos de fabricación .....	18
<b>3.3 Visualización de datos y optimización .....</b>	<b>23</b>
3.3.1 Pantalla de login .....	24
3.3.2 Pantalla de cuadro de mando del sistema .....	25
3.3.3 Pantalla de configuración de planta .....	26
3.3.4 Pantalla de configuración de usuarios .....	27
3.3.5 Pantalla de simulación de parámetros de configuración .....	28
3.3.6 Pantalla de supervisión del sistema .....	29
3.3.7 Pantalla de visualización de alertas .....	30
3.3.8 Pantalla de visualización de datos de los sensores .....	31
3.3.9 Pantalla de consulta de histórico de alertas .....	32
3.3.1 Pantalla de consulta de detalle de alerta .....	33
<b>4. Validación del desarrollo .....</b>	<b>34</b>
<b>5. Referencias bibliográficas .....</b>	<b>34</b>

## 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.
- **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.
- **Motor de correlación de datos:** software que implementa un conjunto de técnicas de Machine Learning capaces de procesar datos con el objetivo de realizar predicciones, detectar anomalías u obtener indicadores de optimización.

## 2. Introducción

### 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.

## Objetivo del presente documento

El principal objetivo del entregable E4.2 es el de recoger los **resultados de la implementación realizada del Sistema de Optimización** de la Fábrica del Futuro. Para ello, se hará una descripción de los resultados obtenidos en cada uno de los puntos definidos en el diseño del sistema previamente definido.

Los puntos a describir serían: la (3.1) *Infraestructura Big Data* que soporte la aplicación de técnicas de Machine Learning mediante un (3.2) *Motor de Correlación de Datos* que alimente a una herramienta de (3.3) *Visualización de Datos* para la toma de decisiones de optimización y mejora de la eficiencia.

Se pondrá especial atención a los resultados obtenidos 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. Sistema de Optimización

El sistema de optimización implementado tiene como objetivo el almacenamiento, tratamiento y procesado de datos masivos a fin de visualizar y configurar datos relativos a sensores de la Fábrica del futuro, para ello se han implementado los siguientes componentes previamente definidos:

- **Infraestructura Big data:** Proporciona la capacidad para el almacenamiento masivo de datos y su procesado distribuido así como la capacidad de consulta e integración de los mismos con el resto de componentes del sistema.
- **Motor de correlación de datos:** Proporciona las capacidades de análisis de los datos de sensores almacenados a fin de generar predicciones y derivar parámetros de configuración que den como resultado optimizaciones en la fabricación.
- **Visualización de datos:** Permite la visualización de los datos de los sensores sin procesar, así como la configuración de parámetros de optimización derivados del análisis masivo de los datos ingerido.



Figura 1: Arquitectura del Sistema de Optimización

## Infraestructura Big Data

A fin de ejecutar el proyecto se ha proporcionado una infraestructura Big Data con el objetivo de procesar y almacenar los datos de sensorización capturados, esta infraestructura consta de dos componentes:

- **Clúster de cómputo distribuido:** Consistente en un clúster virtualizado en OpenStack con siete nodos de cómputo y una capa de servicios gestionados mediante una distribución Cloudera.
- **Clúster de persistencia distribuida:** Consistente en una base de datos Cassandra desplegada en modo clúster con siete nodos sobre el conjunto de instancias virtuales proporcionadas por OpenStack.

### 3.1.1 Clúster de cómputo

Para la implementación del clúster de cómputo distribuido 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, así 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.

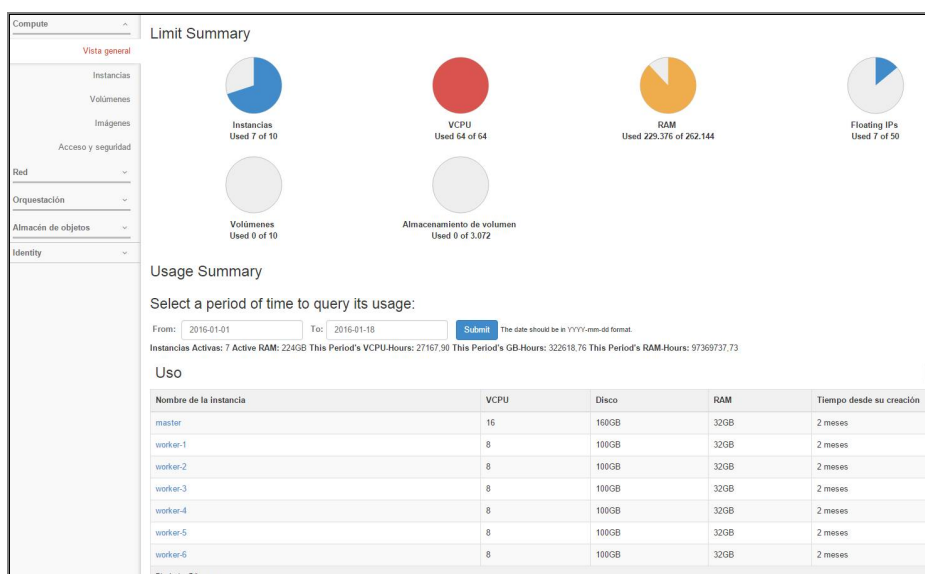


Figura 2: Listado de instancias en OpenStack

## 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.

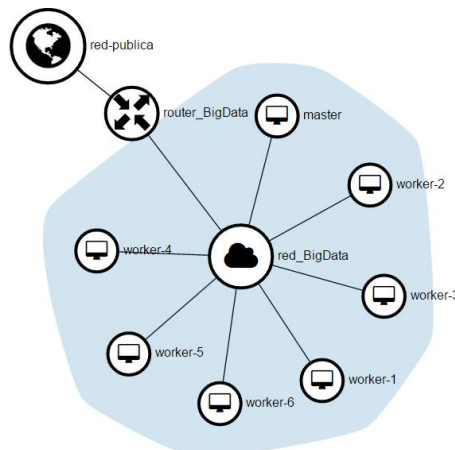


Figura 3: Topología de red en OpenStack

## 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 máquina ("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 cómputo, 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 4: Servicios instalados en Cloudera

La siguiente figura muestra el conjunto de nodos controlados por Cloudera, mostrando un resumen de sus características como nombre, IP, número de roles (por servicios), último latido, promedio de carga, uso de disco y uso de memoria RAM, el conjunto de recursos de cómputo sobrantes tras la instalación de los servicios quedan disponibles para tareas de cómputo distribuido, del mismo modo es posible la ampliación o decomisado de nodos para ampliar o reducir las capacidades de cómputo 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 5: Resumen de nodos instalados

### 3.1.1 Clúster de persistencia distribuida

Para la implementación del clúster de persistencia distribuida hemos optado por implantar una base de datos Cassandra distribuida sobre los siete nodos virtualizados disponibles en OpenStack.

La topología del clúster implementado se define en el fichero de propiedades `cassandra-topology.properties`, a tal efecto se ha configurado el clúster como un solo rack (conjunto de nodos) asociando las IPs públicas disponibles de los nodos del siguiente modo:

```
# Cassandra Node IP = Data Center : Rack
192.168.180.29=dc1: rack1
192.168.180.25=dc1: rack1
192.168.180.26=dc1: rack1
192.168.180.27=dc1: rack1
192.168.180.28=dc1: rack1
192.168.180.24=dc1: rack1
192.168.180.22=dc1: rack1
192.168.4.157=dc1: rack1
```

Adicionalmente a la configuración de la topología se han customizado los siguientes atributos del clúster a fin de habilitar el descubrimiento de los nodos del clúster y dar robustez especificando varios seeds (nodos iniciales a los que se conecta externamente).

```
seed_provider (class_name): org.apache.cassandra.locator.SimpleSeedProvider
parameters (seeds): "192.168.180.22, 192.168.180.24, 192.168.180.26"
```

En caso de caída de alguno de los nodos con rol “seed”, existirán otros dos que permitirían la conexión al clúster habilitando de este modo la HA (**Alta disponibilidad**).

Posteriormente se ha definido el esquema de almacenaje de los datos, en vocabulario de Cassandra “keyspace” (conjunto de tablas con ciertas características), estas características hacen referencia al factor de replicación de los datos en el clúster en nuestro caso tres, cada fragmento de información se divide en 255 fragmentos que se triplican y distribuyen entre los nodos del clúster.

```
CREATE KEYSPACE optiman
WITH durable_writes = true
AND replication = {
    'class' : 'SimpleStrategy',
    'replication_factor' : 3
};
```

De este modo conseguimos **tolerancia a fallos**, en caso de que algún fragmento se corrompa por la razón que sea, existen dos copias no corruptas del mismo y gracias a Cassandra y sus mecanismos internos, de forma **transparente** al administrador, ese fragmento es restaurado garantizando de este modo la calidad de los datos. Este parámetro puede ser aumentado o disminuido según las necesidades.

Para almacenar la información se han definido dos **tablas** una para las **medidas** y otra para las **predicciones**, la tabla medidas se particiona según datasource, sensor y se agrupa temporalmente según (timestamp) de forma descendente, por otro lado la tabla predicciones se particiona según datasource, sensor, channelid y al igual que medidas se agrupa temporalmente según (timestamp) descendientemente.

El esquema general de las tablas sería el siguiente:

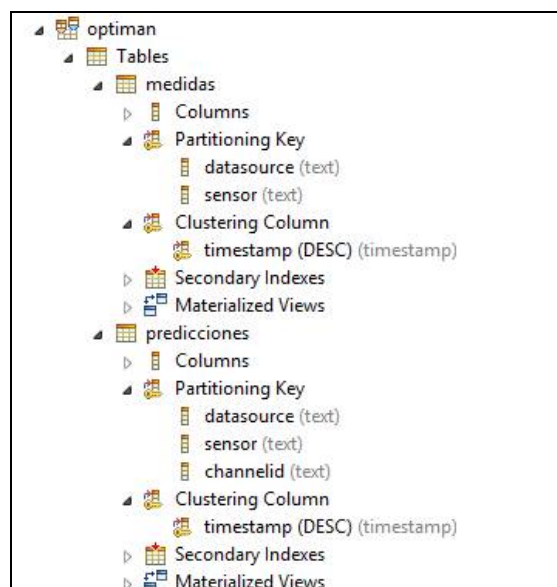


Figura 6: Esquema de tablas

La definición de la tabla de medidas sería:

```
CREATE TABLE IF NOT EXISTS optiman.medidas(
  datasource text, sensor text, timestamp timestamp, SZAPx float,
  SZx float, Trigge float, APHu float, APVs float,
  ACPv float, ZSx float, ZUs float, H7x float,
  ZDx float, H1x float, Pakt1 float, ZDvo float,
  Pakt7 float, H8x float, Pakt8 float, H9x float,
  Pakt9 float, H10x float, Pakt10 float, H2x float,
  SZGx float, ASZ float, SZAEEx float, StZAEEx float,
  StZAPx float, Pakt2 float, H6x float, Pakt6 float,
  H3x float, Pakt3 float, H4x float, Pakt4 float,
  H5x float, Pakt5 float, ZAusw float, ACPn float,
  FZx float, iwdIs float, DZs float, ZNach float,
  EA float, Mm float, ACPx float, APSS float,
  Ms float, APNs float, iwzausblx float, ASSx float,
  AC3u float, Svo float, IDKi1 Mic float, IDKi2 Mic float,
  PPMinx float, SKg float, PPx float, SFs float,
  SKs float, ZFx float, ASU float, FCycAct float,
  Z4x float, ZDry float, ZEvV float, ZEoef float,
  Z0eff float, ZEsch float, ZSchl float, ZQIx float,
  ZQDx float, ZMSx float, ZKern float, ZEve float,
  ZSKab float, ZSKa float, ZUaxs float, ZEx float, StZx float,
  PRIMARY KEY ((datasource, sensor), timestamp)) WITH CLUSTERING ORDER BY (timestamp DESC)
AND bloom_filter_fp_chance = 0.01
AND comment = ''
AND crc_check_chance = 1.0
AND dclocal_read_repair_chance = 0.1
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.0
AND speculative_retry = '99.OPERCENTILE'
AND caching = {
  'keys' : 'ALL',
  'rows_per_partition' : 'NONE'
}
AND compression = {
  'chunk_length_in_kb' : 64,
  'class' : 'LZ4Compressor',
  'enabled' : true
}
AND compaction = {
  'class' : 'SizeTieredCompactionStrategy',
  'max_threshold' : 32,
  'min_threshold' : 4
};
```

Y la definición de la tabla de predicciones sería:

```
CREATE TABLE optiman.predicciones(
  datasource text, sensor text, channelid text,
  timestamp timestamp, error1 float, error2 float,
  error3 float, id_modelo text, linea text,
  mensaje text, planta text, valor1 float,
  valor2 float, valor3 float,
  PRIMARY KEY (( datasource, sensor, channelid ), timestamp)
) WITH CLUSTERING ORDER BY ( timestamp DESC )
AND bloom_filter_fp_chance = 0.01
AND comment = ''
AND crc_check_chance = 1.0
AND dclocal_read_repair_chance = 0.1
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
```

```

AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.0
AND speculative_retry = '99.0PERCENTILE'
AND caching = {
  'keys' : 'ALL',
  'rows_per_partition' : 'NONE'
}
AND compression = {
  'chunk_length_in_kb' : 64,
  'class' : 'LZ4Compressor',
  'enabled' : true
}
AND compaction = {
  'class' : 'SizeTieredCompactionStrategy',
  'max_threshold' : 32,
  'min_threshold' : 4
};

```

Tras la definición, implementación y carga de los datos en el clúster descrito, podemos observar el estado del keyspace optiman en el clúster, donde cada nodo se hace responsable del 42% de los datos gracias a la replicación definida.

```

nodetool status optiman
Datacenter: dc1
=====

```

	Address	Load	Tokens	Owns	Host	ID
Rack						
UN	10.5.0.16	1.46 GB 256 41.5%	963abc48-8789-44a0-9846-94deb81afb25	rack1		
UN	10.5.0.13	1.5 GB 256 44.6%	f0728647-eb07-4df6-83d5-ddcab2360020	rack1		
UN	10.5.0.12	3.26 GB 256 42.8%	1d10f59e-1063-4b04-8a00-171aed6d2365	rack1		
UN	10.5.0.15	2.38 GB 256 41.4%	322bb2a7-6bce-41d6-80bf-130df909cadd	rack1		
UN	10.5.0.14	4.2 GB 256 44.5%	407dec67-f544-456b-a5f8-8adae973edf6	rack1		
UN	10.5.0.11	2.82 GB 256 43.9%	4022e04f-69dc-4355-9875-d661ad1e2fe4	rack1		
UN	10.5.0.10	902.5 MB 256 41.2%	4568936f-77fd-4cd5-9f38-b31c997445fd	rack1		

## 3.2 Motor de Correlación de Datos

El motor de correlación de datos se encarga de dar solución e implementación de cuatro objetivos

- **Detección de anomalías** en los valores de los sensores y generación de alertas.
- **Predicción** de los valores futuros en los sensores.
- **Optimización** de parámetros y objetivos de fabricación a fin de ser simulados.

A tal fin se han implementado técnicas estadísticas alimentadas con el histórico de los datos de los sensores que se detallan en los siguientes puntos:

### 3.2.1 Detección de Anomalías

El objetivo final consiste en implementar un sistema estadístico que permitan detectar comportamientos anómalos en las mediciones de los diferentes sensores para poder actuar a tiempo y evitar así posibles anomalías futuras, basándose para ello en conceptos objetivos y medibles, bien establecidos en el campo de la estadística.

Para acometer este objetivo se van a implementar técnicas de control unidimensional y multidimensional, que dan respuesta a nivel estadístico, sobre cambios de funcionamiento significativos en las mediciones.

En términos generales, un control estadístico se basa en técnicas de inferencia estadística y contraste de hipótesis que confirman o refutan una hipótesis inicial ( $H_0$ ) que considera que todo está bajo control y funcionando correctamente.

Para ello, se toma como referencia una distribución de probabilidad ( $D$ ), que depende del estadístico  $X$  que se va a evaluar y se le pregunta sobre las probabilidad que hay de que asumiendo que el sensor o sensores y dadas las evidencias actuales que se acaban de medir se esté funcionando de la manera esperada.

### 3.2.1.1 SPC Unidimensional

Una variable estadística puede ser sintetizada mediante características de posición que están relacionados con los niveles de funcionamiento, características de dispersión que dan cuenta del grado de variabilidad existente en dicha variable y características de forma que expresan numéricamente el grado de apuntamiento y simetría de las distribuciones de dichas variables.

Los SPC unidimensionales se utilizan, sobre todo, a la hora de predecir incrementos o decrementos de posición significativos o incrementos y decrementos de la variabilidad en el comportamiento de una única señal procedente de una medición.

A modo de ejemplo se muestran dos gráficos SPC-unidimensional, que asumen un modelo  $D$  normal, a partir del cual se establecen los límites de control que delimitarán la zona de alertas. El gráfico superior representa a la señal real, por contra el gráfico inferior nos muestra la media muestral de la señal real. Se puede apreciar como ofrece un número menor de falsos positivos en las alertas pues su variable se crea a partir de un cúmulo de evidencias de la propia señal durante el intervalo en estudio.

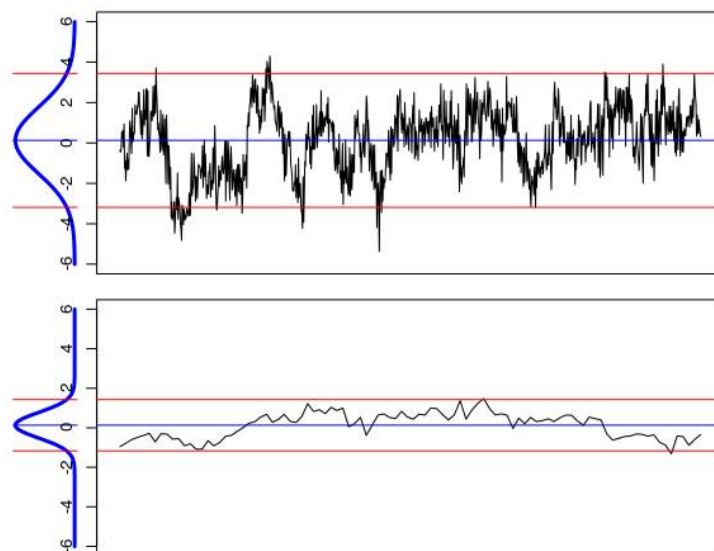


Figura 7: Gráfico de control clásico y basado en las desviaciones

Para la implementación del modelo se hace uso de la librería stats de scikit learn y se detalla el código principal del programa generado.

## 1. Lectura de parámetros de configuración

```
parametros="parametros.cnf"  
if (len(sys.argv) > 1):  
    parametros=sys.argv[1]  
  
lAux=sys.argv[0].split(os.path.sep)  
nombre=lAux[len(lAux)-1].split(".")[0]
```

## 2. Se crean los directorios que almacenaran los modelos a ser entrenados

```
vd=Trai nSPC()  
vd.configureParameters(parametros)  
vd.createDirectorios(vd.db.dirLogs)  
vd.createDirectorios(vd.db.dirModelos)  
vd.createDirectorios(vd.db.dirReports)  
  
vd.logger=vd.create_timed_rotating_log(vd.db.dirLogs+os.path.sep+nombre+".log")  
  
vd.logger.info("Opening Connection.")  
vd.createCon()  
  
vd.logger.info("Reading models (estimation)")  
lModelos=vd.l eeModelos(vd.db.dirModelos)
```

## 3. Se estiman los modelos unidimensionales a partir del histórico de cada sensor configurado

```
if (vd.db.recomputeModels==1 or len(lModelos)==0):  
    lSensores=vd.db.sensores  
  
    for sensor in lSensores:  
        vd.logger.info("Computing SPC models of station " + sensor)  
        vd.obtenModelos(sensor)
```

## 4. Se generan predicciones (alertas) a partir de los datos del sensor actual y de los modelos entrenados

```
vd.logger.info("Reading models (prediction)")  
lModelos=vd.l eeModelos(vd.db.dirModelos)  
vd.logger.info("Making predictions")  
vd.predice(vd.db.dirModelos,lModelos)  
vd.logger.info("Closing connection (estimation) .")  
vd.closeCon()  
  
# Like a service  
tEspera=vd.db.timeSleep
```

## 5. Bucle de generación de alertas

```
while (tEspera>0):  
    vd.logger.info("Sleeping " + str(tEspera) + " seg.")  
    time.sleep(tEspera)  
  
    #print "MODELO SPC BUSCANDO ALERTAS"  
    vd.logger.info("Opening connection (prediction)")  
    vd.createCon()  
    vd.logger.info("Reading models (prediction)")  
    lModelos=vd.l eeModelos(vd.db.dirModelos)
```

```
# MODELO PREDICIENDO
vd.logger.info("Making predictions")
vd.preds(vd.db.dirModelos, lModelos)
vd.logger.info("Closing connection (prediction)")
vd.closeCon()

vd.destroy()
```

### 3.2.1.2 SPC Multidimensional

En ocasiones existen alertas que no son apreciables desde un punto de vista unidimensional, pues se producen a nivel multidimensional cuando los valores de un sensor, dejan de serlo cuando se considera en relación con los valores del resto de sensores obtenidos durante ese periodo. Esto se produce, bien porque se rompe la estructura de relación existente entre el conjunto de variables o bien porque el punto en el espacio multidimensional está muy separado del resto de puntos.

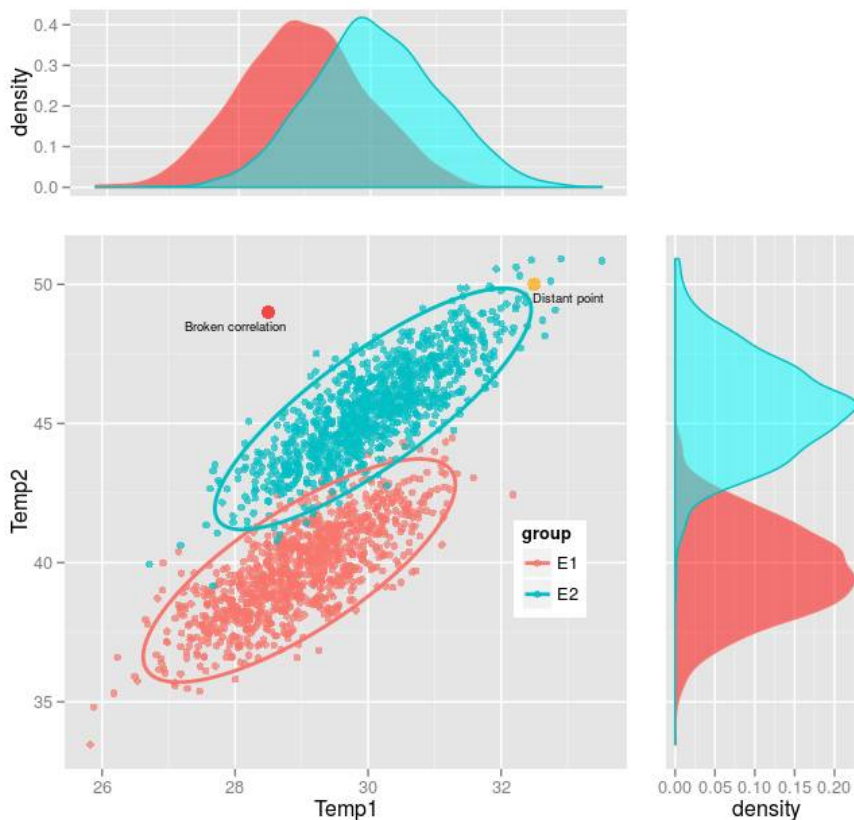


Figura 8: Gráfico de control clásico y basado en las desviaciones

La figura muestra gráficamente la problemática planteada arriba. En dicha figura se puede ver el comportamiento de dos sensores de temperatura (Temp1 y Temp2) en dos estados de comportamiento diferentes (E1 y E2). En la gráfica se señalan dos puntos (rojo, naranja) que muestran datos anómalos, no detectables desde un plano unidimensional, que resaltan a nivel multidimensional.

Para la implementación se puede hacer uso de  $T^2$  de Hotelling como estadístico que ofrece una única medida unidimensional representativa de los valores multidimensionales. En el cálculo del modelo se hace uso de la librería stats de scikit learn y se detalla el código principal del programa generado.

La siguiente figura nos muestra un gráfico de control multidimensional, de los parámetros de fabricación del plástico. Se puede apreciar la rotura de correlación entre variables al principio y final de los datos, debido quizás a la fase previa al régimen estacionario de fabricación y a posibles anomalías al final junto con señales bajo control durante la fase estacionaria.

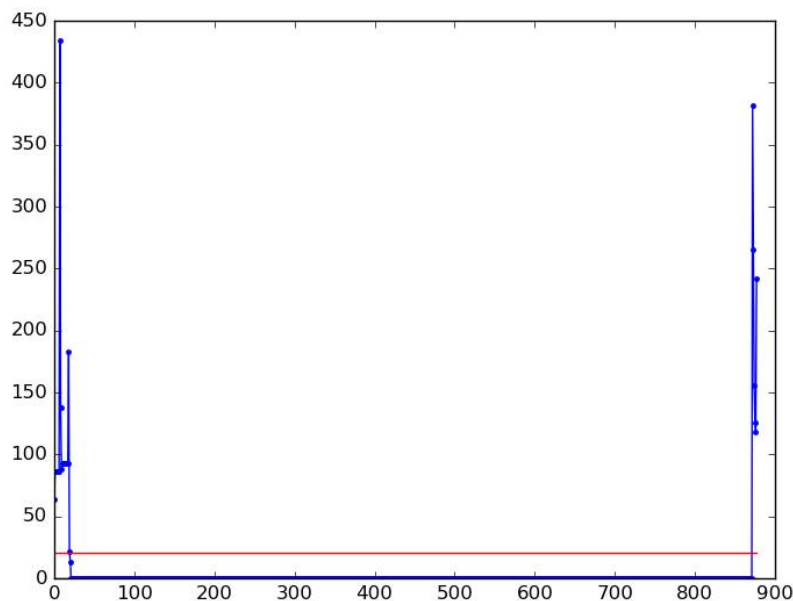


Figura 9: Gráfico T2 de sensores inyectora de plástico

Para la implementación del modelo se hace uso de la librería stats de scikit learn y se detalla el código principal del programa generado.

## 1. Lectura de parámetros de configuración

```
parametros="parametros.cnf"
if (len(sys.argv) > 1):
    parametros=sys.argv[1]
```

```
IAux=sys.argv[0].split(os.path.sep)
nombre=IAux[len(IAux)-1].split(".")[0]
```

*Se crean los directorios que almacenaran los modelos a ser entrenados*

```
vd=SpcMultidimensionalSensores()
vd.configureParameters(parametros)
vd.createDirectorios(vd.db.dirLogs)
vd.createDirectorios(vd.db.dirModelos)
vd.createDirectorios(vd.db.dirReports)
```

```
vd.logger=vd.create_timed_rotating_log(vd.db.dirLogs+os.path.sep+nombre+".log")
```

```
vd.logger.info("Opening Connection.")
vd.createCon()
```

```
# SET USING CONSISTENCY QUORUM TO CASSANDRA IF THERE IS MORE THAN A NODE IN THE CLUSTER.
#lCon=vd.db.server.split("|")
#if (len(lCon)>1):
#   vd.logger.info("Setting: USING CONSISTENCY QUORUM")
#   instruction="USING CONSISTENCY QUORUM;"
#   vd.db.con.execute(instruction)

vd.logger.info("Reading models (estimation)")
lModelos=vd.lleeModelos(vd.db.dirModelos)
```

## 2. Se estiman los modelos multidimensionales a partir del histórico de los sensores configurados

```
if (vd.db.recomputeModels==1 or len(lModelos)==0):
    lSensores=vd.db.sensores

    for sensor in lSensores:
        vd.logger.info("Computing models for station " + sensor)
        vd.obtenModelos(sensor)
```

## 3. Se generan predicciones (alertas) a partir de los datos de los sensores actuales y de los modelos entrenados previamente

```
vd.logger.info("Reading models (prediction)")
lModelos=vd.lleeModelos(vd.db.dirModelos)
vd.logger.info("Making predictions")
vd.predice(vd.db.dirModelos, lModelos)
vd.logger.info("Closing connection (estimation) .")
vd.closeCon()

# Like a service
tEspera=vd.db.timeSleep
```

## 4. Bucle de generación de alertas

```
while (tEspera>0):
    vd.logger.info("Sleeping " + str(tEspera) + " seg.")
    time.sleep(tEspera)

    vd.logger.info("Opening connection (prediction)")
    vd.createCon()
    vd.logger.info("Reading models (prediction)")
    lModelos=vd.lleeModelos(vd.db.dirModelos)

    vd.logger.info("Making predictions")
    vd.predice(vd.db.dirModelos, lModelos)
    vd.logger.info("Closing connection (prediction)")
    vd.closeCon()

vd.destroy()
```

### 3.2.2 Predicción de valores futuros.

Uno de los objetivos en el modelado estadístico de series temporales, consiste en la obtención de modelos que permitan predecir los valores de determinados sensores en el presente o en el futuro próximo, con objeto de conocer los valores siguientes, para por ejemplo tomar las acciones correctivas oportunas previamente antes de que algo ocurra.

Los modelos de series temporales puros, se basan en la idea de que los factores influyentes sobre una variable están contenidos en los valores que dicha variable tiene en instantes pasados, es por ello que utilizan dichos valores pasados para predecir los valores futuros de la propia variable. De manera muy general, se puede ver como un modelo de regresión cuyas variables explicativas son determinados instantes pasados de la propia variable a predecir.

El número y concreción de instantes pasados a utilizar se realiza tomando como objetivo la obtención de un modelo final que sea estacionario, o lo que es lo mismo, que los residuos del modelo final no estén correlacionados con residuos en instantes anteriores (autocorrelación nula), que los residuos tengan varianza constante (homocedasticidad) y que estén distribuidos normalmente (distribución gaussiana).

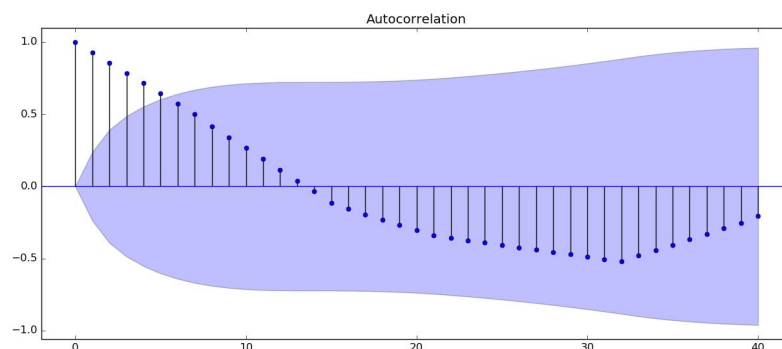


Figura 10: Gráfico de autocorrelación de la variable H7x de la inyectora de plástico ENGEL

La figura nos muestra las autocorrelaciones de la variable H7x del mecanizado de AIMPLAS antes de crearse un modelo de para ella. A la vista del gráfico se puede observar que un modelo de series temporales podría ser candidato a modelar dicha señal pues se observa una clara relación de instantes anteriores sobre el instante actual.

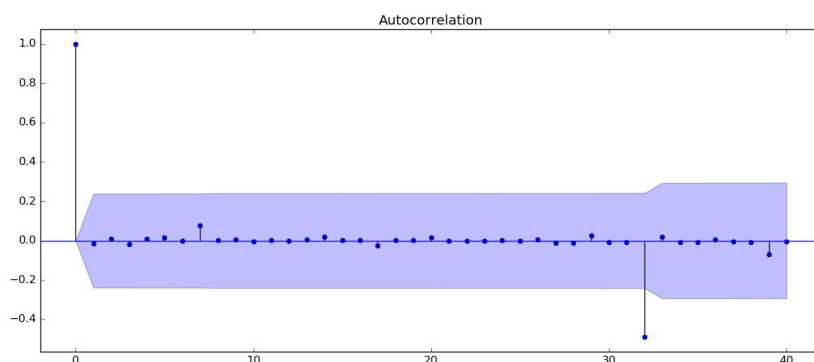


Figura 11: Gráfico de autocorrelación de los residuos del modelo de series temporales estimado para la variable H7x de la inyectora de plástico ENGEL

La anterior gráfica nos muestra el gráfico de autocorrelación de los residuos producidos por el modelo de la señal H7x. Este gráfico nos indica una dependencia nula con los instantes más recientes en el pasado lo que nos valida el modelo respecto a la parte correspondiente de autocorrelación para conseguir el objetivo final de estacionariedad.

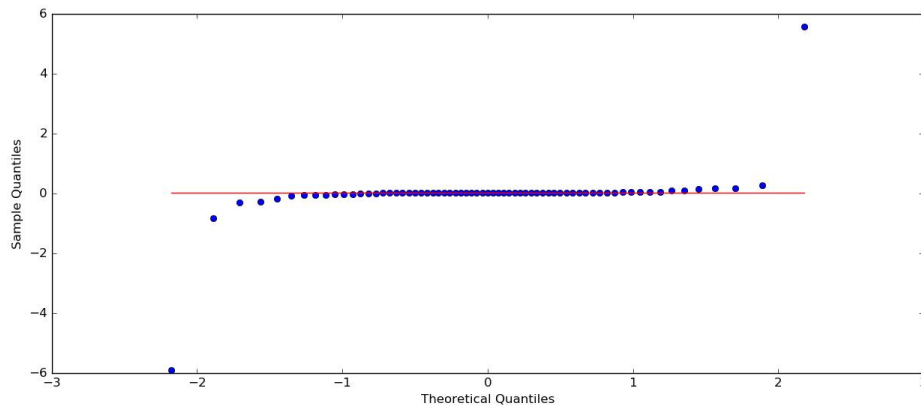


Figura 12: Gráfico de cuantiles de los residuos de la variable H7x de la inyectora de plástico ENGEL

La anterior imagen nos muestra los cuantiles de los residuos. En dicha gráfica se puede ver que a excepción de 2 residuos anómalos el resto de residuos tienen un valor cercano a cero indicativo de un bajo error en el modelo final y además la pendiente de la distribución de estos podría considerarse nula.

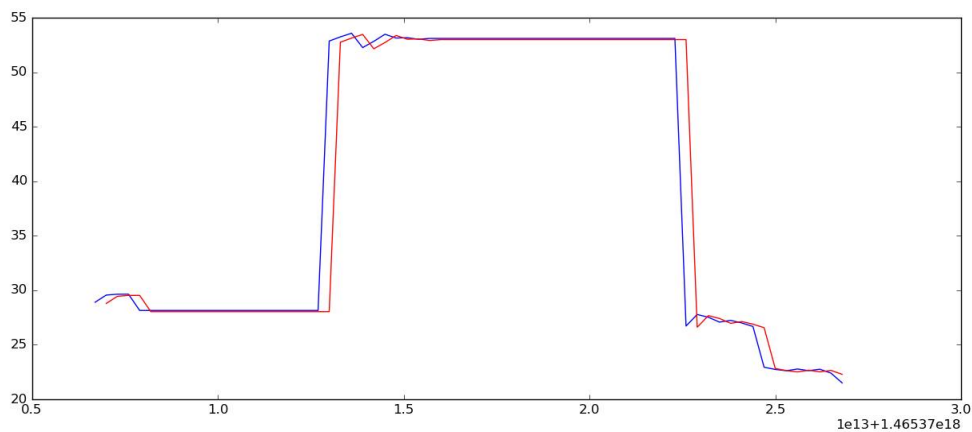


Figura 13: Modelo predictor de la variable H7x de la inyectora de plástico ENGEL

La anterior imagen nos muestra los datos reales (azul) y predichos (rojo), de la señal H7x, mediante un modelo basado en series temporales.

### 3.2.3 Optimización de parámetros y objetivos de fabricación

A fin de detectar los parámetros a configurar en relación a las variables objetivo a optimizar se ha realizado un análisis de los datos mediante el cálculo de una matriz de confusión, de las 78 características proporcionadas en los datos donde se detectan como están correlacionados entre sí.

Para el caso de uso de fabricación de hormas de calzado, se observan los datos espectrales de dos sensores de vibración, situados en diferentes puntos de la máquina, bajo tres situaciones diferentes, la primera nos muestra datos cuando la máquina ha hecho un ciclo en vacío sin fabricación de pieza alguna, la segunda nos muestra una situación en el que la pieza obtenida ha sido correcta desde el punto de vista de la calidad, mientras que la última nos muestra los datos espectrales en los que la pieza fabricada no tenía la calidad final necesaria.

A la vista de las imágenes de espectro y energía en los diferentes puntos, se puede observar como aparecen diferentes frecuencias para los tres tipos de producción, vacío, calidad y no calidad comentadas anteriormente. Este estudio previo, nos ofrece evidencias sobre la importancia de la medición de las vibraciones a la hora de extraer características que nos permitan crear modelos que den cuenta de los parámetros de calidad según los parámetros de calidad previamente establecidos.

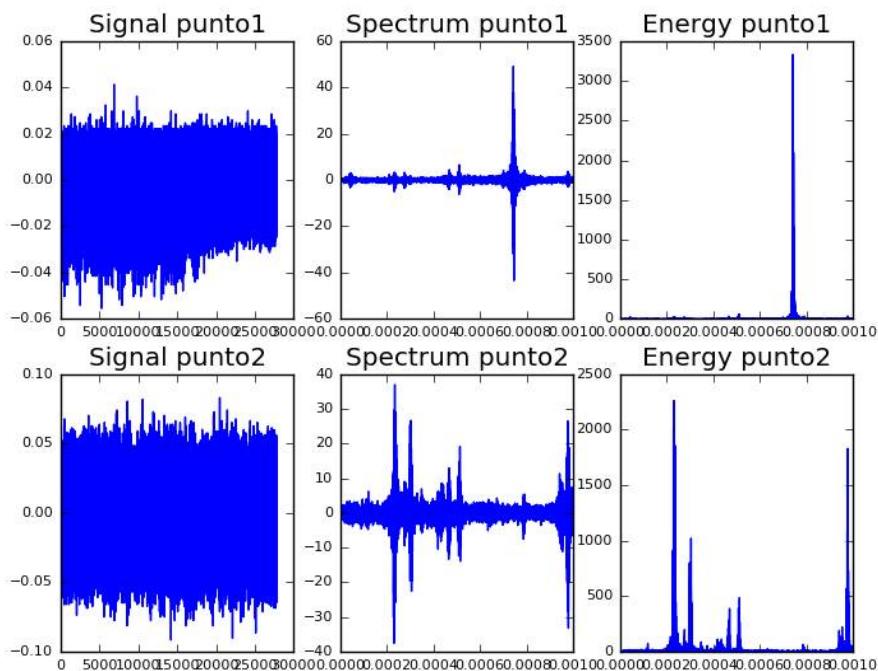


Figura 14: Espectro de la señal del sensor de vibración en vacío

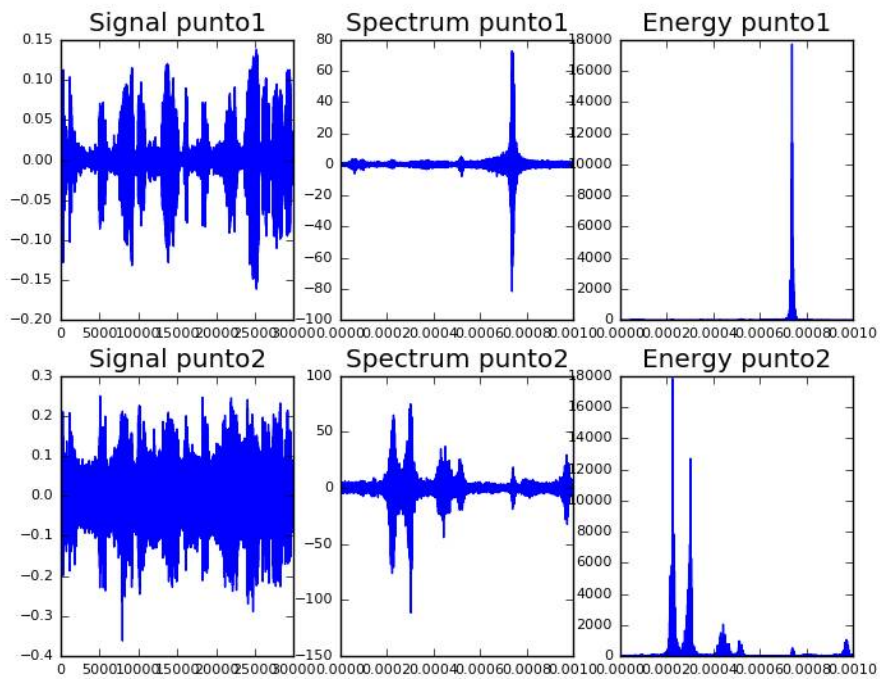


Figura 15: Espectro de la señal del sensor de vibración con funcionamiento válido

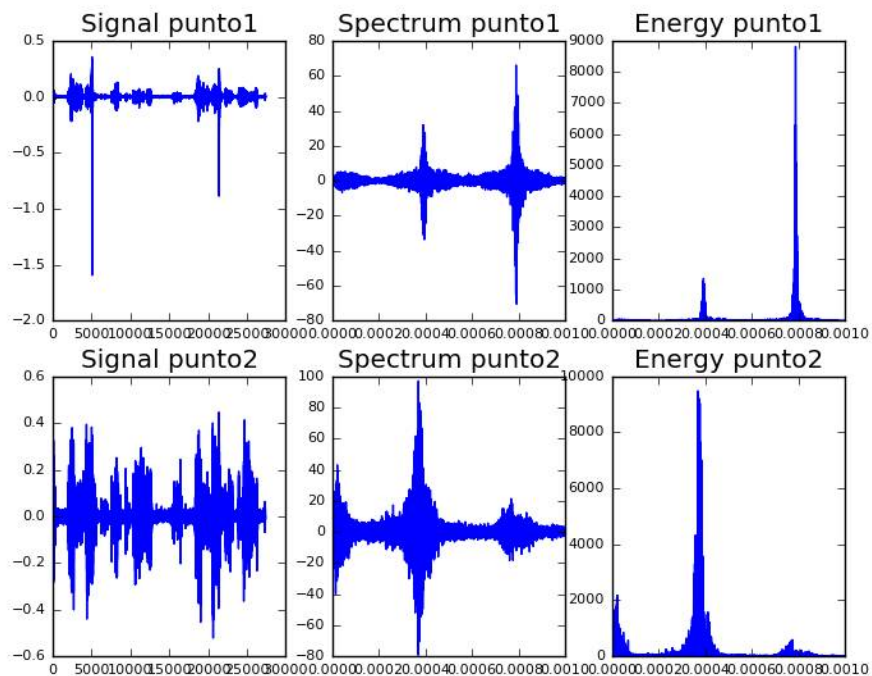


Figura 16: Espectro de la señal del sensor de vibración con funcionamiento incorrecto

En todo proceso de fabricación, puede ser bastante útil disponer de una herramienta que permita simular los valores de ciertas variables objetivo de interés, que habitualmente nos informarán sobre determinados valores de calidad del objeto producido, en función de los valores impuestos a ciertos parámetros de funcionamiento durante la fabricación. Para ello es necesario establecer modelos que relacionen dichas variables objetivo con los parámetros de fabricación. Dichos modelos toman la forma de una función,  $y=f(x)$ , donde una variable objetivo,  $y$ , se relaciona con los parámetros  $x$  a través de la función  $f$ . Dicha función  $f$ , no es más que un modelo que haber sido aprendida a través de un modelo de regresión, red neuronal, máquinas de soporte vectorial o cualquier algoritmo de Machine Learning conocido. También puede ser interesante establecer funciones en el sentido inverso,  $x=g(y)$ , que nos permitirán conocer los valores que se han de imponer a los parámetros controlables de la máquina que fabrica la pieza para obtener los objetivos de calidad requeridos.

En el caso que nos ocupa se ha hecho uso de diferentes modelos de regresión para el aprendizaje de la función  $f$  y  $g$ . A continuación mostramos los diferentes modelos entrenados y sus interdependencias así como su varianza y error, que permitirán averiguar el valor a imponer a los parámetros para unos objetivos (ZUs, APHu, ACPv, ZSx) dados (f) y viceversa (g).

Estos **parámetros** son:

- Valor punta presión (H1x)
- Volumen de inyección (H2x)
- Temperatura del aceite (H3x)
- Zona cilindro mecánica 1 (H4x)
- Zona cilindro mecánica 2 (H5x)
- Zona cilindro mecánica 3 (H6x)
- Traversa (H7x)
- Temperatura del molde (SVo)
- Valor punta presión específica inyección (APVs)

Las **variables objetivo** serían las siguientes:

- Presión de espesura al conmutar (ZUs)
- Cojín de material final postpresion (APHu)
- Valor real tiempo de inyección (ACPv)
- Tiempo de ciclo hasta el final del desmoldeo (ZSx)

A continuación se muestra el resultado de la matriz de correlación de las variables sensorizadas en la fabricación de plástico.

Dicha tabla nos es útil como método descriptivo a la hora de conocer que variables guardan relación entre sí. Esta información nos será útil a la hora de establecer las variables candidatas a formar parte de nuestro modelo. Así pues, mientras que la intensidad de color nos indica el grado de relación lineal entre las variables implicadas, el color nos indica el signo de dicha relación. Cabe destacar que la diagonal está a la máxima intensidad y con el color negro pues

nos muestra la relación de una variable consigo misma, que siempre será positiva y de intensidad máxima.

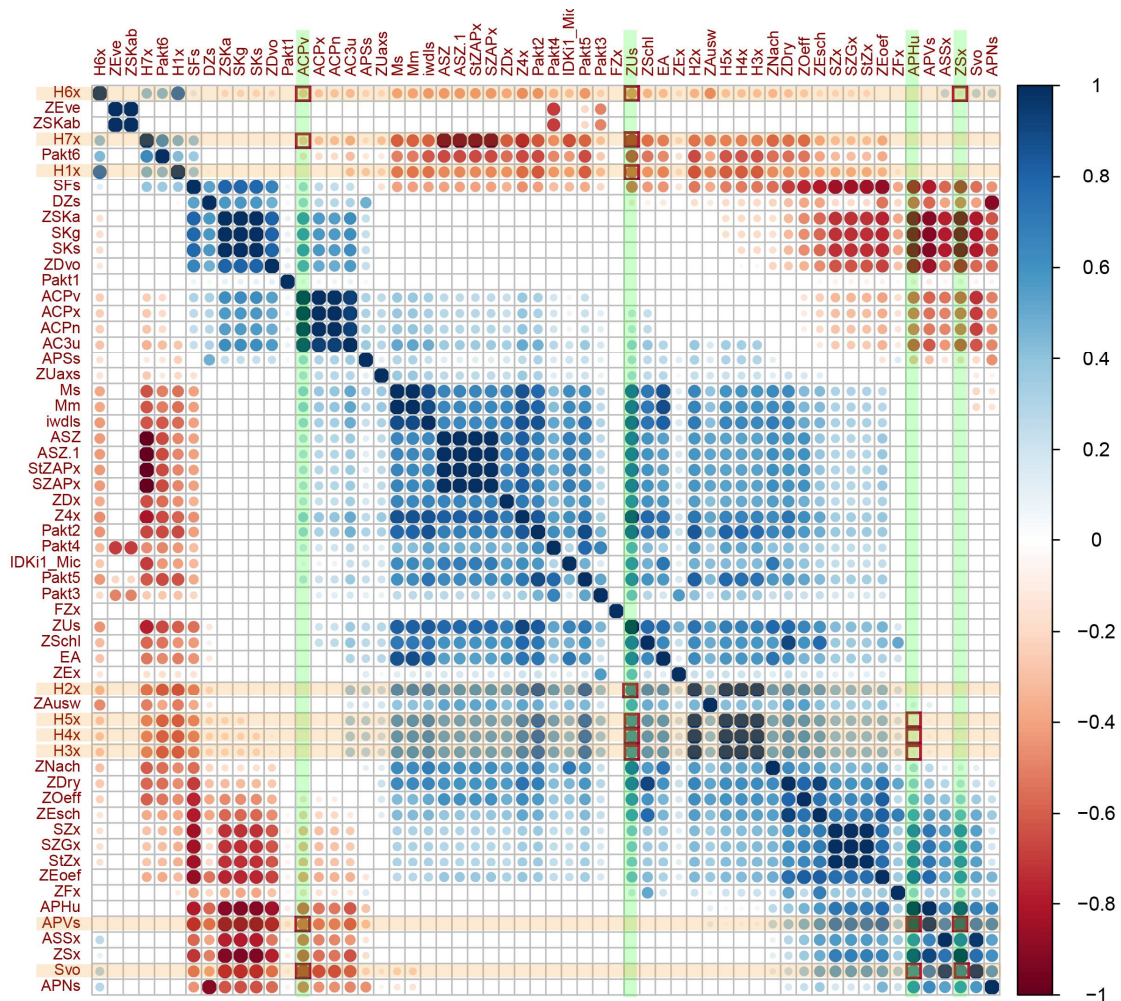


Figura 17: Matriz de correlación de las variables de la inyectora de plástico

### Los modelos de parámetros:

- $APVs = g(APHu, ACPv, ZSx)$   
 $STD\text{-variable: } 1086.036$   
 $STD\text{-error: } 27.40373$
- $SVo = g(ACPv, APHu, ZSx, APVs)$   
 $STD\text{-variable: } 9.552752$   
 $STD\text{-error: } 2.634062$
- $H2x = g(APHu, ZUs, ACPv, APVs)$   
 $STD\text{-variable: } 47.50874$

*STD-error: 21. 21798*

- $H3x = g(\text{APHu}, \text{ZUs}, \text{ACPv}, \text{APVs}, \text{H2x})$

*STD-variabl e: 47. 50874*

*STD-error: 21. 21798*

- $H4x = g(\text{APHu}, \text{ZUs}, \text{ACPv}, \text{APVs}, \text{H2x}, \text{H3x})$

*STD-variabl e: 45. 99958*

*STD-error: 1. 186222*

- $H5x = g(\text{APHu}, \text{ZUs}, \text{ACPv}, \text{APVs}, \text{H2x}, \text{H3x}, \text{H4x})$

*STD-variabl e: 42. 23718*

*STD-error: 0. 7415898*

- $H6x = g(\text{ACPv}, \text{ZUs}, \text{ZSx}, \text{H2x}, \text{H3x}, \text{H4x}, \text{H5x})$

*STD-variabl e: 4. 356975*

*STD-error: 1. 689341*

- $H7x = g(\text{ACPv}, \text{ZUs}, \text{H2x}, \text{H3x}, \text{H4x}, \text{H5x}, \text{H6x})$

*STD-variabl e: 520. 6997*

*STD-error: 242. 0908*

- $H1x = g(\text{ZUs}, \text{H2x}, \text{H3x}, \text{H4x}, \text{H5x}, \text{H6x}, \text{H7x})$

*STD-variabl e: 4. 33565*

*STD-error: 1. 049538*

#### **Los modelos de variables objetivo:**

- $\text{ZUs} = f(\text{H1x}, \text{H2x}, \text{H3x}, \text{H4x}, \text{H5x}, \text{H6x})$

*STD-variabl e: 1. 369222*

*STD-error: 0. 7302088*

- $\text{APHu} = f(\text{APVs}, \text{SVo}, \text{H4x}, \text{H5x})$

*STD-variabl e: 1087. 818*

*STD-error: 15. 57537*

- $\text{ACPv} = f(\text{APHu}, \text{SVo}, \text{APVs}, \text{H6x}, \text{H7x})$

*STD-variabl e: 2. 929129*

*STD-error: 0. 3637341*

- $\text{ZSx} = f(\text{ACPv}, \text{APHu}, \text{APVs}, \text{SVo}, \text{H6x})$

*STD-variabl e: 0. 7152787*

*STD-error: 0. 2037526*

### 3.3 Visualización de datos y optimización

El sistema de optimización dispone de un componente de visualización de datos y optimización consistente en una aplicación web en la que los usuarios podrán identificarse para acceder a cinco funcionalidades principales:

1. **Dashboard:** panel que recoge un conjunto de widgets que resumen el estado del mecanizado en materia de detección de anomalías e indicadores de optimización. Será posible llegar hasta el detalle de un sensor para ver las alertas asociadas y sus medidas.
2. **Alertas:** mostrará el histórico de alertas detectadas por el motor de correlación de datos y notificadas por el sistema experto. Permitirá la búsqueda por diferentes parámetros y acceder al detalle de la alerta para su gestión.
3. **Sensores:** permitirá acceder al conjunto de sensores que forman parte del mecanizado para graficar sus medidas a la par que un conjunto de indicadores de funcionamiento tales como el gráfico de control generado por el motor de correlación de datos.
4. **Simulación:** funcionalidad que permitirá simular los parámetros de entrada del mecanizado para optimizar su funcionamiento. Será posible realizar una función inversa para determinar qué parámetros debemos configurar para obtener unos resultados determinados.
5. **Configuración:** conjunto de funcionalidades que permitirán al usuario desde configurar el mecanizado, sus sensores, parámetros de funcionamiento de la herramienta de visualización, así como conocer el estado del sistema de sensorización utilizado.

A continuación se muestra el árbol de navegación que se ha implementado en la herramienta, así como una descripción detallada de cada una de las funcionalidades implementadas.

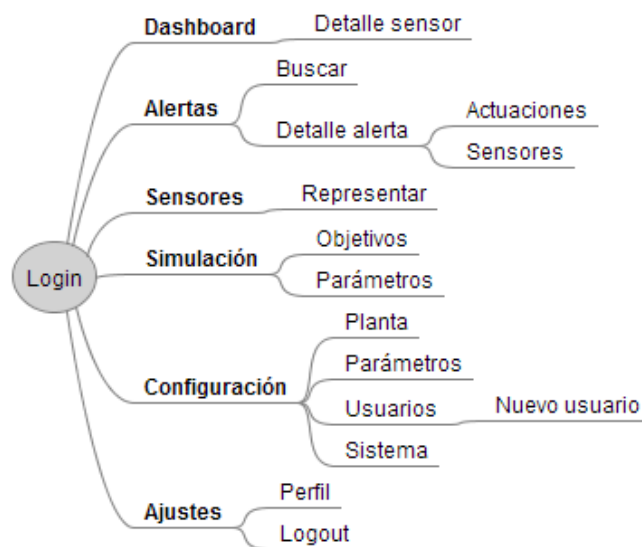
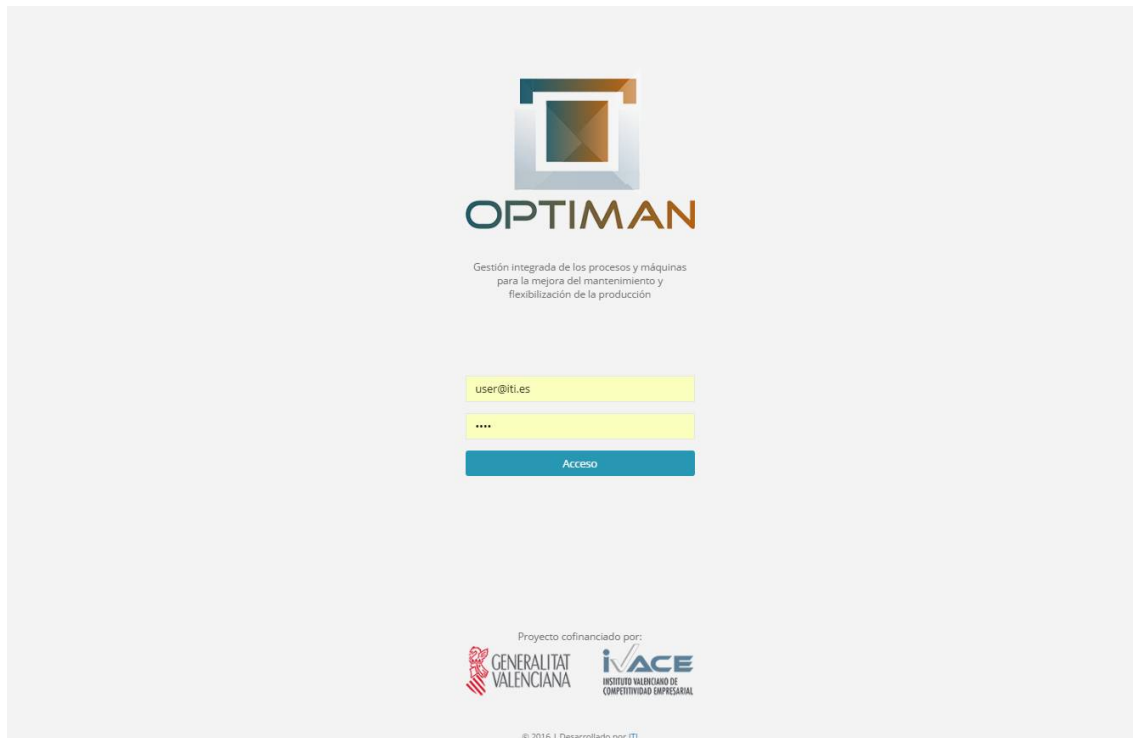


Figura 18: Árbol de navegación de la herramienta de visualización de datos

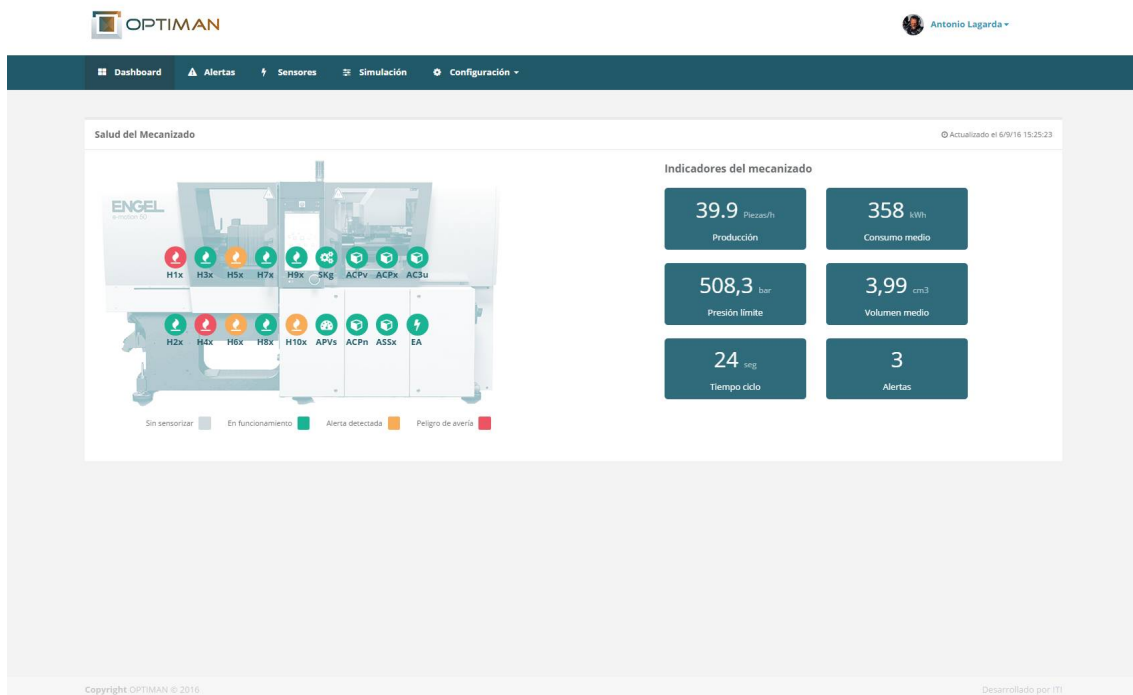
### 3.3.1 Pantalla de login



Esta pantalla permite el acceso al sistema de optimización, requiere la creación de una cuenta de usuario previa al acceso, esta cuenta constara de correo electrónico y una contraseña, la validación de la misma se realiza contra el backend REST implementado en Node.js.

Una vez validadas las credenciales del usuario se accederá a la pantalla del cuadro de mando del sistema, que permite a su vez del acceso al resto del sistema. Atendiendo al usuario, mostrará un mecanizado y otro (INESCOP o AIMPLAS).

### 3.3.2 Pantalla de cuadro de mando del sistema

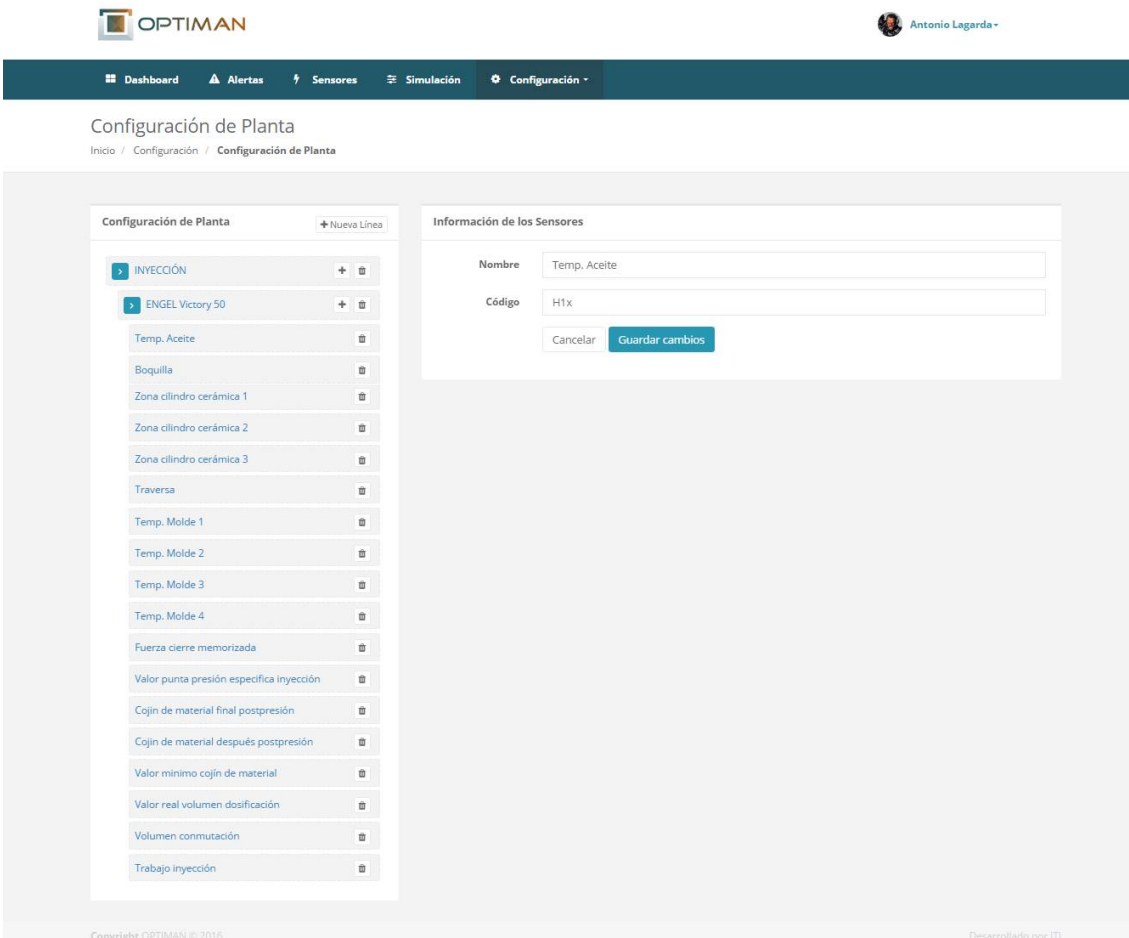


La pantalla de cuadro de mando del sistema de optimización muestra un resumen del estado del sistema, consta de dos componentes:

- **Grafico del sistema:** Muestra un resumen de estado de los distintos sensores instalados en el sistema de producción, asocia un icono a cada sensor y un código de colores en función del número y tipo de alertas detectadas por sensor. **Verde** indica que no existen alertas detectadas, **Amarillo** que hay alertas de baja criticidad y **Rojo** que existen alertas de criticidad alta
- **Cuadro de indicadores:** Muestra un conjunto de KPI's (indicadores) preestablecidos como número de piezas producidas por hora, consumo medio, presión límite, volumen medio, tiempo límite y numero de alertas detectadas.

En la barra superior se muestra el menú de acceso al resto de pantallas de la aplicación, Dashboard (Actual), Alertas, Sensores, Simulación y Configuración. Esta barra es común a toda la aplicación permitiendo la navegabilidad en cualquier momento al resto de pantallas del sistema.

### 3.3.3 Pantalla de configuración de planta

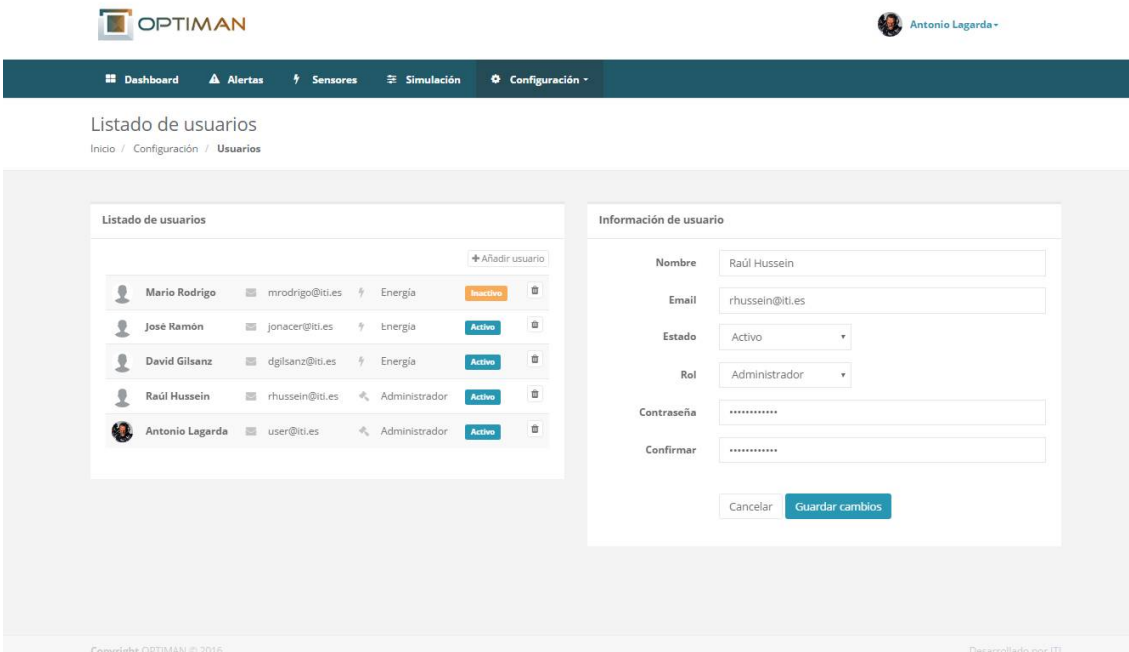


Copyright OPTIMAN © 2015 Desarrollado por ITI

Esta pantalla permite la configuración de los elementos de la planta, la planta cuenta con distintas líneas de producción que a su vez disponen de elementos de fabricación que constan de un conjunto de sensores, la pantalla permite la definición de nuevas líneas, elementos de fabricación, sensores o la edición de los existentes.

Todo elemento del sistema constara de una descripción que facilite su identificación para el usuario y un código único dentro del sistema, ambos editables desde esta pantalla.

### 3.3.4 Pantalla de configuración de usuarios

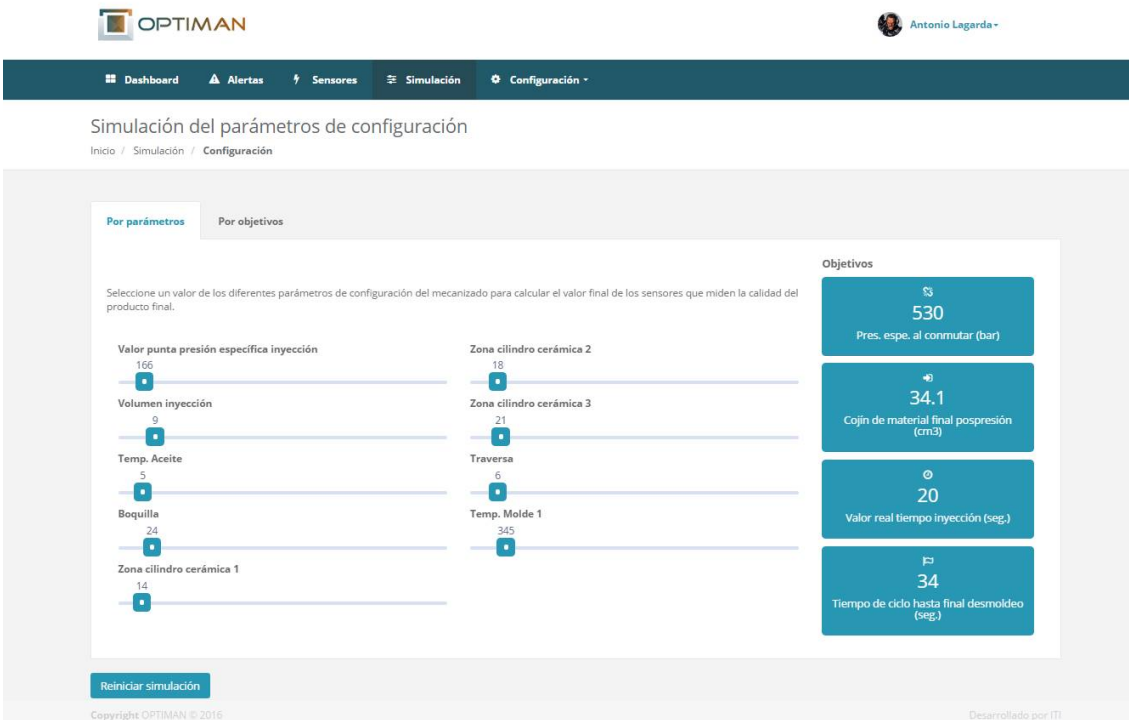


Copyright OPTIMAN © 2016 Desarrollado por ITI

Esta pantalla permite la configuración de los usuarios que podrán acceder al sistema, permite el alta, modificación y borrado de los mismos, se configurara por usuario los siguientes datos:

- Nombre
- Email (login del sistema)
- Estado (Activo/Desactivado)
- Contraseña (contraseña del sistema)
- Confirmación de contraseña

### 3.3.5 Pantalla de simulación de parámetros de configuración



Esta pantalla permite la simulación de parámetros de configuración de producción para un elemento de fabricación, a partir del modelo de simulación aprendido con el histórico de datos recogidos por el sistema, se permite mediante un conjunto de sliders modificar ciertos parámetros dependientes entre si y visualizar como afectan a un conjunto de variables objetivo dependientes de los mismos.

Esta simulación se puede realizar a la inversa, partiendo de un conjunto de variables objetivo introducidas por el usuario, el sistema simula cuales deberían ser los parámetros de configuración para la fabricación a fin de alcanzar los objetivos fijados.

Estos parámetros son:

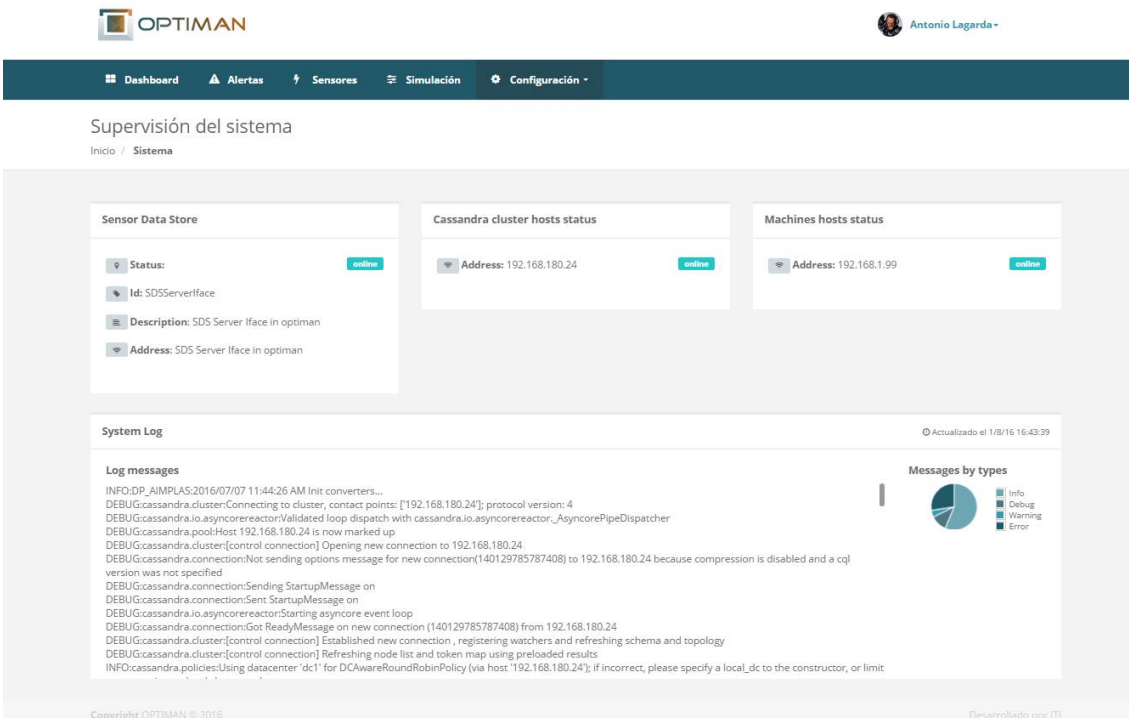
- Valor punta presión
- Volumen de inyección
- Temperatura del aceite
- Zona cilindro mecánica 1
- Zona cilindro mecánica 2
- Zona cilindro mecánica 3

- Traversa
- Temperatura del molde

Las variables objetivo serían las siguientes:

- Presión de espesura al conmutar
- Cojín de material final postpresion
- Valor real tiempo de inyección
- Tiempo de ciclo hasta el final del desmoldeo

### 3.3.6 Pantalla de supervisión del sistema

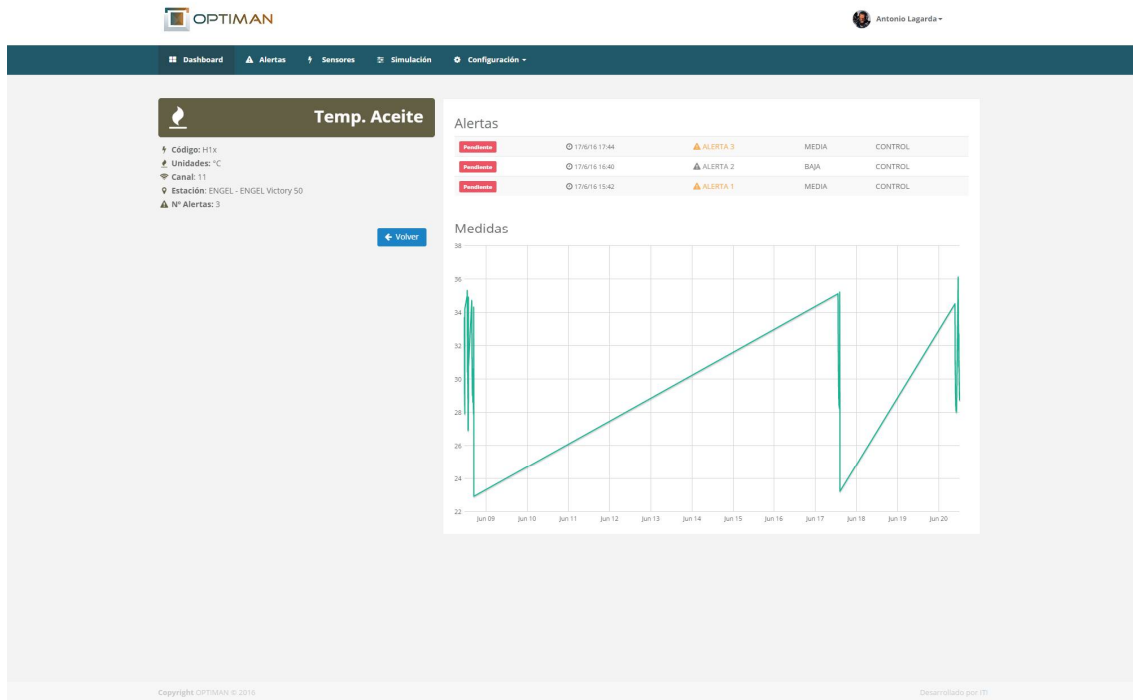


The screenshot displays the 'Supervisión del sistema' interface. At the top, there's a navigation bar with 'Dashboard', 'Alertas', 'Sensores', 'Simulación', and 'Configuración'. The main content area is titled 'Supervisión del sistema' and includes a breadcrumb 'Inicio / Sistema'. Three status cards are visible: 'Sensor Data Store' (Status: online, Id: SDSServerIface, Description: SDS Server Iface in optiman, Address: SDS Server Iface in optiman), 'Cassandra cluster hosts status' (Address: 192.168.180.24, Status: online), and 'Machines hosts status' (Address: 192.168.1.99, Status: online). Below these is a 'System Log' section with a 'Log messages' list and a 'Messages by types' pie chart. The pie chart shows counts for Info, Debug, Warning, and Error. The dashboard also includes a navigation menu with 'Dashboard', 'Alertas', 'Sensores', 'Simulación', and 'Configuración'.

Esta pantalla muestra el estado general del sistema desde el punto de vista de los elementos que lo componen, de este modo se visualiza el estado (Activo/No activo) del sistema de recolección de datos (Sensor data store), sistema de persistencia distribuido (clúster de Cassandra) y host del sistema (máquina que lo aloja).

Adicionalmente se muestran los logs del sistema, mostrando los últimos eventos producidos así como un pequeño indicador en forma de pie chart que muestra el conteo por tipo de log (Info, Warning, Error, etc.)

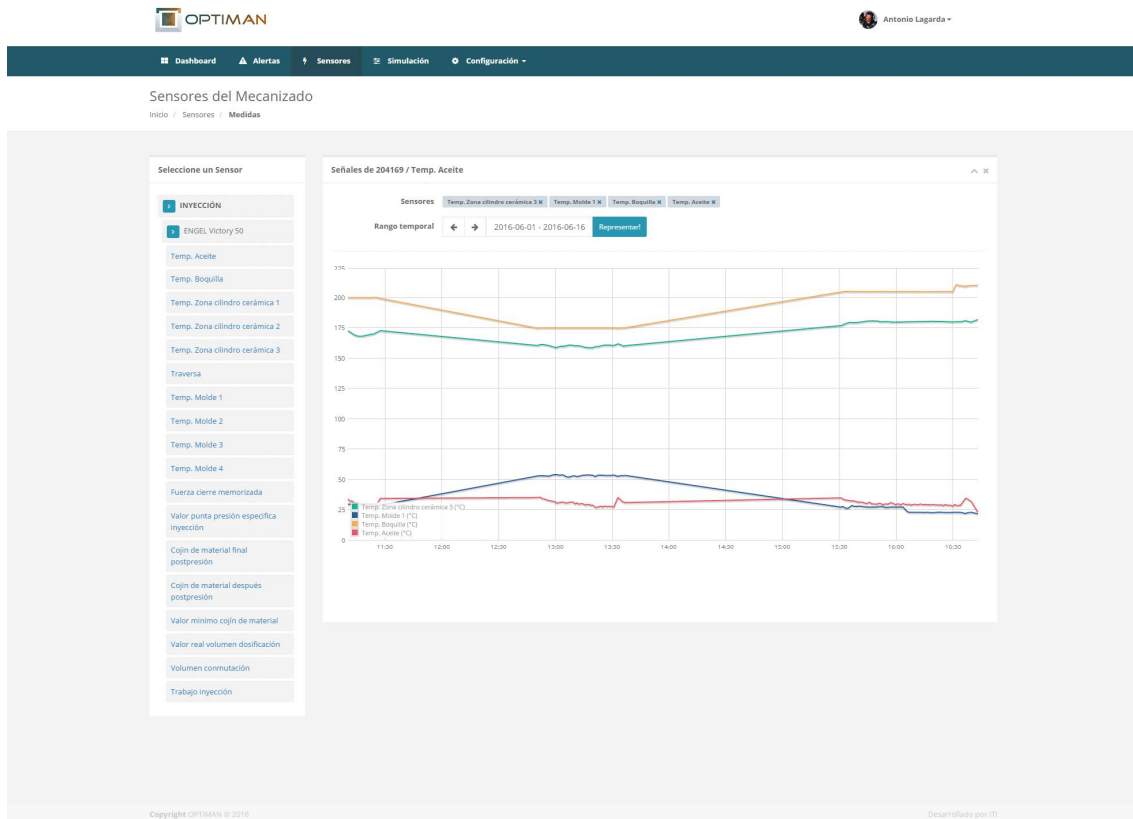
### 3.3.7 Pantalla de visualización de alertas



Esta pantalla muestra el listado de alertas asociadas a un sensor, para cada alerta se muestra un resumen de la misma indicando el estado de la misma, fecha de alta en el sistema, tipo de alerta, criticidad y descripción de la misma.

A su vez se muestran las últimas medidas del sensor, a fin de facilitar la interpretación de la alerta detectada.

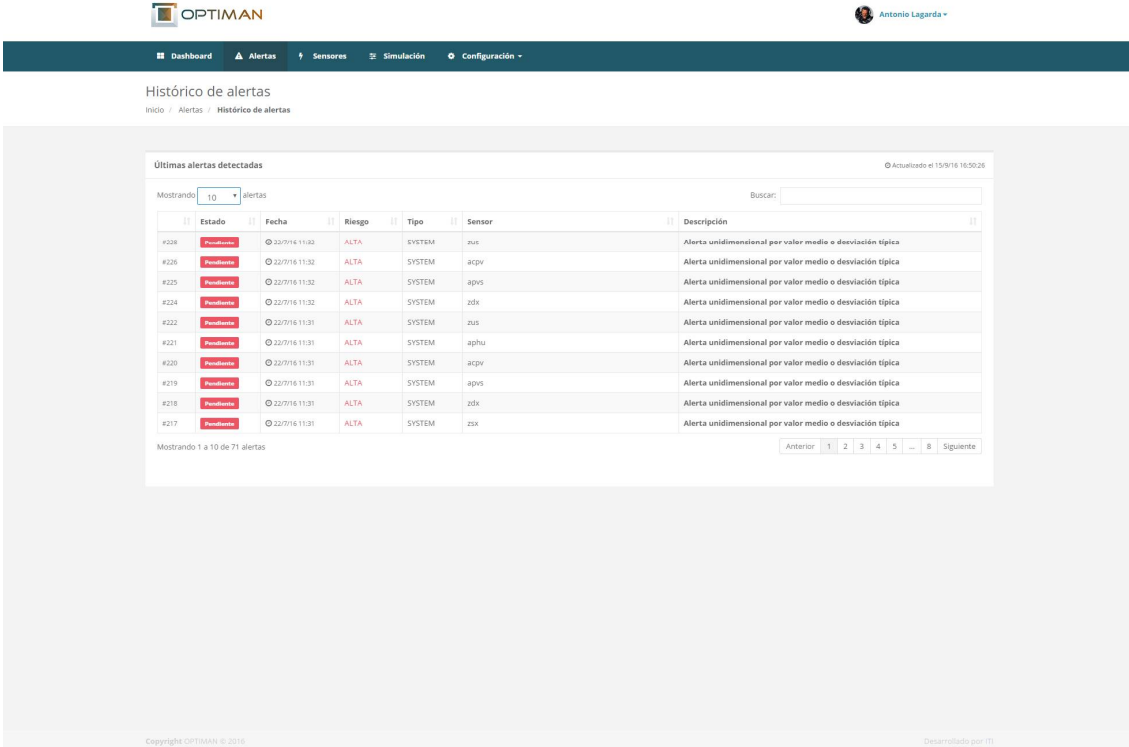
### 3.3.8 Pantalla de visualización de datos de los sensores



Esta pantalla permite la visualización y comparativa de la información de sensores disponible para un elemento de fabricación.

Permite la búsqueda filtrada por fechas realizando una consulta agregada contra el clúster de Cassandra a un determinado número de puntos por grafica (1500-2000) preconfigurado, de este modo la visualización de datos es independiente del volumen de datos almacenados permitiendo al usuario la consulta de históricos de larga duración sin afectar al rendimiento del sistema.

### 3.3.9 Pantalla de consulta de histórico de alertas



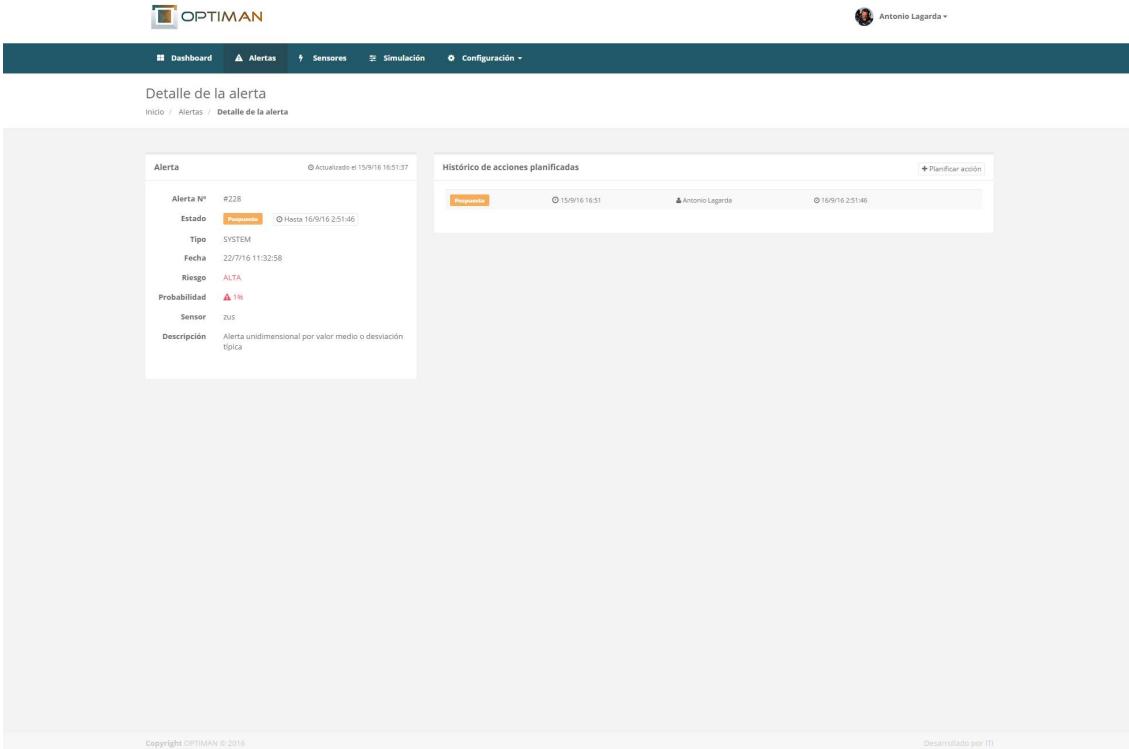
Copyright OPTIMAN © 2016. Desarrollado por ITI.

Esta pantalla muestra el histórico de alertas producidas en el sistema, permite una búsqueda paginada así como una búsqueda por descripción de la alerta.

El listado de alertas muestra información sobre el estado de la alerta, la fecha de alta en el sistema, la criticidad de la alerta, el tipo de la alerta, el sensor o sensores a los que está asociada la alerta así como una descripción de la misma sobre la que se permite la búsqueda.

Pulsando sobre una alerta en concreto navegamos a la pantalla de detalle de la alerta que describimos a continuación.

### 3.3.1 Pantalla de consulta de detalle de alerta



Esta pantalla muestra en detalle la información de una alerta dada, amplia la información de la pantalla anterior proporcionando información del histórico de cambios de estado de la misma, así como parámetros relativos al modelo estadístico que genere la alerta informando detalles como la probabilidad y confianza en la predicción.

Adicionalmente se permite la manipulación del estado de la alerta, permitiendo al usuario:

- Planificar la alerta, marca la alerta como procesada y planifica una actuación sobre el sensor o sensores asociados a la misma.
- Posponer la alerta, aplaza temporalmente el tratamiento de la alerta a la espera de un evento con mayor confianza o probabilidad.
- Descartar la alerta, marca la alerta como eliminada del sistema, sin tratamiento de la misma.

## 4. Validación del desarrollo

Para Raúl

## 5. Referencias bibliográficas

[Mashup] [https://es.wikipedia.org/wiki/Mashup\\_\(aplicaciónn\\_web\\_hibrida\)](https://es.wikipedia.org/wiki/Mashup_(aplicaciónn_web_hibrida))

[REST] [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)

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

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

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

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

[Modelos de regresión] [https://es.wikipedia.org/wiki/Regresión\\_lineal](https://es.wikipedia.org/wiki/Regresión_lineal)

[Análisis de Series Temporales] [https://es.wikipedia.org/wiki/Serie\\_temporal](https://es.wikipedia.org/wiki/Serie_temporal)

[Modelos de clustering] [https://en.wikipedia.org/wiki/Cluster\\_analysis](https://en.wikipedia.org/wiki/Cluster_analysis)

[Modelos avanzados de aprendizaje supervisado]  
[https://en.wikipedia.org/wiki/Supervised\\_learning](https://en.wikipedia.org/wiki/Supervised_learning)

[Modelos de reducción de dimensiones y posterior clasificación]  
[https://en.wikipedia.org/wiki/Dimensionality\\_reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction)