



Máster en Ingeniería Industrial

Trabajo fin de máster

Gestión de activos en IBM Máximo

Autor: Nora Segura Vallejo

Director: Adiba El Boujdaini

MADRID

Julio 2022

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Gestión de activos en IBM Máximo

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2021/2022 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Nora Segura Vallejo

Fecha: 17/07/2022



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Adiba El Boujdaini



Fdo.: Adiba El Boujdaini

Fecha: 18/07/2022



Máster en Ingeniería Industrial

Trabajo fin de máster

Gestión de activos en IBM Máximo

Autor: Nora Segura Vallejo

Director: Adiba El Boujdaini

MADRID

Julio 2022

RESUMEN DEL PROYECTO

Introducción

En la actualidad, todas las empresas invierten su tiempo y su dinero en tratar de optimizar la vida útil de sus activos. Es por ello, que llevar a cabo un buen plan de mantenimiento de dichos activos se ha convertido en una parte esencial para las mismas. Como consecuencia del progreso tecnológico sin precedentes de los últimos años, han surgido nuevas tecnologías capaces de mejorar la eficiencia en el ámbito del mantenimiento, así como diferentes softwares especializados en llevar el mantenimiento a otros niveles y, de esta manera, añadir valor a la empresa.

En este proyecto, se va a utilizar el software IBM Máximo, utilizado a nivel mundial, que permite digitalizar y gestionar todos los activos de una empresa para poder optimizar su vida útil.

Dicho software se utilizará para crear un sistema que pueda gestionar diferentes activos de una organización y su mantenimiento de forma práctica. Se utilizarán distintas partes de Máximo como las *Escalations* o los *Automation Scripts* para gestionar las diferentes órdenes de trabajo que tengan los activos, además de tramitar las órdenes de trabajo generadas debido al mantenimiento preventivo

Objetivos

Como ya se ha comentado, el proyecto va a consistir en la creación de un sistema completo desde 0, que contenga activos y distintos elementos con sus propias órdenes de trabajo, *Escalations* y *Automation Scripts*. Todo esto encadenado con la aplicación que tiene el software sobre mantenimiento preventivo. Por tanto, los principales objetivos del proyecto serán los siguientes:

- Crear una Organización con todos los elementos que haga falta para poder automatizar órdenes de trabajo y mantenimiento preventivo. (*Storerooms, Assets, Items, Job Plans, Work Orders...*)
- Crear *Escalations* mediante condiciones (SQuirreL) para cambiar estados de las órdenes de trabajo de los *Assets* e *Items*.
- Crear *Automation Scripts* (Python/Jython) para generar las órdenes de trabajo y actualizar la información de estas. Los *Automation Scripts* son similares a las *Escalations*, pero sirven para gestionar temas algo más complejos que en las *Escalations* no se podrían gestionar de forma eficiente.
- Crear el Plan de Mantenimiento Preventivo (PMP) mediante frecuencias y medidas de los activos. En caso de que se tenga más tiempo, se podrían simular lecturas de distintos parámetros de los activos en cuestión, y que lleguen a Máximo mediante otro programa.
- Crear gráficos de los distintos parámetros de los activos para que el usuario pueda verlos en el *Start Center*.

Diseño

Para la realización de este proyecto se tomará como ejemplo un taller de coches en el que se deban gestionar los activos de los clientes y los propios activos del taller. En este caso, habrá que gestionar 2 coches y una moto de 3 usuarios distintos y 2 máquinas del taller.

Organización, *site* y sus elementos

Antes de comenzar con el plan de mantenimiento y todas las automatizaciones que se requieren para este, se crean todos los elementos y usuarios que van a participar en el sistema:

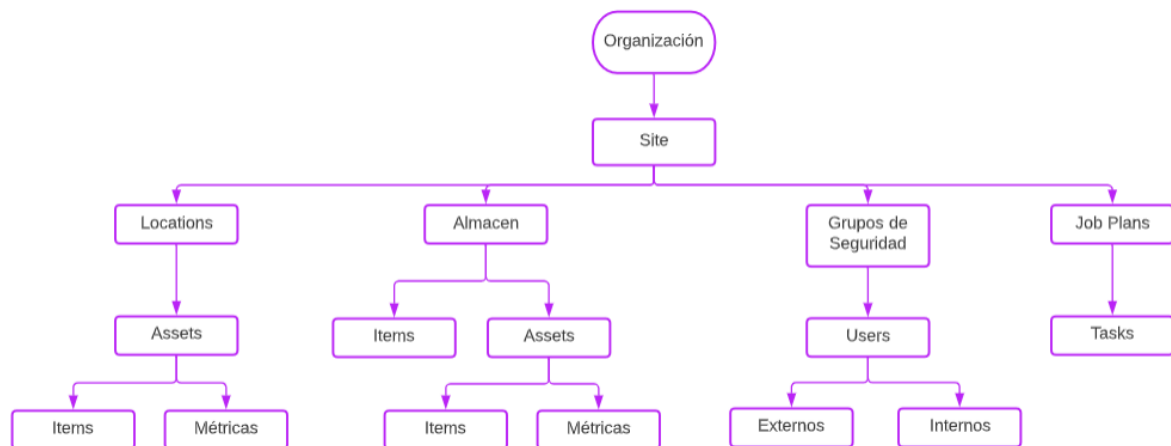


Ilustración 1: Organización, *site* y sus elementos

Estos elementos se crean desde las distintas aplicaciones que tiene IBM Máximo.

Habrà 3 *locations*, una para cada lugar de operación de los *assets* de los clientes, y 1 almacén, que será el taller en sí. Los coches y la moto estarán formados por diferentes *items*, como los neumáticos o el sistema de frenado. Estos *items* estarán también en el taller, para utilizarlos de recambio en caso de necesidad.

Los *assets* tendrán distintas métricas que servirán para activar tanto el plan de mantenimiento según el kilometraje como las diferentes automatizaciones que se van a realizar.

Además, se crearán 2 grupos de seguridad para los clientes del taller y los operadores de este, y se les darán distintos permisos. En el caso del operador del taller, tendrá acceso a su inventario, a los *assets* del sistema, a las órdenes de trabajo, a las *Service Request* de los clientes y a la nueva aplicación que se va a crear para gestionar el mantenimiento. Los clientes tendrán acceso a las *Service Request*, ya que podrán crearlas en caso de necesidad. Además, podrán ver los datos de la nueva aplicación para conocer la situación de sus *assets*.

Por último, se crearán los diferentes Job Plans con sus tareas, que servirán para poder crear las órdenes de trabajo automáticamente en base a estos.

Plan de Mantenimiento Preventivo

Habrà 5 planes de mantenimiento preventivo, uno para cada *asset* que se va a crear. Los 3 *assets* de los clientes se gestionarán mediante frecuencias basadas en tiempo y frecuencias

basadas en métricas. La métrica que se va a utilizar es el kilometraje de los vehículos. En el caso de las 2 máquinas del taller, solamente se utilizarán frecuencias basadas en tiempo.

Estos planes generarán distintos órdenes de trabajo, que podrá ver el operador, basadas en los *Job Plans* ya creados y asociados a estos.

Nueva Aplicación

Se creará una nueva aplicación para que el operador haga las revisiones de los vehículos cuando salte el plan de mantenimiento. Esta nueva aplicación cogerá algunos datos de las órdenes de trabajo que se creen mediante los PMs y tendrá distintos campos a rellenar según el tipo de vehículos que sea. El operador tendrá que rellenarlos, y, una vez hecho esto, se crearán distintas órdenes de trabajo hijas de la anterior, dependiendo de si los valores introducidos son correctos o no. Además, si el operador introduce valores desorbitados, la aplicación le avisará con un mensaje de error. Para conseguir esto, se utilizan varios *Automation Scripts*.

Automation Scripts

Se van a crear 3 scripts diferentes para realizar la automatización de la gestión del mantenimiento de los vehículos:

1. **Nueva aplicación:** Automatizará que, cuando se genere un orden de trabajo desde los PMs, se genere una nueva revisión en la nueva aplicación que se ha creado. Además, rellenará automáticamente algunos campos de la revisión basándose en la orden de trabajo, como el número de esta o el *asset* al que hay que hacerle la revisión.
2. **Mensajes de error:** Automatizará que se creen distintos mensajes de error en algunos campos de la nueva aplicación cuando el operador añada valores que no son lógicos.
3. **Nuevas órdenes de trabajo:** Automatizará que se creen nuevas órdenes de trabajo hijas una vez que el operador haya añadido los valores que se piden en la nueva aplicación.

Escalations

Se van a crear 2 Escalations distintas para otras automatizaciones más sencillas:

1. **Service Request:** Se va a crear una automatización para que, cuando los clientes creen una nueva SR y el operador la resuelva y cambie su estado a 'RESOLVED', se mande una notificación automáticamente a los clientes para que sepan que pueden ir a recoger su vehículo.
2. **Máquinas del taller:** Cuando se genera una orden de trabajo mediante los PMs para cualquier máquina del taller, esta tendrá 3 tareas, la primera la de analizar la máquina en cuestión. Si el operador realiza un análisis favorable y cambia el estado de la primera tarea a 'TERMINADA', la Escalation cancelará las 2 tareas restantes y cambiará el estado de estas a 'CAN'. En caso de que simplemente cambie el estado a 'COMPLETED', el operador tendrá que seguir con las 2 tareas restantes.

Resultados

Se crean las órdenes de trabajo desde la aplicación de Mantenimiento Preventivo basadas en el *Job Plan*:

Work Order: 3034 Revisión Coche Program: ICAI Attachments

Location: CLIENTE1 Lugar de operación del coche 1 Class: WORKORDER Status: WAPPR

Asset: 1017 Coche 1 Work Type: Status Date: 7/14/22 00:00:00

Configuration Item: GL Account: Inherit Status Changes? ☐

Feature: Failure Class: Accepts Charges? ☒

Feature Label: Problem Code: Is Task? ☐

Parent WO: Storeroom Material Status: Under Flow Control? ☐

Classification: Direct Issue Material Status: Suspend Flow Control? ☐

Class Description: Work Package Material Status: Flow Action: Flow Action Assist? ☐

Launch Entry Name: Material Status Last Updated: Job Details Asset Details Priority

Job Plan: 2363 Asset Up? ☐ Asset/Location Priority:

Ilustración 2: PM

Al mismo tiempo, se crea automáticamente la revisión en la nueva aplicación y al rellenarse se verifica que salen los mensajes de error:

Find Rev_vehiculosid Find Navigation Item List View Main

Go To Applications Available Queries All Records EN PROGRESO NUEVO REV INPROG USUARIO REV TERMINADA USUARIO TERMINADO Common Actions New REVVEHC Save REVVEHC Clear Changes More Actions Delete REVVEHC Add to Bookmarks

Work Order NUM: 3034 Description: Revisión Coche Fecha: 7/14/22 00:00:00 Status: NUEVO

Usuario: Cliente: Asset: 1017 Coche 1 Site: ICAI

Tareas

Revisión Neumáticos

Presion_neum: 2.5 Desgaste_neum: BIEN

Revisión Luces

Luminosidad: BIEN

Revisión Batería

Voltaje_bat: 100

Revisión Sistema Frenado

Nivel_liq_frenos: Desgaste_past: Fd: Ft: Fmax: Fm: Fd: Ft: Fm: Fm:

You entered: 100
Valor del voltaje de la batería (VOLTAJE_BAT=100.0) muy superior a cualquier valor normal de voltaje. Por favor, completa el campo con otro valor. (BMXZZ7149)
Click "Edit My Value" to change the value you entered or "Go Back" to the value that was there before.
Edit My Value Go Back

Ilustración 3: Nueva aplicación

Una vez rellena, se crean automáticamente 2 órdenes nuevas de trabajo, ya que los valores de la batería y del sistema de frenado no son los correctos:

Work Order	Reported Date	Actual Finish	Location	Classification	Description	Status	Owner	Job Plan	Asset	PN	SN	Structural Code	Is dynamic?	Dynamic Job Plan
3173	7/14/22 00:41:56		CLIENTE1	Recambio Sistema de Frenado	WAPPR				1017	3	5		NO	
3172	7/14/22 00:41:55		CLIENTE1	Cambiar Batería	WAPPR				1017	3	5		NO	

Ilustración 4: Nuevas órdenes de trabajo

Por otra parte, la *Escalation* de las máquinas del taller cancela automáticamente la segunda y tercera tarea si la primera está en 'TERMINADA':

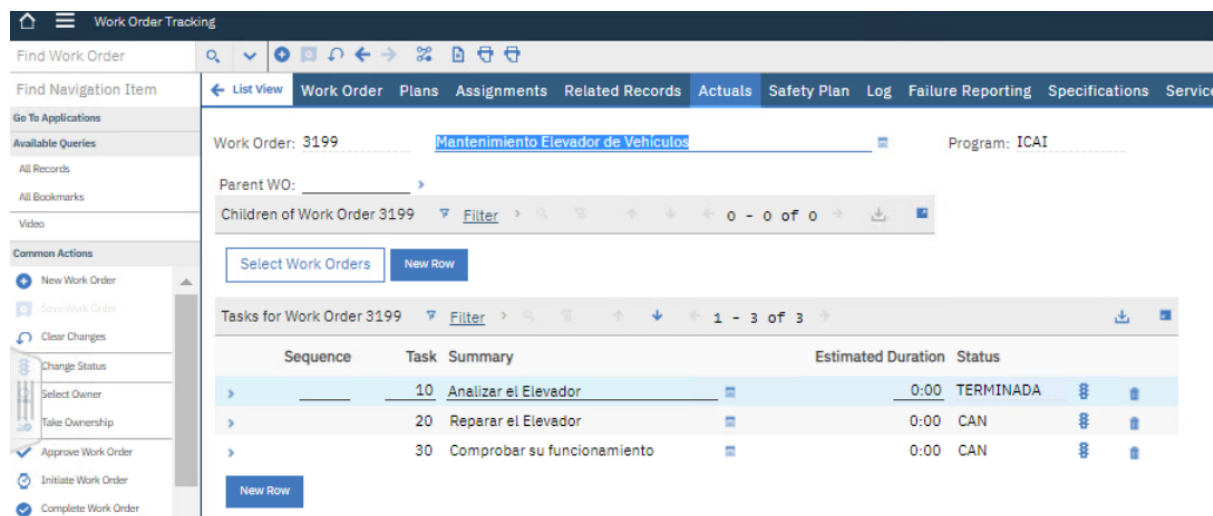


Ilustración 5: Tareas máquinas taller

La *Escalation* de las *Service Request* no puede ser probada, ya que, en el entorno en el que se ha desarrollado este sistema, no hay servicio de correo.

Conclusión

El proyecto concluye habiendo diseñado y creado un sistema de gestión de activos con mantenimiento preventivo. Los objetivos finales que se propusieron para crear este sistema en Máximo se han cumplido, y, además, se han añadido al sistema más elementos de los que se habían propuesto en un principio. Se decidió crear un sistema basado en algo real, como es un taller de coches, y ha sido todo un acierto, ya que, con este ejemplo, se entiende mucho mejor el funcionamiento del software en sí y de todo el sistema creado.

Cuando se comenzó el proyecto, estos eran los principales objetivos a realizar con el software IBM Máximo:

- Crear una Organización con todos los elementos
- Crear *Escalations*
- Crear *Automation Scripts*
- Crear el Plan de Mantenimiento Preventivo
- Crear el *Start Center*

Se han cumplido todos, pero, además, se ha añadido el objetivo de crear una nueva aplicación para que el sistema fuera más automático y sencillo de utilizar para el operador. Gracias al hecho de crear esta nueva aplicación de revisión de coches, se han utilizado múltiples aplicaciones de Máximo que se interconectan entre sí.

ABSTRACT

Introduction

Nowadays, all companies invest time and money in trying to optimize the useful life of their assets. This is why a good maintenance plan for these assets has become an essential part of their business. As a consequence of the unprecedented technological progress of the last few years, new technologies capable of improving efficiency in the field of maintenance have emerged, as well as different software specialized in taking maintenance to other levels and, in this way, adding value to the company.

In this project, the IBM Maximo software, used worldwide, will be used to digitize and manage all the assets of a company in order to optimize their useful life.

This software will be used to create a system capable of managing different assets of an organization and their maintenance in a practical way. Different parts of Maximo, such as Escalations or Automation Scripts, will be used to manage the different work orders that the assets have, as well as processing the work orders generated due to preventive maintenance.

Objectives

As already mentioned, the project will consist of the creation of a complete system from scratch, containing assets and different elements with their own work orders, Escalations and Automation Scripts. All of this will be linked to the application of the software Preventive Maintenance. Therefore, the main objectives of the project will be the following:

- Create an Organization with all the necessary elements to be able to automate work orders and preventive maintenance (Storerooms, Assets, Items, Job Plans, Work Orders...).
- Create Escalations using conditions (SQuirreL) to change the status of Assets and Items work orders.
- Create Automation Scripts (Python/Jython) to generate the work orders and update the information of these. Automation Scripts are similar to Escalations, but they are used to manage more complex issues that Escalations would not be able to manage efficiently.
- Create the Preventive Maintenance Plan (PMP) by means of frequencies and measurements of the assets. In case of having more time, it could be simulated readings of different parameters of the assets and have them reach Maximo by another program.
- Create graphs of the different asset parameters so that the user can view them in the Start Center.

Design

For the realization of this project the example of a car workshop will be taken, in which the assets of the customers and the workshop's own assets must be managed. In this case, it will be necessary to manage 2 cars and a motorbike belonging to 3 different users and 2 machines of the workshop.

Organization, site and its elements

Before starting with the maintenance plan and all the automations required for it, all the elements and users that are going to participate in the system must be created:

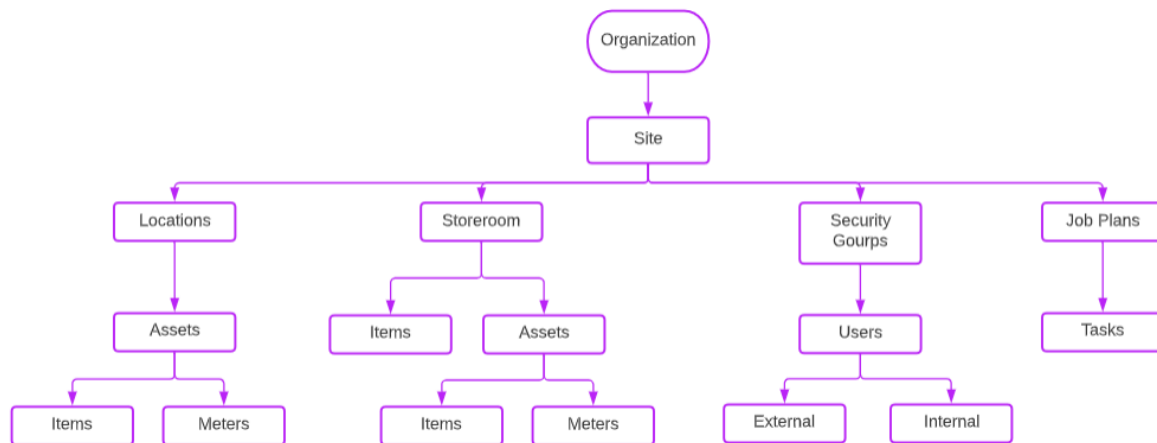


Figure 11: Organization, site and its elements

These elements are created from the different applications that IBM Maximo has.

There will be 3 locations, one for each place of operation of the customer's assets, and 1 warehouse, which will be the workshop itself. The cars and the motorbike will be made up of different items, such as tires or the braking system. These items will also be in the workshop, to be used as spare parts in case of need.

The assets will have different metrics that will be used to activate both the maintenance plan according to the mileage and the different automations that will be carried out.

In addition, 2 security groups will be created for workshop customers and workshop operators, and they will be given different permissions. In the case of the workshop operator, he will have access to his inventory, system assets, work orders, customer service requests and the new application that will be created to manage maintenance. Customers will have access to Service Requests, as they will be able to create them if necessary. They will also be able to view the data of the new application to know the status of their assets.

Finally, the different Job Plans will be created with their tasks, which will be used to automatically create work orders based on them.

Preventive Maintenance Plan

There will be 5 preventive maintenance plans, one for each asset to be created. The 3 customer assets will be managed by time-based frequencies and metric-based frequencies. The metric to be used is the mileage of the vehicles. For the 2 machines in the workshop, only time-based frequencies will be used.

These plans will generate different work orders, which can be viewed by the operator, based on the Job Plans already created and associated to them.

New application

A new application will be created for the operator to check the vehicles when the maintenance plan is triggered. This new application will take some data from the work orders created by the PMs and will have different fields to be filled in depending on the type of vehicles. The operator will have to fill them in, and, once this is done, different work orders will be created from the previous one, depending on whether the values entered are correct or not. In addition, if the operator enters excessive values, the application will warn him with an error message. To achieve this, some Automation Scripts are used.

Automation scripts

Three different scripts will be created to automate the management of vehicle maintenance:

1. **New application:** It will automate that, when a work order is generated from the PMs, a new revision will be generated in the new application that has been created. In addition, it will automatically fill in some fields of the review based on the work order, such as the work order number or the asset to be reviewed.
2. **Error messages:** It will automate the creation of different error messages in some fields of the new application when the operator adds values that are not logical.
3. **New work orders:** This will automate the creation of new child work orders once the operator has added the values requested in the new application.

Escalations

Two different Escalations will be created for other simpler automations:

1. **Service Request:** An automation is going to be created so that when customers create a new SR and the operator resolves it and changes its status to 'RESOLVED', a notification is automatically sent to the customers so that they know they can go and pick up their vehicle.
2. **Workshop machines:** When a work order is generated via PMs for any machine in the workshop, it will have 3 tasks, the first one is to analyze the machine. If the operator performs a favorable analysis and changes the status of the first task to 'TERMINADA', the Escalation will cancel the remaining 2 tasks and change their status to 'CAN'. In case it simply changes the status to 'COMPLETED', the operator will have to continue with the remaining 2 tasks.

Results

Work orders are created from the Preventive Maintenance application based on the Job Plan:

The screenshot shows the 'Work Order' form in the Preventive Maintenance application. The form is titled 'Revisión Coche' and includes fields for 'Work Order: 3034', 'Location: CLIENTE1', 'Asset: 1017', 'Program: ICAI', 'Class: WORKORDER', 'Status: WAPPR', and 'Status Date: 7/14/22 00:00:00'. There are also fields for 'Configuration Item', 'Feature', 'Feature Label', 'Parent WO', 'Classification', 'Class Description', 'Launch Entry Name', 'Job Details', 'Asset Details', 'Priority', 'Job Plan: 2363', 'Asset Up?', and 'Asset/Location Priority:'. The form is divided into sections for 'Job Details', 'Asset Details', and 'Priority'.

Figure 2: PMs

At the same time, the revision is automatically created in the new application and when it is filled in, the error messages are checked:

The screenshot shows the 'Revisión Vehículo' application. The form is titled 'Revisión Vehículo' and includes fields for 'Work Order NUM: 3034', 'Description: Revisión Coche', 'Fecha: 7/14/22 00:00:00', 'Status: NUEVO', 'Usuario:', 'Cliente:', 'Asset: 1017', 'Coche 1', and 'Site: ICAI'. There are sections for 'Revisión Neumáticos', 'Revisión Luces', 'Revisión Batería', and 'Revisión Sistema Frenado'. A validation error message is displayed over the 'Voltaje_bat:' field, stating: 'You entered: 100. Valor del voltaje de la batería (VOLTAJE_BAT=100.0) muy superior a cualquier valor normal de voltaje. Por favor, completa el campo con otro valor. (BMXZZ7149)'. The message also includes instructions to click 'Edit My Value' to change the value or 'Go Back' to return to the previous state.

Figure 3: New application

Once filled in, 2 new work orders are automatically created, as the battery and braking system values are not correct:

The screenshot shows the 'Work Order Tracking' application. The table displays the following data:

Work Order	Reported Date	Actual Finish	Location	Classification	Description	Status	Owner	Job Plan	Asset	PN	SN	Structural Code	Is dynamic?	Dynamic Job Plan
3173	7/14/22 00:41:56		CLIENTE1		Recambio Sistema de Frenado	WAPPR			1017	3	5			NO
3172	7/14/22 00:41:55		CLIENTE1		Cambiar Batería	WAPPR			1017	3	5			NO

Figure 4: New work orders

On the other hand, the Escalation of the workshop machines automatically cancels the second and third tasks if the first one is set to 'TERMINADA':

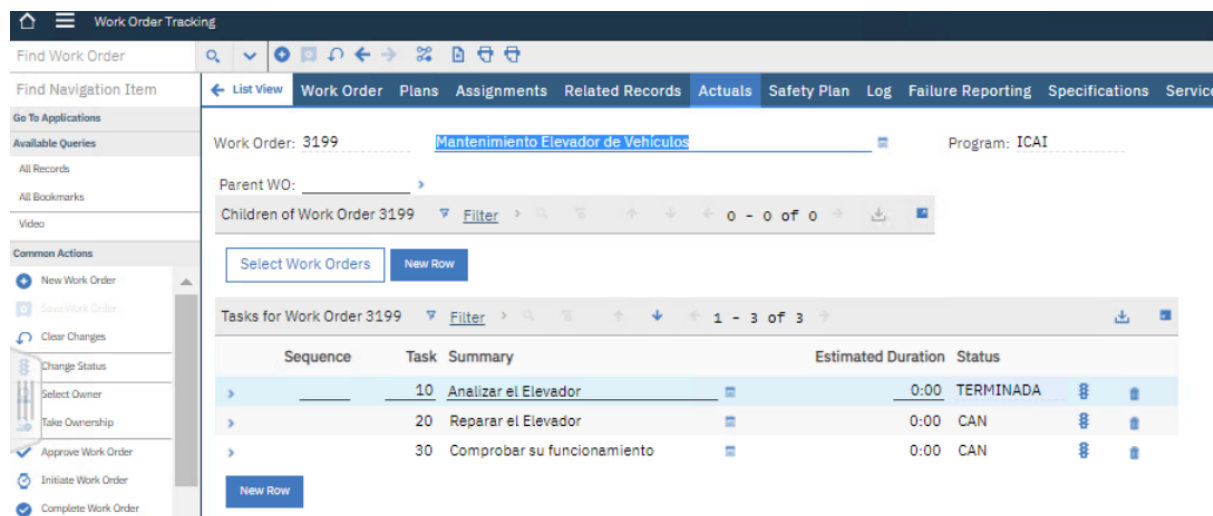


Figure 5: Workshop's machine tasks

The Escalation of Service Requests cannot be tested, as there is no mail service in the environment in which this system has been developed.

Conclusion

The project concludes having designed and created an asset management system with preventive maintenance. The final objectives that were proposed to create this system in Máximo have been met, and, in addition, more elements have been added to the system than had been proposed at the beginning. It was decided to create a system based on something real, such as a car workshop, and it has been a success, since, with this example, the functioning of the software itself and the entire system created is much better understood.

When the project started, these were the main objectives to be achieved with the IBM Maximo software:

- Create an Organisation with all the elements
- Create Escalations
- Create Automation Scripts
- Create the Preventive Maintenance Plan
- Create the Start Center

All of them have been fulfilled, but, in addition, the objective of creating a new application to make the system more automatic and easier to use for the operator has been added. Thanks to the fact of creating this new car revision application, multiple Maximo applications have been used that are interconnected with each other.

índice

Introducción	20
Estado de la cuestión	21
Mantenimiento correctivo	26
Mantenimiento preventivo	26
Mantenimiento predictivo	27
Niveles de Mantenimiento.....	28
Objetivos	29
Diseños conceptuales del sistema	30
Diseño de la Organización	30
<i>Locations</i>	32
Almacén.....	32
Grupos de Seguridad	34
Usuarios, assets e items	34
<i>Job Plans</i>	37
Diseño del Plan de Mantenimiento Preventivo	42
Frecuencias.....	42
Métricas.....	42
Automatismos de los <i>Automation Scripts</i>	43
Diseño de la nueva aplicación para el operador	43
Nueva tabla.....	43
Mockup de la aplicación	45
Automatismos de las <i>Escalations</i>	49
Máquinas del taller	49
<i>Service Requests</i>	49
Start Center	50
Operador.....	50
Usuarios	50
Creación de la Organización y sus elementos	51
Organización.....	51
<i>Site</i>	51
<i>Locations</i>	52

Métricas.....	52
<i>Items</i>	53
<i>Assets</i>	54
<i>Storeroom</i>	57
<i>Job Plans</i>	58
<i>Security Groups</i>	59
Creación del Plan de Mantenimiento Preventivo.....	62
Frecuencias basadas en tiempo	63
Frecuencias basadas en métricas.....	63
<i>Forecast</i>	64
Creación de la nueva aplicación.....	65
<i>Database Configuration</i>	65
<i>Messages</i>	67
<i>Domains</i>	69
<i>Lookup</i>	69
<i>Automation Scripts</i>	70
Conditional Experession Manager	70
Application Designer	71
Lookups.....	73
Resultado final	74
Creación de los <i>Automation Scripts</i>	76
NUEVAAPP.....	76
ERRORESVEHICULOS	77
WOCHILD	78
Creación de las <i>Escalations</i>	80
Máquinas del taller.....	80
<i>Actions</i>	81
<i>Service Request</i>	81
Communication Templates	82
Creación del <i>Start Center</i>	83
Operador	85
Usuarios.....	85
Resultados.....	86

Escalation Máquinas	86
Preventive Maintenance	88
Automation Scripts.....	89
NUEVAAP	89
ERRORES	90
WOCHILD	91
Start Centers.....	92
Conclusiones	93
Anexos.....	94
Anexo I: Objetivos de Desarrollo Sostenible.....	94
Anexo II: XML de la nueva aplicación.....	95
Anexo III: Script para generar revisiones en la nueva aplicación.....	103
Anexo IV: Script para generar mensajes de error	105
Anexo V: Script para generar órdenes de trabajo hijas	107
Bibliografía	112

Índice de Ilustraciones

Ilustración 1: Taxonomía de la Transformación Digital (3).....	23
Ilustración 2: Gráfica de probabilidad de fallo de un activo (4)	24
Ilustración 3: Diferencia entre Entretenimiento y Mantenimiento (5)	25
Ilustración 4: Factores de cambio (5).....	25
Ilustración 5: Ejecución del mantenimiento (5).....	26
Ilustración 6: Comparación de los tipos de mantenimiento (6)	27
Ilustración 7: Niveles 1, 2 y 3	28
Ilustración 8: Niveles 3 y 4	28
Ilustración 9: Esquema Organización.....	30
Ilustración 10: Esquema taller	32
Ilustración 11: Esquema usuarios y assets.....	34
Ilustración 12: Esquema coche	35
Ilustración 13: Esquema moto	35
Ilustración 14: Esquema Job Plans.....	37
Ilustración 15: PMs	42
Ilustración 16: Pantalla nueva Aplicación	45
Ilustración 17: Acciones de la ap	45
Ilustración 18: Información a rellenar	46
Ilustración 19: Tareas a realizar	46
Ilustración 20: Medidas a rellenar	47
Ilustración 21: Elementos de la nueva App	47
Ilustración 22: Organización	51
Ilustración 23: Site	52
Ilustración 24: Locations	52
Ilustración 25: Métricas	53
Ilustración 26: Items	53
Ilustración 27: Ejmplo creación item	54
Ilustración 28: Assets	54
Ilustración 29: Ejemplo creación asset 1	55
Ilustración 30: Métricas del asset 1	55
Ilustración 31: Ejemplo creación asset 2	56
Ilustración 32: Métricas del asset 2	56
Ilustración 33: Ejemplo creación asset 3	57
Ilustración 34: Storeroom con sus items	57
Ilustración 35: Storeroom con sus assets	58
Ilustración 36: Item en Inventory	58
Ilustración 37: Job Plans.....	58
Ilustración 38: Ejemplo creación de Job Plan	59
Ilustración 39: Tareas Job Plan.....	59
Ilustración 40: Security Groups.....	60
Ilustración 41: Ejemplo de permisos de los SG.....	60

Ilustración 42: Ejemplo Data Restrictions de los SG	61
Ilustración 43: Ejemplo de los usuarios del SG	61
Ilustración 44: PMs	62
Ilustración 45: Job Plan del PM	62
Ilustración 46: Frecuencias de tiempo en los PMs	63
Ilustración 47: Frecuencias de métricas en los PMs	64
Ilustración 48: Forecast de los PMs	64
Ilustración 49: Nueva tabla en la base de datos	65
Ilustración 50: Atributos de la nueva tabla	66
Ilustración 51: Ejemplo de atributo	66
Ilustración 52: Relación con otras tablas	67
Ilustración 53: Mensajes de error	68
Ilustración 54: Ejemplo mensaje de error	68
Ilustración 55: Dominios	69
Ilustración 56: Condición para mostrar u ocultar campos	71
Ilustración 57: Campos para filtrar	71
Ilustración 58: Pantalla principal aplicación	72
Ilustración 59: Mostrar/Ocultar campos	73
Ilustración 60: Asociar dominios	73
Ilustración 61: Relación de tablas	74
Ilustración 62: Revisión coche	75
Ilustración 63: Revisión moto	75
Ilustración 64: Launchpoint NUEVAAPP	76
Ilustración 65: Launchpoints ERRORVEHICULOS	77
Ilustración 66: Launchpoint WOCHILD	78
Ilustración 67: Escalation Máquinas	80
Ilustración 68: Acción Escalation	81
Ilustración 69: Escalation Service Request	81
Ilustración 70: Escalation Point Service Request	82
Ilustración 71: Communication Template	82
Ilustración 72: Tipos de tablas en el Start Center	83
Ilustración 73: Atributos de cada tabla	84
Ilustración 74: Query 'SRINPROG1'	84
Ilustración 75: Start Center Operador	85
Ilustración 76: Start Center Usuarios	85
Ilustración 77: Tareas orden de trabajo elevador	86
Ilustración 78: Tarea 1 'TERMINADA'	86
Ilustración 79: Tareas canceladas	87
Ilustración 80: Orden de trabajo creada automáticamente	88
Ilustración 81: Hijas de la orden de trabajo	88
Ilustración 82: Nueva Revisión	89
Ilustración 83: Errores	90
Ilustración 84: Valores añadidos en la revisión	91
Ilustración 85: 2 nuevas órdenes de trabajo	91

Ilustración 86: Órdenes de trabajo hijas.....	92
Ilustración 87: Resultado Start Center Operador	92
Ilustración 88: Resultado Start Center Usuarios.....	92
Ilustración 89: Objetivos de Desarrollo Sostenible.....	94

Introducción

En la actualidad, todas las empresas invierten su tiempo y su dinero en tratar de optimizar la vida útil de sus activos. Es por ello, que llevar a cabo un buen plan de mantenimiento de dichos activos se ha convertido en una parte esencial para las mismas. Como consecuencia del progreso tecnológico sin precedentes de los últimos años, han surgido nuevas tecnologías capaces de mejorar la eficiencia en el ámbito del mantenimiento, así como diferentes softwares especializados en llevar el mantenimiento a otros niveles y, de esta manera, añadir valor a la empresa.

La base para poder lograr este objetivo es la digitalización. “A nivel conceptual, la digitalización, en su sentido más literal, hace referencia al registro, codificación, almacenamiento y transporte de datos de forma digital, es decir, mediante la combinación de bits o unidades básicas de información en los sistemas de numeración binario” (1). Llevamos años inmersos en la transformación digital, que beneficia a las empresas, y a la sociedad, en muchos aspectos. Sin embargo, no es tan sencillo conseguir dicha transformación, ya que hacerlo implica un cambio en las formas de hacer y de ser de la empresa.

En este proyecto, se va a utilizar el software IBM Máximo, utilizado a nivel mundial, que permite digitalizar y gestionar todos los activos de una empresa para poder optimizar su vida útil.

Dicho software se utilizará para crear un sistema que pueda gestionar diferentes activos de una organización y su mantenimiento de forma práctica. Se utilizarán distintas partes de Máximo como las *Escalations* o los *Automation Scripts* para gestionar las diferentes órdenes de trabajo que tengan los activos, además de tramitar las órdenes de trabajo generadas debido al mantenimiento preventivo. Para esta última aplicación del software se utilizarán tanto frecuencias como medidas, las cuales podrán ser enviadas desde otros programas a Máximo simulando lecturas reales de distintos parámetros.

Estado de la cuestión

Hoy en día la digitalización se ha convertido en una parte importantísima para cualquier sector empresarial. Esta transformación digital comenzó hace años y muchas empresas han ido digitalizándose poco a poco. Sin embargo, hace 2 años, cuando la OMS (Organización Mundial de la Salud) declaró la pandemia del COVID-19, esta transformación se aceleró. Ciertamente que el COVID ha afectado a la sociedad de muchas formas negativas, pero en lo referente a este tema, ha conseguido que en las empresas se produzca una verdadera revolución tecnológica. Desde entonces se han empezado a introducir innumerables nuevas estrategias en diferentes ámbitos como la producción, el empleo, las actividades administrativas y la comercialización. De estos ámbitos, las partes más afectadas por el COVID han sido las actividades administrativas y la comercialización. Se ha producido un crecimiento exponencial en cuanto al teletrabajo, y, además, se han empezado a utilizar distintas plataformas digitales para la comercialización. Esto ha afectado a gran parte de la sociedad, y, por tanto, ha creado una nueva conciencia al respecto y una nueva cultura digital. (2)

La transformación digital se basa en utilizar nuevas tecnologías en beneficio de una empresa o de la sociedad en general. Una definición bastante acertada es la de Bockschecker, Hackstein, & and Baumöl: “Proceso de cambio de una organización o la sociedad habilitado por innovaciones y desarrollos de las tecnologías de información y comunicaciones. Incluye la habilidad de adoptar tecnologías rápidamente e incidir en elementos sociales y técnicos de los modelos de negocio, procesos, productos y estructura organizacional”. Esto se podría categorizar en 3 elementos distintos: tecnológico, organizacional y social. Como ya se ha comentado antes, lo social es una parte bastante importante, ya que, para dar el paso a la digitalización, primero debe haber esta cultura en la empresa, algo que no se adquiere tan fácilmente. Para cualquier organización es imprescindible crearla en todos sus empleados y clientes y, para ello, hay 3 pasos importantes que se deben dar: “articular el cambio requerido, activar el liderazgo y el compromiso de los empleados, y alinear la organización para apropiarse de la nueva cultura digital”. (3)

Esta transformación se basa en las nuevas tecnologías que han surgido en la cuarta revolución industrial, y, por tanto, es muy importante decidir qué tecnologías son adecuadas para cada tarea o proyecto. Las más implicadas en este cambio tecnológico son la Nube, el *Big Data*, la Inteligencia Artificial, el Internet de las Cosas, la Robótica Colaborativa, la impresión 3D y el *Blockchain*. A continuación, se explican las tecnologías que más tienen que ver con este proyecto:

- **La Nube:** Es la tecnología más básica que habilita este cambio digital. Ofrece un almacenamiento casi ilimitado, soluciones rápidas, mayor facilidad para escalar servicios y la posibilidad de dar nuevos servicios a los clientes. Además de esto, las organizaciones se han visto forzadas a crear nuevos modelos de negocio, como el pago por uso. Esta tecnología se ha acercado mucho a la red y a derivado en algo que llaman la niebla, que es una capa intermedia, y que consigue reducir la latencia y el tráfico en la red. La parte más llamativa de esta nueva tecnología es la reducción de la latencia, ya que para muchos proyectos en tiempo real es muy beneficioso.

- **El Big Data:** Esta tecnología se basa en la manipulación de grandes cantidades de datos. Hoy en día los datos se han convertido en una fuente de información muy necesaria para cualquier empresa. Lo que consigue el *Big Data* es generar información que añada valor a la empresa analizando y manipulando estos datos. Muchas veces, es muy complicado extraer información que verdaderamente genere beneficios a la empresa, y es por esto que se utiliza esta tecnología con sus analíticas.
- **El Internet de las Cosas (IoT):** Se basa en la conexión de objetos físicos con la red. Son redes de objetos que llevan sensores, softwares, actuadores y otras muchas cosas más con el fin de conectar estas cosas a la red y generar un tráfico de información que permite analizar datos, generar órdenes a los actuadores de estos objetos físicos y muchas más cosas. También sirve para poder intercambiar datos entre dispositivos. Por tanto, esta tecnología se puede utilizar para tomar decisiones inteligentes en tiempo real.

El software IBM Máximo utiliza muchas bases de datos, analiza los datos y automatiza órdenes de trabajo de los activos de una empresa. En este proyecto no se va a utilizar nada físico, pero este software puede recibir datos de objetos físicos aplicando *IoT* y actuar en consecuencia.

Otra parte importante de esta transformación digital es el modelo de negocio, ya que consigue analizar cómo crear valor con estas nuevas tecnologías. Los componentes de los modelos de negocio se pueden agrupar en 3 grupos más generales:

Creación de valor	Entrega de valor	Captura de valor
• Desarrollo de nuevos servicios basados en captura, seguimiento e interpretación de datos. • Infraestructuras conectadas a las infraestructuras claves de otros actores del ecosistema. • Necesidad de nuevos recursos físicos, humanos e intelectuales. • Trazabilidad de productos y servicios, gestión más transparente y monitoreo en tiempo real.	• Flexibilidad de la oferta y personalización en masa. • Segmentación basada en análisis de datos. • Relaciones más directas, eficientes y duraderas con el ecosistema digital. • Ventas digitales mejoradas con la experiencia del cliente y la omnicanalidad. • Productos inteligentes. • Co-creación e innovación abierta.	• Optimización del costo debido a la eficiencia, lo que se manifiesta en toda la cadena de valor. • Precios dinámicos, pago-por uso, pagos on-line, etc.

Tabla 1: Componentes de modelos de negocio (3)

Los principales impactos que tiene como consecuencia la transformación digital, se pueden clasificar en 3 (3):

- La eficiencia interna de la empresa. Utilizar estas tecnologías para crear procesos más eficientes, más seguros y que aporten valor a la empresa.
- Las nuevas oportunidades externas que se crean. Se han creado nuevos modelos de negocio, se llega a la gente de diferente forma con estas nuevas tecnologías...
- Los cambios disruptivos que todo esto conlleva, así como en los roles de cualquier empresa.

Uno de los desafíos más importantes al que se enfrenta una empresa si quiere digitalizarse es el de tener capacidad dinámica. Lo que esto quiere decir es que se necesita “agilidad” para que la empresa, en cualquier momento, pueda reconfigurar y modificar sus activos y dispositivos digitales. Esto significa que se requieren bajos niveles de jerarquías con capacidades analíticas y de tecnología de la información internalizadas en la organización. Esto es muy importante para poder adaptarse a los rápidos cambios que ocurren y para que se puedan aprovechar todas las oportunidades de negocio que surjan de estos cambios. Además de esto, estar al tanto de las últimas tecnologías que salen al mercado y ser capaz de innovar, son características muy necesarias en este mundo de la digitalización.

Por último, se muestra un mapa conceptual que contempla todas las partes que residen en una transformación digital:

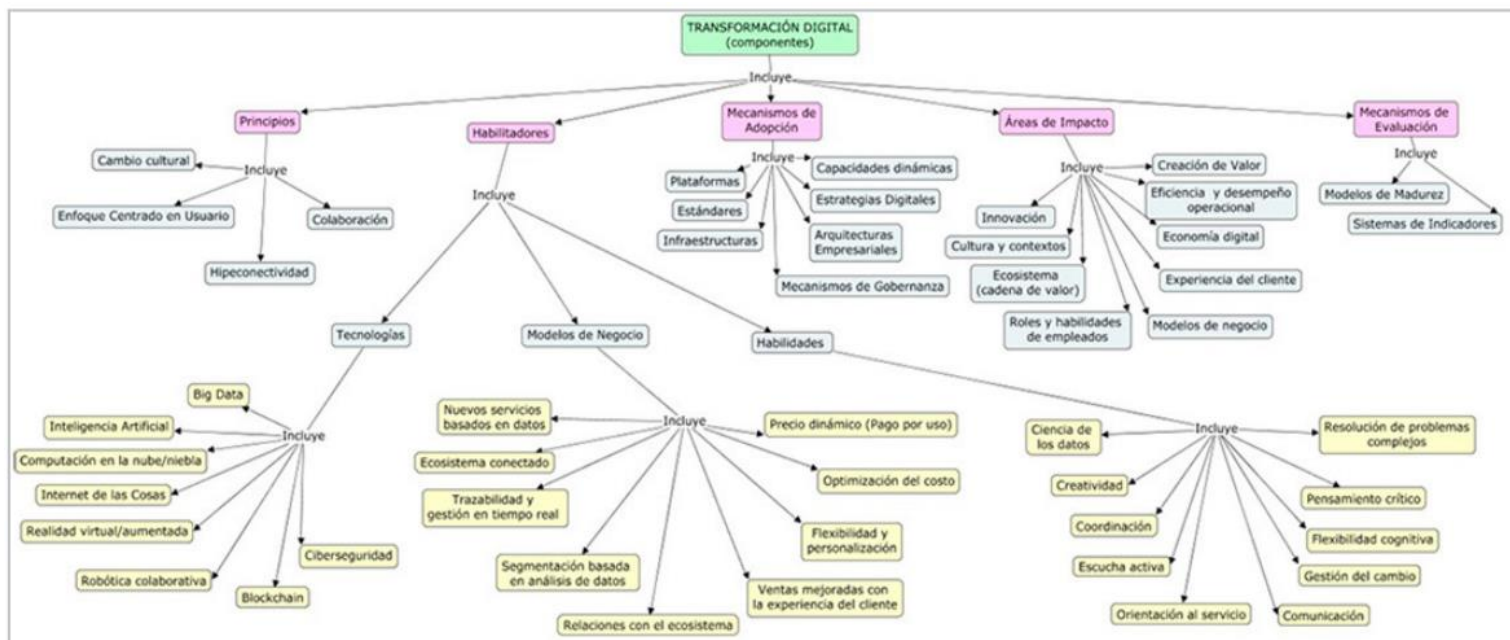


Ilustración 2: Taxonomía de la Transformación Digital (3)

Como ya se ha explicado, la digitalización es una parte fundamental para cualquier empresa hoy en día. Gracias a ello, se pueden aplicar métodos como los de mantenimiento de activos y gestionarlos de una forma eficiente y segura.

El problema al que se intenta buscar solución con el tema del mantenimiento es el de la aparición de fallos en los diferentes equipos o máquinas, ya que esto constituye una gran parte de la ineficiencia de cualquier industria. Si, por ejemplo, una máquina falla en un momento crítico, esto hace perder tiempo y, sobre todo, recursos económicos a la empresa. Por tanto, antes de dar comienzo a la gestión del mantenimiento, hay que comprender mejor cuándo es más probable que ocurran estas averías en los activos. Por supuesto, hay que analizar la vida útil de cada activo por separado, ya que cada activo es distinto. En la Ilustración 1 se muestra una gráfica con los 3 periodos más importantes de cualquiera de ellos como la base para después poder analizar cada activo con sus características.

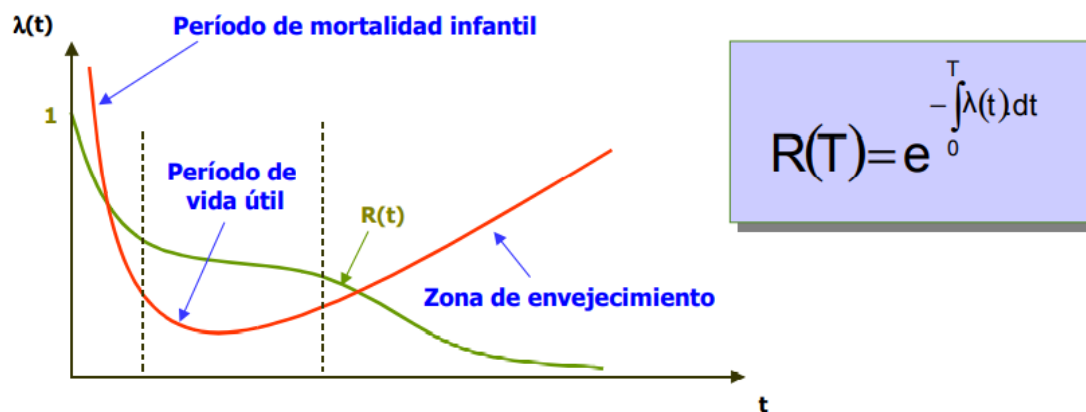


Ilustración 3: Gráfica de probabilidad de fallo de un activo (4)

La fiabilidad de cualquier dispositivo, $R(T)$, es lo que mide la resistencia al fallo de este, basada en la tasa de fallos $\lambda(t)$. $R(T)$ es una función decreciente, que, inicialmente ($t=0$), es igual a 1, considerando que el dispositivo está en perfectas condiciones. Las 3 fases por las que pasa cualquier dispositivo son: un primer periodo de mortalidad infantil, el período de vida útil en el que hay que dar importancia a evitar los primeros fallos que puedan ocurrir, y, finalmente, la zona de envejecimiento del dispositivo. (4)

Una vez hecho este pequeño análisis sobre la probabilidad de fallo de los dispositivos, es importante conocer los distintos métodos de mantenimiento que existen para elegir el más adecuado según el tipo de activo y su criticidad.

Para empezar, es necesario diferenciar lo que es el entretenimiento del mantenimiento. La principal diferencia es que el entretenimiento se refiere al mantenimiento correctivo, algo que a la larga no es tan eficiente como lo son otros mantenimientos.

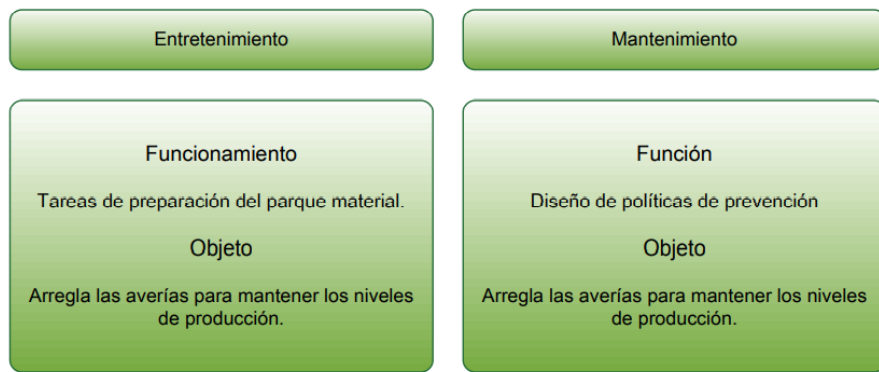


Ilustración 4: Diferencia entre Entretenimiento y Mantenimiento (5)

Antes, todas las empresas utilizaban el entretenimiento, pero cada vez más empresas están empezando a utilizar el mantenimiento para tratar de ser más eficientes y evitar gastos innecesarios.

Los factores más importantes que han llevado a las empresas a pasar del Entretenimiento al Mantenimiento son los siguientes:

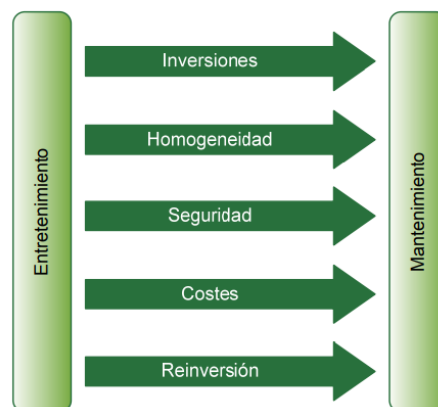


Ilustración 5: Factores de cambio (5)

Por una parte, las inversiones que una empresa hace en sus activos son tan grandes, que estas han comenzado a darle mucha importancia a alargar la vida útil de sus activos. Además de esto, los costes que puede generar tener una máquina parada en un proceso de producción pueden ser muy elevados, y por tanto se intenta que esto no ocurra gestionando un buen mantenimiento.

El mantenimiento se puede clasificar en diferentes grupos, dependiendo de a qué tipo de activos se les haga el mantenimiento y si se hace internamente en la empresa o externamente a los clientes que han adquirido productos de esta. En este último tipo, los clientes firman contratos de garantía con la empresa en caso de que quieran que la empresa les siga haciendo el mantenimiento. Pueden ser contratos para un determinado periodo de tiempo, para toda la vida útil del activo, solamente para gestionar el mantenimiento de algunas partes más críticas del activo...

Estas son las diferentes clasificaciones que puede haber respecto al mantenimiento de los activos:



Ilustración 6: Ejecución del mantenimiento (5)

Mantenimiento correctivo

Este tipo de mantenimiento se refiere al que se hace una vez que ha fallado el dispositivo. Su aplicación como mantenimiento no suele ser la mejor opción, pero cabe destacar que en cualquier plan de mantenimiento hay errores, y, por tanto, este tipo de mantenimiento se hace necesario para reparar las averías que surgen en el menor tiempo posible. Dentro de este tipo de mantenimiento existen 2 subgrupos:

- **Paliativo:** Es la reparación de un activo en el menor tiempo posible para que pueda seguir funcionando sin generar un retraso considerable en una cadena de producción, por ejemplo. No es una reparación del activo a largo plazo.
- **Curativo:** En este caso, el mantenimiento sí que es una acción definitiva. Al igual que en el paliativo se busca que el activo genere las mínimas pérdidas posibles a la empresa, pero en este caso, se arregla pensando a largo plazo.

Mantenimiento preventivo

Este mantenimiento se basa en prevenir los fallos o averías de los dispositivos. Por tanto, el mantenimiento como tal se ejecuta antes de que se creen las averías, al contrario de lo que ocurre en el correctivo. Con esto se evita la degradación de los dispositivos y muchos de los posibles incidentes que suceden en los dispositivos por diferentes causas. En este caso, el mantenimiento también se puede subdividir en 2 grupos:

- **Sistemático:** Este es el más básico de los 2. Simplemente se establece un plan de mantenimiento con una frecuencia determinada dependiendo del activo. Esta frecuencia suele darse como tiempo o también como el número de veces que se ha utilizado el dispositivo.

- **Condicional:** Este tipo de mantenimiento preventivo se basa en mediciones que se hacen a los activos. Si, por ejemplo, un mes, una medición de temperatura de una máquina ha subido mucho en relación con otros meses, este mantenimiento lo detecta y gestiona alguna parada del activo para realizar el mantenimiento. Es un ejemplo muy simple, pero, en cualquier caso, se añaden distintas condiciones para saber que si un parámetro sobrepasa algún valor hay que ejecutar el mantenimiento. Se pueden añadir distintas condiciones al mismo tiempo.

Mantenimiento predictivo

El mantenimiento predictivo se basa en predecir los fallos que van a ocurrir y el momento en el que van a ocurrir para, de esta manera, hacer el mantenimiento justo antes de la posible avería. Trata de evitar mantenimientos innecesarios, y, por tanto, evita costes que pueden generar otros tipos de mantenimientos, como es el caso del mantenimiento preventivo. La desventaja que tiene este tipo de mantenimiento es la inversión de los equipos que realizan la captura de datos y el análisis de estos, ya que requieren de nuevas tecnologías y son más complejos. Estas son sus 3 componentes principales:

- Sensores y dispositivos conectados (*IoT*).
- Diferentes softwares para analizar datos (*Big Data*) y el almacenamiento en la Nube.
- Modelos predictivos que utilizan el *Machine Learning* para predecir posibles fallos con los datos que se sacan de los dispositivos.

La siguiente Ilustración muestra los momentos en los que se aplican los diferentes mantenimientos y las señales que se detectan:

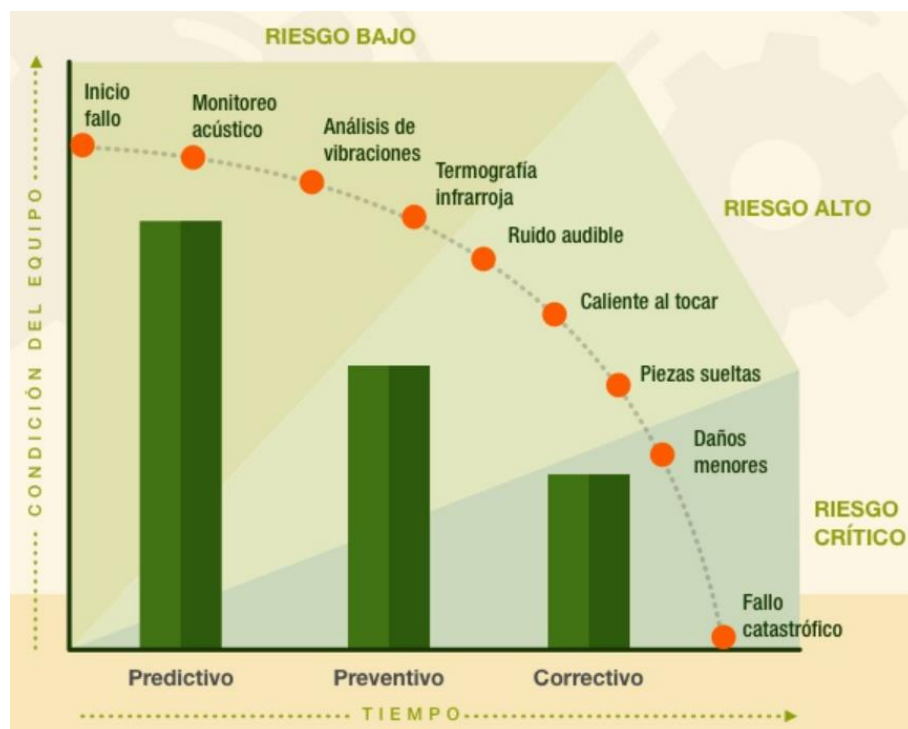


Ilustración 7: Comparación de los tipos de mantenimiento (6)

Niveles de Mantenimiento

Además de los tipos de mantenimiento que hay, las acciones que realiza el departamento de mantenimiento se pueden clasificar en 5 niveles dependiendo del grado de profundidad de la tarea realizada. Los niveles más bajos representan acciones más superficiales, como ajustes, y los más altos pueden llegar a ser tareas que necesiten de personal cualificado exterior.

A continuación, se muestran pequeños esquemas explicando brevemente cada nivel de mantenimiento:



Ilustración 8: Niveles 1, 2 y 3



Ilustración 9: Niveles 3 y 4

Objetivos

Como ya se ha comentado, el proyecto va a consistir en la creación de un sistema completo desde 0, que contenga activos y distintos elementos con sus propias órdenes de trabajo, *Escalations* y *Automation Scripts*. Todo esto encadenado con la aplicación que tiene el software sobre mantenimiento preventivo. Por tanto, los principales objetivos del proyecto serán los siguientes:

- Crear una Organización con todos los elementos que haga falta para poder automatizar órdenes de trabajo y mantenimiento preventivo. (*Storerooms, Assets, Items, Job Plans, Work Orders...*)
- Crear *Escalations* mediante condiciones (SQuirreL) para cambiar estados de las órdenes de trabajo de los *Assets* e *Items*.
- Crear *Automation Scripts* (Python/Jython) para generar las órdenes de trabajo y actualizar la información de estas. Los *Automation Scripts* son similares a las *Escalations*, pero sirven para gestionar temas algo más complejos que en las *Escalations* no se podrían gestionar de forma eficiente.
- Crear el Plan de Mantenimiento Preventivo (PMP) mediante frecuencias y medidas de los activos. En caso de que se tenga más tiempo, se podrían simular lecturas de distintos parámetros de los activos en cuestión, y que lleguen a Máximo mediante otro programa.
- Crear gráficos de los distintos parámetros de los activos para que el usuario pueda verlos en el *Start Center*.

Diseños conceptuales del sistema

Antes de comenzar a crear todo el sistema, se van a crear distintos esquemas conceptuales con todas las aplicaciones que se usarán del software e interconexiones que habrá entre estas. Además, se creará una lista con los *assets* e *items* que se van a crear y con todos los *job plans* (que generarán las *work orders*) que van a tener asociados. Por último, se diseñará el plan de mantenimiento preventivo de los *assets*.

Diseño de la Organización

La primera parte consiste en decidir qué elementos se van a añadir a la organización. En el siguiente esquema, se pueden observar todos ellos:

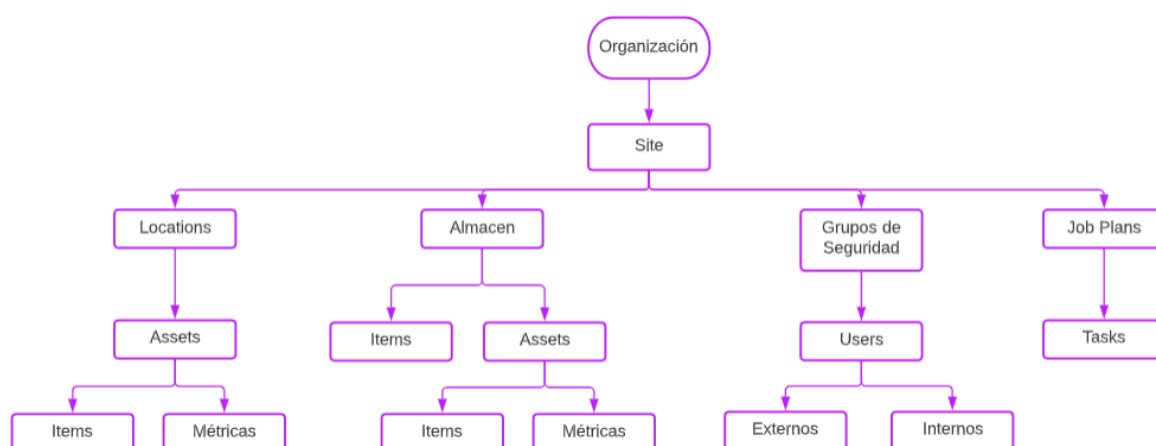


Ilustración 10: Esquema Organización

Por una parte, hay que crear la organización en sí y un *site* con sus elementos. Una organización puede contener diferentes *sites*. El site creado tendrá diferentes elementos básicos:

- **Locations:** Contendrán los diferentes *assets* según dónde opere cada uno de ellos. Además, se crearán métricas para estos para poder gestionar el plan de mantenimiento y las automatizaciones necesarias dependiendo de estas.
- **Almacen:** Contendrá los *items* que se van a gestionar en el proyecto. Desde el almacén se gestionará la compra de los *items* necesarios cuando queden pocos. Tendrá sus propios *assets* como ejemplo de gestión de *asset* propios. Además, los *assets* de los clientes pasarán por el almacén cuando haya que repararlos o hacerles un mantenimiento.
- **Grupos de Seguridad:** En Máximo esta parte es muy importante, ya que, cada Grupo de Seguridad contendrá diferentes usuarios y les dará distintos permisos según las necesidades de cada tipo de usuario. En este caso se creará un grupo de seguridad para los usuarios de la empresa y otro para los usuarios externos de esta, esto es, para sus clientes.
- **Job Plans:** Son muy importantes, ya que estos se asocian a las *Escalations* para crear diferentes órdenes de trabajo. Pueden contener distintas tareas.

Esta es la parte más genérica del proyecto. Para que este se entienda mejor, se creará todo el sistema en base a un ejemplo. En este caso, se va a simular un taller de coches, donde el almacén será el taller, y las distintas *locations* serán los lugares donde se utilizan los coches o motos cuyo mantenimiento es gestionado por el taller. Además, el propio taller tendrá que gestionar sus propios *assets* e *items*. Se escogerán diferentes partes de un vehículo para utilizarlos como ejemplo en este proyecto.

Locations

Como se ha explicado, serán las zonas de operación de los *assets* de los clientes del taller y se crearán el mismo número de *Locations* que número de usuarios que tengan un vehículo que gestione el taller. A cada una se le asociará uno de los vehículos.

Almacén

Como ya se ha comentado, el almacén simulará el taller de coches donde se realizarán las distintas operaciones de los *assets*. En el taller habrá distintos tipos de *items* como ruedas de coches, u otros elementos. Además, el taller tendrá 2 máquinas a las que se les creará un plan de mantenimiento preventivo como ejemplo. Se ha decidido utilizar los siguientes items como ejemplo:

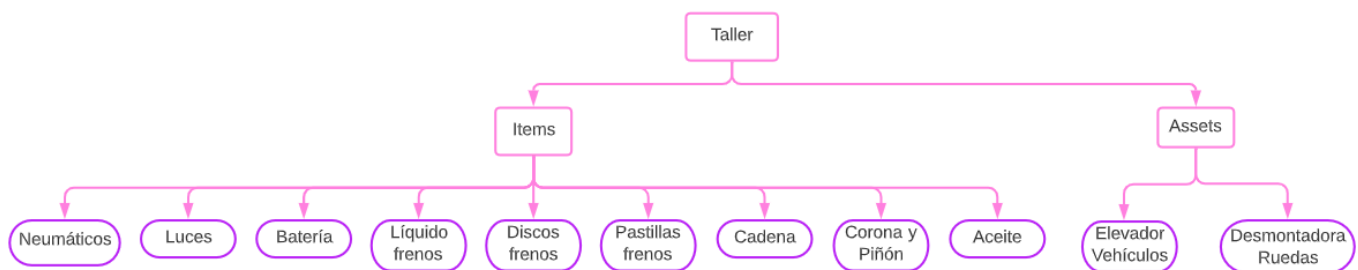


Ilustración 11: Esquema taller

Estos *items* también formarán parte de los *assets* de los clientes, en la sección de usuarios, *assets* e *items* se explica en detalle la subestructura que tendrán los coches y la moto que el taller va a gestionar. Por ello, estos *items* van a estar en el taller, por si se necesitan como reemplazos en cualquier mantenimiento que se les haga a los *assets* de los clientes.

A continuación, se muestra una tabla con las cantidades de cada tipo de item que se van a crear en Máximo, además de tener en cuenta que los items, en general, se han duplicado, ya que para cada tipo habrá de moto y de coche:

Tipo	Items/Assets	Cantidad
<i>Items de tipo COCHE</i>	Neumáticos	20
	Luces	30
	Batería	10
	Líquido de frenos*	20
	Discos de freno	20
	Pastillas de freno	20
<i>Items de tipo MOTO</i>	Neumáticos	10
	Luces	10
	Líquido de frenos*	10
	Discos de freno	10
	Pastillas de freno	10
	Cadena	10
	Corona y Piñón	10
	Aceite*	5
Assets del TALLER	Elevador de Vehículos	1
	Desmontadora de ruedas	1

Tabla 2: Items y Assets en el TALLER

*Para los items que son líquidos: 1 unidad de líquido de frenos equivale a 1L, mientras que 1 unidad de aceite equivale a 5L.

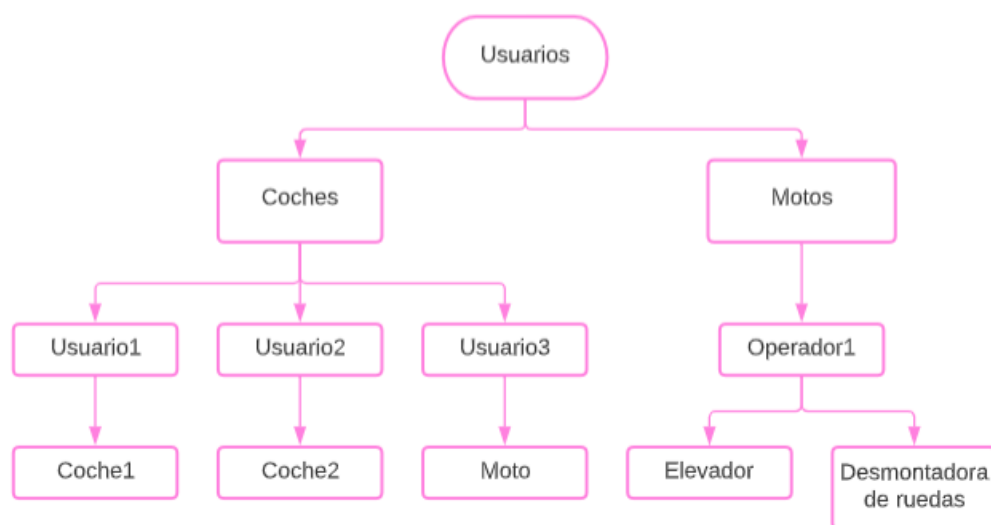
Grupos de Seguridad

Habr  2 Grupos de Seguridad, uno para el operario que gestiona el taller y el mantenimiento, tanto de los assets del taller como de los assets de los clientes, que tendr n permiso para crear y ver Work Orders, Purchase Orders y ver las Service Request de los clientes. Habr  un segundo grupo para los clientes del taller que quieran crear distintas *Service Request* para que el operario del taller las gestione. Este segundo grupo estar  formado por 3 usuarios.

Usuarios, assets e items

Habr  un operador que gestione el taller y los distintos items, *assets* y veh culos de los clientes. Habr  3 usuarios con veh culos, 2 de ellos tendr n un coche y el tercero una moto. En el taller, como ya se ha comentado, habr  2 assets permanentes, en este caso un elevador de coches y una desmontadora de ruedas, para que sirvan como ejemplo de assets propios de una empresa a los que se les crea un plan de mantenimiento preventivo. Adem s, estar n los distintos items que se necesitar n para distintas reparaciones o recambios. Se asumir  que el taller tendr  otras m quinas o dispositivos, que no se gestionar n en este proyecto, para realizar distintas reparaciones o an lisis de los veh culos de los clientes del taller.

Este es el esquema principal de los usuarios y sus *assets*:



Ilustraci n 12: Esquema usuarios y assets

Cada coche o moto tendr  una subestructura creada con diferentes *items*, que estar n en el taller tambi n, como por ejemplo las ruedas de los veh culos. Los 2 coches y la moto ser n los *assets* padres de sus distintos elementos, y estos elementos ser n los que contengan las m tricas para realizar las reparaciones o mantenimientos. En este proyecto se han escogido varias partes de los veh culos como ejemplo, se asume que hay muchos m s elementos que deben ser revisados en un taller, pero se considera que los elementos escogidos sirven como ejemplo suficiente para este proyecto. Estos son los esquemas de un coche y una moto y su estructura:

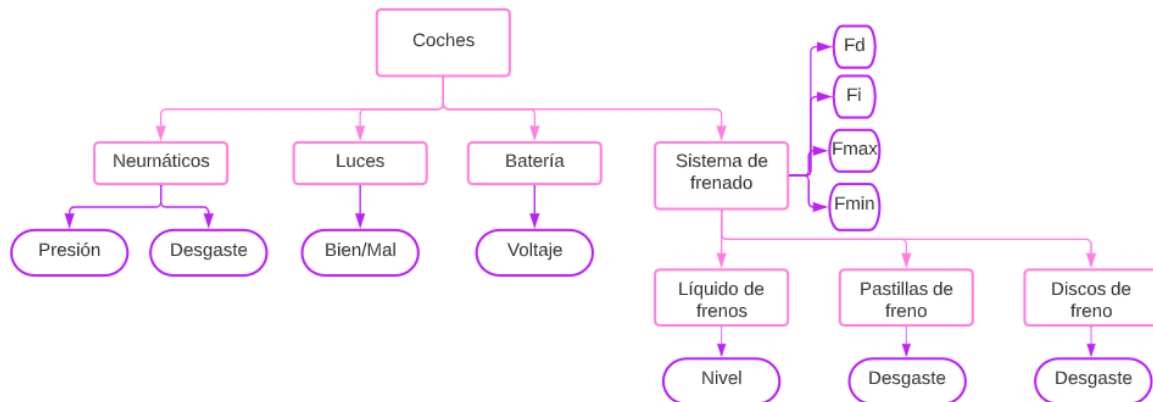


Ilustración 13: Esquema coche

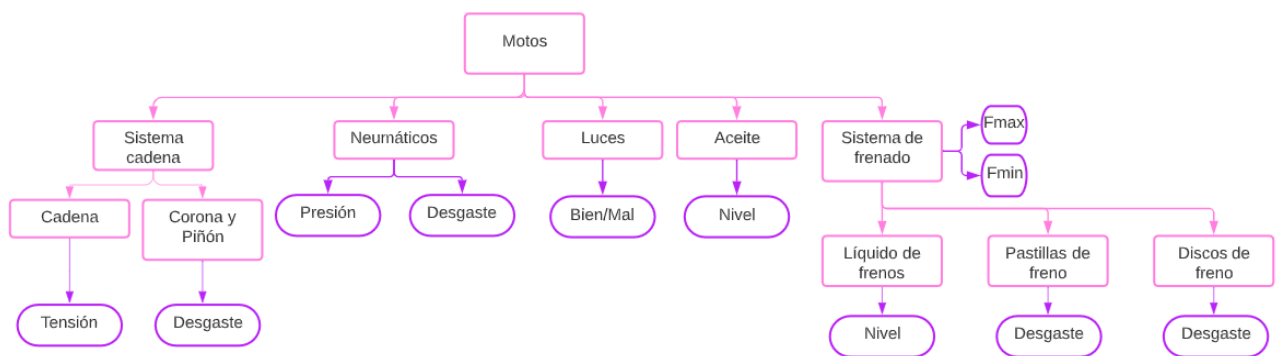


Ilustración 14: Esquema moto

Para cada tipo de vehículo y sus elementos hay distintas métricas. Se va a explicar cada una de ellas y el tipo de métrica que se ha escogido.

Hay 3 tipos de métricas, las de tipo *continuous*, *gauge* y *characteristics*. El primer tipo sirve para métricas numéricas, como puede ser el kilometraje del vehículo. El segundo tipo sirve para métricas como el nivel de aceite, por ejemplo. El último simplemente se basa en añadir características, como, por ejemplo, si algo está bien o mal, o el grado de defecto que tiene un elemento.

A continuación, se añade una tabla explicando cada métrica y su utilización:

Métrica	Descripción	Tipo de métrica	Utilización
KM	Kilometraje Vehículo	<i>Continuous</i>	Se va a utilizar como parámetro para el Mantenimiento Preventivo de los vehículos.
P	Presión Neumáticos	<i>Continuous</i>	Se va a utilizar para saber si las ruedas del vehículo necesitan ser hinchadas o no.
V	Voltaje Batería	<i>Continuous</i>	Se va a utilizar para saber si la batería del coche tiene buen funcionamiento o hay que cambiarla.
N	Nivel Líquido	<i>Gauge</i>	Se va a utilizar para saber si el nivel de aceite en las motos y el nivel del líquido de frenos en todos los vehículos es el correcto o no.
DES	Desgaste Piezas	<i>Characteristic</i>	Se va a utilizar para saber si las distintas piezas de los vehículos están desgastadas y hay que cambiarlas o no.
L	Luminosidad	<i>Characteristic</i>	Se va a utilizar para saber si las luces de los vehículos están en buenas condiciones o no y si hay que cambiarlas.
T	Tensión Cadena	<i>Characteristic</i>	Se va a utilizar para saber si la cadena de la moto está lo suficientemente tensa o si hay que reajustarla.
F_i	Fuerza Freno Izquierdo	<i>Continuous</i>	Se utilizarán para calcular las últimas 3 métricas de esta tabla, que se explican más adelante, para saber si el sistema de frenado funciona correctamente. Las métricas F_i y F_d se utilizarán solo en el caso de los coches.
F_d	Fuerza Freno Derecho	<i>Continuous</i>	
F_{max}	Fuerza Máxima Freno	<i>Continuous</i>	
F_{min}	Fuerza Mínima Freno	<i>Continuous</i>	
E	Eficacia Sistema de Frenos	<i>Characteristic</i>	Mide si los frenos en su conjunto son eficaces o no. Según el resultado de la fórmula se considerará un resultado normal, con defecto leve, o con defecto grave.
D	Desequilibrio Sistema de Frenos	<i>Characteristic</i>	En el caso de los coches, mide si los frenos izquierdos y los derechos tienen desequilibrios. Dependiendo del resultado se considerará un valor normal, defecto leve, defecto grave y defecto muy grave.
O	Ovalidad Sistema de Frenos	<i>Characteristic</i>	Mide si los frenos de cada rueda tienen un valor normal o no de ovalidad. Se considerará un resultado normal, defecto leve y defecto grave.

Tabla 3: Métricas

Job Plans

Se crearán diferentes *Job Plans* que contengan varias tareas para ejecutar. Además, se crearán también el mismo número de clasificaciones que de *Job Plans*, una para cada *Job Plan*, ya que servirán para asociar estos a las *Escalations* y que se generen las órdenes de trabajo automáticamente. Habrá 2 *Job Plans* principales, el primero será un plan que revise todos los elementos del coche, y el segundo los de la moto. Los demás *Job Plans* serán específicos para cada elemento del coche o de la moto. Además, habrá otros 2 más para revisar las máquinas del taller y repararlas si necesario. Este es el esquema:

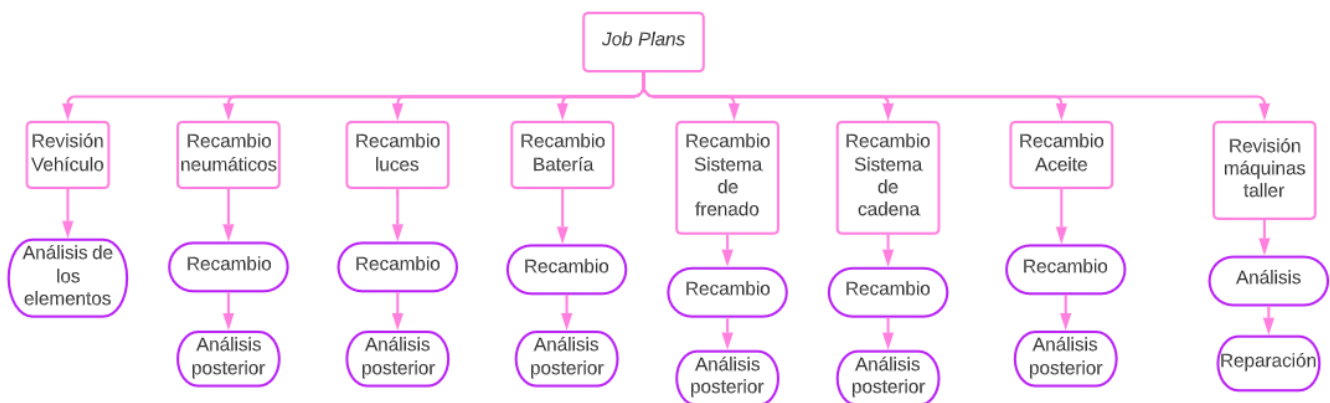


Ilustración 15: Esquema Job Plans

Estos son los Job Plans con sus tareas en detalle:

1. Revisión Coche:

- Neumáticos:** se hará una revisión de los neumáticos para ver si la presión es correcta y si están muy desgastados. En caso de que alguna de estas métricas no sea correcta, se activará el *Job Plan* que repare este elemento.
- Luces:** se revisará si las luces del coche están en buenas condiciones o no. Al igual que con los neumáticos, se activará el *Job Plan* para reemplazar las luces por unas nuevas.
- Batería:** se analizará si el voltaje de la batería es correcto. De no ser así se activará el *Job Plan* para cambiar la batería.
- Sistema de frenado:** se hará una revisión de este sistema, comenzando por el análisis del nivel del líquido de frenos, y siguiendo por el desgaste de los discos de freno y las pastillas de freno. En caso de que alguno de estos elementos no esté en buenas condiciones, se activará el *Job Plan* para arreglarlo. Además de este análisis, se tomarán medidas como las fuerzas máximas y mínimas del sistema, fuerzas de las ruedas izquierdas y derechas, y se calcularán distintos parámetros para realizar un análisis en mayor profundidad de este sistema:

- **Eficacia:** $E = \frac{F}{M.M.A * g} * 100$, donde F es la suma de las fuerzas, y M.M.A. el peso del vehículo.

Los valores normales de este parámetro son:

- **58-100%:** es el mínimo valor aceptado. Un valor igual o mayor se considera un valor normal, y, por tanto, no es necesario intervenir.
- **29-58%:** se considera un defecto grave. Habrá que intervenir en el sistema de frenado y cambiar alguna pieza o todas.
- **0- 29%:** se considera defecto muy grave. En este caso, habrá que cambiar todo el sistema de frenado.

- **Desequilibrio:** $D = \frac{100*(F_d - F_i)}{F_d}$, donde F_d es la fuerza de la rueda derecha y F_i de la izquierda.

Los valores normales de este parámetro son:

- **0-20%:** no se considera defecto ni es necesario intervenir en el sistema de frenado.
- **20-30%:** se considera defecto leve y se aconseja revisar el sistema de frenado para ver qué origina este desequilibrio.
- **30-80%:** se considera defecto grave. Es necesario revisar el sistema de frenado y cambiar alguna de las partes.
- **80-100%:** se considera defecto muy grave. El sistema entero ha de ser cambiado.

- **Ovalidad:** $d = \frac{100*(F_{max} - F_{min})}{F_{max}}$

Los valores normales de este parámetro son:

- **0-30%:** la ovalidad de esa rueda es mínima. No es necesario hacer nada.
- **30-55%:** se considera defecto leve. Habrá que volver a revisar bien el sistema para saber qué está produciendo esta ovalidad.
- **55-100%:** se considera defecto grave. Habrá que cambiar el sistema de frenado al completo.

2. Revisión Moto:

- Neumáticos:** se hará una revisión de los neumáticos para ver si la presión es correcta y si están muy desgastados. En caso de que alguna de estas métricas no sea correcta, se activará el *Job Plan* que repare este elemento.
- Luces:** se revisará si las luces de la moto están en buenas condiciones o no. Al igual que con los neumáticos, se activará el *Job Plan* para reemplazar las luces por unas nuevas.

- c. **Aceite:** se analizará si el nivel de aceite de la moto es el correcto. De no ser así se activará el *Job Plan* para añadir el aceite necesario a la moto.
- d. **Sistema de frenado:** se hará una revisión de este sistema, comenzando por el análisis del nivel del líquido de frenos, y siguiendo por el desgaste de los discos de freno y las pastillas de freno. En caso de que alguno de estos elementos no esté en buenas condiciones, se activará el *Job Plan* para arreglarlo. Además de este análisis, se tomarán medidas como las fuerzas máximas y mínimas del sistema, y se calcularán distintos parámetros para realizar un análisis en mayor profundidad de este sistema:

- **Eficacia:** $E = \frac{F}{M.M.A * g} * 100$, donde F es la suma de las fuerzas, y M.M.A. el peso del vehículo.

Los valores normales de este parámetro son:

- **58-100%:** es el mínimo valor aceptado. Un valor igual o mayor se considera un valor normal, y, por tanto, no es necesario intervenir.
- **29-58%:** se considera un defecto grave. Habrá que intervenir en el sistema de frenado y cambiar alguna pieza o todas.
- **0- 29%:** se considera defecto muy grave. En este caso, habrá que cambiar todo el sistema de frenado.

- **Ovalidad:** $d = \frac{100 * (F_{max} - F_{min})}{F_{max}}$

Los valores normales de este parámetro son:

- **0-30%:** la ovalidad de esa rueda es mínima. No es necesario hacer nada.
- **30-55%:** se considera defecto leve. Habrá que volver a revisar bien el sistema para saber qué está produciendo esta ovalidad.
- **55-100%:** se considera defecto grave. Habrá que cambiar el sistema de frenado al completo.

- e. **Sistema de cadena:** se analizará si la cadena tiene la tensión suficiente y si la corona y el piñón están muy desgastados o no. En caso de que alguno falle, se ejecutará el *Job Plan* para arreglar este sistema.

3. Recambio Neumáticos:

- a. **Cambiar:** se cambiarán las ruedas si el desgaste de estas es grave.
- b. **Comprobación:** Se comprobará que las ruedas funcionan correctamente una vez cambiadas.

4. **Hinchar Neumáticos:**

- a. **Hinchar:** se hincharán las ruedas si la presión de estas es baja.
- b. **Comprobación:** Se comprobará que las ruedas funcionan correctamente una vez cambiadas.

5. **Recambio Luces:**

- a. **Cambiar:** se cambiarán las luces que se considere que ya no tienen un correcto funcionamiento.
- b. **Comprobación:** Se comprobará que estas funcionan correctamente una vez cambiadas.

6. **Recambio Batería:**

- a. **Cambiar:** se cambiará la batería en caso de no tener un correcto funcionamiento.
- b. **Comprobación:** Se comprobará que la batería funciona correctamente una vez cambiada.

7. **Atender Estado Batería:**

- a. **Analizar:** se analizará el estado de la batería.
- b. **Reparación:** Se cargará o reparará si es necesario.

8. **Recambio Aceite:**

- a. **Añadir:** se añadirá aceite si se considera que el depósito tiene un nivel bajo.
- b. **Comprobación:** Se comprobará que el coche funciona correctamente con el nuevo nivel de aceite.

9. **Recambio Sistema de frenado:**

- a. **Cambiar:** se añadirá líquido de frenos si es necesario, y se cambiarán los discos y pastillas en caso de que estos estén desgastados. Si la eficacia, desequilibrio u ovalidad tienen valores extraños, se cambiará todo el sistema, aunque a simple vista parezca que se encuentra en buen estado.
- b. **Comprobación:** Se comprobará que el sistema funciona correctamente una vez cambiado todo.

10. **Recambio Sistema de cadena:**

- a. **Cambiar:** se cambiarán la corona y el piñón en caso de que estén desgastados.
- b. **Limpiar:** se limpiarán bien todas las partes del sistema.
- c. **Engrasar:** se engrasará la cadena para su correcto funcionamiento.
- d. **Tensar:** se tensará la cadena para acabar el mantenimiento de este sistema.
- e. **Comprobación:** Se comprobará que el sistema en su conjunto funciona correctamente.

11. Elevador de Vehículos:

- a. **Análisis:** se analizará el buen funcionamiento del elevador de vehículos.
- b. **Reparación:** En caso de que el análisis haya sido desfavorable, se reparará la máquina.
- c. **Comprobación:** Una vez reparada, se analizará otra vez para saber si funciona correctamente o no. En caso de que siga sin funcionar habrá que desechar la máquina vieja y comprar una nueva.

12. Desmontadora de Ruedas:

- a. **Análisis:** se analizará el buen funcionamiento de la desmontadora de ruedas.
- b. **Reparación:** En caso de que el análisis haya sido desfavorable, se reparará la máquina.
- c. **Comprobación:** Una vez reparada, se analizará otra vez para saber si funciona correctamente o no. En caso de que siga sin funcionar habrá que desechar la máquina vieja y comprar una nueva.

Diseño del Plan de Mantenimiento Preventivo

Una vez diseñado el sistema completo, se pretende dar una idea más general sobre cómo va a funcionar el mantenimiento de los distintos *assets* que se han creado.

Para ejecutar el plan de mantenimiento no sólo se va a utilizar la aplicación que tiene IBM Máximo de Mantenimiento Preventivo. En este plan se incluyen ciertos automatismos para los cuales se utilizarán 2 aplicaciones distintas, la aplicación *Escalations* y la aplicación *Automation Scripts*.

La aplicación de Mantenimiento Preventivo se utilizará tanto para los coches y la moto como para las 2 máquinas del taller. Esta aplicación generará automáticamente órdenes de trabajo para que el operario sepa cuando tiene que hacer una revisión o reparación. Para que se genere una orden de trabajo en Máximo para realizar el mantenimiento de cualquier *asset*, se utilizarán tanto frecuencias como métricas.

Frecuencias

Se utilizarán frecuencias para que cada 2 años los coches y la moto tengan que ir al taller para realizar el mantenimiento. En cuanto a las máquinas del taller, cada año se generará una orden de trabajo para hacerles el mantenimiento.

Métricas

Las métricas solamente se van a utilizar para los coches y la moto. En este caso, se utilizará el kilometraje para generar la orden de trabajo del mantenimiento. Para el coche, cada 15.000km se generará la orden, y para la moto cada 5.000km.

Este es el esquema de los Mantenimientos Preventivos que se van a crear en la aplicación de Máximo, con los Job Plans que van a tener asociados:

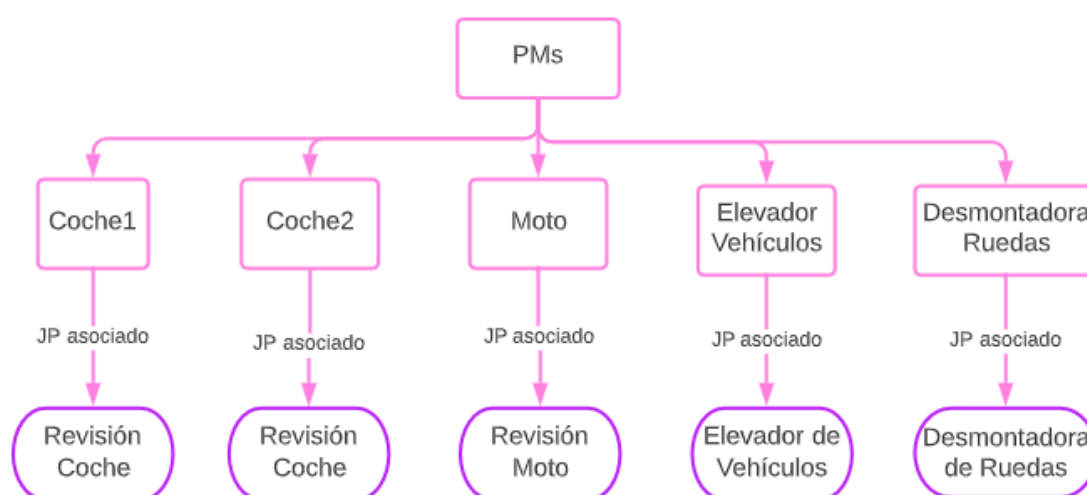


Ilustración 16: PMs

Automatismos de los *Automation Scripts*

Esta aplicación utiliza el lenguaje Python para crear automatismos. Al fin y al cabo, es crear un programa. En este proyecto se va a utilizar para gestionar todo el mantenimiento de los 2 coches y de la moto.

La orden de trabajo que se va a generar desde la aplicación de mantenimiento preventivo dependerá de si el asset es un coche o una moto. Cada cual tiene su Job Plan asociado, y, por tanto, el operador recibirá distintas órdenes de trabajo para hacer las revisiones de los *assets*.

Cuando este haga la revisión, tendrá que ir añadiendo los valores de las métricas, para que así, el programa, genere o no distintas órdenes de trabajo para reparar o cambiar elementos de los *assets*. Esto es lo que va a automatizar esta aplicación. Se va a crear una nueva aplicación en Máximo para que al operador le salgan en pantalla los valores que ha de meter en el programa dependiendo de si es un coche o una moto. El programa cogerá esos valores y los comparará con los valores normales de operación de los *assets*. En el caso del sistema de frenado, el programa calculará mediante las fórmulas los valores de eficacia, desequilibrio y ovalidad, una vez que el operador meta los valores de las fuerzas de frenado que se requieren.

En el caso de que alguno de los valores de las métricas no sea el adecuado, el programa generará, a través de los Job Plans ya creados, distintas órdenes de trabajo para que el operador repare o cambie los elementos de los *assets* en mal estado.

Primero se meterán todos los valores requeridos, el operador seguirá el Job Plan de revisión del coche o moto hasta acabarlo, y, una vez que todos los valores requeridos estén en el programa, este creará las órdenes de trabajo necesarias.

Diseño de la nueva aplicación para el operador

Como ya se ha comentado, se va a crear una nueva aplicación en Máximo para que el operador pueda introducir todos los valores de las métricas necesarias dependiendo de si hace la revisión de un coche o de una moto.

Para crear esta nueva aplicación, primero se va a crear una nueva tabla en la base de datos de Máximo. Para esto, se va a utilizar la aplicación de *Database Configuration*.

Nueva tabla

Esta nueva tabla de la base de datos de Máximo va a contener todas las métricas necesarias e información extra para después utilizarla en el programa que se va a crear en la aplicación de *Automation Scripts*. Estos serán los campos que se van a crear:

Campos	Utilidad
Usuario	Estos 7 campos servirán para llevar un buen control sobre cada revisión que se haga. Podría ocurrir que un cliente tuviera más de un <i>asset</i> , que hubiera más de un operador, más de un site...
Cliente	
<i>Asset</i>	
<i>Site</i>	
Fecha	
<i>Status</i>	
<i>Work Order</i>	
Presión Neumáticos	Revisión de los Neumáticos
Desgaste Neumáticos	
Luminosidad	Revisión de las Luces
Voltaje Batería	Revisión de la Batería (sólo para los coches)
Nivel Aceite	Revisión del Aceite (sólo para las motos)
Nivel Líquido Frenos	Revisión del Sistema de Frenado
Desgaste Discos	
Desgaste Pastillas	
F_d	Fuerzas para calcular eficacia, desequilibrio y ovalidad
F_i	
F_{max}	
F_{min}	
Tensión Cadena	Revisión del Sistema de Cadena (sólo para motos)
Desgaste Corona y Piñón	

Tabla 4: Nuevos campos

Es importante saber que, el operador, rellenará a mano las medidas, pero no los campos de información que se utilizan para llevar un buen control sobre las revisiones de los vehículos. Por ello, se ha añadido el campo *Work Order*, donde el operador añadirá el número de *Work Order* que se ha generado desde la aplicación de Mantenimiento Preventivo, y mediante otro *Automation Script*, pasará toda la información que hay en la *Work Order* a esta nueva aplicación automáticamente.

Mockup de la aplicación

Para diseñar la pantalla que el operador va a ver y va a tener que completar, se ha hecho un mockup en Excel. Este es el resultado final de cómo debería quedar la pantalla de la nueva aplicación:

Revisión Vehículo

Operador

Find Navigation Item

Go To Applications

Available Queries

Common Actions

New

Save

Clear Changes

Change Status

More Actions

List View

Revisión y Parámetros

Usuario: Site: Fecha: Status:

Cliente: Asset: Work Order:

Tarea	Medidas
Revisión Neumáticos	Presión: Desgaste:
Revisión Luces	Luminosidad:
Revisión Batería	Voltaje:
Revisión Aceite	Nivel Aceite:
Revisión Sistema de Frenado	Nivel Líquido Frenos: Desgaste Discos: Desgaste Pastillas:
	Fi: Fd: Fmax: Fmin:

Ilustración 17: Pantalla nueva Aplicación

Por una parte, está la zona de la izquierda, donde se pueden realizar distintas operaciones. En el caso de esta aplicación se podrán hacer las básicas, guardar la revisión del vehículo, crear una nueva, borrarla y cambiar el estado de la misma de *new* a *in progress* y finalmente a *resolved*:

Find Navigation Item

Go To Applications

Available Queries

Common Actions

New

Save

Clear Changes

Change Status

More Actions

Ilustración 18: Acciones de la ap

Por otra parte, el operador, tendrá que rellenar la siguiente información necesaria:

The screenshot shows a software interface with a top navigation bar. On the left, there is a 'List View' button. The main header is 'Revisión y Parámetros'. Below the header, there are seven input fields arranged in two rows. The first row contains 'Usuario:', 'Site:', 'Fecha:', and 'Status:'. The second row contains 'Cliente:', 'Asset:', and 'Work Order:'. Each field has a text input area and a blue arrow icon to its right, indicating a dropdown or search function.

Ilustración 19: Información a rellenar

Finalmente, en la zona de debajo a la izquierda se indican las tareas a realizar, y a la derecha las medidas que el operador tendrá que rellenar:

The screenshot shows a software interface with a list of tasks. The list is contained within a light gray box. The first item in the list is 'Revisión Neumáticos'. Below it are several empty rows, followed by 'Revisión Luces'. Below that are more empty rows, followed by 'Revisión Batería'. Below that are more empty rows, followed by 'Revisión Aceite'. Below that are more empty rows, followed by 'Revisión Sistema de Frenado'. Below that are more empty rows, followed by 'Revisión Sistema de Cadena'. Below that are more empty rows. The list is preceded by a header 'Tarea' in a blue font.

Ilustración 20: Tareas a realizar

Medidas			
Presión:	_____	Desgaste:	_____ >
Luminosidad:	_____ >		
Voltaje:	_____		
Nivel Aceite:	_____		
Nivel Líquido Frenos:	_____	Desgaste Discos:	_____ >
		Desgaste Pastillas:	_____ >
Fi:	_____	Fd:	_____
		Fmax:	_____
		Fmin:	_____
Tensión:	_____ >		Desgaste Corona y Piñón:
			_____ >

Ilustración 21: Medidas a rellenar

Las medidas a introducir por el operador en esta última parte dependerán de si el *asset* al que se le está haciendo la revisión es un coche o una moto. Habrá algunos campos, que, dependiendo del campo *asset*, saldrán en pantalla o no, como la revisión de la batería, del nivel de aceite o del sistema de cadena.

Estos son todos los elementos que se necesitan para crear esta nueva aplicación:

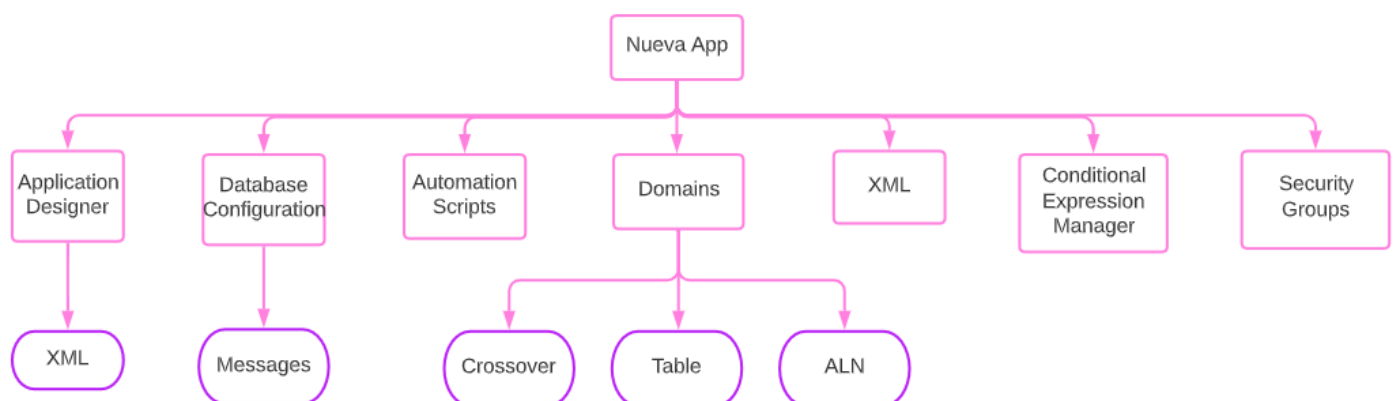


Ilustración 22: Elementos de la nueva App

Además de lo ya explicado anteriormente, se van a explicar otros elementos, que, aunque no sean tan importantes, serán muy necesarios para crear la nueva aplicación en Máximo.

- **Application Designer → XML:** Una vez que esté diseñada la aplicación, se utilizarán el fichero XML de esta para corregir los tamaños de las diferentes secciones y gestionar mejor la parte más visual de la app.
- **Database Configuration → Messages:** Esta es una de las opciones que tiene la aplicación *Database Configuration*. Con ella, se pueden crear mensajes de error para los campos de la nueva aplicación que se quieran. Después, se utilizará un *Automation Script* para automatizar que salgan estos mensajes de error si el operador introduce valores que no tengan sentido en los campos de las medidas.
- **Domains →** Esta aplicación sirve para que en la nueva aplicación, en algunos campos, el operador no tenga que escribir los datos, simplemente seleccionarlos desde una tabla, o, incluso, que se rellenen solos desde otra aplicación.
Hay 3 tipos de *Domains*: los de tipo ALN, TABLE y CROSSOVER. El primer tipo simplemente es crear una tabla añadiendo los elementos que se quieran meter en ella, para que el Operador escoja entre ellos para rellenar un campo. El segundo tipo es una tabla cuyos elementos se cogen desde otra aplicación de Máximo mediante una condición. Por ejemplo, la condición podría ser que el operador pueda rellenar un campo con los usuarios que estén contenidos en un *site* específico. El último tipo consiste en una pequeña automatización para que cuando se rellene un campo, como el número de *Work Order* por ejemplo, otros campos se rellenen automáticamente sacando la información desde la aplicación *Work Order*.
- **XML:** Se utilizará para crear los *lookups* que se quieran. Es algo muy similar a los *Domains*, pero se hace automáticamente desde un fichero XML, que después se añade a los campos que se han creado en el *Application Designer*.
- **Conditional Expression Manager:** Esta aplicación sirve para crear distintas condiciones. Generalmente, estas se añaden a los permisos que se les da a los distintos *Security Groups*. En este caso, se van a utilizar para, según el *asset*, mostrar u ocultar los campos de la aplicación que no se necesiten ver. Por ejemplo, si es un coche, no se mostrará la tarea ni de revisar el aceite ni de revisar el sistema de cadena.

Automatismos de las *Escalations*

Máquinas del taller

Como ya se ha comentado, las 2 máquinas que se han puesto de ejemplo como *assets* del taller, tendrán un plan de mantenimiento preventivo que creará una orden de trabajo para realizar el mantenimiento correspondiente. Se generarán desde los 2 *Job Plans* creados para esta labor. Sin embargo, se va a necesitar una *Escalation*, ya que, los *Job Plans*, tienen como tareas la de analizar las máquinas, repararlas y hacer una comprobación final. En el caso de que la revisión de las máquinas sea favorable, una *Escalation* hará que la orden de trabajo finalice una vez acabada la primera tarea, sin haber pasado por las otras 2. Si la revisión es desfavorable, el operador tendrá que realizar las demás tareas de la orden de trabajo.

Service Requests

Por otra parte, los clientes podrán crear *Service Requests* si sus coches o moto se averían. En este caso, el operador verá en pantalla la SR que han creado y actuará en consecuencia arreglando estos *assets*. La *Escalation* automatizará que se notifique a los clientes como va esta *Service Request*, ya que el operador cambiará el estado de esta de *new* a *in progress* cuando comience a trabajar en ello, y, finalmente, a *resolved* cuando acabe. Para ello, habrá que crear distintos puntos de escalabilidad para cada usuario, poniendo como condición que se notifique a la persona que ha abierto la *Service Request*. Además, habrá que crear los correos automáticos que se enviarán mediante la aplicación *Communication Templates*. Estos se enviarán una vez que el operador ha terminado con la tarea indicada en la SR y cambie el estado a *resolved*.

Start Center

Esta es una parte importante para todos los usuarios del sistema, ya que, será lo primero que vean al entrar en el programa. Verán tablas o gráficos de las aplicaciones a las que tengan permiso desde los *Security Groups*. Para ello, habrá que crear 2 *Templates* distintos para el operador del taller y para los usuarios del taller.

Operador

El operador tendrá que ver en pantalla todas las *Work Orders* que se le vayan creando automáticamente, para poder ejecutarlas en el taller. Además, podrá ver las nuevas revisiones de vehículos que se creen también automáticamente en la nueva aplicación para poder ejecutarlas y completarlas con los datos necesarios. Por último, también verá las *Service Request* de los distintos usuarios del taller. De estas 3 aplicaciones podrá ver los estados nuevos y en progreso, que es lo que le interesa al Operador, saber que le falta por hacer o incluso comenzar.

Usuarios

Los usuarios podrán ver sus propias *Service Request*, tanto las que estén en progreso y las que estén finalizadas, y, además verán los estados de las revisiones de sus coches que tengan esos 2 estados también.

Creación de la Organización y sus elementos

A continuación, se procede a crear la Organización, el *site* y todos los elementos que contiene este.

Organización

Antes de crear la Organización, ha sido necesario crear tanto el *item set* como el *company set* para asociarlos a la organización, y, después, poder utilizar estos *sets* para asociar distintos elementos a la organización directamente. Además, se ha establecido el euro como moneda principal para esta organización, aunque en este proyecto no se vaya a utilizar. Finalmente, se activa la Organización:

The screenshot displays the 'Organizations' application interface. The top navigation bar includes a home icon, a menu icon, and the title 'Organizations'. Below this is a search bar labeled 'Find Organization' and a toolbar with icons for search, dropdown, add, refresh, and navigation. The left sidebar contains sections for 'Go To Applications', 'Available Queries' (with 'All Records' and 'All Bookmarks'), 'Common Actions' (with 'New Organization', 'Save Organization', 'Clear Changes', and 'Create Report'), and 'More Actions' (with 'Work Order Options', 'Inventory Options', 'Crew Assignment Options', 'Asset Options', 'Drilldown Options', and 'DM Options'). The main content area is titled 'List View' and shows the 'Organization' tab. The organization details for 'COMILLAS' are displayed, including 'Comillas Talleres' as the name. The 'Base Currency 1' is set to 'EUR'. The 'Base Currency 2' is empty. The 'Item Set' is 'COM_ITEM', associated with 'Item Set Comillas'. The 'Company Set' is 'COM_COM', associated with 'Company Set Comillas'. The 'Default Item Status' is 'ACTIVE'. The 'Default Stock Category' is 'STK'. The 'Clearing Account' is '100-100-100-1'. The 'Active?' checkbox is checked.

Ilustración 23: Organización

Site

El site se crea en la misma aplicación que la organización. Como se puede ver, se pueden crear muchos sites dentro de esta, aunque en este proyecto se ha decidido crear solamente uno. Es importante saber que se podría haber creado un contrato de garantía en otra aplicación, y asociar el contrato con el site. Finalmente, se activa el site para poder utilizarlo:

The screenshot shows the 'Organizations' application interface. On the left is a sidebar with navigation options like 'Go To Applications', 'Available Queries', 'Common Actions', and 'More Actions'. The main area displays the 'Sites' tab for the 'COMILLAS' organization. It shows a table with columns: Program, Description, Site Contact, Site Contact Group, and Active?. A row is selected for 'ICAI' with description 'Taller ICAI'. Below the table is a 'Details' section with fields for Program, Site Code, Site Contact, Site Contact Group, Active? (checked), and Used? (unchecked). A 'New Row' button is at the bottom.

Ilustración 24: Site

Locations

Se crean 3 *Locations*, una para cada usuario, ya que sus *assets* se asociarán a cada una de ellas:

The screenshot shows the 'Locations' application interface. It displays a table with columns: Location, Description, Type, Status, Priority, and Program. There are 4 rows of data, all with 'OPERATING' status and 'ICAI' program.

Location	Description	Type	Status	Priority	Program
CLIENTE1	Lugar de operacion del coche 1	OPERATING	OPERATING		ICAI
CLIENTE2	Lugar de operaciòn del coche 2	OPERATING	OPERATING		ICAI
CLIENTE3	Lugar de operacion de la moto	OPERATING	OPERATING		ICAI

Ilustración 25: Locations

Métricas

Antes de comenzar a crear los *assets*, se crean las distintas métricas que se van a utilizar para medir y determinar los ajustes que hay que hacerles a los vehículos. Como ya se ha comentado en la parte de diseño, se van a crear 3 tipos de métricas. En la siguiente ilustración se observan todas ellas con su tipo de métrica:

<u>KM</u>	Kilometraje Vehículo	CONTINUOUS	+
<u>P</u>	Presión Neumáticos	CONTINUOUS	+
<u>V</u>	Voltaje batería	CONTINUOUS	+
<u>N</u>	Nivel líquido	GAUGE	+
<u>DES</u>	Desgaste piezas	CHARACTERISTIC	+
<u>L</u>	Luminosidad	CHARACTERISTIC	+
<u>I</u>	Tensión cadena	CHARACTERISTIC	+
<u>FI</u>	Fuerza freno izquierdo	CONTINUOUS	+
<u>FD</u>	Fuerza freno derecho	CONTINUOUS	+
<u>FMAX</u>	Fuerza máxima freno	CONTINUOUS	+
<u>FMIN</u>	Fuerza mínima freno	CONTINUOUS	+
<u>E</u>	Eficacia sistema de frenos	CHARACTERISTIC	+
<u>D</u>	Desequilibrio Sistema de frenos	CHARACTERISTIC	+
<u>O</u>	Ovalidad Sistema de frenos	CHARACTERISTIC	+

Ilustración 26: Métricas

Items

Los *items* sirven para 2 cosas, la primera es para utilizarlos como partes de recambio en el taller y meterlos en el *Storeroom* que simulará el taller. Además de esto, al crear cada *asset* la aplicación pide un *item* en el que basarse, y, por tanto, hay que crear los coches y moto también como *items* para poder crear los *assets* padre que serán estos vehículos.

Item Master		norasegura	
Find Item	Advanced Search	Save Query	Bookmarks
Find Navigation Item	Items	Filter	1 - 20 of 22
Go To Applications	Item	Description	Commodity Group Rotating? Kit? Condition Enabled? Status
Available Queries			ACTIVE
All Records	1011	Neumático Coche	ACTIVE
All Bookmarks	1013	Neumático Moto	ACTIVE
Common Actions	1014	Coche 1	ACTIVE
New Item	1015	Coche 2	ACTIVE
Change Status	1016	Moto	ACTIVE
Create KPI	1018	Luces Coche	ACTIVE
Create Report	1019	Luces Moto	ACTIVE
More Actions	1020	Batería Coche	ACTIVE
Add Items To Storeroom	1021	Líquido frenos Coche	ACTIVE
Unit of Measure and Conversion	1023	Discos freno Coche	ACTIVE
Add/Modify Commodity Codes	1024	Pastillas freno Coche	ACTIVE
Attachment Library/Folders			
Run Reports			
Cognos Analytics			

Ilustración 27: Items

Al crearlo, es muy importante poner un *Part Number* distinto para cada uno, ya que este campo se utiliza como id que define a los *items* en cada *site*, ICAI en este caso. Los items se relacionan con el *site* por medio del *item set* creado anteriormente.

The screenshot shows the 'Item Master' application interface. The 'Item' tab is selected, displaying details for Item 1011. The form includes the following fields and values:

- Item: 1011
- SAP Code: (empty)
- Description: Neumático Coche
- Details: (empty text area)
- Item Set: COM_ITEM
- Commodity Group: (empty)
- Commodity Code: (empty)
- Meter Group: (empty)
- Meter: (empty)
- Lot Type: NOLOT
- Maximum Quantity Issued: (empty)
- Order Unit: NR
- Issue Unit: NR
- Item Type: CI
- MSDS: (empty)
- Part Number: 1
- Status: ACTIVE
- Rotating?: ☒
- Condition Enabled?: ☐
- Kit?: ☐
- Capitalized?: ☐
- Inspect on Receipt?: ☐
- Add as Spare Part?: ☐
- Attach to Parent Asset on Issue?: ☐
- Manufacturer P/N: (empty)
- Supplier P/N: (empty)
- Supplier Description: (empty)
- All the Storerooms?: ☒
- Tax Exempt?: ☐
- CI: (empty)
- Item Nuc: (empty)

Ilustración 28: Ejemplo creación item

Assets

Para crear los vehículos, que son los *assets* principales de este proyecto, hay que crear cada elemento de estos, basándose primero en los *items* ya creados. Cada *asset* se asocia a una de las *Locations* de los usuarios, dependiendo de a qué vehículo pertenezca. Finalmente, se han creado 62 assets distintos:

The screenshot shows the 'Assets' application interface with a list of 62 assets. The table below represents the data shown in the screenshot:

Asset	Part Number	State of change	Serial #	Description	Location	PN Parent	Rotating Item	Ref.Des.	Structural Code	Is SES the owner?	Status	Program
1003	1	ND	ND	Rueda Coche	ICAI		1011			☒	NOT READY	ICAI
1004	1	ND	ND	Rueda Coche	CLIENTE1		1011			☒	NOT READY	ICAI
1006	1	ND	1	Rueda Coche	CLIENTE1	3	1011			☒	ACTIVE	ICAI
1007	1	ND	2	Rueda Coche	CLIENTE1	3	1011			☒	ACTIVE	ICAI
1008	1	ND	3	Rueda Coche	CLIENTE1	3	1011			☒	ACTIVE	ICAI
1009	1	ND	4	Rueda Coche	CLIENTE1	3	1011			☒	ACTIVE	ICAI
1010	1	ND	1	Rueda Coche	CLIENTE2	4	1011			☒	ACTIVE	ICAI
1011	1	ND	2	Rueda Coche	CLIENTE2	4	1011			☒	ACTIVE	ICAI
1012	1	ND	3	Rueda Coche	CLIENTE2	4	1011			☒	ACTIVE	ICAI
1013	1	ND	4	Rueda Coche	CLIENTE2	4	1011			☒	ACTIVE	ICAI
1014	2	ND	1	Rueda Moto	CLIENTE3	5	1013			☒	ACTIVE	ICAI
1015	2	ND	2	Rueda Moto	CLIENTE3	5	1013			☒	ACTIVE	ICAI
1017	3	ND	5	Coche 1	CLIENTE1		1014			☒	ACTIVE	ICAI
1018	4	ND	10	Coche 2	CLIENTE2		1015			☒	ACTIVE	ICAI
1019	6	ND	1	Luces Coche	CLIENTE1	3	1018			☒	ACTIVE	ICAI
1020	6	ND	2	Luces Coche	CLIENTE1	3	1018			☒	ACTIVE	ICAI
1021	6	ND	3	Luces Coche	CLIENTE1	3	1018			☒	ACTIVE	ICAI
1022	6	ND	4	Luces Coche	CLIENTE1	3	1018			☒	ACTIVE	ICAI
1023	6	ND	5	Luces Coche	CLIENTE1	3	1018			☒	ACTIVE	ICAI
1024	6	ND	1	Luces Coche	CLIENTE2	4	1018			☒	ACTIVE	ICAI
1025	6	ND	2	Luces Coche	CLIENTE2	4	1018			☒	ACTIVE	ICAI

Ilustración 29: Assets

Para crear cualquier *asset*, es importante añadir el *Serial Number*, que, al igual que pasaba con el *Part Number* de los *items*, es el id con el que identificar cada *asset* que se basa en uno de los *items*. Como ya se ha explicado en la parte de diseño de este proyecto, se van a crear los 3 vehículos basándose en una jerarquía de *assets*. La siguiente ilustración muestra uno de los elementos que tiene como *asset* padre el Coche1. Forma parte de la parte más baja de esta jerarquía:

The screenshot shows the 'Assets' application interface. The main form displays details for Asset 1006, 'Rueda Coche'. The form is organized into sections: 'Asset Details' (Asset: 1006, Status: ACTIVE, Part Number: 1, State of Change: ND, Nuc: , Type of Obsolescence: , Multiplicity: , Is MSI?), 'Asset Hierarchy' (Asset Parent: 1017 - Coche 1, Item Parent: 1014, Part Number Parent: 3, State of Change Parent: ND, Maintain Hierarchy?), 'Location' (Location: CLIENTE1 - Lugar de operacion del coche 1, Birc: , Rotating Item: 1011 - Neumático Coche, SAP Code: , Condition Code: , Meter Group:), 'Asset Information' (Program: ICAI, Asset Type: E, Hierarchical Level: , Manufacturer Part Number: , Alternative Part Number: , Alternative Description: , SN Manufacturer: , SN Supplier: , Infrastructure Category: , Asset Template:), 'Calendar' (Calendar: , Turn: , Priority:), 'Serial Number' (Serial Number: 1, Serialnumber Status: Not Verified, Failure Class: , Item Type: CI, Tool Rate: , CMB: , Ref Des.: , Structural Code:), and 'Attachments' (Attachments, Moved?, Returned To Vendor?).

Ilustración 30: Ejemplo creación asset 1

En este caso, el *asset* es una de las ruedas del Coche1, y, por tanto, tendrá 2 métricas distintas, la presión y el desgaste de los neumáticos. Estas 2 métricas ya creadas, se van a asociar al *asset* en la pestaña *Meters*. Esto se hará con todos los *assets* que se creen.

The screenshot shows the 'Meters' tab for Asset 1006, 'Rueda Coche'. The table lists two meters:

Sequence	Meter	Description	Meter Type	Unit of Measure	Active?	End Measure
1	P	Presión Neumáticos	CONTINUOUS	BAR	☒	
2	DES	Desgaste piezas	CHARACTERISTIC		☒	

Buttons: New Row, Filter, 1 - 2 of 2

Ilustración 31: Métricas del asset 1

El siguiente ejemplo de *asset* tiene también como padre *asset* el Coche1, pero, además, tiene varios hijos *assets*, que serán los componentes que forman todo el sistema de frenado del coche:

Asset	Description	Location	Description
1031	Líquido frenos Coche	CLIENTE1	Lugar de operacion del coche 1
1032	Discos freno Coche	CLIENTE1	Lugar de operacion del coche 1
1033	Discos freno Coche	CLIENTE1	Lugar de operacion del coche 1
1034	Discos freno Coche	CLIENTE1	Lugar de operacion del coche 1
1035	Discos freno Coche	CLIENTE1	Lugar de operacion del coche 1
1036	Pastillas freno Coche	CLIENTE1	Lugar de operacion del coche 1

Ilustración 32: Ejemplo creación asset 2

Además, el sistema de frenado en su conjunto contendrá las 4 métricas de fuerzas que el operador va a medir:

Sequence	Meter	Description	Meter Type	Unit of Measure	Active?	End Measure
1	F1	Fuerza freno izquierdo	CONTINUOUS KN		☒	
2	FD	Fuerza freno derecho	CONTINUOUS KN		☒	
3	FMAX	Fuerza máxima freno	CONTINUOUS KN		☒	
4	FMIN	Fuerza mínima freno	CONTINUOUS KN		☒	

Ilustración 33: Métricas del asset 2

Por último, se pone como ejemplo el Coche1 con todos sus subelementos. Este *asset* ya no forma parte de otro, y por tanto no tiene un *asset* padre:

Asset	Description	Location	Description
1006	Rueda Coche	CLIENTE1	Lugar de operacion del coche 1
1007	Rueda Coche	CLIENTE1	Lugar de operacion del coche 1
1008	Rueda Coche	CLIENTE1	Lugar de operacion del coche 1
1009	Rueda Coche	CLIENTE1	Lugar de operacion del coche 1
1019	Luces Coche	CLIENTE1	Lugar de operacion del coche 1
1020	Luces Coche	CLIENTE1	Lugar de operacion del coche 1

Ilustración 34: Ejemplo creación asset 3

Storeroom

Todos los items que se han creado se han añadido al Storeroom, aunque esto no significa que haya esos items en el Storeroom, ya que, una vez añadidos, se especifica la cantidad de cada uno de la que se dispone. En los casos en los que el item es un coche o una moto esta cantidad se pone a 0. Este es el Storeroom con sus *items*:

Itemnum	Partnumber	State of change	Description
1011	1	ND	Neumático Coche
1013	2	ND	Neumático Moto
1014	3	ND	Coche 1
1015	4	ND	Coche 2
1016	5	ND	Moto
1018	6	ND	Luces Coche
1019	7	ND	Luces Moto
1020	8	ND	Batería Coche
1021	9	ND	Líquido frenos Coche
1023	10	ND	Discos freno Coche
1024	11	ND	Pastillas freno Coche
1025	12	ND	Sistema de frenado Coche
1026	13	ND	Sistema de cadena Moto
1027	14	ND	Cadena Moto
1028	15	ND	Corona y Piñón Moto
1029	16	ND	Aceite Moto
1030	17	ND	Sistema de frenado Moto
1032	18	ND	Líquido frenos Moto
1033	19	ND	Pastillas freno Moto
1034	20	ND	Discos freno Moto

Ilustración 35: Storeroom con sus items

Además de los *items*, el *Storeroom* también contiene los 2 *assets* que habrá en el taller, el elevador de vehículos y la desmontadora de ruedas:

Itemnum	Assetnum	Partnumber	State of change	Serial Number	Description+
1036	1067	22	ND	1	Desmontadora de ruedas
1035	1066	21	ND	1	Elevador Vehiculos

Ilustración 36: Storeroom con sus assets

En la aplicación de *Inventory*, se puede especificar la cantidad de cada *item* que hay en el *Storeroom*. En la siguiente ilustración se ejemplifica esto con el *item* neumático, ya que habrá 20 inicialmente en el taller:

Item: 1011 Neumático Coche Program: ICAI

Storeroom: TALLER Taller de reparacion donde se encuentran las maquinas Status: ACTIVE

Lot Type: NOLOT Default Bin: 20

Issue Cost Type: STANDARD Default Stage Bin:

Receipt Tolerance %:

Capitalized? ☐ Kit? ☐

Ilustración 37: Item en Inventory

Job Plans

Una vez creados los distintos elementos, se crean los Job Plans en los que se basarán las órdenes de trabajo que se creen automáticamente mediante los distintos Scripts.

Job Plan	Description	Revision	Duration	Supervisor	Status	Template Type	Organization	Contract
2363	Revisión Coche	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2365	Revisión Moto	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2367	Cambiar Neumáticos	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2368	Hinchar Neumáticos	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2369	Cambiar Luces	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2370	Cambiar Batería	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2372	Cambiar Aceite	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2373	Cambiar/Reparar Sistema de Frenado	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2374	Cambiar/Reparar Sistema de Cadena	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2375	Mantenimiento Elevador de Vehículos	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2376	Mantenimiento Desmontadora de Ruedas	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	
2377	Atender Estado Batería	0	0:00		ACTIVE	MAINTENANCE	COMILLAS	

Ilustración 38: Job Plans

Se pone como ejemplo el primer Job Plan que es 'Revisión Coche'. Es de tipo mantenimiento, al igual que los demás:

The screenshot shows the 'Job Plans' application interface. On the left is a sidebar with navigation items like 'Go To Applications', 'Available Queries', 'All Records', 'All Bookmarks', 'Common Actions' (including 'New Job Plan', 'Save Job Plan', 'Clear Changes', 'Change Status', 'Create Report'), and 'More Actions' (including 'View Status History', 'View Costs', 'Attachment Library/Folders'). The main area has tabs for 'List View', 'Job Plan', 'Work Assets', and 'Specifications'. The 'Job Plan' tab is active, showing details for Job Plan 2363, 'Revisión Coche', Organization 'COMILLAS', and Contract. The 'Details' section includes fields for Status (ACTIVE), Template Type (Maintenance), Duration (0:00), Classification, Class Description, Default WO Class, WO Priority, Interruptible?, Interruptible shift, Flow Controlled?, Supervisor, Crew, and Lead. A 'Responsibility' tab is also visible.

Ilustración 39: Ejemplo creación de Job Plan

Para este mismo Job Plan, se añaden sus 4 tareas como se ve a continuación:

The screenshot shows the 'Job Plans' application interface with the 'Job Plan' tab active. It displays a list of tasks for Job Plan 2363, 'Revisión Coche'. The tasks are:

Sequence	Task	Description	Nested Job Plan	Duration	Meter
10	Revisión de Neumáticos			0:00	
20	Revisión de Luces			0:00	
30	Revisión de Batería			0:00	
40	Revisión de Sistema de Frenado			0:00	

Below the table is a 'Details' section for the selected task (Sequence 10, Task 'Revisión de Neumáticos'). It includes fields for Organization (COMILLAS), Program (ICAI), Task, Sequence, Nested Job Plan, Classification, Class Description, Duration (0:00), Meter, Owner, Owner Group, Crew Work Group, Flow Controlled?, and Suspend Flow Control?. There is also an 'Attachments' link.

Ilustración 40: Tareas Job Plan

Las 4 tareas serán los Job Plan hijos del Job Plan 'Revisión Coche'. Una vez creadas, se activa el Job Plan para poder utilizarlo en las distintas aplicaciones.

Security Groups

A continuación, se crean los usuarios y sus Security Groups. Esta parte es muy importante, ya que los usuarios tendrán distintos permisos y restricciones según estos grupos.

En la primera pestaña se añade el Template del Start Center que verá cada usuario al meterse en Máximo con su usuario. Esto se explicará más adelante.

Security Groups

Find Group

Find Navigation Item

Go To Applications

Available Queries

All Records

All Bookmarks

Common Actions

New Group

Go to Setup

Clear Changes

Create Report

More Actions

Hide Password Duration

Security Controls

Group: Clientes_ICAI **Clientes del taller**

Start Center Template: 115 Start Center Clientes ICAI

Independent of Other Groups? ☒

Display Side Navigation Menu? ☒

Default Application

① Users with exactly one Default Application defined according to their group membership will automatically launch to that application upon login.

Default Application:

Ilustración 41: Security Groups

En la pestaña de *Applications* se decide a qué aplicaciones se les da permisos y a cuales no. Además, en cada aplicación se les puede dar permisos para ejecutar distintas acciones según su grupo. En el caso del grupo de los clientes, se le da acceso completo a la aplicación para crear *Service Request* y acceso de lectura para la nueva aplicación que se va a crear que utilizará el operador. En cuanto al grupo del operador, se le da acceso a la aplicación de *Service Request* y se añade que pueda cambiar el estado de estas. También se le da acceso a las *Work Orders*, al inventario, a los *assets* e *items* y acceso completo a la nueva aplicación.

Security Groups

Find Group

Find Navigation Item

Go To Applications

Available Queries

All Records

All Bookmarks

Common Actions

New Group

Go to Setup

Clear Changes

Create Report

More Actions

Hide Password Duration

Security Controls

Authorize Group Reassignment

Global Data Restrictions

Standard Service Authorization

Duplicate Group

Delete Group

Add to Bookmarks

Grant Access to Multiple Catalogs

Revoke Access to Multiple Offerings

Run Reports

Cognos Analytics

Description	Application Type	Main Object/Table
service req		
Create Service Request	Application	The SR view
Create Service Request (Leonardo)	Application	The SR view
Create Service Request (SP)	Application	The SR view
Service Requests	Tools and Tasks	
Service Requests	Application	The SR view
Service Requests (HSE)	Application	The SR view
Service Requests (Leonardo)	Application	The SR view
Service Requests (SP)	Application	The SR view
View Service Requests	Application	The SR view
View Service Requests (Leonardo)	Application	The SR view

Options for Create Service Request (Leonardo) Filter 1 - 2 of 2

Grant Listed Options for This Application

Description	Grant Access?	Condition
> New Service Request	☒	>
> Read access to Create Service Request	☒	>

Filter 41 - 44 of 44

Description	Grant Access?	Condition
> Show auto classification attributes	☒	>
> Show auto classification attributes	☒	>
> Show the field PHASE	☒	>
> Show/Hide the Impact Field	☒	>

Ilustración 42: Ejemplo de permisos de los SG

En cuanto a la pestaña de *Data Restrictions*, simplemente se añaden los elementos con los que va a trabajar cada usuario y se añade la condición de que sean de su *site*, en este caso del *site* ICAI. A continuación, se observa el ejemplo del grupo del operador, que trabajará con *Work Orders*, *Service Request*, *Locations* y *Assets*:

Group: Clientes_ICAI Cientes del taller

Object Restrictions Attribute Restrictions Collection Restrictions

Objects	Filter	Type	Condition
ASSET	QUALIFIED	COND_ICAI	
LOCATIONS	QUALIFIED	COND_ICAI	
SR	QUALIFIED	COND_ICAI	
WORKORDER	QUALIFIED	COND_ICAI	

Details

Object: ASSET The ASSET Table

Application: Condition: COND_ICAI Condición de que sea Site ICAI

Type: QUALIFIED Condition Class: Expression: (SITEID=ICAI)

Reevaluate? ☒

New Row

Ilustración 43: Ejemplo Data Restrictions de los SG

Finalmente, se encuentra la pestaña de los *users* en la que habrá que añadir cada usuario que pertenezca a cada grupo. En la siguiente ilustración se muestran los usuarios del grupo de los clientes:

Group: Clientes_ICAI Cientes del taller

Users Filter 1 - 3 of 3

User	Person	Name	Status	Type
USUARIO1	USUARIO1	Usuario Coche 1	ACTIVE	SCCD-AUTH
USUARIO2	USUARIO2	Usuario Coche 2	ACTIVE	SCCD-AUTH
USUARIO3	USUARIO3	Usuario Moto	ACTIVE	SCCD-AUTH

Details

User: USUARIO1 First Name: Last Name: Usuario Coche 1

Person: USUARIO1 Type: SCCD-AUTH

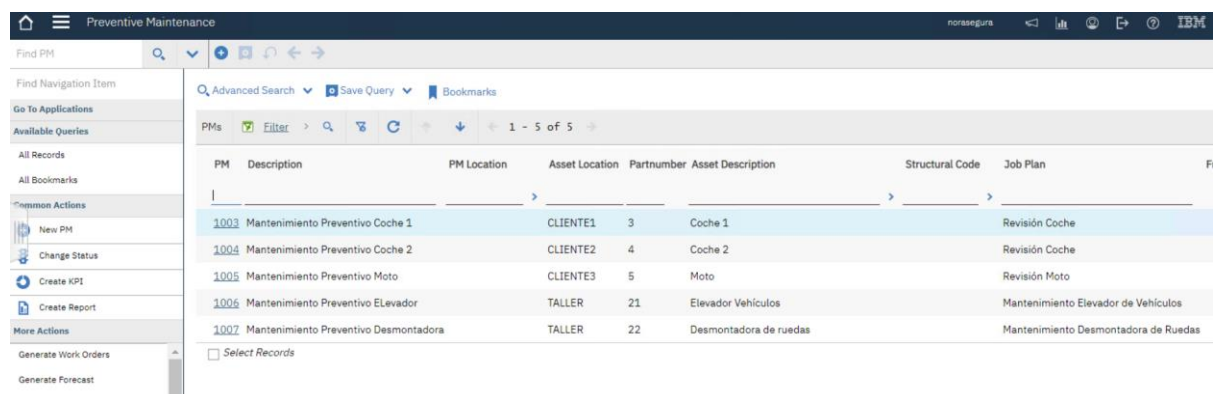
Name: Usuario Coche 1 Status: ACTIVE

Select Users New Row

Ilustración 44: Ejemplo de los usuarios del SG

Creación del Plan de Mantenimiento Preventivo

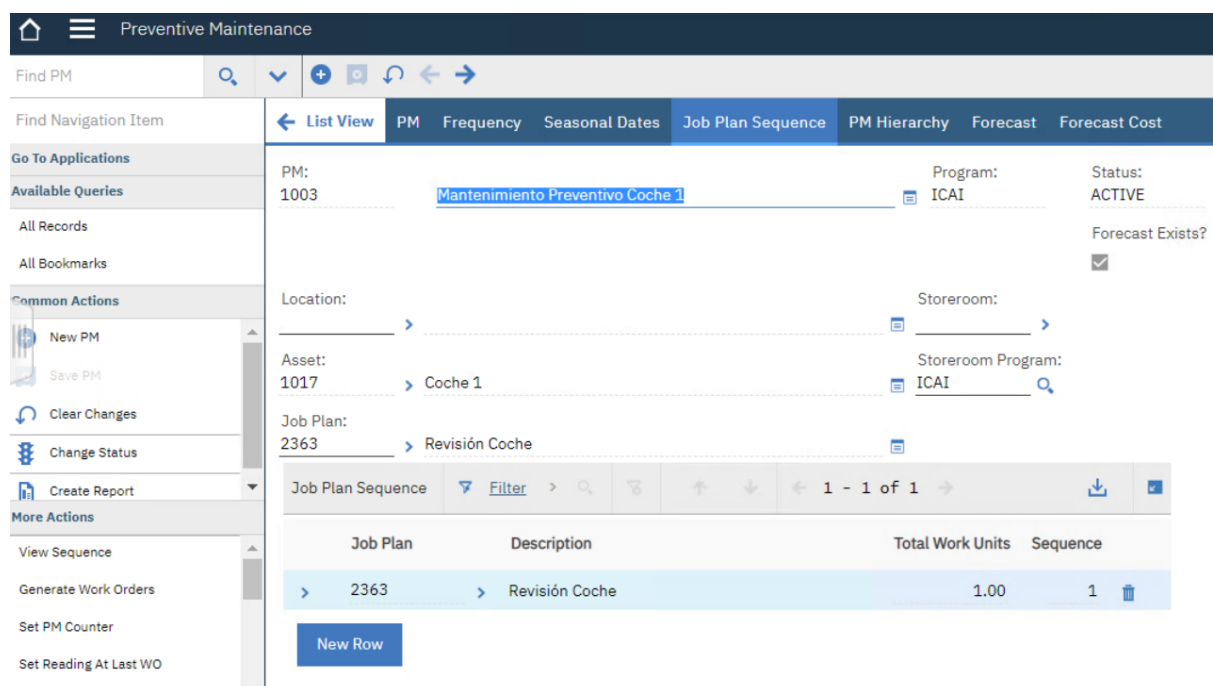
Como ya se ha comentado en la parte de diseño, se han creado 5 planes de mantenimiento preventivo para los 5 *assets* que se han creado:



PM	Description	PM Location	Asset Location	Partnumber	Asset Description	Structural Code	Job Plan
1003	Mantenimiento Preventivo Coche 1		CLIENTE1	3	Coche 1		Revisión Coche
1004	Mantenimiento Preventivo Coche 2		CLIENTE2	4	Coche 2		Revisión Coche
1005	Mantenimiento Preventivo Moto		CLIENTE3	5	Moto		Revisión Moto
1006	Mantenimiento Preventivo Elevador		TALLER	21	Elevador Vehículos		Mantenimiento Elevador de Vehículos
1007	Mantenimiento Preventivo Desmontadora		TALLER	22	Desmontadora de ruedas		Mantenimiento Desmontadora de Ruedas

Ilustración 45: PMs

Se pone como ejemplo el primero de ellos, el mantenimiento del Coche1. Se añade el *Job Plan* en el que se va a basar la orden de trabajo que se va a crear automáticamente desde esta aplicación:



Job Plan	Description	Total Work Units	Sequence
2363	Revisión Coche	1.00	1

Ilustración 46: Job Plan del PM

Una vez creado, se añaden los parámetros que van a establecer cuándo se crea la orden de trabajo. En el caso de los coches y la moto se van a utilizar tanto frecuencias basadas en tiempo como en métricas. En cuanto a las máquinas del taller, simplemente se utilizarán frecuencias basadas en tiempo.

Frecuencias basadas en tiempo

En el caso de los vehículos, cada 2 años se creará una orden de trabajo para hacer el mantenimiento. Hay que añadir la próxima fecha estimada la primera vez que se crea el plan. Las siguientes veces se generará sola teniendo en cuenta la fecha en la que se ha creado la orden de trabajo y la frecuencia con la que esta se crea. Se pone una alerta de días, en la que la aplicación alerta a los usuarios de que se acerca la próxima fecha del mantenimiento. En este caso se ha puesto una alerta 15 días antes.

Para poder utilizar ambos tipos de frecuencias, hay que marcar las 3 casillas que se observan en la siguiente ilustración:

The screenshot shows a form titled "Work Order Generation Information". It contains three checkboxes, all of which are checked: "Use Last Work Order's Start Date to Calculate Next Due Date?", "Generate Work Order Based on Meter Readings (Do Not Estimate)?", and "Generate Work Order When Meter Frequency is Reached?". Below these is a tabbed interface with two tabs: "Time Based Frequency" (which is selected) and "Meter Based Frequency". Under the "Time Based Frequency" tab, there are four input fields: "Frequency:" with the value "2", "Alert Lead (Days):" with the value "15", "Extended Date:" with a calendar icon, and "Target Start Time:" with the value "00:00:0" and a clock icon. Below these are three more fields: "Frequency Units:" with the value "YEARS", "Estimated Next Due Date:" with the value "7/13/24" and a calendar icon, and "Adjust Next Due Date?" with an unchecked checkbox.

Ilustración 47: Frecuencias de tiempo en los PMs

Frecuencias basadas en métricas

Para añadir este tipo de frecuencia, primero la métrica ha de estar asociada al *asset* con el que se asocia el PM. Se selecciona la métrica y se establece la frecuencia con la que se generará la orden de trabajo. En este caso, al ser un coche, cada 15000km se generará la orden de trabajo de mantenimiento. También se establece un valor con el que se alerta a los usuarios, 14000km. Automáticamente, la aplicación calcula los kilómetros que le faltan al coche para que se genere la orden de trabajo, en este caso 15000km. El coche ya tiene una primera medida para el kilometraje de 15000km. En la zona de debajo, se observa la última medida y su fecha. Además, la aplicación calcula cual será la próxima medida necesaria para que salte la orden de trabajo, 30000km.

Meter Based Frequency						
Filter > 1 - 1 of 1						
Meter	Description	Frequency	Units to Go	Generate WO Ahead By	Alert Lead	
KM	Kilometraje Vehiculo	15,000.00	15,000.00		14,000.00	
Details						
Meter: KM		Average Units/Day: 0.00				
Frequency: 15,000.00		Rollover:				
Alert Lead: 14,000.00						
Generate WO Ahead By:						
Last Work Order Information		Next Work Order Projections				
Meter Reading: 15,000.00		Next Meter Reading: 30,000.00				
Meter Reading Date: 7/13/22 22:02:12		Units to Go: 15,000.00				

Activate Go to Setti

Ilustración 48: Frecuencias de métricas en los PMs

Forecast

Finalmente, hay que generar el *Forecast* para los días que se quiera llevar a cabo este plan. Como ejemplo se ha generado un *Forecast* de 19 días:

Forecast Details							
Filter > 1 - 19 of 19							
Forecast Date	Job Plan	New Date	Changed By	Changed Date	Remarks	Reforecast Pending?	
> 7/13/22	2363					☐	
> 7/14/22	2363					☐	
> 7/15/22	2363					☐	
> 7/16/22	2363					☐	
> 7/17/22	2363					☐	
> 7/18/22	2363					☐	
> 7/19/22	2363					☐	

Ilustración 49: Forecast de los PMs

Creación de la nueva aplicación

Para este proyecto se va a crear una nueva aplicación en Máximo que sirva para facilitar la vida del operador cuando ejecute las órdenes de trabajo y tenga que meter todas las métricas en el programa. Esta aplicación se va a llamar Revisión Vehículos. Para crearla hay que utilizar distintas aplicaciones de Máximo.

Database Configuration

La primera parte consiste en crear la nueva tabla con todos sus campos en la base de datos de Máximo. Es importante crearla a nivel de sistema para que después se pueda crear la aplicación asociada a esta tabla de la base de datos:

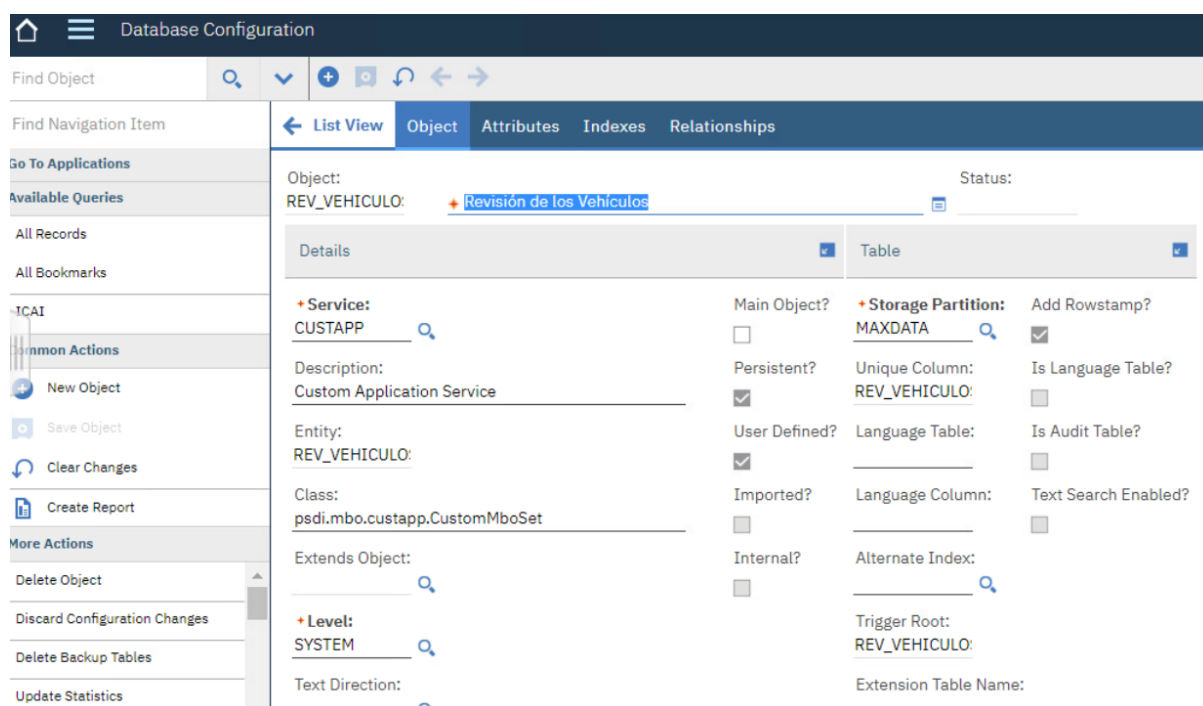


Ilustración 50: Nueva tabla en la base de datos

En la pestaña de atributos se crean todos los campos necesarios para la nueva aplicación. En la primera parte de este proyecto se encuentra una tabla con todos ellos. Además, se ha creado otro atributo más, 'TIPO_VEHIC', que, dependiendo del *Job Plan* que tenga la orden de trabajo, tendrá el valor de 'COCHE' o 'MOTO'.

Status	Attribute	Description	Type	Length
>	ASSET	Asset	ALN	50
>	CLIENTE	Cliente	ALN	50
>	DESCRIPTION	Description	ALN	50
>	DESCRIPTION_LONGDESCRIPTION	Description_longdescription	LONGALN	32,000
>	DESGASTE_COR_PI	Desgaste_cor_pi	ALN	50
>	DESGASTE_DISCO	Desgaste_disco	ALN	50
>	DESGASTE_NEUM	Desgaste_neum	ALN	50
>	DESGASTE_PAST	Desgaste_past	ALN	50
>	FD	Fd	ALN	50
>	FECHA	Fecha	DATETIME	10

Ilustración 51: Atributos de la nueva tabla

Como ejemplo se pone el campo de desgaste de los neumáticos. En este caso, será de tipo ALN, que significa que podrá ser alfanumérico. Además, este campo se va a rellenar mediante un nuevo dominio que se va a crear de tipo ALN también. Por ello, se añade el dominio ICAI_ESTADO al nuevo campo. Esto mismo ocurre con otros campos.

Attribute: DESGASTE_NEUM	+ Title: Desgaste_neum
+ Description: Desgaste_neum	Class:
+ Type: ALN	Domain: ICAI_ESTADO
+ Length: 50	Default Value:
Scale: 0	Alias: DESGASTE_NEUM
Required? ☐	Status:

Ilustración 52: Ejemplo de atributo

Para esta tabla se ha añadido un campo que es el número de *asset* al cual se le está haciendo el mantenimiento. Para que directamente se añada su descripción se crea una relación con la tabla de *assets* de la base de datos. De esta forma, una vez que el campo *asset* esté relleno en la nueva aplicación, su descripción aparecerá automáticamente.

Para que esta relación funcione, se añade una condición de que tanto el site como el número de *assets* sean igual en esta nueva tabla y en la tabla de los *assets*:

← List View

Object

Attributes

Indexes

Relationships

Object:

REV_VEHICULO: Revisión de los Vehículos

Status:

Relationships

Filter

>

1 - 1 of 1

Relationship	Child Object	Where Clause
ASSET	ASSET	ASSETNUM=:ASSET AND SITEID=:SITE

Details

Relationship:

ASSET

Where Clause:

ASSETNUM=:ASSET AND SITEID=:SITE

Child Object:

ASSET

Remarks:

New Row

Ilustración 53: Relación con otras tablas

Una vez creada la tabla, hay que añadir estos cambios estructurales a la base de datos.

Messages

Además de crear la tabla, en la aplicación Database Configuration, se pueden crear mensajes de error para campos en los que se meta un valor que no tiene sentido. En este caso, se han creado 6 mensajes diferentes para los campos de presión, voltaje y fuerzas. Para que en la aplicación se generen estos errores hay que crear un script que se explicará más adelante.

Para cada campo se han establecido valores inferiores y superiores que no son lógicos:

Messages

Messages ☒ Filter > 🔍 🔄 ⬆️ ⬇️ ⬅️ 1 - 5 of 6 ➡️ 📄 📌

Message Group	Message Key	Message ID	Value
rev%			
> REV_VEHIC	ErrorFuerzaInf	BMXZZ7152E	Valor de fuerza de frenado (F={0}) muy i
> REV_VEHIC	ErrorFuerzaSup	BMXZZ7151E	Valor de fuerza de frenado (F={0}) muy
> REV_VEHIC	ErrorPresionInf	BMXZZ7153E	Valor de la presión de los neumáticos (F
> REV_VEHIC	ErrorPresionSup	BMXZZ7154E	Valor de la presión de los neumáticos (F
> REV_VEHIC	ErrorVoltajeInf	BMXZZ7150E	Valor del voltaje de la batería (VOLTAJE

New Row

OK Cancel

Ilustración 54: Mensajes de error

Primero se crea el grupo de mensajes REV_VEHIC donde estarán contenidos todos los mensajes. Como *display method* se escoge el 'MSGBOX', que significa que aparecerán en un recuadro. En la descripción se añade un {0} que se utilizará en el script para añadir el valor que el operador ha rellenado en ese campo y que no concuerda con este.

Messages

Details

Message Group:
REV_VEHIC

Message Key:
ErrorFuerzaInf

• Display Method:
MSGBOX 🔍

• Message ID Prefix:
BMXZZ 🔍

• Message ID Suffix:
E 🔍

Message ID:
BMXZZ7152E

Display ID?
☒

Value:
Valor de fuerza de frenado (F={0}) muy inferior a cualquier valor normal de fuerza. Por favor, completa el campo con otro valor.

Ilustración 55: Ejemplo mensaje de error

Domains

Esta aplicación sirve para que el operador pueda rellenar los campos seleccionando valores desde una tabla. Finalmente se han creado 3 dominios de tipo ALN para rellenar todos los campos que tengan que ver con el desgaste, con el nivel de líquido y con el estado en el que se encuentra la revisión como tal.

A continuación, se puede observar el ejemplo del dominio que se ha creado para establecer el nivel de líquido. En este caso, se han creado 5 valores entre los que elegir. Simularán el porcentaje de líquido que hay, desde un 0% hasta un 100%. Es importante que el tipo de datos del dominio coincida con el tipo de datos que tiene el campo al que se le quiere añadir el dominio.

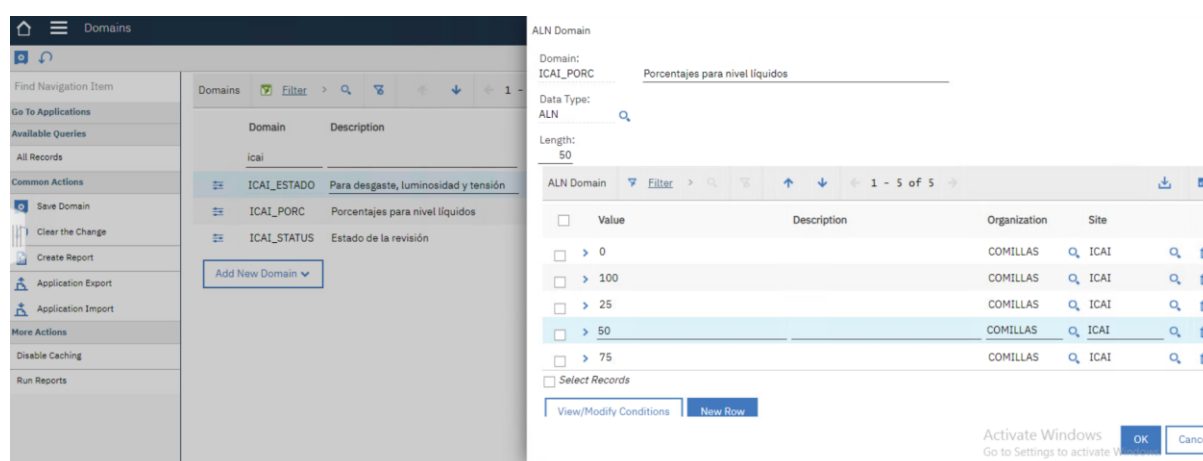


Ilustración 56: Dominios

Lookup

En el caso de los campos del usuario y del cliente, el operador podrá elegir también desde una tabla. En este caso, no se van a crear los posibles valores, si no que se cogerán desde otra tabla, los usuarios. Para ello, se ha exportado desde Máximo el XML de los *lookups* y se ha añadido un *lookup* llamado 'revvehic'. El 'mboname' se refiere al objeto desde el cual se van a coger los valores, el objeto es la tabla de la base de datos, 'maxuser'. Como condición para los valores que le van a aparecer al operador para escoger, se indica que su site por defecto sea ICAI, su 'defsite'.

Esta es la parte que se ha añadido al XML de los lookups que hay en Máximo:

```
<table id="revvehic" inputmode="readonly" mboname="maxuser"
selectmode="single" whereclause=" defsite = 'ICAI' ">
  <tablebody displayrowsperpage="20" filterable="true"
filterexpanded="true" id="revvehic_lookup_tablebody">
    <tablecol dataattribute="userid"
id="revvehic_lookup_tablebody_col_1" mxevent="selectrecord"
mxevent_desc="Go To %1" sortable="true" type="link"/>
  </tablebody>
</table>
```

Automation Scripts

Para crear revisiones de vehículos en esta nueva aplicación se van a crear 2 scripts. El primero va a servir para que se creen automáticamente cuando se genere una orden de trabajo desde la aplicación de mantenimiento preventivo. Además de esto, cogerá los valores de los campos de la orden de trabajo y los añadirá automáticamente a la nueva revisión que se crea. Rellenará los siguientes campos:

- El número de la orden de trabajo desde el que se crea
- Su descripción
- El *asset* para el que se está creando la nueva revisión
- La fecha en la que se ha creado la orden de trabajo, y, por tanto, la revisión
- El site en el que se ha creado

Además, añadirá el valor de 'NUEVO' al campo de estatus, para que cuando el operador comience a hacer la revisión pueda cambiarlo a 'EN PROGRESO' o 'TERMINADO'.

El Segundo script va a generar automáticamente los mensajes de error que se han creado en la aplicación *Database Configuration*. Cuando el operador escriba un valor que esté fuera de los límites, aparecerá una 'x' roja en el campo y saldrá el mensaje de error ya creado, dependiendo del campo y de si el valor está por debajo o por encima de los establecidos.

Ambos scripts se explicarán más adelante.

Conditional Expression Manager

Antes de crear la aplicación, se crean 2 condiciones para que algunos campos solo se muestren en función del tipo de vehículos que es, como por ejemplo la revisión de la batería, que solo se mostrará cuando sea un vehículo de tipo 'COCHE', o el nivel de aceite, cuando sea de tipo 'MOTO'. A continuación, se muestra la condición para añadir a los campos que solamente se muestren cuando es de tipo 'COCHE':

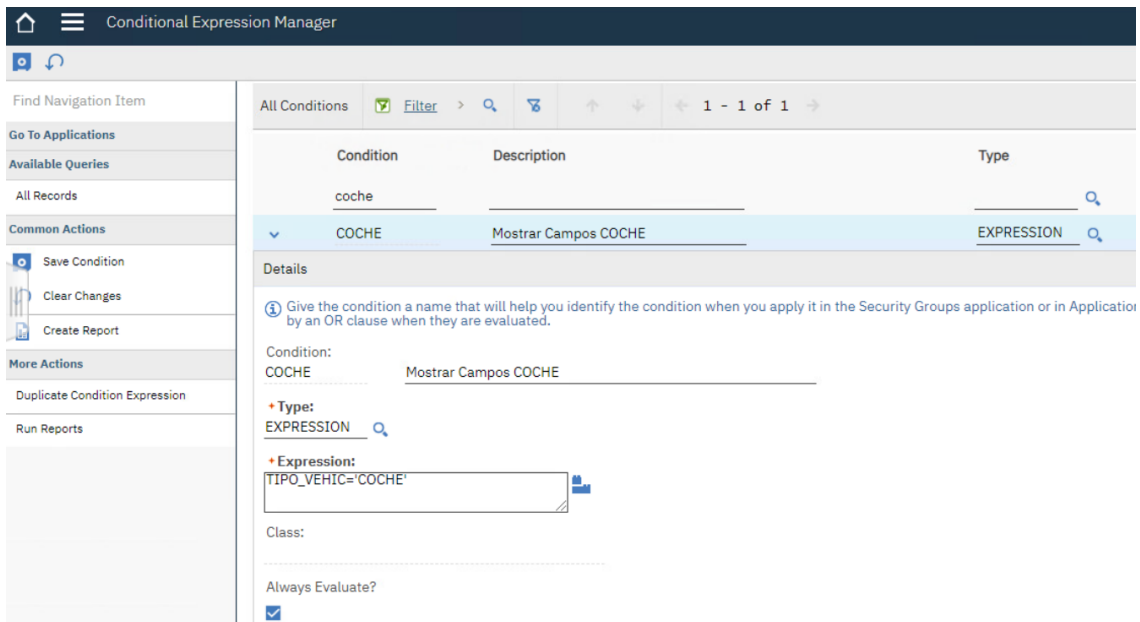


Ilustración 57: Condición para mostrar u ocultar campos

Además, estas condiciones habrá que meterlas en los permisos de la nueva aplicación de ambos *Security Groups*.

Application Designer

Finalmente, hay que crear la nueva aplicación en el Application Designer. Para ello, además de las herramientas que se muestran a continuación, se ha utilizado el XML que se ha ido generando para facilitar este diseño. Este XML se encuentra en el Anexo II adjunto.

Lo primero de todo es crear la tabla en la que se van a filtrar las revisiones que se quieran por medio de los siguientes campos:

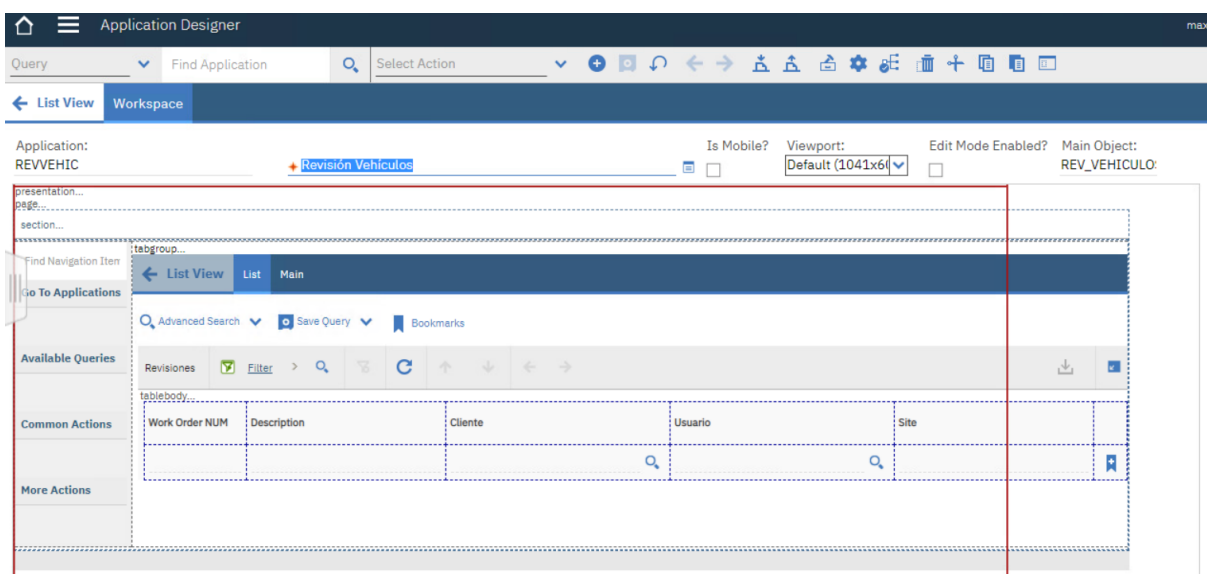


Ilustración 58: Campos para filtrar

En la parte del *main* se va a crear la pantalla con los campos que va a tener que rellenar el operador:

The screenshot shows the 'Application Designer' interface. At the top, there's a header with a home icon, a menu icon, and the text 'Application Designer'. Below this is a toolbar with a 'Query' dropdown, a 'Find Application' search bar, a 'Select Action' dropdown, and several icons for adding, deleting, undo, redo, and other actions. The main workspace is titled 'Application: REVVEHIC' and 'Revisión Vehículos'. On the left, there's a sidebar with 'Common Actions' and 'More Actions'. The main area contains a form with several sections: 'Usuario:', 'Cliente:', 'Asset:', and 'Site:' at the top. Below these are two 'section...' labels. Then there's a 'Tareas' section. This is followed by a 'Revisión Neumáticos' section containing 'Presion_neum:', 'Desgaste_neum:', and another 'section...'. Below that is a 'Revisión Luces' section with 'Luminosidad:' and two 'section...' labels. Finally, there's a 'Revisión Batería' section with 'Voltaje_bat:'.

Ilustración 59: Pantalla principal aplicación

Como se ha comentado anteriormente, en algunos campos habrá que añadir la condición que se ha creado en la aplicación de *Conditional Expression Manager* para que sólo aparezcan cuando sea necesario:

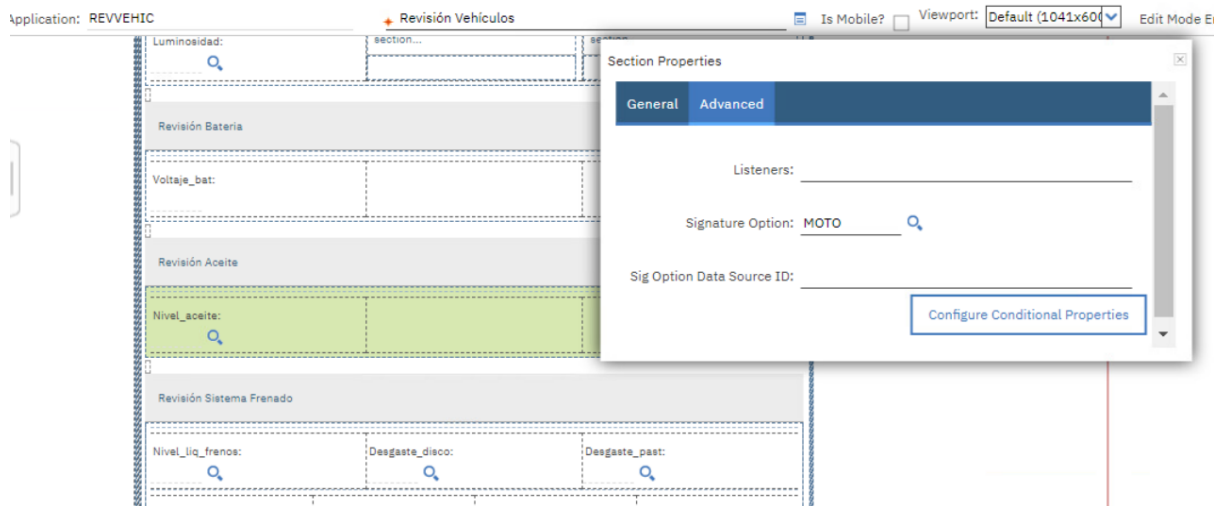


Ilustración 60: Mostrar/Ocultar campos

Lookups

Además, habrá que añadir el símbolo de la lupa a los campos en los que el operador pueda elegir un valor entre varios. Por una parte, se encuentran los campos que tienen asociado uno de los dominios que se ha creado, como, por ejemplo, el campo de desgaste de los neumáticos. En la parte de *lookup* se añade 'VALUELIST', que significa que, en la aplicación, asociará a este campo el dominio que tiene asociado en la tabla de la base de datos:

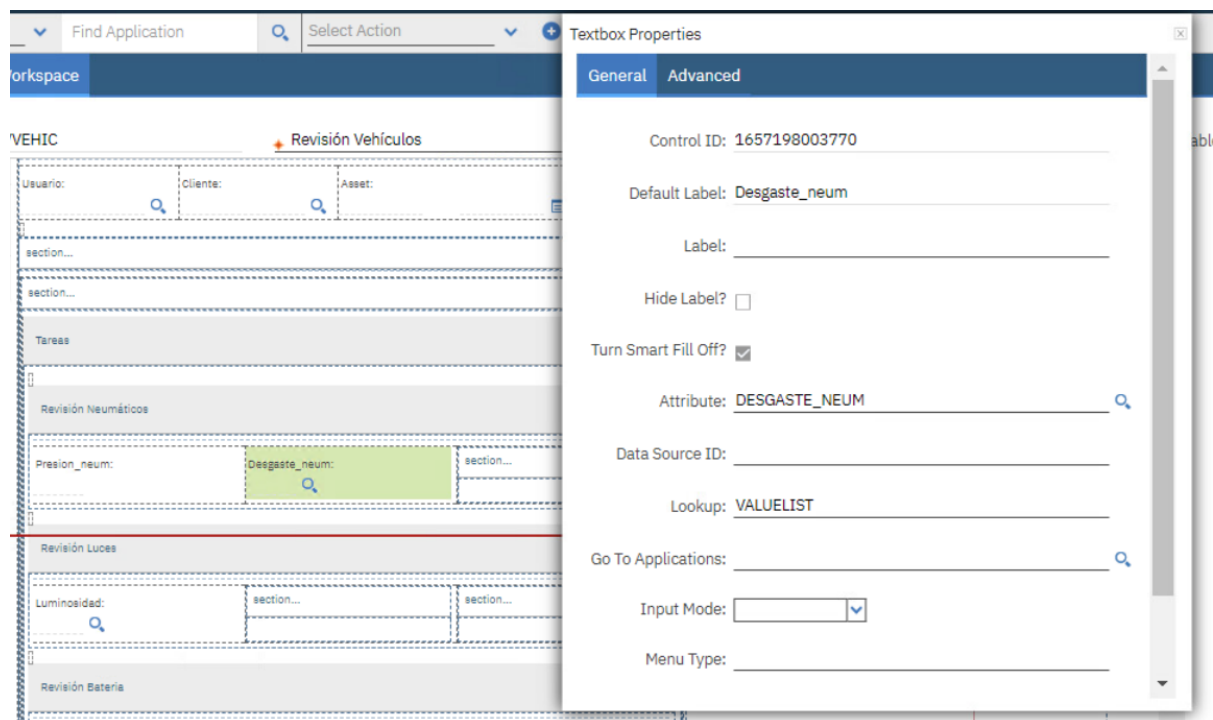


Ilustración 61: Asociar dominios

En el caso de los campos cliente y usuario, al haberse creado en el XML un nuevo *lookup* llamado 'REV_VEHIC', habrá que escribir ese nombre en vez de 'VALUELIST'.

En cuanto al campo *asset* y su descripción, se crean como un 'Multipart Textbox', que significa que coge la relación que se ha creado en la tabla de la base de datos y añade el campo de la descripción dependiendo del número de *asset* que haya:

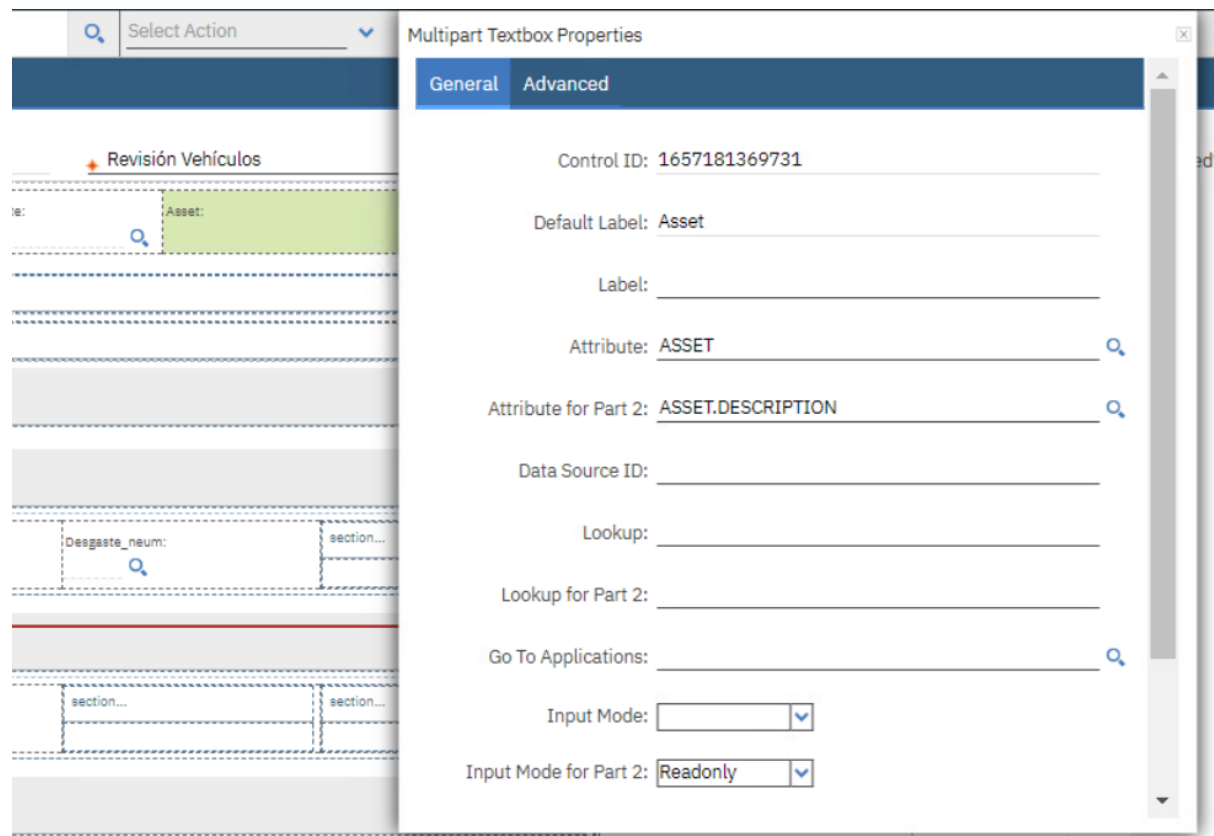


Ilustración 62: Relación de tablas

Resultado final

Esto es lo que se verá cuando la revisión sea de un coche:

Revisión Vehículos

Find Rev_vehiculosid

Find Navigation Item

Go To Applications

Available Queries

All Records

EN PROGRESO

NUEVO

REV INPROG USUARIO

REV TERMINADA USUARIO

TERMINADO

Common Actions

New REVVEHIC

Save REVVEHIC

Clear Changes

More Actions

Delete REVVEHIC

Add to Bookmarks

Work Order NUM: 15029 Description: Revisión Coche Fecha: 7/16/22 18:11:25 Status: NUEVO

Usuario: Cliente: Asset: 1018 Coche 2 Site: ICAI

Tareas

Revisión Neumáticos

Presion_neum: Desgaste_neum:

Revisión Luces

Luminosidad:

Revisión Bateria

Voltaje_bat:

Revisión Sistema Frenado

Nivel_liq_frenos: Desgaste_disco: Desgaste_past: Fmax: Fmin: Fd: Fi:

Ilustración 63: Revisión coche

Cuando sea de una moto se verá los siguientes campos:

Revisión Vehículos

Find Rev_vehiculosid

Find Navigation Item

Go To Applications

Available Queries

All Records

EN PROGRESO

NUEVO

REV INPROG USUARIO

REV TERMINADA USUARIO

TERMINADO

Common Actions

New REVVEHIC

Save REVVEHIC

Clear Changes

More Actions

Delete REVVEHIC

Add to Bookmarks

Work Order NUM: 15037 Description: Revisión Moto Fecha: 7/16/22 18:11:48 Status: NUEVO

Usuario: Cliente: Asset: 1065 Moto Site: ICAI

Tareas

Revisión Neumáticos

Presion_neum: Desgaste_neum:

Revisión Luces

Luminosidad:

Revisión Aceite

Nivel_aceite:

Revisión Sistema Frenado

Nivel_liq_frenos: Desgaste_disco: Desgaste_past: Fmax: Fmin: Fd: Fi:

Revisión Sistema Cadena

Tension_cad: Desgaste_cor_pi:

Ilustración 64: Revisión moto

Creación de los *Automation Scripts*

Se van a crear tres scripts, dos de ellos ya se han explicado brevemente. Estos scripts se encuentran en los Anexos III, IV y V.

Para crear un script mediante esta aplicación, primero hay que crear el *launchpoint* basado en un objeto o tabla de la base de datos, o en un atributo de una de estas tablas. El *launchpoint* le dice a esta aplicación cuando debe ejecutar el script. Una vez que se crea el *launchpoint* con las condiciones que se desean, se escribe el código que se va a lanzar.

En cuanto al código, la inicialización es la misma para todos. Se crea una función log para poder sacar por la pantalla de los logs lo que se quiera en caso de que algo no funcione y haya que hacer un *debug*, ya que esta aplicación no lo hace. Además, hay que inicializar el objeto o atributo en sí, el mbo, que se utilizará para sacar los datos que haya que utilizar en el código.

```
# Helper function for printing
def log(e):
    logger.debug(launchPoint + ":" + scriptName + (" >> %s" % e) + "\r" )

# Init
mxServer = MXServer.getMXServer()
userInfo = mxServer.getSystemUserInfo()
micService = MicService(mxServer)
micService.init()
micUserInfo = micService.getNewUserInfo()
logger = MXLoggerFactory.getLogger("maximo.script")
```

NUEVAAPP

Este es el primer script que se ha creado y del que ya se ha hablado. El *launchpoint* es a nivel de objeto. Se escoge la tabla de *Work Order*. Como condición se establece que escoja solamente las órdenes de trabajo que sean del site ICAI y que se generen mediante los 2 *Job Plans* de revisión de vehículos. Además, se indica que el script ha de ser ejecutado cuando se añada la orden de trabajo, si se cambia algo en esta y se guarda, no se volverá a crear otra revisión en la nueva aplicación.

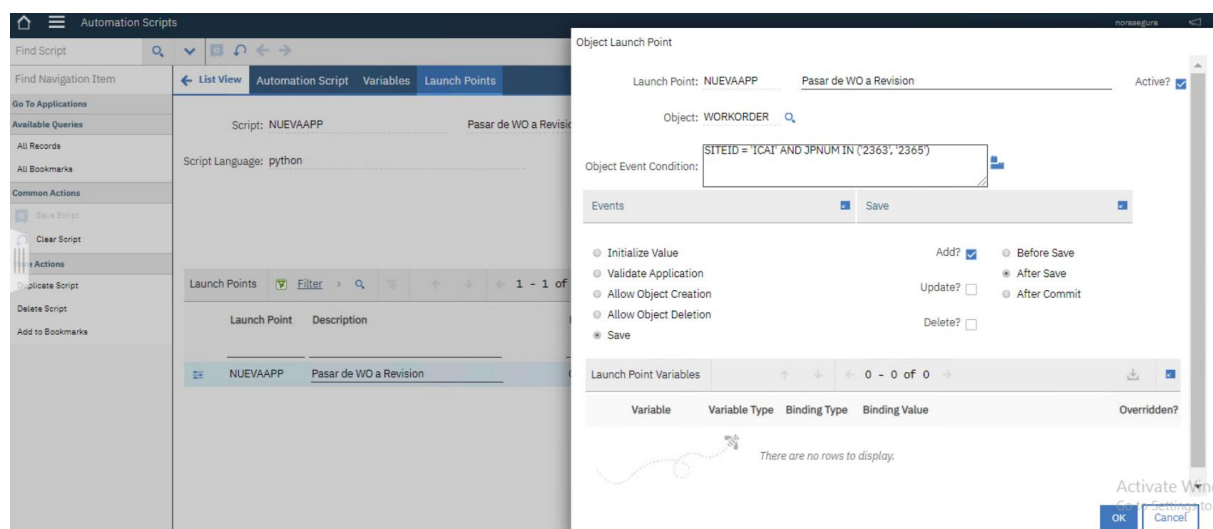


Ilustración 65: Launchpoint NUEVAAPP

Una vez hecho esto, se escribe el código. Se inicializa todo, y se sacan los valores que se quieren copiar en la nueva revisión. Para sacar los valores se utiliza la siguiente función del mbo:

```
wonum = mbo.getString("WONUM")
```

Para añadir estos valores, se crea otro nuevo objeto asociándolo a la nueva tabla de la base de datos que se ha creado:

```
revision= mxServer.getMboSet("REV_VEHICULOS",userInfo)
revisionWhere = "1 = 1"
revision.setWhere(revisionWhere)
revisionmbo = revision.getMbo(0)
```

Una vez hecho esto, se utiliza una función para añadir valores a los campos de este nuevo objeto:

```
newrevision = revision.add()

newrevision.setValue("DESCRIPTION",description,
MboConstants.NOVALIDATION | MboConstants.NOACCESSCHECK)
```

Finalmente, se establece en la base de datos el valor del campo 'TIPO_VEHIC', dependiendo del *Job Plan*, ya que este campo se utilizará en el tercer script.

ERRORESVEHICULOS

Al igual que pasaba con el anterior, ya se ha explicado el objetivo de este script. En este caso, se han creado 6 *launchpoints* a nivel de atributo, ya que, se quiere que cada vez que se escriba un valor incorrecto en los diferentes campos, salte el error correspondiente:

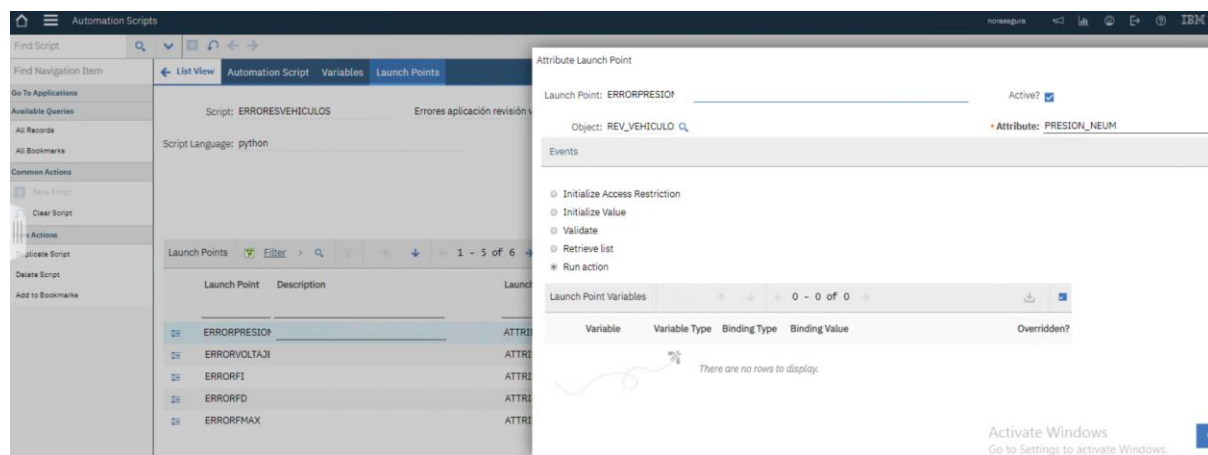


Ilustración 66: Launchpoints ERRORVEHICULOS

Estos *launchpoints* se utilizarán en el script a modo de condición.

El código se inicializa igual, pero añadiendo una función que crea el error:

```
# Helper function to set an error
def setMessage(errGroup, errKey, errParams):
    global errorkey, errorgroup, params
    errorkey=errKey
    errorgroup=errGroup
    params=errParams
```

Una vez hecho esto, se cogen los valores de interés por medio del mbo, que es la tabla de la nueva aplicación. Se utilizan diferentes *ifs* teniendo en cuenta el launchpoint y el valor del campo para añadir el error adecuado. El siguiente *if* sirve para añadir el mensaje de que el valor es muy superior al adecuado al campo de presión:

```
if (launchPoint=="ERRORPRESION" and presion>5):
    setMessage("REV_VEHIC", "ErrorPresionSup", [presion])
    log("presion: %s" % presion)
```

Además, se añade el valor que el operador ha metido en el campo al mensaje, en la parte de este en la que se había añadido el {0}. También se añade un log con el valor de la presión para asegurarse de que está cogiendo bien los datos de la tabla.

WOCHILD

Por último, se ha creado el script WOCHILD. El objetivo de este script es el de crear órdenes de trabajo hijas cada vez que el operador mete nuevos valores en los campos de cualquier revisión y los guarda. Para ello, se ha creado un *launchpoint* a nivel de objeto, en el que utilizar la nueva tabla que se ha creado. En este caso, se añade la condición de que se ejecute el script cada vez que se actualice y se guarde cualquier campo de la revisión.

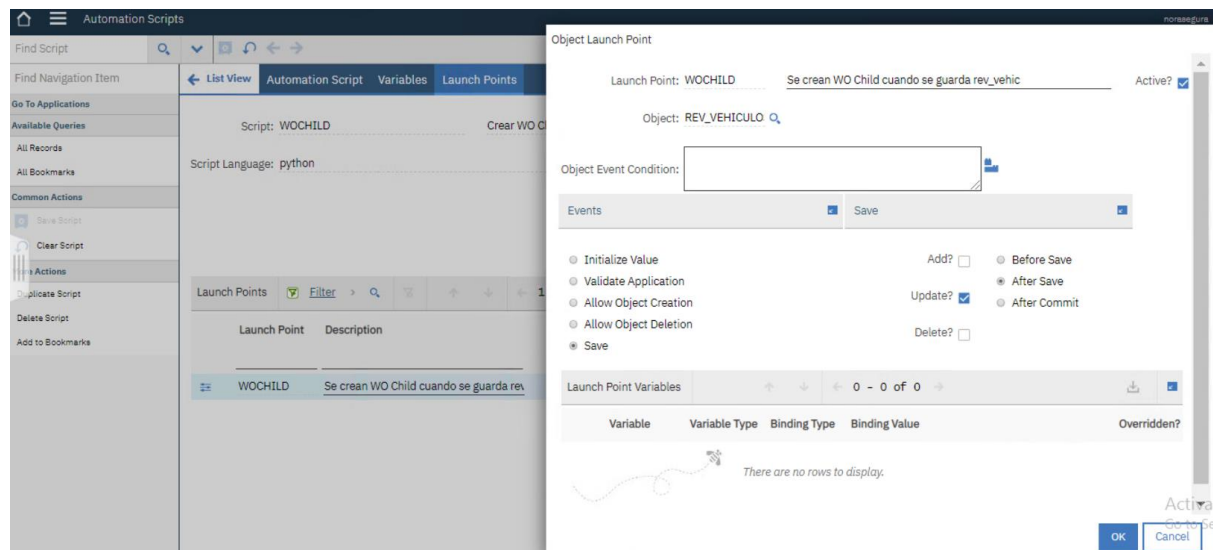


Ilustración 67: Launchpoint WOCHILD

El script se inicializa igual que el primero, y se cogen los siguientes valores de las revisiones de vehículos:

- La descripción
- El site, para añadirlo a la nueva orden de trabajo
- El *asset*, para añadirlo a la nueva orden de trabajo
- El *wonum*, para añadirlo como orden de trabajo padre de la nueva orden
- Todos los valores de las medidas que añade el operador
- El tipo de vehículo

Primero, se calculan los valores de eficacia, ovalidad y desequilibrio del sistema de frenado, mediante las fuerzas que el operador a añadido. A continuación, se asignan valores a estas nuevas variables según lo explicado en la parte de diseño de este proyecto. Dependiendo del tipo de vehículo se utilizan todas o 2 fórmulas. Por ejemplo, la ovalidad de las ruedas:

```
O = ((fmax-fmin)/fmax)*100

if O < 30:
    ovalidad = "BIEN"
elif O > 30 and O < 55:
    ovalidad = "LEVE"
elif O > 55:
    ovalidad = "GRAVE"
```

Una vez obtenidos estos nuevos valores, se procede a crear las órdenes de trabajo basadas en los *Job Plans* de reparación, siguiendo distintas condiciones. Por ejemplo, en el caso de la presión de los neumáticos, se establece que, si esta es menor de 2,2, el script ha de crear una nueva orden de trabajo con el *Job Plan* de ‘Hinchar las Ruedas’:

```
if presion < 2.2:
    log("Hinchar ruedas")

    newwoadded = wo.add()

    newwoadded.setValue("JPNUM", "2368", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
    newwoadded.setValue("SITEID", site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
    newwoadded.setValue("ASSETNUM", asset, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
    newwoadded.setValue("PARENT", wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

    wo.save()
```

Finalmente, se consigue crear las órdenes de trabajo hijas de la orden de revisión del vehículo, si las medidas que ha añadido el operador no son adecuadas y los elementos necesitan ser reparados o cambiados.

Creación de las *Escalations*

Máquinas del taller

Esta *Escalation* tiene que ver con las órdenes de trabajo que se crean desde los PMs de las 2 máquinas del taller. Estas órdenes de trabajo se crean en base a los *Job Plans* del mantenimiento de las máquinas, que tienen 3 tareas cada una. Si el operador ejecuta la primera tarea, que es la de análisis de las máquinas, y observa que todo funciona correctamente, pondrá esta tarea en 'TERMINADA', y, automáticamente las otras 2 tareas se cancelarán. Sin embargo, si pone esta tarea en el estado 'RESOLVED' o 'COMPLETED', tendrá que ejecutar las tareas restantes.

Para que ocurra esto automáticamente, primero se indica que la *Escalation* se aplica en la tabla 'WOACTIVITY', ya que es la tabla que contiene las tareas, hijas de la orden de trabajo.

A continuación, se establece la condición para ejecutarse, la cual dice que el Job Plan debe ser uno de los 2 que se encargan del mantenimiento de las máquinas, y que, además, la tarea 10, que es la primera, tiene un status igual a 'TERMINADA'.

Si se cumplen estas condiciones, la *Escalation* entra en el *Escalation Point* y accede a las otras 2 tareas restantes. En esta parte es donde se añade la acción que va a cambiar el estado de las otras 2 tareas a 'CANCELED'.

The screenshot displays the SAP Escalation configuration interface. The 'Condition' field contains the SQL query: `parent in (select wonum from workorder where jpnun in ('2375','2376')) and parent in (select parent from woactivity where taskid='10' and status='TERMINADA')`. The 'Escalation Point' is set to 1. The 'Elapsed Time Attribute' is set to `taskid in ('20','30')`. The 'Description' field is set to 'Cambia estado tareas ICAI'. The 'Actions' section shows a single action with ID 1950, description 'Cambia estado tareas ICAI', type 'CHANGESTATUS', and sequence 10.

Ilustración 68: *Escalation Máquinas*

La *query* que se está utilizando es la siguiente:

```
select parent, wonum from woactivity where parent in (select wonum from workorder where jpnun in ('2375','2376')) and parent in (select parent from woactivity where taskid='10' and status='TERMINADA') and taskid in ('20','30')
```

Actions

La acción se crea para el objeto *Work Order* y de tipo *Change Status*. El valor al que se cambia es 'CAN':

Ilustración 69: Acción Escalation

Service Request

La segunda *Escalation* que se crea sirve para avisar a los usuarios mediante un correo de que su *Service Request* ha sido completada. Para ello, se aplica a las SRs y se ejecuta solamente si el *site* es ICAI y si su estado es 'RESOLVED'. Se crean 3 *Escalation Points*, ya que hay 3 usuarios y se quiere enviar a cada uno su correo cada vez que se cierre una de sus *Service Request*.

Ilustración 70: Escalation Service Request

En cada *Escalation Point*, se añade la condición de que el *owner* de la *Service Request* sea el propio usuario, y se añade el *Communication Template* de cada uno:

Details

Escalation Point: 1

Elapsed Time Attribute: STATUSDATE

Escalation Point Condition: OWNER='USUARIO1'

Description: SR del USUARIO1

New Row

Actions Notifications

Notifications Filter 1 - 1 of 1

Template	Role/Recipient	Subject
1469	AFFECTEDBY, USUARIO1	

New Row

Ilustración 71: Escalation Point Service Request

Communication Templates

Para ello, hay que crear 3 Communication Templates que se aplican en las SRs, cada uno dirigido a un usuario. Se añade que el correo es enviado por el Operador y se escribe el correo que les va a llegar:

Escalations / Communication Templates

Find Template

Find Navigation Item

Available Queries

All Records

All Bookmarks

Common Actions

New Communication Template

Save Communication Template

Clear Changes

Change Status

Create KPI

Actions

Import Library/Folders

Create Template

Delete Template

Add to Bookmarks

Template: 1469 Notificación de que el asset está listo Created By: NORASEGURA Status: ACTIVE Date: 7/11/22 18:36:54

Applies To: SR Accessible From: ALL

Track Failed Messages? []

Comm Log Entry? [x]

Template Details

To: AFFECTEDBY, USUARIO1

CC:

bcc:

Send From: Operador1

Reply To:

Subject:

Message:

Hola [SR],
Su correo se encuentra totalmente reparado y puede venir a recogerlo cuando quiera.
En su caso, FAVOR DE REVISAR.

Ilustración 72: Communication Template

Creación del *Start Center*

Para crear los 2 *Start Centers* de los 2 *Security Groups*, primero, se decide que tipo de tablas tiene que haber:

The screenshot shows the 'Layout and Configuration' interface. The top bar indicates the layout is 'Narrow-Wide' and the description is 'Start Center Clientes ICAI'. The left sidebar lists various navigation items. The main area is divided into two columns: 'Left Column' and 'Right Column'. The 'Left Column' contains a table with one row: 'Favorite Applications' with a 'Display Name' of 'Favorite Applications' and an 'Order' of 1. The 'Right Column' contains a table with four rows: 'Service Request En Progreso' (Order 1), 'Service Request Resuelta' (Order 2), 'Revisión Vehículos En Progreso' (Order 3), and 'Revisión Vehículos Terminada' (Order 4). Both columns have a 'Select Content' button below the table. The bottom bar has 'Finished' and 'Cancel' buttons.

Ilustración 73: Tipos de tablas en el *Start Center*

A continuación, se selecciona para cada tabla la aplicación que se asocia a ella y los atributos que se quieren ver. Además, hay que añadir una *query* que dependerá de cada usuario o *Security Group*. Estas *queries* se crean directamente en cada aplicación poniendo condiciones para filtrar los datos que tengan las aplicaciones. En la siguiente ilustración se muestra el ejemplo de la aplicación *Service Request* y la *query* 'SRINPROG1'.

Result Set Setup

Find Navigation Item

Go To Applications

My Recent Applications

Administration

Analytics

Asset Configuration Manag...

Assets

Change

Change (HSE)

Contracts

Financial

Flight Log (CM)

Infrastructure

Integration

Inventory

Operations (HSE)

Planning

Preventive Maintenance

Purchasing

Release

Safety and Quality Manage...

Security

Self Service

Service Desk

Service Level

Service Provider (SP)

System Configuration

Task Management

Work Orders

Portlet Type: Result Set

Display Name: Service Request En Progreso

Application: ZSR

Query: SRINPROG1

SR INPROG USUARIO1

Field Configuration

Chart Options

Color Options

To choose the fields that appear in the portlet, first select an application in the Application field, then select an object from the Object List.

Object List

Service Requests (Leonardo)

Available Fields

Filter

1 - 6 of 336

Description	Name
Accumulated Hold Time(HH-MM)	ACCUMULATEDHOLDTIME
Actual Labor Cost	ACTLABCOST
Actual Labor Hours	ACTLABHRS
Actual Contact	ACTUALCONTACTDATE
Actual Finish	ACTUALFINISH
Actual Start	ACTUALSTART

Add Selected

Add All

Selected Fields

1 - 6 of 6

Order	Description	Name
1	Service Request	TICKETID
2	Summary	DESCRIPTION
3	Asset Description	ASSET_DESCRIPTION
4	Reported Date	REPORTDATE
5	Reported To	PLUSGREPORTEDTO
6	Reported By	REPORTEDBY

Remove Selected

Remove All

Ilustración 74:Atributos de cada tabla

Para cada tabla del *Start Center* se ha creado una *query* condicionando los datos que van a poder observar los usuarios.

Por ejemplo, la *query* 'SRINPROG1' indica que los usuarios solo puedan ver las *Service Request* que estén 'INPROGRESS', del *site* ICAI y que sean suyas.

View/Manage Queries

Queries

Filter

1 - 2 of 2

Query Name	Description	Default?	Public?	Contract
srinpr				
SRINPROG	SR ICAI INPROG	☐	☒	
SRINPROG1	SR INPROG USUARIO1	☐	☒	

Clause:

(status = 'INPROG' and siteid like '%ICAI%' and owner=:user)

Contract:

OK

Cancel

Ilustración 75: Query 'SRINPROG1'

Operador

A continuación, se muestra cómo queda el *Template* final del grupo de seguridad del operador para el *Start Center*:

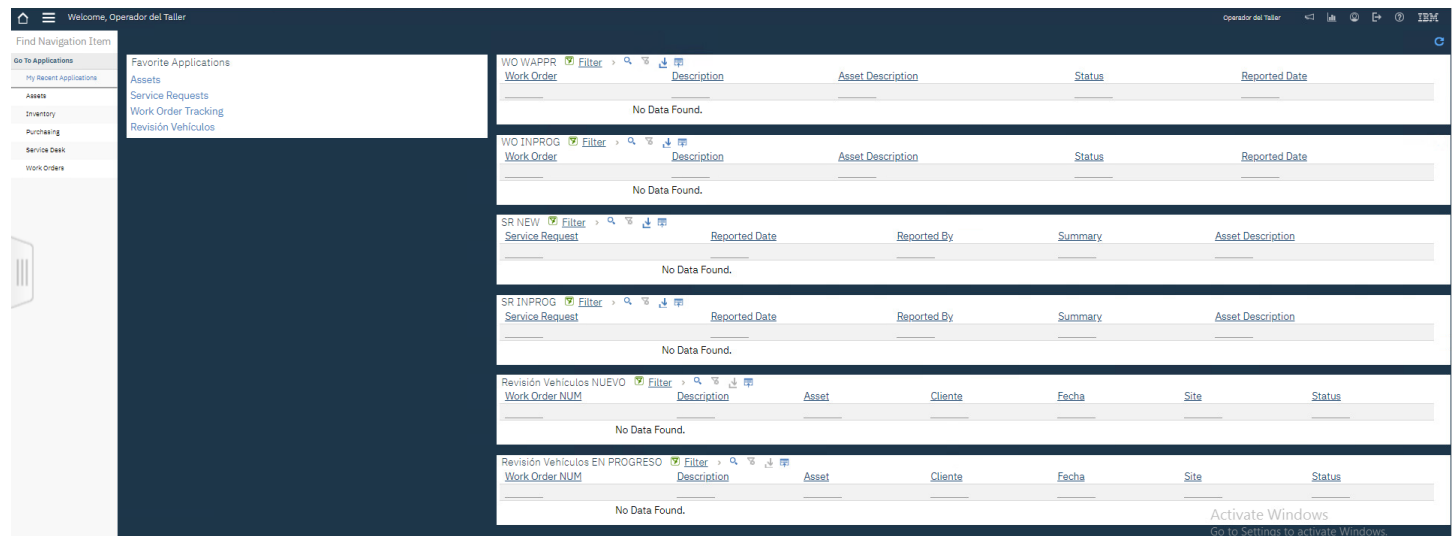


Ilustración 76: Start Center Operador

Usuarios

En la siguiente ilustración se observa el resultado final del *Template* de los usuarios del taller:

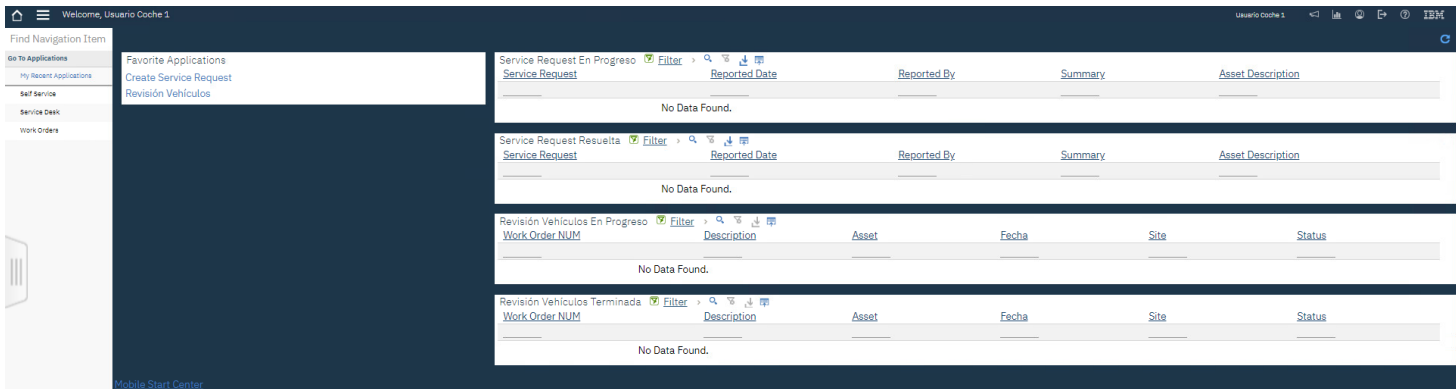


Ilustración 77: Start Center Usuarios

Resultados

Se van a mostrar los resultados finales de cómo actuarían las automatizaciones dentro de Máximo. La única automatización que no se puede mostrar es la *Escalation* de los *Service Request*, ya que este sistema se ha creado en un entorno de desarrollo que no contiene servicio de correo, y, por tanto, la *Escalation* no hará nada.

Escalation Máquinas

Para comprobar esta *Escalation* se ha creado una orden de trabajo manualmente, con el *Job Plan* del elevador de vehículos asociado a esta. En la siguiente imagen se observa que los estados de las tareas están en 'WAPPR', que significa esperando a ser aprobadas.

The screenshot shows the 'Work Order Tracking' interface. The main header displays 'Work Order: 3199' and 'Mantenimiento Elevador de Vehículos'. The status is 'WAPPR'. Below this, a table lists tasks for Work Order 3199:

Sequence	Task	Summary	Estimated Duration	Status
10	Analizar el Elevador		0:00	WAPPR
20	Reparar el Elevador		0:00	WAPPR
30	Comprobar su funcionamiento		0:00	WAPPR

Ilustración 78: Tareas orden de trabajo elevador

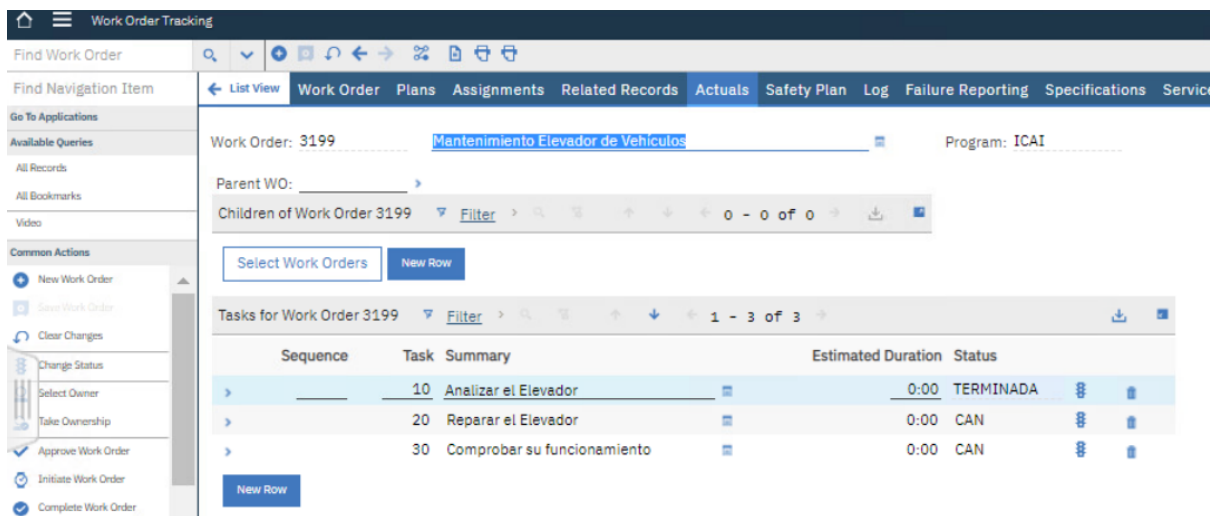
Una vez que el operador aprueba la primera tarea, que es la de análisis, y la realiza de forma favorable, cambia su estado a 'TERMINADA' en vez de 'RESOLVED' o 'COMPLETED'.

The screenshot shows the same 'Work Order Tracking' interface, but the status of the first task has been updated. The table now shows:

Sequence	Task	Summary	Estimated Duration	Status
10	Analizar el Elevador		0:00	TERMINADA
20	Reparar el Elevador		0:00	WAPPR
30	Comprobar su funcionamiento		0:00	WAPPR

Ilustración 79: Tarea 1 'TERMINADA'

Una vez hecho esto, la *Escalation* entra en el *Escalation Point* porque las condiciones se cumplen, y accede a las otras 2 tareas. La acción asociada cambia el valor del estado de las 2 tareas restantes a 'CAN':



The screenshot displays the 'Work Order Tracking' application interface. The main content area shows the details for Work Order 3199, titled 'Mantenimiento Elevador de Vehiculos'. Below this, there is a section for 'Tasks for Work Order 3199' with a table listing three tasks. The first task, 'Analizar el Elevador' (Sequence 10), is marked as 'TERMINADA'. The second task, 'Reparar el Elevador' (Sequence 20), and the third task, 'Comprobar su funcionamiento' (Sequence 30), are both marked as 'CAN' (Canceled). The interface includes a sidebar with navigation options and a top navigation bar with tabs for different views.

Sequence	Task	Summary	Estimated Duration	Status
10	Analizar el Elevador		0:00	TERMINADA
20	Reparar el Elevador		0:00	CAN
30	Comprobar su funcionamiento		0:00	CAN

Ilustración 80: Tareas canceladas

Preventive Maintenance

Se ha cambiado el PM del Coche1 para que se ejecute con una frecuencia de 1 día y se muestre que la orden de trabajo se crea automáticamente a las 00:00 de la noche:

Work Order: 3034 Revisión Coche

Location: CLIENTE1 Lugar de operación del coche 1

Asset: 1017 Coche 1

Program: ICAI

Class: WORKORDER

Status: WAPPR

Status Date: 7/14/22 00:00:00

Configuration Item:

Feature:

Feature Label:

Parent WO:

Classification:

Class Description:

Launch Entry Name:

Job Details

Asset Details

Priority

Job Plan: 2363

Asset Up?

Asset/Location Priority:

Ilustración 81: Orden de trabajo creada automáticamente

Al ser creada en base a un Job Plan, las tareas de este se convertirán en las hijas de la orden de trabajo:

Work Order: 3034 Revisión Coche

Program: ICAI

Status: WAPPR

Parent WO:

Children of Work Order 3034

Select Work Orders

New Row

Sequence	Task	Summary	Estimated Duration	Status
10	Revisión de Neumáticos		0:00	WAPPR
20	Revisión de Luces		0:00	WAPPR
30	Revisión de Batería		0:00	WAPPR
40	Revisión de Sistema de Frenado		0:00	WAPPR

Ilustración 82: Hijas de la orden de trabajo

Automation Scripts

Para comprobar los scripts se va a seguir con el resultado obtenido en el apartado de *Predictive Maintenance*.

NUEVAAP

Al generarse la orden de trabajo número '3034' desde la aplicación de PM, se genera sola la Revisión en la nueva aplicación de revisiones. Además, como ya se ha comentado, se traspasan los valores del número de la orden de trabajo, su descripción, la fecha, el *asset* y su *site* desde la orden de trabajo. También se establece el estado como 'NUEVO':

The screenshot displays the 'Revisión Vehículos' application interface. The top navigation bar includes a home icon, a menu icon, and the title 'Revisión Vehículos'. Below this is a search bar labeled 'Find Rev_vehiculosid'. The main content area is divided into a left sidebar and a main panel. The sidebar contains sections for 'Find Navigation Item', 'Go To Applications', 'Available Queries', and 'Common Actions'. The main panel shows a form for a new revision. At the top, it displays 'Work Order NUM: 3034', 'Description: Revisión Coche', 'Fecha: 7/14/22 00:00:00', and 'Status: NUEVO'. Below this, there are input fields for 'Usuario', 'Cliente', 'Asset: 1017', 'Coche 1', and 'Site: ICAI'. The form is organized into sections: 'Tareas', 'Revisión Neumáticos' (with fields for 'Presion_neum' and 'Desgaste_neum'), 'Revisión Luces' (with 'Luminosidad'), 'Revisión Bateria' (with 'Voltaje_bat'), and 'Revisión Sistema Frenado' (with fields for 'Nivel_liq_frenos', 'Desgaste_disco', 'Desgaste_past', 'Fmax', 'Fmin', 'Fd', and 'Fi').

Ilustración 83: Nueva Revisión

Se verifica que, al ser el *Job Plan* 'Revisión Coche', campos como el nivel de aceite o el sistema de cadena no aparecen.

ERRORES

Se meten distintos valores en los campos y se comprueba, que, por ejemplo, añadiendo un 100 en el campo del voltaje de la batería del coche, aparece el error que se ha creado en la aplicación *Database Configuration*:

The screenshot shows the 'Revisión Vehículo' application interface. The top navigation bar includes a home icon, a menu icon, and the title 'Revisión Vehículo'. Below this is a search bar with the text 'Find Rev_vehiculosid'. The main content area is divided into a left sidebar and a right main panel. The sidebar contains sections for 'Find Navigation Item', 'Go To Applications', 'Available Queries', 'All Records', 'EN PROGRESO', 'NUEVO', 'REV INPROG USUARIO', 'REV TERMINADA USUARIO', 'TERMINADO', 'Common Actions', and 'More Actions'. The main panel displays a form for vehicle inspection. At the top, there are fields for 'Work Order NUM: 3034', 'Description: Revisión Coche', 'Fecha: 7/14/22 00:00:00', and 'Status: NUEVO'. Below these are fields for 'Usuario:', 'Cliente:', 'Asset: 1017', 'Coche 1', and 'Site: ICAI'. The form is organized into sections: 'Tareas', 'Revisión Neumáticos', 'Revisión Luces', 'Revisión Batería', and 'Revisión Sistema Frenado'. In the 'Revisión Batería' section, the 'Voltaje_bat:' field has a red error icon next to it. A modal dialog box is open over this field, displaying the message: 'You entered: 100. Valor del voltaje de la batería (VOLTAJE_BAT=100.0) muy superior a cualquier valor normal de voltaje. Por favor, completa el campo con otro valor. (BMXZZ7149). Click "Edit My Value" to change the value you entered or "Go Back" to the value that was there before.' The dialog box has two buttons at the bottom: 'Edit My Value' and 'Go Back'.

Ilustración 84: Errores

Además, el mensaje de error lleva escrito el valor que el operador ha metido erróneamente en el campo.

WOCHILD

Finalmente, se comprueba que, una vez que se han añadido los valores de todos los campos y se ha guardado la revisión, se crean las órdenes de trabajo hijas de la orden de revisión previa.

Según los datos añadidos, se observa que, tanto la batería como el sistema de frenado, no están en buenas condiciones. Lo que falla en el sistema de frenado es el cálculo de la eficacia teniendo en cuenta las fuerzas que el operador ha metido:

Find Rev_vehiculosid

Find Navigation Item

Go To Applications

Available Queries

All Records

EN PROGRESO

NUEVO

REV INPROG USUARIO

REV TERMINADA USUARIO

TERMINADO

Common Actions

New REVVEHC

Save REVVEHC

Clear Changes

More Actions

Delete REVVEHC

Add to Bookmarks

Work Order NUM: 3034 Description: Revisión Coche Fecha: 7/14/22 00:00:00 Status: EN PROGRESO

Usuario: OPERADOR1 Cliente: USUARIO1 Asset: 1017 Coche 1 Site: ICAI

Tareas

Revisión Neumáticos

Presion_neum: 2.5 Desgaste_neum: BIEN

Revisión Luces

Luminosidad: BIEN

Revisión Batería

Voltaje_bat: 10

Revisión Sistema Frenado

Nivel_liq_frenos: 75 Desgaste_disco: BIEN Desgaste_past: BIEN

Fmax: 2.4 Fmin: 2.3 Fd: 2.5 Fi: 2.3

Ilustración 85: Valores añadidos en la revisión

Si el operador entra en la aplicación *Work Order Tracking*, comprueba que se han creado 2 nuevas *Work Orders* a las que se les ha asociado 2 *Job Plans*: 'Recambio Sistema de Frenado' y 'Cambiar Batería'.

Work Order	Reported Date	Actual Finish	Location	Classification	Description	Status	Owner	Job Plan	Asset	PN	SN	Structural Code	Is dynamic?	Dynamic Job Plan
3173	7/14/22 00:41:56		CLIENTE1	Recambio Sistema de Frenado	WAPFR				1017	3	5		NO	
3172	7/14/22 00:41:55		CLIENTE1	Cambiar Batería	WAPFR				1017	3	5		NO	

Ilustración 86: 2 nuevas órdenes de trabajo

Finalmente, se observa que las *Work Orders* contienen datos de la revisión en cuestión, como el *asset* y el *site*, y que, además, tienen una orden de trabajo padre, que es la '3034', la primera en generarse mediante la aplicación de PM.

Work Order Tracking

Find Work Order

Find Navigation Item

Go To Applications

Available Queries

All Records

All Bookmarks

All Work Orders

HF2000 - View all workorder

WO - View all Workorder (T REPA)

Work Orders with status equal WA

Work Orders associated with the c

All opened and not complete

New Work Order

Clear Changes

Change Status

Work Order: 3173 [Recambio Sistema de Frenado](#)

Program: ICAI Taller ICAI

External Reference ID: _____

Contract: _____

Location: CLIENTE1 > Lugar de operacion del coche 1

Phase: _____

Asset: 1017 > Coche 1

Status: WAPPR Waiting on approval

Part Number: _____

Status Date: 7/14/22 00:41:56

State of Change: _____

Work Type: _____

Serial Number: _____

Technical Operation Report: _____

Structural Code: _____

Intervention Report: _____

Parent WO: 3034 >

Failure Report: _____

Ilustración 87: Órdenes de trabajo hijas

Start Centers

Esto verá el operador cuando tenga distintas órdenes de trabajo, SRs y Revisiones asignadas:

Operator de Taller

Find Navigation Item

Go To Applications

Favorite Applications

Assets

Service Requests

Work Order Tracking

Revisión Vehículos

Assets

Inventory

Purchasing

Service Desk

Work Orders

WO WAPPR	Filter	Description	Asset Description	Status	Reported Date
14194		Revisión Coche	Coche 1	WAPPR	7/16/22 13:40:18
14189		Revisión Coche	Coche 2	WAPPR	7/16/22 13:39:46
15029		Revisión Coche	Coche 2	WAPPR	7/16/22 18:11:25
14217		Revisión Moto	Moto	WAPPR	7/16/22 13:43:06
14226		Hinchar ruedas	Moto	WAPPR	7/16/22 13:44:08
14227		Recambio luces	Moto	WAPPR	7/16/22 13:44:09
15037		Revisión Moto	Moto	WAPPR	7/16/22 18:11:48
14202		Mantenimiento Elevador de Vehículos	Elevador Vehículos	WAPPR	7/16/22 13:40:36
14206		Mantenimiento Desmontadora de Ruedas	Desmontadora de ruedas	WAPPR	7/16/22 13:41:20

1 - 9 of 9

WO INPRG	Filter	Description	Asset Description	Status	Reported Date
14126		Revisión Coche	Coche 2	INPRG	7/16/22 13:21:46

1 - 1 of 1

SR NEW	Filter	Reported Date	Reported By	Summary	Asset Description
32803		7/16/22 11:55:20	USUARIO3	La luz delantera no funciona	Moto

1 - 1 of 1

SR INPRG	Filter	Reported Date	Reported By	Summary	Asset Description
32804		7/16/22 11:56:27	USUARIO1	Rueda pinchada	Coche 1
32802		7/16/22 11:54:11	USUARIO2	El coche no arranca	Coche 2

1 - 2 of 2

Revisión Vehículos NUEVO	Filter	Description	Asset	Cliente	Fecha	Site	Status
14189		Revisión Coche	1018		7/16/22 13:39:46	ICAI	NUEVO

Activate Windows
Go to Settings to activate Windows.

Ilustración 88: Resultado Start Center Operador

Los usuarios verán lo siguiente:

Usuario Coche 1

Find Navigation Item

Go To Applications

Favorite Applications

Create Service Request

Revisión Vehículos

Self Service

Service Desk

Work Orders

Service Request En Progreso	Filter	Reported Date	Reported By	Summary	Asset Description
32804		7/16/22 11:56:27	USUARIO1	Rueda pinchada	Coche 1

1 - 1 of 1

Service Request Resueta	Filter	Reported Date	Reported By	Summary	Asset Description
32801		7/15/22 14:06:34	USUARIO1		

1 - 1 of 1

Revisión Vehículos En Progreso	Filter	Description	Asset	Fecha	Site	Status
No Data Found.						

Revisión Vehículos Terminada	Filter	Description	Asset	Fecha	Site	Status
2447		Revisión Coche	1017	7/13/22 20:32:39	ICAI	TERMINADO
3034		Revisión Coche	1017	7/14/22 00:00:00	ICAI	TERMINADO
14000		Revisión Coche	1017	7/16/22 12:48:11	ICAI	TERMINADO
14077		Revisión Coche	1017	7/16/22 13:08:39	ICAI	TERMINADO

1 - 4 of 4

Ilustración 89: Resultado Start Center Usuarios

Conclusiones

El proyecto concluye habiendo diseñado y creado un sistema de gestión de activos con mantenimiento preventivo. Los objetivos finales que se propusieron para crear este sistema en Máximo se han cumplido, y, además, se han añadido al sistema más elementos de los que se habían propuesto en un principio. Se decidió crear un sistema basado en algo real, como es un taller de coches, y ha sido todo un acierto, ya que, con este ejemplo, se entiende mucho mejor el funcionamiento del software en sí y de todo el sistema creado.

Cuando se comenzó el proyecto, estos eran los principales objetivos a realizar con el software IBM Máximo:

- Crear una Organización con todos los elementos
- Crear *Escalations*
- Crear *Automation Scripts*
- Crear el Plan de Mantenimiento Preventivo
- Crear los *Start Centers*

Se han cumplido todos, pero, además, se ha añadido el objetivo de crear una nueva aplicación para que el sistema fuera más automático y sencillo de utilizar para el operador. Gracias al hecho de crear esta nueva aplicación de revisión de coches, se han utilizado múltiples aplicaciones de Máximo que se interconectan entre sí.

Anexos

Anexo I: Objetivos de Desarrollo Sostenible

Hoy en día cualquier empresa trata de alinearse con uno o varios Objetivos de Desarrollo Sostenible. Es muy importante tener un buen plan de sostenibilidad, ya que beneficia a la empresa, pero también beneficia a toda la sociedad.

Este proyecto se podría alinear principalmente con el ODS 9, industria, innovación e infraestructura, con el 11, ciudades y comunidades sostenibles y con el 12 producción y consumo responsables.

- **Industria, innovación e infraestructura:** Considero que el proyecto está totalmente alineado con este objetivo, ya que este trata sobre la innovación en la industria, utilizando un software y optimizando la vida útil de los activos, además de crear una infraestructura digital para los activos de una empresa.
- **Ciudades y comunidades sostenibles:** Este proyecto trata de gestionar los activos de tal forma que sean lo más sostenibles posible. Si un activo falla porque no se le ha hecho un correcto mantenimiento muchas veces se desecha, algo que va en contra de la sostenibilidad.
- **Producción y consumo responsables:** Este objetivo lo identifico con el proyecto por una razón similar a la del objetivo número 11. Considero que una producción y consumo responsable de unos activos se da cuando se optimiza el uso de estos activos al máximo.



Ilustración 90: Objetivos de Desarrollo Sostenible

Anexo II: XML de la nueva aplicación

```
<presentation id="revvehic" ismobile="false" mboname="REV_VEHICULOS"
resultstableid="results_showlist" version="6.0.0" viewport="1041x600"
whereclause="TIPO_VEHIC IS NOT NULL">
    <page id="mainrec" scroll="false">
        <include controltoclone="pageHeader" id="INCLUDE-
pageHeader"/>
        <clientarea id="clientarea">
            <tabgroup id="maintabs" style="form">
                <tab default="true" id="results"
label="List" type="list">
                    <menubar event="search"
id="actiontoolbar1" sourcemethod="getAppSearchOptions"/>
                    <table id="results_showlist"
inputmode="readonly" label="Revisiones" mboname="REV_VEHICULOS">
                        <tablebody
displayrowsperpage="20" filterable="true" filterexpanded="true"
id="results_showlist_tablebody">
                            <tablecol
dataattribute="WONUM" id="results_showlist_column_link"
mxevent="selectrecord" mxevent_desc="Go To %1" type="link"
usefieldsizegroup="false"/>
                            <tablecol
dataattribute="DESCRIPTION" id="1657127824399"/>
                            <tablecol
dataattribute="USUARIO" id="1657127984393" lookup="VALUELIST"/>
                            <tablecol
dataattribute="CLIENTE" id="1657128027715" lookup="VALUELIST"/>
                            <tablecol
dataattribute="SITE" id="1657128015739"/>
                            <tablecol
filterable="false" id="results_bookmark" mxevent="BOOKMARK"
mxevent_desc="Add to Bookmarks" mxevent_icon="btn_addtobookmarks.gif"
sortable="false" type="event"/>
                        </tablebody>
                    </table>
                </tab>
                <tab id="main" label="Main" type="insert">
                    <section id="1657016533204">
                        <section id="1656933057054">
                            <sectionrow
id="1656933090728">
                                <sectioncol
id="1657129076183">
                                    <textbox dataattribute="WONUM" id="1657125982529" size="5"/>
                                </sectioncol>
                                <sectioncol
id="1656934620097">
                                    <textbox dataattribute="DESCRIPTION" id="1657125985318" size="20"/>
                                </sectioncol>
                                <sectioncol
id="1657128293791">
```

```

        <textbox dataattribute="FECHA" id="1657128377335" size="15"/>
        </sectioncol>
    </sectioncol>
    id="1657128284245">
        <textbox dataattribute="STATUS" id="1657128846630"
        lookup="VALUELIST" size="12"/>
        </sectioncol>
    </sectionrow>
    <sectionrow
    id="1656934663080">
        <sectioncol
        id="1656934900603">
            <textbox dataattribute="USUARIO" id="1657126005862"
            lookup="REVVEHIC" size="15"/>
            </sectioncol>
        <sectioncol
        id="1656934674533">
            <textbox dataattribute="CLIENTE" id="1657126003279"
            lookup="REVVEHIC" size="15"/>
            </sectioncol>
        <sectioncol
        id="1657128798690">
            <multiparttextbox dataattribute="ASSET"
            descdataattribute="ASSET.DESCRPTION" descinputmode="readonly"
            id="1657181369731" size="10"/>
            </sectioncol>
        <sectioncol
        id="1657128784512">
            <textbox dataattribute="SITE" id="1657198340188" size="8"/>
            </sectioncol>
        </sectionrow>
        <blankline
        id="1656948409469"/>
    </section>
    <section id="1656936052726">
        <sectionrow
        id="1656936058071">
            <sectioncol
            id="1656936063519">
                <section id="1656936069809">
                    <sectionrow id="1657185100293">
                        <sectioncol id="1657185105664">
                            <section border="false" id="1657188623356"
                            label="Tareas">
                                <blankline id="1657194839043a"/>
                                <section border="true" id="1657194853085"
                                label="Revisión Neumáticos">
                                    <sectionrow id="1657197981342">

```

```

                                <sectioncol
id="1657197989627">

                                <textbox
dataattribute="PRESION_NEUM" id="1657197996323" size="6"
smartfilloff="false"/>

                                </sectioncol>

                                <sectioncol
id="1657197986056">

                                <textbox
dataattribute="DESGASTE_NEUM" id="1657198003770" lookup="VALUELIST"
size="6"/>

                                </sectioncol>

                                <sectioncol
id="1657197981349">

                                <section
id="1657199124848"/>

                                </sectioncol>

                                </sectionrow>

                                </section>

                                <blankline id="1657194839043b"/>

                                <section border="true" id="1657196153206"
label="Revisión Luces">

                                <sectionrow id="1657198015441">

                                <sectioncol
id="1657198021088">

                                <textbox
dataattribute="LUMINOSIDAD" id="1657198029389" lookup="VALUELIST"
size="6"/>

                                </sectioncol>

                                <sectioncol
id="1657198018314">

                                <section
id="1657199129712"/>

                                </sectioncol>

                                <sectioncol
id="1657198015447">

                                <section
id="1657199133071"/>

```

```

        </sectioncol>

    </sectionrow>

</section>

<blankline id="1657194839043c"/>

    <section border="true" id="1657196192522"
label="Revisión Bateria" sigoption="COCHE">

        <sectionrow id="1657198036843">

            <sectioncol
id="1657198041878">

                <textbox
dataattribute="VOLTAJE_BAT" id="1657198048826" sigoption="COCHE" size="6"/>

            </sectioncol>

            <sectioncol
id="1657198039235"/>

            <sectioncol
id="1657198036850"/>

        </sectionrow>

    </section>

    <blankline id="1657194839043d"/>

    <section border="true" id="1657196192522a"
label="Revisión Aceite" sigoption="MOTO">

        <sectionrow id="1657198056043">

            <sectioncol
id="1657198067088">

                <textbox
dataattribute="NIVEL_ACEITE" id="1657198075626" lookup="VALUELIST"
sigoption="MOTO" size="6"/>

            </sectioncol>

            <sectioncol
id="1657198063857"/>

            <sectioncol
id="1657198056050"/>

        </sectionrow>

    </section>

    <blankline id="1657194839043g"/>

```

```

        <section border="true" id="1657196192522b"
label="Revisión Sistema Frenado">

        <sectionrow id="1657198083378">

            <sectioncol
id="1657198090288">

                <textbox
dataattribute="NIVEL_LIQ_FRENOS" id="1657198100752" lookup="VALUELIST"
size="6"/>

            </sectioncol>

            <sectioncol
id="1657198087593">

                <textbox
dataattribute="DESGASTE_DISCO" id="1657198107337" lookup="VALUELIST"
size="6"/>

            </sectioncol>

            <sectioncol
id="1657198083382">

                <textbox
dataattribute="DESGASTE_PAST" id="1657198117010" lookup="VALUELIST"
size="6"/>

            </sectioncol>

        </sectionrow>

        <sectionrow id="1657198157387">

            <sectioncol
id="1657265242700">

                <textbox
dataattribute="FMAX" id="1657198190411" size="6"/>

            </sectioncol>

            <sectioncol
id="1657198166914">

                <textbox
dataattribute="FMIN" id="1657198197115" size="6"/>

            </sectioncol>

            <sectioncol
id="1657198164565">

                <textbox
dataattribute="FD" id="1657198177621" sigoption="COCHE" size="6"/>

            </sectioncol>

```

```

                                <sectioncol
id="1657198157393">

                                <textbox
dataattribute="FI" id="1657198184175" sigoption="COCHE" size="6"/>

                                </sectioncol>

                                </sectionrow>

                                <sectionrow id="1657198161793">

                                <sectioncol

id="1657198171991"/>

                                <sectioncol

id="1657198169450"/>

                                <sectioncol

id="1657198161797"/>

                                </sectionrow>

                                </section>

                                <blankline id="1657194839043f"/>

                                <section border="true" id="1657196192522c"
label="Revisión Sistema Cadena" sigoption="MOTO">

                                <sectionrow id="1657198125517">

                                <sectioncol

id="1657198133116">

                                <textbox
dataattribute="TENSION_CAD" id="1657198139465" lookup="VALUELIST"
sigoption="MOTO" size="6"/>

                                </sectioncol>

                                <sectioncol

id="1657198130571">

                                <textbox
dataattribute="DESGASTE_COR_PI" id="1657198145931" lookup="VALUELIST"
sigoption="MOTO" size="6"/>

                                </sectioncol>

                                <sectioncol

id="1657198125522">

                                <section

id="1657199226939"/>

                                </sectioncol>

                                </sectionrow>

```



```

        </section>

        <blankline id="1657194839043e"/>

    </section>

</sectioncol>

</sectionrow>

</section>

</sectioncol>
</sectionrow>
</section>
</section>
</tab>
</tabgroup>
</clientarea>
<include controltoclone="pageFooter" id="INCLUDE-
pageFooter"/>
</page>

    <dialog beanclass="psdi.webclient.system.beans.QbeBean"
id="searchmore" inputmode="query" label="More Search Fields">
    <section id="searchmore_1">
        <textbox dataattribute="REV_VEHICULOSID"
id="searchmore_attr1"/>
        <textbox dataattribute="DESCRIPTION"
id="searchmore_attr2"/>
        <textbox dataattribute="ASSET"
id="searchmore_attr3"/>
        <textbox dataattribute="CLIENTE"
id="searchmore_attr4"/>
        <textbox dataattribute="DESCRIPTION_LONGDESCRIPTION"
id="searchmore_attr5"/>
        <textbox dataattribute="DESGASTE_COR_PI"
id="searchmore_attr6"/>
        <textbox dataattribute="DESGASTE_DISCO"
id="searchmore_attr7"/>
        <textbox dataattribute="DESGASTE_NEUM"
id="searchmore_attr8"/>
        <textbox dataattribute="DESGASTE_PAST"
id="searchmore_attr9"/>
        <textbox dataattribute="FD" id="searchmore_attr10"/>
        <textbox dataattribute="FECHA"
id="searchmore_attr11"/>
        <textbox dataattribute="FI" id="searchmore_attr12"/>
        <textbox dataattribute="FMAX"
id="searchmore_attr13"/>
        <textbox dataattribute="FMIN"
id="searchmore_attr14"/>
        <checkbox dataattribute="HASLD"
id="searchmore_attr15"/>
        <textbox dataattribute="LUMINOSIDAD"
id="searchmore_attr16"/>
        <textbox dataattribute="NIVEL_ACEITE"
id="searchmore_attr17"/>
        <textbox dataattribute="NIVEL_LIQ_FRENOS"
id="searchmore_attr18"/>

```

```

        <textbox dataattribute="PRESION_NEUM"
id="searchmore_attr19"/>
        <textbox dataattribute="SITE"
id="searchmore_attr20"/>
        <textbox dataattribute="STATUS"
id="searchmore_attr21"/>
        <textbox dataattribute="TENSION_CAD"
id="searchmore_attr22"/>
        <textbox dataattribute="USUARIO"
id="searchmore_attr23"/>
        <textbox dataattribute="VOLTAJE_BAT"
id="searchmore_attr24"/>
        <textbox dataattribute="WONUM"
id="searchmore_attr25"/>
    </section>
    <buttongroup id="searchmore_buttongroup">
        <pushbutton default="true"
id="searchmore_buttongroup_1" label="Find" mxevent="dialogok"/>
        <pushbutton id="qbe_restoreappdefault_button"
label="Restore Application Defaults" mxevent="qbeclear"/>
        <pushbutton id="qbe_revisebutton" label="Revise"
menutype="SEARCHMOREREVISE" targetid="searchmore"/>
        <pushbutton id="searchmore_buttongroup_3"
label="Cancel" mxevent="qbecancel"/>
    </buttongroup>
</dialog>

    <dialog beaclass="psdi.webclient.system.beans.ViewBookmarksBean"
id="searchbook" label="My Bookmarks" mboname="REV_VEHICULOS">
        <table id="bookmarks_table" inputmode="readonly"
selectmode="single">
            <tablebody displayrowsperpage="20" filterable="true"
id="bookmarks_table_tablebody">
                <tablecol dataattribute="WONUM"
id="bookmarks_tablecolumn_1" mxevent="selectrecord" type="link"/>
                <tablecol dataattribute="description"
id="bookmarks_tablecolumn_2"/>
                <tablecol filterable="false"
id="bookmarks_tablecolumn_3" mxevent="instantdelete" mxevent_desc="Delete"
mxevent_icon="btn_delete.gif" sortable="false" type="event"/>
            </tablebody>
        </table>
        <buttongroup id="bookmarks_2">
            <pushbutton default="true" id="bookmarks_2_1"
label="OK" mxevent="dialogok"/>
        </buttongroup>
    </dialog>
    <configurationblock id="datastore_configurationblock">
    </configurationblock>
</presentation>

```

Anexo III: Script para generar revisiones en la nueva aplicación

```
# Autoscript          NUEVAAPP
# Description          Pasar de WO a Revision
# Author              NORA
# Created              06-07-2022
# Log level            ERROR
# Launch point        WO
#-----
# Version             Date             Changed by             Description
#-----
#          0.1          06/07/22          NORA          Initial version
#-----

from math import ceil, fabs
from java.lang import String, Math, Integer
from java.text import SimpleDateFormat as sdf
from java.util import Date, Calendar
from psdi.mbo import Mbo, MboConstants, SqlFormat
from psdi.server import MXServer
from psdi.security import UserInfo
from psdi.iface.mic import MicService
from psdi.util.logging import MXLogger
from psdi.util.logging import MXLoggerFactory

# Helper function for printing
def log(e):
    logger.debug(launchPoint + ":" + scriptName + (" >> %s" % e) + "\r" )

# Init
mxServer = MXServer.getMXServer()
userInfo = mxServer.getSystemUserInfo()
micService = MicService(mxServer)
micService.init()
micUserInfo = micService.getNewUserInfo()
logger = MXLoggerFactory.getLogger("maximo.script")

log("----- BEGIN SCRIPT NORA1234 -----")

wonum = mbo.getString("WONUM")
asset = mbo.getString("ASSETNUM")
description = mbo.getString("DESCRIPTION")
siteid = mbo.getString("SITEID")
reportdate = mbo.getString("REPORTDATE")
jp = mbo.getString("JPNUM")

if (onadd):

    log("Detecto la integracion y que es un add")

    revision= mxServer.getMboSet("REV_VEHICULOS",userInfo)
    revisionWhere = "1 = 1"
    revision.setWhere(revisionWhere)
    revisionmbo = revision.getMbo(0)

    log ("Detecto el mbo, el vehiculo y creo linea")
```

```

newrevision = revision.add()

newrevision.setValue("DESCRIPTION",description,
MboConstants.NOVALIDATION | MboConstants.NOACCESSCHECK)
newrevision.setValue("ASSET",asset, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
newrevision.setValue("WONUM",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
newrevision.setValue("SITE",siteid, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
newrevision.setValue("FECHA",reportdate, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
newrevision.setValue("STATUS","NUEVO")

if jp == "2363":
    newrevision.setValue("TIPO_VEHIC","COCHE",
MboConstants.NOVALIDATION | MboConstants.NOACCESSCHECK)

elif jp == "2365":
    newrevision.setValue("TIPO_VEHIC","MOTO", MboConstants.NOVALIDATION
| MboConstants.NOACCESSCHECK)

revision.save()

log("----- END SCRIPT NORA4321 -----")

```

Anexo IV: Script para generar mensajes de error

```
# Autoscript                ERRORESVEHICULOS
# Description                Errores aplicación revisión vehículos
# Author                    NORA
# Created                   07-07-2022
# Log level                 ERROR
# Launch point              PRESION, VOLTAJE Y FUERZAS
#-----
# Version      Date      Changed by      Description
#-----
#      0.1      07/07/22      NORA
#      Initial version
#-----
from math import ceil, fabs
from java.lang import String, Math, Integer
from java.text import SimpleDateFormat as sdf
from java.util import Date, Calendar
from psdi.mbo import Mbo, MboConstants, SqlFormat
from psdi.server import MXServer
from psdi.security import UserInfo
from psdi.iface.mic import MicService
from psdi.util.logging import MXLogger
from psdi.util.logging import MXLoggerFactory

# Helper function for printing
def log(e):
    logger.debug(launchPoint + ":" + scriptName + (" >> %s" % e) + "\r" )

# Helper function to set an error
def setMessage(errGroup, errKey, errParams):
    global errorkey, errorgroup, params
    errorkey=errKey
    errorgroup=errGroup
    params=errParams

# Init
mxServer = MXServer.getMXServer()
userInfo = mxServer.getSystemUserInfo()
micService = MicService(mxServer)
micService.init()
micUserInfo = micService.getNewUserInfo()
logger = MXLoggerFactory.getLogger("maximo.script")

presion = mbo.getFloat("PRESION_NEUM")
voltaje = mbo.getFloat("VOLTAJE_BAT")
fi = mbo.getFloat("FI")
fd = mbo.getFloat("FD")
fmax = mbo.getFloat("FMAX")
fmin = mbo.getFloat("FMIN")

log("----- BEGIN123 ERRORES VEHICULOS-----")
log("mboname: %s" % mboname)
log("launchpoint: %s" % launchPoint)
log("user: %s" % user)

if (launchPoint=="ERRORPRESION" and presion>5):
    setMessage("REV_VEHIC", "ErrorPresionSup", [presion])
```

```

log("presion: %s" % presion)

if (launchPoint=="ERRORPRESION" and presion<1):
    setMessage("REV_VEHIC", "ErrorPresionInf", [presion])
    log("presion: %s" % presion)

if (launchPoint=="ERRORVOLTAJE" and voltaje>15):
    setMessage("REV_VEHIC", "ErrorVoltajeSup", [voltaje])
    log("voltaje: %s" % voltaje)

if (launchPoint=="ERRORVOLTAJE" and voltaje<8):
    setMessage("REV_VEHIC", "ErrorVoltajeInf", [voltaje])
    log("voltaje: %s" % voltaje)

if (launchPoint=="ERRORFI" and fi>10):
    setMessage("REV_VEHIC", "ErrorFuerzaSup", [fi])
    log("fi: %s" % fi)

if (launchPoint=="ERRORFI" and fi<0.5):
    setMessage("REV_VEHIC", "ErrorFuerzaInf", [fi])
    log("fi: %s" % fi)

if (launchPoint=="ERRORFD" and fd>10):
    setMessage("REV_VEHIC", "ErrorFuerzaSup", [fd])
    log("fd: %s" % fd)

if (launchPoint=="ERRORFD" and fd<0.5):
    setMessage("REV_VEHIC", "ErrorFuerzaInf", [fd])
    log("fd: %s" % fd)

if (launchPoint=="ERRORFMAX" and fmax>10):
    setMessage("REV_VEHIC", "ErrorFuerzaSup", [fmax])
    log("fmax: %s" % fmax)

if (launchPoint=="ERRORFMAX" and fmax<0.5):
    setMessage("REV_VEHIC", "ErrorFuerzaInf", [fmax])
    log("fmax: %s" % fmax)

if (launchPoint=="ERRORFMIN" and fmin>10):
    setMessage("REV_VEHIC", "ErrorFuerzaSup", [fmin])
    log("fmin: %s" % fmin)

if (launchPoint=="ERRORFMIN" and fmin<0.5):
    setMessage("REV_VEHIC", "ErrorFuerzaInf", [fmin])
    log("fmin: %s" % fmin)

log("----- END SCRIPT -----")

```

Anexo V: Script para generar órdenes de trabajo hijas

```
# Autoscript          WOCHILD
# Description          Crear WO Child
# Author              NORA
# Created              10-07-2022
# Log level            ERROR
# Launch point         REV_VEHICULOS
#-----
# Version             Date             Changed by             Description
#-----
#          0.1          10/07/22          NORA          Initial version
#-----

from math import ceil, fabs
from java.lang import String, Math, Integer
from java.text import SimpleDateFormat as sdf
from java.util import Date, Calendar
from psdi.mbo import Mbo, MboConstants, SqlFormat
from psdi.server import MXServer
from psdi.security import UserInfo
from psdi.iface.mic import MicService
from psdi.util.logging import MXLogger
from psdi.util.logging import MXLoggerFactory

# Helper function for printing
def log(e):
    logger.debug(launchPoint + ":" + scriptName + (" >> %s" % e) + "\r" )

# Init
mxServer = MXServer.getMXServer()
userInfo = mxServer.getSystemUserInfo()
micService = MicService(mxServer)
micService.init()
micUserInfo = micService.getNewUserInfo()
logger = MXLoggerFactory.getLogger("maximo.script")

log("----- BEGIN SCRIPT NORA CHILD WO -----")

des = mbo.getString("DESCRIPTION")
site = mbo.getString("SITE")
asset = mbo.getString("ASSET")
wonum = mbo.getString("WONUM")
presion = mbo.getFloat("PRESION_NEUM")
desgaste_neum = mbo.getString("DESGASTE_NEUM")
luminosidad = mbo.getString("LUMINOSIDAD")
voltaje = mbo.getFloat("VOLTAJE_BAT")
n_aceite = mbo.getString("NIVEL_ACEITE")
n_liq_frenos = mbo.getString("NIVEL_LIQ_FRENOS")
desgaste_disco = mbo.getString("DESGASTE_DISCO")
desgaste_past = mbo.getString("DESGASTE_PAST")
fi = mbo.getFloat("FI")
fd = mbo.getFloat("FD")
fmax = mbo.getFloat("FMAX")
fmin = mbo.getFloat("FMIN")
tension_cad = mbo.getString("TENSION_CAD")
desgaste_cor_pi = mbo.getString("DESGASTE_COR_PI")
tipo = mbo.getString("TIPO_VEHIC")
```

```
if tipo == "COCHE" and fd != 0.0 and fmax != 0.0:
```

```
    E = ((fi+fd)/850/9.8)*100
    D = ((fd-fi)/fd)*100
    O = ((fmax-fmin)/fmax)*100
```

```
    if E > 58:
        eficacia = "BIEN"
    elif E < 58 and E > 29:
        eficacia = "GRAVE"
    elif E < 29:
        eficacia = "MUY GRAVE"
```

```
    if D < 20:
        desequilibrio = "BIEN"
    elif D > 20 and D < 30:
        desequilibrio = "LEVE"
    elif D > 30 and D < 80:
        desequilibrio = "GRAVE"
    elif D > 80:
        desequilibrio = "MUY GRAVE"
```

```
    if O < 30:
        ovalidad = "BIEN"
    elif O > 30 and O < 55:
        ovalidad = "LEVE"
    elif O > 55:
        ovalidad = "GRAVE"
```

```
elif tipo == "MOTO" and fmax != 0.0:
```

```
    E = ((fmax)/850/9.8)*100
    O = ((fmax-fmin)/fmax)*100
```

```
    if E > 58:
        eficacia = "BIEN"
    elif E < 58 and E > 29:
        eficacia = "GRAVE"
    elif E < 29:
        eficacia = "MUY GRAVE"
```

```
    if O < 30:
        ovalidad = "BIEN"
    elif O > 30 and O < 55:
        ovalidad = "LEVE"
    elif O > 55:
        ovalidad = "GRAVE"
```

```
if (onupdate):
```

```
    #log("Detecto la integracion y que es un add")
```

```
    wo = mxServer.getMboSet("WORKORDER",userInfo)
    wombo = wo.getMbo(0)
```

```
    if presion < 2.2:
```



```

        log("Hinchar ruedas")

        newwoadded = wo.add()

        newwoadded.setValue("JPNUM","2368", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
        newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
        newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
        newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

        wo.save()

        if desgaste_neum == "MAL":
            log("Recambio ruedas")

            newwoadded = wo.add()

            newwoadded.setValue("JPNUM","2367", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
            newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
            newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
            newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

            wo.save()

            if luminosidad == "MAL":
                log("Recambio luces")

                newwoadded = wo.add()

                newwoadded.setValue("JPNUM","2369", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

                wo.save()

            if n_liq_frenos != "100" or n_liq_frenos != "75" or desgaste_disco ==
"MAL" or desgaste_past == "MAL":
                log("Sistema de frenado")

                newwoadded = wo.add()

                newwoadded.setValue("JPNUM","2373", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

```

```

        newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

        wo.save()

        if tipo == "COCHE":

            if (voltaje <= 12 and voltaje > 10):
                log("Atender estado bateria")

                newwoadded = wo.add()

                newwoadded.setValue("JPNUM","2377", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION
| MboConstants.NOACCESSCHECK)
                newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

                wo.save()

            elif (voltaje <= 10):
                log("Cambiar bateria")

                newwoadded = wo.add()

                newwoadded.setValue("JPNUM","2370", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION
| MboConstants.NOACCESSCHECK)
                newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

                wo.save()

            if desequilibrio == "GRAVE" or desequilibrio == "MUY GRAVE" or
eficacia == "GRAVE" or eficacia == "MUY GRAVE" or ovalidad == "GRAVE":
                log("Sistema de frenado")

                newwoadded = wo.add()

                newwoadded.setValue("JPNUM","2373", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION
| MboConstants.NOACCESSCHECK)
                newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

                wo.save()

        if tipo == "MOTO":

            if n_aceite == "0" or n_aceite == "25":
                log("Recambio aceite")

```

```

        newwoadded = wo.add()

        newwoadded.setValue("JPNUM","2372", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
        newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
        newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION
| MboConstants.NOACCESSCHECK)
        newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

        wo.save()

        if eficacia == "GRAVE" or eficacia == "MUY GRAVE" or ovalidad ==
"GRAVE":

            log("Sistema de frenado")

            newwoadded = wo.add()

            newwoadded.setValue("JPNUM","2373", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
            newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
            newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION
| MboConstants.NOACCESSCHECK)
            newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

            wo.save()

            if (tension_cad == "MAL" or desgaste_cor_pi == "MAL"):
                log("Sistema cadena")

                newwoadded = wo.add()

                newwoadded.setValue("JPNUM","2374", MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("SITEID",site, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)
                newwoadded.setValue("ASSETNUM",asset, MboConstants.NOVALIDATION
| MboConstants.NOACCESSCHECK)
                newwoadded.setValue("PARENT",wonum, MboConstants.NOVALIDATION |
MboConstants.NOACCESSCHECK)

                wo.save()

```

Bibliografía

1. **Universidad de Alcalá.** LAS PRÁCTICAS DE DIGITALIZACIÓN IMPLEMENTADAS EN LAS EMPRESAS. *Revista Prisma Social*. [En línea] 21 de Diciembre de 2020.
<https://dialnet.unirioja.es/servlet/articulo?codigo=7742143>.
2. **Martínez, Adriana Martínez.** Impacto de la COVID-19 en la producción, empleo y digitalización de empresas en Guanajuato: una primera aproximación. *SCIELO*. [En línea] 18 de Mayo de 2021. http://www.scielo.org.mx/scielo.php?pid=S2007-07052021000300201&script=sci_arttext.
3. **Delgado, Tatiana Fernández.** Taxonomía de Transformación Digital. *Revista de Transformación Digital Cubana*. [En línea] 30 de Marzo de 2020.
https://www.researchgate.net/profile/Tatiana-Delgado-3/publication/341051958_Taxonomia_de_transformacion_digital/links/5eab007c92851cb26768fa55/Taxonomia-de-transformacion-digital.pdf.
4. **IBERDROLA.** OPTIMIZACIÓN DEL MANTENIMIENTO. [En línea]
https://d1wqtxts1xzle7.cloudfront.net/36903895/_09__RCM_en_MAXIMO_IBERDROLA-with-cover-page-v2.pdf?Expires=1652864667&Signature=adQFDRZRZXZDMvcz8xa8UXVuqgS8rs9VBIN7RwLv92S4csEvRkiCMo2J1ljF8iva1EEa4ytX5ibXmeOAIdtiu2rTRbembu24EHfvBd3f4h~WQGbOCqT6ZQI553N6Ouac.
5. **Grupo SANVALERO.** Gestión de Mantenimiento. *Estudios abiertos SEAS*. [En línea]
https://d1wqtxts1xzle7.cloudfront.net/36903895/_09__RCM_en_MAXIMO_IBERDROLA-with-cover-page-v2.pdf?Expires=1652864667&Signature=adQFDRZRZXZDMvcz8xa8UXVuqgS8rs9VBIN7RwLv92S4csEvRkiCMo2J1ljF8iva1EEa4ytX5ibXmeOAIdtiu2rTRbembu24EHfvBd3f4h~WQGbOCqT6ZQI553N6Ouac.
6. **Iberdrola.** Mantenimiento predictivo: la técnica basada en datos clave para anticipar errores. *Iberdrola*. [En línea] <https://www.iberdrola.com/innovacion/mantenimiento-predictivo>.
7. **IBM.** IBM Maximo Application Suite. *IBM*. [En línea] <https://www.ibm.com/es-es/products/maximo>.