



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Diseño de una base de datos para la evaluación y seguimiento de la calidad de los proveedores en FORD Almussafes

Trabajo Fin de Grado

Autor: Sadi M. Yilmaz Soñora

Curso: 2018/2019

Titulación: Ingeniería Electrónica Industrial y Automática

Cotutores: Óscar Trull Domínguez | Carmen Romero Montenegro(FORD)

Índice

1. Introducción y objetivos del trabajo.....	2
2. Situación actual y requerimientos	2
1.1 Factoría Ford Almussafes.....	2
1.2 El departamento de Incoming y su función.....	3
1.3 Necesidad de control.....	4
3. Sistema de penalizaciones por puntos.....	5
3.1 Funcionamiento.....	5
3.2 Parámetros de la fórmula.....	6
3.3 Ejemplos.....	8
4. Estructura de la base de datos.....	9
4.1 Introducción.....	9
4.2 Tablas.....	9
4.3 Código principal.....	12
5. Utilidades programadas.....	26
5.1 Exportación de datos.....	26
5.2 Actualizaciones.....	32
5.3 Futuro desarrollo.....	34
6. Conclusión.....	37
7. Bibliografía.....	38

1. Introducción y objetivos del trabajo

El proyecto consiste en el desarrollo e implementación de un sistema de medición y regulación del rendimiento de las empresas terceras que trabajan en Ford. Para lograrlo, se programará una base de datos en Microsoft Access 2016 que tenga integrado el sistema y almacene los datos necesarios.

El programa está en uso actualmente y en constante desarrollo mediante actualizaciones según nuevas necesidades. Está programado de forma que sea dinámico y fiable asegurando que el departamento de IT de la factoría pueda añadir nuevas funcionalidades en un futuro con facilidad.

2. Situación actual y requerimientos

2.1 Factoría Ford Almussafes

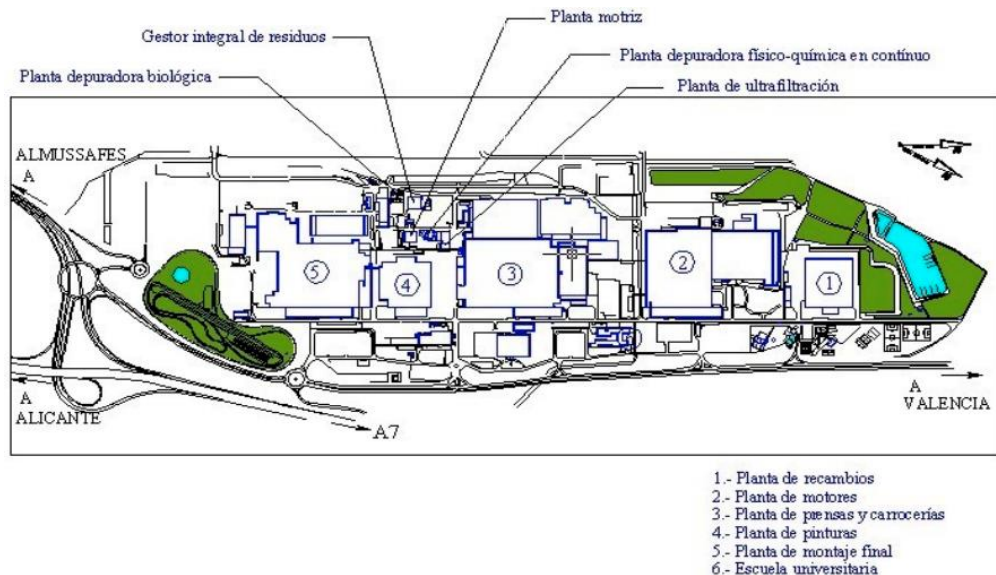


Ilustración 1. Mapa de la factoría FORD Almussafes.

La fábrica de FORD Almussafes se inauguró en 1976, con el lanzamiento del Ford Fiesta. Actualmente se fabrican los modelos *Kuga*, *Mondeo*, *Galaxy*, *S-max* y *Transit Connect*.

La factoría dispone de **5 plantas**, ocupando un total de 2.7 millones de metros cuadrados:

-Planta de montaje: Se montan los coches, mediante una producción en cadena. Partiendo de la carrocería hasta acabar el exterior, los coches salen acabados y listos para vender.

-**Planta de recambios:** Se almacenan los repuestos de piezas.

-**Planta de motores:** Se montan los motores, la mayoría de los cuales no se utilizan en la planta de montaje, sino que se exportan a países como Estados Unidos.

-**Planta de pinturas:** Se pintan las carrocerías.

-**Planta de prensas y carrocerías:** Se trabaja el acero y se forman las carrocerías.

Alrededor de la planta se encuentra un polígono industrial, en el cual se sitúan empresas auxiliares a Ford que suministran piezas directamente a través de túneles de transporte con balancinas consiguiendo un JIT o “Just in time”, producción sin necesidad de almacenamiento.

2.2 El departamento de Incoming y su función

Incoming (“recepción” en inglés) se encarga de la administración y calidad de las piezas recibidas por proveedores. Esta formado por un equipo de inspectores e ingenieros separados en dos grupos, un grupo de inspección que se encarga de la investigación del material posiblemente afectado y determinar la causa del problema y otro grupo de gestión que comunica con los proveedores y dicta los pasos a seguir tras analizar el problema.

La inspección de los lotes de piezas está estandarizada mediante las normas **UNE-ISO 2859-1:2012** y **UNE-ISO 3951-1:2012** para asegurar un balance adecuado entre nivel de calidad del producto y riesgo al proveedor. Se escoge una muestra del lote para inspeccionar (la cantidad escogida depende del grado de calidad que se quiere conseguir) y si se alcanza un número de piezas malas se rechaza el lote entero. Adicionalmente se utiliza un análisis estadístico para determinar el grado de inspección necesario según el historial del proveedor, de forma que la inspección es más rigurosa si ha habido lotes rechazados de la misma pieza recientemente.

El método de muestreo de lotes consigue un ahorro en el tiempo y coste de inspección sobre el método de análisis del lote por completo aunque permite la entrada ocasional de piezas en mal estado. Cuando un operario se encuentra con una pieza en mal estado se inicia una inspección de la causa del problema. Si se determina que el problema viene del proveedor se utilizan planos y especificaciones de la pieza para decidir que la pieza está fuera de la especificación diseñada. De modo que el departamento comunica al proveedor de la pieza que ha habido un *Rechazo* y las medidas de contención que debe adoptar siendo el responsable de las medidas adoptadas el proveedor, que deberá contactar a unas empresas terceras en Ford para realizar la inspección y contención del error. El grado de contención depende de factores como el número de piezas afectadas, el punto de detección del problema, severidad del problema, etc. Existen diferentes inspecciones de calidad y pruebas en diferentes puntos de la línea de montaje (FCPA, CAL, REP, FAL...) que logran minimizar la posibilidad de escape mediante un sistema de alertas. Por ejemplo una detección al final de la línea de montaje resultaría en una Alerta de FAL con medidas de contención la inspección completa del stock de la pieza, inspección al final de la línea y de todos los vehículos ya montados situados en la campa. Se inspeccionarán también todas las piezas recibidas hasta que lleguen con un certificado del proveedor que asegure las piezas en buen estado. Los costes de las contenciones y la pieza rechazada son cobrados al proveedor.

Para documentar todas las medidas adoptadas, el proveedor deberá enviar un **documento 8D**, que detalla los 8 pasos + disciplinas:

- D1: ESTABLECER UN GRUPO PARA SOLUCIÓN DEL PROBLEMA
- D2: DESCRIPCIÓN DEL PROBLEMA
- D3: DESARROLLAR UNA SOLUCIÓN TEMPORAL (ICA)
- D4: ANÁLISIS DE CAUSA RAÍZ
- D5: DESARROLLAR SOLUCIONES PERMANENTES (PCA)
- D6: IMPLEMENTAR Y VALIDAR SOLUCIONES
- D7: PREVENIR LA RECURRENCIA
- D8: CERRAR EL PROBLEMA Y RECONOCER CONTRIBUCIONES

El documento debe recibirse completo dentro de un intervalo de 1 mes desde el rechazo inicial pero se urge el envío del documento relleno hasta el D3 lo antes posible para proteger el producto final. Una vez enviado debe ser revisado y aprobado por el departamento de Incoming.

Las medidas de contención se detallan en el D3, también llamado ICA o Interim Containment Action, y son llevadas a cabo por empresas terceras ya que FORD no ofrece servicios de *sorting* (comprobación del estado) de piezas. El proveedor se pone en contacto con una de las 8 empresas disponibles y contrata el *sorting* y *rework* de la pieza en las instalaciones de FORD.

Estas empresas también pueden realizar reparaciones o *retrabajos* (del inglés *Reworks*) de las piezas afectadas redactando un documento **QPS**, Quality Process Sheet, que indica los pasos para la reparación y debe ser aprobado por el departamento de Incoming y el departamento responsable de la pieza (carrocerías, eléctrica, etc.). Una vez se realiza la reparación, las piezas se reintroducen a la línea de montaje para su nuevo uso.

2.3 Necesidad de control

Los retrabajos realizados por las empresas terceras no están controlados, las piezas se reintroducen en la línea de montaje sin un control de calidad previo a veces dando lugar al rechazo de las mismas piezas y disminuyendo la calidad del producto final.

No existe una responsabilidad en estas empresas, los errores que se cometen no tienen repercusión y los costes de ellos son absorbidos por Ford. El responsable de la planta de montaje ha concluido que para proteger la calidad del producto se debe crear un nuevo sistema con el objetivo de monitorizar y gestionar los fallos cometidos por estas empresas y su rendimiento. El sistema diseñado conseguirá mejorar la calidad de los retrabajos, penalizar la mala actuación de las empresas y minimizar costes de gestión de fallos.

Se establecerá un sistema de penalizaciones por puntos, si una empresa se queda sin puntos de calidad (Quality Points) será penalizada con la imposibilidad de realizar retrabajos durante un periodo de tiempo. De esta forma se asegura una mayor regulación y la calidad óptima de los retrabajos realizados.

Para poder almacenar todos los datos y visualizarlos de forma sencilla, se programará una base de datos relacionando los datos sobre rechazos almacenados en los servidores SQL de Ford y nuevos datos sobre los fallos en los retrabajos de estas empresas.

3. Sistema de penalizaciones por puntos

Se ha diseñado un **nuevo sistema por puntos** con la ayuda de los ingenieros del departamento, con el objetivo de modelizar y monitorizar el rendimiento de las empresas de retrabajo.

3.1 Funcionamiento

Cada empresa de retrabajos dispondrá de **10 Quality Points** iniciales, cada penalización a un retrabajo realizado supondrá una pérdida de puntos. Para recuperar puntos se establece un ciclo de **6 meses**, pasados 6 meses **sin recibir penalizaciones** se reiniciarán los puntos a 10. Si por el contrario se pierden todos los puntos la empresa será penalizada sin poder retrabajar en la factoría un periodo de tiempo dado por la cantidad de veces que se ha quedado sin puntos en el mismo ciclo, según la tabla:

Penalización según veces sin puntos:
1 semana
2 semanas
1 mes
2 meses
4 meses
6 meses
1 año

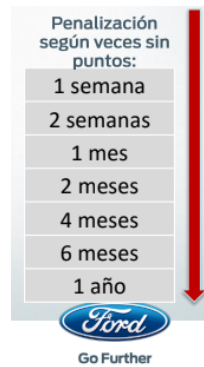


Tabla 1. Periodo de penalización según veces sin puntos.

La cantidad de puntos que se pierden en una penalización viene dada por la fórmula diseñada con la ayuda de los ingenieros del departamento de Incoming:

$$Q = \sqrt{N \cdot P^2 \cdot \alpha^R}$$

Con Q suponiendo una pérdida de **QP** (Quality Points) según la tabla:

Q	
Q = 0	0 pts
$Q \leq 12$	- 1 pts
$12 < Q \leq 25$	- 2 pts
$25 < Q \leq 50$	- 4 pts
$50 < Q \leq 90$	- 6 pts
$90 < Q \leq 150$	- 8 pts
$150 < Q$	- 10 pts

Tabla 2. Cantidad de QP restados según valor de Q.

3.2 Parámetros de la fórmula

Son los factores que definen la penalización y la cantidad de puntos perdidos. Los parámetros son independientes entre ellos. Son cuatro, tres objetivos y uno subjetivo.

N – Número de partes/vehículos afectados

Es la cantidad de partes rechazadas, en caso de que el rechazo sea de vehículos montados, será la cantidad de vehículos rechazados. Este parámetro es objetivo, la cantidad viene dada en el rechazo y su valor es un entero positivo mayor de 0. De los parámetros de la fórmula, es el que menos varía Q, siendo su derivada parcial la raíz cuadrada de N.

P – Punto de detección

Un valor del 1 al 8 según donde se ha detectado la parte defectuosa, según la tabla de valores:

1.	Almacén.
2.	Punto de uso.
3.	Montado en vehículo
4.	OK-line/ FAL
5.	FCPA/GRC/4Poster...
6.	After Station-7 (TSI)
7.	Campa/Puertos
8.	StopShip

Tabla 3. Valor de P según punto de detección.

Lógicamente el valor de P aumenta cuanto más cerca del final de la línea de montaje. El fallo se puede detectar en los múltiples controles de calidad desde el almacén hasta el StopShip (vehículo vendido y pasa a ser del cliente). Como en la fórmula P está elevado al cuadrado, se puede factorizar fuera del radical, de forma que su derivada parcial será P.

α – Severidad

Un parámetro subjetivo, su valor será diferente según el tipo de severidad:

$\alpha =$	Severidad C $\rightarrow \alpha = 0$
	Severidad B $\rightarrow \alpha = 3.2$
	Severidad A $\rightarrow \alpha = 9.6$
	Severidad BLITZ $\rightarrow \alpha = 38.8$

Tabla 4. Valor de alfa según severidad.

Las diferentes severidades corresponden con las severidades de las alertas de calidad emitidas por FCPA (control de calidad a vehículos ya montados y fuera de la línea de montaje) aunque los valores de cada tipo están diseñados para mejorar como afecta a la fórmula. Como es un parámetro subjetivo, el penalizador decidirá el valor apropiado para cada penalización.

Una severidad tipo C corresponde a una penalización que no supone una pieza defectuosa o fuera de especificación, sino un defecto reclamable por un cliente, como por ejemplo la tonalidad del color de la pieza. Se asigna un valor de α de 0, de forma que Q será 0 cuando haya severidad C. Aunque no suponga una pérdida de puntos, al ser una penalización reiniciará el contador de ciclo de 6 meses para la recuperación de puntos.

Severidades B y A tienen valores de α de 3.2 y 9.6 respectivamente, siendo la severidad B la más común y la A normalmente para todos los casos más graves que los de severidad B.

La severidad BLITZ indica una gravedad extrema, un defecto a gran escala que pone en peligro la calidad a un nivel mucho mayor que los de severidad A. Su valor en α es de 38.8, muy superior al 9.6 de la severidad A. Una severidad de este tipo afecta mucho al valor de Q.

R – Repetitividad

Es la cantidad de veces que la misma empresa ha repetido el mismo defecto en un mismo ciclo. En principio es un parámetro objetivo, pero debido a la inmensa cantidad de piezas diferentes y diferentes defectos, que se repita el mismo defecto más de una vez por ciclo es muy improbable. Por tanto, el valor lo establecerá el penalizador siendo conocedor de las anteriores penalizaciones a la empresa de retrabajos en un mismo ciclo y de sus similitudes.

R es el parámetro que más afecta al valor de Q, marca el valor de productos de α dentro del radical. De forma que, si la severidad es alta y es una penalización repetitiva, el valor de Q será muy alto.

3.3 Ejemplos

A continuación se presentan ejemplos reales de fallos en retrabajos:

		Q	QP
CASO 1	Severidad B	5,1	-1
	2. Point of fit		
	2 partes		
	Repet 1		

Defecto: Guías fuera de especificación después de sorting. Valor especificado: $17,4 > x < 18,8$. Medido: 17,1 y 17,02.

La severidad del problema es baja porque se ha detectado en la línea de montaje en el punto de uso. Teniendo en cuenta la magnitud del stock (pueden haber más de 1000 piezas de un mismo modelo) el escape de 2 piezas por descuido es probable. El problema resulta en una penalización de **1 QP**.

		Q	QP
CASO 2	Severidad A	21,5	-2
	4. FAL		
	3 partes		
	Repet 1		

Defecto: FELT MAL POSICIONADO EN GUANTERA (REWORK FAILURE). AFECTANDO GRAN PARTE DE LA CAMPA.

En este caso la severidad escogida es de tipo A teniendo en cuenta que el defecto se ha detectado al final de la línea de montaje (FAL) y ha afectado gran parte de la campaña (almacén de coches montados). El montaje incorrecto ha sido causa de la mala actuación del personal de la empresa de retrabajado resultando en una pérdida de **2 QP**.

		Q	QP
CASO 3	Severidad C	0	0
	4. FAL		
	1 partes		
	Repet 1		

Defecto: Revisión mal realizada, pieza NOK detectada en FAL, siendo posterior al VIN de corte.

La severidad es de tipo C ya que se trata de una sola pieza que ha escapado la revisión del stock, siendo detectada en la revisión de final de línea como medida de contención del problema y realizada por la misma empresa. Ha escapado una revisión pero se ha detectado en la siguiente sin más casos por tanto no resulta en pérdida de puntos.

4. Estructura de la base de datos

4.1 Introducción

La base de datos se programará en Microsoft Access 2016, ya que es la versión instalada en los ordenadores y portátiles que pertenecen a Ford. Se hará uso del servidor SQL que almacena información sobre todos los rechazos generados por el departamento de Incoming.

El funcionamiento al que se quiere llegar:

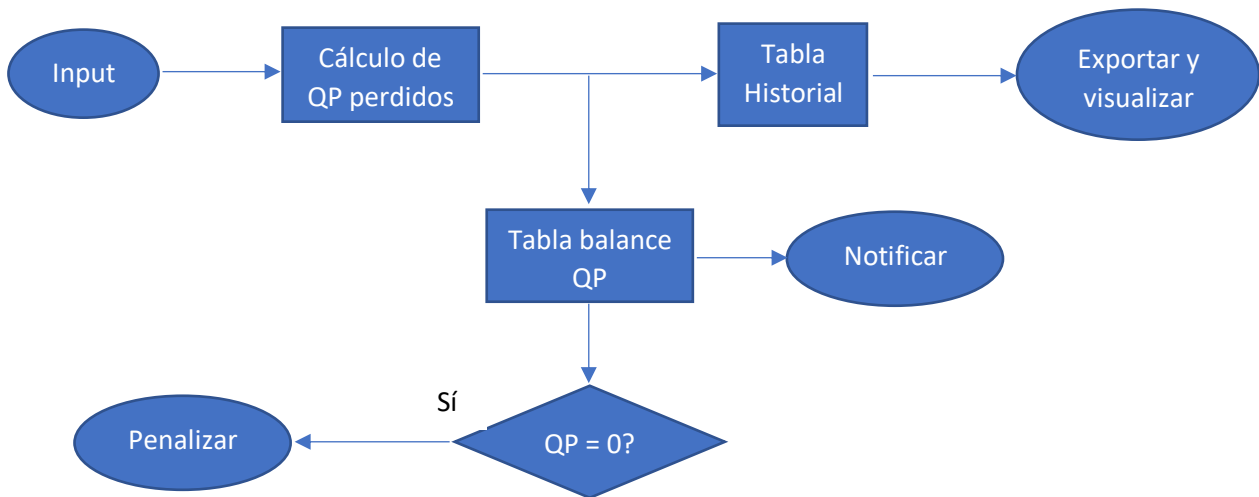


Ilustración 2. Diagrama de funcionamiento de la base de datos.

4.2 Tablas

-Tabla Historial

En esta tabla se almacenarán los datos sobre penalizaciones realizadas, serán necesarios para la exportación y visualización de datos y para llevar un seguimiento de la evolución de los puntos de cada empresa.

Diseño de las columnas de la tabla

Las columnas de la tabla Historial serán los datos que queremos guardar de cada penalización, en primer lugar, se añade una columna "PRIMARY KEY" o clave principal, que servirá para identificar cada penalización y no podrá ser la misma para 2 penalizaciones diferentes. A este campo lo llamaremos NumPenal y se establecerá en Autonumeración, de forma que cada nuevo registro añadido a la tabla se le asigne un NumPenal mayor que el anterior.

En la segunda columna de la tabla se utilizará un campo de Texto Corto llamado RW para identificar la empresa de reemplazo que se penaliza, será una abreviación del nombre completo de la empresa.

La tercera columna será el código de rechazo, CodRechazo, es la PRIMARY KEY de la tabla SQL que contiene la información de rechazos, un campo varchar(6) con un código del tipo:

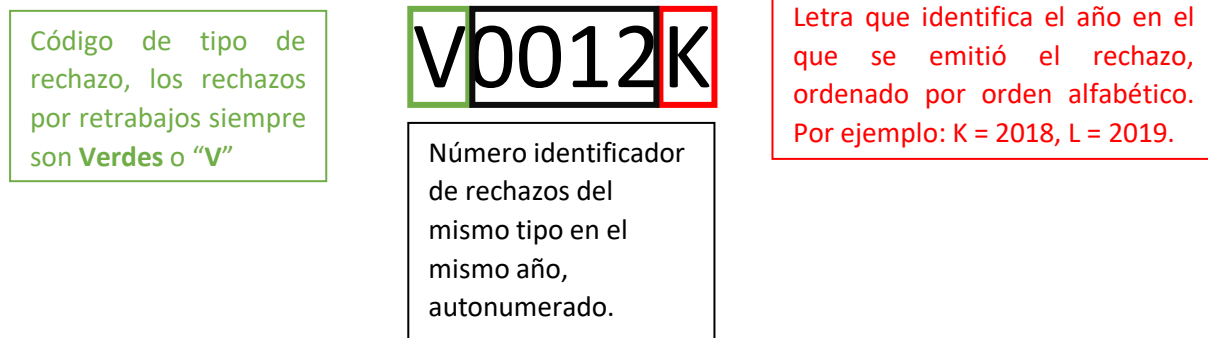


Ilustración 3. Estructura de código de rechazo.

Las cuatro siguientes columnas serán los parámetros introducidos para el cálculo de los puntos perdidos: N, P, Alfa, R. La siguiente columna será la cantidad de puntos perdidos por la penalización PuntosRestados, y la que viene la cantidad de puntos restantes a la empresa tras la penalización, PuntosAct.

Por último, la fecha de penalización se guardará en un campo tipo Fecha llamado FechaPenal, será la misma fecha que la del rechazo si hay un rechazo adjunto a la penalización, sino se introducirá al realizarla.

La tabla diseñada resultante:

Historial	
Nombre del campo	Tipo de datos
NumPenal	Autonumeración
RW	Texto corto
CodRechazo	Texto corto
NumPart	Número
PuntoDetec	Texto corto
Severidad	Texto corto
Repetitividad	Número
PuntosRestados	Número
PuntosAct	Número
FechaPenal	Fecha/Hora

-Tabla Balance de Puntos (BalPuntos)

La tabla BalPuntos se utilizará para guardar información sobre el estado actual del balance de puntos, siendo actualizada periódicamente. También almacenará las diferentes empresas que realizan retrabajos en Ford, siendo posible agregar o eliminar según sea necesario.

Diseño de las columnas de la tabla

De forma similar a la tabla Historial, la primera columna se reserva para un número de identificación, el campo Id, que servirá de clave principal y se establecerá en Autonumeración.

Para identificar las diferentes empresas de retrabajos se utilizará un campo de texto como en la tabla Historial, el campo EmpresaRW. También se almacenará el código de proveedor de Ford de cada empresa, que servirá para determinar quien ha realizado el retrabajo desde la información de rechazos del servidor interno de Ford.

Se añadirá un campo numérico llamado Puntos para guardar la cantidad de puntos, otro campo numérico para la cantidad de veces que se ha quedado sin puntos una empresa llamado VecesPen, y un campo Booleano "Penalizado" para indicar si esta en período de penalización sin poder realizar retrabajos.

De forma adicional se almacena también la fecha de última actualización "UltimaActualizacion" y otro campo para indicar cuando se reiniciarían los puntos o se acabaría la penalización.

La tabla BalPuntos diseñada:

	Nombre del campo	Tipo de datos
Id	Id	Autonumeración
EmpresaRW	EmpresaRW	Texto corto
Puntos	Puntos	Número
UltimaActualizacion	UltimaActualizacion	Fecha/Hora
codprov	codprov	Texto corto
VecesPen	VecesPen	Número
DaysLeft	DaysLeft	Texto corto
Penalizado	Penalizado	Sí/No

4.3 Código principal

Como el funcionamiento principal del programa es recoger datos, se empezará programando la pantalla de input de datos, se diseña un nuevo formulario llamado Nueva_Penal que servirá de pantalla para crear penalizaciones y calcular los puntos.

Nueva Penalización

Código de Rechazo Verde

Empresa RW

Numero de partes afectadas

Punto de detección

Severidad

Repetitividad

Desde esta pantalla se introduce el código de rechazo verde anteriormente creado que corresponde con un fallo por retrabajo de alguna de las empresas de retrabajo. Pulsando el botón Buscar se busca el rechazo correspondiente en las tablas SQL del servidor interno de Ford mediante una conexión ODBC (Open DataBase Connectivity).

```
Private Sub BtonBuscar_Click()  
  
    On Error GoTo salir  
  
    If IsNull(Me.CodVerde) Then  
  
        MsgBox ("Introduzca el código de rechazo")  
  
        GoTo salir  
  
    End If
```

Si el campo de texto CodVerde esta vacío se emite un aviso y se aborta la operación.

```
Dim rst As DAO.Recordset
```

```
Set rst = CurrentDb.OpenRecordset("SELECT dbo_Rechazos.defectuosas, dbo_Rechazos.[Fecha rechazo],
    dbo_Rechazos.[Codigo proveedor]," _
    & "BalPuntos.EmpresaRW FROM dbo_Rechazos LEFT JOIN BalPuntos ON dbo_Rechazos.[Codigo proveedor] =
    BalPuntos.codprov" _ & "WHERE dbo_Rechazos.[Numero rechazo] ='" & Me.CodVerde & "' AND
    dbo_Rechazos.[Numero rechazo] LIKE 'V%'", dbOpenSnapshot)
```

Se extrae la información necesaria de la tabla SQL `dbo_Rechazos` con una instrucción `SELECT`: la fecha del rechazo emitido, la cantidad de piezas afectadas y el código de proveedor.

Se realiza también un `LEFT JOIN` a nuestra tabla `BalPuntos` donde coincidan el campo [código proveedor] de la tabla `Rechazos` y el campo [codprov] de la tabla `BalPuntos`. Un `LEFT JOIN` relaciona las dos tablas según una condición y siempre devolverá las filas de la primera tabla, aunque no se cumpla la condición en la segunda, de forma que si no coincidiese el código de proveedor con alguno de la tabla `BalPuntos` se devolvería Nulo el campo `EmpresaRW`.

Para el criterio se utiliza la sentencia `WHERE`, donde el [Numero rechazo] será el introducido en el campo de texto por el usuario. Como en la tabla `rechazos` se guarda la información de todos los tipos de rechazo, limitamos la búsqueda solo para rechazos verdes, cuyos [Numero rechazo] empiezan por V, mediante el operador `LIKE` se seleccionan solo los que su primer letra es V.

El tipo de recordset será `dbOpenSnapshot` ya que solo queremos extraer información y no hace falta editar.

```
If rst.EOF Then
    MsgBox ("No se encuentra el rechazo verde")
    rst.Close
    GoTo salir
End If
```

`Rst.EOF` (End of file del inglés) es un boolean que se vuelve `True` cuando no hay más información en el recordset. Si no hay ningún registro con esos criterios no existe el rechazo en la tabla `Rechazos`, se podrá penalizar igualmente, pero rellenando todos los campos manualmente.

```
Me.NumParts = rst![defectuosas]

fecharech = rst![Fecha rechazo]

Me.RWcomp = rst!EmpresaRW

rst.Close

salir:

    Set rst = Nothing

End Sub
```

Los datos extraídos se sitúan en los campos correspondientes, el número de partes afectadas en el campo de texto NumParts y la empresa de retrabajos en la lista desplegable RWcomp. La fecha se almacena en una variable global de tipo Fecha del formulario, para poder usarla posteriormente.

Se pulsará el botón Aceptar una vez se hayan rellenado todos los campos:

```
Private Sub ButAcep_Click()

    On Error Resume Next

    Dim conrech As Boolean

    If IsNull(Me.NumParts) Then

        MsgBox ("Numero de partes afectadas no válido")

        GoTo salir

    End If

    If IsNull(Me.Repet) Then

        MsgBox ("Repetitividad no válido")

        GoTo salir

    End If

    If IsNull(Me.Ptdetec) Then

        MsgBox ("Punto de detección no válido")

        GoTo salir

    End If
```

```

If IsNull(Me.Severity) Then

    MsgBox ("Severidad no válida")

    GoTo salir

End If

If IsNull(Me.RWcomp) Then

    MsgBox ("No existe la empresa RW seleccionada")

    GoTo salir

End If

```

Se realiza una comprobación inicial de todos los campos, si alguno no se ha rellenado correctamente se avisa al usuario y se aborta la operación.

```

conrech = True

If IsNull(Me.CodVerde) Then

    If MsgBox("Desea emitir una penalización sin el código de rechazo verde?", vbYesNo +
vbQuestion) = vbYes Then

        fecharech = Date

        conrech = False

    Else: GoTo salir

End If

End If

```

Si el campo de código rechazo no se rellena, se puede crear una penalización sin código de rechazo. Utilizaremos la variable Boolean conrech para determinar si tiene o no un código de rechazo verde asociado. Como no se ha extraído la fecha de ningún rechazo, se coge la fecha de la penalización como la fecha actual del usuario.

```

fecha:

fech = InputBox("Introduzca la fecha (dd/mm/yyyy)", "Fecha de penalización", Nz(fecharech, Date))

If (Format(fech, "dd/mm/yyyy") <> fech) Or StrPtr(fech) = 0 Then

    GoTo fecha

End If

```

La fecha se introduce mediante una ventana que tendrá como valor predeterminado la variable Global fecharech, de forma que si se ha buscado la información del rechazo con el botón Buscar, la fecha será la fecha de emisión del rechazo. Se utiliza la función Nz() para indicar un valor predeterminado de fecha actual del usuario si la variable fecharech fuese nula. El campo fecha es obligatorio, si la fecha indicada no es una fecha válida se vuelve a abrir la ventana.

Código de Rechazo Verde

Empresa RW

Numero de partes afectadas

Punto de detección

Severidad

Repetitividad

Fecha de penalización

Introduzca la fecha (dd/mm/yyyy)

Se crea una función `CalcularPuntos()` con argumentos los parámetros de la fórmula que devuelve el número de puntos a restar.

```
QP = CalcularPuntos(Me.NumParts, (Me.Ptdetec.ListIndex + 1), Severity.ListIndex, Me.Repet)
```

```
If conrech Then
```

```
Set rst = CurrentDb.OpenRecordset("SELECT CodRechazo FROM Historial WHERE CodRechazo = " & Me.CodVerde & "", dbOpenSnapshot)
```

```
If Not rst.EOF Then
```

```
If MsgBox("Ya existe una penalizacion para ese rechazo, desea modificarla?", vbYesNo + vbQuestion) = vbYes Then
```

```
If SysCmd(acSysCmdGetObjectState, acForm, "Modificar") <> 0 Then
```

```
DoCmd.Close acForm, "Modificar", acSaveNo
```

```
DoCmd.OpenForm "Modificar", acNormal, , , , Me.CodVerde
```

```
Else
```

```
DoCmd.OpenForm "Modificar", acNormal, , , , Me.CodVerde
```

```
End If
```

```
Else
```

```
rst.Close
```

```
GoTo salir
```

```
End If
```

```
End If
```

```
End If
```

```
Set rst = CurrentDb.OpenRecordset("Historial", dbOpenDynaset)
```

Se utiliza un recordset para determinar si hay alguna penalización anterior en la tabla de Historial con el mismo número de rechazo, si es así no se penaliza y se da la opción de modificar la penalización.

Una vez completados todos los requerimientos se añade un nuevo registro con los datos de la penalización a la tabla Historial y se actualizan los puntos de la empresa afectada en la tabla BalPuntos. También se comprueba si la empresa se ha quedado sin puntos, de ser así se abre una ventana para elegir periodo de penalización:

Penalización

R21

Veces sin puntos:

Periodo

Sin penalizar

1 semana

2 semanas

1 mes

2 meses

4 meses

6 meses

1 año

Desde Fecha

Hoy

Fecha de rechazo

Otra

Desde Fecha:

Fecha Fin:

Se podría automatizar la elección de la penalización, pero para mayor libertad se ha decidido hacerlo manual, de forma que si por algún motivo la penalización es diferente de la que debería ser se pueda cambiar fácilmente. En la ventana se muestra las veces que la empresa se ha quedado sin puntos, una lista de botones de opción para elegir el periodo de penalización y otra para elegir la fecha desde que se penaliza.

```
Select Case Me.DesdeFecha.Value
```

```
Case 1
```

```
fecha = Date
```

```
Case 2
```

```
fecha = CDate(Nz(Me.OpenArgs, 0))
```

```
If fecha = 0 Then Exit Sub
```

```
Case 3
```

```
If IsNull(Me.Fechaini) Then Exit Sub
```

```
fecha = Me.Fechaini
```

```
Case Else
```

```
Exit Sub
```

```
End Select
```

```
diasreinicio = DateDiff("d", Me.Fechaini, Me.txtFechaFin)

If diasreinicio <= 0 Then

    MsgBox "La fecha fin ya ha pasado. Indique una fecha correcta!", vbCritical, "Error en la fecha"

    Exit Sub

End If

If MsgBox("Desea penalizar " & Me.EmpresaRW & " por un total de " & diasreinicio & " días, hasta la fecha " &
Me.txtFechaFin & "?", vbYesNo, "Confirmar") = vbYes Then

    balrst.Edit

    balrst!Penalizado = True

    balrst!DaysLeft = DateAdd("d", diasreinicio, fecha) & "(" & diasreinicio & " días)"

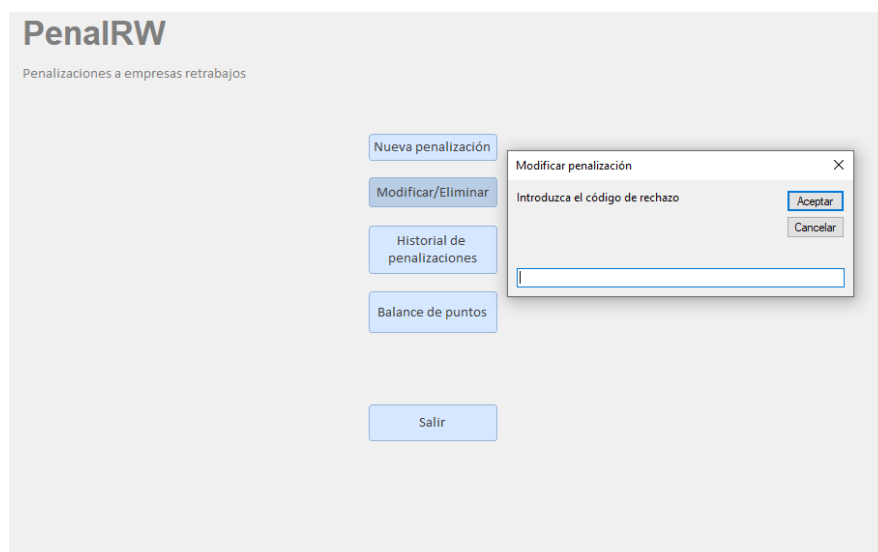
    balrst.Update

End If
```

El cálculo de la fecha final se hace automáticamente una vez se elige un periodo y fecha inicial, al pulsar el botón de Penalizar se actualiza el estado de la empresa de reemplazo en la tabla BalPuntos a Penalizado y el campo de días restantes a los días restantes de penalización. El usuario que ha penalizado se hace responsable de comunicar a la empresa que no va a poder realizar reemplazos hasta la fecha final indicada. El envío del correo se podría haber automatizado pero por simplicidad y por orden de los ingenieros se ha preferido hacerlo manual.

Modificar penalizaciones

Para corregir errores o penalizaciones incorrectas debe haber un sistema de modificación o eliminación de penalizaciones anteriores, desde el menú principal se añade un botón que permite realizar modificaciones a penalizaciones anteriores y ajusta el nivel de puntos según sea necesario.



Para modificar una penalización, se pide el código de rechazo de la penalización mediante un InputBox. Se comprueba que el rechazo indicado exista en la tabla Historial mediante un Recordset y si es así se abre el formulario de modificar con argumentos de apertura el código de rechazo.

Al cargar el formulario, se rellenan los campos con la información sobre la penalización mediante otro Recordset de la tabla Historial con una condición WHERE para que el código de rechazo sea igual al argumento de apertura (OpenArgs) del formulario:



El botón Modificar del formulario actualiza la información en la tabla Historial de la penalización si se ha cambiado algún valor de sus parámetros, para ello se utiliza el evento Dirty del formulario, de forma que se volverá Verdadera una variable booleana llamada “modif”.

```
Private Sub Form_Dirty(Cancel As Integer)
```

```
    modif = True
```

```
End Sub
```

Se ha elegido utilizar el método de Recordset en vez de campos vinculados directamente a la tabla (es posible desde Access) para minimizar errores, problemas cuando dos usuarios editan el mismo registro a la vez, problemas de conectividad con las tablas, etc.

```

ModifPuntos = False

newpuntos = CalcularPuntos(Me.NumParts, (Me.Ptdetec.ListIndex + 1), Me.Severity.ListIndex, Me.Repet)

Set histrst = CurrentDb.OpenRecordset("SELECT * FROM Historial WHERE CodRechazo = " & Codigoverde & "",
    dbOpenDynaset)

If Not histrst.EOF Then

    histrst.Edit

    histrst!NumPart = Me.NumParts

    histrst!PuntoDetec = (Me.Ptdetec.ListIndex + 1) & ". " & Me.Ptdetec

    histrst!Severidad = Me.Severity

    histrst!Repetitividad = Me.Repet

    histrst!PuntosRestados = newpuntos

    histrst.Update

End If

histrst.Close

Set histrst = Nothing

```

Como la tabla Historial tiene una columna de puntos actuales “PuntosAct”, es necesario calcular de nuevo los puntos restados y si el resultado es diferente, cambiar los puntos de la penalización y el valor de la columna PuntosAct de las penalizaciones posteriores.

```

Dim idreset As Integer

If newpuntos <> puntosantes Then

    diffpuntos = puntosantes - newpuntos

    If MsgBox("Los puntos restados pasarán de " & puntosantes & " Puntos a " & newpuntos & " Puntos. Aceptar?",
        vbYesNo + vbQuestion, "Cambio en los puntos") = vbYes Then

```

Se calculan de nuevo los puntos con los argumentos modificados y si son diferentes se comunica al usuario.

```

Set histrst = CurrentDb.OpenRecordset("SELECT NumPenal, RW, PuntosAct FROM Historial WHERE PuntosAct =
    10 AND RW = " & Me.RWcomp & " AND NumPenal > " & idpen & " ORDER BY NumPenal ASC",
    dbOpenSnapshot)

```

Se abre un Recordset de la tabla historial que busca si ha habido un reinicio de puntos después de penalización a modificar. Con la instrucción “ORDER BY NumPenal ASC” se consigue que el primer registro sea el reinicio que sucedió después de la penalización, de forma que solo hay que modificar la columna PuntosAct de las penalizaciones entre la penalización modificada

y el reinicio de puntos. En caso contrario si no hay reinicio, se modifican todas las penalizaciones posteriores a la modificada y también la tabla de balance de puntos BalPuntos:

```

If histrst.EOF Then

    ModifPuntos = True

    SQL = "UPDATE Historial SET PuntosAct = If( PuntosAct + " & diffpuntos & " > 10, 10, PuntosAct + " &
diffpuntos & ") WHERE RW = " & Me.RWcomp & " AND NumPenal >= " & idpen

Else

    idreset = histrst!NumPenal

    SQL = "UPDATE Historial SET PuntosAct = If( PuntosAct + " & diffpuntos & " > 10, 10, PuntosAct + " &
diffpuntos & ") WHERE RW = " & Me.RWcomp & " AND NumPenal BETWEEN " & idpen & " AND " & (idreset -
1)

End If

```

Se utiliza idreset para identificar el reinicio, las penalizaciones a modificar serán las que tengan NumPenal entre idpen y (idreset-1).

```

DoCmd.SetWarnings False

DoCmd.RunSQL (SQL)

DoCmd.SetWarnings True

```

Si la modificación realizada ha causado un cambio en el balance de puntos de la empresa, se edita el valor de puntos en la tabla BalPuntos y se actualiza el estado del boolean Penalizado si hace falta:

```

If ModifPuntos Then

    Set balrst = CurrentDb.OpenRecordset("SELECT * FROM BalPuntos WHERE EmpresaRW = " & Me.RWcomp &
"", dbOpenDynaset)

    pts = balrst!Puntos + diffpuntos

    If pts > 10 Then pts = 10

```

```

balrst.Edit

balrst!Puntos = pts

If pts >= 0 And balrst!Penalizado = True Then

    balrst!Penalizado = False

    balrst!DaysLeft = ""

End If

balrst.Update

End If

msg = MsgBox("Se ha modificado la penalizacion con éxito.", , "Modificación realizada")

```

El botón Eliminar funciona de manera similar, se comprueba si hay un cambio de puntos, y si lo hay se comprueba si hay un reinicio de puntos después de la penalización:

```

updt = False

NoRstPuntos = False

If puntosantes <> 0 Then

    Set histrst = CurrentDb.OpenRecordset("SELECT * FROM Historial WHERE Historial.NumPenal > " & idpen & "
    AND RW = " & Me.RWcomp & " AND PuntosAct = 10 ORDER BY NumPenal ASC", dbOpenSnapshot)

    If histrst.EOF Then

        updt = True

        histrst.Close

        SQL = "UPDATE Historial SET Puntos0Act = IIf( PuntosAct + " & puntosantes & " > 10, 10, PuntosAct + " &
        puntosantes & ") WHERE RW = " & Me.RWcomp & " AND NumPenal > " & idpen

    Else

        updt = True

        NoRstPuntos = True

        idreset = histrst!NumPenal

        SQL = "UPDATE Historial SET PuntosAct = IIf( PuntosAct + " & puntosantes & " > 10, 10, PuntosAct + " &
        puntosantes & ") WHERE RW = " & Me.RWcomp & " AND NumPenal BETWEEN " & idpen & " AND " &
        (idreset - 1)

        histrst.Close

    End If

```

Una vez se han realizado las comprobaciones previas se elimina el registro de la penalización y se ejecuta la instrucción SQL para actualizar la columna PuntosAct de la tabla Historial:

elim:

```
If MsgBox("Desea eliminar la penalización al rechazo " & UCase(Codigoverde) & " ?", vbYesNo + vbQuestion)
= vbYes Then

Set histrst = CurrentDb.OpenRecordset("SELECT * FROM Historial WHERE CodRechazo = " & Codigoverde &
" AND RW =" & Me.RWcomp & """, dbOpenDynaset)

histrst.Delete

histrst.Close

If updt Then

    DoCmd.SetWarnings False

    DoCmd.RunSQL (SQL)

    DoCmd.SetWarnings True
```

Si la penalización eliminada suponía un cargo de puntos y no ha habido un reinicio posterior a ella se suman los puntos al balance en la tabla BalPuntos:

```
If Not NoRstPuntos Then

    Set balrst = CurrentDb.OpenRecordset("SELECT * FROM BalPuntos WHERE EmpresaRW =" &
Me.RWcomp & """, dbOpenDynaset)

    balrst.Edit

    If balrst!Puntos <= 0 Then balrst!VecesPen = balrst!VecesPen - 1

    balrst!Puntos = IIf((balrst!Puntos + puntosantes) > 10, 10, balrst!Puntos + puntosantes)

    balrst.Update

    balrst.Close

End If

End If

MsgBox ("Penalización eliminada.")

DoCmd.OpenForm "Menu"
```


Balance de puntos

Se crea un formulario que contiene la información de la tabla BalPuntos en una interfaz de usuario simple. Se compone de campos vinculados a las columnas de la tabla en vista de formularios continuos, de forma que para cada empresa hay una fila con su información. De modo que si hace falta añadir una nueva empresa a la base de datos se crea un nuevo registro en la tabla con la información necesaria.

Balance de Puntos						
Id	Empresa RW	Puntos	Ultima Actualización	N. Veces sin puntos	Días restantes para reinicio	Penalizado
1	INT	8	06/04/2019	0	17/05/2019 (42 días)	No
2	AKT	7	25/06/2019	0	26/10/2019 (123 días)	No
3	R21	10	18/06/2019	0		No
4	KH	8	02/04/2019	0	11/05/2019 (39 días)	No
5	WP	10	27/02/2019	0		No
6	EA	10		0		No
7	FD	8	11/06/2019	0	28/09/2019 (109 días)	No
8	TRI	10		0		No

Historial de penalizaciones

De forma similar al balance de puntos, se establece un formulario con campos vinculados a la tabla Historial por defecto limitado a los 300 últimos registros para mayor comodidad, aunque se habilita un botón para ver el historial entero:

Historial de penalizaciones									
Id Penalizacion	RW	Cod. Rechazo	Num. partes	Punto de detección	Severidad	Repet.	Fecha Penal.	Ptos. Restados	Ptos. Restantes
34	R21		0			0	21/08/2019	0	10
33	AKT		1	2. PointofFit	B	2	25/06/2019	1	7
32	R21	V0241L	9	4. Ok-Line/FAL	A	0	14/06/2019	1	0
29	FD	V0234L	12	4. Ok-Line/FAL	A	0	11/06/2019	2	8
28	AKT	V0173L	6	4. Ok-Line/FAL	B	0	13/05/2019	1	8
27	R21	V0166L	1	2. PointofFit	C	1	07/05/2019	0	1
26	INT	V0134L	1	4. Ok-Line/FAL	B	1	05/04/2019	1	8
25	INT	V0125L	6	4. Ok-Line/FAL	C	1	03/04/2019	0	9
24	KH	V0120L	1	2. PointofFit	B	1	02/04/2019	1	8
16	KH	V0080L	1	4. Ok-Line/FAL	C	1	05/03/2019	0	9
14	INT	V0068L	4	2. PointofFit	B	1	27/02/2019	1	9
13	AKT	V0072L	1	2. PointofFit	B	1	28/02/2019	1	9
12	R21	V0069L	3	4. Ok-Line/FAL	A	1	27/02/2019	2	1
9	WP	V0042L	1	5. FCPA/GRC/4Poster	B	1	04/02/2019	1	9
7	R21	V0061L	3	2. PointofFit	B	1	20/02/2019	1	3
6	R21	V0052L	1	5. FCPA/GRC/4Poster	B	1	11/02/2019	1	4
5	R21	V0045L	1	3. Montado	B	1	05/02/2019	1	5
4	R21	V0044L	2	2. PointofFit	B	1	05/02/2019	1	6
3	KH	V0010L	1	2. PointofFit	B	1	15/01/2019	1	9
2	R21	V0006L	2	3. Montado	B	1	11/01/2019	1	7
1	R21	V0004L	10	5. FCPA/GRC/4Poster	B	1	10/01/2019	2	8

 Fecha inicial
 Fecha final

 Cod. Rechazo
 Empresa RW

Mostrando últimos 300

Adicionalmente, se añade un sistema de filtrado por código de rechazo y empresa, seleccionar una empresa de la lista desplegable y pulsar el botón Filtrar modifica el origen de registro del formulario principal:

```
If Not IsNull(Me.filtrech) Then

    Me.RecordSource = "SELECT * FROM Historial WHERE (((Historial.NumPenal) > 0)) AND
    CodRechazo = " & Me.filtrech & " ORDER BY Historial.NumPenal DESC;"

    Me.filtrech = Null

    Me.filtrw = Null

    Me.Requery

Elseif Not IsNull(Me.filtrw) Then

    Me.RecordSource = "SELECT * FROM Historial WHERE (((Historial.NumPenal) > 0)) AND
    Historial.RW = " & Me.filtrw & " ORDER BY Historial.NumPenal DESC;"

    Me.filtrech = Null

    Me.filtrw = Null

    Me.Requery

Else

    Me.RecordSource = "SELECT TOP 300 * FROM Historial WHERE (((Historial.NumPenal) > 0)) ORDER BY
    Historial.NumPenal DESC;"

    Me.Requery

End If
```

Pulsar el botón sin indicar código de rechazo o empresa de reemplazo limpia el filtro actual y vuelve al estado predeterminado del formulario.

5. Utilidades programadas

5.1 Exportación de datos

Uno de los objetivos a conseguir mediante la base de datos es visualizar y monitorizar el rendimiento de las empresas terceras que retrabajan piezas en la planta de montaje. Por ello se ha programado una exportación personalizada a Excel accesible desde el formulario de Historial:



Se crea una función ExportarExcel que tiene como argumentos la fecha inicial y la fecha final que se llama al pulsar el botón.

```

Funcion ExportarExcel(Fechaini As Date, Fechafin As Date)

DoCmd.Hourglass (True)

'Exportar todas las penalizaciones entre las fechas seleccionadas

Set rst = CurrentDb.OpenRecordset("SELECT Historial.*, dbo_Rechazos.descripcion FROM Historial LEFT JOIN
    dbo_Rechazos ON Historial.CodRechazo = dbo_Rechazos.[Numero Rechazo]" _
    & "WHERE Historial.[FechaPenal] BETWEEN #" & Format(Fechaini, "mm/dd/yyyy") & "# AND #" &
    Format(Fechafin, "mm/dd/yyyy") & "# ORDER BY [FechaPenal] DESC, [NumPenal] DESC", dbOpenSnapshot)

Set rst2 = CurrentDb.OpenRecordset("BalPuntos", dbOpenSnapshot)

```

Se utiliza un Recordset de la tabla Historial con una instrucción JOIN a la tabla en el servidor SQL Rechazos para añadir también la descripción del defecto a cada rechazo y un Recordset diferente para la tabla BalPuntos. Ambos Recordsets se abren en modo Snapshot (modo lectura) porque solo se requiere lectura de datos en las tablas.

```

Set appExcel = CreateObject("Excel.application")

appExcel.Visible = False

appExcel.Workbooks.Add

Set wksNew = appExcel.Worksheets(1)

```

Una de las funcionalidades del lenguaje VBA es la compatibilidad entre las otras aplicaciones de Microsoft, se utiliza la función CreateObject para crear una instancia de Microsoft Excel y se mantiene oculta al usuario hasta el fin de la función.

Se da formato a las columnas y celdas de la hoja activa utilizando comandos de Excel como ColumnWidth para el ancho de columnas y Range para seleccionar rangos. Se procede a pasar los datos de la tabla Historial a la hoja activa mediante un bucle en el que cada registro es una fila utilizando una variable i como iterador:

```
i = 2

Do While Not rst.EOF

    .Range("A" & i).Value = Nz(rst!NumPenal, 0)

    .Range("B" & i).Value = Nz(rst!RW, "")

    .Range("C" & i).Value = Nz(rst!NumPart, 0)

    .Range("D" & i).Value = Nz(rst!PuntoDetec, "")

    .Range("E" & i).Value = Nz(rst!Severidad, "")

    .Range("F" & i).Value = Nz(rst!Repetitividad, 0)

    .Range("G" & i).Value = Nz(rst!PuntosRestados, 0)

    .Range("H" & i).Value = Nz(rst!PuntosAct, 0)

    .Range("I" & i).Value = Nz(rst!FechaPenal, Empty)

    .Range("J" & i).Value = Nz(rst!descripcion, "")

    .Range("A" & i & ":" & i).HorizontalAlignment = -4108

    .Range("J" & i).WrapText = True

    i = i + 1

    rst.MoveNext

Loop
```

ID	RW	Numero de partes	Punto de deteccion	Severidad	Repetitividad	QP restados	QP	Fecha	Descripción
35	WP	0			0	0	10	21/08/2019	
34	R21	0			0	0	10	21/08/2019	
33	AKT	1	2. PointofFit	B	2	1	7	25/06/2019	
32	R21	9	4. Ok-Line/FAL	A	0	1	0	14/06/2019	Alojamiento para switch pack derecho sin retrabajar. Botón start-stop se atasca. CAL 2019-719
29	FD	12	4. Ok-Line/FAL	A	0	2	8	11/06/2019	Retrabajo mal hecho de FormeID en el aireador central derecho del CD4.2 La alerta de retrabajo es A138766063 piezas mal verificadas dimensionalmente. Especificación: 1291+/-10 mm.
28	AKT	6	4. Ok-Line/FAL	B	0	1	8	13/05/2019	RH: 1267 mm,1270 mm, 1265 mm. LH: 1262 mm, 1260 mm,1255 mm
27	R21	1	2. PointofFit	C	1	0	1	07/05/2019	CROMIADO DEFECTUOSO, PIEZA MAL REVISADA
26	INT	1	4. Ok-Line/FAL	B	1	1	8	05/04/2019	escape por parte de Integrale por stock y por parkings en revision por tonalidad.
25	INT	6	4. Ok-Line/FAL	C	1	0	9	03/04/2019	Aparecen manetas tonalidad NOK despues del VIN de corte
24	KH	1	2. PointofFit	B	1	1	8	02/04/2019	PIEZA MAL REVISADA, TORNILLO EXTRA EN EINTERIOR DEL CALEFACTOR. RETRABAJO REFERENTE AL RECHAZO R0719L
16	KH	1	4. Ok-Line/FAL	C	1	0	9	05/03/2019	Revisión mal realizada, pieza NOK detectada en FAL, siendo posterior al VIN de corte.
13	AKT	1	2. PointofFit	B	1	1	9	28/02/2019	DEFECTO PINTURA, MALA REVISION
14	INT	4	2. PointofFit	B	1	1	9	27/02/2019	Motor de limpieza no se queda en posición 0 tras revisión .(marcas de macheo no coinciden)
12	R21	3	4. Ok-Line/FAL	A	1	2	1	27/02/2019	FELT MAL POSICIONADO EN GUANTERA (REWORK FAILURE). AFECTANDO GRAN PARTE DE LA CAMPA.

Otro bucle añade la información de la tabla BalPuntos y cuenta el número de empresas en la variable NumRW:

```
i = 2
```

```
NumRW = 0
```

```
Do While Not rst2.EOF
```

```
.Range("K" & i).Value = Nz(rst2!Id, 0)
```

```
.Range("L" & i).Value = Nz(rst2!EmpresaRW, "")
```

```
.Range("S" & i).Value = rst2!Puntos
```

```
i = i + 1
```

```
NumRW = NumRW + 1
```

```
rst2.MoveNext
```

```
Loop
```

Por cada empresa se crea una nueva hoja y se abre un Recordset de Historial ordenado por empresa, Fecha, y Numero de penalización respectivamente. Se decide usar un solo Recordset en vez de uno por cada empresa con una condición WHERE para minimizar las consultas SQL al servidor y aumentar el rendimiento.

```

Set rst = CurrentDb.OpenRecordset("SELECT Historial.*, dbo_Rechazos.descripcion FROM Historial LEFT JOIN
    dbo_Rechazos ON Historial.CodRechazo = dbo_Rechazos.[Numero Rechazo]" _
    & "WHERE Historial.[FechaPenal] BETWEEN #" & Format(Fechaini, "mm/dd/yyyy") & "# AND #" &
    Format(Fechafin, "mm/dd/yyyy") & "# ORDER BY [RW] DESC, [FechaPenal] DESC, [NumPenal] DESC",
    dbOpenSnapshot)

rework = rst!RW

appExcel.Worksheets("" & rst!RW & "").Activate

```

Para crear gráficas más complejas sobre el número de penalizaciones de cada empresa se utilizan dos arrays ArrMes y Arrrw:

```

Dim ArrMes(1 To 30, 1 To 12) As Integer

Dim Arrrw(1 To 30) As String

r = 1

Arrrw(r) = rework

```

En Arrrw se guardan el nombre de cada empresa y en ArrMes la cantidad de penalizaciones por mes de las empresas. De forma que los dos arrays se relacionan para un mismo argumento x tal que Arrrw(x) es el nombre y ArrMes(x, 1 to 12) es la cantidad por mes de la misma empresa.

```

Do While Not rst.EOF

    If rework <> rst!RW Then

        appExcel.Worksheets("" & rst!RW & "").Activate

        i = 2

        r = r + 1

        Arrrw(r) = rst!RW

    End If

```

Para cada registro primero se comprueba si la empresa es la misma que la anterior, si no fuera así se cambia de hoja a la correspondiente y se añade un nuevo nombre al array Arrrw. Después de dar formato a cada hoja se copia la información de cada penalización a las celdas:

```
.Range("A" & i).Value = Nz(rst!NumPenal, 0)

.Range("B" & i).Value = Nz(rst!CodRechazo, "")

.Range("C" & i).Value = Nz(rst!NumPart, 0)

.Range("D" & i).Value = Nz(rst!PuntoDetec, "")

.Range("E" & i).Value = Nz(rst!Severidad, "")

.Range("F" & i).Value = Nz(rst!Repetitividad, 0)

.Range("G" & i).Value = Nz(rst!PuntosRestados, 0)

.Range("H" & i).Value = rst!PuntosAct

.Range("I" & i).Value = CDate(rst![FechaPenal])

ArrMes(r, Month(rst!FechaPenal)) = ArrMes(r, Month(rst!FechaPenal)) + 1

.Range("J" & i).Value = rst!descripcion

.Range("A" & i & ":I" & i).HorizontalAlignment = -4108

.Range("J" & i).WrapText = True

End With

i = i + 1

rework = rst!RW

rst.MoveNext

Loop
```

Se cuenta el número de penalizaciones en cada mes usando la función Month, que devuelve el número de mes (del 1 al 12) de un argumento Fecha. Al finalizar todos los registros del Recordset se obtiene completos los dos arrays Arrrw y ArrMes y las hojas para cada empresa.

En la hoja principal ("Overview") se extrae el número de penalizaciones de cada severidad para cada empresa mediante la función de Excel CONTAR.SI.CONJUNTO, que devuelve el número de veces que se cumple una condición múltiple, con argumentos el nombre de empresa y el tipo de severidad:

```
.Range("M2").Select

.ActiveCell.FormulaR1C1 = "=COUNTIFS(C[-11],RC[-1],C[-8],""C"")"

.Selection.AutoFill Destination:=.Range("M2:M" & (NumRW + 1)), Type:=xFillDefault

.Range("N2").Select

.ActiveCell.FormulaR1C1 = "=COUNTIFS(C[-12],RC[-2],C[-9],""B"")"

.Selection.AutoFill Destination:=.Range("N2:N" & (NumRW + 1)), Type:=xFillDefault

.Range("O2").Select

.ActiveCell.FormulaR1C1 = "=COUNTIFS(C[-13],RC[-3],C[-10],""A"")"

.Selection.AutoFill Destination:=.Range("O2:O" & (NumRW + 1)), Type:=xFillDefault

.Range("P2").Select

.ActiveCell.FormulaR1C1 = "=COUNTIFS(C[-14],RC[-4],C[-11],""BLITZ"")"

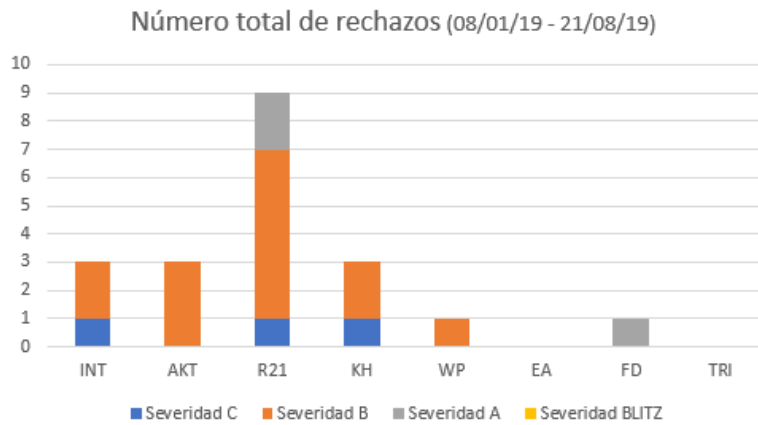
.Selection.AutoFill Destination:=.Range("P2:P" & (NumRW + 1)), Type:=xFillDefault

.Range("Q2").Select

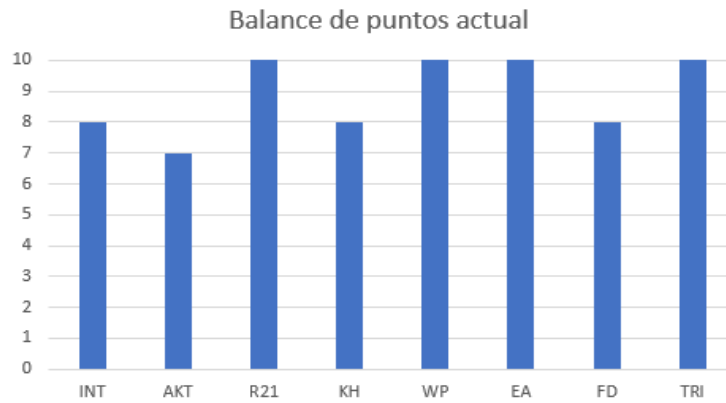
.ActiveCell.FormulaR1C1 = "=SUM(RC[-4]+RC[-3]+RC[-2]+RC[-1])"

.Selection.AutoFill Destination:=.Range("Q2:Q" & (NumRW + 1)), Type:=xFillDefault
```

Con esta información organizada, se utiliza una gráfica de barras apiladas para visualizar la cantidad de penalizaciones de cada empresa separadas por severidad:



Otra gráfica que muestra el balance de puntos actual:



Se inserta también la información de los arrays Arrrw y ArrMes en la hoja principal mediante el código:

```

i = 2

For m = 1 To 12

  For r = 1 To NumRW

    If (ArrMes(r, m) <> 0) Then

      If (mesconrech <> MonthName(m, False)) Then

        mesconrech = MonthName(m, False)

        If i <> 2 Then

          .Range("T" & i).Value = " "

          i = i + 1

        End If

        .Range("T" & i).Value = mesconrech

      End If

      .Range("U" & i).Value = Arrrw(r)

      .Range("V" & i).Value = ArrMes(r, m)

      i = i + 1

    End If

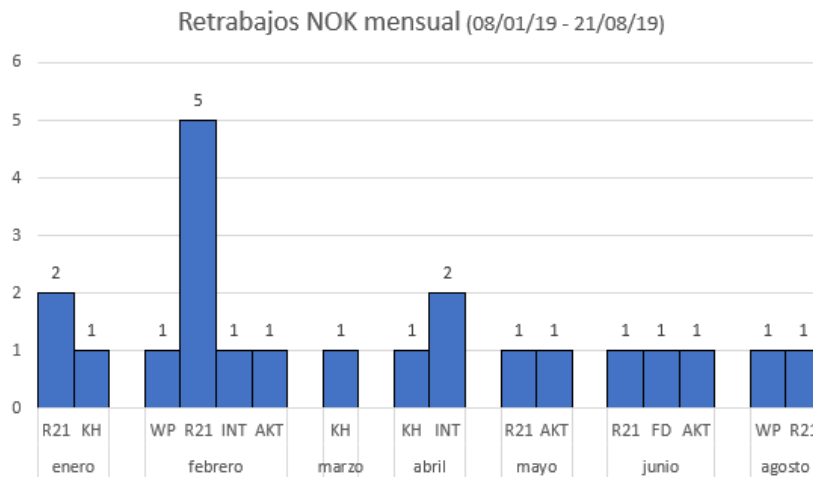
  Next

Next

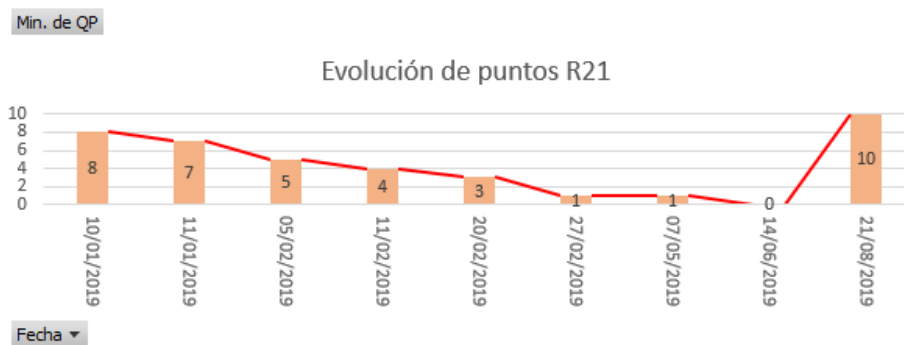
Next

```

Resultando en un gráfico de barras por meses:



Por último, una gráfica que muestra la evolución de los puntos de cada empresa en el tiempo:



5.2 Actualizaciones

Como la aplicación será utilizada por diferentes usuarios en diferentes equipos, se dividirá la base de datos utilizando el divisor de bases de datos de Microsoft Access. Por tanto cada usuario tendrá su propia copia del archivo “front-end” que contiene el código y los formularios como interfaz mientras que un archivo “back-end” situado en un disco de red contiene los datos de las tablas. Mediante este aislamiento se consigue mayor seguridad de los datos y el archivo de cada usuario es independiente de los demás.

Para mayor seguridad, también se limita el acceso del fichero de los usuarios convirtiéndolo en formato .ACCDE, que compila el código evitando poder editarlo y esconde los paneles de navegación obligando a usar los botones de los formularios de la aplicación.

Se implementa un sistema de actualizaciones para asegurar que la versión del programa de cada usuario es la misma utilizando una tabla local en cada front-end y una tabla vinculada al back-end. De manera que al encender la aplicación se realiza una comprobación de la versión indicada en el back-end y la indicada en el front-end:

```

Dim versionrst As DAO.Recordset

PathtoBat = "C:\PenalRW\Actualizar_PenalRW.bat"

On Error GoTo eh

Set versionrst = CurrentDb.OpenRecordset("SELECT VersionLocal.Version, VersionW.Version FROM
VersionLocal, VersionW WHERE VersionLocal.Version = VersionW.Version", dbOpenSnapshot)

If versionrst.EOF Then

    If MsgBox("Existe una actualización de PenalRW. Desea actualizar?", vbYesNo + vbQuestion) = vbYes Then

        Shell "cmd /k " & PathtoBat

        DoCmd.Quit

    End If

End If

```

Si la versión en la tabla en red es diferente de la local se ejecuta el archivo "Actualizar_PenalRW.bat", un archivo por lotes que abre la consola de comandos de Windows, y se cierra la aplicación. El archivo también se puede utilizar por un usuario para instalar el programa en su equipo. Se ejecuta:

```

:actualizacion

REM /* Eliminar elementos antiguos */

IF EXIST C:\PenalRW\PenalRW*.accdb DEL /F /Q C:\PenalRW\PenalRW*.accdb

IF EXIST C:\PenalRW\PenalRW*.accde DEL /F /Q C:\PenalRW\PenalRW*.accde

IF EXIST C:\PenalRW\PenalRW*.lnk DEL /F /Q C:\PenalRW\PenalRW*.lnk

REM /* Crear carpetas de la aplicación */

IF NOT EXIST C:\PenalRW MD C:\PenalRW

```

Se eliminan los elementos antiguos en la carpeta de instalación del programa y se crean los directorios si no existían.

```

REM /* Copiar elementos nuevos */

IF EXIST C:\PenalRW\PenalRW.zip DEL /F /Q C:\PenalRW\PenalRW.zip

COPY W:\****\PenalRW\PenalRW.zip C:\PenalRW

7z x -aoa -bso0 -y C:\PenalRW\PenalRW.zip -oC:\PenalRW

IF ERRORLEVEL 2 PAUSE

DEL /F /Q C:\PenalRW\PenalRW.zip

```

Se copia el archivo zip con la nueva actualización desde el disco de red a la carpeta local y se descomprime.

```

REM /* Actualizar acceso directo en Escritorio */

DEL /F /Q %USERPROFILE%\Desktop\PenalRW*.lnk

COPY "C:\PenalRW\PenalRW - Acceso directo.lnk" "%USERPROFILE%\Desktop\"

REM /* Log de esta instalación */

START cmd /c W:\****\PenalRW\logme.bat %0

```

Para llevar un seguimiento de los usuarios que actualizan se utiliza otro .bat, llamado “logme.bat”, que simplemente escribe el usuario actual y la fecha en el archivo en red.

```

ECHO *****

ECHO Actualizacion completa. Iniciando 'PenalRW'.

ECHO Puede cerrar esta ventana.

ECHO *****

START /MAX MSACCESS.EXE "C:\PenalRW\PenalRW.accd"

```

Una vez finalizado se abre el nuevo fichero front-end actualizado.

5.3 Futuro desarrollo

Para facilitar el mantenimiento del programa y su desarrollo posterior, se ha programado un nuevo ejecutable “GenerarActPenalRW.bat” que genera una nueva versión y prepara todos los archivos a partir del archivo .ACCDB original:

```

IF EXIST W:\****\PenalRW\test\PenalRW.zip (

DEL /F /Q W:\****\PenalRW\test\PenalRWlast.zip

RENAME W:\****\PenalRW\test\PenalRW.zip PenalRWlast.zip

```

Se actualiza la copia de seguridad situada en el disco de red. Si la nueva versión tuviese algún error se renombra el archivo PenalRWlast.zip a PenalRW.zip para revertir el cambio.

Para lograr acceder a las tablas del back-end se llama a un script de Visual Basic situado en la misma carpeta con nombre “ActVersiones.vbs” con el parámetro /WAIT, que espera a que haya finalizado antes de seguir ejecutando el resto del .bat:

```

START /WAIT WScript "ActVersiones.vbs"

Set app = CreateObject("Access.Application")

```

En el script se crea un objeto de Microsoft Access del archivo front-end actualizado y se ejecuta un macro “SubirVersion” que aumenta la versión de la tabla local:

```
app.OpenCurrentDatabase "W:\***\PenalRW\test\PenalRW.accdb"
```

'Ejecuta el macro que sube la version, pidiendo actualizar a todas las FE

```
app.DoCmd.RunMacro "SubirVersion"
```

```
Wscript.sleep 2000 'Por si no se actualiza al instante
```

Se guarda el número de versión en una variable llamada versionL y se convierte el archivo a .ACCDE mediante el comando SYSCMD 603:

```
versionL = app.Run("getversion")
```

```
app.CloseCurrentDatabase
```

```
app.SysCmd 603, "W:\***\PenalRW\test\PenalRW.accdb", "W:\***\PenalRW\test\PenalRW.accde"
```

Finalmente se abre el archivo back-end y se establece la nueva versión en la tabla:

```
app.OpenCurrentDatabase "W:\***\PenalRW\Database\PenalRW_be.accdb"
```

```
app.Run "setversion", versionL
```

```
app.CloseCurrentDatabase
```

```
app.Quit
```

Desde el archivo por lotes .bat se crea una copia de seguridad del back-end y el archivo comprimido .zip con el programa.

```
IF EXIST W:\***\PenalRW\Database\PenalRW_be.accdb COPY
    W:\***\PenalRW\Database\PenalRW_be.accdb W:\***\PenalRW\Database\PenalRW_be_backup.accdb

7z a -aoa W:\***\PenalRW\test\PenalRW.zip W:\***\PenalRW\test\PenalRW.accde

7z a -aoa W:\***\PenalRW\test\PenalRW.zip W:\***\PenalRW\test\PenalRW*.lnk

7z a -aoa W:\***\PenalRW\test\PenalRW.zip W:\***\PenalRW\test\import

IF EXIST W:\***\PenalRW\PenalRW.zip DEL /F /Q W:\***\PenalRW\PenalRW.zip

COPY W:\***\PenalRW\test\PenalRW.zip W:\***\PenalRW\

PAUSE
```

Una vez creado se reemplaza el existente .zip por el actualizado y se finaliza.

Las instrucciones que seguir han sido detalladas en documentos de texto en la misma carpeta, de forma que un usuario técnico pueda entender y realizar la actualización sin problema.

6. Conclusión

Mediante el diseño del nuevo sistema por puntos se conseguirá:

- **Una mejora indirecta en la calidad** de los retrabajos realizados por estas empresas terceras que intentan prevenir la pérdida de puntos.
- **Mayor responsabilidad** en las empresas retrabajos. Se considera la dificultad del retrabajo antes de aceptarlo y se forma al personal más estrictamente para evitar pérdidas de puntos.
- **Menor coste a Ford**. La gestión de los fallos cometidos y el coste de la pieza mal retrabajada son tiempo y costes que paga Ford y deben minimizarse.
- **Medibles de rendimiento y control** de las empresas terceras. Mediante la acumulación y exportación de datos se consigue visualizar de forma simple el rendimiento de cada empresa.
- **Mayor oportunidad** para las empresas menos contratadas. Para recuperar los puntos debe pasar 6 meses sin ninguna penalización, de forma que las empresas que realicen más retrabajos tendrán una mayor probabilidad de quedarse sin puntos y no poder retrabajar durante un período de tiempo, de modo que las demás empresas podrán aceptar más retrabajos.

Se ha programado una base de datos en Access para la implementación del sistema: recopilando datos sobre penalizaciones, visualizando rendimiento en forma de gráfica en Excel con exportación de datos y regulando el balance de puntos de cada empresa. La aplicación está en uso actualmente y recibirá futuro soporte por parte de los técnicos informáticos del equipo IT de la factoría.

Para facilitar el mantenimiento y futuro desarrollo se han programado scripts adicionales en archivos BATCH(.bat) y VBS(Visual Basic Script) y se han redactado instrucciones de uso. El objetivo es la creación de un entorno de programación accesible por cualquier usuario de nivel técnico para poder añadir nuevas funcionalidades al programa según sea necesario en un futuro.

La aplicación se ha programado mediante el feedback de los ingenieros del departamento, logrando funciones personalizadas según la necesidad solicitada. Durante el proceso se han detectado errores que han sido resueltos por medio del apoyo del equipo de IT de la factoría.

7. Bibliografía

1. *Procedimientos de muestreo para la inspección por atributos. Parte 1: Planes de muestreo para las inspecciones lote por lote, tabulados según el límite de calidad de aceptación (LCA). UNE-ISO 2859-1:2012.*
2. *Procedimientos de muestreo para la inspección por variables. Parte 1: Especificaciones para los planes de muestreo simples tabulados según el nivel de calidad aceptable (NCA) para la inspección lote por lote para una característica de calidad única y un nivel de calidad aceptable (NCA) único. UNE-ISO 3951-1:2012.*
3. Le Guen, Frédéric: *Macros y Lenguaje VBA. Aprender A Programar Con Excel.* Barcelona, 2013.
4. Marian, Thierry: *VBA Access 2016. Aprenda a crear aplicaciones profesionales.* Barcelona, 2016.
5. Amelot, Michele: *VBA Access 2013. Programar En Access.* Paris, 2013.
6. Ayala, Luis y Yessy Arriaga: *Visual Basic 2017: Nivel Intermedio.* México, 2017.