

DICIEMBRE  
2016

CHAVEZ MONTOYA  
ERIK JABNEEL



# ACTUALIZACIÓN Y MANTENIMIENTO DEL SISTEMA "RGP" (ROUTING GENERATION PROGRAM).



# **INSTITUTO TECNOLÓGICO DE PABELLÓN DE ARTEAGA.**

**Ingeniería en Tecnologías de la Información y las  
Comunicaciones.**

**'ACTUALIZACIÓN Y MANTENIMIENTO DEL SISTEMA "RGP"  
(ROUTING GENERATION PROGRAM).'**

**MAHLE Componentes de Motor de México, S. de R.L. DE C.V.**

**Erik Jabneel Chavez Montoya.**

**Asesor Interno: MC. Renato Eduardo Reyes González.**

**Asesor Externo: Ing. Jorge Humberto Rodríguez Barrientos.**

**Pabellón de Arteaga, Ags. Diciembre 2016.**

## **Dedicatoria.**

A mi madre: C. Rosa Montoya Esparza por todo el apoyo incondicional que día con día me brindo, por darme los mejores consejos llenos de sabiduría y esperanza cuando los necesite, por escucharme y alentarme a tomar siempre la mejor decisión y a buscar siempre lo mejor para mí, por dejarme en claro que no hay mayor satisfacción que la de realizar todas tus metas.

A mis maestros: MC. Francisco Sánchez Mares, Lic. José de Jesús Pérez Limón, MC. Renato Eduardo Reyes González por brindarme siempre su apoyo cuando se los solicite, por sus consejos y enseñanzas, por mostrarme que siempre hay que dar lo mejor de uno tanto académicamente como profesionalmente.

## **Agradecimientos.**

Al Lic. Jorge López Flores jefe de Recursos Humanos, por haberme brindado la oportunidad de desarrollar mi Residencia profesional en Mahle, Componentes para Motor de México.

Al Ing. Jorge Humberto Rodríguez Barrientos por brindarme la confianza para trabajar de la mano, permitiéndome aprender de su experiencia y conocimientos a lo largo de todo este tiempo que desempeñe mis labores en el departamento de Ingeniería de rutas.

A la toda comunidad Docente de la máxima casa de estudios en Pabellón de Arteaga, el "Instituto Tecnológico de Pabellón de Arteaga".

## Resumen

Se realizó reingeniería y actualizaciones dentro del sistema "ROUTING GENERATION PROGRAM" en el cual calculan los tiempos estándar y la salida al "SAP" de los expansores, anillos, large bore, aceros y transmisión.

Dentro del departamento de Ingeniería de rutas manejan un sistema en el cual generan hojas de ruta de anillos, segmentos, expansores, large bore y franklin. Las actualizaciones y reingeniería que se le realizaron al sistema tienen como propósito hacer más ágil y eficiente el sistema "RGP" actualizando las fórmulas para calcular las diferentes operaciones por las cuales pasa el expansor, anillo, large bore, franklin o segmento y así generar el nuevo componente de una manera más rápida y certera.

## INDICE.

DEDICATORIA.....	3
AGRADECIMIENTOS.....	3
RESUMEN.....	4
INDICE.....	5
LISTA DE TABLAS.....	8
LISTA DE FIGURAS.....	9
CAPITULO I.....	14
1. INTRODUCCIÓN.....	14
1.1 Antecedentes.....	15
1.1.1 Historia MAHLE componentes para motor de México.....	16
1.1.2 Política de calidad.....	19
1.1.2.1 Objetivos de Calidad.....	19
1.1.3 Principios y valores de la compañía.....	19
1.1.3.1 Principios.....	19
1.1.3.2 Valores.....	19
1.1.4 Nuestros Clientes.....	19
1.1.5 Organigrama Funcional de Mahle Componentes para Motor de México Planta Aguascalientes.....	20
1.1.5.1 Organigrama Ingeniería.....	20
1.2 Planteamiento del Problema.....	20
1.3 Objetivos.....	21
1.3.1 General.....	21
1.3.2 Específicos.....	21
1.4 Justificación.....	21
CAPITULO II.....	22
2. MARCO TEÓRICO.....	22
2.1.1 Mantenimiento del software.....	22
2.1.1.1 Definiciones.....	22
2.1.1.2 Naturaleza del trabajo de mantenimiento.....	23
2.1.1.3 Factores que afectan el esfuerzo de mantenimiento.....	23
2.1.2 Reingeniería.....	23
2.1.2.1 Propósitos de la Reingeniería.....	24
2.1.2.2 Tipos de Reingeniería.....	24
2.1.2.2.1 Reestructuración.....	24
2.1.2.2.2 Ingeniería Inversa.....	25
2.1.2.2.3 Administración de configuración y cambios.....	25
2.1.2.3 Reingeniería contra reemplazo.....	25
2.1.3 Tiempo Estándar.....	26
2.1.3.1 Aplicación del Tiempo Estándar.....	27
2.1.3.2 Tiempo Básico.....	27
2.1.3.3 Tiempo Suplementario.....	28
2.1.3.4 Tiempo Improductivo.....	28
2.1.3.5 Tolerancias.....	28
2.1.3.6 Mantenimiento de tiempos estándar.....	29
2.1.3.7 Sistemas de tiempos predeterminados.....	29
2.1.3.8 Cálculo del Tiempo Estándar.....	29
2.1.3.9 De los Tiempos Observados al Tiempo Estándar.....	30

2.1.4	ERP (Enterprise Resource Planning).....	30
2.1.4.1	Características.....	30
2.1.4.2	Integralidad.....	31
2.1.4.3	Modularidad.....	32
2.1.4.4	Adaptabilidad.....	32
2.1.4.5	Ventajas de un ERP.....	32
2.1.5	SAP (Systems, applications and products).....	33
2.1.5.1	¿Que son los módulos SAP?.....	35
2.1.5.2	Costos.....	36
2.1.5.3	Costos de la implementación de los procesos.....	36
CAPITULO III.....		37
3.	METODOLOGÍA.....	37
3.1.1	Agregar Nuevos Registros en la Base de Datos.....	37
3.1.2	Acceder a Modificar, Actualizar o Agregar Nuevos Centros de Trabajo.....	41
3.1.3	Pattern2.vb – Placas Modelo.....	42
3.1.4	ExportarPlacasModelo.vb.....	43
3.1.5	Anillos.....	45
3.1.5.1	CentroTrabajo120.vb – HEAT TREAT PREM. GASOLINE.....	45
3.1.5.2	CentroTrabajo130.vb – SPLITTER CASTINGS.....	48
3.1.5.3	CentroTrabajo170.vb – FINISH GRIND (MICRO).....	51
3.1.5.4	CentroTrabajo236.vb – MACHINE CENTER KOMATSU.....	52
3.1.5.5	CentroTrabajo255.vb – GAP SIZER WET (SIM HUMEDA).....	54
3.1.5.6	CentroTrabajo261.vb – LASSER ENGRAVE.....	55
3.1.5.7	CentroTrabajo265.vb – ENGRAVE HIGH RUNNERS.....	55
3.1.5.8	CentroTrabajo280.vb – BATES BORE.....	59
3.1.5.9	CentroTrabajo290.vb – PRUEBA DE MANDRIL.....	60
3.1.5.10	CentroTrabajo410.vb – FINISH MILL.....	61
3.1.5.11	CentroTrabajo415.vb – FINISH MILL (HIGH RUNNERS).....	61
3.1.5.12	CentroTrabajo420.vb – AUTO. FINISH TURN.....	65
3.1.5.13	CentroTrabajo424.vb – AUTO FINISH TURN GANG KATAOKA.....	65
3.1.5.14	CentroTrabajo425.vb – AUTO. FINISH TURN (HIGH RUNNERS).....	66
3.1.5.15	CentroTrabajo427.vb – LINTECH A.F.T.....	69
3.1.5.16	CentroTrabajo428.vb – AXIAL FORM.....	70
3.1.5.17	CentroTrabajo450.vb – I.D. BRUSH.....	70
3.1.5.18	CentroTrabajo465.vb – SCOTCHBRITE RINGS.....	70
3.1.5.19	CentroTrabajo490.vb – GRINDING O.D. KATAOKA.....	71
3.1.5.20	CentroTrabajo495.vb – ROLADORA DE ANILLOS.....	71
3.1.5.21	CentroTrabajo500.vb – POINT BEND.....	71
3.1.5.22	CentroTrabajo501.vb – ID POINT CHAMFER.....	72
3.1.5.23	CentroTrabajo767.vb – WHEELABRATOR GASOLINE.....	72
3.1.5.24	CentroTrabajo810.vb – VISUAL INSPECTION PREM. GASOLINE.....	73
3.1.5.25	CentroTrabajo815.vb – FINAL INSPECTION RINGS PREM. GASOLINE.....	73
3.1.5.26	CentroTrabajo820.vb – 100% GAP INSPECTION (AUTOGAP) GASOLINE.....	73
3.1.5.27	CentroTrabajo830.vb – FINAL INSPECTION RINGS.....	74
3.1.6	Large Bore.....	76
3.1.6.1	CentroTrabajo9201.vb – Coil Steel L.B.....	76
3.1.6.2	CentroTrabajo9202.vb – Coil Stainless Steel.....	77
3.1.6.3	CentroTrabajo9203.vb – Corte de Bobinas (CAMPBELL CUT OFF).....	78
3.1.6.4	CentroTrabajo9204.vb – STRESS REL&HEAT SHAP#1 Y STRESS REL&HEAT SHAP#2.....	78
3.1.6.5	CentroTrabajo9208.vb – Stack Size LB.....	80

3.1.6.6	CentroTrabajo9210.vb – Visual Inspection L.B.....	81
3.1.6.7	CentroTrabajo9215.vb – Rough Grind. ....	82
3.1.6.8	CentroTrabajo9221.vb – Side Grind Besly.....	85
3.1.6.9	CentroTrabajo9222.vb – Band Rings & Unband LB. ....	86
3.1.6.10	CentroTrabajo9223.vb – 6 NOTCH.....	87
3.1.6.11	CentroTrabajo9224.vb – Counterbore-Southben LB.....	88
3.1.6.12	CentroTrabajo9225.vb – Groove Slot LB. ....	90
3.1.6.13	CentroTrabajo9241.vb – Moly Coating L.B. ....	91
3.1.6.14	CentroTrabajo9242.vb – O.D. Sheffield L.B.....	92
3.1.6.15	CentroTrabajo9265.vb – Splitter L.B. ....	94
3.1.6.16	CentroTrabajo9267.vb – Shape & Notch L.B.....	95
3.1.6.17	CentroTrabajo9282.vb – Frm Turn/Bore >6.5 L.B. Iron.....	96
3.1.6.18	CentroTrabajo9283.vb – FORM TRN/BORE DUCTIL. ....	97
<b>3.1.7</b>	<b>Expansores.....</b>	<b>98</b>
3.1.7.1	CentroTrabajo499.vb – „GAS NITRIDING EXPANDERS“.....	98
3.1.7.2	CentroTrabajo560.vb – „EXPANDER SS-50 PUNCHING “.....	98
3.1.7.3	CentroTrabajo563.vb – „LANCE KEY“.....	99
3.1.7.4	CentroTrabajo565.vb – „EXPANDER ES-80 FORMING “.....	99
3.1.7.5	CentroTrabajo566.vb – „WIRE LATCH“.....	99
3.1.7.6	CentroTrabajo581.vb – „EXPANDER 98-22 FORMING“.....	101
3.1.7.7	CentroTrabajo721.vb – „BLACKENING EXPANDERS“.....	101
3.1.7.8	CentroTrabajo740.vb – „ELECTROPOLISH EXPANDERS“.....	102
3.1.7.9	CentroTrabajo745.vb – „CHEMICAL POLISH EXPANDERS “.....	102
3.1.7.10	CentroTrabajo832.vb – „FINAL INSP. EXPANDER“.....	103
<b>3.1.8</b>	<b>Franklin.....</b>	<b>104</b>
3.1.8.1	CentroTrabajo2005.vb – „FIRST ROUGH GRIND NORTON“.....	104
3.1.8.2	CentroTrabajo2020.vb – „ROUGH BORE 1 CUTS SPLIT BUSH“.....	105
3.1.8.3	CentroTrabajo2030.vb – „SECOND ROUGH GRIND“.....	105
3.1.8.4	CentroTrabajo2035.vb–„SECOND ROUGH GRIND NORTON“.....	108
3.1.8.5	CentroTrabajo2070.vb – „DEGREASE“.....	109
3.1.8.6	CentroTrabajo2080.vb – „FIN. GRD.“.....	110
3.1.8.7	CentroTrabajo2110.vb – „AUTOLOAD CTB HIGH VOL. CELL“.....	110
3.1.8.8	CentroTrabajo2120.vb – „MILL HOOK LAP-AUT; MAC“.....	110
3.1.8.9	CentroTrabajo2130.vb – „MILL CHAMF AT GAP - AUTOM.“.....	110
3.1.8.10	CentroTrabajo2140.vb – „HOOKLAP / HOOKER HIGH VOL. C“.....	111
3.1.8.11	CentroTrabajo2150.vb – „HOOK LAP ASSEMBLY“.....	112
3.1.8.12	CentroTrabajo2160.vb – „HOOK LAP DISASSEMBLY“.....	112
3.1.8.13	CentroTrabajo2170.vb – „MILL ANG. SEAL-1ST SIDE O.D“.....	113
3.1.8.14	CentroTrabajo2180.vb – „ENGRAVE“.....	114
3.1.8.15	CentroTrabajo2190.vb – „BATES BORE“.....	115
3.1.8.16	CentroTrabajo2210.vb – „FIN. BORE - HOOK LAP - SPLIT“.....	116
3.1.8.17	CentroTrabajo2260.vb – „VIBRATRON“.....	116
3.1.8.18	CentroTrabajo2270.vb – „TIN PLATE“.....	117
3.1.8.19	CentroTrabajo2330.vb – „MILL LAP JOINT - AUT.; MA“.....	117
3.1.8.20	CentroTrabajo2350.vb – „100% GAP INSPECTION (AUTOGAP)“.....	118
	RESULTADOS.....	119
	CONCLUSIONES.....	120
	COMPETENCIAS DESARROLLADAS.....	121
	PROGRAMA DE ACTIVIDADES CRONOGRAMA DE ACTIVIDADES.....	122
	REFERENCIAS.....	123

# Lista de Tablas

TABLA 1 TIEMPOS CICLO POR TIPO DE MATERIAL.- CT- 130.....	48
TABLA 2 VALORES DE ESPACIADORES Y ANILLOS POR CÉLULA. ....	49
TABLA 3 ESQUEMA DE VALORES NECESARIOS ESPACIADORES.- CT- 130 .....	50
TABLA 4 TIEMPOS CICLO OPTIMIZADOS.- CT- 410.....	61
TABLA 5 VALORES DE BLAST Y BLOWOFF .....	91



# Lista de Figuras

ILUSTRACIÓN 1 FACTORES DEL TIEMPO ESTÁNDAR.....	27
ILUSTRACIÓN 2 ICONO DE SQL EN EL MENÚ DE INICIO.....	37
ILUSTRACIÓN 3 PANTALLA INICIAL DEL PROGRAMA.....	37
ILUSTRACIÓN 4 SERVIDORES DISPONIBLES.....	38
ILUSTRACIÓN 5 MENSAJE DE ALERTA DE CONEXIÓN.....	38
ILUSTRACIÓN 6 MENSAJE DE ADVERTENCIA.....	38
ILUSTRACIÓN 7 SERVIDOR QUE ALOJA LA BD.....	38
ILUSTRACIÓN 8 CARPETAS INCLUIDAS EN EL SERVIDOR.....	38
ILUSTRACIÓN 9 BASES DE DATOS DENTRO DEL SERVIDOR.....	39
ILUSTRACIÓN 10 SELECCIONAR UNA TABLA DE LA BASE DE DATOS.....	39
ILUSTRACIÓN 11 ABRIR TABLAS DE LA BD SELECCIONADA.....	39
ILUSTRACIÓN 12 ABRIR TABLA DE SQL PARA VER DATOS.....	40
ILUSTRACIÓN 13 TABLA CENTROTRABAJOCLASIFICACIONHERRAMIENTALES.....	40
ILUSTRACIÓN 14 MENÚ DESPLEGABLE CON LAS OPCIONES DE LA TABLA.....	40
ILUSTRACIÓN 15 POSICIONAMIENTO DEL CURSOR PARA AGREGAR NUEVOS REGISTROS.....	41
ILUSTRACIÓN 16 ICONO DE SHARPDEVELOP4.4 EN EL MENÚ DE INICIO.....	41
ILUSTRACIÓN 17 VENTANA DE BUSQUEDA.....	41
ILUSTRACIÓN 18 PÁGINA DE INICIO SHARPDEVELOP.....	41
ILUSTRACIÓN 19 PROYECTO ABIERTO Y LISTO PARA SU EDICIÓN U ACTUALIZACIÓN.....	42
ILUSTRACIÓN 20 INTERFACE GRÁFICA (FORM).- PLACAS MODELO.....	43
ILUSTRACIÓN 21 FUNCIÓN DEL BOTÓN EXPORTAR EXCEL.....	43
ILUSTRACIÓN 22 MÉTODO GETFILE( ), VARIABLES NECESARIAS Y QUERY REALIZADO A LA BD.- EXPORTARPLACASMODELO.....	44
ILUSTRACIÓN 23 ENCABEZADOS DE LA NUEVA HOJA DE EXCEL.- EXPORTARPLACASMODELO.....	44
ILUSTRACIÓN 24 LECTORES PARA INSERTAR LOS DATOS EN LA HOJA EXCEL.-EXPORTARPLACASMODELO.....	44
ILUSTRACIÓN 25 FORMATO DE LA NUEVA HOJA EXCEL.- EXPORTARPLACASMODELO.....	45
ILUSTRACIÓN 26 FORM ACTUALIZADO.- CT-120.....	45
ILUSTRACIÓN 27 FORM ANTES DE ACTUALIZACIÓN.- CT-120.....	45
ILUSTRACIÓN 28 MÉTODO LLENAR_MATERIAL( ). CT-120.....	46
ILUSTRACIÓN 29 NUEVAS VARIABLES. CT-120.....	46
ILUSTRACIÓN 30 FUNCIÓN TEST( ). CT-120.....	47
ILUSTRACIÓN 31 MÉTODO CALCULAR( ). - CT -120.....	47
ILUSTRACIÓN 32 FROM ANTES DE LA ACTUALIZACIÓN.- CT -130.....	48
ILUSTRACIÓN 33 FORM ACTUALIZADO.- CT -130.....	48
ILUSTRACIÓN 34 ACTUALIZACIÓN DE LOS VALORES DE LA VARIABLE "T_CICLO". - CT-130.....	49
ILUSTRACIÓN 35 CONDICIÓN IF PARA OBTENER EL VALOR DE ANILLOSxCELULA.- CT-130.....	49
ILUSTRACIÓN 36 VARIABLE „TIEMPO_MAQUINA“ FORMULA ACTUALIZADA.-CT-130.....	50
ILUSTRACIÓN 37 FUNCIÓN CLICK( ) BOTÓN ACEPTAR.-CT -130.....	50
ILUSTRACIÓN 38 FORM ANTES DELA ACTUALIZACIÓN.- CT-170.....	51
ILUSTRACIÓN 39 FORM ACTUALIZADO.-CT -170.....	51
ILUSTRACIÓN 40 EVENTO LOAD.- CT-170.....	51
ILUSTRACIÓN 41 MÉTODO CALCULAR( ), Y TIEMPOS CICLO ACTUALIZADOS.- CT-170.....	52
ILUSTRACIÓN 42 FORM ANTES DE LA ACTUALIZACIÓN.- CT-236.....	52

ILUSTRACIÓN 43 FORM ACTUALIZADO. - CT-236. ....	52
ILUSTRACIÓN 44 FUNCIÓN TEST( ) - CT-236. ....	53
ILUSTRACIÓN 45 VRIABLES NUEVAS Y MÉTODO CALCULAR( ) - CT-236. ....	53
ILUSTRACIÓN 46 MÉTODO CALCULAR( ) Y TIEMPOS CICLO ACTUALIZADOS.- CT-255. ....	54
ILUSTRACIÓN 47 CONDICIONANTE IF DENTRO DEL MÉTODO CALCULAR( ) - CT-255. ....	54
ILUSTRACIÓN 48 MÉTODO CALCULAR( ), TIEMPOS ACTUALIZADOS Y NUEVOS VALORES DE "A_CICLICAS" - CT-261. ....	55
ILUSTRACIÓN 49 AGREGAR NUEVO CENTRO DE TRABAJO. - CT-265. ....	56
ILUSTRACIÓN 50 AGREGAR NUEVO CENTRO DE TRABAJO.- CT-265. ....	57
ILUSTRACIÓN 51 IMPORTS LOGICA. -CT-265. ....	57
ILUSTRACIÓN 52 PROPIEDADES DE LA INTERFACE. - CT-265. ....	58
ILUSTRACIÓN 53 FUNCIÓN TEST Y PASARVALORESDELISTAS. - CT-265. ....	58
ILUSTRACIÓN 54 PROPIEDAD CENTROTRABAJO, MÉTODO CALCULAR( ) Y PROPIEDAD FACTORLABOR.- CT-265. ....	59
ILUSTRACIÓN 55 MÉTODO CALCULAR( ) - CT- 280. ....	59
ILUSTRACIÓN 56 MÉTODO "SACA_TIPOHIER" PARA OBTENER TIEMPO CICLO. - CT- 280. ....	60
ILUSTRACIÓN 57 NUEVA VARIABLE "t_ciclo" - CT- 290. ....	60
ILUSTRACIÓN 58 MÉTODO CALCULAR( ) - CT- 290. ....	60
ILUSTRACIÓN 59 MÉTODO CALCULAR( ), TIEMPOS CICLO ACTUALIZADOS.- CT- 410. ....	61
ILUSTRACIÓN 60 AGREGAR NUEVO CENTRO DE TRABAJO.- CT- 415. ....	62
ILUSTRACIÓN 61 AGREGAR CENTRO DE TRABAJO. CT- 415. ....	63
ILUSTRACIÓN 62 IMPORTS LOGICA.- CT- 415. ....	63
ILUSTRACIÓN 63 INTERFACE GRÁFICA.- CT - 415. ....	64
ILUSTRACIÓN 64 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 420. ....	65
ILUSTRACIÓN 65 VARIABLE TIEMPO CICLO ACTUALIZADA. - CT-420. ....	65
ILUSTRACIÓN 66 MÉTODO CALCULAR( ) - CT- 420. ....	65
ILUSTRACIÓN 67 MÉTODO CALCULAR( ) - CT- 424. ....	66
ILUSTRACIÓN 68 EVENTO LOAD.- CT- 424. ....	66
ILUSTRACIÓN 69 CÓDIGO NECESARIO QUE DEBE LLEVAR TODO CENTRO DE TRABAJO.- CT- 425. ....	67
ILUSTRACIÓN 70 NUEVO FORM.- CT- 425. ....	67
ILUSTRACIÓN 71 VARIABLE PARA ALMACENAR EL VALOR DE ENTRADA.- CT- 425. ....	68
ILUSTRACIÓN 72 CONSTRUCTOR DONDE SE AGREGAN LOS VALORES A LAS LISTAS.- CT- 425. ....	68
ILUSTRACIÓN 73 FUNCIÓN PASARVALORESDELISTAS( ) - CT- 425. ....	68
ILUSTRACIÓN 74 CONSTRUCTOR Y PARÁMETROS DE LA CLASE.- CT- 425. ....	68
ILUSTRACIÓN 75 MÉTODO CALCULAR( ) - CT- 425. ....	69
ILUSTRACIÓN 76 EVENTO CLICK DEL BOTÓN ACEPTAR.- CT- 425. ....	69
ILUSTRACIÓN 77 MÉTODO CALCULAR( ) - CT- 428. ....	70
ILUSTRACIÓN 78 MÉTODO CALCULAR( ) - CT- 450. ....	70
ILUSTRACIÓN 79 MÉTODO CALCULAR( ) - CT- 465. ....	70
ILUSTRACIÓN 80 MÉTODO CALCULAR( ) - CT- 490. ....	71
ILUSTRACIÓN 81 MÉTODO CALCULAR( ) - CT- 495. ....	71
ILUSTRACIÓN 82 MÉTODO CALCULAR( ) - CT- 500. ....	72
ILUSTRACIÓN 83 MÉTODO CALCULAR( ) -CT- 501. ....	72
ILUSTRACIÓN 84 MÉTODO CALCULAR( ) - CT- 767. ....	72
ILUSTRACIÓN 85 MÉTODO CALCULAR( ) - CT- 810. ....	73
ILUSTRACIÓN 86 MÉTODO CALCULAR( ) - CT- 815. ....	73
ILUSTRACIÓN 87 MÉTODO CALCULAR( ) - CT- 820. ....	74
ILUSTRACIÓN 88 FUNCIÓN TEST( ) - CT- 830. ....	74

ILUSTRACIÓN 89 PROCEDIMIENTO SUB PASARVALORESDELISTAS.- CT- 830. ....	74
ILUSTRACIÓN 90 PROCEDEMIENTO SUB NEW( ).- CT- 830. ....	75
ILUSTRACIÓN 91 MÉTODO CALCULAR( ).- CT- 830. ....	75
ILUSTRACIÓN 92 MÉTODO SUB LLENAR_MATERIAL( ) Y EVENTO LOAD( ).- CT- 830. ....	76
ILUSTRACIÓN 93 MÉTODO CALCULAR( ).- CT-9201. ....	77
ILUSTRACIÓN 94 MÉTODO CALCULAR( ).- CT- 9202. ....	77
ILUSTRACIÓN 95 MÉTODO CALCULAR( ).- CT- 9203. ....	78
ILUSTRACIÓN 96 VARIABLES NECESARIAS.- CT- 9204. ....	78
ILUSTRACIÓN 97 NUEVOS TIEMPOS CICLO EN RECETAS.- CT- 9204. ....	79
ILUSTRACIÓN 98 NUEVOS TIEMPOS CICLO EN RECETAS.- CT- 9204. ....	79
ILUSTRACIÓN 99 OPERACIONES NECESARIAS Y NUEVA FÓRMULA DE TIEMPO ESTÁNDAR.- CT-9204. ....	79
ILUSTRACIÓN 100 OPERACIONES NECESARIAS Y NUEVA FÓRMULA DE TIEMPO ESTÁNDAR.- CT- 9204. ....	80
ILUSTRACIÓN 101 NUEVA INTERFACE GRÁFICA.- CT- 9208. ....	80
ILUSTRACIÓN 102 MÉTODO SUB PASARVALORESDELISTAS.- CT- 9208. ....	80
ILUSTRACIÓN 103 MÉTODO SUB NEW( ).- CT- 9208. ....	81
ILUSTRACIÓN 104 NUEVAS VARIABLES.- CT- 9208. ....	81
ILUSTRACIÓN 105 MÉTODO CALCULAR( ).- CT- 9208. ....	81
ILUSTRACIÓN 106 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 9208. ....	81
ILUSTRACIÓN 107 MÉTODO CALCULAR( ).- CT- 9210. ....	82
ILUSTRACIÓN 108 NUEVA INTERFACE GRÁFICA.- CT- 9215. ....	82
ILUSTRACIÓN 109 FUNCIÓN TEST( ).- CT- 9215. ....	83
ILUSTRACIÓN 110 MÉTODO PASARVALORESDELISTAS( ).- CT- 9215. ....	83
ILUSTRACIÓN 111 MÉTODO LLENAR_MATERIAL( ).- CT- 9215. ....	83
ILUSTRACIÓN 112 MÉTODO SUB NEW( ).- CT- 9215. ....	84
ILUSTRACIÓN 113 MÉTODO CALCULAR( ), NUEVAS VARIABLES Y CONDIONANTE IF.- CT- 9215. ....	84
ILUSTRACIÓN 114 NUEVAS OPERACIONES MÉTODO CALCULAR( ).- CT- 9215. ....	84
ILUSTRACIÓN 115 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 9215. ....	85
ILUSTRACIÓN 116 EVENTO LOAD( ).- CT- 9215. ....	85
ILUSTRACIÓN 117 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 9221. ....	85
ILUSTRACIÓN 118 FORM ACTUALIZADO.- CT- 9221. ....	85
ILUSTRACIÓN 119 MÉTODO CALCULAR( ).- CT- 9221. ....	86
ILUSTRACIÓN 120 MÉTODO CALCULAR( ).- CT-9222. ....	86
ILUSTRACIÓN 121 FUNCIÓN TEST( ).- CT- 9223. ....	87
ILUSTRACIÓN 122 MÉTODO CALCULAR( ).- CT- 9223. ....	87
ILUSTRACIÓN 123 FORM ACTUALIZADO.- CT- 9224. ....	88
ILUSTRACIÓN 124 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 9224. ....	88
ILUSTRACIÓN 125 FUNCIÓN TEST( ).- CT-9224. ....	88
ILUSTRACIÓN 126 VARIABLES NECESARIAS.- CT- 9224. ....	89
ILUSTRACIÓN 127 MÉTODO CALCULAR( ).- CT- 9224. ....	89
ILUSTRACIÓN 128 FORMULA ACTUALIZADA.- CT-9224. ....	89
ILUSTRACIÓN 129 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT-9224. ....	90
ILUSTRACIÓN 130 FORM ACTUALIZADO.- CT-9225. ....	90
ILUSTRACIÓN 131 FORM ANTES DE LA ACTUALIZACIÓN.- CT-9225. ....	90
ILUSTRACIÓN 132 FUNCIÓN TEST( ).- CT-9225. ....	90
ILUSTRACIÓN 133 MÉTODO CALCULAR( ).- CT- 9225. ....	91
ILUSTRACIÓN 134 MÉTODO CALCULAR( ).- CT- 9241. ....	92

ILUSTRACIÓN 135 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 9241.	92
ILUSTRACIÓN 136 NUEVA INTERFACE GRÁFICA.- CT- 9242.	92
ILUSTRACIÓN 137 NUEVAS VARIABLES AGREGADAS.- CT- 9242.	93
ILUSTRACIÓN 138 MÉTODO CALCULAR( ).- CT- 9224.	93
ILUSTRACIÓN 139 FORM ACTUALIZADO.- CT- 9265.	94
ILUSTRACIÓN 140 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 9265.	94
ILUSTRACIÓN 141 VARIABLES NECESARIAS.- CT-9265.	94
ILUSTRACIÓN 142 EVENTO LOAD( ).- CT- 9265.	94
ILUSTRACIÓN 143 MÉTODO CALCULAR( ).- CT- 9265.	94
ILUSTRACIÓN 144 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 9265.	95
ILUSTRACIÓN 145 NUEVA INTERFACE GRÁFICA.- CT- 9267.	95
ILUSTRACIÓN 146 VARIABLES NECESARIAS.- CT- 9267.	95
ILUSTRACIÓN 147 MÉTODO CALCULAR( ).- CT-9267.	96
ILUSTRACIÓN 148 FUNCIÓN CLIC DEL BOTON ACEPTAR.- CT- 9267.	96
ILUSTRACIÓN 149 FORM ACTUALIZADO.- CT- 9282.	96
ILUSTRACIÓN 150 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 9282.	96
ILUSTRACIÓN 151 EVENTO LOAD( ).- CT- 9282.	96
ILUSTRACIÓN 152 MÉTODO CALCULAR( ).- CT- 9282.	97
ILUSTRACIÓN 153 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 9282.	97
ILUSTRACIÓN 154 MÉTODO CALCULAR( ).- CT- 9283.	97
ILUSTRACIÓN 155 MÉTODO CALCULAR( ).- CT- 499.	98
ILUSTRACIÓN 156 VARIABLES Y MÉTODO CALCULAR( ).- CT- 560.	98
ILUSTRACIÓN 157 VARIABLES.- CT- 563.	99
ILUSTRACIÓN 158 MÉTODO CALCULAR( ).- CT- 565.	99
ILUSTRACIÓN 159 NUEVA INTERFACE GRÁFICA.- CT- 566.	99
ILUSTRACIÓN 160 MÉTODO SUB CARGARINTERFAZ( ).- CT- 566.	99
ILUSTRACIÓN 161 PUBLIC SUB NEW( ) Y PUBLIC SUB PASARVALORESDELISTAS( ).- CT- 566.	100
ILUSTRACIÓN 162 VARIABLES NECESARIAS.- CT- 566.	100
ILUSTRACIÓN 163 MÉTODO CALCULAR( ).- CT- 566.	100
ILUSTRACIÓN 164 MÉTODO CALCULAR( ).- CT- 581.	101
ILUSTRACIÓN 165 VARIABLES.- CT -581.	101
ILUSTRACIÓN 166 MÉTODO CALCULAR( ).- CT- 721.	101
ILUSTRACIÓN 167 MÉTODO CALCULAR( ).- CT- 740.	102
ILUSTRACIÓN 168 VARIABLES NECESARIAS Y MÉTODO CALCULAR( ).- CT- 745.	102
ILUSTRACIÓN 169 NUEVA VARIABLE Y MÉTODO CALCULAR( ).- CT- 832.	103
ILUSTRACIÓN 170 MÉTODO CALCULAR( ).- CT- 832.	103
ILUSTRACIÓN 171 MÉTODO CALCULAR( ).- CT- 832.	103
ILUSTRACIÓN 172 NUEVA FÓRMULA.- CT- 832.	104
ILUSTRACIÓN 173 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 832.	104
ILUSTRACIÓN 174 MÉTODO CALCULAR( ).- CT- 2005.	105
ILUSTRACIÓN 175 MÉTODO CALCULAR( ).- CT- 2020.	105
ILUSTRACIÓN 176 FORM ACTUALIZADO.- CT- 2030.	105
ILUSTRACIÓN 177 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 2030.	105
ILUSTRACIÓN 178 FUNCIÓN TEST( ).- CT- 2030.	106
ILUSTRACIÓN 179 PROCEDIMIENTO SUB PASARVALORESDELISTAS( ) Y SUB NEW( ).- CT- 2030.	106
ILUSTRACIÓN 180 NUEVAS VARIABLES.- CT- 2030.	106

ILUSTRACIÓN 181 MÉTODO CALCULAR( ).- CT- 2030.....	107
ILUSTRACIÓN 182 EVENTO LOAD Y PROCEDIMIENTO SUB LLENAR_MATERIAL( ).- CT- 2030. ....	107
ILUSTRACIÓN 183 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 2030. ....	108
ILUSTRACIÓN 184 FORM ACTUALIZADO.- CT- 2035.....	108
ILUSTRACIÓN 185 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 2035.....	108
ILUSTRACIÓN 186 PROCEDIMIENTO SUB PASARVALORESDELISTAS( ) Y SUB NEW( ).- CT- 2035.....	108
ILUSTRACIÓN 187 VARIABLES NUEVAS.- CT-2035. ....	109
ILUSTRACIÓN 188 MÉTODO CALCULAR( ).- CT- 2035.....	109
ILUSTRACIÓN 189 MÉTODO CALCULAR( ).-CT- 2070. ....	109
ILUSTRACIÓN 190 MÉTODO CALCULAR( ).- CT- 2120.....	110
ILUSTRACIÓN 191 MÉTODO CALCULAR( ).- CT- 2130.....	110
ILUSTRACIÓN 192 PROCEDIMIENTO SUB CARGARINTERFAZ( ).- CT- 2140.....	111
ILUSTRACIÓN 193 PROCEDIMIENTO SUB PASARVALORESDELISTAS( ) Y SUB NEW( ).- CT- 2140.....	111
ILUSTRACIÓN 194 FUNCIÓN CLIC DEL BOTÓN CALCULAR Y MÉTODO CALCULAR( ).- CT- 2140. ....	112
ILUSTRACIÓN 195 MÉTODO CALCULAR( ).- CT- 2150.....	112
ILUSTRACIÓN 196 MÉTODO CALCULAR( ).- CT- 2160.....	113
ILUSTRACIÓN 197 NUEVA INTEFAZ GRÁFICA.- CT- 2170. ....	113
ILUSTRACIÓN 198 PROCEDIMIENTO SUB CARGARINTERFAZ( ).- CT- 2170.....	113
ILUSTRACIÓN 199 VARIABLES NUEVAS.- CT- 2170. ....	113
ILUSTRACIÓN 200 MÉTODO CALCULAR( ).-CT- 2170. ....	114
ILUSTRACIÓN 201 FUNCIÓN CLIC DEL BOTÓN ACEPTAR.- CT- 2170. ....	114
ILUSTRACIÓN 202 MÉTODO CALCULAR( ).- CT- 2180.....	114
ILUSTRACIÓN 203 FORM ACTUALIZADO.- CT- 2190.....	115
ILUSTRACIÓN 204 FORM ANTES DE LA ACTUALIZACIÓN.- CT- 2190.....	115
ILUSTRACIÓN 205 PROCEDIMIENTO SUB PASARVALORESDELISTAS.- CT-2190.....	115
ILUSTRACIÓN 206 MÉTODO CALCULAR( ).- CT- 2190.....	115
ILUSTRACIÓN 207 FUNCIÓN CLIC DEL BOTON ACEPTAR.- CT- 2190. ....	116
ILUSTRACIÓN 208 EVENTO LOAD( ).- CT- 2190. ....	116
ILUSTRACIÓN 209 MÉTODO CALCULAR( ).- CT- 2210.....	116
ILUSTRACIÓN 210 MÉTODO CALCULAR( ).- CT- 2260.....	117
ILUSTRACIÓN 211 MÉTODO CALCULAR( ).- CT- 2270.....	117
ILUSTRACIÓN 212 MÉTODO CALCULAR( ).- CT-2330. ....	117
ILUSTRACIÓN 213 MÉTODO CALCULAR( ).- CT- 2350.....	118

# CAPITULO I

## 1. Introducción

Hoy en día podemos observar que a nivel mundial la mayoría de las empresas tienen como objetivo ofrecer productos de gran calidad a bajo costo, manteniendo altos estándares de calidad para satisfacer las necesidades específicas que el cliente requiere. Mediante un estudio de trabajo que permite la obtención de estándares que se utilizan para mejorar aspectos relacionados con labores operativas y gerenciales de la producción.

Uno de los resultados que se logra con el estudio del trabajo (por medición de trabajo) es la definición de los tiempos estándar, que se refieren al tiempo que invierte un trabajador calificado en llevar a cabo una tarea manual definida, efectuando según la norma de ejecución definida.

En el caso de las empresas que se dedican a la fabricación de autopartes para automóviles, el mercado se encuentra muy competido y solo sobreviven aquellas empresas que ofrecen el mejor producto y de mayor calidad.

Siendo uno de los problemas que enfrentan las empresas en el campo de la computación es el alto costo que representa el desarrollo del software. Sin embargo, los problemas con los productos de programación no se limitan a fallas durante la ejecución, si no que incluyen aquellos que surgen en el desarrollo, la administración y el mantenimiento.

Durante años se esperó que los problemas de mantenimiento del software y el envejecimiento de los sistemas, desaparecieran con la introducción de nuevas tecnologías de desarrollo y el remplazo gradual de las ya existentes. Sin embargo, la historia y la experiencia nos muestran que esta es una falsa esperanza.

La reingeniería es una medida directa auxiliada con herramientas automáticas para el mantenimiento del software. En principio, es un reconocimiento de que el problema existe y de que ignorarlo no va a resolverlo. La reingeniería nos brinda una perspectiva en la que el software existente concibe como un activo que debe ser administrado y protegido adecuadamente, que puede ser utilizado nuevamente. Incorpora una visión integral del ciclo de vida del software en el cual el mantenimiento y desarrollo son actividades conjuntas, dado que comparten las mismas metodologías y los mismos tipos de herramientas.

Es importante destacar que la reingeniería no es un procedimiento rígido y lineal, sino un conjunto de estrategias que pueden ser aplicadas total o parcialmente en un problema dado, de acuerdo con las características de este y al criterio del grupo de trabajo que está encargado de realizar la tarea de mantenimiento. (Héctor Francisco Bautista Gonzalez, Cuauthemoc Freyre Mercado, Norma Susana Zavala Carrasco, 1995)

A nivel mundial grupo MAHLE ésta catalogado entre los proveedores automotrices más importantes del sector y ha logrado posicionarse entre los tres principales productores de partes para sistemas de pistones y cilindros. (Canales de Información de México, 2014)

Es por esto que gracias a este tipo de empresas automotrices se generan muchos de los empleos en la entidad, generando un fuerte impacto a la economía del estado.

Actualmente en la ciudad de Aguascalientes, se encuentra Mahle componentes para motor de México S. de R.L. de C.V., empresa dedicada a la fabricación de anillos para pistón, y tienen como uno de sus principales objetivos la calidad de sus productos con lo cual asegura el prestigio internacional con sus clientes. Dentro de su diversidad de productos se encuentra la producción de anillos para el mercado doméstico, equipo original, anillos para locomotora y plantas de energía. Esta empresa actualmente exporta sus productos a diferentes partes del mundo.

Mahle componentes para motor de México S. de R.L. de C.V. compite contra otras empresas ofreciendo productos de alta calidad bajo estándares internacionales. Para el mercado industrial produce componentes para sistemas de refrigeración, filtración y aire acondicionado, y también participa destacadamente en el segmento de grandes motores como trenes, barcos y generadores de energía.

Como uno de los 30 proveedores automotrices a nivel mundial y como un socio líder en el desarrollo global de la industria automotriz y del motor, MAHLE ofrece sistemas únicos de competencia en motores de combustión y en componentes periféricos del motor. (Canales de Información de México, 2014, pág. 46.)

## **1.1 Antecedentes.**

MAHLE es una empresa alemana de carácter global conformada por tres unidades de negocio:

- Componentes de motor

- Sistemas de Filtración y Componentes Periféricos
- La unidad de negocio Aftermarket

Fue fundada en 1920 por los hermanos Ernst y Herman Mahle, empezando a trabajar en producción con 6 personas en Stuttgart, Alemania. Hoy en día, uno de cada dos automóviles que se están produciendo en el mundo, contiene componentes MAHLE, además la empresa está presente en los cinco continentes y cuenta con 140 centros de producción y genera más de 65 mil empleos a nivel global. (Canales de Información de México, 2014)

Mahle componentes para motor de México S. de R.L. de C.V. es una empresa integrante del Grupo Mahle Alemania, la cual inicia operaciones en 1953, cuando a iniciativa de un grupo de inversionistas mexicanos se crea la empresa “Conductores Eléctricos, S.A. de C.V.”, en asociación con industriales de Estados Unidos de Norteamérica.

Arranca así en 1954 la fabricación de Conductores Eléctricos en la colonia Industrial Vallejo, del Distrito Federal, generando empleo para 48 personas. Pocos meses después se asocia con industriales italianos, por lo cual en la década de los 60's, la empresa logra ser fabricante de conductores eléctricos sustituyendo la importación de estos productos, y convirtiéndose asimismo en exportador.

En la década de los 70's, la empresa inicia su diversificación hacia la fabricación de plásticos, desde las materias primas, productos intermedios, hasta los artículos manufacturados. (Juárez López, 2011)

Actualmente Mahle cuenta con más de 6 plantas industriales distribuidas a lo largo del territorio nacional, y genera más 4,000 empleos directos en el territorio nacional. Por lo anterior en Mahle se contempla con gran optimismo el futuro de su grupo, consiente de las áreas en que se desenvuelven son de gran importancia para el desarrollo del país. (Canales de Información de México, 2014)

### **1.1.1 Historia MAHLE componentes para motor de México.**

Mahle pertenece a uno de los grupos industriales más importantes de México, Grupo Mahle, al cual a su vez pertenecen innumerables empresas de renombre en nuestro país y el extranjero, entre las de mayor relevancia encontramos Mahle camisa Saltillo, Clemex entre muchas otras más.

En México más de cuatro mil empleados conforman la familia de Mahle componentes para Motor de México, los centros de trabajo están distribuidos en 6 plantas, 2 ubicadas



en el Estado de México; Naucalpan y Lerma, donde, además, está el Centro de Distribución After Market para mercado de repuesto y equipo original, una en Nuevo León (Santa Catarina), 2 en Coahuila (Ramos Arizpe) y una más en la ciudad de Aguascalientes. (Canales de Información de México, 2014)

En el sector industrial también están presentes con los anillos de gran diámetro o “Large Bore” utilizados en motores de barcos, trenes y generadores de energía, así como los anillos de diámetro pequeño para podadoras, equipos de refrigeración o compresores compactos, por mencionar solo algunos. (Canales de Información de México, 2014)

MAHLE componentes de Motor planta Aguascalientes ocupa hoy una superficie de más 104 mil metros cuadrados, de los cuales 30,000 están edificados y destinados para oficinas técnicas, administrativas y áreas productivas. ubicada en el Km. 0.3 Carretera Maravillas – Jesús María Aguascalientes, México. Siendo el gerente de planta el Ing. Julián Carrillo, y el coordinador del departamento de ingeniería de rutas el Ing. Jorge Humberto Rodríguez Barrientos. En este departamento es donde se llevan a cabo todos los procesos de maquinado de los anillos, expansores, segmentos, franklin y Large Bore, tanto de compresión como de control de aceite.

Su misión es el diseño, desarrollo, fabricación y comercialización de anillos para pistón, dirigido al mercado de al mercado de motores de combustión interna y compresores, garantizando la satisfacción de sus clientes, su personal y accionistas. (Canales de Información de México, 2014)

La tecnología utilizada en la fabricación de los anillos proviene de Estados Unidos y Alemania, y sus operaciones en México datan de 1958, precisamente en Naucalpan Estado de México, para posteriormente trasladar las operaciones de maquinados a Aguascalientes, donde se iniciaron operaciones de producción en el año 1986. (Juárez López, 2011)

Por siempre esta empresa se ha distinguido por la calidad de sus productos, su liderazgo en los mercados y sobre todo por su filosofía de trabajo, basada en lograr la satisfacción amplia y consistente de las necesidades y expectativas de los clientes y también de un auténtico interés por el desarrollo y la superación de su personal, lo cual la han llevado a recibir innumerables reconocimientos nacionales e internacionales.

Gracias a la dedicación permanente y al compromiso de toda su gente, MAHLE Aguascalientes ha sido reconocida con las siguientes certificaciones: ISO TS 16949

(anteriormente QS9000) distinción alcanzada en septiembre de 1996 y que se mantiene hasta la fecha, C-TPAT, OHSAS 18001, INDUSTRIA LIMPIA e ISO 14001.

Además, en 2013 obtuvo también la certificación Q1 de Ford y el reconocimiento como Proveedor de Excelencia por General Motors, galardones conseguidos por lograr menos de un PPM, entregas al 99% y la reducción a cero de backlog en Aftermarket. (Canales de Información de México, 2014)

Esta empresa de autopartes en el estado de Aguascalientes, enorgullece al grupo Mahle, al reconocer que el desarrollo alcanzado se debe a la cooperación, lealtad y entrega del personal que la integra.

La empresa inicio sus operaciones en 1958, en Naucalpan, Edo. de México, como Sealed Power de México, con capital 100% extranjero y tecnología de Sealed Power Corporation.

En 1986 el grupo Condumex adquirió el 60% de la empresa, manteniéndose el 40% restante bajo el control de Sealed Power Division USA y continuando con la relación de Licencia Tecnológica.

En 1986 se iniciaron las operaciones en planta Aguascalientes con la razón social de SPIMEX, fabricando juegos completos de anillos para pistón, para el mercado de repuestos.

Entre 1988 y 1989 se transfirieron de Naucalpan Edo. de México a Aguascalientes las líneas de segmentos y de maquinado final de anillo de compresión.

Durante 1993 se realizó la transferencia de las operaciones iniciales de maquinado, manteniéndose en Naucalpan la fundición del hierro gris.

La planta de Naucalpan suministra "Castings" (nombre que recibe la materia prima) a la planta en Aguascalientes para la fabricación de anillos de compresión de hierro. Ambas plantas tienen la misma razón social, pero para fines operativos su relación es la de cliente – proveedor. (Juárez López, 2011)

Durante Julio del 2007 la empresa es comprada por el grupo industrial MAHLE, la cual tiene presencia en todo el mundo. Iniciado una nueva era de crecimiento y expansión que los ha consolidado como el fabricante número uno en estos componentes en toda América Latina. (Canales de Información de México, 2014)

### **1.1.2 Política de calidad.**

En Mahle componentes de motor de México el compromiso es lograr satisfacer en forma amplia y consistente las necesidades de los clientes, a través del suministro de productos y servicios competitivos.

Tenemos el compromiso de fortalecer la empresa, a través de un proceso planeado y sistemático de mejoramiento continuo en todas las fases del negocio, basados en los siguientes objetivos:

#### **1.1.2.1 Objetivos de Calidad.**

- Mantener un sistema de calidad certificado (TS-16949) como base del mejoramiento continuo en toda la organización.
- Implantar y mantener sistemas de desarrollo para todo el personal con el fin de aumentar la productividad y asegurar su permanencia.
- Generar una cultura permanente de reducción de costos.
- Fomentar el trabajo en equipo.
- Comprometer a nuestros proveedores clave, a cumplir los requerimientos del TS-16949.

### **1.1.3 Principios y valores de la compañía.**

#### **1.1.3.1 Principios.**

- Asegurar la presencia dominante de la empresa, en el mercado nacional y buscar una creciente participación en el mercado internacional.
- Cuidar contantemente la rentabilidad de los productos

#### **1.1.3.2 Valores.**

- Orientar al Grupo MAHLE hacia la competitividad y la rentabilidad.
- Satisfacer con eficiencia, calidad y servicio las necesidades de los clientes.

### **1.1.4 Nuestros Clientes.**

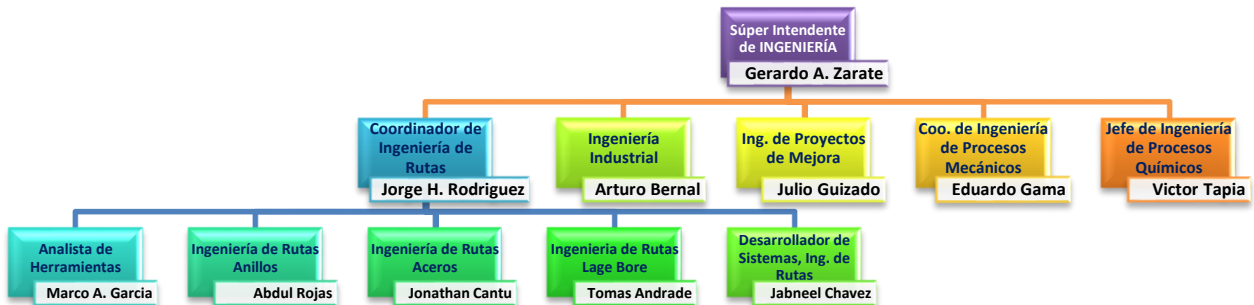
Más de 140 fabricantes de motores y de vehículos con renombre equipan sus motores de serie con los componentes de motor y filtros de MAHLE. Por mencionar algunos he aquí una pequeña selección:

Alfa Romeo, Audi, BMW, Case New Holland, Caterpillar, Citroën, Cummins, Daewoo, DAF, Deutz, Ducati, Ferrari, Fiat, Ford, General Motors, Harley Davidson, Hatz, Perkins, Peugeot, Chrysler entre otros. (Juárez López, 2011)

### 1.1.5 Organigrama Funcional de Mahle Componentes para Motor de México Planta Aguascalientes.



#### 1.1.5.1 Organigrama Ingeniería.



## 1.2 Planteamiento del Problema.

Dentro del departamento de ingeniería de rutas cuentan con un sistema que les permite crear un nuevo formato con las especificaciones de las cotizaciones realizadas, agilizando el tiempo en que realizan los cálculos necesarios por los que pasa cada componente. Al pasar el tiempo se han detectado varios errores dentro del sistema “RGP”, y por la constante actualización del tiempo estándar por los procesos de mejora continua, ha surgido la necesidad de aplicar reingeniería al sistema “RGP” para actualizar las fórmulas que calculan el tiempo estándar de cada uno de los centros de trabajo activos en planta al realizar las cotizaciones del herramental necesario en las hojas de ruta y la salida al SAP.

Las hojas de ruta requieren del cálculo del tiempo estándar, que actualmente se obtiene mediante una serie de operaciones realizadas por el sistema "ROUTING GENERATION PROGRAM". Las fórmulas con las que cuenta el sistema actualmente están obsoletas, por la mejora continua de la empresa, se han estandarizado y reducido los tiempos estándar, en comparación con los que contaban en los centros de trabajo dentro del sistema.

### **1.3 Objetivos.**

#### **1.3.1 General**

Implementar reingeniería y la actualización de las fórmulas necesarias para realizar el cálculo del tiempo estándar de expansores, anillos, segmentos, franklin y Large Bore en el sistema "ROUTING GENERATION PROGRAM" al momento de realizar las cotizaciones de los herramientas requeridos y generar la hoja de ruta. Así como la elaboración de nuevas interfaces para los centros de trabajo inexistentes en el sistema.

#### **1.3.2 Específicos**

- Actualizar las fórmulas para calcular los tiempos estándar dentro del sistema "ROUTING GENERATION PROGRAM" para cada uno de los centros de trabajo activos en planta.
- Generar nuevas interfaces para los centros de trabajo faltantes en el sistema.
- Aplicar reingeniería al sistema para extender su utilidad y tiempo de vida en planta.

### **1.4 Justificación.**

Con la finalidad de que el sistema "ROUTING GENERATION PROGRAM" siga teniendo un buen funcionamiento y este se mantenga actualizado en las fórmulas para obtener el tiempo estándar de manera automatizada, a través de los datos generados mediante el cálculo de la hoja de ruta en el sistema, para agilizar los cálculos. Se optó por realizar las siguientes actividades:

- Actualizar y revisar las fórmulas para realizar el cálculo de los tiempos estándar por operación para anillos, large bore, franklin, segmentos y expansores.
- Generar nuevos centros de trabajo que no se encuentran en existencia dentro del sistema, pero si están en operación en planta.
- Dar solución a problemas encontrados eventualmente dentro del sistema "RGP".

# CAPITULO II

## 2. Marco Teórico

### 2.1.1 Mantenimiento del software

Los problemas de mantenimiento de un producto de programación han crecido gracias que se cree que el mantenimiento es más sencillo de llevar a cabo que el desarrollo, por lo que puede ser realizado por personal con menos experiencia, herramientas y dirección. Por lo contrario, el mantenimiento del software generalmente presenta más retos que el desarrollo. ¿Por qué?, ¿Qué es exactamente lo que hacen las personas de mantenimiento? (Olgín, 1997)

#### 2.1.1.1 Definiciones

El mantenimiento es el último proceso en el ciclo de vida del software; a través de los años se han propuesto muchas maneras de definirlo, a continuación, se listan algunas.

Mantenimiento es la modificación de un producto de software para mejorar su desempeño y demás atributos o adaptarlo para el cambio a un nuevo ambiente. (IEEE Standard Glossary Software Engineering Terminology)

Mantenimiento es el mecanismo para combatir el deterioro del software que con el tiempo tienda a ser no estructurado, ilegible o resistente a cambios.

Mantenimiento incluye actualizaciones a las aplicaciones existentes, consiste en hacer cambios a un programa de software después de que ha sido puesto en producción.

Mantenimiento es adaptar el software contantemente para las necesidades de la empresa. (Olgín, 1997)

Estas definiciones identifican tres actividades fundamentales:

- Corregir los errores
- Revisar los requerimientos originales.
- Mejorar el funcionamiento, aumentar su desempeño e introducir funciones nuevas.

### **2.1.1.2 Naturaleza del trabajo de mantenimiento**

Frecuentemente el mantenimiento se relaciona con la corrección de errores, el argumento es que el mantenimiento correctivo es la actividad que más se realiza. Sin embargo, muchos estudios muestran que esto es incorrecto.

La mayor parte de los trabajos de investigación en mantenimiento han identificado al trabajo de mantenimiento adaptivo y perfectivo como las actividades dominantes en las organizaciones. (Olgín, 1997)

### **2.1.1.3 Factores que afectan el esfuerzo de mantenimiento**

Los factores que incrementan el esfuerzo de mantenimiento son:

- El tamaño del sistema.
- La edad.
- La familiaridad del personal con el sistema.
- El nivel de experiencia.

Los sistemas de software más grandes requieren de mayor esfuerzo de mantenimiento que los sistemas más pequeños. Existe una curva de aprendizaje asociada con los sistemas grandes; entre más grandes son, las funciones son más complicadas y complejas. También, los sistemas viejos requieren de mayor mantenimiento que los sistemas nuevos. Los sistemas de software tienden a crecer con el tiempo, a encontrarse menos estructurados al ser modificados, y ser menos entendibles cuando se cambia el grupo de personas que lo manejan. Una gran cantidad de trabajo de mantenimiento se refiere al mantenimiento de corrección, especialmente rutinas de corrección de errores para programas grandes y viejos.

Así mismo, una gran parte del trabajo de mantenimiento se gasta en mantenimiento correctivo por caídas del sistema. Personas de mantenimiento que no formaban parte del grupo original que desarrollo el sistema, dedican más tiempo tratando de entender el sistema, ya que no están familiarizados con la estructura de éste ni con la forma de trabajar de los autores originales.

Estos factores predicen un incremento en el mantenimiento del sistema conforme este va creciendo o habiendo cambios en el personal. (Olgín, 1997)

## **2.1.2 Reingeniería**

La reingeniería es la manera de realizar el mantenimiento de una forma automática, aplicando herramientas, técnicas y metodologías para extender la vida útil de un sistema a un bajo costo. La reingeniería de programación considera el mejoramiento

del proceso de mantenimiento al sugerir una estrategia a largo plazo, en lugar de ejecutar los cambios como se van presentando.

La reingeniería es el proceso de examina un sistema de software ya existente (programa) y modificarlo con la ayuda de herramientas automáticas para:

- Incrementar la disposición del software que recibirá mantenimiento.
- Incrementar su nivel tecnológico.
- Extender sus expectativas de vida.
- Capturar sus componentes en una biblioteca donde las herramientas case se puedan utilizar para su soporte.
- Incrementar la productividad en el mantenimiento. (Olgín, 1997)

#### **2.1.2.1 Propósitos de la Reingeniería**

- Mejorar el manejo de los sistemas existentes.
- Proveer asistencia automatizada para el mantenimiento.
- Reducir errores y costos de mantenimiento.
- Hacer sistemas fáciles de entender, modificar y analizar.
- Otorgar sistemas de conversión y migración.
- Forzar a utilizar estándares en los desarrollos (nuevos y antiguos).
- Mejorar el tiempo de respuesta a las solicitudes de mantenimiento.
- Mejorar el mantenimiento.
- Proteger y extender la vida de los sistemas.
- Utilizar case para soportar los sistemas actuales.
- Reutilizar los componentes de los sistemas actuales.

La reingeniería puede ayudarnos a entender sistemas existentes y a descubrir componentes (arquitectura, estructura de datos) de software que sean comunes a lo largo de los mismos sistemas. Estos componentes pueden ser usados -o reusados- en la fabricación de nuevos sistemas, disminuyendo así el tiempo de desarrollo. (Olgín, 1997)

#### **2.1.2.2 Tipos de Reingeniería**

##### **2.1.2.2.1 Reestructuración**

Es el proceso de cambiar la forma del software (e. g., nombres de datos y definiciones y fuentes de código de programa) sin alterar su funcionalidad. La razón principal de la reestructuración es hacer que el programa sea más fácil de entender. (Olgín, 1997)



#### **2.1.2.2.2 Ingeniería Inversa**

Es el proceso de analizar el sistema para construir una descripción de sus componentes y de sus interrelaciones entre sí. El resultado es una descripción de alto nivel (diagramas de flujo, diagramas entidad relación, código fuente, etc.) del programa a partir de sus niveles más bajos de información (en muchos casos el código fuente). El propósito de la ingeniería inversa es el de actualizar la documentación o volver a documentar el sistema y descubrir su diseño como una ayuda para hacer el programa más entendible o para migrarlo a una nueva tecnología. (Olgín, 1997)

#### **2.1.2.2.3 Administración de configuración y cambios**

Es la actividad de generar y organizar la información referente a la evolución de los programas que se encuentran en etapas de desarrollo o de mantenimiento. Permite administrar las bibliotecas que almacenan estos datos y controlar los cambios que se efectúan a las diferentes versiones del producto.

Aunque los conceptos envueltos en la reingeniería no son nuevos, la idea de desarrollar los sistemas viejos y actualizarlos dentro de una nueva estrategia, utilizando los mismos conceptos que en el desarrollo si lo es. Los elementos aplicados a la reingeniería incorporan varias de las mejores ideas del pasado y del presente de la ingeniería de software, lo que permite que se puedan aplicar a una gran variedad de empresas, no importando la metodología, el hardware, ni mucho menos el lenguaje utilizado en sus sistemas. Sin embargo, si deberá existir una metodología de desarrollo y un ciclo de vida de software, así como políticas y procedimientos para aceptar y controlar los cambios realizados. Al contar con estos requisitos, la labor de reingeniería puede ser implementada utilizando las mismas técnicas que se emplean actualmente en el desarrollo de nuevos sistemas, pero aplicándose a los sistemas existentes.

Para muchas organizaciones la reingeniería no es una opción, es una necesidad si quieren proveerse de software con costos moderados que les ayuden a alcanzar una ventaja competitiva.

Probablemente el beneficio más importante que la reingeniería lleve consigo, es el unir las funciones de desarrollo del software y de mantenimiento en una sola, esto se logra incorporando las mismas herramientas y técnicas en las dos fases del ciclo de vida del sistema. Información acerca del sistema, de sus componentes y de las interrelaciones entre si después de aplicar la reingeniería pueden ser utilizados por herramientas CASE para proveer un soporte en el futuro. (Olgín, 1997)

#### **2.1.2.3 Reingeniería contra reemplazo**

Sistemas frágiles que son altos candidatos para la reingeniería, son sistemas que:

- Son de notable importancia para la corporación.
- Son el blanco de frecuentes trabajos de mantenimiento y requieren de un gran porcentaje de los recursos asignados al mantenimiento.
- Solamente uno o unas cuantas personas selectas entienden el sistema y pueden realizar cambios en el.
- Contiene errores que nadie puede encontrar.
- Se requiere actualizarlos o incrementar su tecnología.

(Olgín, 1997)

### **2.1.3 Tiempo Estándar**

Según la norma ANSI STANDARD Z94.0-1982, se define el tiempo estándar como: el valor de una unidad de tiempo para la realización de una tarea como lo determina la aplicación apropiada de las técnicas de medición de trabajo efectuada por el personal calificado. Por lo general se establece aplicando las tolerancias apropiadas al tiempo normal.

El tiempo normal es “el tiempo que requiere un operario calificado para realizar una tarea, a un ritmo normal, para completar un elemento ciclo u operación usando un método prescrito”. (Rojano, 2011)

La tolerancia es “el valor o porcentaje de tiempo mediante el cual se aumenta el tiempo normal, para la cantidad de tiempo improductivo aplicada, para compensar las causas justificables o los requerimientos de normas generales que necesita un tiempo de desempeño que no se mide en forma directa para cada elemento o tarea.

Teóricamente, para la determinación de un tiempo estándar las condiciones de producción deben estar estables, de tal forma que no existan problemas de diseño, reprocesos, retrasos de máquinas, debe haber equilibrio entre fuerza laboral, materiales y capacidad de producción. Sin embargo, en la práctica estas condiciones no siempre existen por lo tanto deben considerarse otros factores denominados tolerancias (tiempo improductivo).

Para establecer el tiempo estándar se usan los datos estándar, que consisten en la organización de los elementos de trabajo en bloques constructivos útiles y bien definidos, cuyo número depende de la exactitud deseada de la naturaleza del trabajo y de la flexibilidad necesaria. Esta información generalmente se usa como base para elaborar los estándares de tiempo en un trabajo que es semejante a aquel de donde se hizo el estudio, sin la necesidad de determinar nuevamente los tiempos. (Escuela de Administración de Negocios, 2016)

### 2.1.3.1 Aplicación del Tiempo Estándar

Los tiempos estándar o tiempos tipo de fabricación o prestación, son la base para una serie de aplicaciones a nivel industrial y de servicio, aplicaciones sin las cuales las organizaciones difícilmente subsisten y entre ellas tenemos:

- Determinación de las necesidades y los costos de mano de obra directa en un producto, lo que permite estimar el costo de producción.
- Para obtener información en base a programación y control de la producción, lo que permite optimizar la utilización de los recursos y minimizar los tiempos de manufactura.
- Para evaluar métodos de producción alternos como opción para elaborar otros artículos en la búsqueda de una mejor eficiencia.
- Para determinar un tiempo de trabajo aceptable y poder aplicar a los trabajadores un programa de incentivos por producción.
- Para estimar los tiempos de producción cuando existe algún cambio en la materia prima.
- Para incluir mejoras en procesos de baja eficiencia, operación lenta y o costos excesivos.
- Para determinar la capacidad de planta (distribución)
- En la evaluación de compra de equipo más productivo.
- En la estimación de tiempos confiables de entrega a los clientes.
- En la determinación del costo total de producción.
- Para la determinación del precio de venta.
- En la realización de programas y presupuestos de producción.
- Implementación de controles de mano de obra.
- Balanceo de línea.

El tiempo estándar ésta compuesto de varios factores, según se muestra en la siguiente figura:



Ilustración 1 Factores del tiempo estándar

### 2.1.3.2 Tiempo Básico.

El tiempo básico se define como “tiempo mínimo reducible que se calcula a partir de los tiempos elementales de una tarea de trabajo”. Una tarea de trabajo es un conjunto de actividades necesarias para completar la ejecución de un proceso o producto. Cada tarea está compuesta de varios movimientos elementales.

### **2.1.3.3 Tiempo Suplementario.**

Todo proceso de producción está sujeto a variaciones inevitables que se originan de acuerdo a las características humanas y de los sistemas involucrados. El tiempo suplementario es el tiempo que se consume por deficiencias en los productos y procesos, diseños y fatiga.

El tiempo suplementario se calcula a partir de un porcentaje sobre el tiempo básico y se establece a partir de un estudio particular de cada empresa.

### **2.1.3.4 Tiempo Improductivo.**

A pesar de que forma parte del tiempo estándar, es importante separarlo porque se origina en forma independiente de aspectos como diseño, método y especificaciones del producto. Se divide básicamente en dos aspectos:

Por deficiencia de la dirección: corresponden a los retrasos ocasionados por circunstancias operativas no previstas entre las que se pueden citar: Falta de planificación, cambios improvisados en el proceso productivo y malas condiciones de trabajo.

Por deficiencia de los trabajadores: tiempos improductivos causados por el personal involucrado directamente en los procesos de manufactura; por ejemplo: llegadas tardías o pérdida de tiempo, ausencias, repeticiones por descuido del trabajador o accidentes por negligencia.

### **2.1.3.5 Tolerancias.**

Para determinar los porcentajes de tiempos suplementarios e imprevistos de tiempo total estándar, se recurre a las tolerancias, las cuales son la magnitud adicional tolerable que se aplica al tiempo normal.

Es un aspecto muy controvertido, debido a que depende de los elementos, no son negociables con los trabajadores y si son poco realistas pueden invalidar el tiempo estándar. Lo ideal es obtener los datos que se registran en la empresa en aspectos como necesidades personales, fatiga, demoras, etc. Existen clasificaciones de tolerancias a saber:

Necesidades personales: tomar agua, usar servicios sanitarios, etc. Se recomienda emplear 5% que equivale a 24 minutos en una jornada de 8 horas.

Fatiga: corresponde a la disminución de la capacidad de ejecución de un trabajo por causas físicas y psicológicas, producidos por factores como cantidad de luz, temperatura, humedad, ruido, salud, edad, dieta, etc.

En general se recomienda el 4% sobre el tiempo normal, sin embargo, puede pasar de valores que van desde 2% (estar de pie) a 22% (empleo de fuerza muscular al levantar 60 libras).

Demoras evitables: se originan por interrupciones, irregularidad de materiales, interferencias de máquinas, etc. Estos se calculan por muestreo de trabajo. Demoras evitables: son causa de actividades como visitas a otros empleados, ociosidad, fumar o comer en horas de trabajo, etc.

Extraordinarias: situaciones especiales que rara vez se presentan en el trabajo.

#### **2.1.3.6 Mantenimiento de tiempos estándar.**

Debido a que los tiempos estándar están basados en un método específico, estos deben estar sujetos a mantenimiento debido a que generalmente se incluyen mejoras con el tiempo y se vuelven obsoletos por los cambios progresivos que se hace en los procesos.

#### **2.1.3.7 Sistemas de tiempos predeterminados.**

Los estándares predeterminados son el tiempo de trabajo que se establece de acuerdo a la definición de una labor en termino de elementos muy pequeños, donde la suma total de elementos define el tiempo básico. Por lo general, los tiempos predeterminados se reconocen como los más importantes desde el punto de vista de especificación de los métodos y de la exactitud.

Existen varios métodos para la elaboración de tiempos predeterminados, los cuales se conocen por sus siglas en inglés como se describen a continuación.

MTA (Motion Time Analysis), WF (Work Factor), BTM (Basic Motion Time Study), MTM 1,2,3 (Methods Time Measurement), UAS (Universal Analysing System), MOST (Maynard's Operation Sequence Techniques), MICRO (Micro Motion Analysis), MODAPTS (Modular Arrangement of PTS), MACRO (Macro Motion Analysis) MCD (Master Clerical Data).

Aunque se conoce que dichos procedimientos tienen una confiabilidad de 5%, los más utilizados son los métodos MOST y MTM. (Escuela de Administración de Negocios, 2016)

#### **2.1.3.8 Cálculo del Tiempo Estándar**

La etapa del cálculo del tiempo estándar marca el inicio del trabajo de oficina en el estudio de tiempos, aunque es muy probable que el especialista en medio del análisis

considere necesario apoyarse nuevamente en la observación de las operaciones. Esta fase no requiere gran dominio aritmético, por lo que consiste en cálculos comunes y corrientes que puede efectuar el analista en muy poco tiempo, una aplicación o una hoja de cálculo.

Requiere eso sí, de una gran capacidad de análisis de consistencia de los datos obtenidos en la fase de observación, y un evidente conocimiento de las medidas a tomar dependiendo de la situación que se presente.

### **2.1.3.9 De los Tiempos Observados al Tiempo Estándar**

El hecho de convertir una serie de tiempos observados en tiempos tipo o estándar, requiere de la aplicación sistemática de una serie de pasos en los que se hará importante que el analista tenga claridad respecto a la base teórica del cronometraje del trabajo, la valoración del ritmo y los suplementos del estudio.

Al efectuar el cálculo del Tiempo Estándar se debe considerar lo siguiente:

Como se asignarán los elementos contingentes.

Si debe concederse el tiempo de preparación y retiro.

El factor de interferencia cuando se presente un ciclo de trabajo estudiado. (López, Cálculo del Tiempo Estándar o Tipo - Ingeniería Industrial, 2016)

### **2.1.4 ERP (Enterprise Resource Planning).**

Conocido por sus siglas en inglés como “ERP” o sistemas de planificación de recursos de la empresa, se refiere al software que modifica la estructura interna de la empresa, ya que son sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa.

#### **2.1.4.1 Características**

Se caracterizan por estar compuestos de diferentes partes que tienen diferente uso y están integradas en una única aplicación, por lo que solo se puede definir el “ERP” como la integración de estas partes:

Producción, Ventas, Compras, Logística, Contabilidad (de varios tipos), Gestión de Proyectos, Sistema de Información Geográfica (GIS), Inventarios y control de almacenes, pedidos, nominas, entre otros.

El “ERP” integra todo lo necesario para el funcionamiento de los procesos de negocio de la empresa. No podemos hablar de ERP en el momento en que solo se integra un

negocio o una pequeña parte de los procesos de negocio. La definición de “ERP” indica la necesidad de que: esté disponible toda la información para todo el mundo todo el tiempo. (Hernández, 2014)

Lo más destacable de un “ERP” es que unifica y ordena toda la información de la empresa en un solo lugar, de este modo cualquier suceso queda a la vista de forma inmediata, posibilitando la toma de decisiones de forma más rápida y segura, acortando los ciclos productivos. Con un “ERP” tendremos la empresa bajo control e incrementaremos la calidad de nuestros servicios y productos.

La implantación de un “ERP” conlleva la eliminación de barreras inter departamentales, la información fluye por toda la empresa eliminando la improvisación por falta de información.

Los “ERP” son una evolución de sistemas MRP, los cuales estaban enfocados únicamente a la planificación de materiales y capacidades productivas. Los ERP disponen de herramientas para efectuar la planificación de los trabajos en planta. Está planificación se efectúa enfrentando los requerimientos materiales y capacidad de los productos a fabricar contra las existencias y capacidades sin asignar.

Los “ERP” son el núcleo de otras aplicaciones como pueden ser el CRM (Gestión de las Relaciones con los Clientes), Data Mining (Conversión de datos en información útil), etc. (LIDER Integrated Technology Consulting S.A., 2016)

Un sistema “ERP” debe cumplir al menos las siguientes características que se destacan al estudiar los autores consultados.

#### **2.1.4.2 Integralidad**

Los sistemas “ERP” permiten controlar los diferentes procesos de la compañía bajo la premisa de que todos los departamentos de una empresa se relacionan entre sí, es decir que el resultado de un proceso es punto de inicio del siguiente. Controlan y ejecutan las transacciones que constituyen la actividad habitual de la empresa. Esa característica permite efectuar mediciones y seguimiento a los resultados de la compañía y determinar el cumplimiento de los objetivos propuestos. (Lopez Hermoso, 2000)

#### **2.1.4.3 Modularidad**

Los sistemas “ERP” pueden instalarse de acuerdo con los requerimientos del cliente. En la medida que se requieran las funcionalidades se determinan cuáles son los módulos necesarios de configurar. (Muñiz, 2000)

#### **2.1.4.4 Adaptabilidad**

Los sistemas “ERP” pueden adaptarse a la idiosincrasia de cada empresa. Esto se logra por medio de la configuración o parametrización de los procesos de acuerdo con las salidas que necesite cada uno. (Carmen de Palos Heredero, Lopez Hemoso Agius, Martin Romo Romero, Santiago, Sonia Medina Salgado, 2004)

Los sistemas “ERP” tienen entre sus objetivos principales el satisfacer las diferentes necesidades de información de la empresa para conseguir que los gestores empresariales dispongan de un soporte para tomar decisiones y controlar el cumplimiento de objetivos.

Los sistemas “ERP” son sistemas integrados en contraposición con el software a la medida diseñado para un cliente en particular. Esto implica que cuando se adquiere un sistema ERP, se obtiene una versión estándar de un producto. Este producto no está diseñado para la empresa que lo compra, pero si tiene algunas características que lo hacen adaptable a las operaciones centrales de la empresa. Para adaptarlo a las necesidades de la empresa, es necesario realizar modificaciones y parametrizaciones; estas adaptaciones, por regla general, no deben superar el 30% de la solución informática completa para garantizar el éxito de la implantación. (Vásquez, 2011)

#### **2.1.4.5 Ventajas de un ERP.**

- Disminuyen los problemas sobre la veracidad de la información.
- Mejor manejo de inventarios (producto terminado, materia prima).
- Mejores transacciones entre empresas.
- Mejor comunicación dentro de la empresa.
- Pronósticos de cambio.
- Control de calidad.
- Tiene una sola base de datos centralizada.
- Los componentes de ERP interactúan entre si consolidando todas las operaciones.
- En un sistema ERP los datos se integran solo una vez y son consistentes, completos y comunes.



- Aunque a veces las empresas que lo implantan suelen tener que modificar alguno de sus procesos para alinearlos con los del sistema ERP, en muchos casos no es necesario.
- La tendencia actual es ofrecer aplicaciones especializadas para determinadas empresas, los que se conoce como versiones sectoriales o aplicaciones sectoriales especialmente indicadas o preparadas para determinados procesos de negocio de un sector (los más utilizados).

El mercado de sistemas “ERP” a nivel mundial está dominado por SAP de Alemania con el mayor nivel de facturación en el año 2010.

### **2.1.5 SAP (Systems, applications and products)**

Fundada en 1972, SAP es el líder mundial de soluciones de software de negocios colaborativas para todo tipo de industria y para cada uno de los principales mercados. Con 12 millones de usuarios, 84,000 instalaciones y 1,500 socios de negocio, SAP es la compañía de software empresarial más importante, y el tercer proveedor independiente de software en el mundo.

SAP emplea a más de 30,000 personas en más de 50 países. Nuestros profesionales están comprometidos en proveer soporte y servicios de alto nivel a los clientes. Las siglas SAP (System, Applications and Products) identifican a una compañía de sistemas informáticos con sede en Alemania, que se introdujo en el mercado de los sistemas de información con un producto denominado SAP R72, antecesor al SAP R/3. Estas son algunas cifras de Éxito en el mundo: 5º mayor proveedor de software en el mundo. Líder de mercado con 34% de cuota de mercado en soluciones de gestión empresarial. 25 años de experiencia en negocio. Más de 6,000 clientes en más de 50 países. Más de 8,000 instalaciones de SAP R/3 y más de 2,200 instalaciones de SAP R/2.

Las ventas de grupo sumaron más de 1,800 millones de dólares en 1995; el 70% fue generado fuera de Alemania. La versión SAP R/3 se desarrolló en 1989. Desde entonces hasta ahora, no ha dejado de evolucionar.

Las principales diferencias técnicas respecto a la versión anterior SAP R/2, son: Arquitectura cliente/servidor, Entorno grafico puede desarrollarse sobre diferentes tipos de plataformas informáticas y sistemas de bases de datos; todas estas características permiten al usuario de SAP disponer de un sistema más potente, rápido y eficaz, enfocado a facilitar su operativa diaria.

Desde el punto de vista funcional y de su arquitectura técnica, SAP R/3 puede definirse como un software abierto, basado en la tecnología cliente/servidor, diseñado para manejar las necesidades de información de una empresa.

SAP R/3 es el software de estas características de mayor divulgación en todo el mundo, contando con más de 18,000 instalaciones en más 100 países. Es la versión mejorada de un producto anterior (sistema R/2) que ha permitido a SAP AG convertirse en la empresa líder de software empresarial, sin embargo, no se limita a ser un simple paquete de programas informáticos; SAP R/3 va más allá: supone todo un equipo (de personal, programas, comunicaciones, Partners...) trabajando 24 horas al día para la empresa en que se instale. El sistema SAP R/3 es un sistema "On-line" y en tiempo real, diseñado para cubrir de forma global las necesidades de gestión o información de corporaciones de tipo medio/grande.

Consta de un conjunto de módulos totalmente integrados que cubren una amplia variedad de funciones de negocio entre las que se incluyen: Gestión Económico Financiera (Contabilidad General, Contabilidad Analítica, Activos Fijos, Módulo Financiero, etc.), Logística, Comercial y Distribución, Producción (Planificación, Control, Sistemas de Producción en serie, lotes, JIT, etc.), Control de Calidad, Mantenimiento, Gestión integrada de Proyectos, Recursos Humanos, Workflow, etc. En definitiva, puede afirmarse que cubre todas las áreas funcionales de la empresa.

SAP es muy flexible. Permite agilizar las tareas diarias de cualquier empresa independientemente del sector y del país en que trabaje, de su tamaño (si bien es cierto que parece estar dirigido más bien a grandes empresas) y de otros factores que pueden suponer un problema con otro software.

Las principales características de SAP son: Información "On-line" esta característica significa que la información se encuentra disponible al momento, sin necesidad de esperar largos procesos de actualización y procesamiento habituales en otros sistemas.

Jerarquía de la información, esta forma de organizar la información permite obtener informes desde diferentes vistas. Integración esta es la característica más destacable de SAP y significa que la información se comparte entre todos los módulos de SAP que la necesite y que pueden tener acceso a ella.

La información se comparte, tanto entre módulos, como entre todas las áreas.

La integración en SAP se logra a través de la puesta en común de la información de cada uno de los módulos y por la alimentación de una base de datos común. Por lo tanto, debemos tener en cuenta que toda la información que introducimos en SAP repercutirá, al momento, a todos los demás usuarios con acceso a la misma.

Este hecho implica que la información siempre debe estar actualizada, debe ser completa y debe ser correcta.

#### **2.1.5.1 ¿Que son los módulos SAP?**

el sistema SAP está compuesto de una serie de áreas funcionales o módulos, que responden de forma completa, y en tiempo real, a los procesos operativos de las compañías.

Aunque pueden ser agrupados en tres grandes áreas (financiera, logística y de recursos humanos), funcionan de un modo integrado, dado que existe una conexión natural entre los diferentes procesos.

En el área financiera, el módulo FI proporciona una visión completa de las funciones contables y financieras, e incluye un amplio sistema de información y de generación de reportes para facilitar la toma de decisiones.

El módulo CO (Controlling) se utiliza para representar la estructura de costos y los factores que tienen influencia, lo que genéricamente se conoce como contabilidad interna de las compañías.

Otros módulos son IM (Gestión de Inversiones) y TR (Tesorería).

En el área de logística, el módulo LO brinda herramientas e informes necesarios para analizar y gestionar el estado de la logística de la compañía y realizar previsiones en la cadena de suministros.

El módulo MM (Gestión de materiales) se integra completamente a las otras áreas funcionales de SAP y da soporte a todas las fases de gestión de materiales: planificación de necesidades y control, compras, entrada de mercaderías, gestión de stocks y verificación de facturas.

Business Warehouse (BW) o Business Intelligence (BI): Datawarehousing, Business Objects: Herramienta d BI para la creación de reportes, dashboards y gráficos, ventas y Distribución (SD): Facturación, Pedidos de Ventas, solicitudes, pedidos abiertos; etcétera.

### **2.1.5.2 Costos**

la implementación de SAP comprende los siguientes costos: costo del hardware, costo del software, costo de software no-SAP (sistema operativo, software de respaldo) y hardware adicional (equipo de lectura de código de barras).

Costo de implementación de procesos, costo de entrenamiento de los operadores, costo de mantenimiento del sistema, costo de oportunidad por fallas de sistema, costo de actualización periódica del sistema, costo de hardware: SAP R/3 requiere de un equipo de cómputo adecuado para poder funcionar, esto implica servidores poderosos y buenas estaciones de trabajo.

### **2.1.5.3 Costos de la implementación de los procesos.**

este costo consiste en: costo de ayuda experta, costo de suplir al personal de la empresa que se encuentra dedicado al 100% en el proyecto, costo de distraer personal de la empresa para trabajar con los expertos en sesiones esporádicas, capacitación del personal, pruebas del sistema, documentación, etc.

# CAPITULO III

## 3. Metodología

Dentro de este capítulo se darán a conocer cada uno de los pasos a seguir en la realización de este proyecto, describiendo los elementos que han servido de base para su elaboración.

Al inicio de este proyecto se realizó una inspección detallada al sistema “RGP” que se enfocó al análisis del código de los centros de trabajo en el sistema actual. Donde se destaca

- Conocimiento de los procedimientos realizados para calcular el tiempo estándar de cada uno de los centros de trabajo en operación dentro del sistema RGP.
- Conocimiento de cómo se generan las hojas de ruta.
- Conocimiento de la base de datos que utiliza el sistema RGP
- Conocimiento del herramental necesario para cada operación.

Una vez terminado el periodo de capacitación, se determinó que se podía proceder a dar inicio con las actividades de actualizaciones al código del sistema RGP.

### 3.1.1 Agregar Nuevos Registros en la Base de Datos.

Para cargar nuevos Herramentales a la base de datos se tiene que acceder al servidor “AGUAPSERV-OLD\SHR” desde SQL Server Enterprise Manager en el menú inicio (Ilustración 2).

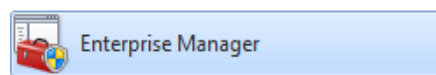


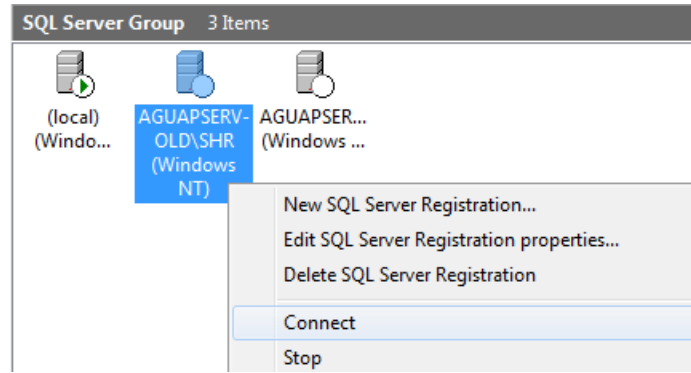
Ilustración 2 Icono de SQL en el menú de inicio.

Una vez iniciado el programa debes dirigirte al ítem SQL Server Group que se muestra en la pantalla Console Root\Microsoft SQL Servers y das doble clic (Ilustración 3).

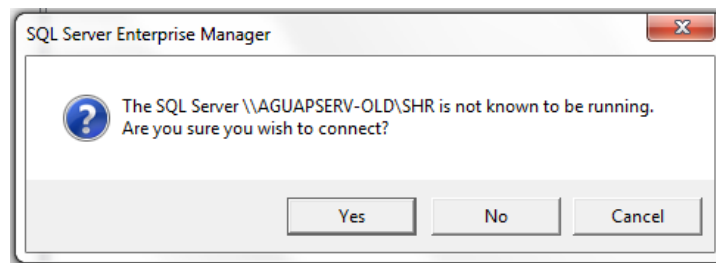


Ilustración 3 Pantalla inicial del programa

Te mostrara los servidores disponibles en la red local, das clic con el botón secundario y te desplegará un sub menú de acciones, ahora das clic en connect (Ilustración 4), te mostrará un cuadro de dialogo para confirmar si te quieres conectar; das clic en yes, si no hay ningún error listo ya estas conectado al servidor “AGUAPSERV-OLD\SHR” (Ilustración 5).

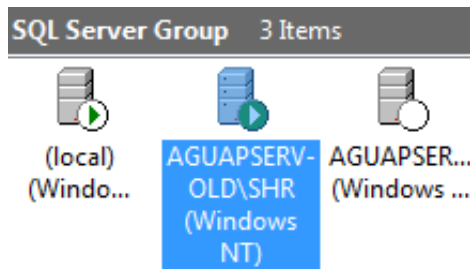


**Ilustración 4 Servidores Disponibles**

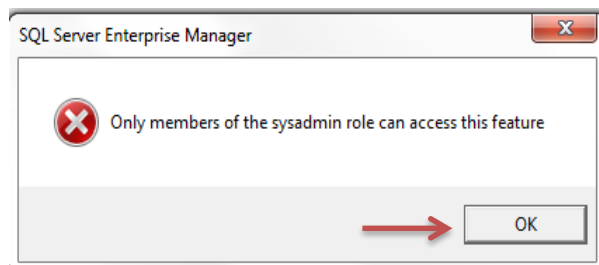


**Ilustración 5 Mensaje de Alerta de Conexión**

Debe aparecer una flechita blanca dentro de un círculo verde en el icono del servidor al que te conectaste, si es así, da doble clic sobre el icono de servidor **“AGUAPSERV-OLD\SHR”** (Ilustración 7), te mostrara un cuadro de dialogo (Ilustración 6) pidiendo confirmación para acceder das clic en OK y listo. Te mostrara el contenido del servidor (Ilustración 8).

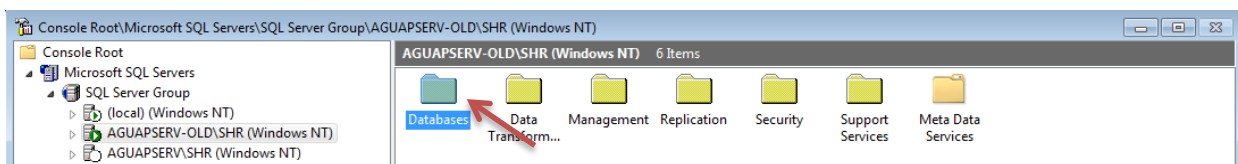


**Ilustración 7 Servidor que aloja la BD.**

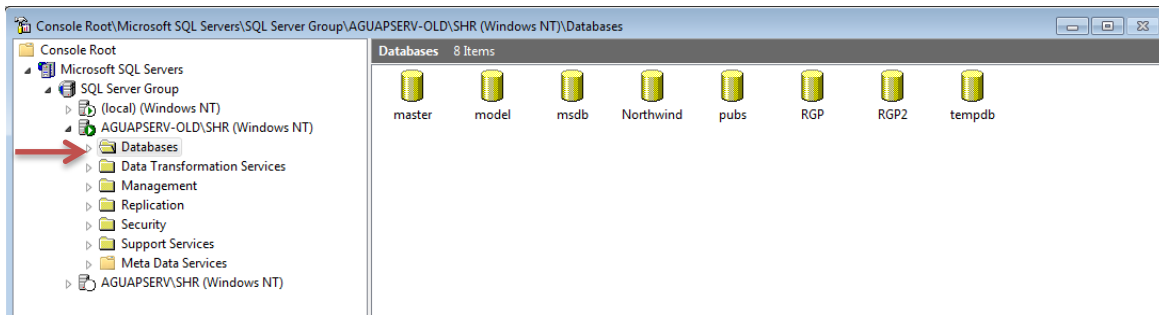


**Ilustración 6 Mensaje de Advertencia.**

Una vez dentro del servidor dar doble clic sobre la carpeta **“Databases”** (Ilustración 8). te mostrará las Bases de Datos incluidas en el servidor (Ilustración 9).

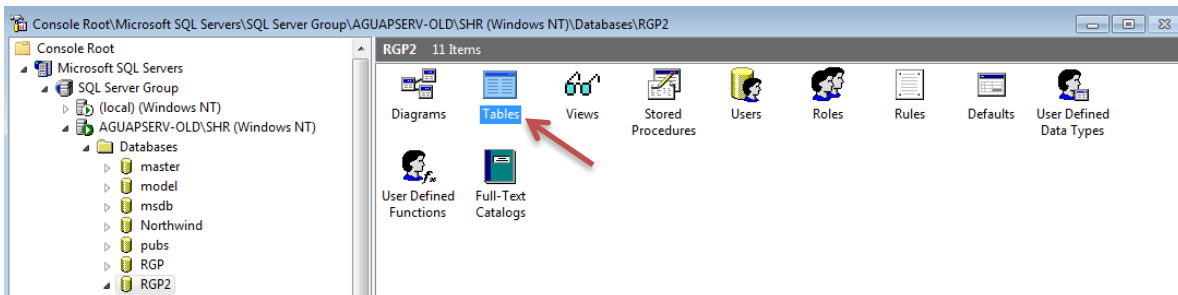


**Ilustración 8 Carpetas Incluidas en el Servidor**

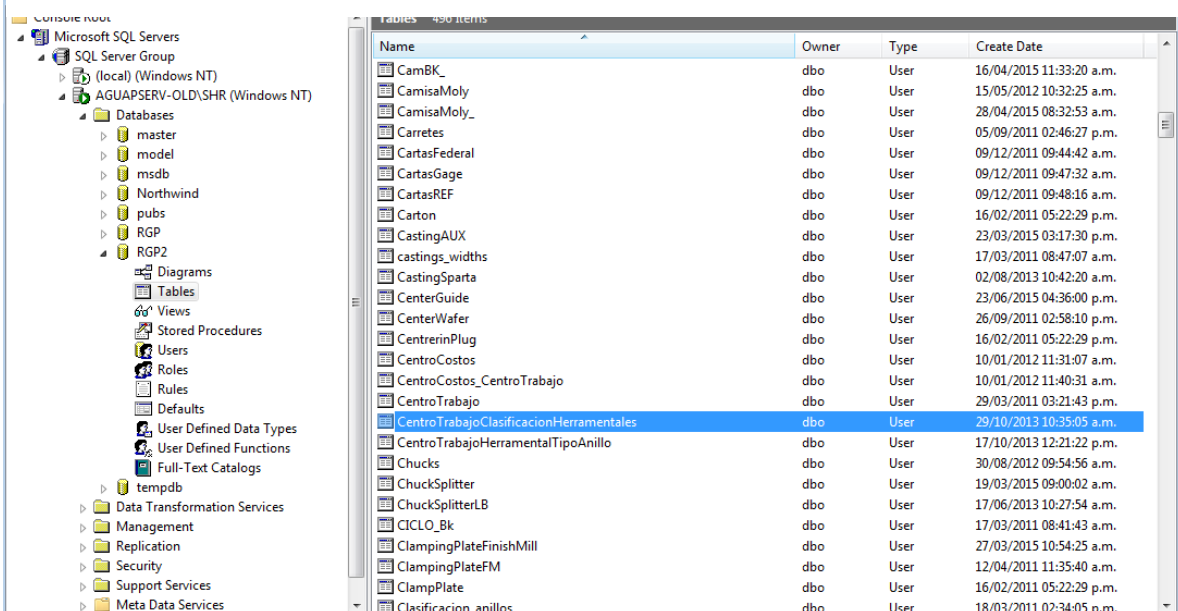


**Ilustración 9 Bases de Datos dentro del Servidor**

En el menú lateral selecciona la carpeta “**Databases**” que te despliega un sub-menú con las bases de datos dentro del servidor; Ahora si seleccionas la base de datos “**RGP2**” en el menú lateral y te mostrara el contenido de la base de datos, das doble clic sobre el ítem “**Tables**” (Ilustración 10), se desplegara un listado de tablas (Ilustración 10), después se selecciona la tabla a la que se desea agregar los nuevos registros en este caso es la tabla “**CentroTrabajoClasificacionHerramentales**” donde se va a cargar el nuevo herramental que ocupa el centro de trabajo deseado en este caso el “**421**”.(Ilustración 11)

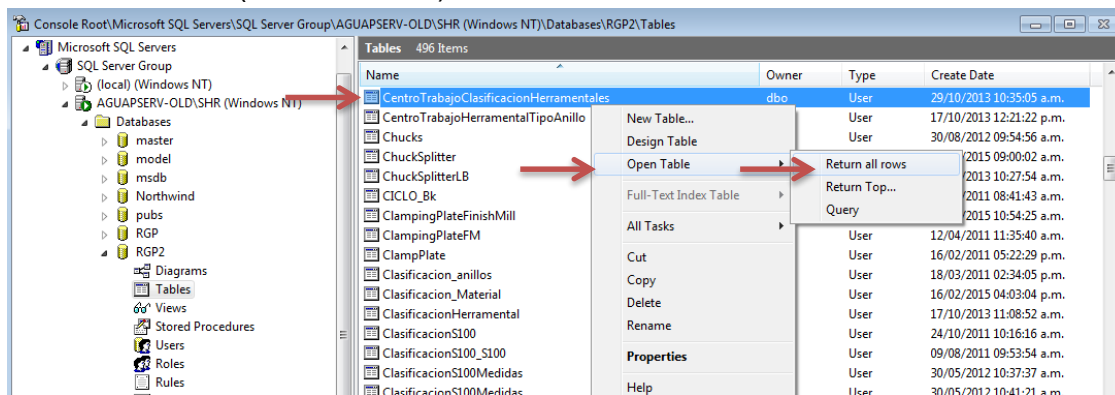


**Ilustración 11 Abrir tablas de la BD seleccionada.**



**Ilustración 10 Seleccionar una Tabla de la Base de Datos**

Una vez encontrada la tabla das clic con el botón secundario el cual desplegara un sub-menú donde darás clic con el botón secundario y seleccionarás “Open Table”, seguido de “Return all rows” (Ilustración 12).



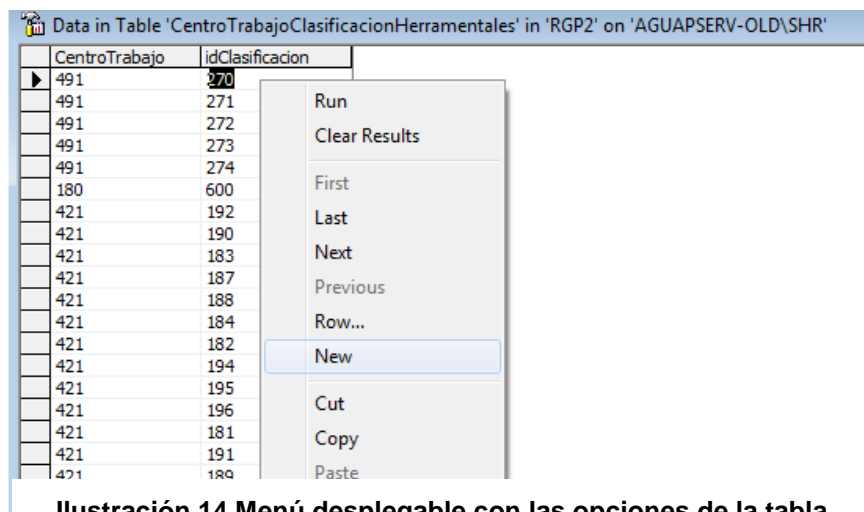
**Ilustración 12** Abir Tabla de SQL para ver Datos

Una vez realizado este procedimiento se abrirá una nueva ventana con la información incluida en tabla seleccionada (Ilustración 13).

CentroTrabajo	idClasificacion
491	270
491	271
491	272
491	273
491	274
180	600
421	192
421	190
421	183

**Ilustración 13** Tabla CentroTrabajoClasificacionHerramentales.

Para agregar un nuevo campo das clic con el botón secundario en cualquier registro dentro de la tabla y seleccionas “New” en el menú despegable (Ilustración 14), y listo aparecerá el puntero en la última fila de la tabla lista para agregar nuevos registros (Ilustración 15).



**Ilustración 14** Menú desplegable con las opciones de la tabla.



Data in Table 'CentroTrabajoClasificacionHerramentales' in 'RGP2' on 'AGUAPSERV-OLD\SHR'		
	CentroTrabajo	idClasificacion
	9363	138
	9363	139
	9363	140
	9363	141
	9363	142
	421	441

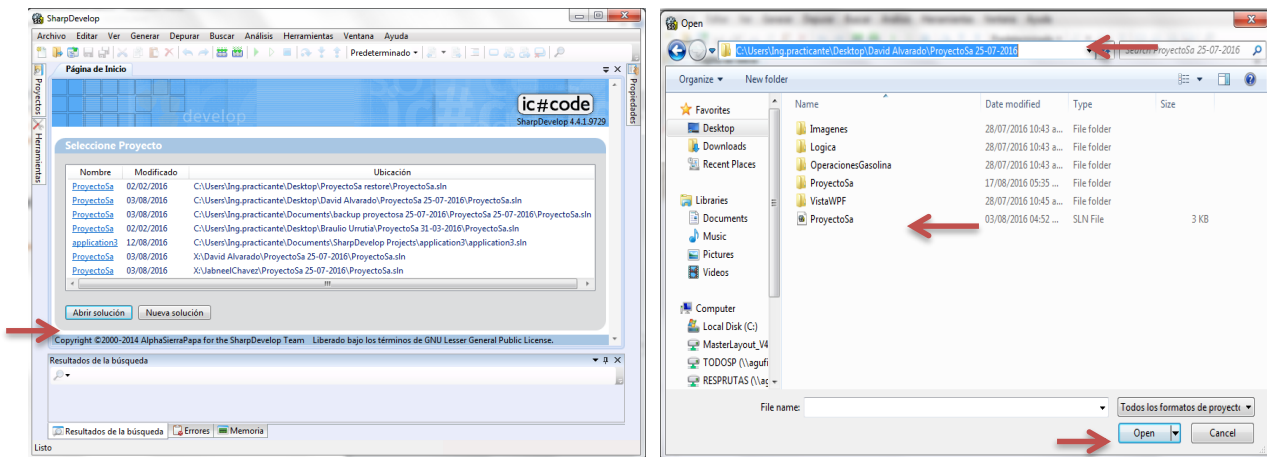
**Ilustración 15** Posicionamiento del cursor para agregar nuevos registros.

### 3.1.2 Acceder a Modificar, Actualizar o Agregar Nuevos Centros de Trabajo.

Nota: para acceder a modificar cualquier clase o archivo del proyecto debe dirigirse a un entorno de desarrollo, en nuestro caso es “SharpDevelop4.4” en el menú de inicio. Después seleccionar la opción “Abrir solución” que se muestra en la página de inicio (Ilustración 18), se abrirá una nueva ventana de búsqueda para que ingrese la ruta de la carpeta contenedora del sistema (Ilustración 17).



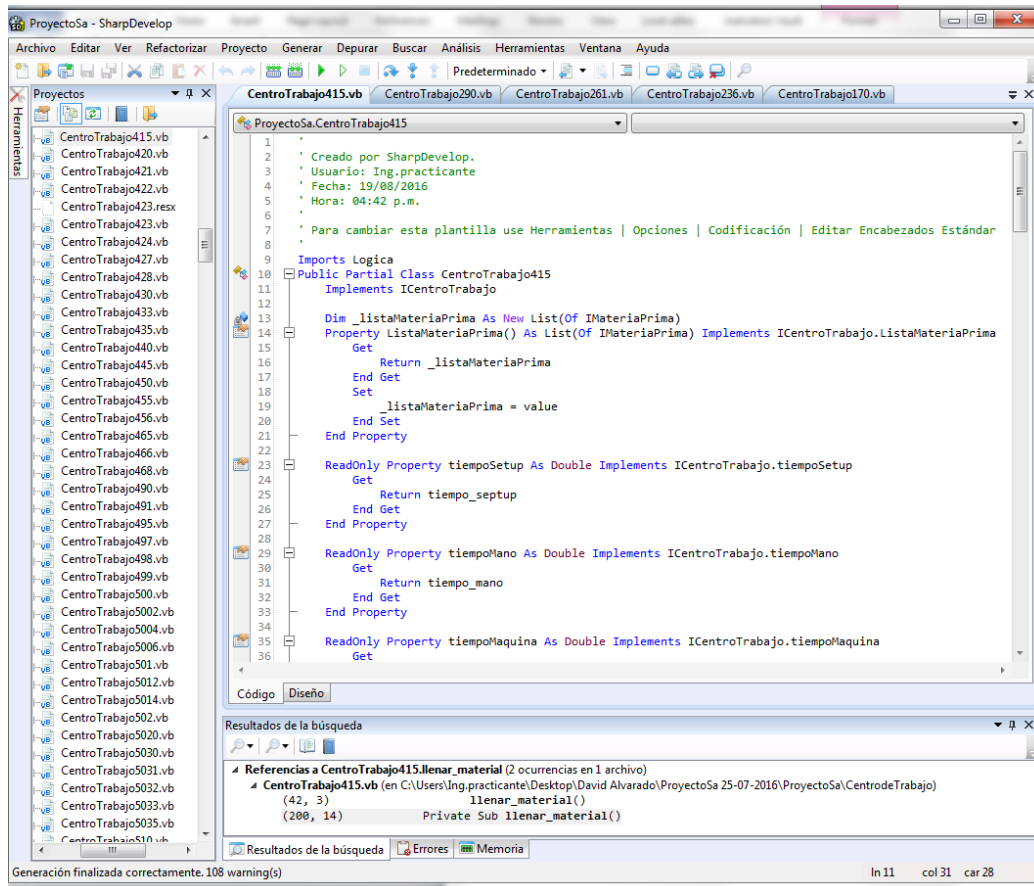
**Ilustración 16** Icono de SharpDevelop4.4 en el menú de inicio.



**Ilustración 18** Página de inicio SharpDevelop.

**Ilustración 17** Ventana de Búsqueda.

Una vez en la carpeta contenedora seleccionas el archivo .SLN y das clic en open y listo te mostrara todos los archivos incluidos en el proyecto, buscas el que quieres editar, das doble clic sobre él y listo.



**Ilustración 19 Proyecto abierto y listo para su edición u actualización.**

### 3.1.3 Pattern2.vb – Placas Modelo.

La actualización realizada en este formulario se llevó a cabo ya que nació la necesidad de exportar la tabla de las placas modelo para trabajar en un archivo de Excel. Para esto se agregó un nuevo botón para realizar la exportación de la tabla.

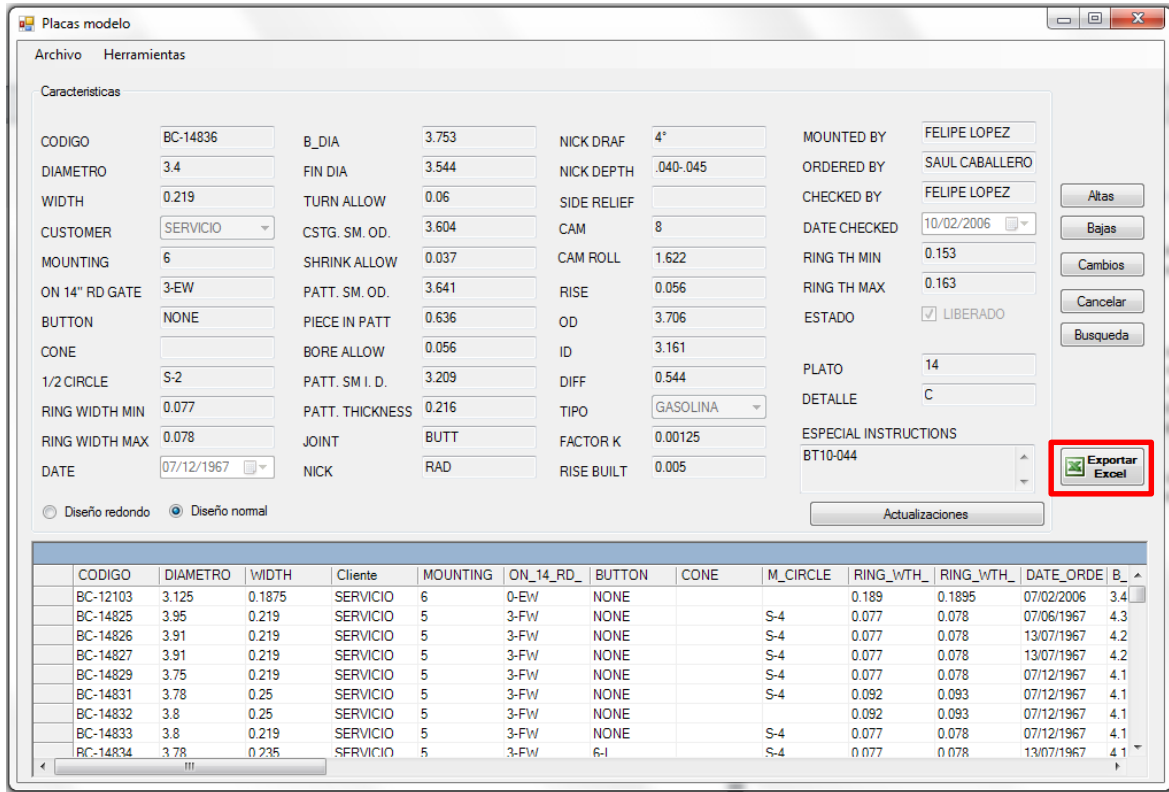


Ilustración 20 Interface gráfica (form).- Placas Modelo.

Aquí también se agregó la función del botón que manda llamar el método creado en la clase “ExportarPlacasModelo” declarando una nueva variable para extraer el método.

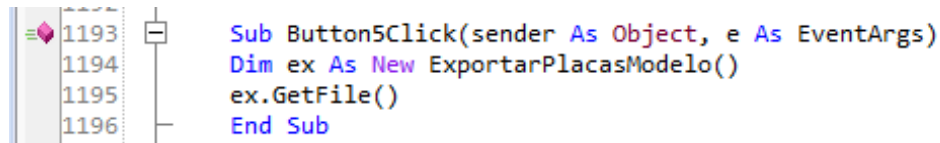


Ilustración 21 Función del botón Exportar Excel

### 3.1.4 ExportarPlacasModelo.vb

Para poder llevar a cabo la exportación de la tabla se tuvo que crear una nueva clase llamada ExportarPlacasModelo donde se importaron las utilerías del office necesarias para realizar la importación de la tabla, después se declaró el método GetFile() donde se realizara la exportación de los datos almacenados en la tabla, dentro del mismo método se declararon cuatro nuevas variables que son para enviar los datos al archivo de Microsoft Excel; después se realiza un query para extraer los datos de la tabla y los ordene según el Código.

```

9 Imports Spring.Context
10 Imports Spring.Context.Support
11 Imports Logica
12 Public Class ExportarPlacasModelo
13
14 Public Sub GetFile()
15     Dim exApp As New Microsoft.Office.Interop.Excel.Application
16     Dim exLibro As Microsoft.Office.Interop.Excel.Workbook
17     Dim exHoja As Microsoft.Office.Interop.Excel.Worksheet
18     Dim adldCI As System.Globalization.CultureInfo = System.Threading.Thread.CurrentThread.CurrentCulture
19     System.Threading.Thread.CurrentThread.CurrentCulture = New System.Globalization.CultureInfo("en-US")
20     exLibro = exApp.Workbooks.Add
21     exHoja = exLibro.Worksheets("Sheet1")
22
23     Dim cmd As SqlCommand
24     Dim lector As SqlDataReader
25     Dim query As String
26     query = "Select * " & _
27           "From Pattern2 " & _
28           "Order By codigo"
29     cmd = conecta(query)

```

**Ilustración 22 método GetFile( ), variables necesarias y query realizado a la BD.- ExportarPlacasModelo**

(1) se declaran los encabezados que creara en el archivo Excel para la tabla, indicando en que columna va el mismo (Ilustración 23 líneas de la 32 a la 78, en la línea 79 se declara el color de fondo del rango de celdas que ocuparan los títulos).

```

31 lector = cmd.ExecuteReader
32 exHoja.Cells.Item(1,1)= "CODIGO"
33 exHoja.Cells.Item(1,2)= "DIAMETRO"
34 exHoja.Cells.Item(1,3)= "WIDTH"
35 exHoja.Cells.Item(1,4)= "Cliente"
36 exHoja.Cells.Item(1,5)= "MOUTING"
37 exHoja.Cells.Item(1,6)= "ON 14 RD_GATE"
38 exHoja.Cells.Item(1,7)= "MONTAJE"
39 exHoja.Cells.Item(1,8)= "MONTAJE"
40 exHoja.Cells.Item(1,9)= "MONTAJE"
41 exHoja.Cells.Item(1,10)= "MONTAJE"
42 exHoja.Cells.Item(1,11)= "MONTAJE"
43 exHoja.Cells.Item(1,12)= "MONTAJE"
44 exHoja.Cells.Item(1,13)= "MONTAJE"
45 exHoja.Cells.Item(1,14)= "MONTAJE"
46 exHoja.Cells.Item(1,15)= "MONTAJE"
47 exHoja.Cells.Item(1,16)= "MONTAJE"
48 exHoja.Cells.Item(1,17)= "MONTAJE"
49 exHoja.Cells.Item(1,18)= "MONTAJE"
50 exHoja.Cells.Item(1,19)= "MONTAJE"
51 exHoja.Cells.Item(1,20)= "MONTAJE"
52 exHoja.Cells.Item(1,21)= "MONTAJE"
53 exHoja.Cells.Item(1,22)= "MONTAJE"
54 exHoja.Cells.Item(1,23)= "MONTAJE"
55 exHoja.Cells.Item(1,24)= "MONTAJE"
56 exHoja.Cells.Item(1,25)= "MONTAJE"
57 exHoja.Cells.Item(1,26)= "MONTAJE"
58 exHoja.Cells.Item(1,27)= "MONTAJE"
59 exHoja.Cells.Item(1,28)= "MONTAJE"
60 exHoja.Cells.Item(1,29)= "MONTAJE"
61 exHoja.Cells.Item(1,30)= "MONTAJE"
62 exHoja.Cells.Item(1,31)= "MONTAJE"
63 exHoja.Cells.Item(1,32)= "MONTAJE"
64 exHoja.Cells.Item(1,33)= "MONTAJE"
65 exHoja.Cells.Item(1,34)= "MONTAJE"
66 exHoja.Cells.Item(1,35)= "MONTAJE"
67 exHoja.Cells.Item(1,36)= "MONTAJE"
68 exHoja.Cells.Item(1,37)= "MONTAJE"
69 exHoja.Cells.Item(1,38)= "MONTAJE"
70 exHoja.Cells.Item(1,39)= "MONTAJE"
71 exHoja.Cells.Item(1,40)= "MONTAJE"
72 exHoja.Cells.Item(1,41)= "MONTAJE"
73 exHoja.Cells.Item(1,42)= "MONTAJE"
74 exHoja.Cells.Item(1,43)= "MONTAJE"
75 exHoja.Cells.Item(1,44)= "MONTAJE"
76 exHoja.Cells.Item(1,45)= "MONTAJE"
77 exHoja.Cells.Item(1,46)= "MONTAJE"
78 exHoja.Cells.Item(1,47)= "Diseno"
79 exHoja.Range("A1:AU1").Interior.ColorIndex= 10
80

```

**Ilustración 23 Encabezados de la nueva Hoja de Excel.- ExportarPlacasModelo.**

(2) Se declaran los lectores que insertaran en la celda correspondiente la información de la tabla (Ilustración 24).

```

81 Dim f As Integer = 2
82 While lector.Read
83     exHoja.Cells.Item(f,1) = Lector(0).ToString
84     exHoja.Cells.Item(f,2) = Lector(1).ToString
85     exHoja.Cells.Item(f,3) = Lector(2).ToString
86     exHoja.Cells.Item(f,4) = Lector(3).ToString
87     exHoja.Cells.Item(f,5) = Lector(4).ToString
88     exHoja.Cells.Item(f,6) = Lector(5).ToString
89     exHoja.Cells.Item(f,7) = Lector(6).ToString
90     exHoja.Cells.Item(f,8) = Lector(7).ToString
91     exHoja.Cells.Item(f,9) = Lector(8).ToString

```

**Ilustración 24 lectores para insertar los datos en la hoja Excel.- ExportarPlacasModelo.**

(3) para darle más formato al archivo Excel se declara el color de las celdas de los títulos, también se le da formato en negritas, centrado, que las celdas sean autoajustables. (Ilustración 25 líneas de la 32 a la 78).

```

128         exHoja.Cells.Item(f,46) = Lector(45).ToString
129         exHoja.Cells.Item(f,47) = Lector(46).ToString
130         f += 1
131         exHoja.Rows.Item(1).Font.Bold = 1
132         exHoja.Rows.Item(1).HorizontalAlignment = 3
133         exHoja.Columns.AutoFit()
134         exHoja.Columns.HorizontalAlignment = 3
135         exHoja.Columns.VerticalAlignment = 2
136     End While
137     cn.Close()
138 Catch er As Exception
139     MsgBox("No se pudo Exportar el Archivo a Excel" & vbCrLf & er.Message,MsgBoxStyle.Critical,"Ing de Rutas")
140 End Try
141 exApp.Application.Visible = True
142 exHoja = Nothing
143 exLibro = Nothing
144 exApp = Nothing
145 End Sub
146 End Class

```

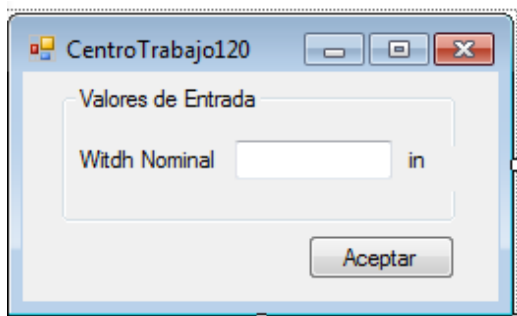
**Ilustración 25 formato de la nueva hoja Excel.- ExportarPlacasModelo.**

También se programó una excepción que mandara un mensaje de alerta en caso de error, notificando al usuario que el archivo no pudo ser exportado (Ilustración 25 línea 139).

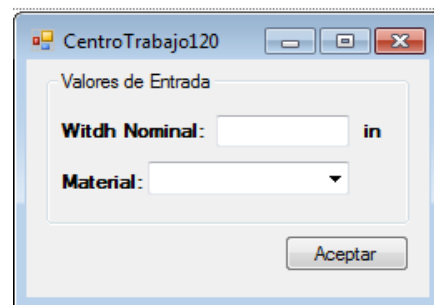
### 3.1.5 Anillos

#### 3.1.5.1 CentroTrabajo120.vb – HEAT TREAT PREM. GASOLINE

Este centro de trabajo se actualizo modificando la interfaz gráfica ya que en un principio no contaba con los elementos necesario, se tuvo que agregar un nuevo label (Material:) y un ComboBox, para los valores de entrada que se necesitan en la nueva fórmula a utilizar en el centro de trabajo.



**Ilustración 27 form antes de actualización.  
- CT-120.**



**Ilustración 26 form actualizado. - CT-120.**

La nueva fórmula utilizada para sacar el tiempo maquina requiere del Width Nominal del anillo y del tiempo ciclo de calentamiento en horno por tipo de material a utilizar, por eso fue necesario agregar los campos anteriormente descritos para saber el tipo de material a utilizar, y así poder extraer el tiempo ciclo de la operación declarado en la variable t\_ciclo de cada material; para esta actualización se agregó el método “llenar\_material()” que es el encargado de crear un listado de los materiales disponibles para la operación y mostrarlos en el comboBox declarado anteriormente, al cual le cambiamos el nombre por “cbo\_material”.

```

227 Private Sub llenar_material()
228     Dim l As New List(Of String)
229     l.Add("HIERRO GRIS INTERMEDIO")
230     l.Add("HIERRO GRIS ALTO MODULO")
231     l.Add("HIERRO GRIS")
232     l.Add("HIERRO GRIS CENTRIFUGADO")
233     l.Add("HIERRO DUCTIL")
234     l.Add("ACERO INOXIDABLE")
235     l.Add("ACERO AL CARBON")
236     For Each m In obtenerEspecTsTipoMaterial(l)
237         cbo_material.Items.Add(m)
238     Next
239 End Sub
240

```

**Ilustración 28 método llenar\_material( ). CT-120.**

Se crearon nuevas variables llamadas “\_material”, “tipo” y “banElementoFijo” para almacenar los datos requeridos por la operación (Ilustración 29).

```

Dim tipo As String
Dim banElementoFijo As Boolean = False

Public Sub New (ByVal p_width As Double)
    width = p_width
End Sub

'-----variables que se nesesitan-----
Dim width As Double
Dim _width As Double
dim _material as string

```

**Ilustración 29 Nuevas Variables. CT-120.**

En la función Test se agregó la variable material que obtiene los valores ingresados por el usuario del tipo de material del listado de materiales, y la variable tipo que busca el tipo de material específico al que se refiere la operación; también se declaró un IF para validar cual tipo de material fue seleccionado por el usuario y mandar un mensaje de alerta en caso de que el material seleccionado no esté disponible para trabajar en este centro de trabajo (Ilustración 30).

```

95 Public Function Test(ByVal IDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
96     _material = Module1.obtenerValorDatoString (lDatosString, "MATERIAL")
97     tipo = module1.sacar_tipo_material(_material)
98
99     If tipo = "HIERRO GRIS ALTO MODULO" Or tipo = "HIERRO GRIS INTERMEDIO" Or tipo = "ACERO" Or tipo = "HIERRO DUCTIL" Or tipo = "HIERRO GRIS" Then' Or
100
101     Else
102         msgbox("Material " & tipo & " No Disponible para el C.T " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
103             "Por Favor seleccione un tipo de material disponible para poder calcular el centro de trabajo",MsgBoxStyle.Critical,"Ing. de Rutas")
104         Dim listaOpcionales As New List(Of String)
105         listaOpcionales.Add("HIERRO GRIS ALTO MODULO")
106         listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
107         listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
108         listaOpcionales.Add("HIERRO GRIS")
109         'listaOpcionales.Add("HIERRO DUCTIL")
110         'listaOpcionales.Add("ACERO INOXIDABLE")
111         'listaOpcionales.Add("ACERO AL CARBON")
112         Dim f As New Seleccionar(listaOpcionales)
113         If f.ShowDialog() = DialogResult.Ok Then
114             tipo = f.elementoSeleccionado
115             banElementoFijo = True
116         End If
117         Exit Function
118     End If
119     Return True
120 End Function

```

**Ilustración 30 función Test( ). CT-120.**

En la función calcular() se actualizo en código agregando la función “buscar\_septup()” que se encarga de buscar el tiempo Setup del centro de trabajo en la base de datos; se creó la variable “t\_ciclo” de tipo Double; se declaró un IF para condicionar la variable “banElementoFijo” que es de tipo boolean a que sea igual False entonces mostrará el listado de material, en caso de que la variable “tipo” sea igual a “HIERRO GRIS ALTO MOD” tomara el siguiente valor para la variable “t\_ciclo = 7166.03”, en caso de que “tipo” sea igual a “HIERRO GRIS INTERMEDIO” la variable “t\_ciclo = 4500”, si no se seleccionó ninguno de los anteriores mostrara un mensaje de advertencia y cerrara la función saliendo del form; en caso contrario de seleccionar alguno de los anteriores procederá a realizar los cálculos para obtener el tiempo máquina(Ilustración 31).

```

166 Public Sub calcular() implements ICentroTrabajo.calcular
167
168     buscar_septup()
169     Dim t_ciclo As Double
170
171     If banElementoFijo = False Then
172         tipo = Module1.sacar_tipo_material(_material)
173     End If
174     If tipo = "HIERRO GRIS ALTO MODULO"
175         t_ciclo = 7166.03
176     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
177         t_ciclo = 4500
178     Else
179         msgbox("Material " & tipo & " no disponible para el C.T 130, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
180         Exit Sub
181     End If
182
183     tiempo_septup = buscar_setupin(centro)
184     tiempo_maquina = math.Round(((width*(t_ciclo+18669.41))/20304)*100,3)
185     tiempo_mano=tiempo_maquina
186 End Sub

```

**Ilustración 31 método calcular( ). - CT -120.**

### 3.1.5.2 CentroTrabajo130.vb – SPLITTER CASTINGS.

Para este centro de trabajo se realizaron las siguientes actualizaciones de la interfaz gráfica de tal forma que se agregaron tres nuevos labels uno para la etiqueta de “Diámetro Nominal:”, otro par la etiqueta “Espaciador1:” y por ultimo otro para la etiqueta “Espaciador 2:”; se agregó un TextBox para capturar el diámetro ingresado por el usuario y dos nuevos comboBox para mostrar y que elija los valores deseados el usuario; estos valores son necesarios para realizar el cálculo del tiempo máquina.

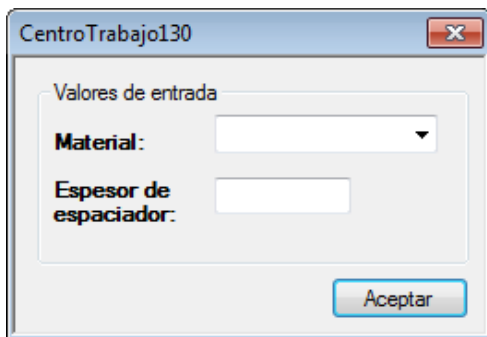


Ilustración 32 from antes de la actualización. - CT -130.

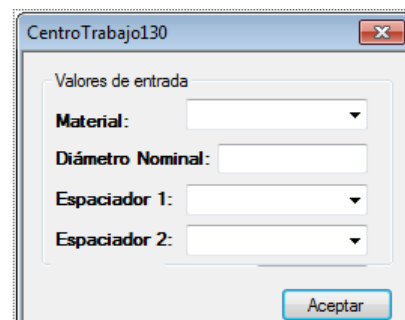


Ilustración 33 form actualizado. - CT -130.

Para la actualización de la formula se tuvo que modificar parte del código, agregando nuevas variables llamadas “\_dmtro”, “\_esp1”, “\_esp2”, “dia\_tipoMat” y “anillosxCelula” que son de tipo Double, que se utilizaran para realizar el cálculo del tiempo\_maquina; para ello se modificaron los valores de la variable “t\_ciclo” de cada tipo de material que puede ser utilizado en el centro de trabajo, de los que ya tenían declarados anteriormente por los valores de la siguiente tabla según sea el caso del tipo de material.

Tabla 1 Tiempos Ciclo por Tipo de Material.- CT- 130.

TIPO DE MATERIAL	Tiempo ciclo	Diámetro por tipo de Material
HIERRO DUCTIL	N/A	N/A
HIERRO GRIS	64.94	3.77990
HIERRO GRIS ALTO MOD	82.48	3.74020
HIERRO GRIS INTERMEDIO	101.66	3.77950



```

If banElementoFijo = False Then
    tipo = Module1.sacar_tipo_material(_material)
End If
If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
    t_ciclo = 64.94
    dia_tipoMat = 3.77990
ElseIf tipo = "HIERRO GRIS ALTO MODULO"
    t_ciclo = 82.48
    dia_tipoMat = 3.74020
ElseIf tipo = "HIERRO GRIS INTERMEDIO"
    t_ciclo = 101.66
    dia_tipoMat = 3.77950

```

**Ilustración 34 Actualización de los valores de la variable "t\_ciclo". - CT-130.**

Se agregó una condición IF para validar dentro de que rango se encuentran los valores de los espaciadores ingresados por el usuario y así obtener el valor de los anillos que se utilizan por célula para ello se utilizaron los registros de la siguiente tabla.

Base de datos total de anillos por célula				
Tipo	Espaciador 1	Espaciador 2	castings	anillos
CUADRUPLE	0.0648	0.1296	5	20
CUADRUPLE	0.1451	0.0726	5	20
CUADRUPLE				
DOBLES	0.1451		10	20
DOBLES	0.1776		9	18
DOBLES	0.2086		8	16
DOBLES	0.2686		6	12
DOBLES	0.2946		6	12
DOBLES	0.3316		5	10
DOBLES	0.1362		11	22
DOBLES	0.3855		4	8
DOBLES	0.1526		10	20

**Tabla 2 valores de espaciadores y anillos por célula.**

```

If _esp1 = 0.0648 and _esp2 = 0.1296 Then
    anillosxCelula = 20
    MsgBox("Seleccionaste un tipo de anillos por celula 'CUADRUPLE'")
ElseIf _esp1 = 0.1451 and _esp2 = 0.0726 Then
    anillosxCelula = 20
    MsgBox("Seleccionaste un tipo de anillos por celula 'CUADRUPLE'")
ElseIf _esp1 = 0.1451 And _esp2 = 0 Then
    anillosxCelula = 20
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.1776 And _esp2 = 0 Then
    anillosxCelula = 18
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.2086 And _esp2 = 0 Then
    anillosxCelula = 16
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.2686 And _esp2 = 0 Then
    anillosxCelula = 12
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.2946 And _esp2 = 0 Then
    anillosxCelula = 12
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.3316 And _esp2 = 0 Then
    anillosxCelula = 10
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.1362 And _esp2 = 0 Then
    anillosxCelula = 22
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.3855 And _esp2 = 0 Then
    anillosxCelula = 8
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
ElseIf _esp1 = 0.1526 And _esp2 = 0 Then
    anillosxCelula = 20
    MsgBox("Seleccionaste un tipo de anillos por celula 'DOBLE'")
Else
    MsgBox ("Imposible Calcular Verifica las Medidas de los Espaciadores",MsgBoxStyle.Critical,"Ing. de Rutas")
    Exit Sub
End If

```

**Ilustración 35 Condición IF para obtener el valor de anillosxCelula. - CT-130.**

Después de realizar la condición IF ahora si se procede a utilizar la nueva fórmula en la variable tiempo\_maquina que quedo de la siguiente manera (Ilustración 36).

```
tiempo_maquina =((((_dmTRO*t_ciclo)/dia_tipoMat)+62.40)/(36*anillosxCelula))*100
tiempo_mano = math.Round(tiempo_maquina * factorLabor,3)
```

**Ilustración 36 Variable „tiempo\_maquina“ formula actualizada.-CT-130.**

En la función ButtonClick se agregaron a la condición IF con la que ya contaba los dos comboBox que se agregaron anteriormente condicionándolos a que los valores ingresados por el usuario tienen que ser valores numéricos, si es así se procede a realizar los cálculos matemáticos correspondientes, no sin antes haber verificado la comprobación de que los valores ingresados en los espaciadores por el usuario tienen que ser iguales a los del siguiente esquema, en caso de no cumplir con alguna de las condiciones mostrará un mensaje de advertencia para verificar los valores.

Base de datos total de anillos por célula				
Tipo	Espaciador 1	Espaciador 2	castings	anillos
CUADRUPLE	0.0648	0.1296	5	20
CUADRUPLE	0.1451	0.0726	5	20
CUADRUPLE				
DOBLES	0.1451		10	20
DOBLES	0.1776		9	18
DOBLES	0.2086		8	16
DOBLES	0.2686		6	12
DOBLES	0.2946		6	12
DOBLES	0.3316		5	10
DOBLES	0.1362		11	22
DOBLES	0.3855		4	8
DOBLES	0.1526		10	20

**Tabla 3 Esquema de valores necesarios Espaciadores.- CT- 130.**

```
Sub Button1Click(sender As Object, e As EventArgs)
    If IsNumeric(Me.textBox2.Text) And IsNumeric(Me.comboBox1.Text) And IsNumeric(Me.comboBox2.Text) Then
        Me.material = Me.cb_material.Text
        Me_espesor = Me.textBox1.Text
        _esp1 = Me.comboBox1.Text
        _esp2 = comboBox2.Text
        me_dmTRO = Me.textBox2.Text
        calcular()
        Close()

        ElseIf _esp1 = 0.0648 and _esp2 = 0.1296
        ElseIf _esp1 = 0.1451 and _esp2 = 0.0726
        ElseIf _esp1 = 0.1451 And _esp2 = 0
        ElseIf _esp1 = 0.1776 And _esp2 = 0
        ElseIf _esp1 = 0.2086 And _esp2 = 0
        ElseIf _esp1 = 0.2686 And _esp2 = 0
        ElseIf _esp1 = 0.2946 And _esp2 = 0
        ElseIf _esp1 = 0.3316 And _esp2 = 0
        ElseIf _esp1 = 0.1362 And _esp2 = 0
        ElseIf _esp1 = 0.3855 And _esp2 = 0
        ElseIf _esp1 = 0.1526 And _esp2 = 0
        Else
            msgbox("Imposible Calcular Verifica que los Valores son Correctos",MsgBoxStyle.Exclamation,"Ing. de Rutas")
        End If
End Sub
```

**Ilustración 37 función click( ) botón aceptar.-CT -130.**

### 3.1.5.3 CentroTrabajo170.vb – FINISH GRIND (MICRO).

El centro de trabajo 170 se actualizo en su interfaz gráfica ya que en el form que ya tenían diseñado tenía la opción de tomar tres valores de entrada, y en la nueva fórmula solo se necesita el tipo de material a utilizar, así que decidió ocultar los labels y los TextBox que no se necesitaban para ello se declaró el evento CentroTrabajo170Load (sender...). En el cual se declara cuáles son los elementos a ocultar, y se le asigna una nueva posición a los que se van a utilizar (Ilustración 40).

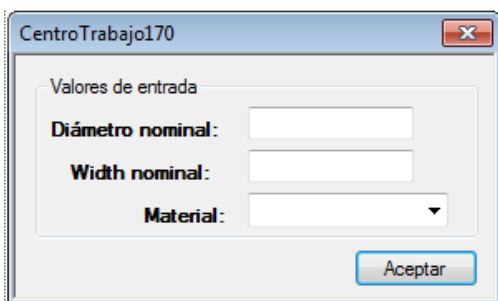


Ilustración 38 form antes de la actualización.- CT-170.

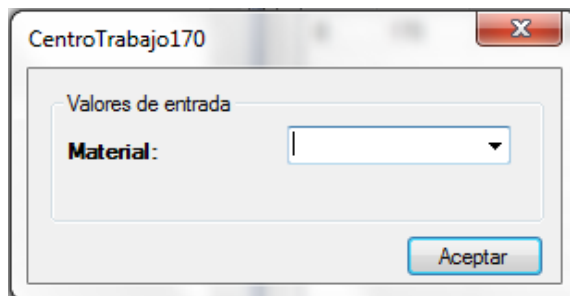


Ilustración 39 form actualizado.-CT -170.

```
Sub CentroTrabajo170Load(sender As Object, e As EventArgs)
    label2.Visible = False
    textBox2.Visible = False
    label1.Visible = False
    textBox1.Visible = False |
    label3.Location = New System.Drawing.Point(6, 26)
    cbo_material.Location = New System.Drawing.Point (128,19)
End Sub
End Class
```

Ilustración 40 Evento Load.- CT-170.

En el método calcular(), se actualizo el código de manera que se eliminaron del condicionamiento los materiales que no se utilizaban en el centro y se actualizo el tiempo ciclo de los siguientes materiales: "ACERO INOXIDABLE" y "ACERO AL CARBON" t\_ciclo=0.04636; y del "HIERRO GRIS" y "HIERRO GRIS CENTRIFUGADO" t\_ciclo= 0.2840.

La fórmula se modificó quedando de la siguiente manera:  $((0.06+t\_ciclo)/36)*100$ ; nota la fórmula que se ve en color verde en la ilustración 41 es la fórmula obsoleta utilizada anteriormente en el centro de trabajo170.

```

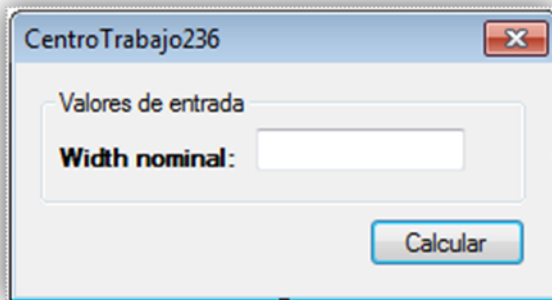
Public Sub calcular() implements ICentroTrabajo.calcular
    tiempo_septup=buscar_setup(centro)
    Dim t_ciclo As Double=0
    If banElementoFijo = False Then
        tipo = Module1.sacar_tipo_material(_material)
    End If
    If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Then
        t_ciclo = .2840
    ElseIf tipo = "ACERO INOXIDABLE" or tipo = "ACERO AL CARBON"
        t_ciclo = .4636
    Else
        msgbox("Material " & tipo & " no disponible para el C.C 170, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
        Exit Sub
    End If
    tiempo_maquina = ((0.06+t_ciclo)/36)*100
    '((( ( 0.3181 + ( ( t_ciclo * _d1 ) / 3.9173 ) * ( 4.5753/_h1 ) ) ) * _h1 ) / 164.7108)*100,3)
    tiempo_mano = tiempo_maquina
End Sub

```

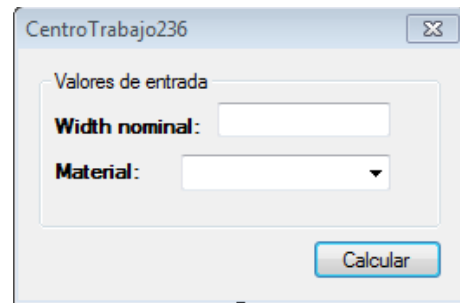
**Ilustración 41 método calcular(), y tiempos ciclo actualizados.- CT-170.**

### 3.1.5.4 CentroTrabajo236.vb – MACHINE CENTER KOMATSU

Este centro de trabajo se actualizo modificando la interfaz ya que un principio la interfaz de este centro de trabajo solo requería de un valor de entrada que era el Width nominal; para su actualización fue necesario incorporar un nuevo label llamado (Material) y un comboBox para pedir al usuario ingrese el material a utilizar en la operación.



**Ilustración 42 form antes de la actualización. - CT-236.**



**Ilustración 43 form actualizado. - CT-236.**

Para la actualización fue necesario modificar el código incorporando nuevas variables llamadas “tipo”, “banElementoFijo” y “\_material”. En la función Test se agregó la variable “\_material” que obtiene los valores del tipo de material del listado de materiales, y la variable “tipo” que saca el tipo de material específico al que se refiere; se declaró un IF que condiciona el tipo de material a utilizar ya que dependiendo de si este tiene seleccionado como respuesta un material de la lista, este se encarga de buscar que tipo de material que se está usando y en caso de seleccionar un material no disponible para el centro de trabajo este mandaría un mensaje de advertencia (Ilustración 44).

También fue necesario agregar la función llenar\_material() que es la encargada de cargar el listado de materiales disponibles en la planta para mostrarlos en el ComboBox declarado anteriormente, al cual le cambiamos el nombre por: “cbo\_material”.

```
Public Function Test(ByVal lDatos As List(Of Data),ByVal lDatosBool As List(Of DataBoolean),ByVal lDatosString As List(Of DataString)) As Boolean Implements ICentroTrabajo.Test
    _material = module1.obtenerValorDataString(lDatosString,"MATERIAL")
    Tipo = module1.sacar_tipo_material(_material)

    If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO GRIS INTERMEDIO" Or tipo = "HIERRO DUCTIL" Or tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Then
    Else
        MsgBox("Material " & tipo & " no disponible para el C.C " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
            "Por favor seleccione un tipo de material disponible para poder calcular el centro de Trabajo",MsgBoxStyle.Critical,"Ing. de Rutas")
        Dim listaOpcionales As New List(Of String)
        listaOpcionales.Add("HIERRO GRIS")
        listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
        listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
        listaOpcionales.Add("HIERRO GRIS ALTO MODULO")
        listaOpcionales.Add("HIERRO DUCTIL")
        listaOpcionales.Add("ACERO INOXIDABLE")
        listaOpcionales.Add("ACERO AL CARBON")
        Dim f As New seleccion(listaOpcionales)
        If f.ShowDialog() = DialogResult.OK Then
            tipo = f.elementoSeleccionado
            banElementoFijo = True
        End If
    End If
    Exit Function
End If
Return True
End Function
```

**Ilustración 44 función Test().- CT-236.**

la función calcular() se modificó actualizando el tiempo ciclo de los hierros a utilizar, se agregó un if para condicionar la selección de un tipo de material disponible para el Centro de Trabajo236, y se agregó la variable t\_ciclo del tipo double donde se almacenaran los tiempos ciclo de cada uno de los materiales disponibles para este centro de trabajo; en caso de no seleccionar ninguno de los materiales disponibles para la operación mostrará un mensaje de alerta y termina con la función que está realizando cerrando el form. En caso contrario solo pasaría a realizar la operación matemática (Ilustración 45).

```
Dim _h1 As Double
Dim _material As String
-----
Dim centro As String = "236"
Public Property CentroTrabajo As String Implements ICentroTrabajo.CentroTrabajo
    Get
        Return centro
    End Get
    Set
        centro = value
    End Set
End Property

public Sub calcular() implements ICentroTrabajo.calcular
    tiempo_septup = buscar_setupin(centro)
    Dim t_ciclo As Double
    If banElementoFijo = False Then
        tipo = Module1.sacar_tipo_material(_material)
    End If
    If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
        t_ciclo = 180.63

    ElseIf tipo = "HIERRO GRIS INTERMEDIO"
        t_ciclo = 194.30

    ElseIf tipo = "HIERRO DUCTIL"
        t_ciclo = 194.30

    Else
        MsgBox("Material " & tipo & " no disponible para el C.C 170, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
        Exit Sub
    End If
    tiempo_maquina = (((144.78+t_ciclo)*_h1)/216)*100
    '((( ( 124.7549 + t_ciclo ) * _h1 ) / 216 ) * 100,3)
    tiempo_mano = tiempoMaquina
End Sub
```

**Ilustración 45 Vriables nuevas y método calcular().- CT-236.**

Comparándola con la fórmula anteriormente utilizada, esta fórmula se actualizó modificando solo un valor y quitando el redondeo utilizado.

### 3.1.5.5 CentroTrabajo255.vb – GAP SIZER WET (SIM HUMEDA).

En este centro de trabajo se actualizaron los valores del tiempo ciclo almacenado en la variable “t\_ciclo” que tenían anteriormente para cada tipo de material que se utiliza en el centro (Ilustración 46).

```
Public Sub calcular() implements ICentroTrabajo.calcular
    buscar_septup()
    Dim t_ciclo, dim_guillotina As Double
    If banElementoFijo = False Then
        tipo = Module1.sacar_tipo_material(_material)
    End If
    If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Then
        t_ciclo = 5.06
    ElseIf tipo = "HIERRO DUCTIL"
        t_ciclo = 8.320
    ElseIf tipo = "ACERO INOXIDABLE" or tipo = "ACERO AL CARBON"
        t_ciclo = 12.076
    ElseIf tipo = "HIERRO GRIS ALTO MODULO"
        t_ciclo = 7.07
    ElseIf tipo = "HIERRO GRIS INTERMEDIO"
        t_ciclo = 10.79
    Else
        msgbox("Material " & tipo & " no disponible para el C.T. 255, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
    End If
    Exit Sub
End Sub
```

**Ilustración 46 método calcular() y tiempos ciclo actualizados.- CT-255.**

En la misma función se declaró un IF para validar si los valores ingresados por el usuario están dentro de un rango el cual dependiendo de las medidas obtiene cual es la guillotina a utilizar; también se actualizó la antigua fórmula por la siguiente:  $((t\_ciclo+1.43)/(36*dim\_guillotina/_h1))*100$  (Ilustración 47).

```
-----se agrego esta condicion para que tomara un valor dentro del rango y asi poder utilizar la guillotina necesaria---
    If _h1 >=0.020 And _h1 <0.0460 Then
        dim_guillotina =0.2740
    ElseIf _h1 >=0.0460 And _h1 <0.0580 then
        dim_guillotina = 0.2740
    ElseIf _h1 >=0.0580 And _h1 <0.0625 Then
        dim_guillotina = 0.3520
    ElseIf _h1 >= 0.0625 And _h1 <0.0680 Then
        dim_guillotina = 0.3730
    ElseIf _h1 >=0.0680 And _h1 <0.0780 Then
        dim_guillotina = 0.4060
    ElseIf _h1 >=0.0780 And _h1 <0.0935 Then
        dim_guillotina = 0.3880
    ElseIf _h1 >=0.0935 And _h1 <0.0960 Then
        dim_guillotina = 0.4650
    ElseIf _h1 >= 0.0960 And _h1 <0.0980 Then
        dim_guillotina = 0.3250
    ElseIf _h1 >=0.0980 And _h1 <0.1180 Then
        dim_guillotina = 0.3880
    ElseIf _h1 >=0.1180 And _h1 <0.1250 Then
        dim_guillotina = 0.3520
    ElseIf _h1 >=0.1250 And _h1 <0.1370 Then
        dim_guillotina = 0.3730
    ElseIf _h1 >=0.1370 And _h1 <0.1550 Then
        dim_guillotina = 0.2740
    ElseIf _h1 >=0.1550 And _h1 <0.1570 Then
        dim_guillotina = 0.3100
    ElseIf _h1 >=0.1570 And _h1 <0.1870 Then
        dim_guillotina = 0.4650
    ElseIf _h1 >=0.1870 And _h1 <0.1950 Then
        dim_guillotina = 0.3730
    ElseIf _h1 >=0.1950 And _h1 <0.2490 Then
        dim_guillotina = 0.3880
    ElseIf _h1 >=0.2490 And _h1 < 0.3000 Then
        dim_guillotina = 0.4990
    End If
    tiempo_maquina =(((t_ciclo+1.43)/(36*dim_guillotina/_h1))*100
    '(((67.4651 + t_ciclo) * _h1) / 133.2) * 100,3)
    tiempo_mano = tiempo_maquina
End Sub
```

**Ilustración 47 condicionante IF dentro del método calcular(). - CT-255.**

### 3.1.5.6 CentroTrabajo261.vb – LASSER ENGRAVE.

La actualización de este centro se llevó a cabo solo modificando el código de la función calcular(), ya que se actualizó el tipo de marcado, anteriormente aparecían solo las iniciales del tipo de marcado, ahora aparece el nombre completo de la operación, se declaró una nueva variable llamada “a\_ciclicas” y se le asignaron nuevos valores a la variable “t\_ciclo\_caracter” junto con “a\_ciclicas” para cada uno de los tipos de marcado que se utilizan, también se agregó un nuevo tipo de marcado que es “sin recubrimiento” (Ilustración 48).

```
Public Sub calcular() implements ICentroTrabajo.calcular
    tiempo_septup = buscar_setupin(centro)
    Dim t_ciclo_caracter As Double = 0
    Dim t_ciclo_logotipo As Double = 0
    Dim a_ciclicas As Double = 0

    If _tipoMarcado = "LASSER AUTOMATICO FOSFATO" Then
        t_ciclo_caracter = 0.36
        a_ciclicas = 0.07
        't_ciclo_caracter = 0.1184
        If _llevaLogotipo Then
            t_ciclo_logotipo = 0.1191
        End If
    ElseIf _tipoMarcado = "LASSER MANUAL FOSFATO"
        t_ciclo_caracter = 0.59
        a_ciclicas = 3.67
        't_ciclo_caracter = 0.1105
        If _llevaLogotipo Then
            t_ciclo_logotipo = 0.1359
        End If
    ElseIf _tipoMarcado = "LASSER AUTOMATICO CROMO"
        t_ciclo_caracter = 1.78
        a_ciclicas = 0.07
        't_ciclo_caracter = .1618
        If _llevaLogotipo Then
            t_ciclo_logotipo = 0.1685
        End If
    ElseIf _tipoMarcado = "SIN RECUBRIMIENTO"
        t_ciclo_caracter = 1.78
        a_ciclicas = 0.07
    End If

    tiempo_maquina = (((t_ciclo_caracter * _NoCaracteres) + a_ciclicas + 0.02)/36)*100
    'tiempo_maquina = math.Round((3.62111+(( _NoCaracteres * t_ciclo_caracter) + (t_ciclo_logotipo)))/36,3)
    tiempo_mano = tiempo_maquina
End Sub
```

**Ilustración 48 método calcular( ), tiempos actualizados y nuevos valores de "a\_ciclicas".- CT-261.**

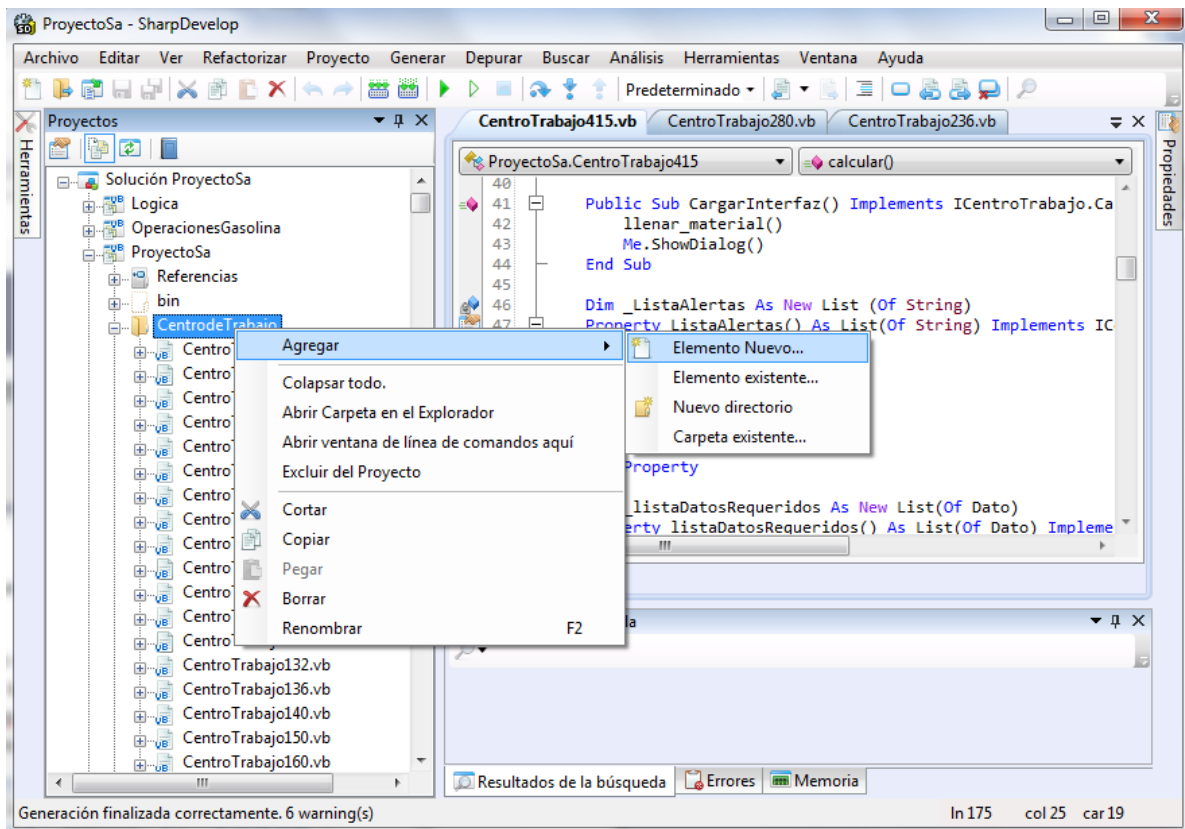
La fórmula actualizada quedo de la siguiente forma:

$$(((t\_ciclo\_caracter * \_NoCarateres) + a\_ciclicas + 0.02) / 36) * 100.$$

### 3.1.5.7 CentroTrabajo265.vb – ENGRAVE HIGH RUNNERS.

Este centro se agregó desde cero ya que no se contaba con él en el software, pero si está en funcionamiento en planta.

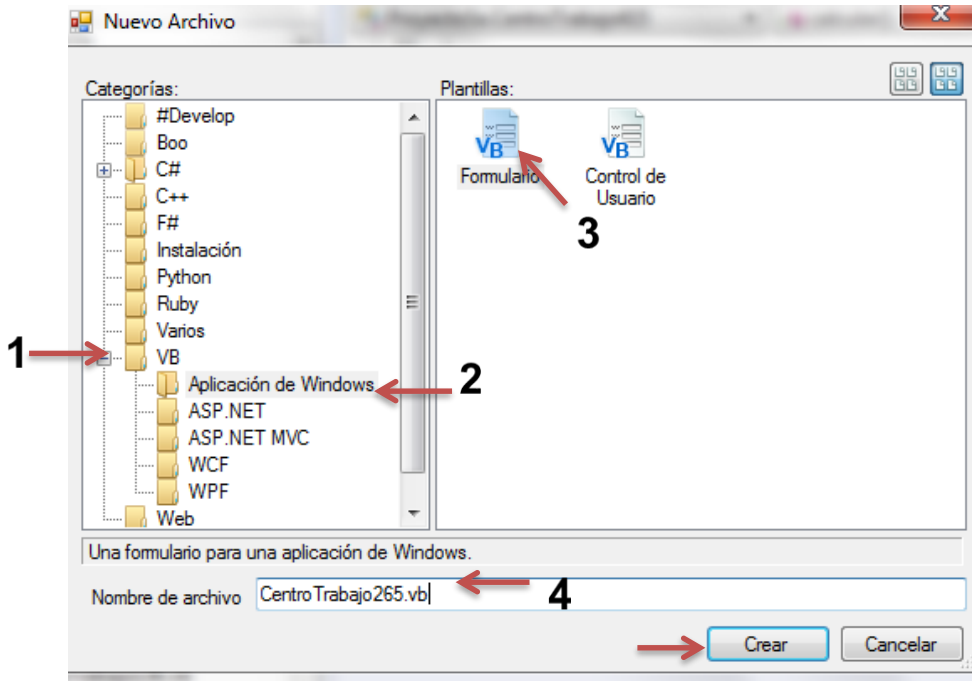
Para agregar un nuevo centro de trabajo hay que buscar la carpeta de nombre “CentrodeTrabajo” en el proyecto, una vez en ella darás clic con el botón secundario para que te despliegue el menú de opciones, después das clic en Agregar y por último en elemento nuevo (Ilustración 49).



**Ilustración 49 Agregar nuevo centro de trabajo. - CT-265.**

Una vez realizado este procedimiento se abrirá una nueva ventana con las opciones de Categorías y plantillas que se pueden agregar al proyecto, para este proyecto es necesario (1) un archivo de la categoría VB (visual Basic), (2) seleccionando la sub carpeta de nombre Aplicación de Windows te mostrará las plantillas incluidas en esa sub categoría; (3)seleccionas la plantilla Formulario, (4) le asignas un nuevo nombre en este caso es CentroTrabajo265.vb, (5) das clic en crear esperas un poco y listo se creó el nuevo form en blanco listo para codificarlo y aparece en pantalla con solo el código requerido para su ejecución sin ninguna función (Ilustración 50).





**Ilustración 50 Agregar nuevo centro de trabajo.- CT-265.**

En este centro se codificarán todas las funciones necesarias para el buen funcionamiento del mismo, empezando por importar las funciones de Logica, después se implementará la interface Logica.ICentroTrabajo dentro de la clase patriarcal mediante el siguiente código “implements ICentroTrabajo” que contiene las propiedades necesarias para obtener y procesar los valores ingresados por el usuario.

```

9 Imports Logica
10 Public Partial Class CentroTrabajo265
11     Implements ICentroTrabajo
12

```

**Ilustración 51 Imports Logica. -CT -265.**

Después se declara la variable `_ListaMateriaPrima` que es la que genera una nueva lista de los materiales a utilizar extrayendo el código de materia prima, la cantidad que se utiliza en el componente, unidad de medida del SAP y la descripción, implementado la interface de Logica.IMateriaPrima; y la propiedad `ListaMateriaPrima()` que también genera una lista pero solo de las materias primas utilizadas en este centro de trabajo, implementando las funciones de `ICentroTrabajo.ListaMateriaPrima` que son las encargadas de crear la lista necesaria para este centro de trabajo. Se agregaron las propiedades `tiempoSetup`, `tiempoMano` y `tiempoMaquina` que se utilizan para leer y mostrar los valores de los tiempos. Se declaró la función `CargarInterfaz()` implementada para este centro y se codificó con el código `Me.calcular()` para que en lugar de mostrar la interfaz gráfica solo realizara el cálculo necesario (Ilustración 52).

```

13 Dim _ListaMateriaPrima As New List(Of IMateriaPrima)
14 Property ListaMateriaPrima() As List (Of IMateriaPrima) Implements ICentroTrabajo.ListaMateriaPrima
15     Get
16         Return _ListaMateriaPrima
17     End Get
18     Set
19         _ListaMateriaPrima = value
20     End Set
21 End Property
22
23 ReadOnly Property tiempoSetup As Double Implements ICentroTrabajo.tiempoSetup
24     Get
25         Return tiempo_septup
26     End Get
27 End Property
28
29 ReadOnly Property tiempoMano As Double Implements ICentroTrabajo.tiempoMano
30     Get
31         Return tiempo_mano
32     End Get
33 End Property
34
35 ReadOnly Property tiempoMaquina As Double Implements ICentroTrabajo.tiempoMaquina
36     Get
37         Return tiempo_maquina
38     End Get
39 End Property
40
41 Public Sub CargarInterfaz() Implements ICentroTrabajo.CargarInterfaz
42     Me.calcular()
43 End Sub
44 Dim _ListaAlertas As New List (Of String)
45 Property ListaAlertas() As List (Of String) Implements ICentroTrabajo.ListaAlertas
46     Get
47         Return _ListaAlertas
48     End Get
49     Set
50         _ListaAlertas = Value
51     End Set
52 End Property
53
54 Dim _listaDatosRequeridos As New List (Of Dato)
55 Property listaDatosRequeridos() As List (Of Dato) Implements ICentroTrabajo.listaDatosRequeridos
56     Get
57         Return _listaDatosRequeridos
58     End Get
59     Set
60         listaDatosRequeridos = value

```

**Ilustración 52 Propiedades de la interface. - CT-265.**

Se agregan las funciones Test y “pasarValoresDeListas” que convierten los datos dependiendo del valor ingresado (Ilustración 53).

```

83 Public function Test(ByVal lDatos As List(Of Dato), byVal lDatosBool as List(of DatoBoolean), Byval lDatosString as List(of DatoString)) as boolean implements ICentroTrabajo.Tr
84     Return True
85 End Function
86
87 Public Sub pasarValoresDeListas(ByVal lDatos as List(of Dato),ByVal lDatosBool as List(of DatoBoolean),ByVal lDatosString as List (Of DatoString)) Implements ICentroTrabajo.pa:
88 End Sub
89

```

**Ilustración 53 función Test y pasarValoresDeListas. - CT-265.**

Después se declara la variable “centro” del tipo String y se le asigna el número de centro que se está creando, en seguida se utiliza la propiedad CentroTrabajo que genera un String implementando “ICentroTrabajo.CentroTrabajo” para regresar el valor declarado en la variable centro y mostrarlo; en las líneas siguientes se declaran las variables “tiempo\_maquina”, “tiempo\_septup” y “tiempo\_mano” que almacenaran el valor numérico de los tiempos generados con los cálculos (Ilustración 54).

En seguida se declara la función calcular( ) donde se realizan las operaciones matemáticas para calcular los tiempos y se agrega la formula necesaria (Ilustración 54).

```

96 Dim centro As String ="265"
97 Public Property CentroTrabajo As String Implements ICentroTrabajo.CentroTrabajo
98     Get
99         return centro
100     End Get
101     Set
102         centro = value
103     End Set
104 End Property
105 Public tiempo_maquina As Double
106 Public tiempo_septup As Double
107 Public tiempo_mano As Double = 0
108
109 Public Sub calcular() Implements ICentroTrabajo.calcular
110     tiempo_septup = buscar_setupin(centro)
111     tiempo_maquina = Math.Round (0.0347580608*100,3)
112     tiempo_mano =0
113
114 End Sub
115 Dim _factorLabor As Double
116 Public Property factorLabor As Double Implements ICentroTrabajo.factorLabor
117     Get
118         Return _factorLabor
119     End Get
120     Set
121         _factorLabor = value
122     End Set
123 End Property
124 End Class

```

Ilustración 54 propiedad CentroTrabajo, método calcular( ) y propiedad factorLabor.- CT-265.

### 3.1.5.8 CentroTrabajo280.vb – BATES BORE.

Prácticamente en este centro de trabajo solo se actualizo la parte de código donde se encuentra la fórmula que se actualizo por la siguiente:  $(\text{Width Nominal} * (\text{Tiempo Ciclo por tipo de Material} + 4.72)/36/2)$ , y los tiempos ciclo de cada uno de los hierros utilizados por el centro de trabajo (Ilustración 55).

```

177
178 Public Sub calcular() implements ICentroTrabajo.calcular
179     buscar_septup()
180     Dim cicloxmate As Double
181     cicloxmate = saca_tipohier(_material)
182
183     tiempo_maquina = (_h1 * (cicloxmate + 4.72)/36)/2
184     tiempo_maquina = math.Round(tiempo_maquina * 100,3)
185     tiempo_mano = tiempo_maquina
186 End Sub

```

Ilustración 55 método calcular( ). - CT- 280.

```

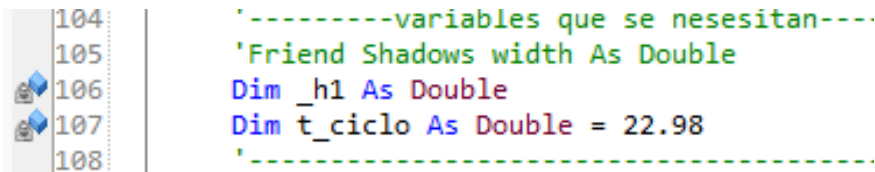
Public Function saca_tipohier(ByVal mat As String) As Double
    Dim vhie As Double = 0
    If banElementoFijo = False Then
        tipo = Module1.sacar_tipo_material(_material)
    End If
    If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Then
        vhie = 19.60'125.105538
    ElseIf tipo = "HIERRO GRIS INTERMEDIO"
        vhie = 19.60'374.907
    ElseIf tipo = "HIERRO GRIS ALTO MODULO"
        vhie = 19.60'374.9070
    ElseIf tipo = "HIERRO DUCTIL"
        vhie = 19.60
    Else
        msgbox("Material " & vhie & " no disponible para el C.C 280, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
    Exit Function
    End If
    Return vhie

```

**Ilustración 56 método "saca\_tipohier" para obtener tiempo ciclo. - CT- 280.**

### 3.1.5.9 CentroTrabajo290.vb – PRUEBA DE MANDRIL.

La actualización de este centro de trabajo solo fue en la parte del código donde se encuentra la fórmula para obtener los tiempos ya que solo se requirió de cambiar o actualizar un valor, ya que se optimizaron los tiempos de ejecución y se redujo en decimas el valor actualizado; para ello se tuvo que identificar el método calcular() y modificar la formula con la que cuanta el centro con los nuevos valores necesarios, quedando de la siguiente forma  $((81.13 + \text{Tiempo Ciclo de Material}) * \text{width nominal}) / 734.4$  (Ilustración 58), también se declaró una variable nueva llamada "t\_ciclo" que almacena el valor del tiempo ciclo de todos los materiales ya que es el mismo en todos los casos y es necesario para la realización de los cálculos (Ilustración 57).



```

104 | '-----variables que se nesesitan----
105 | 'Friend Shadows width As Double
106 | Dim _h1 As Double
107 | Dim t_ciclo As Double = 22.98
108 | '-----

```

**Ilustración 57 Nueva Variable "t\_ciclo".- CT- 290.**

```

Public Sub calcular() implements ICentroTrabajo.calcular
    tiempo_septup=buscar_setupin(centro)
    tiempo_maquina =Math.Round(((81.13+t_ciclo)*_h1 )/734.4,5)*100
    tiempo_mano=tiempo_maquina
End Sub

```

**Ilustración 58 método calcular().- CT- 290.**

### 3.1.5.10 CentroTrabajo410.vb – FINISH MILL.

La actualización de este centro de trabajo se realizó actualizando el tiempo ciclo de los materiales a utilizar y la formula ya que se optimizaron los tiempos en comparación con los anteriores. Para ello tenemos que encontrar el método “calcular( )” en el código del centro de trabajo para poder actualizar los valores de los tiempos ciclo y la formula con los valores optimizados que se muestran en la siguiente tabla.

Tabla 4 tiempos ciclo optimizados.- CT- 410.

TIPO DE MATERIAL	TIEMPO CICLO
HIERRO DUCTIL	21.28
HIERRO GRIS	12.27
HIERRO GRIS ALTO MOD	12.67
HIERRO GRIS INTERMEDIO	12.67

Como se puede observar en la ilustración 59, Tiempos ciclo y formula actualizados – CT410 la fórmula que se utilizaba anteriormente se ve de color verde, y la fórmula actualizada con los tiempos optimizados se muestra en la línea 191 enseguida de la formula pasada.

```
172 Public Sub calcular() implements ICentroTrabajo.calcular
173     buscar_septup()
174     Dim t_ciclo As Double
175     If banElementoFijo = False Then
176         tipo = Module1.sacar_tipo_material(_material)
177     End If
178     If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Then
179         t_ciclo = 12.27
180     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
181         t_ciclo = 12.67
182     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
183         t_ciclo = 12.67
184     ElseIf tipo = "HIERRO DUCTIL"
185         t_ciclo = 21.28
186     Else
187         MsgBox("Material " & tipo & " no disponible para el C.C 410, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
188         Exit Sub
189     End If
190     'tiempo_maquina = math.Round((( ( 52.49 + t_ciclo ) * _h1) / 198) * 100,3)
191     tiempo_maquina= Math.Round(((t_ciclo+41.83)*_h1)/226.8,4)*100
192     tiempo_mano = tiempo_maquina
193 End Sub
```

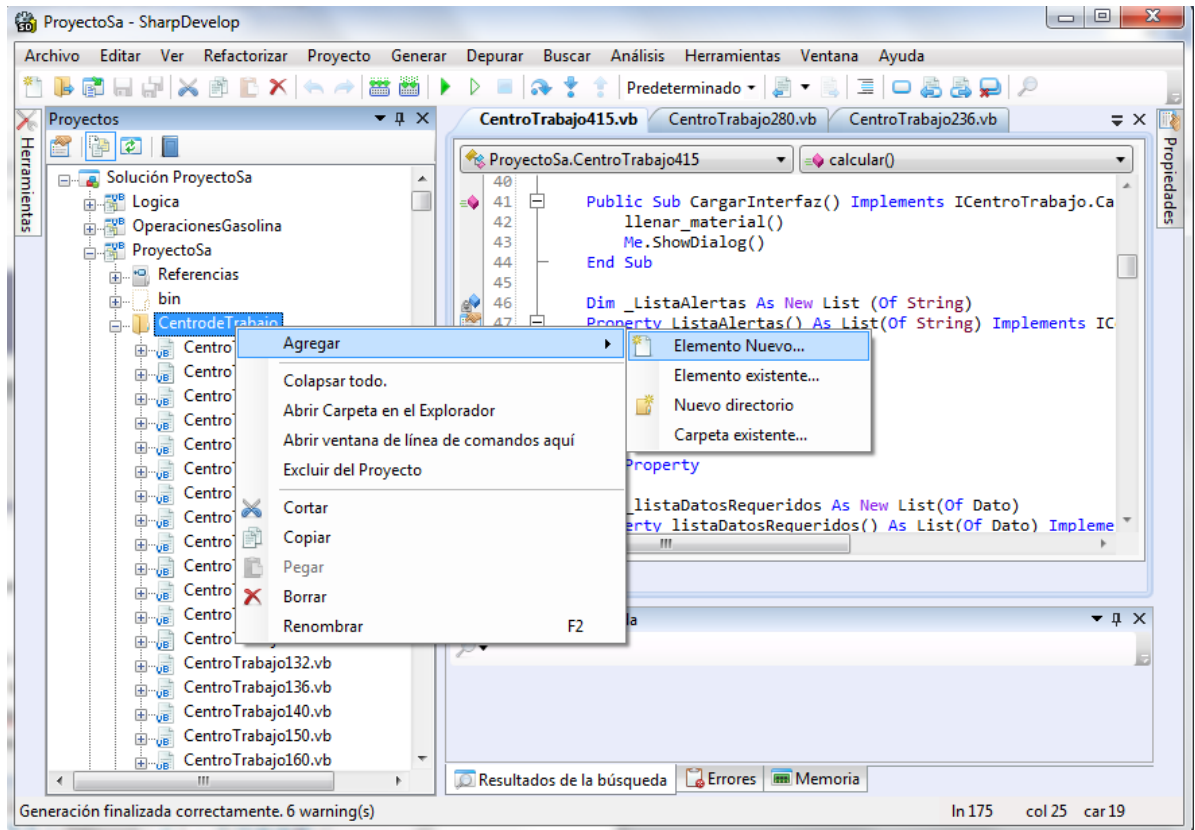
Ilustración 59 método calcular( ), tiempos ciclo actualizados.- CT- 410.

### 3.1.5.11 CentroTrabajo415.vb – FINISH MILL (HIGH RUNNERS).

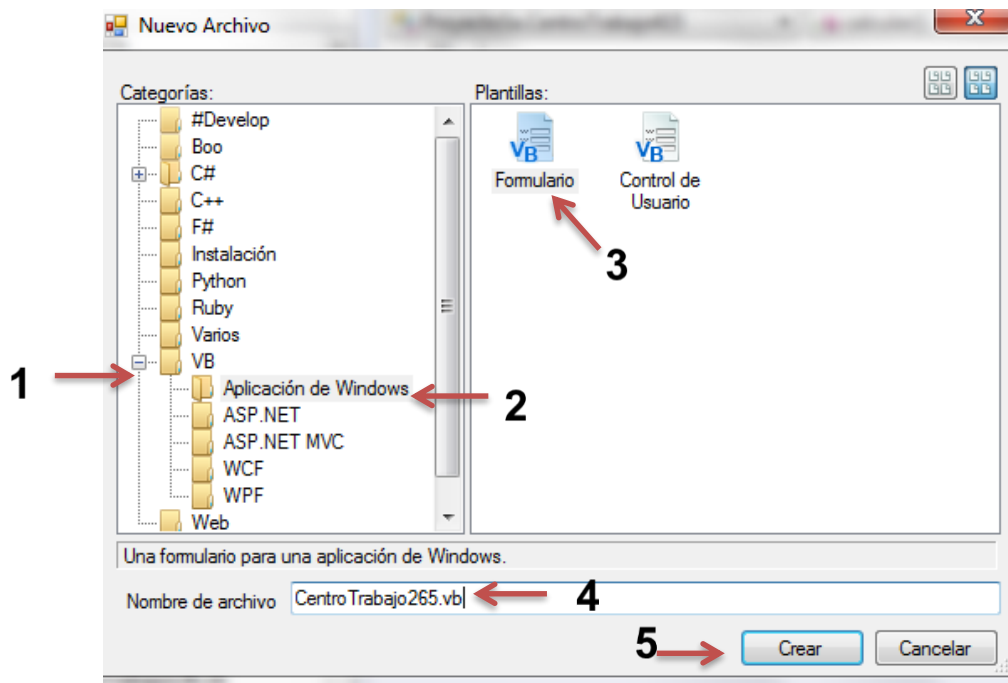
Este centro de trabajo se agregó desde cero ya que el sistema no contaba con el pero si está en funcionamiento en la planta.

Para agregar el nuevo centro de trabajo hay que buscar la carpeta llamada “CentrodeTrabajo” en el proyecto, una vez en ella darás clic con el botón secundario sobre ella para que te despliegue el menú de opciones, después das clic en agregar y por último en elemento nuevo (Ilustración 60).

Una vez realizado este procedimiento se abrirá una nueva ventana con las opciones de categorías y plantillas que se pueden agregar al proyecto, para este proyecto es necesario (1) un archivo de la categoría VB (Visual Basic), (2) seleccionándola subcarpeta de nombre “Aplicación de Windows“ te mostrará las plantillas incluidas en esa sub categoría, (3) seleccionas la plantilla „Formulario“, (4) le asignas un nuevo nombre en este caso „CentroTrabajo415.vb, (5) das clic en crear, esperas un poco y listo se creó un nuevo formulario en blanco listo para codificarlo y aparece en pantalla con solo el código requerido para su ejecución sin ninguna función (Ilustración 61).



**Ilustración 60 Agregar nuevo Centro de Trabajo.- CT- 415.**



**Ilustración 61 Agregar Centro de Trabajo. CT- 415.**

En este centro se codificarán todas las funciones necesarias para el buen funcionamiento del mismo, empezando por importar las funciones de Logica, después se implementará la interface Logica.ICentroTrabajo dentro de la clase patriarcal mediante el siguiente código “implements ICentroTrabajo” que contiene las propiedades necesarias para obtener y procesar los valores ingresados por el usuario.

```

9 Imports Logica
10 Public Partial Class CentroTrabajo415
11 Implements ICentroTrabajo

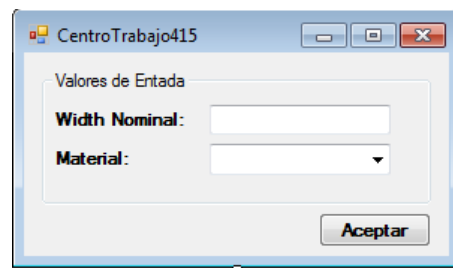
```

**Ilustración 62 Imports Logica.- CT- 415.**

Después se declara la variable `_ListaMateriaPrima` que es la que genera una nueva lista de los materiales a utilizar extrayendo el código de materia prima, la cantidad que se utiliza en el componente, unidad de medida del SAP y la descripción, implementado la interface de Logica.IMateriaPrima; y la propiedad `ListaMateriaPrima()` que también genera una lista pero solo de las materias primas utilizadas en este centro de trabajo, implementando las funciones de `ICentroTrabajo.ListaMateriaPrima` que son las encargadas de crear la lista necesaria para este centro de trabajo. Se agregaron las propiedades `tiempoSetup`, `tiempoMano` y `tiempoMaquina` que se utilizan para leer y mostrar los valores de los tiempos.

Se declaró la función CargarInterfaz() implementada para este centro y se codificó mandando llamar el método llenar\_material() y Me.ShowDialog(), para que realice las funciones del método llenar\_material() y muestre en pantalla la interfaz gráfica del form(formulario).

En la interfaz gráfica se modificó agregando 2 labels (material y Width Nominal) al form, un TextBox y un comboBox para los valores necesarios que tiene que ingresar el usuario quedando de la siguiente manera (Ilustración 63).



**Ilustración 63 Interface Gráfica.- CT - 415.**

En la codificación se tuvo que empezar por importar las librerías necesarias para el buen funcionamiento del sistema; también se declararon las variables que utilizan los métodos declarados en el modulo1.

En el método Text se declara un IF para condicionar el tipo de material a utilizar en el centro de trabajo, y en caso de no ser uno de los que aparecen en la condición mandara un mensaje de error indicando que no se pueden hacer los cálculos para el centro de trabajo

En el método calcular( ) se declara una condicionante IF para saber cuál es el tipo de material a utilizar en el centro de trabajo, en caso de seleccionar uno de los que no aparecen en la condición mandara un mensaje de error indicando que no se pueden realizar los cálculos para el centro de trabajo utilizado y cerrara el form, en caso contrario procederá a realizar las operaciones y cálculos matemáticos para obtener los tiempos deseados.

En la función del botón se programó que al hacer clic sobre él y si no ha seleccionado algún tipo de material te mandará un mensaje de alerta para que revise los campos una vez cumplidas todas las condiciones procederá a realizar los cálculos necesarios para la obtención de los tiempos estándar.



### 3.1.5.12 CentroTrabajo420.vb – AUTO. FINISH TURN

Inicialmente como se ha hecho con los demás centros de trabajo, se debe de verificar si existe o no algún cambio en la interfaz gráfica del formulario. Observe la siguiente imagen:

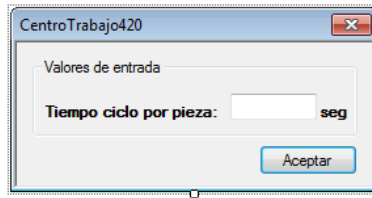


Ilustración 64 form antes de la actualización.- CT- 420.

A este centro de trabajo se le eliminaron los valores de entrada ya que el tiempo ciclo es el mismo para todos los materiales utilizados en este centro de trabajo. Lo cual indica que el valor del tiempo estándar no será variante y siempre será el mismo. Entonces solo se tiene que cambiar la formula y asignarle el valor del tiempo ciclo a la variable ciclo.

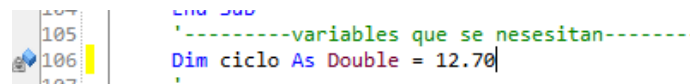


Ilustración 65 Variable tiempo ciclo actualizada. - CT -420.

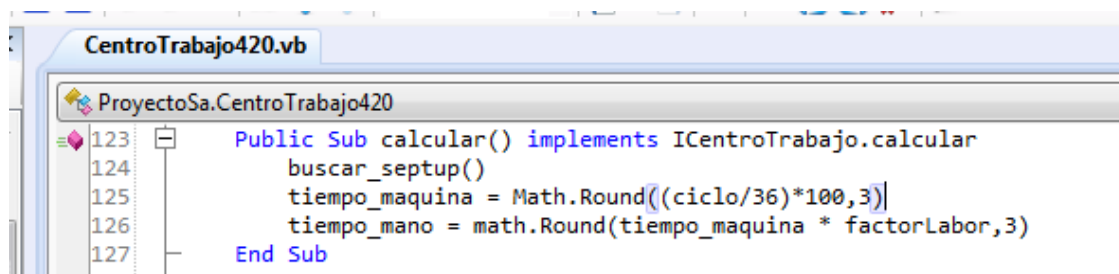


Ilustración 66 método calcular( ). - CT- 420.

Una vez que se cambió la formula este centro de trabajo queda actualizado.

### 3.1.5.13 CentroTrabajo424.vb – AUTO FINISH TURN GANG KATAOKA.

La actualización de este centro de trabajo se realizó modificando en el método calcular( ), la fórmula para obtener el tiempo máquina y eliminando la lista de materiales que no se utilizan ya que el tipo de material es el mismo porque la maquina esta esclavizada al mismo componente y el tiempo ciclo no cambia (Ilustración 67). Por esta razón también se decidió ocultar los objetos que no se utilizan que son el label1 (Tiempo ciclo por

pieza), label3 (seg.) y el textBox1 que era donde se introducía el valor del tiempo ciclo por el usuario; esta instrucción se declara en el evento CentroTrabajo424Load (Ilustración 68).

```

146
147 Public Sub calcular() Implements ICentroTrabajo.calcular
148     buscar_septup()
149     Dim t_ciclo As Double
150
151     tipo = module1.sacar_tipo_material(_material)
152     If tipo = "HIERRO GRIS" Then
153         t_ciclo = 28.25
154     Else
155         msgbox("Material " & tipo & " no disponible para el C.C 424, imposib:
156         Exit Sub
157     End If
158     tiempo_maquina = math.Round( 0.793605* 100,3)
159     '(( ( 36.8009 + t_ciclo ) * _h1 ) / 30.24)
160     tiempo_mano = tiempo_maquina * factorLabor
161 End Sub

```

Ilustración 67 método calcular().- CT- 424.

```

:56 Sub CentroTrabajo424Load(sender As Object, e As EventArgs)
:57     label1.Visible = False
:58     textBox1.Visible = False
:59     label3.Visible =False
:60 End Sub
:61 End Class
:62

```

Ilustración 68 Evento Load.- CT- 424.

### 3.1.5.14 CentroTrabajo425.vb – AUTO. FINISH TURN (HIGH RUNNERS)

Este centro de trabajo se agregó desde cero ya que no se contaba con el anteriormente en el software, pero si está en funcionamiento en planta.

Para crear este centro de trabajo se agregó un nuevo formulario al directorio centrodetrabajo que se encuentra dentro del ProyectoSa, con el nombre CentroTrabajo425.vb. Una vez que se creó se le agregaron las siguientes líneas de código las cuales debe llevar todo centro de trabajo para su buen funcionamiento. A continuación, se muestra el código Ilustración 69.

```

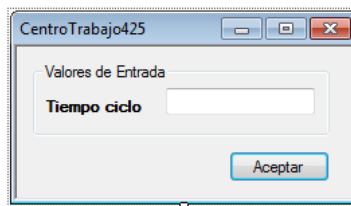
CentroTrabajo425.vb
ProyectoSa.CentroTrabajo425
Imports System.Data
Imports System.Data.OleDb
Imports System.Data.SqlClient
Imports Logica
Public Partial Class CentroTrabajo425
    Implements ICentroTrabajo
16 Dim _ListaMateriaPrima As New List(Of IMateriaPrima)
17 Property ListaMateriaPrima() As List(Of IMateriaPrima) Implements ICentroTrabajo.ListaMateriaPrima
18 Get
19     Return _ListaMateriaPrima
20 End Get
21 Set
22     _ListaMateriaPrima = value
23 End Set
24 End Property
25
26 ReadOnly Property tiempoSetup As Double Implements ICentroTrabajo.tiempoSetup
27 Get
28     Return tiempo_septup
29 End Get
30 End Property
31
32 ReadOnly Property tiempoMano As Double Implements ICentroTrabajo.tiempoMano
33 Get
34     Return tiempo_mano
35 End Get
36 End Property
37
38 ReadOnly Property tiempoMaquina As Double Implements ICentroTrabajo.tiempoMaquina
39 Get
40     Return tiempo_maquina
41 End Get
42 End Property
43
44 Public Sub CargarInterfaz() Implements ICentroTrabajo.CargarInterfaz
45     Me.ShowDialog()
46 End Sub
47
48 Dim _ListaAlertas As New List(Of String)
49 Property ListaAlertas() As List(Of String) Implements ICentroTrabajo.ListaAlertas
50 Get
51     Return _ListaAlertas
52 End Get
53 Set
54     _ListaAlertas = value
55 End Set
56 End Property
57
58 Dim _listaDatosRequeridos As New List(Of Dato)
59 Property listaDatosRequeridos() As List(Of Dato) Implements ICentroTrabajo.listaDatosRequeridos
60 Get
61     Return _listaDatosRequeridos
62 End Get
63 Set
64     _listaDatosRequeridos = value
65 End Set
66 End Property
67
68 Dim _listaDatosBooleanosRequeridos As New List(Of DatoBoolean)
69 Property listaDatosBooleanosRequeridos() As List(Of DatoBoolean) Implements ICentroTrabajo.listaDatosBooleanosRequeridos
70 Get
71     Return _listaDatosBooleanosRequeridos
72 End Get
73 Set
74     _listaDatosBooleanosRequeridos = value
75 End Set
76 End Property
77
78 Dim _listaDatosStringRequeridos As New List(Of DatoString)
79 Property listaDatosStringRequeridos() As List(Of DatoString) Implements ICentroTrabajo.listaDatosStringRequeridos
80 Get
81     Return _listaDatosStringRequeridos
82 End Get
83 Set
84     _listaDatosStringRequeridos = value
85 End Set
86 End Property
87
88 Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatoString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
89     Return True
90 End Function
91

```

**Ilustración 69 código necesario que debe llevar todo centro de trabajo.- CT- 425.**

El código que se muestra en la imagen anterior se compone de propiedades y métodos que se implementan con las propiedades de las clases de la librería lógica.

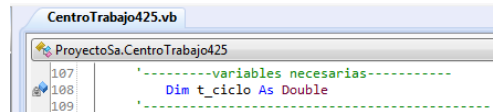
Después de haber agregado el código anterior, se creó la siguiente interfaz gráfica.



**Ilustración 70 Nuevo form.- CT- 425.**

Como se puede apreciar en la imagen anterior solamente se agrego un solo valor de entrada, el cual es tiempo ciclo.

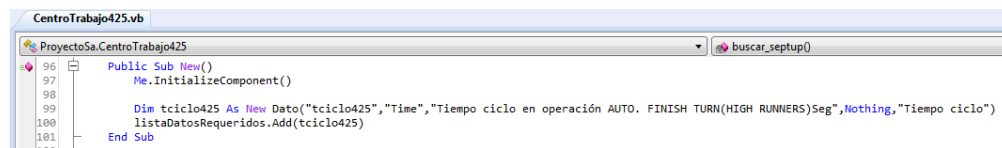
Una vez que se creó la interfaz gráfica del centro de trabajo se agregó la variable donde se almacenara el valor de entrada que el usuario ingrese al formulario.



```
CentroTrabajo425.vb
ProyectoSa.CentroTrabajo425
107 '-----variables necesarias-----
108 Dim t_ciclo As Double
109 '-----'
```

**Ilustración 71 variable para almacenar el valor de entrada.- CT- 425.**

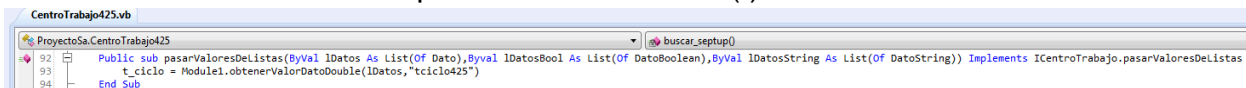
Despues se agrego una variable de tipo Dato y se agrego a la listaDatosRequeridos.



```
CentroTrabajo425.vb
ProyectoSa.CentroTrabajo425 buscar_septup()
96 Public Sub New()
97 Me.InitializeComponent()
98
99 Dim tciclo425 As New Dato("tciclo425","Time","Tiempo ciclo en operación AUTO. FINISH TURN(HIGH RUNNERS)Seg",Nothing,"Tiempo ciclo")
100 listaDatosRequeridos.Add(tciclo425)
101 End Sub
```

**Ilustración 72 Constructor donde se agregan los valores a las listas.- CT- 425.**

Una vez que se realizo lo anterior se debe igualar la variable correspondiente al tiempo ciclo con el metodo obtenerValorDatoDouble perteneciente al module1 del ProyectoSa, ademas se debe pasar el nombre de la variable tipo dato como parametro. Todo esto se realiza dentro del metodo pasarValoresDeListas( ).

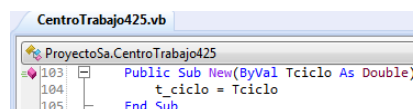


```
CentroTrabajo425.vb
ProyectoSa.CentroTrabajo425 buscar_septup()
32 Public sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBooleen),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
33 t_ciclo = Module1.obtenerValorDatoDouble(lDatos,"tciclo425")
34 End Sub
```

**Ilustración 73 función pasarValoresDeListas( ).- CT- 425.**

Nota: el metodo pasarValoresDeListas se debe de agregar cuando se agregan las propiedades y los metodos necesarios para el buen funcionamiento del centro de trabajo.

Posteriormente se agrego el parametro del tiempo ciclo al constructor de la clase, ademas de que dicho parametro se igualo con la variable correspondiente al valor de entrada.

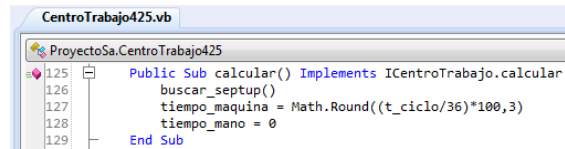


```
CentroTrabajo425.vb
ProyectoSa.CentroTrabajo425
103 Public Sub New(ByVal Tciclo As Double)
104 t_ciclo = Tciclo
105 End Sub
```

**Ilustración 74 Constructor y parámetros de la clase.- CT- 425.**

Ahora que se ya se tiene todo lo anterior se procede a crear el metodo calcular, el cual tambien implementa un metodo llamado igual pero con la diferencia que se encuentra en la clase ICentroTrabajo.vb del proyecto logica.

Una vez que se creo el metodo se agrego el siguiente codigo.

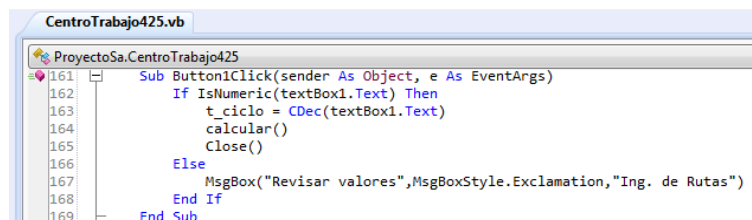


```
CentroTrabajo425.vb
ProyectoSa.CentroTrabajo425
125 Public Sub calcular() Implements ICentroTrabajo.calcular
126     buscar_septup()
127     tiempo_maquina = Math.Round((t_ciclo/36)*100,3)
128     tiempo_mano = 0
129 End Sub
```

**Ilustración 75 método calcular( ).- CT- 425.**

Lo primero que se realiza dentro del metodo es obtener el valor del tiempo setup mediante un metodo llamado buscar\_septup() el cual extrae el valor de la tabla CentroTrabajo de la base de datos RGP2. Despues se almaceno la formula para calcular el tiempo estandar en la variable tiempo\_maquina. Dicha formula depende del valor de tiempo ciclo que el usuario ingrese.

Despues de haber realizado el metodo calcular() se agrego el evento clic al boton del formulario y dentro del evento se agrego el siguiente codigo.



```
CentroTrabajo425.vb
ProyectoSa.CentroTrabajo425
161 Sub Button1Click(sender As Object, e As EventArgs)
162     If IsNumeric(textBox1.Text) Then
163         t_ciclo = CDec(textBox1.Text)
164         calcular()
165         Close()
166     Else
167         MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
168     End If
169 End Sub
```

**Ilustración 76 Evento click del botón aceptar.- CT- 425.**

Lo que se realiza primero en este codigo es una condicion if en donde si los datos que ingreso al formulario son numericos, entonces estos mismos se almacenaran en la variables que se les asignaron, y posteriormente se realizara el metodo calcular y se obtendra el valor del tiempo estandar, y finalmente se cerrara el formulario.

### **3.1.5.15 CentroTrabajo427.vb – LINTECH A.F.T.**

En este centro lo que se realizó fue agregar a la lista de materiales los componentes del tipo “HIERRO GRIS INTERMEDIO” y modificar el tiempo ciclo de cada uno de los materiales utilizados en este centro; se agregó a la condición IF para saber qué tipo de material selecciono si escogió uno de los que están en el condicionamiento procederá a realizar las operaciones y cálculos matemáticos para obtener el tiempo estándar. También se cambio la fórmula que se encuentra en el método calcular( ) del centro de trabajo.

### 3.1.5.16 CentroTrabajo428.vb – AXIAL FORM.

En este centro de trabajo solo se actualizo la fórmula matemática para obtener el tiempo máquina en el método calcular( ) quedando de la siguiente manera: 0.123516085\*100 este valor solo se optimizo en unas décimas de segundo como se puede apreciar en la Ilustración 77 comparado con la formula anterior, la fórmula que aparece de color verde es la fórmula anterior.

```
111  
112 Public Sub calcular() implements ICentroTrabajo.calcular  
113     tiempo_septup = buscar_setupin(centro)  
114     tiempo_maquina = math.Round(0.123516085*100,3)  
115     '(0.125344978116953 * 100,3)  
116     tiempo_mano = tiempo_maquina * factorLabor  
117 End Sub  
118
```

Ilustración 77 método calcular( ).- CT- 428.

### 3.1.5.17 CentroTrabajo450.vb – I.D. BRUSH.

En este centro de trabajo se modificó el método calcular( ) donde solo se cambió la formula anterior por la nueva que es más rápida y eficaz (Ilustración 78).

```
121  
122 Public Sub calcular() Implements ICentroTrabajo.calcular  
123     tiempo_septup=buscar_setupin(centro)  
124     tiempo_maquina = ((61.057*_h1)/226.8)*100  
125     tiempo_mano=tiempo_maquina  
126 End Sub  
127
```

Ilustración 78 método calcular( ).- CT- 450.

### 3.1.5.18 CentroTrabajo465.vb – SCOTCHBRITE RINGS.

La actualización de este centro de trabajo se llevó a cabo mediante la actualización de la fórmula matemática en el método calcular( ), solamente se actualizaron los antiguos valores por los más nuevos (Ilustración 79).

Nota: la fórmula que se ve de color verde es la fórmula que se utilizaba anteriormente en el sistema (Ilustración 79).

```
6: 128  
7: 129  
8: 130 Public Sub calcular() implements ICentroTrabajo.calcular  
9: 131     tiempo_septup=buscar_setupin(centro)  
10: 132     tiempo_maquina = math.Round((((_d1 *21.50)/3.622)+98.45)*_h1 )/141.732,3)*100  
11: 133     '((( ( 95.633 + ( ( _d1 * 20.86 ) / 4.25 ) ) * _h1 ) / 117) * 100,3)  
12: 134     tiempo_mano=tiempo_maquina  
13: 135 End Sub  
14: 136  
15: 137  
16: 138
```

Ilustración 79 método calcular( ).- CT- 465.

### 3.1.5.19 CentroTrabajo490.vb – GRINDING O.D. KATAOKA.

Para realizar la actualización en este centro de trabajo fue necesario cambiar solamente la fórmula matemática utilizada para obtener el tiempo máquina del centro, para ello se tiene que dirigir al método calcular( ) y cambiar la antigua formula por la nueva (Ilustración 80).

Nota: la fórmula que se en color verde es la fórmula que anteriormente se utilizaba ilustración 80.

```
120 |
121 |
122 | Public Sub calcular() implements ICentroTrabajo.calcular
123 |     tiempo_septup=buscar_setupin(centro)
124 |     tiempo_maquina =((128.932*_h1)/28.28736)*100
125 |     '((( ( 36.8009 + 78.11 ) * _h1 ) / 30.24) * 100,3)
126 |     tiempo_mano=tiempo_maquina * factorLabor
127 | End Sub
```

Ilustración 80 método calcular( ).- CT- 490.

### 3.1.5.20 CentroTrabajo495.vb – ROLADORA DE ANILLOS.

Para realizar la actualización en este centro de trabajo fue necesario cambiar solamente la fórmula matemática utilizada para obtener el tiempo máquina del centro, para ello se tiene que dirigir al método calcular( ) y cambiar la antigua formula por la nueva (Ilustración 81).

Nota: la fórmula que se en color verde es la fórmula que anteriormente se utilizaba.

```
121 |
122 | Public Sub calcular() implements ICentroTrabajo.calcular
123 |     tiempo_septup=buscar_setupin(centro)
124 |     tiempo_maquina =((0.0120+((2.06*diametro)/3.91730))/36)*100
125 |     '((( 0.0916 + ( ( 2.2 * diametro ) / 5.1181 ) ) / 36 ) * 100,3)
126 |     tiempo_mano=tiempo_maquina * factorLabor
127 | End Sub
```

Ilustración 81 método calcular( ).- CT- 495.

### 3.1.5.21 CentroTrabajo500.vb – POINT BEND.

Para la actualización de este centro se tiene que buscar el método calcular( ) ya que dentro de este se encuentran la operaciones matemáticas necesarias para obtener el tiempo maquina en este caso de tener inspección será un valor y en caso de no ser así será otro valor (Ilustración 82).

Nota: la fórmula que se en color verde es la fórmula que anteriormente se utilizaba.

```

122
123
124 Public Sub calcular() implements ICentroTrabajo.calcular
125     tiempo_septup=buscar_setupin(centro)
126     If b = True Then
127         tiempo_maquina = math.Round(1.3324 * 100,3)
128         '(1.35067 * 100,3)
129     Else
130         tiempo_maquina = math.Round(.6636 * 100,3)
131         '(.6819 * 100,3)
132     End If
133     tiempo_mano=tiempo_maquina
134 End Sub

```

Ilustración 82 método calcular( ).- CT- 500.

### 3.1.5.22 CentroTrabajo501.vb – ID POINT CHAMFER.

Para la actualización de este centro se tiene que buscar el método calcular( ) ya que dentro de este se encuentran la operaciones matemáticas necesarias para obtener el tiempo maquina en este caso solo se cambió el valor del redondeo en la operación matemática para que tomara más valores y fuera más exacto (Ilustración 83).

Nota: la fórmula que se en color verde es la fórmula que anteriormente se utilizaba.

```

112 Public Sub calcular() implements ICentroTrabajo.calcular
113     tiempo_septup=buscar_setupin(centro)
114     tiempo_maquina = math.Round(0.684584065934066 * 100,4)
115     '(0.684584065934066 * 100,3)
116     tiempo_mano=tiempo_maquina
117 End Sub

```

Ilustración 83 método calcular( ). -CT- 501.

### 3.1.5.23 CentroTrabajo767.vb – WHEELABRATOR GASOLINE.

Para la actualización de este centro se tiene que buscar el método calcular( ) ya que dentro de este se encuentran la operaciones matemáticas necesarias para obtener el tiempo maquina en este caso si los ciclos adicionales son igual a cero realizara una operación de no ser así realizara otra (Ilustración 84).

Nota: la fórmula que se en color verde es la fórmula que anteriormente se utilizaba.

```

130 Public Sub calcular() implements ICentroTrabajo.calcular
131     buscar_septup()
132     If _CiclosAdicionales = 0 Then
133         tiempo_maquina =Math.Round(((620.21*_width)/717.9894*100,3)
134         '((( 565.65 * _width ) / 622.8) * 100,3)
135     Else
136         tiempo_maquina = math.Round (((((620.21+(102.56*_CiclosAdicionales ))*_width)/718)*100,3)
137         '((( ( 565.65 + ( 102.56 * _CiclosAdicionales ) ) * _width) / 622.8) * 100,3)
138     End If
139     tiempo_mano = tiempo_maquina
140 End Sub

```

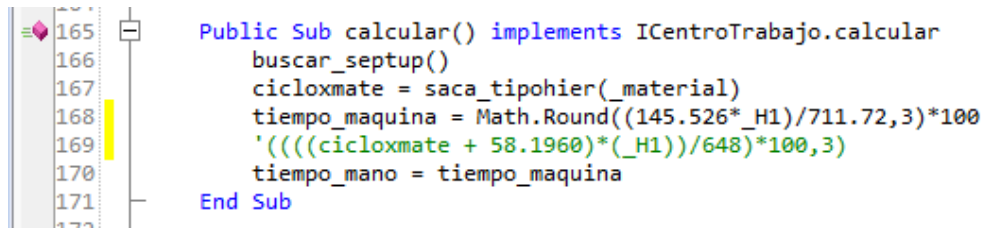
Ilustración 84 método calcular( ).- CT- 767.



### 3.1.5.24 CentroTrabajo810.vb – VISUAL INSPECTION PREM. GASOLINE.

Para la actualización de este centro se tiene que buscar el método calcular( ) ya que dentro de este se encuentran las operaciones matemáticas necesarias para obtener el tiempo máquina en este caso fue necesario cambiar solamente la fórmula matemática utilizada por la fórmula nueva (Ilustración 85).

Nota: la fórmula que se encuentra en color verde es la fórmula que anteriormente se utilizaba.



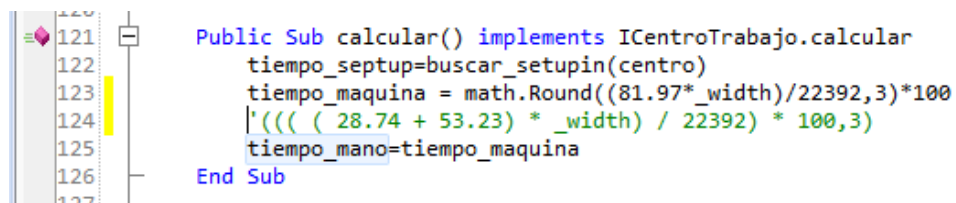
```
165 Public Sub calcular() implements ICentroTrabajo.calcular
166     buscar_septup()
167     cicloxmate = saca_tipohier(_material)
168     tiempo_maquina = Math.Round((145.526*_H1)/711.72,3)*100
169     '(((cicloxmate + 58.1960)*(_H1))/648)*100,3)
170     tiempo_mano = tiempo_maquina
171 End Sub
```

Ilustración 85 método calcular( ).- CT- 810.

### 3.1.5.25 CentroTrabajo815.vb – FINAL INSPECTION RINGS PREM. GASOLINE.

Para la actualización de este centro se tiene que buscar el método calcular( ) ya que dentro de este se encuentran las operaciones matemáticas necesarias para obtener el tiempo máquina en este caso fue necesario cambiar solamente la fórmula matemática utilizada por la fórmula nueva (Ilustración 86).

Nota: la fórmula que se encuentra en color verde es la fórmula que anteriormente se utilizaba.



```
121 Public Sub calcular() implements ICentroTrabajo.calcular
122     tiempo_septup=buscar_setupin(centro)
123     tiempo_maquina = math.Round((81.97*_width)/22392,3)*100
124     '((( ( 28.74 + 53.23) * _width) / 22392) * 100,3)
125     tiempo_mano=tiempo_maquina
126 End Sub
```

Ilustración 86 método calcular( ).- CT- 815.

### 3.1.5.26 CentroTrabajo820.vb – 100% GAP INSPECTION (AUTOGAP) GASOLINE.

Para la actualización de este centro se tiene que buscar el método calcular( ) ya que dentro de este se encuentran las operaciones matemáticas necesarias para obtener el tiempo máquina en este caso fue necesario cambiar solamente la fórmula matemática utilizada por la fórmula nueva (Ilustración 87).

Nota: la fórmula que se encuentra en color verde es la fórmula que anteriormente se utilizaba.

```

122
123 Public Sub calcular() implements ICentroTrabajo.calcular
124     tiempo_septup=buscar_setupin(centro)
125     tiempo_maquina =(,0763279345*100)
126     '(,072*100,3)
127     tiempo_mano = tiempo_maquina
128 End Sub

```

Ilustración 87 método calcular( ).- CT- 820.

### 3.1.5.27 CentroTrabajo830.vb – FINAL INSPECTION RINGS.

Para la actualización de este centro de trabajo fue necesario modificar la interface gráfica, ya que en un principio este centro de trabajo solicitaba al usuario ingresara varios valores para los diferentes tipos de objetos en el form, se modificó el código en la función Test( ) de donde se eliminó todo el código incluido dentro de la función solo se dejó el Return = True, en seguida fuera de esta función se encuentran declaradas dos variables, una llamada “tipo” y otra llamada “banElementoFijo”, estas dos variables también se eliminan (Ilustración 88).

```

87 Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
88     _material = module1.obtenerValorDatoString(lDatosString,"MATERIAL")
89     tipo = Module1.sacar_tipo_material(_material)
90
91     If tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Or tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO DUCTIL" Then
92
93     Else
94         MsgBox("Material " & tipo & " no disponible para el C.C " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
95             "Por favor seleccione un tipo de material disponible para poder calcular el centro de Trabajo",MsgBoxStyle.Critical,"Ing. de Rutas")
96     Dim listaOpcionales As New List(Of String)
97     listaOpcionales.Add("HIERRO GRIS")
98     listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
99     listaOpcionales.Add("HIERRO DUCTIL")
100    listaOpcionales.Add("HIERRO GRIS ALTO MODULO")
101    listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
102    listaOpcionales.Add("ACERO INOXIDABLE")
103    listaOpcionales.Add("ACERO AL CARBON")
104    Dim f As New Selectionar(listaOpcionales)
105    If f.ShowDialog() = DialogResult.OK Then
106        tipo = f.elementoSeleccionado
107        banElementoFijo = True
108    End If
109    Exit Function
110 End If
111 Return True
112 End Function
113
114 Dim tipo As String
115 Dim banElementoFijo As Boolean = False
116

```

Ilustración 88 función Test( ).- CT- 830.

En la función sub pasarValoresDeListas( ) se borra la condicionante IF que utilizaba la variable \_material (Ilustración 89).

```

117 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
118     _banPintura = module1.obtenerValorDatoBoolean(lDatosBool,"llevaPintura")
119     _noFranjasPintura = module1.obtenerValorDatoDouble(lDatos,"numeroDeFranjasPintura")
120     _banAceite = module1.obtenerValorDatoBoolean(lDatosBool,"llevaAceite")
121     _pasosAceite = module1.obtenerValorDatoDouble(lDatos,"numeroPasosDeAceite")
122     _resorte = module1.obtenerValorDatoBoolean(lDatosBool,"llevaResorte")
123     _di = module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")
124     If banElementoFijo = False Then
125         _material = module1.obtenerValorDatoString(lDatosString,"MATERIAL")
126     End If
127     _d1 = module1.obtenerValorDatoDouble(lDatos,"diaNominalAnillo")
128 End Sub

```

Ilustración 89 procedimiento Sub pasarValoresDeListas.- CT- 830.

Más adelante hay que ubicar la función Sub New ( ) y aquí se eliminan de las listas de herramental y datos requeridos (Ilustración 90).

```

130 | Public Sub New()
131 |     Me.InitializeComponent()
132 |     Dim llevaPintura As New DatoBoolean("llevaPintura","Indica si el componente lleva pintura","¿Pintar?")
133 |     Dim numeroDeFranjasPintura As New Dato("numeroDeFranjasPintura","Cantidad","Numero de franjas de pintura",Nothing,"No. de franjas pintura:")
134 |     Dim llevaAceite As New DatoBoolean("llevaAceite","Indica si el componente lleva aceite","¿Poner aceite?")
135 |     Dim numeroPasosDeAceite As New Dato("numeroPasosDeAceite","Cantidad","Numero de pasos de Aceite",Nothing,"No. pasos aceite:")
136 |     Dim llevaResorte As New DatoBoolean("llevaResorte","Indica si el componente lleva resorte","¿Colocar resorte?")
137 |     Dim widthNominalAnillo As New Dato("widthNominalAnillo","Distance","Width nominal del anillo (Plano)",Nothing,"Width nominal de anillo:")
138 |     Dim material_ As New DatoString("MATERIAL","MATERIAL Componente","Material:")
139 |     Dim diaNominalAnillo As New Dato("diaNominalAnillo","Distance","Diámetro nominal del anillo (Plano)",Nothing,"Diámetro nominal del anillo:")
140 |
141 |     'listaDatosRequeridos.Add(diaNominalAnillo)
142 |     'listaDatosStringRequeridos.Add(material_)
143 |     listaDatosRequeridos.Add(widthNominalAnillo)
144 |     listaDatosBooleanosRequeridos.Add(llevaPintura)
145 |     listaDatosRequeridos.Add(numeroDeFranjasPintura)
146 |     listaDatosBooleanosRequeridos.Add(llevaAceite)
147 |     listaDatosRequeridos.Add(numeroPasosDeAceite)
148 |     listaDatosBooleanosRequeridos.Add(llevaResorte)
149 | End Sub

```

**Ilustración 90 procedimiento Sub New( ).- CT- 830.**

Ahora nos dirigimos al método calcular ( ), y en esta parte de código hay que ubicar la condicionante IF que condiciona el tipo de materia seleccionado y proceder a borrar la condicionante. Posterior a esto se actualizan las fórmulas para realizar los cálculos del tiempo estándar cuando se le coloca o no resorte a la pieza (Ilustración 91).

```

189 | Public Sub calcular() implements ICentroTrabajo.calcular
190 |     buscar_septup()
191 |     Dim t_ciclo As Double
192 |     If _Resorte = True Then
193 |         Dim mResorte As New MateriaPrimaGenerica
194 |         With mResorte
195 |             .Cantidad = 10000
196 |             .CodigoMateriaPrima = obtenerSpring(_D1)
197 |             .Descripcion = "RESORTE"
198 |             .Measurement = "EA"
199 |         End With
200 |         ListaMateriaPrima.Add(mResorte)
201 |     End If
202 |
203 |     ' If banElementoFijo = False Then
204 |     '     tipo = Module1.sacar_tipo_material(material)
205 |     ' End If
206 |     ' If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Then
207 |     '     t_ciclo = 50.642*.99
208 |     ' ElseIf tipo = "HIERRO GRIS INTERMEDIO"
209 |     '     t_ciclo = 50.642*1.71
210 |     ' ElseIf tipo = "HIERRO GRIS ALTO MODULO"
211 |     '     t_ciclo = 50.642*.99
212 |     ' ElseIf tipo = "HIERRO DUCTIL"
213 |     '     t_ciclo = 50.642*1.71
214 |     ' ElseIf tipo = "ACERO INOXIDABLE" OR tipo = "ACERO AL CARBON"
215 |     '     t_ciclo = 50.642*.87
216 |     '
217 |     ' MsgBox("Material " & tipo & " no disponible para el C.C 830, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
218 |     ' Exit Sub
219 |     '
220 |
221 |     If _Resorte = True Then
222 |         tiempo_maquina = Math.Round((((98.274 + (4.102857 * _NoFranjasPintura) + (1.0924 * _PasosAceite) + tcicloResorte) * _H1) / 144) * 100,3)
223 |         'tiempo_maquina = math.Round((( ( ( 0.34 * ( 5 / _H1 ) ) + ( _PasosAceite * 0.029867 * ( 5 / _H1 ) ) + ( _NoFranjasPintura * 0.117858 * ( 5 / _H1 ) ) ) + ( 7
224 |
225 |     Else
226 |         tiempo_maquina = Math.Round((((98.274 + (4.102857 * _NoFranjasPintura) + (1.0924 * _PasosAceite)) * _H1) / 144) * 100,3)
227 |         'tiempo_maquina = math.Round((( ( ( 0.34 * ( 5 / _H1 ) ) + ( _PasosAceite * 0.029867 * ( 5 / _H1 ) ) + ( _NoFranjasPintura * 0.117858 * ( 5 / _H1 ) ) + ( t_ciclo * ( 5 / _H1 ) ) ) * _H1
228 |     End If
229 |     tiempo_man0 = tiempo_maquina
230 | End Sub

```

**Ilustración 91 método calcular( ).- CT- 830.**

Por ultimo hay que quitar todo rastro de que el centro de trabajo solicitaba al usuario ingresara el diámetro y el tipo de material a utilizar. Se quita de la función clic del botón aceptar la igualación de la variable “\_material = cbo\_material.text” y se elimina la función Sub llenar\_material( ). En la función load del centro de trabajo se ocultan los objetos que ya no son utilizados en el form (Ilustración 92).

```

402 ' Private Sub llenar_material()
403 '     Dim l As New List(Of String)
404 '     l.Add("HIERRO GRIS INTERMEDIO")
405 '     l.Add("HIERRO GRIS ALTO MODULO")
406 '     l.Add("HIERRO GRIS")
407 '     l.Add("HIERRO GRIS CENTRIFUGADO")
408 '     l.Add("HIERRO DUCTIL")
409 '     l.Add("ACERO INOXIDABLE")
410 '     l.Add("ACERO AL CARBON")
411 '     For Each m In obtenerEspeccsTipoMaterial(l)
412 '         cbo_material.Items.Add(m)
413 '     Next
414 ' End Sub
415
416 Sub CentroTrabajo830Load(sender As Object, e As EventArgs)
417     label3.Visible = False
418     txt_d1.Visible = False
419     'txt_h1.Location = new System.Drawing.Point (138,130)
420     'label2.Location = New System.Drawing.Point (6,137)
421
422 End Sub

```

**Ilustración 92 método Sub llenar\_material( ) y Evento Load( ).- CT- 830.**

## 3.1.6 Large Bore

### 3.1.6.1 CentroTrabajo9201.vb – Coil Steel L.B.

Para la actualización de este centro de trabajo fue necesario cambiar la fórmula que actualmente usaba por la nueva fórmula estandarizada para el buen funcionamiento del sistema; el cambio o actualización se llevó a cabo en la parte del código donde se encuentra el método calcular( ) del centro de trabajo 9201, donde se agregaron nuevas variables llamadas “carga”, “cXc” y “tciclo”, todas ellas de tipo Double. Enseguida se realizan las operaciones para obtener el valor de las variables anteriormente agregadas. Estos valores son muy necesarios para poder calcular el tiempo estándar con la formula directa para evitar errores con los decimales dicha fórmula actualizada se encuentra en la línea 143. Posterior a ella se encuentran las formulas simplificada línea 146 y obsoleta línea 149 que era la que se utilizaba en este centro de trabajo (Ilustración 93).

```

127 Public Sub calcular() Implements ICentroTrabajo.calcular
128     tiempo_septup=buscar_setupin(centro)
129
130     '-----VARIABLES QUE SE NECESITAN-----
131     Dim carga As Double
132     Dim cXc As Double
133     Dim tciclo As Double
134
135     '-----OPERACIONES PARA REALIZAR EL CALCULO DIRECTO-----
136
137     carga = (7/widths)-2]
138     tciclo = ((diametro * 138.04) / 9.062)
139     cXc = (tciclo + 35.2229071429 + 0.00)
140
141     '-----FORMULA DIRECTA ACTUALIZADA 03/10/2016-----
142
143     tiempo_maquina =(100 * ((cXc) / 3600)/(carga))*100
144
145     '-----FORMULA SIMPLIFICADA ACTUALIZADA-----
146     '(((35.22+(diametro*138.04)/9.062))/((7/widths)-2)*(36))*100)
147
148     '-----FORMULA OBSOLETA-----
149     '((( 115.28 + ( ( diametro * 210 ) / 9.062 ) ) / ( ( 7 / width ) - 1 ) * 36 )) * 100, 3)
150
151     tiempo_mano=tiempo_maquina
152 End Sub

```

**Ilustración 93 método calcular( ).- CT-9201.**

### 3.1.6.2 CentroTrabajo9202.vb – Coil Stainless Steel.

Para la actualización de este centro de trabajo fue necesario cambiar la fórmula que actualmente usaba por la nueva fórmula estandarizada para el buen funcionamiento del sistema; el cambio o actualización se llevó a cabo en la parte del método calcular ( ) del centro de trabajo 9202 donde se agregaron nuevas variables llamadas “carga”, “cXc” y “tc”, cada una de ellas de tipo Double. Enseguida se realizan las operaciones para obtener el valor de las variables anteriormente agregadas. Estos valores son muy necesarios para poder calcular el tiempo estándar con la formula directa para evitar errores con los decimales, dicha fórmula actualizada se encuentra en la línea 144. Posterior a ella se encuentra la formula obsoleta línea 147 que era la que anteriormente se utilizaba en este centro de trabajo (Ilustración 94).

```

128 Public Sub calcular() Implements ICentroTrabajo.calcular
129     '-----VARIABLES QUE SE NECESITAN-----
130     Dim carga As Double
131     Dim tc As Double
132     Dim cXc As Double
133
134     '-----OPERACIONES NECESARIAS PARA REALIZAR EL CALCULO DIRECTO-----
135
136     carga = (7.31 / widths)
137     tc = ((240.64 * d1 )/9.062)
138     cXc = (tc + 72.88 + 11.06)
139
140     Try
141         tiempo_septup=buscar_setupin(centro)
142         Dim t_ciclo As Double = 240.64
143
144     '-----FORMULA DIRECTA ACTUALIZADA 03/10/2016-----
145     tiempo_maquina = (100 * ((cXc) / 3600) / (carga)) * 100
146
147     '-----FORMULA OBSOLETA-----
148     'tiempo_maquina = math.Round((( ( 83.94 + ( ( t_ciclo * d1 ) / 9.062 ) ) * widths ) / 263.16) * 100,3)
149
150     tiempo_mano=tiempo_maquina
151     Catch er As Exception
152         ListaAlertas.Add(Name & " : " & er.Message)
153     End Try
154 End Sub

```

**Ilustración 94 método calcular( ).- CT- 9202.**

### 3.1.6.3 CentroTrabajo9203.vb – Corte de Bobinas (CAMPBELL CUT OFF).

Básicamente en este centro de trabajo solo fue necesario la actualización de la fórmula; para realizar esta actualización tienes que dirigirte al método calcular() del centro de trabajo 9203 y agregar las variables necesarias para realizar los cálculos, llamadas “carga” “t\_ciclo” y “cXc”, todas ellas de tipo double, la variable “t\_ciclo” se iguala a “462.72”. En seguida se realizan las operaciones para obtener el valor de las variables agregadas anteriormente líneas 128 y 129. Después se agrega la nueva fórmula para obtener el tiempo estándar del CT-9203 línea 136. Posterior a ella se encuentran las formulas simplificada línea 140 y obsoleta línea 143 que era la que se utilizaba anteriormente en este centro de trabajo (Ilustración 95).

```
121 Public Sub calcular() Implements ICentroTrabajo.calcular
122     '-----VARIABLES NECESARIAS-----
123     Dim carga As Double
124     Dim cXc As Double
125     Dim t_ciclo As Double = 462.72
126
127     '-----OPERACIONES NECESARIAS PARA REALIZAR EL CALCULO DIRECTO-----
128     carga = ( 6.8 / widths ) - 2
129     cXc = (t_ciclo + 69.02965125 + 303.52533750)
130
131     Try
132         tiempo_septup=buscar_setupin(centro)
133
134     '-----FORMULA DIRECTA ACTUALIZADA 03/10/2016-----
135
136         tiempo_maquina = (100 * ((cXc) / 3600)/(carga)) * 100
137
138     '-----FORMULA SIMPLIFICADA ACTUALIZADA-----
139
140         'tiempo_maquina = math.Round((835.27/(36*((6.8/widths)-2)))*100,3)
141
142     '-----FORMULA OBSOLETA-----
143         '((( ( 239.53 + 380 ) * widths ) / 244.8) * 100,3)
144
145         tiempo_mano=tiempo_maquina
146     Catch er As Exception
147         ListaAlertas.Add(Name & ": " & er.Message)
148     End Try
149 End Sub
```

Ilustración 95 método calcular().- CT- 9203.

### 3.1.6.4 CentroTrabajo9204.vb – STRESS REL&HEAT SHAP#1 Y STRESS REL&HEAT SHAP#2.

Básicamente en este centro de trabajo fue necesario actualizar las formulas y los tiempos ciclo con los que contaba el código en el método calcular (), se agregaron las recetas con las que no contaba el centro de trabajo y se agregaron las siguientes variables “carga” y “cXc”, de tipo double, líneas 141 y 142 (Ilustración 96).

```
140     '-----VARIABLES NECESARIAS-----
141     Dim carga As Double
142     Dim cXc As Double
```

Ilustración 96 Variables necesarias.- CT- 9204.

Ya que los nuevos tiempos son más eficientes se actualizaron quedando de la siguiente manera: (Ilustraciones 97 y 98).

```

145 tiempo_septup=buscar_setupin(centro)
146 Dim t_ciclo_horno As Double
147 If _shap1 = True Then
148   If _Receta = 1 Then
149     t_ciclo_horno = 7200
150   ElseIf _Receta = 2
151     t_ciclo_horno = 10800
152   ElseIf _Receta = 3
153     t_ciclo_horno = 10800
154   ElseIf _Receta = 4
155     t_ciclo_horno = 5400
156   ElseIf _Receta = 5
157     t_ciclo_horno = 10800
158   ElseIf _Receta = 6
159     t_ciclo_horno = 9000
160   ElseIf _Receta = 7
161     t_ciclo_horno = 3600
162   ElseIf _Receta = 8
163     t_ciclo_horno = 0
164   Else
165     msgbox("Receta no reconocida, Imposible calcular!!!",MsgBoxStyle.Critical,"Ing. de Rutas")
166     me.ListaAlertas.Add(Name & "No. de reseta no reconocida, verificar con Ing. Industrial")
167     Exit Sub
168   End If

```

**Ilustración 97 Nuevos tiempos ciclo en Recetas.- CT- 9204.**

```

179 Else
180   If _Receta = 1 Then
181     t_ciclo_horno = 7200
182   ElseIf _Receta = 2
183     t_ciclo_horno = 10800
184   ElseIf _Receta = 3
185     t_ciclo_horno = 5400
186   ElseIf _Receta = 4
187     t_ciclo_horno = 5400
188   ElseIf _Receta = 5
189     t_ciclo_horno = 10800
190   ElseIf _Receta = 6
191     t_ciclo_horno = 9000
192   ElseIf _Receta = 7
193     t_ciclo_horno = 3600
194   ElseIf _Receta = 8
195     t_ciclo_horno = 0
196   Else
197     msgbox("Receta no reconocida, Imposible calcular!!!",MsgBoxStyle.Critical,"Ing. de Rutas")
198     me.ListaAlertas.Add(Name & "No. de reseta no reconocida, verificar con Ing. Industrial")
199     Exit Sub
200   End If

```

**Ilustración 98 Nuevos tiempos ciclo en Recetas.- CT- 9204.**

Posterior a estos pasos se procede a realizar las operaciones para obtener el valor de las variables “carga” y “cXc” ya que son muy importantes para realizar la operación del tiempo estándar, enseguida se actualiza la fórmula para calcular el tiempo estándar para cuando selecciona la casilla del shap1 (Ilustración 99).

```

169 '-----OPERACIONES NECESARIAS PARA REALIZAR EL CALCULO DIRECTO-----
170     carga = ((6.5/ widths) * 6)
171     cXc = ( t_ciclo_horno + 7.04 + 291.76126667 + 5394.10)
172
173 '-----FORMULA DIRECTA ACTUALIZADA 03/10/2016-----
174     tiempo_maquina = ( 100 * ((cXc) / 3600)/(carga)) * 100
175
176 '-----FORMULA SIMPLIFICADA ACTUALIZADA-----
177     'tiempo_maquina = math.Round((( 5692.9013 + t_ciclo_horno ) / 36 ) / ( ( 6.5 / widths ) * 6 )) * 100,3)
178

```

**Ilustración 99 Operaciones necesarias y nueva fórmula de tiempo estándar.- CT-9204.**

En caso de no seleccionar el shap1 se procede a realizar los siguientes cálculos que son para cuando selecciona shap2 en la interface gráfica. De igual manera que cuando selecciona shap1 se realizan las operaciones para obtener el valor de las variables “carga” y “cXc” ya que son muy importantes para realizar la operación del tiempo

estándar, enseguida se actualiza la fórmula para calcular el tiempo estándar para cuando selecciona la casilla del shap2. En las líneas 178 y 211 se encuentran las formulas simplificadas para la actualización, pero para evitar el error con los decimales se optó por utilizar la formula directa (Ilustraciones 99 y 100).

```

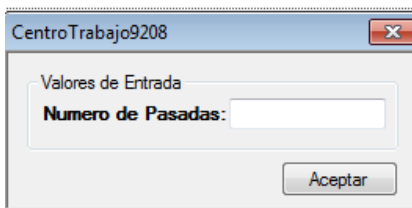
201 | .....-OPERACIONES NECESARIAS PARA REALIZAR EL CALCULO DIRECTO-.....
202 |
203 |         carga = ((6.8 / widths) * 6)
204 |         cXc = (t_ciclo_horno + 7061.98 + 4.8680 + 443.0146277778 + 65.1283333333)
205 |
206 | .....-FORMULA DIRECTA ACTUALIZADA 03/10/2016-.....
207 |         tiempo_maquina =( 100* ((cXc) / 3600) / (carga)) * 100
208 |
209 | .....-FORMULA SIMPLIFICADA ACTUALIZADA-.....
210 |
211 |         'tiempo_maquina =math.Round(((( 7574.99 + t_ciclo_horno ) / 36 ) / ( ( 6.8 / widths ) * 6 )) * 100,3)
212 |

```

**Ilustración 100 Operaciones necesarias y nueva fórmula de tiempo estándar.- CT- 9204.**

### 3.1.6.5 CentroTrabajo9208.vb – Stack Size LB.

Para realizar la actualización de este centro de trabajo fue necesario cambiar la interface gráfica del mismo añadiendo un nuevo campo para que pueda tomar los valores ingresados por el usuario ya que anteriormente el sistema solo realizaba el cálculo directo sin pedir ningún valor de entrada. Para realizar esto te diriges al CT que deseas modificar, te diriges a la pestaña de diseño y después agregas un groupBox, un label, un TextBox y un botón. Cambias la propiedad text de cada uno de ellos por el texto que deseas que se muestre en la interface gráfica del centro de trabajo.



**Ilustración 101 Nueva Interface gráfica.- CT- 9208.**

Después de agregar los nuevos campos en el form será necesario pasar a la codificación; para esto te diriges a la pestaña Código y buscas el método Sub “pasarValoresDeListas” para agregar la variable que almacenara el valor ingresado por el usuario (Ilustración 102).

```

90 | Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
91 |     NPasadas = Module1.obtenerValorDatoDouble(lDatos,"NumeroDePasadas")
92 |
93 | End Sub

```

**Ilustración 102 método Sub pasarValoresDeListas.- CT- 9208.**

Después en el método sub se declara la variable que añade una descripción de que esta almacenando en un listado de datos requeridos (Ilustración 103).



```

95 | Public Sub New()
96 |     Me.InitializeComponent()
97 |
98 |     Dim NumeroDePasadas As New Dato("NumeroDePasadas", "Distance", "Numero de pasadas", Nothing, "Numero de Pasadas:")
99 |     listaDatosRequeridos.Add(NumeroDePasadas)
100 | End Sub

```

**Ilustración 103 método Sub New( ).- CT- 9208.**

Se declara la nueva variable del tipo Double que almacena el valor ingresado por el usuario para su posterior uso en los cálculos; (Ilustración 104).

```

102 | '-----variables que se necesitan-----
103 | Dim NPasadas As Double
104 | '-----

```

**Ilustración 104 Nuevas Variables.- CT- 9208.**

En el método calcular se agrega la operación o fórmula matemática que realizara el cálculo para obtener el tiempo máquina línea 122 (Ilustración 105).

```

119 | Public Sub calcular() implements ICentroTrabajo.calcular
120 |     Try
121 |         tiempo_septup=buscar_setupin(centro)
122 |         tiempo_maquina = Math.Round((5.9218+(3.705458333*NPasadas))/36*100,3)
123 |         tiempo_mano=tiempo_maquina
124 |     Catch er As Exception
125 |         Me.ListaAlertas.Add(Name & ": " & er.Message)
126 |     End Try
127 |
128 | End Sub

```

**Ilustración 105 método calcular( ).- CT- 9208.**

A la función clic del botón se le agrego un if para condicionar que el valor ingresado por el usuario tiene que ser un numero de no ser así manda un mensaje de alerta, en caso contrario procede a realizar la operaciones en el método calcular( ) (Ilustración 106).

```

141 | Sub Button1Click(sender As Object, e As EventArgs)
142 |     If IsNumeric(textBox1.Text) Then
143 |         NPasadas = CDec(Me.textBox1.Text)
144 |         calcular()
145 |         Close()
146 |     Else
147 |         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
148 |     End If
149 | End Sub
150 |

```

**Ilustración 106 función clic del botón aceptar.- CT- 9208.**

### 3.1.6.6 CentroTrabajo9210.vb – Visual Inspection L.B.

Prácticamente en este centro de trabajo se tuvo que actualizar la fórmula matemática que está en uso en sistema, para ello te tienes que dirigir al método calcular ( ) del centro de trabajo 9210. Una vez ahí procedes a agregar las variables necesarias para realizar los cálculos, se crean dos nuevas variables de tipo double llamadas “carga” y “cXc” que serán utilizadas para almacenar valores resultantes de operaciones matemáticas para su posterior uso en la nueva fórmula para calcular el tiempo

estándar. En las líneas 126 y 127 se agregaron las fórmulas para obtener el valor necesario de la variable para su posterior uso. En la línea 133 se agrega la nueva fórmula directa para calcular el tiempo estándar ya que con la simplificada variaba el resultado en décimas. En la línea 137 se puede apreciar la formula simplificada, y en la línea 141 se puede ver la formula obsoleta (Ilustración 107).

```

121 Public Sub calcular() Implements ICentroTrabajo.calcular
122 '-----VARIABLES QUE SE NECESITAN-----
123 Dim carga As Double
124 Dim cXc As Double
125 '-----OPERACIONES PARA REALIZAR EL CALCULO DIRECTO-----
126 carga = (2.5/_h1)
127 cXc = (43.43434667 + 4.903290 + 25.515)
128
129
130 tiempo_septup=buscar_setupin(centro)
131
132 '-----FORMULA DIRECTA ACTUALIZADA 0/09/2016-----
133 tiempo_maquina = (100 * ((cXc) / 3600) / (carga)) * 100
134
135 '-----FORMULA SIMPLIFICADA ACTUALIZADA-----
136
137 *tiempo_maquina =(73.85263667/((2.5/_h1)*36))*100
138
139
140 '-----FORMULA OBSOLETA-----
141 '((( ( 59.91 + ( ( 62.03 * _h1 ) / 0.1865 ) ) * _h1 ) / 114.138) * 100,3)
142 tiempo_mano=tiempo_maquina
143 *tiempo_maquina = 0
144 End Sub

```

**Ilustración 107 método calcular().- CT- 9210.**

### 3.1.6.7 CentroTrabajo9215.vb – Rough Grind.

Para la actualización de este centro de trabajo fue necesario modificar la interface gráfica del form añadiendo un Label y un ComboBox para ingresar los nuevos datos que necesita saber el sistema para realizar las operaciones y cálculos matemáticos para la obtención del tiempo máquina. Después se decidió ocultar los campos que ya no se van a utilizar para la obtención de tiempo máquina.

**Ilustración 108 nueva interface gráfica.- CT- 9215.**

En el código se tuvo que modificar agregando en la función Test( ) del CT- 9215 la variable \_material que obtiene de los procesos en el modulo1 el material y el tipo del que está hecho. También se agregó un condicionamiento if para saber qué tipo de material es seleccionado y es caso de no ser uno de los que se ponen en la condición mandara un mensaje de alerta indicando que el material no está disponible para trabajar en este centro de trabajo. Antes de hacer esto se declararon las variables “banElementoFijo” y “tipo” (Ilustración 109).

```

86 | Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean Implements 1
87 |     _material = Module1.obtenerValorDatoString(lDatosString,"MATERIAL")
88 |     tipo = Module1.sacar_tipo_material(_material)
89 |     If tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Or tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO GRIS INTERMEDIO"
90 |     Else
91 |         MsgBox("Material " & tipo & " no disponible para el C.C " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
92 |         "Por favor seleccione un tipo de material disponible para poder calcular el centro de trabajo",MsgBoxStyle.Critical,"Ing. de Rutas")
93 |         Dim listaOpcionales As New List(Of String)
94 |         listaOpcionales.Add("HIERRO GRIS")
95 |         listaOpcionales.Add("ACERO INOXIDABLE")
96 |         listaOpcionales.Add("ACERO AL CARBON")
97 |         listaOpcionales.Add("HIERRO DUCTIL")
98 |         listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
99 |         listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
100 |         Dim f As New Seleccionar(listaOpcionales)
101 |         If f.ShowDialog() = DialogResult.OK Then
102 |             tipo = f.elementoSeleccionado
103 |             banElementoFijo = True
104 |         End If
105 |         Exit Function
106 |     End If
107 |     Return True
108 | End Function
109 |
110 | Dim tipo As String
111 | Dim banElementoFijo As Boolean = False

```

**Ilustración 109 función Test( ).- CT- 9215.**

En el método sub “pasarValoresDeListas” se agregó la condición if para comparar que la variable “banElementoFijo no sea false, de ser así cargara el ComboBox con el listado de materiales que se pueden utilizar (Ilustración 110).

```

113 | Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBox
114 |
115 |     widths = module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")
116 |     diametro = module1.obtenerValorDatoDouble(lDatos,"diaNominalAnillo")
117 |     _pasadas = module1.obtenerValorDatoDouble(lDatos,"pasadas9215")
118 |     If banElementoFijo = False Then
119 |         _material = Module1.obtenerValorDatoString(lDatosString,"MATERIAL")
120 |     End If
121 | End Sub

```

**Ilustración 110 método pasarValoresDeListas( ).- CT- 9215.**

Se agregó el método llenar\_material() que se encarga de cargar los materiales al comboBox de la interface (Ilustración 111).

```

218 | Private Sub llenar_material()
219 |     Dim l As New List(Of String)
220 |     l.Add("ACERO INOXIDABLE")
221 |     l.Add("ACERO AL CARBON")
222 |     l.Add("HIERRO GRIS")
223 |     l.Add("HIERRO GRIS CENTRIFUGADO")
224 |     l.Add("HIERRO GRIS INTERMEDIO")
225 |     l.Add("HIERRO DUCTIL")
226 |     For Each m In obtenerEspecTipoMaterial(l)
227 |         cbo_material.Items.Add(m)
228 |     Next
229 | End Sub

```

**Ilustración 111 método llenar\_material( ).- CT- 9215.**

En el método sub new se agrega la variable material\_ de tipo DatoString que almacena una breve descripción de la variable para después agregarlo a la lista de datos requeridos para la operación (Ilustración 112).

```

123 Public Sub New()
124     Me.InitializeComponent()
125     Dim pasadas9215 As New Dato("pasadas9215","Cantidad","Indica el numero de pas
126     listaDatosRequeridos.Add(pasadas9215)
127
128     Dim widthNominalAnillo As New Dato("widthNominalAnillo","Distance","Width nor
129     listaDatosRequeridos.Add(widthNominalAnillo)
130
131
132     Dim diaNominalAnillo As New Dato("diaNominalAnillo","Distance","Diámetro nomi
133     listaDatosRequeridos.Add(diaNominalAnillo)
134
135     Dim material_ As New DatoString("MATERIAL","MATERIAL Componente","Material:")
136     listaDatosStringRequeridos.Add(material_)
137
138 End Sub

```

**Ilustración 112 método Sub New( ).- CT- 9215.**

Después pasamos al método calcular( ) donde se agregaran las variables necesarias para realizar los cálculos línea 168 y 169 ilustración 113, también se agregara un if condicionante para que dependiendo del material utilizado tome el valor de la variable t\_ciclo según corresponda al tipo de material línea 177 a 192 ilustración 113.

```

167 Public Sub calcular() Implements ICentroTrabajo.calcular
168     Dim t_ciclo As Double
169     Dim tct As Double
170     Dim CxC As Double
171
172     Try
173         tiempo_septup=buscar_setupin(centro)
174         If banElementoFijo = False Then
175             tipo = Module1.sacar_tipo_material(_material)
176         End If
177         If tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Then
178             t_ciclo = 1.80
179
180             ElseIf tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
181                 t_ciclo =2.5098241667
182
183             ElseIf tipo = "HIERRO GRIS INTERMEDIO" Then
184                 t_ciclo = 2.6683066667
185
186             ElseIf tipo = "HIERRO DUCTIL"Then
187                 t_ciclo = 1.9151761538
188
189             Else
190                 MsgBox("Material " & tipo & " no disponible para el C.C 9215, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
191             Exit Sub
192         End If

```

**Ilustración 113 método calcular( ), nuevas variables y condicionante IF.- CT- 9215.**

Posterior a esto se procedió a agregar las operaciones matemáticas para obtener el valor de las variables anteriormente agregadas líneas 194 y 195 (Ilustración 114) Enseguida en la línea 196 se agregó la fórmula matemática directa para calcular el tiempo estándar. En la línea 197 se encuentra la formula actualizada simplificada, y en la línea 198 la formula obsoleta.

```

193
194     tct = (t_ciclo * _pasadas)
195     CxC = (tct + 2.82644306 +0.83863656)
196     tiempo_maquina =(100 * ((CxC) / 3600)/1) * 100
197     'tiempo_maquina = math.Round((3.67+(t_ciclo*_pasadas))/36*100,3)
198     '((( ( 27.84 + ( ( ( 25.664 * diametro ) * _pasadas ) ) * widths ) / 432) * 100,3)
199     tiempo_mano=tiempo_maquina
200 Catch er As Exception
201     me.ListaAlertas.Add(Name & ": " & er.Message)
202 End Try
203
204 End Sub

```

**Ilustración 114 Nuevas operaciones método calcular( ).- CT- 9215.**

En la función clic del botón se modificó quitando del condicionamiento los dos textbox que se ocultaron y agregando una comparación para que el campo de material no este vacío de ser así procederá a realizar las operaciones del método calcular del CT.9215.

```

216 Sub Button1Click(sender As Object, e As EventArgs)
217     If comboBox1.Text <> "" And cbo_material.Text <> "" Then 'and isnumeric(Me.textBox2.Text) and isnumeric(me.textBox3.Text) Then
218         'Me.widths = CDec(Me.textBox1.Text)
219         'Me.diametro = CDec(Me.textBox2.Text)
220         Me._pasadas = CInt(Me.comboBox1.Text)
221         _material = Me.cbo_material.Text
222         Me.calcular()
223         Me.Close()
224     Else
225         msgbox("Revisar valores", MsgBoxStyle.Exclamation, "Ing. de Rutas")
226     End If
227 End Sub

```

**Ilustración 115 función clic del botón aceptar.- CT- 9215.**

Para terminar, se agregó en el evento load del CT-9215 el código para ocultar los elementos de la interface que ya no se van a utilizar (Ilustración 116).

```

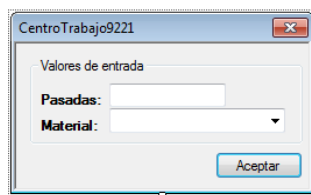
231 Sub CentroTrabajo9215Load(sender As Object, e As EventArgs)
232     textBox1.Visible = False
233     textBox2.Visible = False
234     label1.Visible = False
235     label2.Visible = False
236 End Sub
237 End Class

```

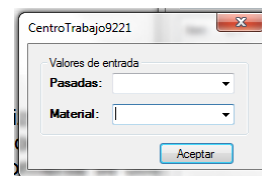
**Ilustración 116 Evento Load( ).- CT- 9215.**

### 3.1.6.8 CentroTrabajo9221.vb – Side Grind Besly.

En este centro de trabajo se realizó una pequeña modificación a la interface gráfica por estética se alinearon y acomodaron del mismo tamaño los ComboBox para que tuviese una mejor vista.



**Ilustración 117 form antes de la actualización.- CT- 9221.**



**Ilustración 118 form actualizado.- CT- 9221.**

En este centro de trabajo se actualizaron los valores de tiempo ciclo por material que se almacenan en la variable t\_ciclo dentro del método calcular() del CT-9221 por los siguientes como se muestra en la (Ilustración 119), los tiempos que anteriormente se utilizaban se muestran de color verde a un costado del t\_ciclo que actualmente se estará usando.

Dentro de este método también se agregaron nuevas variables llamadas “tcf” y “CxC” de tipo double líneas 159 y 160 ilustración 119.

```

156 Public Sub calcular() implements ICentroTrabajo.calcular
157     tiempo_septup=buscar_setupin(centro)
158     Dim t_ciclo As Double
159     Dim tcf As Double
160     Dim CxC As Double
161     If banElementoFijo = False Then
162         tipo = Module1.sacar_tipo_material(_material)
163     End If
164     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
165         t_ciclo = 3.165921'1.612
166     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
167         t_ciclo = 2.79675'2.569
168     ElseIf tipo = "ACERO INOXIDABLE" OR tipo = "ACERO AL CARBON"
169         t_ciclo = 2.467355'2.426
170     ElseIf tipo = "HIERRO DUCTIL"
171         t_ciclo = 3.368530'2.065
172     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
173         t_ciclo = 2.3347683333'2.850
174     Else
175         msgbox("Material " & tipo & " no disponible para el C.C 9221, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
176         Me.ListaAlertas.Add(Name & " El material " & tipo & " no esta disponible para el calculo de tiempos estandar, verificar con Ing. In
177         Exit Sub
178     End If
179     tcf = (t_ciclo * _Pasadas)
180     CxC = (tcf + 1.0625084523 + 3.2493806099)
181     tiempo_maquina = (100 * ((CxC)/3600))*100
182     'tiempo_maquina = math.Round(((4.31+(t_ciclo*_Pasadas))/36)*100,3)
183     '((( 2.56081 + ( t_ciclo * _Pasadas ) ) / 36) * 100,3)
184     tiempo_mano=tiempo_maquina * factorLabor
185 End Sub

```

**Ilustración 119 método calcular( ).- CT- 9221.**

En las líneas 179 y 180 se encuentran las operaciones matemáticas necesarias para obtener el valor de la variable anteriormente descrita.

También se actualizo la fórmula que se utilizaba por una más eficiente, como se puede apreciar en la ilustración 119 la nueva fórmula está en la línea 181 y debajo de esta misma se encuentra la formula simplificada actualizada y debajo de esta la obsoleta que anteriormente se utilizaba para la obtención del tiempo máquina.

### 3.1.6.9 CentroTrabajo9222.vb – Band Rings & Unband LB.

Prácticamente la actualización a este centro de trabajo se llevó a cabo en el método calcular( ) donde solo se actualizaron las fórmulas que utilizaba el CT-9222 por las nuevas como se muestra en la Ilustración 120 las fórmulas que anteriormente utilizaba el sistema se muestran de color verde, debajo de las que actualmente quedaran en uso en las líneas 128 y 131.

```

124 Public Sub calcular() implements ICentroTrabajo.calcular
125     Try
126         tiempo_septup=buscar_setupin(centro)
127         If(_colocarBanda=True) Then
128             tiempo_maquina = math.Round(.140913*100,5)
129             '( .10544*100,3)
130         Else
131             tiempo_maquina = math.Round(.124801*100,5)
132             '( .149052*100,3)

```

**Ilustración 120 método calcular( ).- CT-9222.**

### 3.1.6.10 CentroTrabajo9223.vb – 6 NOTCH.

Para realizar la actualización de este centro de trabajo fue necesario quitar elementos de la lista de materiales que se pueden utilizar en el centro de trabajo y actualizar los tiempos ciclo de cada uno de estos materiales. En la función Test se quitaron de la condicionante if los siguientes elementos: “ACERO INOXIDABLE”, “ACERO AL CARBON”, “HIERRO GRIS ALTO MODULO” que son elementos que ya no se utilizan en la operación a realizar ni en el CT-9223. También se quitan de la lista de opciones que se despliega (Ilustración 121).

```
90 Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean Implements
91 _material = module1.obtenerValorDatoString(lDatosString,"MATERIAL")
92 tipo = Module1.sacar_tipo_material(_material)
93
94 If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO DUCTIL" Or tipo = "HIERRO GRIS INTERMEDIO" Then
95
96 Else
97 MsgBox("Material " & tipo & " no disponible para el C.C " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
98 "Por favor seleccione un tipo de material disponible para poder calcular el centro de trabajo",MsgBoxStyle.Critical,"Ing. de Rutas")
99 Dim listaOpcionales As New List(Of String)
100 listaOpcionales.Add("HIERRO GRIS")
101 listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
102 listaOpcionales.Add("HIERRO DUCTIL")
103 listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
104
105 Dim f As New Seleccionar(listaOpcionales)
106 If f.ShowDialog() = DialogResult.OK Then
107 tipo = f.elementoSeleccionado
108 banElementoFijo = True
109 End If
110 Exit Function
111 End If
112 Return True
113 End Function
```

Ilustración 121 función Test( ).- CT- 9223.

Después de esto se procede a actualizar los valores de la variable t\_ciclo de cada material utilizado en este centro de trabajo y a quitar los materiales que ya no son requeridos dentro del método calcular () en seguida se actualiza la fórmula que almacena la variable tiempo\_maquina para la realización de los cálculos y obtención del tiempo estándar (Ilustración 122).

```
146 Public Sub calcular() implements ICentroTrabajo.calcular
147 tiempo_septup=buscar_setupin(centro)
148 Dim t_ciclo As Double
149 If banElementoFijo = False Then
150 tipo = Module1.sacar_tipo_material(_material)
151 End If
152 If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
153 t_ciclo = 14.4600
154 ElseIf tipo = "HIERRO GRIS INTERMEDIO"
155 t_ciclo = 16.3082*15.52
156 'ElseIf tipo = "HIERRO GRIS ALTO MODULO"
157 t_ciclo = 14.80
158 ElseIf tipo = "HIERRO DUCTIL"
159 t_ciclo = 16.3082*15.52
160 'ElseIf tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON"
161 t_ciclo = 15.52
162 Else
163 MsgBox("Material " & tipo & " no disponible para el C.C 9223, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
164 Exit Sub
165 End If
166 tiempo_maquina = math.Round(((9.4668+t_ciclo)/36)*100,3)
167 '((( 28.5961 + t_ciclo) / 36) * 100,3)
168 tiempo_mano=tiempo_maquina
169 End Sub
```

Ilustración 122 método calcular( ).- CT- 9223.

Nota en la imagen anterior se pueden ver de color verde los valores que anteriormente utilizaban, así como la antigua formula línea 167 y los materiales que ya no utiliza.

### 3.1.6.11 CentroTrabajo9224.vb – Counterbore-Southben LB.

Prácticamente para la actualización en este centro de trabajo fue necesario modificar la interface gráfica del CT-9224, para esto se agregó un label y un comboBox, también se ocultaron los objetos que ya no son utilizados en el centro de trabajo para calcular el tiempo estándar.

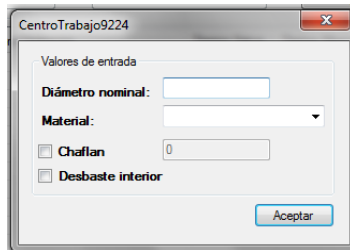


Ilustración 124 form antes de la actualización.- CT- 9224.

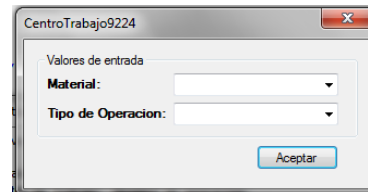


Ilustración 123 form actualizado.- CT- 9224.

Después en la función test se agrega la variable “op” para obtener el nombre de la operación a realizar y se agrega un if para determinar si selecciono una operación, en caso contrario manda un mensaje de alerta. Se eliminan los materiales que no se utilizaran más en este centro de trabajo de la condicionante if al igual que de la lista de materiales a cargar (Ilustración 125).

```
86 Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean Implemer
87     _material = module1.obtenerValorDatoString(lDatosString,"MATERIAL")
88     tipo = Module1.sacar_tipo_material(_material)
89     oper = Module1.obtenerValorDatoString(lDatosString,"OPERACION")
90     op = Module1.obtener_nombre_operacion(oper)
91     If op = "CHAFLÁN" Or op = "BEVEL LADO DEL DASH" Or op = "BEVEL DOS LADOS" Then
92     Else
93         MsgBox("No seleccionaste ninguna operacion")
94     End If
95     If tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Or tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO DUCTIL" Then
96     Else
97         MsgBox("Material " & tipo & " no disponible para el C.C " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" &
100         "Por favor seleccione un tipo de material disponible para poder calcular el centro de trabajo",MsgBoxStyle.Critical,"Ing. de Rutas")
101         Dim listaOpcionales As New List(Of String)
102         listaOpcionales.Add("HIERRO GRIS")
103         listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
104         listaOpcionales.Add("HIERRO DUCTIL")
105         listaOpcionales.Add("ACERO INOXIDABLE")
106         listaOpcionales.Add("ACERO AL CARBON")
107         Dim f As New Seleccionar(listaOpcionales)
108         If f.ShowDialog() = DialogResult.OK Then
109             tipo = f.elementoSeleccionado
110             banElementoFijo = True
111         End If
112         Exit Function
113     End If
114     Return True
115 End Function
```

Ilustración 125 función Test( ).- CT-9224.

Antes de realizar el paso anterior es necesario declarar la nueva variable llamada “op” de tipo string, como se muestra en la ilustración 126.



```

L69: '-----variables que se necesitan-----
L70: Dim _d1 As Double
L71: Dim _material As String
L72: Dim banChaflan As Boolean
L73: Dim MedidaChaflan As Double
L74: Dim _DesbasteInterior As Boolean
L75: Dim _width As Double
L76: Dim oper As String
L77: Dim op As String
L78: Dim a_cicl As Double
L79: '-----

```

**Ilustración 126 Variables necesarias.-  
CT- 9224.**

Posterior a los pasos anteriores hay que buscar el método calcular () del CT-9224. Una vez ahí hay que declarar dos nuevas variables llamadas “t\_ciclo” y “cXc” de tipo doublé, que serán utilizadas para realizar el cálculo del tiempo estándar. Se actualizan los tiempos ciclo y las actividades cíclicas que están almacenadas en las siguientes variables “t\_ciclo” y “a\_cicl” respectivamente para cada tipo de material, Ilustración 127.

```

195: Public Sub calcular() implements ICentroTrabajo.calcular
196: tiempo_septup=buscar_setupin(centro)
197: Dim t_ciclo As Double
198: Dim cXc As Double
199:
200: If op = "CHAFLÁN" Or op = "BEVEL LADO DEL DASH" Or op = "BEVEL DOS LADOS" Then
201: Else
202: MsgBox("No seleccionaste ninguna operacion")
203:
204: End If
205:
206: If banElementoFijo = False Then
207: tipo = Module1.sacar_tipo_material(_material)
208: End If
209: If op = "BEVEL LADO DEL DASH" And tipo = "HIERRO GRIS" Then
210: t_ciclo = 14.17*9.6
211: a_cicl = 10.02294490
212: ElseIf op = "BEVEL LADO DEL DASH" And tipo = "ACERO INOXIDABLE"
213: t_ciclo = 0.00
214: a_cicl = 10.02294490
215: ElseIf op = "BEVEL LADO DEL DASH" And tipo = "ACERO AL CARBON"
216: t_ciclo = 0.00
217: a_cicl = 10.02294490
218: ElseIf op = "CHAFLÁN" And tipo = "ACERO INOXIDABLE"
219: t_ciclo = 10.95*12.025
220: a_cicl = 16.55694490
221: ElseIf op = "CHAFLÁN" And tipo = "ACERO AL CARBON"
222: t_ciclo = 10.95*12.025
223: a_cicl = 16.55694490
224: ElseIf op = "CHAFLÁN" And tipo = "HIERRO DUCTIL"
225: t_ciclo = 12.24*9.6
226: a_cicl = 16.55694490
227: ElseIf op = "BEVEL DOS LADOS" And tipo = "HIERRO GRIS"
228: t_ciclo = 12.32 *12.05
229: a_cicl = 16.55694490
230: ElseIf op = "BEVEL DOS LADOS" And tipo = "HIERRO DUCTIL"
231: t_ciclo = 10.35
232: a_cicl = 16.55694490
233: ElseIf op = "BEVEL DOS LADOS" And tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON"
234: t_ciclo = 10.96
235: a_cicl = 16.55694490
236: Else
237: msgbox("Tipo de Material y Operación No Compatibles,imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
238: listaAlertas.Add(Name & ":" & "El material " & tipo & " no esta disponible en el calculo,varificar con Ing. Industrial")
239: Exit Sub
240: End If

```

**Ilustración 127 método calcular().- CT- 9224.**

Enseguida se agrega la nueva fórmula estandarizada para realizar el cálculo línea 255, ilustración 128.

```

255: cXc = ((a_cicl + 2.37335510 + t_ciclo) / 36)
256: tiempo_maquina = cXc * 100
257:

```

**Ilustración 128 formula actualizada.- CT-  
9224.**

En la función clic del botón aceptar se agregó a la condicionante if que el comboBox sea diferente de cero y se igualo la variable “op” para que tome el valor ingresado en el comboBox1 (Ilustración 129).

```

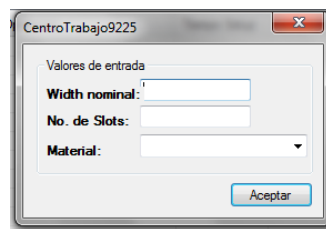
281| Sub Btn_aceptarClick(sender As Object, e As EventArgs)
282|     If cbo_material.Text <> "" And comboBox1.Text <> "" Then 'And Isnumeric(Me.tbWidth.Text) Then
283|
284|         _material = Me.cbo_material.Text
285|         op = CStr(Me.comboBox1.Text)
286|
287|         calcular()
288|         close()
289|     Else
290|         MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
291|     End If
292| End Sub

```

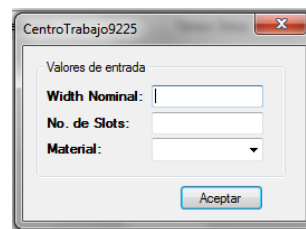
**Ilustración 129 función clic del botón aceptar.- CT-9224.**

### 3.1.6.12 CentroTrabajo9225.vb – Groove Slot LB.

En este centro de trabajo se realizó una pequeña modificación a la interface gráfica, alineando los textBox y el comboBox del mismo tamaño solo para darle más estética al sistema.



**Ilustración 131 form antes de la actualización.- CT-9225.**



**Ilustración 130 form actualizado.- CT-9225.**

Posterior a la modificación de la interface gráfica se procede a buscar la función test del CT-9225, en la cual hay que eliminar de la condicionante If y de la lista de materiales el “HIERRO GRIS ALTO MODULO” ya que para la nueva fórmula actualizada no se utiliza este material en el centro de trabajo (Ilustración 132).

```

86| Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean Imple
87|     _material = module1.obtenerValorDatoString(lDatosString,"MATERIAL")
88|     tipo = Module1.sacar_tipo_material(_material)
89|
90|     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO DUCTIL" Or tipo = "HIERRO GRIS INTERMEDIO" Then
91|
92|     Else
93|         MsgBox("Material " & tipo & " no disponible para el C.C " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "
94|         "Por favor seleccione un tipo de material disponible para poder calcular el centro de Trabajo",MsgBoxStyle.Critical,"Ing. de Rutas")
95|         Dim listaOpcionales As New List(Of String)
96|         listaOpcionales.Add("HIERRO GRIS")
97|         listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
98|         listaOpcionales.Add("HIERRO DUCTIL")
99|         'listaOpcionales.Add("HIERRO GRIS ALTO MODULO")
100|         listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
101|         Dim f As New Seleccionar(listaOpcionales)
102|         If f.ShowDialog() = DialogResult.OK Then
103|             tipo = f.elementoSeleccionado
104|             banElementoFijo = True
105|         End If
106|         Exit Function
107|     End If
108|     Return True
109| End Function

```

**Ilustración 132 función Test( ).- CT-9225.**

Fue necesario agregar una nueva variable llamada “slotsXmat” para almacenar el número de ventilas por tipo de material en el método calcular( ), para ello te diriges al método calcular del CT-9225 y declaras las nuevas variables. En seguida precedes a actualizar los tiempos ciclo y a modificar la fórmula matemática almacenada en la variable tiempo\_maquina (Ilustración 133).

```

161 Public Sub calcular() Implements ICentroTrabajo.calcular
162     Dim slotsXmat As Double
163     Dim carga As Double
164     Dim cXc As Double
165     Dim tc As Double
166
167     tiempo_septup=buscar_setupIn(centro)
168     Dim t_ciclo As Double
169     If banElementoFijo = False Then
170         tipo = Module1.sacar_tipo_material(_material)
171     End If
172     If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Then
173         t_ciclo = 97.93'121.97
174         slotsXmat = 18
175         tc = ((t_ciclo * _slots)/slotsXmat)
176     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
177         t_ciclo = 90.47'90.48
178         slotsXmat = 18
179         tc = ((t_ciclo * _slots)/slotsXmat)
180     ElseIf tipo = "HIERRO DUCTIL"
181         t_ciclo =82.943333'90.48
182         slotsXmat = 12
183         tc = ((t_ciclo * _slots)/slotsXmat)
184     'ElseIf tipo = "HIERRO GRIS ALTO MODULO"
185         t_ciclo = 43.39
186     Else
187         MsgBox("Material " & tipo & " no disponible para el C.C 9225, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
188     Exit Sub
189     End If
190
191     carga = Math.Round(2.495 / _h1,0)
192     cXc = (tc + 135.36126867)
193     tiempo_maquina = ( 100 * ((cXc) / 3600) / (carga) ) * 100
194     'tiempo_maquina = (((((t_ciclo*_slots)/slotsXmat)+135.36126867)*_h1)/89.82*100)
195     '((( 96.51704 + ( ( t_ciclo * _slots ) / 18 ) ) / ( ( ( 2.75 / _h1 ) -1 ) * 36 )) ) * 100,3)
196     tiempo_mano=tiempo_maquina
197 End Sub

```

**Ilustración 133 método calcular( ).- CT- 9225.**

### 3.1.6.13 CentroTrabajo9241.vb – Moly Coating L.B.

Para la actualización de este centro de trabajo fue necesario actualizar la operación matemática que almacena la variable tiempo\_maquina del método calcular( ) del CT-9241, además de actualizar la formula también se tuvo que actualizar el valor del blast y blowoff de la condicionante if de la función clic del botón aceptar del CT-9241 (Ilustración 135). Por los nuevos valores que se muestran en la siguiente tabla:

Width Nominal	Pasadas		
	Blast	Blowoff	Coating
0.1550	6	10	38
0.1839	6	10	38
0.1850	6	10	38
0.1865	6	10	38
0.2490	6	10	38

**Tabla 5 Valores de Blast y Blowoff**

Nota en la ilustración 134 se pueden apreciar ambas fórmulas, la nueva en la línea 129 y la obsoleta que ya no se usa de color verde en la línea siguiente.

```

127 Public Sub calcular() implements ICentroTrabajo.calcular
128 tiempo_septup=buscar_setupin(centro)
129 tiempo_maquina = math.Round(((17.6357*coating)+35.39)*widths/187.992*100,3)
130 '((( ( 369.39 + ( 23.676 * blast ) + ( 35.56 * blowoff ) + ( 20.28527 * coating ) + 774.27 ) ) * widths ) / 194.4) * 100,3)
131 tiempo_mano=tiempo_maquina
132 Dim m As New MateriaPrimaGenerica
133 m.CodigoMateriaPrima = "0505032"
134 m.Descripcion = "MOLY PLASMA PCP-251 GTP P290737"
135 m.Measurement = "LBS"
136 Dim calo As New CalculoPesoMolyLB(widths,coating)
137 m.Cantidad = math.Round(CDec(calo.txt_peso_sap.Text) * 10,3)
138 ListaMateriaPrima.Add(m)
139
140 End Sub

```

**Ilustración 134 método calcular( ).- CT- 9241.**

```

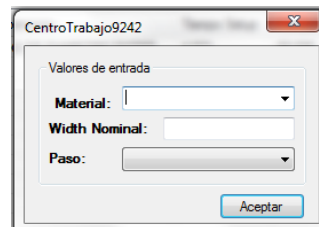
152 Sub Button1Click(sender As Object, e As EventArgs)
153 If isnumeric(Me.textBox1.Text) and isnumeric(Me.textBox2.Text) and isnumeric(Me.textBox3.Text) and isnumeric(Me.textBox4.Text) Then
154 widths= CDec(Me.textBox1.Text)
155 coating = CDec(Me.textBox4.Text)
156 If widths = .155 Then
157 blast =6'4
158 blowoff = 10
159 'coating = 28
160 ElseIf widths = .249
161 blast =6'4
162 blowoff = 10'8
163 'coating = 28
164 ElseIf widths = .1865
165 blast =6
166 blowoff = 10
167 'coating = 34
168 ElseIf widths = .185
169 blast =6
170 blowoff = 10
171 'coating = 52
172 ElseIf widths = .1839
173 blast =6
174 blowoff = 10
175 'coating = 8
176 ElseIf widths = .1865
177 blast =6
178 blowoff = 10
179 'coating = 36
180 End If
181 calcular()
182 Close()
183 Else
184 msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
185 End If
186 End Sub

```

**Ilustración 135 función clic del botón aceptar.- CT- 9241.**

### 3.1.6.14 CentroTrabajo9242.vb – O.D. Sheffield L.B.

Para la actualización se tuvo que modificar la interface gráfica del CT-9242 agregando un label y un TextBox Nuevos para los valores faltantes para realizar los cálculos.



**Ilustración 136 nueva interface gráfica.- CT- 9242.**

Una vez realizado lo anterior es necesario agregar una nueva variable del tipo double para almacenar el valor del Width que ingresara el usuario para la realización de los cálculos necesarios para obtener el tiempo máquina (Ilustración 137).

```

142 | |-----variables que se nesecitan-----
143 | Dim _material As String
144 | Dim _paso As Integer
145 | Dim _h1 As Double
146 | |-----

```

**Ilustración 137 nuevas variables agregadas.- CT- 9242.**

En el método calcular( ) del CT-9242 se actualizaron los valores de la variable t\_ciclo para cada tipo de material en la condicionante IF, se eliminó de la condicionante el tipo de material “HIERRO GRIS ALTO MODULO” para el paso “1” ya que ya no será utilizado en este paso; en el paso “2” solo fue necesario actualizar los valores de la variable t\_ciclo para cada tipo de material en la condicionante IF. En ambos pasos 1 y 2 fue necesaria la actualización de las fórmulas que almacena la variable tiempo\_maquina del CT-9242 (Ilustración 138).

```

162 | Public Sub calcular() implements ICentroTrabajo.calcular
163 |     Try
164 |         tiempo_septup=buscar_setupin(centro)
165 |         Dim t_ciclo As Double
166 |         If banElementoFijo = False Then
167 |             tipo = Module1.sacar_tipo_material(_material)
168 |         End If
169 |         If _paso = 1 Then
170 |             tipo = module1.sacar_tipo_material(_material)
171 |             If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO" Then
172 |                 t_ciclo = 282'238
173 |             ElseIf tipo = "HIERRO GRIS INTERMEDIO"
174 |                 t_ciclo = 428'238
175 |             'ElseIf tipo = "HIERRO GRIS ALTO MODULO"
176 |                 't_ciclo = 238
177 |             ElseIf tipo = "HIERRO DUCTIL"
178 |                 t_ciclo = 235'298
179 |             ElseIf tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON"
180 |                 t_ciclo = 320'298
181 |             Else
182 |                 msgbox("Material " & tipo & " no disponible para el C.C 9242, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
183 |                 Me.ListaAlertas.Add(Name & ": El material & " & tipo & " no se encuentra disponible para el calculo, verificar con Ing. Industrial")
184 |                 Exit Sub
185 |             End If
186 |             tiempo_maquina = math.Round(((t_ciclo+126.51)*_h1/82.08*100,3)'((( 342.68 + t_ciclo ) / 432) * 100,3)
187 |         ElseIf _paso = 2
188 |             If banElementoFijo = False Then
189 |                 tipo = Module1.sacar_tipo_material(_material)
190 |             End If
191 |             If tipo = "HIERRO GRIS" or tipo = "HIERRO GRIS CENTRIFUGADO"Then
192 |                 t_ciclo = 219.48'234
193 |             ElseIf tipo = "HIERRO GRIS INTERMEDIO"
194 |                 t_ciclo = 319.74'234
195 |             ElseIf tipo = "HIERRO GRIS ALTO MODULO"
196 |                 t_ciclo = 319.74'234
197 |             ElseIf tipo = "HIERRO DUCTIL"
198 |                 t_ciclo = 319.74'320.40
199 |             ElseIf tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON"
200 |                 t_ciclo = 219.48'320.40
201 |             Else
202 |                 msgbox("Tipo de material no disponible,imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
203 |                 Me.ListaAlertas.Add(Name & ": El material & " & tipo & " no se encuentra disponible para el calculo, verificar con Ing. Industrial")
204 |                 Exit Sub
205 |             End If
206 |             tiempo_maquina = (((t_ciclo+93.91)*(_h1))/67.14*100)'((( 179.94 + t_ciclo ) / 360) * 100,3)
207 |         Else
208 |             Me.ListaAlertas.Add(Name & ": Error en calculo de tiempos estandar, no se encuentra disponible el num de paso")
209 |         End If
210 |         tiempo_mano=tiempo_maquina
211 |     Catch er As Exception
212 |         Me.ListaAlertas.Add(Name & ": " & er.Message)
213 |     End Try
214 | End Sub

```

**Ilustración 138 método calcular( ).- CT- 9242.**

**NOTA:** En la ilustración 138 lo que se aprecia de color verde son los valores y formulas con las que contaba el CT-.9242 antes de la actualización en el documento se menciona que se eliminaron, pero en realidad solo se dejaron como comentario en el código y son inutilizables.

### 3.1.6.15 CentroTrabajo9265.vb – Splitter L.B.

Para la actualización de este centro de trabajo fue necesario realizar una actualización a la interface gráfica del CT-9265. Agregando un label y un comboBox para los nuevos datos que se requieren para realizar las operaciones, también se decidió ocultar los campos que ya no se utilizarán como se puede apreciar en la ilustración 139 se ocultaron los valores que pedía antes.

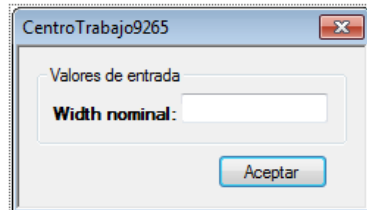


Ilustración 140 form antes de la actualización.- CT- 9265.

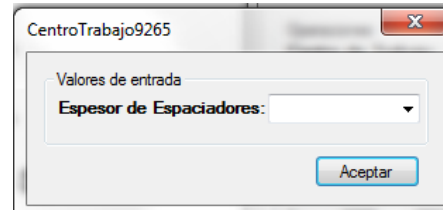


Ilustración 139 form actualizado.- CT- 9265.

Los siguientes cambios en el código del CT-9265. Fueron agregando una nueva variable llamada “\_esp” de tipo double para que almacene los valores ingresados por el usuario (Ilustración 141).

```
104 | '-----variables que se necesitan-----  
105 | Dim _h1 As Double  
106 | Dim _esp As Double  
107 | '-----
```

Ilustración 141 Variables necesarias.- CT- 9265.

En el procedimiento Load del CT-9265. Se agregó el código que oculta los campos que ya no utilizaremos en el form (Ilustración 142).

```
159 | Sub CentroTrabajo9265Load(sender As Object, e As EventArgs)  
160 |     label1.Visible = False  
161 |     textBox1.Visible = False  
162 | End Sub
```

Ilustración 142 Evento Load( ).- CT- 9265.

Posterior a eso se procedió a actualizar la fórmula matemática que almacena la variable tiempo\_maquina en el método calcular( ) del CT-9265. La fórmula que se aprecia de color verde en la ilustración 131 es la fórmula que anteriormente se utilizaba (Ilustración 143).

```
122 | Public Sub calcular() implements ICentroTrabajo.calcular  
123 |     Try  
124 |         tiempo_septup=buscar_setupin(centro)  
125 |         tiempo_maquina = math.Round((434.38/(36*(4.2/(_esp+0.045))))*100,3)  
126 |         '((( ( 85.37422 + 313.26 ) * ( _h1 + 0.045 ) ) / 151.2) * 100,3)  
127 |         tiempo_mano = math.Round(tiempo_maquina * factorLabor,3)  
128 |     Catch er As Exception  
129 |         Me.ListaAlertas.Add(Name & ": " & er.Message)  
130 |     End Try
```

Ilustración 143 método calcular( ).- CT- 9265.

Para terminar la actualización del CT-9265. Se tuvo que cambiar la condicionante IF de la función clic del botón para que tomara el valor del comboBox que se agregó y la variable \_esp pase a tomar el valor ingresado en el comboBox1, se dejó como comentario la línea donde obtenía el valor del Width anteriormente ya que actualmente ya no se usa (Ilustración 144).

```

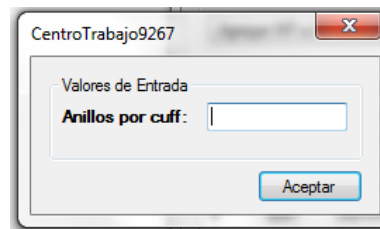
144 Sub Button1Click(sender As Object, e As EventArgs)
145     If isnumeric(Me.comboBox1.Text) Then
146         ' _h1 = CDec(textBox1.Text)
147         _esp = CDec(textBox1.Text)
148         Me.calcular()
149         Me.Close()
150     Else
151         MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
152     End If
153 End Sub

```

**Ilustración 144 función clic del botón aceptar.- CT- 9265.**

### 3.1.6.16 CentroTrabajo9267.vb – Shape & Notch L.B.

Para realizar la actualización de este centro de trabajo fue necesario modificar la interface gráfica del mismo ya que anteriormente no se mostraba la interface, y solo realizaba los cálculos requeridos. Actualmente se necesita saber el número de anillos por cuff para realizar la operación, para ello fue necesario agregar un GroupBox, un label, un TextBox y un botón a la interface.



**Ilustración 145 nueva interface gráfica.- CT- 9267.**

Una vez realizado lo anterior fue necesario agregar una nueva variable llamada “ringsXcuff” que es la que almacenara el valor ingresado por el usuario Ilustración146.

```

98 '-----variables que se necesitan-----
99 Dim ringsXcuff As Integer = 1
100 '-----

```

**Ilustración 146 variables necesarias.- CT- 9267.**

En seguida se procedió a actualizar la fórmula matemática que almacena la variable “tiempo\_maquina” en el método calcular( ) como se aprecia en la línea 118 de la ilustración 147, debajo de esta e puede apreciar la fórmula que anteriormente utilizaba el CT-9267. De color verde

```

115|  Public Sub calcular() implements ICentroTrabajo.calcular
116|      Try
117|          tiempo_septup = buscar_setupin(centro)
118|          tiempo_maquina = ((69.8029697/(36*ringsXcuff))*100)
119|          '(1.96204560897436 * 100,3)
120|          tiempo_mano = math.Round(tiempo_maquina * factorLabor,3)
121|      Catch er As Exception
122|          Me.ListaAlertas.Add(Name & ": " & er.Message)
123|      End Try
124|
125|  End Sub

```

**Ilustración 147 método calcular().- CT-9267.**

En la función del procedimiento clic del botón aceptar del CT-9267 se agregó una condicionante IF para verificar si el valor ingresado por el usuario es numérico, en caso de que no sea numérico mostrara un mensaje de alerta con la leyenda “Revisar valores”. De ser numérico el valor ingresado procederá a realizar los cálculos necesarios incluidos en el método calcular( ) (Ilustración 148).

```

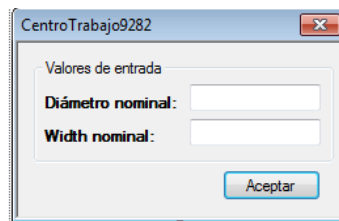
138|  Sub Button1Click(sender As Object, e As EventArgs)
139|      If IsNumeric(textBox1.Text) Then
140|          ringsXcuff = CDec(textBox1.Text)
141|          Me.calcular()
142|          Me.Close()
143|      Else
144|          msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
145|      End If
146|  End Sub
147| End Class

```

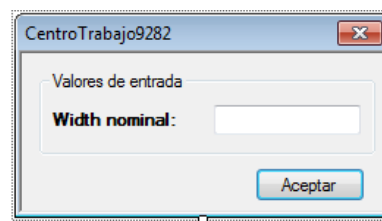
**Ilustración 148 función clic del boton aceptar.- CT- 9267.**

### 3.1.6.17 CentroTrabajo9282.vb – Frm Turn/Bore >6.5 L.B. Iron.

Para realizar la actualización de este centro de trabajo fue necesario modificar la interface gráfica del mismo ya que anteriormente pedía dos valores de entrada, para la actualización será necesario ocultar los campos que ya no se utilizaran para realizar la operación.



**Ilustración 150 form antes de la actualización.- CT- 9282.**



**Ilustración 149 form actualizado.- CT- 9282.**

Una vez realizado lo anterior, en el procedimiento load del centro de trabajo se declaran los objetos que se ocultaran en el form (Ilustración 151).

```

156|  Sub CentroTrabajo9282Load(sender As Object, e As EventArgs)
157|      label3.Visible = False
158|      textBox2.Visible = False
159|  End Sub

```

**Ilustración 151 Evento Load().- CT- 9282.**



Después en el método calcular( ) del centro de trabajo 9282 se modificó la fórmula matemática que almacena la variable “tiempo\_maquina” por la nueva fórmula. Como se puede apreciar en la ilustración 152 la nueva fórmula se encuentra en la línea 130 y la que anteriormente se utilizaba se muestra de color verde en la línea 131.

```

128 Public Sub calcular() implements ICentroTrabajo.calcular
129     tiempo_septup=buscar_setupin(centro)
130     tiempo_maquina = math.Round((1391.31*_h1)/134.46*100,3)
131     '((( ( 151.9025 + ( ( 1530.93 * _d1 ) / 7.4803 ) ) * _h1 ) / 266.4) * 100,3)
132     tiempo_mano=tiempo_maquina
133 End Sub

```

**Ilustración 152 método calcular( ).- CT- 9282.**

Seguido de eso en la función del procedimiento clic del botón aceptar del centro de trabajo 9282 se modificó quitando de la condicionante la condición de que el valor ingresado en el textbox2 tiene que ser un valor numérico y la operación con la cual la variable \_d1 tomaba el valor ingresado por el usuario en el textbox2 (Ilustración 153).

```

145 Sub Button1Click(sender As Object, e As EventArgs)
146     If isnumeric(Me.textBox1.Text) Then 'And isnumeric(Me.textBox2.Text) Then
147         Me._h1 = Me.textBox1.Text
148         'Me._d1 = Me.textBox2.Text
149         calcular()
150         close()
151     Else
152         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
153     End If
154 End Sub

```

**Ilustración 153 función clic del botón aceptar.- CT- 9282.**

### 3.1.6.18 CentroTrabajo9283.vb – FORM TRN/BORE DUCTIL.

Para la actualización de este centro de trabajo solamente fue necesario modificar la fórmula matemática que almacena la variable tiempo\_maquina en el método calcular( ) del CT-9283. Antes de esto se tiene que agregar una nueva variable llamada “t\_ciclo” dentro del mismo método calcular( ) para que almacene e valor del tiempo ciclo del material. Como solo utiliza un material solo se igualo la variable al valor del tiempo ciclo como se puede ver en la ilustración 154 línea 130.

```

129 Public Sub calcular() Implements ICentroTrabajo.calcular
130     Dim t_ciclo As Double = 1652.78
131     Try
132         tiempo_septup=buscar_setupin(centro)
133         tiempo_maquina = (((70.23+(t_ciclo/9.062)*_diametro))*_h1)/134.28*100)
134         '((( ( 205.52173 + ( 2765.5 / 9.062 ) * _diametro ) ) * _h1 ) / 248.76) * 100,3)
135         tiempo_mano=tiempo_maquina
136     Catch er As Exception
137         me.ListaAlertas.Add(Name & ": " & er.Message)
138     End Try
139 End Sub
140

```

**Ilustración 154 método calcular( ).- CT- 9283.**

NOTA: la fórmula que se aprecia de color verde en la ilustración 154 es la que se utilizaba anteriormente, en la línea 133 se aprecia la formula actualizada.

### 3.1.7 Expansores

#### 3.1.7.1 CentroTrabajo499.vb – „GAS NITRIDING EXPANDERS“.

Para la actualización en este centro de trabajo es necesario ingresar al „ProyectoSa” y una vez ahí dentro buscar en el proyecto el „CentroTrabajo499.vb” dar doble clic. En la pantalla que se abra te mostrara el codigo del formulario, cuando estes dentro del codigo busca el método calcular( ) que es donde se encuentran la operaciones que realiza para obtener el tiempo estandar del centro de trabajo, cuando lo hallas ubicado dirigitete a la fórmula que almacena la variable „tiempo\_maquina” y procede a modificar la fórmula matematica (Ilustración 155).

```

121 Public Sub calcular() implements ICentroTrabajo.calcular
122     tiempo_septup = buscar_setupin(centro)
123     '-----Actualizacion realizada 19/10/2016---JabneelChavez-----
124     tiempo_maquina = Math.Round (36031.13 / (36 * (160 * ((6.5825 / width_a ) * 2 ))) * 100, 3)
125     '-----
126     '---FORMULA OBSOLETA+---tiempo_maquina = math.Round((35701.71/(36*(190*((6.55/width_a)*2)))) * 100,3)
127     '---FORMULA ANTERIOR A LA ULTIMA ACTUALIZACION---tiempo_maquina = math.Round((35701.71 / ( 36 * ( 190 * ( ( 6.55 / width_a ) * 2 ) ) ) ) * 100,3)
128     tiempo_mano = tiempo_maquina
129 End Sub

```

Ilustración 155 metodo calcular( ).- CT- 499.

Como se puede apreciar en la ilustración 155 en la línea 124 se encuentra la fórmula del tiempo estándar actualizada, debajo de esta en la línea 126 aparece la formula obsoleta más antigua, y debajo esta la que se usaba anteriormente hasta el día de la actualización.

#### 3.1.7.2 CentroTrabajo560.vb – „EXPANDER SS-50 PUNCHING “.

Prácticamente en este centro de trabajo solo se tuvo que actualizar la fórmula matemática para calcular el tiempo estándar para esto hay que ubicar el método calcular ( ), una vez ahí proceder a modificar la fórmula que almacena la variable „tiempo\_maquina”. Como se puede apreciar en la ilustración 156 línea 131 aparece la nueva fórmula actualizada, y debajo de esta de color verde la formula obsoleta. También en las líneas 126 y 127 se dejaron como comentario las dos constantes que se utilizaban en la formula anterior.

```

126 Dim factor1 As Double =1.6667
127 Dim factor2 as Double =0.00055
128
129 Public Sub calcular() implements ICentroTrabajo.calcular
130     tiempo_septup=buscar_setupin(centro)
131     tiempo_maquina = ((0.1702 + (jorobas * ((_d1 * 4.4954) / 3.1988) / 27)) / 36) * 100
132     'tiempo_maquina = ( ( jorobas * _d1 * 0.081887916549112 )+ 0.17 ) / 36 * 100
133     tiempo_mano= tiempo_maquina * factorLabor
134 End Sub

```

Ilustración 156 variables y método calcular( ).- CT- 560.

### 3.1.7.3 CentroTrabajo563.vb – „LANCE KEY“.

La actualización en este centro de trabajo se realizó solo en la línea 98 del código del formulario „CentroTrabajo563.vb” donde se actualizo el valor de la variable „factor1” por el siguiente: 0.247402, como se puede apreciar en la ilustración 157.

```
98 | Dim factor1 As Double = 0.247402'0.2674
99 | Dim centro As String = "563"
```

Ilustración 157 variables.- CT- 563.

### 3.1.7.4 CentroTrabajo565.vb – „EXPANDER ES-80 FORMING “.

Para realizar la actualización de este centro de trabajo hay que ubicar el método calcular ( ), una vez ahí proceder a modificar la fórmula que almacena la variable „tiempo\_maquina”. Como se puede apreciar en la ilustración 158 la formula obsoleta aparece de color verde en la línea 124 y sobre ella en la línea 123 se aprecia la fórmula del tiempo estándar actualizada.

```
121 | Public Sub calcular() implements ICentroTrabajo.calcular
122 | tiempo_septup=buscar_setupin(centro)
123 | tiempo_maquina =(((1.95 * ((Num_pasos * 2) - 1) /195) + 0.0676)/36)
124 | ' math.Round((1.14+(((1.39*(Num_pasos * 2) -1))/ 135))/36,3)
125 | tiempo_maquina = math.Round(tiempo_maquina * 100,3)
126 | tiempo_mano = tiempo_maquina
127 | End Sub
```

Ilustración 158 método calcular( ).- CT- 565.

### 3.1.7.5 CentroTrabajo566.vb – „WIRE LATCH“.

Para realizar la actualización de este centro de trabajo fue necesario modificar la interface gráfica ya que en un principio en este centro de trabajo no se solicitaba ningún valor para ser ingresado por el usuario.

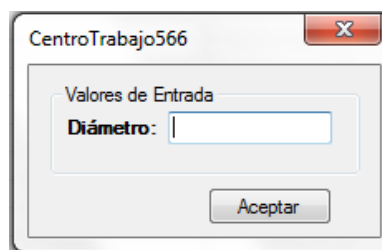


Ilustración 159 nueva interface gráfica.- CT- 566.

En seguida hay que activar la función para que se muestre el formulario y eliminar la función donde solo realizaba las operaciones correspondientes en el método CargarInterfaz ( ) línea 45 ilustración 160.

```
44 | Public Sub CargarInterfaz() Implements ICentroTrabajo.CargarInterfaz
45 | ShowDialog()
46 | End Sub
```

Ilustración 160 método Sub CargarInterfaz( ).- CT- 566.

En seguida hay que declarar la nueva variable que obtendrá el valor y tipo de datos ingresado en el método „public sub pasarValoresDeListas ( )” como se aprecia en la ilustración 161 línea 94. También se agregó en el método Public Sub, el iniciador de los componentes u objetos agregados al formulario línea 98, y la variable que almacenara y enviara la descripción de los datos requeridos para la operación cuando se desea realizar el cálculo mediante la vista WPF líneas 100 y 101.

```

93 | Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasa
94 |     d1 = module1.obtenerValorDatoDouble(lDatos,"diaNominalAnillo")
95 | End Sub
96 |
97 | Public Sub New()
98 |     Me.InitializeComponent()
99 |
100 |     Dim diaNominalAnillo As New Dato("diaNominalAnillo","Distance","Diámetro nominal del anillo (Plano)","Nothing","Diámetro nominal del anillo:")
101 |     listaDatosRequeridos.Add(diaNominalAnillo)
102 | End Sub

```

**Ilustración 161 public Sub New() y Public Sub pasarValoresDeListas() .- CT- 566.**

Se declara la nueva variable que almacenará el valor ingresado por el usuario como se puede apreciar en la ilustración 162 línea 114 - 117.

```

114 | Dim d1 As Double
115 | Dim tciclo As Double
116 | Dim aCiclicas As Double
117 | Dim aNCiclicas As Double
118 |

```

**Ilustración 162 variables necesarias.- CT- 566.**

Posterior hay que buscar el método calcular ( ), y una vez ahí hay que declarar una condicionante IF como podemos ver en la ilustración 163 línea 126 para que dependiendo del valor ingresado por el usuario las variables internas tomen el valor correspondiente para realizar el cálculo, aquí también se le asignó el valor correspondiente a las variables anteriormente declaradas. En la línea 135 se agregó la nueva fórmula que almacena la variable „tiempo\_maquina” actualizada para calcular el tiempo estándar.

```

124 | Public Sub calcular() Implements ICentroTrabajo.calcular
125 |     buscar_septup()
126 |     If d1>=3 Then
127 |         tciclo = 2.75923077
128 |         aCiclicas=2.57060734
129 |         aNCiclicas= 2.04128219
130 |     ElseIf d1<3
131 |         tciclo=2.92545455
132 |         aCiclicas =7.43378870
133 |         aNCiclicas= 0.11551304
134 |     End If
135 |     tiempo_maquina=((tciclo + aCiclicas + aNCiclicas)/36)*100
136 |     '0.01942 * 100
137 |
138 |     tiempo_mano = tiempo_maquina
139 | End Sub

```

**Ilustración 163 método calcular() .- CT- 566.**

### 3.1.7.6 CentroTrabajo581.vb – „EXPANDER 98-22 FORMING“.

Como en los centros de trabajo anteriores es necesario ubicar el método calcular ( ), una vez ahí proceder a modificar la fórmula matemática que almacena la variable „tiempo\_maquina“. Como podemos apreciar en la ilustración 164, en la línea 125 se encuentra la fórmula para calcular el tiempo estándar actualizada y debajo de ella en la línea 126 se ve de color verde la fórmula obsoleta que utilizaba el centro de trabajo para calcular el tiempo estándar.

```
123 | Public Sub calcular() implements ICentroTrabajo.calcular
124 |     tiempo_septup=buscar_setupin(centro)
125 |     tiempo_maquina =((0.4289 + (( 2.8 * jorobas) / 93)) / 36)*100
126 |     '(factor1+(factor2*jorobas)/factor3))*100
127 |     tiempo_mano=tiempo_maquina
128 | End Sub
```

Ilustración 164 método calcular().- CT- 581.

Nota: en este centro de trabajo la formula anterior era muy parecida a la actual y esta tenía los valores no variables almacenados en las variables factor1, factor2 y factor3, estos valores ya no son requeridos en la operación así que se dejaron como comentario para que quede registro de las últimas actualizaciones realizadas (Ilustración 165).

```
106 | Dim factor1 As Double '=0.0119152
107 | Dim factor2 As Double '= 0.030753
108 | Dim factor3 As Double '= 36
109 | Dim centro As String = "581"
```

Ilustración 165 variables.- CT -581.

### 3.1.7.7 CentroTrabajo721.vb – „BLACKENING EXPANDERS“.

Para realizar la actualización en este centro de trabajo hay que ubicar el método calcular ( ), una vez ahí proceder a modificar la fórmula matemática que almacena la variable „tiempo\_maquina“. Como se puede apreciar en la ilustración 166 se observa en la línea 124 la formula actualizada y debajo de esta de color verde en la línea 125 la formula obsoleta que utilizaba el centro de trabajo hasta la actualización.

```
121 | Public Sub calcular() implements ICentroTrabajo.calcular
122 |     tiempo_septup=buscar_setupin(centro)
123 |     tiempo_maquina = ((56.97056)/(12*((12.8/_width)*2)))
124 |     tiempo_maquina = (1467.65 / (36 * ((312 / _width) * 2))) * 100
125 |     'math.Round(tiempo_maquina * 100,3)
126 |     tiempo_mano = tiempo_maquina
127 | End Sub
```

Ilustración 166 método calcular().- CT- 721.

### 3.1.7.8 CentroTrabajo740.vb – „ELECTROPOLISH EXPANDERS“.

Para realizar la actualización es necesario ubicar el método calcular ( ), una vez ahí proceder a modificar la fórmula matemática que almacena la variable „tiempo\_maquina“. Como se puede apreciar en la ilustración 167 se observa en la línea 124 la fórmula actualizada y debajo de esta en la línea 125 la fórmula obsoleta que utilizaba el centro de trabajo hasta la actualización.

```
122 | Public Sub calcular() implements ICentroTrabajo.calcular
123 |     tiempo_septup=buscar_setupin(centro)
124 |     tiempo_maquina=((516.7207 * h1) / 3024)* 100
125 |     |( ( 2.62361 ) / ( ( 13.5 / h1 ) * 2 ) ) * 100
126 |     tiempo_mano = tiempo_maquina
127 | End Sub
128 |
```

Ilustración 167 método calcular( ).- CT- 740.

### 3.1.7.9 CentroTrabajo745.vb – „CHEMICAL POLISH EXPANDERS “.

Para realizar la actualización en este centro de trabajo es necesario ubicar el método calcular ( ), una vez ahí hay que declarar una nueva variable fuera del método para que almacene el valor de la carga con cada width diferente, enseguida hay que cambiar el tipo de dato de String a Double de la variable Espesor. Después de realizar lo anterior hay que proceder ahora si a modificar el código del método calcular para que realice los cálculos y operaciones correspondientes, dentro de este método hay que agregar una condicionante IF para obtener el valor de la carga dependiendo del width ingresado por usuario. En la ilustración 168, línea 151 se encuentra la nueva fórmula actualizada y debajo de esta la fórmula obsoleta de un color verde.

```
105 | Dim Espesor As Double
106 |
107 | Dim carga As Double
108 | Dim factor1 As Double = 0.259032
109 | Dim centro As String = "745"
110 | Public Property CentroTrabajo As String Implements ICentroTrabajo.CentroTrabajo
111 |     Get
112 |         Return centro
113 |     End Get
114 |     Set
115 |         centro = value
116 |     End Set
117 | End Property
118 |
119 | Public tiempo_maquina As Double
120 | Public tiempo_septup As Double
121 | Public tiempo_mano As Double
122 |
123 | Public Sub calcular() Implements ICentroTrabajo.calcular
124 |
125 |     If Espesor = 0.0787 Then
126 |         carga = 25.184
127 |     ElseIf Espesor = 0.0984
128 |         carga = 22.632
129 |     ElseIf Espesor = 0.1102
130 |         carga = 24.244
131 |     ElseIf Espesor = 0.1181
132 |         carga = 24.891
133 |     ElseIf Espesor = 0.125
134 |         carga =21.250
135 |     ElseIf Espesor = 0.1378
136 |         carga = 22.848
137 |     ElseIf Espesor = 0.1562
138 |         carga = 24.992
139 |     ElseIf Espesor = 0.1563
140 |         carga = 25.808
141 |     ElseIf Espesor = 0.1574
142 |         carga = 23.610
143 |     ElseIf Espesor = 0.1575
144 |         carga = 22.850
145 |     ElseIf Espesor = 0.1875
146 |         carga =20.625
147 |     ElseIf Espesor = 0.1968
148 |         carga = 21.648
149 |     End If
150 |     tiempo_septup=buscar_setupin(centro)
151 |     tiempo_maquina = (((399.04) * (Espesor)) / (carga * 36))
152 |     (Espesor*factor1)
153 |     'tiempo_maquina = math.Round(tiempo_maquina * 100,3)
154 |     tiempo_mano = tiempo_maquina
155 | End Sub
```

Ilustración 168 variables necesarias y método calcular( ).- CT- 745.

### 3.1.7.10 CentroTrabajo832.vb – „FINAL INSP. EXPANDER“.

Para llevar a cabo la actualización en este centro de trabajo es necesario acceder a ProyectoSa y buscar el CentroTrabajo832.vb dentro de la carpeta Centrodetrabajo. Una vez que lo ubiques dar doble clic sobre él. Se abrirá una ventana con la parte del código del formulario, una vez ahí hay que agregar dos variables nuevas para que almacene el valor de las actividades cíclicas de pintura y el total de piezas por tubo, la primera llamada „AcP” de tipo Double e igualarla a cero y la segunda llamada „pzsXrollo” de tipo Double. Después se declara una condicionante IF donde dependiendo de si el número de franjas a pintar es mayor que cero la variable „AcP” toma el valor de 2.4410206 (Ilustración 169).

```
173 Dim AcP As Double = 0
174
175 Public Sub calcular() Implements ICentroTrabajo.calcular
176     Dim pzsXrollo As Double
177
178     If _noFranjas > 0 Then
179         AcP = 2.4410206
180     End If
```

Ilustración 169 nueva variable y método calcular().- CT- 832.

En seguida buscar el método calcular ( ), para realizar la actualización hay que modificar la fórmula matemática que almacena la variable „tiempo\_maquina” ubicada en el método calcular() pero antes de esto hay que agregar un condicionante IF para obtener el valor de la variable „pzsXrollo” ya que dependiendo del valor del width ingresado y el tipo de expansor seleccionado por el usuario toma el valor correspondiente de la variable „pzsXrollo” que es necesaria para realizar la operación matemática (Ilustraciones 170 y 171).

```
175 Public Sub calcular() Implements ICentroTrabajo.calcular
176     Dim pzsXrollo As Double
177
178     If _noFranjas > 0 Then
179         AcP = 2.4410206
180     End If
181
182     tiempo_septup = buscar_setupin(centro)
183
184     -----EXPANSOR ES80-----
185
186     If _tipo = "ES80" And espesor = 0.0590 Then
187         pzsXrollo = 380
188     ElseIf _tipo = "ES80" And espesor = 0.0772
189         pzsXrollo = 330
190     ElseIf _tipo = "ES80" And espesor = 0.0787
191         pzsXrollo = 320
192     ElseIf _tipo = "ES80" And espesor = 0.0984
193         pzsXrollo = 230
194     ElseIf _tipo = "ES80" And espesor = 0.1102
195         pzsXrollo = 220
196     ElseIf _tipo = "ES80" And espesor = 0.1181
197         pzsXrollo = 210
198     ElseIf _tipo = "ES80" And espesor = 0.1378
199         pzsXrollo = 160
200     ElseIf _tipo = "ES80" And espesor = 0.1575
201         pzsXrollo = 140
202     ElseIf _tipo = "ES80" And espesor = 0.1577
203         pzsXrollo = 140
204     ElseIf _tipo = "ES80" And espesor = 0.1840
205         pzsXrollo = 110
206     ElseIf _tipo = "ES80" And espesor = 0.1875
207         pzsXrollo = 110
208     ElseIf _tipo = "ES80" And espesor = 0.0590
209         pzsXrollo = 380
210     ElseIf _tipo = "ES80"
211         pzsXrollo = 380
212
```

Ilustración 170 método calcular().- CT- 832.

```
213 -----EXPANSOR S550-----
214
215     ElseIf _tipo = "S550" And espesor = 0.0984
216         pzsXrollo = 140
217     ElseIf _tipo = "S550" And espesor = 0.1100
218         pzsXrollo = 125
219     ElseIf _tipo = "S550" And espesor = 0.1102
220         pzsXrollo = 125
221     ElseIf _tipo = "S550" And espesor = 0.1180
222         pzsXrollo = 120
223     ElseIf _tipo = "S550" And espesor = 0.1181
224         pzsXrollo = 120
225     ElseIf _tipo = "S550" And espesor = 0.1250
226         pzsXrollo = 110
227     ElseIf _tipo = "S550" And espesor = 0.1378
228         pzsXrollo = 110
229     ElseIf _tipo = "S550" And espesor = 0.1560
230         pzsXrollo = 90
231     ElseIf _tipo = "S550" And espesor = 0.1575
232         pzsXrollo = 90
233     ElseIf _tipo = "S550" And espesor = 0.1577
234         pzsXrollo = 90
235     ElseIf _tipo = "S550" And espesor = 0.1772
236         pzsXrollo = 80
237     ElseIf _tipo = "S550" And espesor = 0.1840
238         pzsXrollo = 70
239     ElseIf _tipo = "S550" And espesor = 0.1870
240         pzsXrollo = 75
241     ElseIf _tipo = "S550" And espesor = 0.1875
242         pzsXrollo = 75
243     ElseIf _tipo = "S550" And espesor = 0.1969
244         pzsXrollo = 70
245     ElseIf _tipo = "S550" And espesor = 0.1970
246         pzsXrollo = 70
247     ElseIf _tipo = "S550" And espesor = 0.2480
248         pzsXrollo = 55
249     ElseIf _tipo = "S550" And espesor = 0.2490
250         pzsXrollo = 55
251     ElseIf _tipo = "S550" And espesor = 0.2500
252         pzsXrollo = 55
253     ElseIf _tipo = "S550"
254         pzsXrollo = 55
```

Ilustración 171 método calcular().- CT- 832.

Después hay que ubicar la variable „tiempo\_maquina” para su modificación, esta se encuentra dentro del método calcular( ), (ilustración 172 línea 283) Aquí se cambia la formula por la formula actualizada.

```

282 |         End If
283 |         tiempo_maquina = ((98.855494 + (17.43625 * _noFranjas) + (AcP)) / (36 * pzsXrollo)) * 100
284 |         tiempo_mano = tiempo_maquina
285 |         tiempo_maquina = 0
286 |     End Sub

```

**Ilustración 172 nueva fórmula.- CT- 832.**

Por ultimo en el evento click del boton aceptar se modifica el codigo cambiando los valores de un textBox1 que se elimino y en su lugar se agrego un comboBox1 para que realice las operaciones necesarias (Ilustración 173).

```

302 | Sub Button1Click(sender As Object, e As EventArgs)
303 |     If isnumeric(comboBox1.Text) And isnumeric(textBox2.Text) Then
304 |         espesor = comboBox1.Text
305 |         _noFranjas = textBox2.Text
306 |         If rb_es80.Checked = True Then
307 |             _tipo = "ES80"
308 |             calcular()
309 |         ElseIf rb_ss50.Checked = True
310 |             _tipo = "SS50"
311 |             calcular()
312 |         ElseIf rb9822.Checked = True
313 |             _tipo = "9822"
314 |             calcular()
315 |         Else
316 |             msgbox("Elija un tipo de expansor", MsgBoxStyle.Exclamation, "Ing. de Rutas")
317 |         End If
318 |         Close()
319 |     Else
320 |         msgbox("Revisar valores", MsgBoxStyle.Exclamation, "Ing. de Rutas")
321 |     End If
322 | End Sub

```

**Ilustración 173 función clic del botón aceptar.- CT- 832.**

## 3.1.8 Franklin

### 3.1.8.1 CentroTrabajo2005.vb – „FIRST ROUGH GRIND NORTON“.

Para realizar la actualización de este centro de trabajo es necesario dirigirse a la parte del código donde se encuentre el método calcular( ). Una vez ahí hay que proceder a modificar y actualizar la fórmula matemática necesaria para calcular el tiempo estándar. Como se puede apreciar en la ilustración 174 en la línea 134 se encuentra la nueva fórmula actualizada y debajo de esta de color verde la formula obsoleta.



```

132 Public Sub calcular() implements ICentroTrabajo.calcular
133     buscar_septup()
134     tiempo_maquina = (((30.36 * _numeroPasadas2005) + 17.43) * (_h1)) / 720)
135     '( 89.61 * ( _h1 * _numeroPasadas2005 ) ) / 828
136     tiempo_maquina = math.Round(tiempo_maquina * 100,3)
137     tiempo_mano = tiempo_maquina * factorLabor
138 End Sub

```

Ilustración 174 método calcular().- CT- 2005.

### 3.1.8.2 CentroTrabajo2020.vb – „ROUGH BORE 1 CUTS SPLIT BUSH“.

Para realizar la actualización de este centro de trabajo es necesario ubicar el método calcular( ), una vez que te encuentres ahí hay que proceder a modificar o actualizar la fórmula matemática para realizar el cálculo del tiempo estándar. Como se puede apreciar en la ilustración 175 en la línea 137 se encuentra la nueva fórmula actualizada y debajo de esta de color verde la fórmula obsoleta.

```

135 Public Sub calcular() implements ICentroTrabajo.calcular
136     tiempo_septup = buscar_setupin(centro)
137     tiempo_maquina = ((81.85929429 * _WidthCastingMayor)/154.8)*100
138     'math.Round((( 2.8351 / ( 8.6 / _WidthCastingMayor ) )) * 100,3)
139     tiempo_mano = tiempoMaquina
140 End Sub

```

Ilustración 175 método calcular().- CT- 2020.

### 3.1.8.3 CentroTrabajo2030.vb – „SECOND ROUGH GRIND“.

Para realizar la actualización en este centro de trabajo fue necesario modificar la interfaz gráfica del form ya que se agregaron nuevos objetos que serán útiles para que el usuario pueda identificar a que variable le está ingresando un nuevo valor.

Ilustración 177 form antes de la actualización.- CT- 2030.

Ilustración 176 form actualizado.- CT- 2030.

En la función Test se agregó el método mediante el cual obtiene el tipo de material seleccionado por el usuario, se crea una condicionante if ya que dependiendo del tipo de material seleccionado enviara un mensaje de alerta si no se puede utilizar el

material en el centro de trabajo, en seguida se agregaron las variables que almacenaran el tipo de material y si hay algun elemento seleccionado (Ilustración 178).

```

86 Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean
87     _material = Module1.obtenerValorDatoString(lDatosString,"MATERIAL")
88     tipo = Module1.sacar_tipo_material(_material)
89
90     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO DUCTIL" Then
91
92     Else
93         MsgBox("Material " & tipo & "no disponible para el C.C. & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf
94         "Por favor seleccione un tipo de material disponible para poder calcular el centro de trabajo", MsgBoxStyle.Critical,"Ing. de Rutas")
95         Dim listaOpcionales As New List(Of String)
96         listaOpcionales.Add("HIERRO GRIS")
97         listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
98         listaOpcionales.Add("HIERRO DUCTIL")
99         Dim f As New Seleccionar(listaOpcionales)
100        If f.ShowDialog() = DialogResult.OK Then
101            tipo = f.elementoSeleccionado
102            banElementoFijo = True
103        End If
104        Exit Function
105    End If
106    Return True
107 End Function
108
109 Dim tipo As String
110 Dim banElementoFijo As Boolean = False

```

**Ilustración 178 función Test( ).- CT- 2030.**

Dentro del procedimiento pasarValoresDeListas se añadió una condicionante if donde si el no hay ningún elemento seleccionado de la lista de materiales, entonces va a cargar la información de los materiales disponibles en la variable \_material (Ilustración 179). También en el procedimiento sub New( ) Se declaran nuevas variables y se les asigna el tipo de dato y una pequeña serie de descripciones necesarias para generar una vista WPF (Ilustración 179).

```

112 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString))
113     If banElementoFijo = False Then
114         _material = Module1.obtenerValorDatoString(lDatosString,"MATERIAL")
115     End If
116     _diaCastingMayor = module1.obtenerValorDatoDouble(lDatos,"diaBCasting")
117     rpm = Module1.obtenerValorDatoDouble(lDatos,"RPM")
118
119 End Sub
120
121 Public Sub New()
122     Me.InitializeComponent()
123     'Dim diaBCasting As New Dato("diaBCasting","Distance","Diametro B de la placa modelo",Nothing,"Díámetro B del casting:")
124     'listaDatosRequeridos.Add(diaBCasting)
125
126     Dim material_ As New DatoString("MATERIAL","Material Componente MF012-S,MF013-H,MF053-H,etc..","Material:")
127     listaDatosStringRequeridos.Add(material_)
128
129     Dim RPM As New Dato("RPM","Velocidad","Revoluciones por Minuto",Nothing,"RPM: ")
130     listaDatosRequeridos.Add(RPM)
131 End Sub

```

**Ilustración 179 procedimiento Sub pasarValoresDeListas( ) y Sub New( ).- CT- 2030.**

Posterior hay que agregar 4 nuevas variables necesarias para realizar los cálculos del tiempo estándar (rpm, \_material, rpmM y tciclo), (Ilustración 180).

```

155 |-----Valores que se requieren-----
156 | Dim _diaCastingMayor As Double
157 | Dim rpm As Double
158 | Dim _material As String
159 |
160 | Dim rpmM As Double
161 | Dim tciclo As Double
162 |
163 |-----

```

**Ilustración 180 nuevas variables.- CT- 2030.**

Enseguida hay que ubicar el método calcular( ) en donde se encuentra ubicada la variable “tiempo\_maquina” que es la que almacena la fórmula matemática utilizada para realizar el cálculo y obtener el tiempo estándar. Primero hay que agregar un If para determinar si hay seleccionado un material, y de no ser así que cargue el listado de materiales disponibles; en seguida se agrega otra condicionante IF para que dependiendo del tipo de material seleccionado por el usuario las variables “tciclo” y “rpmM” tomen un valor diferente cada una. Ahora si procedemos a actualizar la fórmula matemática que almacena la variable “tiempo\_maquina” (Ilustración 181).

```

177: Public Sub calcular() implements ICentroTrabajo.calcular
178:     tiempo_septup = buscar_setupin(centro)
179:     If banElementoFijo = False Then
180:         tipo = Module1.sacar_tipo_material(_material)
181:     End If
182:     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
183:         tciclo = 0.38
184:         rpmM = 10
185:     ElseIf tipo = "HIERRO DUCTIL"
186:         tciclo = 0.32
187:         rpmM = 12
188:     End If
189:     tiempo_maquina = (((tciclo * rpmM / rpm) + 0.05398708) / (36))*100
190:     'tiempo_maquina = math.Round((( 0.0563 + ( ( 11.4 / 8 ) * 15 ) ) / ( ( ( 37 / _diaCastingMayor ) * 3.78 ) * 36 ) ) * 100,3)
191:     tiempo_mano = tiempoMaquina
192: End Sub

```

**Ilustración 181 método calcular().- CT- 2030.**

Activamos la función load del centro de trabajo 2030 donde programamos la función para que oculte objetos dentro del form, estos objetos se ocultan con la “propiedad .Visible” y tiene que ser igualada a falso. Agregamos el procedimiento “Sub llenar\_material( )” y se codifica la lista de materiales que se podrán utilizar dentro de este centro de trabajo (Ilustración 182).

```

204: Sub CentroTrabajo2030Load(sender As Object, e As EventArgs)
205:     labell.Visible = False
206:     textBox1.Visible = False
207: End Sub
208:
209: Private Sub llenar_material()
210:     Dim l As New List(Of String)
211:     l.Add("HIERRO GRIS")
212:     l.Add("HIERRO GRIS CENTRIFUGADO")
213:     l.Add("HIERRO DUCTIL")
214:     For Each m In obtenerEspecstipoMaterial(l)
215:         cbo_material.Items.Add(m)
216:     Next
217: End Sub

```

**Ilustración 182 Evento Load y procedimiento Sub llenar\_material().- CT- 2030.**

En la función clic del botón Calcular se modificó el código cambiando los valores de referencia en la condicionante If y añadiendo la igualación de las variables con los objetos del form, también se eliminaron las líneas de código que ya no son utilizadas (Ilustración 183).

```

165 Sub Btn_calcularClick(sender As Object, e As EventArgs)
166     If IsNumeric(Me.textBox2.Text) Then
167         _diaCastingMayor = CDec(textBox1.Text)
168         _material = Me.cbo_material.Text
169         rpm = Me.textBox2.Text
170         calcular()
171         Close()
172     Else
173         MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
174     End If
175 End Sub

```

Ilustración 183 función clic del botón aceptar.- CT- 2030.

### 3.1.8.4 CentroTrabajo2035.vb–„SECOND ROUGH GRIND NORTON“.

Para realizar la actualización en este centro de trabajo fue necesario ocultar objetos con los que contaba el form, ya que ya no se utilizaran más para realizar los cálculos. Como podemos observar la interface gráfica fue modificada ya que se ocultaron los campos en los cuales pedía ingresar el valor del width nominal del anillo.

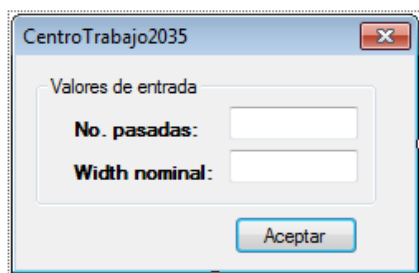


Ilustración 185 form antes de la actualización.- CT- 2035.

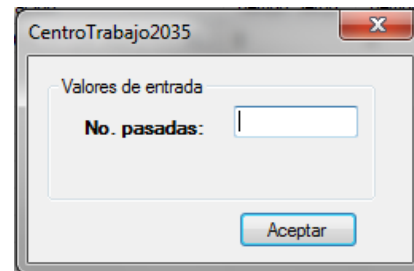


Ilustración 184 form actualizado.- CT- 2035.

Ahora ubicamos el procedimiento Sub PasarValoresDeListas y eliminamos la variable \_h1 y su procedimiento. Enseguida en el procedimiento sub New( ) se elimina la variable widthNominalAnillo, sus descripciones y el proceso que realiza para generar la vista WPF (Ilustración 186).

```

89 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString))
90     ' _h1 = module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")
91     _numeroPasadas2035 = CDb1(Module1.obtenerValorDatoDouble(lDatos,"numeroPasadas2005"))
92 End Sub
93
94 Public Sub New()
95     Me.InitializeComponent()
96
97     'Dim widthNominalAnillo As New Dato("widthNominalAnillo","Distance","Width nominal del anillo (Plano)",Nothing,"Width nominal de anillo:")
98     'listaDatosRequeridos.Add(widthNominalAnillo)
99
100     Dim numeroPasadas2035 As New Dato("numeroPasadas2035","Cantidad","Numero de pasadas en la operación SECOND ROUGH GRIND NORTON (FRANKLIN)",Not
101     listaDatosRequeridos.Add(numeroPasadas2035)
102 End Sub

```

Ilustración 186 procedimiento Sub pasarValoresDeListas( ) y Sub New( ).- CT- 2035.

Proseguimos y agregamos 4 nuevas variables (t\_ciclo, aCiclicas, aNoCiclicas y cXc) que almacenaran los valores constantes y operaciones necesarias para realizar el cálculo (Ilustración 187).

```

132 |         Dim t_ciclo As Double
133 |         Dim aCiclicas As Double = 0.00414163
134 |         Dim aNoCiclicas As Double = 0.10649416
135 |         Dim cXc As Double

```

Ilustración 187 variables nuevas.- CT-2035.

Enseguida nos dirigimos al método calcular( ) y ubicamos la variable “tiempo\_maquina”; aquí se actualizo la fórmula matemática, también se ocultaron los objetos que ya no se utilizan y anteriormente solicitaba el sistema. Para el cálculo del tiempo estándar (Ilustración 188).

```

138 | Public Sub calcular() implements ICentroTrabajo.calcular
139 |     buscar_septup()
140 |     t_ciclo = (0.55333333 * _numeroPasadas2035)
141 |     cXc = (t_ciclo +aCiclicas+aNoCiclicas)
142 |
143 |     tiempo_maquina = ((100 * ((cXc) / 3600)) / 1) * 100
144 |     'FORMULA ACTUALIZADA| ((0.114 + (t_ciclo * _numeroPasadas2035)) / 36) *100
145 |     ' ( 89.61 * ( _h1 * _numeroPasadas2035 ) ) / 828
146 |     'tiempo_maquina = math.Round(tiempo_maquina * 100,3)
147 |     tiempo_mano = tiempo_maquina * factorLabor
148 | End Sub

```

Ilustración 188 método calcular( ).- CT- 2035.

### 3.1.8.5 CentroTrabajo2070.vb – „DEGREASE“.

Prácticamente para realizar la actualización en este centro de trabajo solo es necesario ubicar el método calcular ( ) y dentro de este método buscar la fórmula que almacena la variable “tiempo\_maquina“, una vez que la ubiques hay que proceder a actualizar la fórmula matemática. En esta ocasión solo se actualizo un valor dentro de la formula, como se puede apreciar en la ilustración 189 línea 118, debajo de esta se aprecia de color verde la formula obsoleta.

```

116 | Public Sub calcular() implements ICentroTrabajo.calcular
117 |     tiempo_septup = buscar_setupin(centro)
118 |     tiempo_maquina = math.Round(0.0076474894 * 100,3)]
119 |     '(0.0097 * 100,3)
120 |     tiempo_mano = tiempoMaquina
121 | End Sub

```

Ilustración 189 método calcular( ).-CT- 2070.

### 3.1.8.6 CentroTrabajo2080.vb – „FIN. GRD.“

Se hizo prácticamente lo mismo que en el CentroTrabajo2030.vb

### 3.1.8.7 CentroTrabajo2110.vb – „AUTOLOAD CTB HIGH VOL. CELL“.

Para realizar la actualización de este centro de trabajo es necesario ubicar el método calcular( ), una vez ahí hay que ubicar la variable “tiempo\_maquina” para actualizar la fórmula matemática.

### 3.1.8.8 CentroTrabajo2120.vb – „MILL HOOK LAP-AUT; MAC“.

Para realizar la actualización en este centro de trabajo es necesario ubicar el método calcular( ) y actualizar los valores dentro de la fórmula que almacena la variable “tiempo\_maquina”. Como podemos apreciar en la ilustración 190 la nueva fórmula esta en las líneas 134 y 136 respectivamente al tipo de operación seleccionada por el usuario.

```
131 Public Sub calcular() implements ICentroTrabajo.calcular
132     tiempo_septup = buscar_setupin(centro)
133     If _AutomaticMaclarge = True Then
134         tiempo_maquina = math.Round(0.29314364 * 100,3)'(0.148 * 100,3)
135     Else
136         tiempo_maquina = math.Round(0.15121092 * 100,3)'(0.1388 * 100,3)
137     End If
138     tiempo_mano = tiempoMaquina * factorLabor
```

Ilustración 190 método calcular( ).- CT- 2120

### 3.1.8.9 CentroTrabajo2130.vb – „MILL CHAMF AT GAP - AUTOM.“

Para realizar la actualización en este centro de trabajo es necesario ubicar el método calcular( ) y actualizar los valores dentro de la fórmula que almacena la variable “tiempo\_maquina”. Como podemos apreciar en la ilustración 191 la nueva fórmula está en la línea 118, de color verde a un costado se puede apreciar la formula obsoleta.

```
115 Public Sub calcular() implements ICentroTrabajo.calcular
116     tiempo_septup = buscar_setupin(centro)
117     tiempo_maquina = math.Round(0.130482 * 100,3)'(0.1306 * 100,3)
118     tiempo_mano = tiempoMaquina
119 End Sub
```

Ilustración 191 método calcular( ).- CT- 2130.

### 3.1.8.10 CentroTrabajo2140.vb – „HOOKLAP / HOOKER HIGH VOL. C“.

En este centro de trabajo se tuvo que modificar la interface gráfica del form ya que en un principio el sistema solicitaba a usuario ingresara el diámetro nominal del anillo. Ahora el sistema realiza el cálculo directo sin necesidad de que el usuario ingresa algún dato. Para ello es necesario modificar algunas líneas de código y eliminar el código que ya no se utiliza.

Lo primero que hay que hacer es ubicar la función CargarInterfaz( ) y quitar del código el procedimiento que ejecuta la interfaz del centro de trabajo (“Me.ShowDialog( )”) y agregar el método calcular para que realice esta acción en lugar de mostrar el formulario (Ilustración 192).

```
41 | Public Sub CargarInterfaz() Implements ICentroTrabajo.CargarInterfaz
42 |     'Me.ShowDialog()
43 |     calcular()
44 | End Sub
```

**Ilustración 192 procedimiento Sub CargarInterfaz( ).- CT- 2140.**

En seguida hay que eliminar la variable \_d1 del procedimiento pasarValoresDeListas y la variable “diaNominalAnillo” y sus descripciones del procedimiento Sub New( ), como podemos apreciar en la ilustración 193 se ven de color verde las líneas eliminadas.

```
90 | Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString))
91 |     'd1 = module1.obtenerValorDatoDouble(lDatos,"diaNominalAnillo")
92 | End Sub
93 |
94 | Public Sub New()
95 |     Me.InitializeComponent()
96 |     'Dim diaNominalAnillo As New Dato("diaNominalAnillo","Distance","Diámetro nominal del anillo (Plano)",Nothing,"Diámetro nominal del anillo:")
97 |     'listaDatosRequeridos.Add(diaNominalAnillo)
-- |
```

**Ilustración 193 procedimiento Sub pasarValoresDeListas( ) y Sub New( ).- CT- 2140.**

La variable \_d1 se elimina del centro de trabajo, y en la función click del botón calcular se elimina la condicionante If y la igualdad de la variable con el valor del textBox. Posterior a estos cambios hay que ubicar el método calcular( ) y dentro de este la variable tiempo\_maquina para actualizar la fórmula matemática, en la línea 139 se encuentra la nueva fórmula actualizada y al costado derecho de color verde se puede apreciar la fórmula obsoleta utilizada en el centro de trabajo (Ilustración 194).

```

123 '-----Valores que se requieren-----
124 'Dim _d1 As Double
125 '-----
126
127 Sub Btn_calcularClick(sender As Object, e As EventArgs)
128     'If isnumeric(textBox1.Text) Then
129         '_d1 = CDec(textBox1.Text)
130         calcular()
131         Close()
132     'Else
133         'msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
134     'End If
135 End Sub
136
137 Public Sub calcular() implements ICentroTrabajo.calcular
138     tiempo_septup = buscar_setupin(centro)
139     tiempo_maquina = math.Round(0.12600772 * 100,3)'(( 0.08286349 * _d1 ) * 100,3)
140     tiempo_mano = tiempoMaquina * factorLabor
141 End Sub

```

Ilustración 194 función clic del botón calcular y método calcular().- CT- 2140.

### 3.1.8.11 CentroTrabajo2150.vb – „HOOK LAP ASSEMBLY“.

Para realizar la actualización en este centro de trabajo es necesario ubicar el método calcular( ) y actualizar los valores dentro de la fórmula que almacena la variable “tiempo\_maquina”. Como podemos apreciar en la ilustración 195 la nueva fórmula esta en las líneas 134 y 136 respectivamente al tipo de operación seleccionada por el usuario.

```

131 Public Sub calcular() implements ICentroTrabajo.calcular
132     tiempo_septup = buscar_setupin(centro)
133     If _AutomaticMacLarge = True Then
134         tiempo_maquina = math.Round(0.0185935 * 100,3)'(0.0314 * 100,3)
135     Else
136         tiempo_maquina = math.Round(0.0185935 * 100,3)'(0.0219 * 100,3)
137     End If
138     tiempo_mano = tiempoMaquina
139 End Sub

```

Ilustración 195 método calcular().- CT- 2150.

### 3.1.8.12 CentroTrabajo2160.vb – „HOOK LAP DISASSEMBLY“.

Para realizar la actualización en este centro de trabajo es necesario ubicar el método calcular( ) y actualizar los valores dentro de la fórmula que almacena la variable “tiempo\_maquina”. Como podemos apreciar en la ilustración 196 la nueva fórmula está en la línea 117, y de color verde a un costado se puede apreciar la formula obsoleta.



```

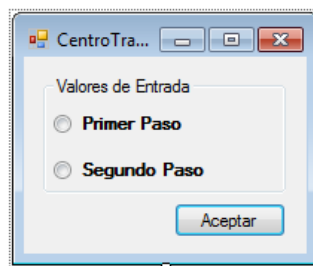
115 | Public Sub calcular() implements ICentroTrabajo.calcular
116 |     tiempo_septup = buscar_setupin(centro)
117 |     tiempo_maquina =Math.Round(0.03893460 * 100,3)' 0.0389 * 100
118 |     tiempo_mano = tiempoMaquina
119 | End Sub

```

**Ilustración 196 método calcular().- CT- 2160.**

### 3.1.8.13 CentroTrabajo2170.vb – „MILL ANG. SEAL-1ST SIDE O.D“.

Para realizar la actualización de este centro de trabajo es necesario modificar la interface del form ya que en un principio este centro de trabajo no solicitaba ningún tipo de información para realizar los cálculos. Ahora es necesario saber cuál paso es que se está realizando.



**Ilustración 197 nueva interfaz gráfica.- CT- 2170.**

Para ello hay que dirigirnos a la parte de código donde se encuentra la función CargarInterfaz( ) y agregar el siguiente código (Me.ShowDialog( )) para que de esta forma se abra el formulario y no solo realice los cálculos (Ilustración 198).

```

41 | Public Sub CargarInterfaz() Implements ICentroTrabajo.CargarInterfaz
42 |     Me.ShowDialog()
43 |     'calcular()
44 | End Sub

```

**Ilustración 198 procedimiento Sub CargarInterfaz().- CT- 2170.**

Después hay que agregar dos nuevas variables de tipo boolean para que almacenen si se seleccionó o no alguna de las opciones indicada por los radio Button (Ilustración 199).

```

119 | '-----Valores que se requieren-----
120 | Dim paso1 As Boolean
121 | Dim paso2 As Boolean
122 | '-----

```

**Ilustración 199 variables nuevas.- CT- 2170.**

Ahora dentro del método calcular( ) se agrega una condicionante If para saber cuál paso se seleccionó para la operación. En caso de seleccionar el paso 1 procederá a ejecutar la nueva fórmula agregada (ilustración 200 línea 127) con los valores para esa operación y en caso de seleccionar el paso2 ejecutará la formula agregada para ese paso (línea 129); como se puede apreciar en la ilustración 187 en la línea 131 se encuentra de color verde la formula obsoleta.

```

123 |
124 | Public Sub calcular() implements ICentroTrabajo.calcular
125 |     tiempo_septup = buscar_setupin(centro)
126 |     If paso1 = True Then
127 |         tiempo_maquina = Math.Round(0.45184603 * 100,3)
128 |     ElseIf paso2 = True
129 |         tiempo_maquina = Math.Round(0.46434603 * 100,3)
130 |     End If
131 |     'tiempo_maquina = 0.5136 * 100
132 |     tiempo_mano = tiempoMaquina
133 | End Sub

```

**Ilustración 200 método calcular( ).-CT- 2170.**

Ahora hay que agregar la función clic del botón calcular donde se igualan las variables agregadas anteriormente con los radioButtons agregados en la interface, se declara el método calcular( ) para que lo ejecute después de haber igualado las variables (Ilustración 201).

```

145 | Sub Button1Click(sender As Object, e As EventArgs)
146 |     paso1 = Me.rbPaso1.Checked
147 |     paso2 = Me.rbPaso2.Checked
148 |     calcular()
149 |     Close()
150 |
151 | End Sub

```

**Ilustración 201 función clic del botón aceptar.- CT- 2170.**

### 3.1.8.14 CentroTrabajo2180.vb – „ENGRAVE“.

Para realizar la actualización en este centro de trabajo es necesario ubicar el método calcular( ) y actualizar los valores dentro de la fórmula que almacena la variable “tiempo\_maquina”. Como podemos apreciar en la ilustración 202 la nueva fórmula está en la línea 117, y de color verde a un costado se puede apreciar la formula obsoleta.

```

115 | Public Sub calcular() implements ICentroTrabajo.calcular
116 |     tiempo_septup = buscar_setupin(centro)
117 |     tiempo_maquina =Math.Round(0.02562550 * 100,3)' 0.0351 * 100
118 |     tiempo_mano = tiempoMaquina
119 | End Sub

```

**Ilustración 202 método calcular( ).- CT- 2180.**

### 3.1.8.15 CentroTrabajo2190.vb – „BATES BORE“.

Para realizar la actualización de este centro de trabajo es necesario modificar la interface gráfica del form ya que para esta operación anteriormente se necesitaba que el usuario ingresara el valor del width nominal y diámetro nominal del anillo; ahora solo es necesario saber el valor del Width nominal.

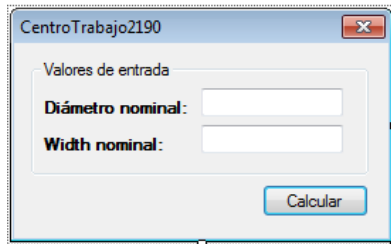


Ilustración 204 form antes de la actualización.- CT- 2190.

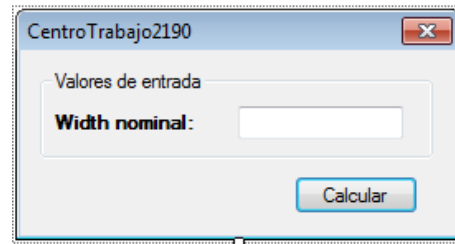


Ilustración 203 form actualizado.- CT- 2190.

En seguida nos dirigimos al procedimiento Sub pasarValoresDeListas( ) y eliminamos del código la variable `_d1` (línea 90), posteriormente procedemos a realizar la misma acción con la variable “`diaNominalAnillo`” del procedimiento Sub New( ), Ilustración 205.

```
89 | Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.  
90 |     _d1 = module1.obtenerValorDatoDouble(lDatos,"diaNominalAnillo")  
91 |     _h1 = module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")  
92 | End Sub  
93 |  
94 | Public Sub New()  
95 |     Me.InitializeComponent()  
96 |     Dim diaNominalAnillo As New Dato("diaNominalAnillo","Distance","Diámetro nominal del anillo (Plano)",Nothing,"Diámetro nominal del anillo:")  
97 |     listaDatosRequeridos.Add(diaNominalAnillo)  
98 |  
99 |     Dim widthNominalAnillo As New Dato("widthNominalAnillo","Distance","Width nominal del anillo (Plano)",Nothing,"Width nominal de anillo:")  
100 |     listaDatosRequeridos.Add(widthNominalAnillo)  
101 |  
102 | End Sub
```

Ilustración 205 procedimiento Sub pasarValoresDeListas.- CT-2190.

Ahora buscamos el método calcular() y ubicamos la variable “`tiempo_maquina`” dentro y procedemos a actualizar la fórmula matemática (Ilustración 206).

```
143 | Public Sub calcular() implements ICentroTrabajo.calcular  
144 |     tiempo_septup = buscar_setupin(centro)  
145 |     tiempo_maquina = math.Round(28.007/(((1 / _h1)* 2)* 36) * 100,3)  
146 |     '((( ( ( ( _d1 * 168.443 ) / 3.0118 ) + 15.28 ) * _h1 ) / 792) * 100,3)  
147 |     tiempo_mano = tiempoMaquina  
148 | End Sub
```

Ilustración 206 método calcular( ).- CT- 2190.

En la función clic del botón calcular se elimina la igualdad de la variable `_d1` (anteriormente eliminada) con el `textBox1` del form y que se oculta dentro del Evento Load del centro de trabajo en cuestión junto con el `label1` (Ilustraciones 207 y 208).

```

132 Sub Btn_calcularClick(sender As Object, e As EventArgs)
133     If isnumeric(textBox2.Text) Then
134         '_d1 = CDec(textBox1.Text)
135         _h1 = CDec(textBox2.Text)
136         calcular()
137         Close()
138     Else
139         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
140     End If
141 End Sub

```

**Ilustración 207 función clic del boton aceptar.- CT- 2190.**

```

160 Sub CentroTrabajo2190Load(sender As Object, e As EventArgs)
161     labell1.Visible = False
162     textBox1.Visible = False
163 End Sub

```

**Ilustración 208 evento Load( ).- CT- 2190.**

### 3.1.8.16 CentroTrabajo2210.vb – „FIN. BORE - HOOK LAP - SPLIT“.

Para realizar la actualización en este centro de trabajo es necesario ubicar el método calcular y dentro del mismo ubicar la variable “tiempo\_maquina” una vez ahí procedemos a modificar la fórmula matemática (línea 137) actualizándola por la nueva fórmula. Como podemos apreciar en la ilustración 209 en la línea 138 se encuentra de color verde la formula obsoleta.

```

135 Public Sub calcular() implements ICentroTrabajo.calcular
136     tiempo_septup = buscar_setupin(centro)
137     tiempo_maquina = math.Round(((200.06 * _h1)/(296.28))*100,3)
138     '(( 7.24805 / ( 8.6 / _h1 ) ) * 100,3)
139     tiempo_mano = tiempoMaquina
140 End Sub

```

**Ilustración 209 método calcular( ).- CT- 2210.**

### 3.1.8.17 CentroTrabajo2260.vb – „VIBRATRON“.

Prácticamente para realizar la actualización en este centro de trabajo solo es necesario ubicar el método calcular () y dentro de este método buscar la fórmula que almacena la variable “tiempo\_maquina“, una vez que la ubiques hay que proceder a actualizar la fórmula matemática. En esta ocasión solo se actualizo un valor dentro de la formula, como se puede apreciar en la ilustración 210 línea 117, a un costado de esta se aprecia de color verde la formula obsoleta.

```

115 |  Public Sub calcular() implements ICentroTrabajo.calcular
116 |         tiempo_septup = buscar_setupin(centro)
117 |         tiempo_maquina = math.Round(0.027242935 * 100,3)'(0.0121 * 100,3)
118 |         tiempo_mano = tiempoMaquina * factorLabor
119 |     End Sub

```

Ilustración 210 método calcular().- CT- 2260.

### 3.1.8.18 CentroTrabajo2270.vb – „TIN PLATE“.

Prácticamente para realizar la actualización en este centro de trabajo solo es necesario ubicar el método calcular () y dentro de este método buscar la fórmula que almacena la variable “tiempo\_maquina“, una vez que la ubiques hay que proceder a actualizar la fórmula matemática. En esta ocasión solo se actualizo un valor dentro de la formula, como se puede apreciar en la ilustración 211 línea 117, a un costado de esta se aprecia de color verde la formula obsoleta.

```

115 |  Public Sub calcular() implements ICentroTrabajo.calcular
116 |         tiempo_septup = buscar_setupin(centro)
117 |         tiempo_maquina = math.Round(0.243140 * 100,3)'(0.2429 * 100,3)
118 |         tiempo_mano = tiempoMaquina * 1
119 |     End Sub

```

Ilustración 211 método calcular().- CT- 2270.

### 3.1.8.19 CentroTrabajo2330.vb – „MILL LAP JOINT - AUT.; MA“.

Prácticamente para realizar la actualización en este centro de trabajo solo es necesario ubicar el método calcular () y dentro de este método buscar la fórmula que almacena la variable “tiempo\_maquina“, una vez que la ubiques hay que proceder a actualizar la fórmula matemática. En esta ocasión solo se actualizo un valor dentro de la formula, como se puede apreciar en la ilustración 212 línea 117, a un costado de esta se aprecia de color verde la formula obsoleta.

```

115 |  Public Sub calcular() implements ICentroTrabajo.calcular
116 |         tiempo_septup = buscar_setupin(centro)
117 |         tiempo_maquina = math.Round(0.13070219 * 100,3)'(0.131 * 100,3)
118 |         tiempo_mano = tiempoMaquina
119 |     End Sub

```

Ilustración 212 método calcular().- CT-2330.

### 3.1.8.20 CentroTrabajo2350.vb – „100% GAP INSPECTION (AUTOGAP)“.

Prácticamente para realizar la actualización en este centro de trabajo solo es necesario ubicar el método calcular () y dentro de este método buscar la fórmula que almacena la variable “tiempo\_maquina“, una vez que la ubiques hay que proceder a actualizar la fórmula matemática. En esta ocasión solo se actualizo un valor dentro de la formula, como se puede apreciar en la ilustración 213 línea 118, a un costado de esta se aprecia de color verde la formula obsoleta.

```
116 | □ Public Sub calcular() implements ICentroTrabajo.calcular
117 |     tiempo_septup = buscar_setupin(centro)
118 |     tiempo_maquina = math.Round(0.05428406 * 100,3)'(0.0453 * 100,3)
119 |     tiempo_mano = tiempoMaquina
120 | End Sub
```

Ilustración 213 método calcular().- CT- 2350.

## Resultados

Mi participación dentro de Mahle componentes de motor México se desarrolló implementando mis conocimientos de programación en el departamento de ingeniería de rutas que es donde manejan el sistema "ROUTING GENERATION PROGRAM" para el cálculo de hojas de ruta para anillos, expansores, segmentos, franklin y large bore dentro de este departamento se realizaron varias actividades para el logro de los objetivos propuestos en esta investigación de las cuales podemos mencionar las siguientes:

- Reingeniería y actualizaciones al sistema RGP en general.
- Actualización a la forma del cromo en los centros de trabajo.
- Revisión y actualización de las fórmulas para calcular el tiempo estándar con las que actualmente cuenta el sistema.
- Actualización de varias tablas dentro de la base de datos.
- Reducción del tiempo que se llevan en realizar las cotizaciones de herramientas para el cálculo de las hojas de ruta.
- Revisión de la información y formulas existentes para el cálculo del tiempo estándar.
- Creación de nuevas interfaces de centros de trabajo para calcular el tiempo estándar de los componentes requeridos de manera automática.
- Se aplicó reingeniería al sistema codificando un nuevo método para extraer y exportar la tabla de placas modelo a un nuevo archivo de Excel.
- Manuales para la actualización de las formulas del tiempo estándar en el sistema "ROUTING GENERATION PROGRAM"

## Conclusiones

Al cumplir satisfactoriamente con los objetivos planteados en este proyecto se concluye que las fórmulas necesarias para realizar el cálculo del tiempo estándar quedaron revisadas y actualizadas en el sistema "ROUTING GENERATION PROGRAM" reduciendo significativamente los tiempos de las cotizaciones de herramientas para los cálculos de las hojas de ruta.

Durante el periodo que desarrollo este proyecto quedaron al descubierto varios problemas con los que contaba el sistema, por tal motivo se tuvieron repartir responsabilidades para poder mantener actualizado el sistema y la información de cada componente creado.

Recomendaciones.

La actualización periódica de los tiempos estándar para el cálculo y cotizaciones de herramental en las hojas de ruta, permitirá que la empresa se mantenga en el camino correcto durante su proceso de mejora continua.

Los conocimientos básicos de programación son realmente necesarios dentro de la organización, ya que no cuentan con un programador de base y la mayoría de los sistemas que utilizan necesitan de actualizaciones o modificaciones y el personal con el que cuentan en ocasiones no tienen ni la más mínima idea de cómo realizar esas actualizaciones.

En este trabajo se ve reflejada la importancia de contar con un desarrollador en la planta, ya que dentro de la organización manejan varios sistemas ERP que están conectados al SAP y en su mayoría estos sistemas necesitan de una constante actualización o modificación para que puedan seguir operando conforme se van actualizando las operaciones dentro de la planta, y esta pueda seguir operando bajo sus procesos de mejora continua.



## **Competencias Desarrolladas.**

Durante la realización de este proyecto obtuve la capacidad de analizar, diseñar, desarrollar y mantener proyectos y aplicaciones de forma segura y eficaz, conocimientos de los procedimientos y algoritmos básicos del entorno .NET, de los lenguajes VB, y C#. Aprendí a utilizar el IDE "SharpDevelop".

## Programa de actividades Cronograma de actividades

Actividades por Quincena	Ago-1a	Ago-2a	Sept – 1a	Sept – 2a	Oct – 1a	Oct-2a	Nov – 1a	Nov. – 2a	Dic-1a
Actualizar Forma de Cromo.									
Agregar Listas de Herramentales a la BD.									
Agregar costo de nuevos Herramentales a la BD.									
Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – ANILLOS.									
Modificación a interfaces de centros de trabajo – ANILLOS.									
Generación de nuevas interfaces para los centros de trabajo faltantes en el área de ANILLOS.									
Elaboración del manual de la Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – ANILLOS.									
Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – LARGE BORE.									
Modificación a interfaces de centros de trabajo – LARGE BORE.									
Generación de nuevas interfaces para los centros de trabajo faltantes en el área de LARGE BORE.									
Elaboración del manual de la Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – LARGE BORE.									
Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – EXPANSORES.									
Modificación a interfaces de centros de trabajo – EXPANSORES.									
Generación de nuevas interfaces para los centros de trabajo faltantes en el área de EXPANSORES.									
Elaboración del manual de la Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – EXPANSORES.									
Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – FRANKLIN.									
Modificación a interfaces de centros de trabajo – FRANKLIN.									
Generación de nuevas interfaces para los centros de trabajo faltantes en el área de FRANKLIN.									
Elaboración del manual de la Actualización de las fórmulas del Tiempo Estándar para los Centros de Trabajo – FRANKLIN.									
Corregir Errores Encontrados en el Sistema.									

## Referencias

- 1983, I. S. (s.f.). IEEE Standard Glossary Software Engineering Terminology. ANSI.
- Canales de Información de México. (2014). Mahle Componentes de Motor de México S.A. de C.V. (C. E. Pérez, Ed.) *Gestión y Desarrollo Nacional*, 46, 47.
- Carmen de Palos Heredero, Lopez Hemoso Agius, Martin Romo Romero, Santiago, Sonia Medina Salgado. (2004). *Informatica y Comunicaciones en la empresa*. Madrid: Esic.
- Escuela de Administración de Negocios, U. d. (22 de 11 de 2016). *tiempos estandar - ucreanop*. Obtenido de tiempos estandar - ucreanop:  
<http://ucreanop.org/descargas/Lecturas/Tiempos%20Estandar.pdf>
- Héctor Francisco Bautista Gonzalez, Cuauthemoc Freyre Mercado, Norma Susana Zavala Carrasco. (1995). Reingeniería de Programación: Una propuesta para la solución de los problemas de mantenimiento de software. *Tesis de Ingeniero en Computacion*. México, México: UNAM.
- Hernández, Z. T. (2014). *Teoría General de la Administración* (2 da ed.). CD. de México: Grupo Editorial Patria.
- Juárez López, F. J. (Octubre de 2011). Migración ERP-SYTELINE / ERP-SAP. *Proyecto de Tesis "Migración ERP-SYTELINE / ERP-SAP"*. Aguascalientes, Aguascalientes, México: Instituto Tecnológico de Aguascalientes.
- LIDER Integrated Technology Consulting S.A. (19 de Noviembre de 2016). *¿Qué es un ERP? - Andece*. Obtenido de ¿Qué es un ERP? - Andece:  
[http://www.andece.org/adheridos/images/stories/LIDER\\_IT/Qu-es-un-ERP.pdf](http://www.andece.org/adheridos/images/stories/LIDER_IT/Qu-es-un-ERP.pdf)
- Lopez Hermoso, J. J. (2000). *Informatica aplicada a la gestión de las Empresas*. Madrid: Esic.
- López, B. S. (19 de Noviembre de 2016). *Aplicación del Tiempo Estándar - Ingeniería Industrial*. Obtenido de Aplicación del Tiempo Estándar:  
<http://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/estudio-de-tiempos/aplicaci%C3%B3n-del-tiempo-est%C3%A1ndar/>
- López, B. S. (19 de Noviembre de 2016). *Cálculo del Tiempo Estándar o Tipo - Ingeniería Industrial*. Obtenido de Ingeniería Industrial:  
<http://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/estudio-de-tiempos/c%C3%A1lculo-del-tiempo-est%C3%A1ndar-o-tipo/>
- Muñiz, L. (2000). *ERP: Guía práctica para la selección e Implementación. Enterprise Resource Planning o Sistema de Planificación de Recursos Empresariales*. Barcelona: Gestión.
- Olgín, H. (1997). *ORGANIZACIÓN Y ADMINISTRACIÓN DE CENTROS DE COMPUTO* (1 ed.). México, Ciudad Universitaria, D.F.: UNAM, Facultad de Ingeniería.
- Rojano, M. F. (Diciembre de 2011). *GESTIÓN DE PROCESOS PARA INCREMENTAR LA CAPACIDAD DE PRODUCCIÓN EN EL ÁREA DE TINTURADO Y ACABADOS EN LA EMPRESA TEXTIL ANDELAS CÍA. LTDA. Tesis GESTIÓN DE PROCESOS PARA INCREMENTAR LA CAPACIDAD DE PRODUCCIÓN EN EL ÁREA DE TINTURADO Y ACABADOS EN LA EMPRESA TEXTIL ANDELAS CÍA. LTDA*. Ambato, Ecuador: Universidad Técnica de Ambato.

Vásquez, M. E. (Diciembre de 2011). Los Beneficios De Implementar un Sistema ERP en las Empresas Colombianas- Estudio de Caso. *Tesis- Los Beneficios De Implementar un Sistema ERP en las Empresas Colombianas- Estudio de Caso*. Bogotá, Colombia: Universidad del Rosario.