

2018



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO
de Pabellón de Arteaga

TEC

Instituto Tecnológico de Pabellón de Arteaga

SUSTENTANTE:

DAVID GIOVANNI ALVARADO MUÑOZ

NOMBRE DE LA EMPRESA:

MAHLE COMPONENTES PARA MOTOR DE MÉXICO S. DE R.L. DE CV.

NOMBRE DEL PROYECTO:

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

NOMBRE DEL ASESOR INTERNO:

M.I.T.C RAFAEL PRECIADO GUTIERREZ

NOMBRE DEL ASESOR EXTERNO:

JORGE HUMBERTO RODRÍGUEZ BARRIETOS

Agradecimientos

Primero antes que nada quiero agradecer a mis padres Ma. de los Ángeles Muñoz Robles y Guillermo Alvarado Esquivel por apoyarme incondicionalmente moral y económicamente en toda mi carrera. En especial a mi madre por enseñarme a nunca rendirme y seguir adelante a pesar de todas las adversidades.

A mis hermanos por el apoyo que siempre me brindaron día a día en el transcurso de toda mi carrera.

A mis maestros por dar siempre lo mejor de ellos, por sus consejos, por sus enseñanzas y por haberme dado oportunidades de crecimiento tanto personales como académicas y profesionales.

Al Lic. Jorge López Flores jefe de Recursos Humanos, por darme la oportunidad de haber desarrollado mis residencias profesionales en Mahle.

Así como también al Ing. Jorge Humberto Rodríguez Barrientos por la confianza que depositó en mí, por lo que me apoyo a lo largo de este tiempo y porque me permitió aprender de su experiencia y conocimientos.

Resumen

Se realizó la actualización de las fórmulas para calcular el tiempo estándar de cada uno de los centros de trabajo que están en funcionamiento en la planta.

También se agregaron nuevos centros que no estaban dados de alta en el sistema RGP, pero que ya están en funcionamiento en la planta.

Además de que se repararon varios errores dentro del software RGP que los usuarios del departamento de ingeniería de rutas ya tenían identificados, e impedían el buen funcionamiento del software.

Además se creó la documentación de cómo se realizó la actualización de cada uno de los centros de trabajo.

Índice

Lista de tablas.....	6
Lista de figuras.....	6
1. Introducción.....	15
1.1 Antecedentes.....	15
1.2 Organigrama funcional de Mahle componentes para motor de México.....	17
1.3 Organigrama de ingeniería.....	17
2. Planteamiento del problema.....	18
3. Objetivos.....	18
3.1 Objetivo general.....	18
3.2 Objetivos específicos.....	18
4. Justificación.....	19
5. Marco teórico.....	19
5.1 ¿Qué es un ERP?.....	19
5.1.1 El propósito fundamental de un ERP.....	20
5.1.2 Ventajas de los ERP.....	20
5.1.3 Componentes del ERP.....	21
5.2 SAP (System Applications and Products).....	21
5.2.1 ¿Qué son los módulos SAP?.....	23
6. Metodología.....	24
6.1 Actualización de la materia prima de los CT 715, 9361 y 9362.....	24
6.1.1 Centro de trabajo 715.....	24
6.1.2 Centro de Trabajo 9361.....	26
6.1.3 Centro de trabajo 9362.....	28
6.2 Actualización de tiempos estándar – Anillos.....	32
6.2.1 CentroTrabajo110.vb – FIRST ROUGH GRIND.....	32
6.2.2 CentroTrabajo112.vb – SECOND ROUGH GRIND.....	34
6.2.3 CentroTrabajo132.vb – SPLITTER CUFFS.....	36
6.2.4 CentroTrabajo140.vb – DEGREASE RINGS.....	40
6.2.5 CentroTrabajo150.vb – FIN. GRD.....	43
6.2.6 CentroTrabajo230.vb – ROUGH CAM TURN.....	45
6.2.7 CentroTrabajo232.vb – ROUGH CAM TURN (HIGH RUNNERS).....	47

6.2.8	CentroTrabajo235.vb – CAM TURN BORE MILL (CTB - 170 MACHINE)	53
6.2.9	CentroTrabajo285.vb – BATES BORE (HIGH RUNNERS)	56
6.2.10	CentroTrabajo421.vb – CNC AUTO. FIN TURN	59
6.2.11	CentroTrabajo423.vb – KATAOKA (ANILLOS THM)	59
6.2.12	CentroTrabajo433.vb – SIMPLEX	61
6.2.13	CentroTrabajo435.vb – GRIND CHANNEL (NORTON)	62
6.2.14	CentroTrabajo440.vb – TAPER SIDE	65
6.2.15	CentroTrabajo460.vb – WASH	67
6.2.16	CentroTrabajo491.vb – O.D. SHEFFIELD GASOLINE	69
6.2.17	CentroTrabajo496.vb – STRESS RELIEF FOR STEEL COMP. RINGS	72
6.2.18	CentroTrabajo715.vb – O.D. CHROME PLATE RINGS	74
6.2.19	CentroTrabajo750.vb – PHOSPHATE GASOLINE	75
6.2.20	CentroTrabajo755.vb – FERROX FILL GASOLINE	76
6.2.21	CentroTrabajo757.vb – HEAD TREAT GASOLINE	77
6.3	Actualización de tiempos estándar – Large Bore	78
6.3.1	CentroTrabajo9226.vb – I.D. GROOVE L.B.	78
6.3.2	CentroTrabajo9227.vb – GROOVE FERROX L.B.	81
6.3.3	CentroTrabajo9243.vb – LAP. O.D. L.B.	82
6.3.4	CentroTrabajo9244.vb – DEGREASE RINGS	84
6.3.5	CentroTrabajo9261.vb – CROP SAW L.B.	85
6.3.6	CentroTrabajo9262.vb – CNC CUFF TO RING & INS L.B.	85
6.3.7	CentroTrabajo9263.vb – CNC C'BORE & PRETTERN L.B.	89
6.3.8	CentroTrabajo9266.vb – TURN O.D., I.D. CUFF L.B.	91
6.3.9	CentroTrabajo9285.vb – ABRASIVE WHEEL L.B.	94
6.3.10	CentroTrabajo9301.vb – FINISH TURN CELL #1 L.B.	96
6.3.11	CentroTrabajo9302.vb – Finish Turn Cell #2	99
6.3.12	CentroTrabajo9305.vb – GAP NOTCH IDENTIFICATION L.B.	100
6.3.13	CentroTrabajo9329.vb – LASER MARK L.B.	102
6.3.14	CentroTrabajo9341.vb – SIDE PLATE L.B.	104
6.3.15	CentroTrabajo9361.vb – CHROME PLATE I.D. & O.D. L.B.	105
6.3.16	CentroTrabajo9363.vb – FERROX FILL L.B.	108
6.3.17	CentroTrabajo9364.vb – FERROX FURN L.B.	108

6.3.18	CentroTrabajo9381.vb – MANGANESE PHOSPHATE L.B.	110
6.4	Actualización de tiempos estándar – Segmentos	111
6.4.1	CentroTrabajo456.vb – Lapping segments	111
6.4.2	CentroTrabajo466.vb – Scotchbrite Segments	112
6.4.3	CentroTrabajo497.vb – Nitrurado de segmentos	113
6.4.4	CentroTrabajo498.vb – Barrell grade segments	115
6.4.5	CentroTrabajo510.vb – Bobinado de segmentos	116
6.4.6	CentroTrabajo514.vb – Desengrase bobinado segmentos	117
6.4.7	CentroTrabajo517.vb – Degrease segments	118
6.4.8	CentroTrabajo520.vb – Normalizado	120
6.4.9	CentroTrabajo550.vb – Cut off vulcan	121
6.4.10	CentroTrabajo555.vb – Cut off Thompson cromado	122
6.4.11	CentroTrabajo558.vb – Gap side brusher	124
6.4.12	CentroTrabajo716.vb – OD Chrome plate segments	125
6.4.13	CentroTrabajo717.vb – ID Chrome plate segments	127
6.4.14	CentroTrabajo720.vb – Blackening	129
6.4.15	CentroTrabajo831.vb – Final insp. Segments	132
6.5	Actualización de tiempos estándar – FRANKLIN	133
6.5.1	CentroTrabajo230.vb – ROUGH CAM TURN (FRANKLIN)	133
6.5.2	CentroTrabajo2000.vb – FIRST ROUGH GRIND (FRANKLIN)	135
6.5.3	CentroTrabajo2060.vb – SPLITTER CUFF (FRANKLIN)	137
6.5.4	CentroTrabajo2195.vb – FINISH MILL (FRANKLIN)	139
6.5.5	CentroTrabajo2230.vb – DUPLEX (FRANKLIN)	140
6.5.6	CentroTrabajo2240.vb – PICK (FRANKLIN)	140
6.5.7	CentroTrabajo2250.vb – I.D. BRUSH (FRANKLIN)	142
6.5.8	CentroTrabajo2255.vb – SCOTCHBRITE RINGS (FRANKLIN)	142
6.5.9	CentroTrabajo2290.vb – PHOSPHATE GASOLINE	143
6.5.10	CentroTrabajo2295.vb – VISUAL INSPECTION (FRANKLIN)	145
7.	Resultados	146
8.	Conclusiones y recomendaciones	147
8.1	Conclusiones	147
8.2	Recomendaciones	147

9. Referencias	147
-----------------------------	-----

Lista de tablas

Tabla 1. Tabla para calcular la cantidad de materia prima.	29
Tabla 2. Tabla de resultados de la cantidad de materia prima a usar.	31
Tabla 3. Tabla de materiales	35
Tabla 4. Tabla de datos correspondiente a los materiales.	39
Tabla 5. Tabla relación diámetro-rieles.....	41

Lista de figuras

Figura 1 Directorio de las clases de los centros de trabajo.....	24
Figura 2 Clase CentroTrabajo715.vb.	25
Figura 3 Método calcular del CT 715 donde se agrega la materia prima.	25
Figura 4 Método calcular_cromo_kg.	26
Figura 5 Variable ca de tipo objeto.	26
Figura 6 Variable tipo MateriaPrimaGenerica.	26
Figura 7 Método calcular del CT 9361 donde se agrega la materia prima.	27
Figura 8 Formulario del CT 9361.	27
Figura 9 Variables de la clase CentroTrabajo9361.	27
Figura 10 Variable ca de tipo objeto.	28
Figura 11 Método calcular_cromo_kg de la clase CalculoGramosCromoMoly.vb.	28
Figura 12 Nuevas líneas de código.	29
Figura 13 Variables del CT 9362.	30
Figura 14 Formulario del CT 9362.	30
Figura 15 Código para igualar una variable con una celda de Excel.	30
Figura 16 Variables tipo MateriaPrimaGenerica del CT 9362.	31
Figura 17 Formulario del CT 110.	32
Figura 18 if que obtiene los tiempos ciclo.	33
Figura 19 Obteniendo la operación para sacar los tiempos ciclo.	33
Figura 20 Variable ciclo por carga.	34
Figura 21 Formula actualizada para calcular el tiempo estándar del CT 110.	34
Figura 22 Formulario del CT 112.	34
Figura 23 Condicional para obtener el tiempo ciclo.	35
Figura 24 Valores para obtener el ciclo por carga.	35
Figura 25 Variable ciclo por carga.	36
Figura 26 Declaración de variables.	36
Figura 27 Formula actualizada.	36
Figura 28 Formulario correspondiente al CT132.....	36
Figura 29 Creando el formulario de la clase.....	37
Figura 30 Variables donde se almacenaran los diferentes valores.....	37

Figura 31 Igualando variables con parámetros.....	37
Figura 32 Variables que se agregan a la lista de datos requeridos.....	38
Figura 33 Pasando valores de listas.....	38
Figura 34 Extrayendo los materiales de la base de datos.....	38
Figura 35 Variables que se deben de declarar antes del método TEST.....	39
Figura 36 Método llenar material.....	39
Figura 37 Método calcular del CT 132.....	40
Figura 38 Formula actualizada para calcular el tiempo estándar.....	40
Figura 39 Interfaz gráfica del centro de trabajo.....	41
Figura 40 Error al insertar un valor de diámetro fuera de rango.....	41
Figura 41 Función que obtiene sacar el número de rieles de acuerdo al valor del diámetro.....	42
Figura 42 Método calcular del CT 140.....	42
Figura 43 Formula actualizada del CT 140.....	42
Figura 44 Evento clic del botón del formulario.....	43
Figura 45 Formulario correspondiente al CT 150.....	43
Figura 46 Condicional donde se obtiene el tiempo ciclo de acuerdo al material.....	44
Figura 47 Forma de obtener la operación para calcular el tiempo ciclo 1 y 2.....	44
Figura 48 Formula actualizada para el cálculo del tiempo estándar correspondiente al CT 150.....	44
Figura 49 Formulario antes de la actualización.....	45
Figura 50 Formulario actual.....	45
Figura 51 Constructor donde se agregan los valores a las listas.....	45
Figura 52 Método pasarValoreDeListas() del CT 230.....	45
Figura 53 Valores de entrada agregados a las listas de datos requeridos.....	46
Figura 54 Constructor y parámetros actuales de la clase.....	46
Figura 55 Variables donde se almacenan los valores de entrada.....	46
Figura 56 Método calcular() del CT 230.....	46
Figura 57 Evento clic del botón del formulario.....	47
Figura 58 Agregar Nuevo Centro de Trabajo. - CT-232.....	47
Figura 59 Agregar Centro de Trabajo. - CT-232.....	48
Figura 60 Imports Logica.....	48
Figura 61 Formulario del CT 232.....	49
Figura 62 Propiedades de la interface ICentroTrabajo.....	50
Figura 63 Variables donde se almacenan los valores de entrada.....	50
Figura 64 Variables necesarias para la función Test.....	51
Figura 65 Función Test del CT 232.....	51
Figura 66 Método llenar_material().....	51
Figura 67 Método calcular del CT 232 con su fórmula actualizada.....	52
Figura 68 Método buscar_septup().....	52
Figura 69 Evento clic del botón del formulario.....	52
Figura 70 Formulario actualizado del CT 235.....	53
Figura 71 Formulario antes de la actualización.....	53
Figura 72 Función Test para agregar los materiales al centro de trabajo.....	53

Figura 73 Variables necesarias para la función Test.	54
Figura 74 Variables necesarias para almacenar los valores que el usuario ingrese al formulario....	54
Figura 75 Agregando un nuevo parámetro al constructor de la clase.	54
Figura 76 Agregando el elemento material a la ListaDatosStringRequeridos.	55
Figura 77 Método publico pasarValoresDeListas.	55
Figura 78 Método llenar material.	55
Figura 79 Método CargarInterfaz en el cual se manda a llamar el método llenar_material.	56
Figura 80 Condición if para obtener el valor del tiempo ciclo.	56
Figura 81 Formula actualizada para calcular el tiempo estándar del CT 235.	56
Figura 82 Formulario del CT 285.	57
Figura 83 Variable donde se almacena el valor de entrada.	57
Figura 84 Igualando variables con parámetros.	57
Figura 85 Variables que se agregan a la lista de datos requeridos.	58
Figura 86 Pasando valores de listas.	58
Figura 87 Método calcular del CT 285.	58
Figura 88 Evento clic del botón del formulario.	58
Figura 89 Formulario antes de la eliminación de sus elementos.	59
Figura 90 Asignando valor al tiempo ciclo.	59
Figura 91 Formula actualizada del CT 421.	59
Figura 92 Formulario actual del CT 423.	60
Figura 93 Variables correspondientes a cada valor de entrada.	60
Figura 94 Pasando valores a la interface ICentroTrabajo del proyecto Logica.	60
Figura 95 Método calcular() donde se lleva a cabo la operación para obtener el tiempo estándar.	61
Figura 96 Evento clic del botón del formulario.	61
Figura 97 Formulario del CT 433.	61
Figura 98 Método calcular y formula del tiempo estándar del CT 433.	61
Figura 99 Formulario antes de la actualización.	62
Figura 100 Formulario actualizado.	62
Figura 101 Variables donde se almacenan los diferentes valores.	62
Figura 102 Igualando variables con parámetros.	62
Figura 103 Variables que se agregan a la lista de datos requeridos.	63
Figura 104 Pasando valores de listas.	63
Figura 105 Variables que se deben de declarar antes del método Test.	63
Figura 106 Agregando materiales a la función Test.	63
Figura 107 Método llenar_material().	64
Figura 108 Método CargarInterfaz() del CT 435.	64
Figura 109 Método calcular donde se encuentra la fórmula para el cálculo del tiempo estándar.	64
Figura 110 Evento clic del botón del formulario del CT 435.	65
Figura 111 Formulario antes de la actualización.	65
Figura 112 Formulario actual.	65
Figura 113 Constructor de la clase actual.	66
Figura 114 Método pasarValoresDeListas() y listaDatosRequeridos actuales.	66

Figura 115 Método calcular() del CT 440.	66
Figura 116 Evento clic del botón del formulario.	67
Figura 117 Formulario correspondiente al CT 460.	67
Figura 118 Variables donde se almacenaran los valores de entrada.	67
Figura 119 Igualando parámetros con variables.	68
Figura 120 Variables que se agregan a la lista de datos requeridos.	68
Figura 121 Pasando valores de listas.	68
Figura 122 Función sacar_num_rieles().	68
Figura 123 Método calcular donde se encuentra la fórmula del tiempo estándar.	69
Figura 124 Evento clic del botón del formulario.	69
Figura 125 Formulario antes de la actualización.	70
Figura 126 Formulario actualizado.	70
Figura 127 Variables donde se almacenan los valores de entrada.	70
Figura 128 Método calcular() del CT 491.	70
Figura 129 Evento clic del botón del formulario.	71
Figura 130 Formulario antes de la actualización.	72
Figura 131 Formulario actualizado.	72
Figura 132 Variables donde se almacenan los valores de entrada.	72
Figura 133 Método sacar_tamaño_de_carga_por_pino().	72
Figura 134 Método calcular del CT 496.	73
Figura 135 Evento clic del botón del formulario.	73
Figura 136 Formulario antes de la actualización.	74
Figura 137 Formulario actualizado.	74
Figura 138 Formula del tiempo estándar del método calcular() (es la misma para el método calcular2()).	74
Figura 139 Formulario del centro de trabajo 750.	75
Figura 140 Método cuantos_bastidores().	75
Figura 141 Método calcular del CT 750.	76
Figura 142 Evento clic del botón del formulario.	76
Figura 143 Formulario del CT 755.	77
Figura 144 Método calcular() del CT 755.	77
Figura 145 Formulario del CT 757.	77
Figura 146 Método calcular() del CT 757.	78
Figura 147 Formulario del CT 9226.	78
Figura 148 Variables necesarias para la clase.	79
Figura 149 Agregando nuevos parámetros al constructor de la clase.	79
Figura 150 Pasando valores de entrada a las listas de datos requeridos.	79
Figura 151 Pasando valores de listas.	79
Figura 152 Agregando lista de materiales al método Test de la clase.	80
Figura 153 Método llenar_material().	80
Figura 154 Método calcular del CT 9226.	81
Figura 155 Evento clic del botón del formulario.	81

Figura 156 Formulario del CT 9227.	82
Figura 157 Método calcular() del CT 9227.	82
Figura 158 Formulario antes de la modificación.	83
Figura 159 Formulario después de la actualización.	83
Figura 160 Variables correspondientes a los valores de entrada.	83
Figura 161 Método calcular correspondiente al CT 9243.	84
Figura 162 Método calcular() del CT 9244.	84
Figura 163 Formulario del CT 9261.	85
Figura 164 Método calcular() del CT 9261.	85
Figura 165 Formulario del CT 9262.	86
Figura 166 Interfaz del formulario al seleccionar la operación Bardon 1 y Okuma 1.	86
Figura 167 Evento CheckedChanged de los radiobuttons del formulario.	86
Figura 168 Variables necesarias para almacenar los valores de entrada.	87
Figura 169 Código del evento clic del botón del formulario.	87
Figura 170 Formula correspondiente a la operación Bardon 1 del CT 9262.	87
Figura 171 Formula correspondiente a la operación Okuma 1 del CT 9262.	88
Figura 172 Código correspondiente a la operación Bardon 2 del CT 9262.	88
Figura 173 Método llenar_proceso().	89
Figura 174 Formulario antes de la actualización.	89
Figura 175 Formulario actualizado.	89
Figura 176 Evento CheckedChanged correspondiente a los radiobutton.	89
Figura 177 Variables necesarias para almacenar los valores de entrada.	90
Figura 178 Código del evento clic del botón del formulario.	90
Figura 179 Método llenar_proceso().	90
Figura 180 Código de la operación Okuma 2.	91
Figura 181 Código de la operación Cincinnati.	91
Figura 182 Formulario antes de la actualización.	92
Figura 183 Formulario actualizado.	92
Figura 184 Evento CheckedChanged de los radiobuttons del formulario.	92
Figura 185 Variables que se necesitan para almacenar los valores de entrada.	93
Figura 186 Evento clic del botón del formulario.	93
Figura 187 Código de la operación Puma del CT 9266.	94
Figura 188 Código de la operación Puma 240 del CT 9266.	94
Figura 189 Formulario del CT 9285.	95
Figura 190 Método calcular() del CT 9285.	95
Figura 191 Método del evento clic del botón del formulario.	96
Figura 192 Formulario antes de la actualización.	96
Figura 193 Formulario actualizado.	96
Figura 194 Evento ChekedChanged de los radiobuttons del formulario.	97
Figura 195 Variables donde se almacenan los valores de entrada.	97
Figura 196 Código del evento clic del botón del formulario.	97
Figura 197 Código para obtener el tiempo estándar de la operación Finish Turn Cell #1.	98

Figura 198 Código para obtener el tiempo estándar de la operación Super Finish.....	98
Figura 199 Formulario antes de la actualización.....	99
Figura 200 Formulario actualizado.....	99
Figura 201 Variables donde se almacenan los diferentes valores de entrada.	99
Figura 202 Método calcular() del CT 9302.	100
Figura 203 Formulario del CT 9305.	100
Figura 204 Función Test del CT 9305.	101
Figura 205 Parámetros del constructor de la clase.....	101
Figura 206 Agregando valores de entrada a la listaDatosStringRequeridos.....	101
Figura 207 Pasando valores de listas.	101
Figura 208 Método del evento clic del botón del formulario.	102
Figura 209 Método calcular del CT 9305.....	102
Figura 210 Formulario antes de la actualización.....	103
Figura 211 Formulario actualizado.....	103
Figura 212 Método del evento clic del formulario.....	103
Figura 213 Método calcular del CT 9329.....	104
Figura 214 Formulario antes de la actualización.....	104
Figura 215 Formulario actualizado.....	105
Figura 216 Cálculos para obtener el tiempo estándar del CT 9341.	105
Figura 217 Formulario antes de la actualización.....	106
Figura 218 Formulario actualizado.....	106
Figura 219 Variables donde se almacenan los valores de entrada.	106
Figura 220 Evento CheckedChanged de los radiobuttons.	106
Figura 221 Código del evento clic del botón del formulario.	107
Figura 222 Método calcular del CT 9361.....	107
Figura 223 Formulario del CT 9363.	108
Figura 224 Método calcular() del CT 9363.	108
Figura 225 Formulario del CT 9364.	108
Figura 226 Parámetros que contiene el constructor de la clase.....	109
Figura 227 Agregando valores de entrada a ListaDatosRequeridos.	109
Figura 228 Pasar valores de listas.	109
Figura 229 Método calcular() del CT 9364.	109
Figura 230 Evento clic del botón del formulario.	109
Figura 231 Formulario del CT 9381.	110
Figura 232 Constructor y parametros del CT 9381.	110
Figura 233 Método pasarValoresDeListas() y listaDatosRequeridos.	110
Figura 234 Método calcular() del CT 9381.	111
Figura 235 Formulario del CT 456.	111
Figura 236 Método calcular() del CT 456.	112
Figura 237 Formulario del CT 466.	112
Figura 238 Método calcular() del CT 466.	113
Figura 239 Formulario antes de la actualización.....	113

Figura 240 Formulario actual.	113
Figura 241 Variables y parámetros del CT 497.....	114
Figura 242 listaDatosBooleanosRequeridos y listaDatosRequeridos.....	114
Figura 243 Método pasarValoresDeListas().	114
Figura 244 Método calcular() del CT 497.	115
Figura 245 Evento clic del botón del formulario.	115
Figura 246 Formulario del CT 498.	116
Figura 247 Método calcular() del CT 498.	116
Figura 248 Formulario del CT 510.	116
Figura 249 Método calcular() del CT 510.	117
Figura 250 Formulario del CT 514.	117
Figura 251 Método calcular() del CT 514.	117
Figura 252 Formulario antes de la actualización.....	118
Figura 253 Formulario actual.	118
Figura 254 Variables necesarias y parámetros del constructor del CT 517.	118
Figura 255 Variables Tipo Dato y DatoBoolean.....	119
Figura 256 Método pasarValoresDeListas().	119
Figura 257 Método sacar_rieles_hurricane().	119
Figura 258 Método calcular() del CT 517.	119
Figura 259 Evento clic del botón del formulario.	120
Figura 260 Formulario del CT 520.	120
Figura 261 Función buscar_tiempo() anterior.	121
Figura 262 Función buscar_tiempo() actual.....	121
Figura 263 Método calcular() del CT 520.	121
Figura 264 Formulario del CT 550.	122
Figura 265 Método calcular() del CT 550.	122
Figura 266 Formulario antes de la actualización.....	122
Figura 267 Formulario actual.	123
Figura 268 Variables y parámetros necesarios para calcular el tiempo estándar.	123
Figura 269 ListaDatosBooleanosRequeridos y ListaDatosRequeridos.	123
Figura 270 Método pasarValoresDeListas().	124
Figura 271 Método calcular() del CT 555.	124
Figura 272 Método calcular() del CT 558.	125
Figura 273 Formulario antes de la actualización.....	125
Figura 274 Formulario actual.	125
Figura 275 Variables y constructor de la clase.....	126
Figura 276 ListaDatosBooleanosRequeridos.....	126
Figura 277 Método pasarValoresDeListas.....	126
Figura 278 Método calcular() del CT 716.	126
Figura 279 Evento clic del botón del formulario.	127
Figura 280 Formulario antes de la actualización.....	127
Figura 281 Formulario actual.	127

Figura 282 Variables y parámetros del CT 717.....	128
Figura 283 Metodo pasarValoresDeListas().	128
Figura 284 Método calcular() del CT 717.	128
Figura 285 Evento clic del botón del formulario.	129
Figura 286 Formulario antes de la actualización.....	129
Figura 287 Formulario actual.	129
Figura 288 Variables y parámetros del CT 720.....	130
Figura 289 Listas de datos requeridos.	130
Figura 290 Método pasarValoresDeListas del CT 720.....	130
Figura 291 Función sacar_rieles().	131
Figura 292 Método calcular() del CT 720.	131
Figura 293 Evento clic del botón del formulario.	132
Figura 294 Formulario del CT 831.	132
Figura 295 Método calcular() del CT 831.	133
Figura 296 Formulario actual del CT 230.	133
Figura 297 Variables del CT 230.	134
Figura 298 Variables tipo Dato, DatoBoolean y DatoString.	134
Figura 299 Método pasarValoresDeListas().	134
Figura 300 Método calcular() del CT 230.	135
Figura 301 Formulario antes de la actualización.....	135
Figura 302 Formulario actual.	135
Figura 303 Variables y parámetros del CT 2000.....	136
Figura 304 Variable tipo Dato y listaDatosRequeridos.	136
Figura 305 Método pasarValoresDeListas del CT 2000.....	136
Figura 306 Método calcular() del CT 2000.	136
Figura 307 Evento clic del botón del formulario.	137
Figura 308 Formulario antes de la actualización.....	137
Figura 309 Formulario actual.	137
Figura 310 Variables y parámetros necesarios del CT 2060.	138
Figura 311 Constructor donde están las variables tipo Dato.	138
Figura 312 Método calcular() del CT 2060.	138
Figura 313 Evento clic del botón del formulario.	139
Figura 314 Formulario del CT 2195.	139
Figura 315 Método calcular() del CT 2195.	139
Figura 316 Formulario del CT 2230.	140
Figura 317 Método calcular() del CT 2230.	140
Figura 318 Formulario antes de la actualización.....	140
Figura 319 Formulario actual.	141
Figura 320 Variables y parámetros del CT 2240.....	141
Figura 321 Variable tipo Dato.	141
Figura 322 Método pasarValoresDeListas().	141
Figura 323 Método calcular() del CT 2240.	141

Figura 324 Evento clic del botón del formulario.....	142
Figura 325 Formulario del CT 2250.	142
Figura 326 Método calcular() del CT 2250.	142
Figura 327 Formulario del CT 2255.	143
Figura 328 Método calcular() del CT 2255.	143
Figura 329 Formulario antes de la actualización.....	143
Figura 330 Formulario actual.	143
Figura 331 Constructor y variables de la clase.....	144
Figura 332 Constructor donde se encuentra la listaDatosRequeridos.	144
Figura 333 Método pasarValoreDeListas.	144
Figura 334 buscar_bastidor().	144
Figura 335 Método calcular() del CT 2290.	145
Figura 336 Evento clic del botón del formulario.....	145
Figura 337 Formulario del CT 2295.	146
Figura 338 Método calcular del CT 2295.....	146

1. Introducción

Posicionada como empresa líder en desarrollo dentro de la industria automotriz y de motores en todo el mundo, MAHLE ofrece a sus clientes sistemas que compiten en las áreas de motores de combustión interna y partes adyacentes al motor.

MAHLE se encuentra entre los 3 proveedores de mayor relevancia a nivel global para sistemas de pistones, componentes de cilindro, sistemas de trenes de válvulas, sistemas de manejo de aire y sistemas de manejo de líquidos. Dentro de nuestros clientes contamos con los fabricantes de motores de combustión interna más reconocidos a nivel mundial.

Su presencia en México incluye alrededor de 2,500 empleados en seis plantas de producción localizadas en Aguascalientes, Lerma, Naucalpan en el estado de México, Puebla y Ramos Arizpe Coahuila (2 plantas).

La planta de Aguascalientes inició operaciones en 1986. Cuenta con una superficie total de 104,396 m², y con 30,000 m² construidos entre áreas productivas, oficinas administrativas y técnicas, en los cuales se desarrollan y fabrican juegos completos de anillos para pistón, incluyendo anillos para control de aceite para diferentes usos dentro de la industria automotriz.

En esta planta también se fabrican anillos para pistón de gran diámetro cuya aplicación es conocida como Super Duty, los cuales son utilizados principalmente para locomotoras a diesel. Actualmente se tiene un proyecto de ampliación y mejora de las instalaciones de la empresa. A la fecha, el 40% de su producción es para exportación y el 60% es para el mercado doméstico.

La capacidad actual de producción de la planta en Aguascalientes es de 145.9 millones de unidades por año. Por su alta tecnología, nivel de servicio así como su estricto control de calidad, principalmente en equipo original, esta planta ha logrado posicionarse como el líder en la fabricación de anillos para pistón en México, obteniendo reconocimientos de sus principales clientes como Ford, Chrysler y General Motors.

1.1 Antecedentes

MAHLE COMPONENTES DE MOTOR DE MEXICO es una empresa del sector auto partes que se dedica a la fabricación y comercialización de anillos para motores de combustión interna. Se integra de 2 unidades, una en Naucalpan,

Estado de México, donde se localiza la fundición; y otra en Aguascalientes, donde se ubica la fábrica de anillos maquinados.

En Aguascalientes, las instalaciones se encuentran en el Km. 0.3 de la carretera Maravillas-Jesús María y es dónde se llevan a cabo todos los procesos de maquinado de los anillos, tanto de compresión como de control de aceite, así como el respectivo juego de segmentos y expansores.

La tecnología usada 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 de 1986.

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 de General Motors.

Esta empresa de auto partes 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 inició 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 División 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 expansores para el mercado de repuesto.

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

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

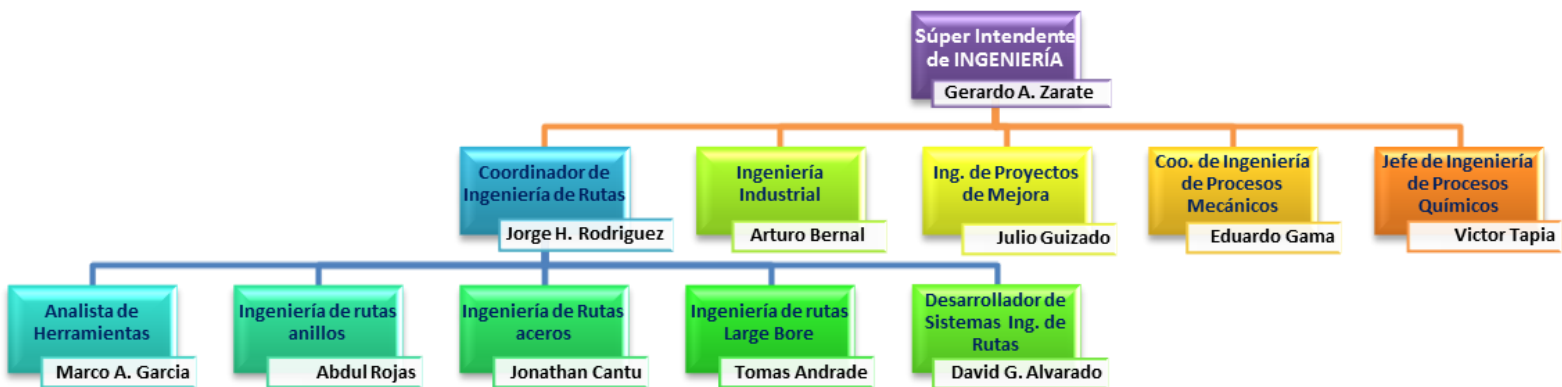
La planta de Naucalpan suministra “castings” (*nombre que recibe la materia prima*) a la planta Aguascalientes para la fabricación de anillos de compresión de hierro. Ambas plantas tienen la misma razón social pero, para fines operativos su relación es de cliente - proveedor.

Durante el año 2008 la empresa es comprada por el grupo industrial Mahle, la cual tiene presencia en todo el mundo.

1.2 Organigrama funcional de Mahle componentes para motor de México



1.3 Organigrama de ingeniería



2. Planteamiento del problema

El departamento de ingeniería de rutas debe entregar a producción un formato con la información necesaria para fabricar los componentes, la información que contiene el formato son las operaciones que necesita el componente, maquinas donde se realizan las operaciones, herramienta requerida en las operaciones, tiempo que el componente debe permanecer en cada operación y datos específicos del componente como materia prima, dimensiones, dureza, resistencia, etc. Para lo cual el departamento de ingenierías de rutas ya cuenta con un sistema que le ayuda a realizar el proceso antes mencionado.

Dicho sistema además cuenta con una herramienta para realizar cotizaciones de los tiempos estándar de operación de cada centro de trabajo, pero el problema con esta herramienta es que las fórmulas con las que se obtiene el valor del tiempo estándar, se encuentran obsoletas, por lo tanto es necesario realizar la actualización con las nuevas fórmulas. También esta herramienta cotiza la cantidad de materia prima a usar de acuerdo a las medidas del anillo que el usuario ingrese al formulario, sin embargo la materia prima de aquellos centros de trabajo ya no está en uso y se debe de actualizar por la nueva.

Además se deben de reparar algunos pequeños errores que no permiten el funcionamiento adecuado del software RGP (Routing Generation Program).

3. Objetivos

3.1 Objetivo general

El principal objetivo es mantener el software RGP (ROUTING GENERATION PROGRAM) actualizado y sin errores.

3.2 Objetivos específicos

- Mantener actualizadas las fórmulas para calcular el tiempo estándar de cada uno de los centros de trabajo que están dados de alta en la base de datos.
- Dar de alta en el sistema nuevos centros de trabajo que ya están en operación en planta.
- Actualizar la materia prima de los centros de trabajo 715, 9361, 9362.
- Reparar los errores que han identificado los usuarios del departamento de ingeniería de rutas.

4. Justificación

El software RGP (Routing Generation Program) debe de estar siempre actualizado y funcionando correctamente debido a que los datos que se obtienen de este, se dan de alta en el ERP de la empresa que en este caso es el SAP.

En especial tiene que estar actualizado y funcionando correctamente la herramienta de cotización, ya que cuando se realiza la cotización de varios centros de trabajo se exportan los tiempos estándar a una hoja de Excel y todos los datos contenidos en dicha hoja se almacenan en el ERP de la empresa.

5. Marco teórico

5.1 ¿Qué es un ERP?

Definición. Los sistemas de planificación de recursos de la empresa (en inglés ERP, Enterprise Resource Planning) 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.

Los sistemas ERP son sistemas integrales de gestión para la empresa. Se caracterizan por estar compuestos por diferentes partes integradas en una única aplicación. Estas partes son de diferente uso, por ejemplo: producción, ventas, compras, logística, contabilidad (de varios tipos), gestión de proyectos, GIS (sistema de información geográfica), inventarios y control de almacenes, pedidos, nóminas, etc. Solo podemos definir un ERP como la integración de todas estas partes.

Lo contrario sería como considerar un simple programa de facturación como un ERP por el simple hecho de que una empresa integre únicamente esa parte. Ésta es la diferencia fundamental entre un ERP y otra aplicación de gestión.

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 que tan sólo se integra uno o una pequeña parte de los procesos de negocio. La propia definición de ERP indica la necesidad de "Disponibilidad de toda la información para todo el mundo todo el tiempo".

Los objetivos principales de los sistemas ERP son:

- Optimización de los procesos empresariales

- Acceso a toda la información de forma confiable, precisa y oportuna (integridad de datos)
- La posibilidad de compartir información entre todos los componentes de la organización
- Eliminación de datos y operaciones innecesarias de reingeniería

5.1.1 El propósito fundamental de un ERP

Es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de operación. Las características que distinguen a un ERP de cualquier otro software empresarial, es que deben de ser sistemas integrales, con modularidad y adaptables Integrales, porque permiten controlar los diferentes procesos de la compañía entendiendo 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. Por ejemplo, en una compañía, el que un cliente haga un pedido representa que se cree una orden de venta que desencadenará el proceso de creación de ese componente.

Si la empresa no usa un ERP, necesitará tener varios programas que controlen todos los procesos mencionados, con la desventaja de que al no estar integrados, la información se duplica, crece el margen de contaminación en la información (sobre todo por errores de captura) y se crea un escenario favorable para malversaciones. Con un ERP, el operador simplemente captura el pedido y el sistema se encarga de todo lo demás, por lo que la información no se manipula y se encuentra protegida.

5.1.2 Ventajas de los ERP

Tanto económica como técnicamente es que la funcionalidad se encuentra dividida en módulos, los cuales pueden instalarse de acuerdo con los requerimientos del cliente.

Ejemplo: ventas, materiales, finanzas, control de almacén, recursos humanos, etc. Adaptables.

Los ERP están creados para 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 se necesiten de cada uno. Por ejemplo, para controlar inventarios, es posible que una empresa necesite manejar la partición de lotes pero otra empresa no.

Los ERP más avanzados suelen incorporar herramientas de programación de 4ª Generación para el desarrollo rápido de nuevos procesos. La parametrización es el valor añadido fundamental que se debe hacer con cualquier ERP para adaptarlo a las necesidades concretas de cada empresa.

Otra de las características destacables de los sistemas ERP son las bases de datos centralizadas.

5.1.3 Componentes del ERP

Interactúan entre sí consolidando todas las operaciones. En un sistema ERP los datos se ingresan sólo una vez y deben ser consistentes, completos y comunes. Las empresas que lo implanten suelen tener que modificar alguno de sus procesos para alinearlos con los del sistema ERP.

Este proceso se conoce como Reingeniería de Procesos, aunque no siempre es necesario. Aunque el ERP pueda tener menús modulares configurables según los roles de cada usuario, es un todo. Esto es un único programa (con multiplicidad de librerías) con acceso a una base de datos centralizada.

No debemos confundir en este punto la definición de un ERP con la de una suite de gestión. La tendencia actual es ofrecer aplicaciones especializadas para determinadas empresas.

Es lo que se denomina versiones sectoriales o aplicaciones sectoriales especialmente indicadas o preparadas para determinados procesos de negocio de un sector (los más utilizados).

5.2 SAP (System Applications and Products)

HISTORIA. 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 del 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 R/2, antecesor al SAP R/3. Estas son algunas cifras del éxito en el mundo: 5º mayor

proveedor de software del 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 del 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 las versión anterior, SAP R/2, son: Arquitectura cliente/servidor Entorno gráfico 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 un 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 de 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 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 necesiten 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.

5.2.1 ¿Qué 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 logística, el módulo LO brinda las 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 de BI para la creación de reportes, dashboards y gráficos, Ventas y Distribución (SD): Facturación, Pedidos de ventas, solicitudes, pedidos abiertos; etcétera.

6. Metodología

6.1 Actualización de la materia prima de los CT 715, 9361 y 9362

6.1.1 Centro de trabajo 715

Para agregar nuevas materias a este centro de trabajo primero se tiene que localizar la clase CentroTrabajo715.vb que se encuentra en la carpeta centros de trabajo del ProyectoSa como se muestra a continuación:

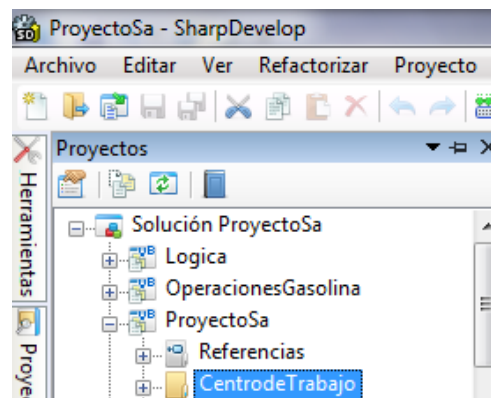


Figura 1 Directorio de las clases de los centros de trabajo.

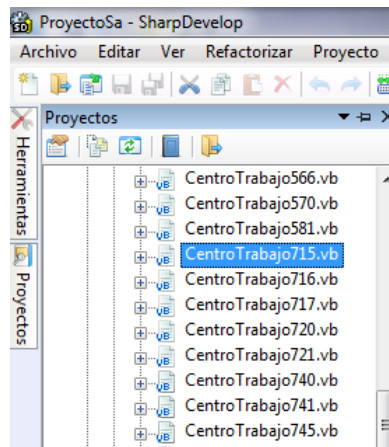


Figura 2 Clase CentroTrabajo715.vb.

Una vez localizada la clase se tiene que dirigir al método llamado calcular dentro de la clase dependiendo de la cantidad de materia prima que se requiera agregar es la cantidad de variables que se van a utilizar como se muestra en la siguiente imagen:

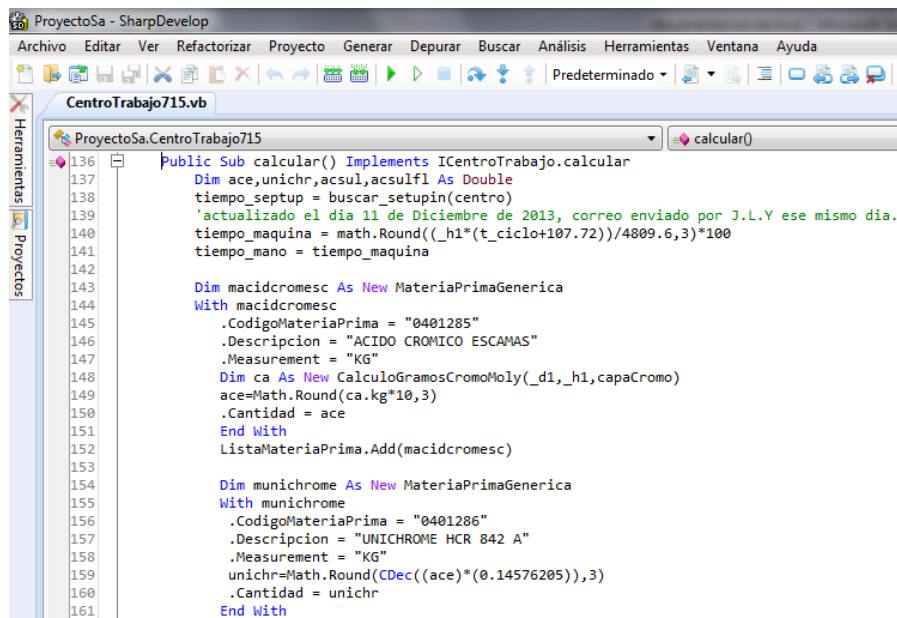


Figura 3 Método calcular del CT 715 donde se agrega la materia prima.

Por ejemplo la variable llamada macidcromesc es una variable tipo objeto que corresponde a la materia prima ácido crómico escamas y en esta variable se guardan las propiedades de dicha materia prima las cuales son CódigoMateriaPrima, Descripción, measurement y la cantidad esta última se calcula mediante una fórmula matemática que se encuentra en otra clase llamada CalculoGramosCromoMoly.vb específicamente en el método llamado calcular_cromo_kg en donde la variable kg contiene la fórmula que calcula la cantidad de materia que se va a utilizar.

```

96 Private Sub calcular_cromo_kg()
97     Dim d21,d22 As Double
98     d21 = CDec(Me.txt_diametro.Text) * Math.PI
99     d22= d21*CDec(txt_width.Text)*CDec(txt_espesor_cromo.Text)
100    kg = CDec(d22*Math.Pow(2.54,3))*1.923076923*7.14/1000)*1000
101    alum= CDec(d22*Math.Pow(2.54,3))*7.14*4.5/100/1000+0.023076923)*1000
102 End Sub

```

Figura 4 Método calcular_cromo_kg.

Para vincular esta fórmula a la clase CentroTrabajo715.vb, primero se tiene que declarar una nueva variable tipo objeto a partir de la clase CalculoGramosCromoMoly.vb como la siguiente:

```

148 Dim ca As New CalculoGramosCromoMoly( d1, h1,capaCromo)

```

Figura 5 Variable ca de tipo objeto.

La cual contiene los parámetros necesarios para que funcione. Una vez que se realizó todo lo anterior se declara una nueva variable en la cual vamos a guardar el valor de la fórmula que se encuentra en la otra clase. Una vez que la variable obtenga el valor de la formula se procederá a agregarla a la propiedad cantidad. Además la variable que guarda las propiedades de la materia prima se agrega a la listaMateriaPrima del proyecto para que la materia prima esté disponible para otras clases.

6.1.2 Centro de Trabajo 9361

Para este centro de trabajo es similar el procedimiento al del centro de anterior, primero localice la clase, una vez localizada se debe de ubicar el método calcular ya que se ubicó se procede a agregar la materia prima que se necesita para este centro de trabajo como se muestra en las siguientes líneas de código.

```

142 Dim macidcromesc As New MateriaPrimaGenerica
143 With macidcromesc
144     .CodigoMateriaPrima = "0401285"
145     .Descripcion = "ACIDO CROMICO ESCAMAS"
146     .Measurement = "KG"
147 Dim ca As New CalculoGramosCromoMoly(_d1,_h1,capaCromo)
148 ace=Math.Round(ca.kg*10,3)
149 .Cantidad = ace
150 End With
151 ListaMateriaPrima.Add(macidcromesc)
152

```

Figura 6 Variable tipo MateriaPrimaGenerica.

```

CentroTrabajo9361.vb
ProyectoSa.CentroTrabajo9361
calculador()
127     Return centro
128 End Get
129 Set
130     centro = value
131 End Set
132 End Property
133
134 Public tiempo_maquina As Double
135 Public tiempo_septup As Double
136 Public tiempo_mano As Double = 0
137
138 Public Sub calcular() Implements ICentroTrabajo.calcular
139     Dim accres,acsulk,acsulfl As Double
140     tiempo_septup = buscar_setupin(centro)
141     tiempo_maquina = math.Round(((3300.247 + _tiempo_cromado) / (36 * (8 * (14.5 / _width)))) * 100,3)
142
143     tiempo_mano = math.Round(tiempo_maquina * factorLabor,3)
144
145     Dim macidchresc As New MateriaPrimaGenerica
146     With macidchresc
147         .CodigoMateriaPrima = "0401285"
148         .Descripcion = "ACIDO CROMICO ESCAMAS"
149         .Measurement = "KG"
150     End With
151     Dim ca As New CalculoGramosCromoMoly(_d1,_width,capaCromo)
152     accres=Math.Round(ca.kg*10,3)
153     .Cantidad = accres
154     End With
155     ListaMateriaPrima.Add(macidchresc)
156
157     Dim macidosulfurico As New MateriaPrimaGenerica
158     With macidosulfurico
159         .CodigoMateriaPrima="0401226A"
160         .Descripcion="ACIDO SULFURICO"
161         .Measurement="L"
162         acsulk=(accres*0.01003547)
163         acsulfl=Math.Round(acsulk/1.04,3)
164         .Cantidad=acsulfl
165     End With
166     ListaMateriaPrima.Add(macidosulfurico)
167 End Sub

```

Figura 7 Método calcular del CT 9361 donde se agrega la materia prima.

En donde cada variable tipo objeto contiene toda la información correspondiente a la materia prima y además contiene la formula con la cual se realiza el cálculo de la materia que se va a usar en el centro de trabajo. Pero para que se lleve a cabo el cálculo hay que insertar primero los valores que el formulario de la clase requiere para realizar el cálculo.

Figura 8 Formulario del CT 9361.

Una vez que el usuario inserta los valores, el software guarda los valores en unas variables, las cuales ya habían sido declaradas.

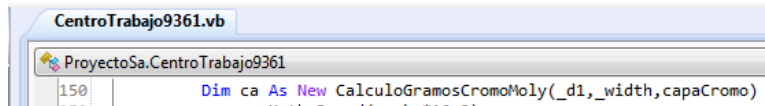
```

CentroTrabajo9361.vb
ProyectoSa.CentroTrabajo9361
118     '-----variables que se necesitan-----
119     Dim _width As Double
120     Dim _tiempo_cromado As Double
121     Dim _d1 As Double
122     Dim capaCromo As Double
123     '-----

```

Figura 9 Variables de la clase CentroTrabajo9361.

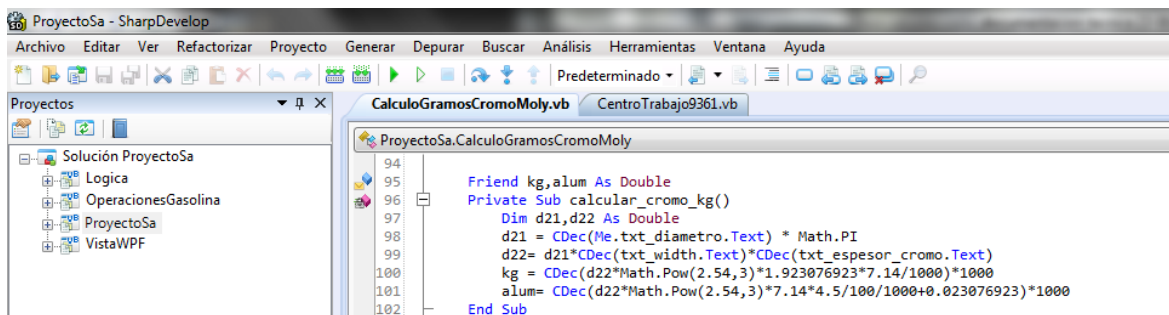
Después estas variables se insertan como parámetros al momento de que se declara una variable tipo objeto la cual requiere de tres parámetros como se muestra a continuación:



```
150 Dim ca As New CalculoGramosCromoMoly(_d1, _width, capaCromo)
```

Figura 10 Variable ca de tipo objeto.

Como en el centro de trabajo anterior la fórmula para calcular la primera materia prima la cual es acido crómico escamas se encuentra en otra clase llamada CalculoGramosCromoMoly.vb, específicamente en el método calculo como lo muestra la siguiente imagen:



```
94
95 Friend kg,alum As Double
96 Private Sub calcular_cromo_kg()
97 Dim d21,d22 As Double
98 d21 = CDec(Me.txt_diametro.Text) * Math.PI
99 d22= d21*CDec(txt_width.Text)*CDec(txt_espesor_cromo.Text)
100 kg = CDec(d22*Math.Pow(2.54,3)*1.923076923*7.14/1000)*1000
101 alum= CDec(d22*Math.Pow(2.54,3)*7.14*4.5/100/1000+0.023076923)*1000
102 End Sub
```

Figura 11 Método calcular_cromo_kg de la clase CalculoGramosCromoMoly.vb.

La fórmula esta almacenada en la variable kg, es muy importante verificar que el resultado de esta fórmula sea correcto ya que la mayoría de las materias primas dependen del valor de esta fórmula para realizar el cálculo de su fórmula.

Una vez que se realizó todo lo anterior solo resta agregar la variable de la materia prima a la ListaMateriaPrima para poder realizar cotizaciones posteriormente.

6.1.3 Centro de trabajo 9362

Para agregar nueva materia prima a este centro de trabajo se debe realizar el mismo procedimiento como en los anteriores centros de trabajo. La única diferencia de este centro de trabajo con los dos anteriores es que las fórmulas para calcular la cantidad de materia prima a usar se encuentran en un archivo de Excel que se encuentra dentro de la carpeta del sistema.

Después de que se localizó el archivo se debe modificar las siguientes líneas de código que se encuentran en el método calcular de la clase CentroTrabajo9362.vb y son las siguientes:

```

CentroTrabajo9362.vb
ProyectoSa.CentroTrabajo9362
ListaMateriaPrima

147
148 Public Sub calcular() Implements ICentroTrabajo.calcular
149     Try
150         tiempo_septup = buscar_setupin(centro)
151         tiempo_maquina = MATH.Round((( 4287 + _TiempoCiclo ) / ( 36 * ( 2 * ( ( 13.5 / _h1 ) * 2 ) ) ) ) * 100,3)
152         tiempo_mano = tiempo_maquina
153
154         'AGREGAMOS LA MATERIA PRIMA
155         Dim oExcel As Excel.ApplicationClass
156         Dim oBook As Excel.WorkbookClass
157         Dim oBooks As Excel.Workbooks
158         Dim oHoja As Excel.Worksheet
159
160         Dim aldCI As System.Globalization.CultureInfo = System.Threading.Thread.CurrentThread.CurrentCulture
161         System.Threading.Thread.CurrentThread.CurrentCulture = New System.Globalization.CultureInfo("en-US")
162
163         oExcel = CreateObject("Excel.Application")
164         oBooks = oExcel.Workbooks
165         oBook = oBooks.Open("\\Agufileserv2\INGENIERIA\RESPRUTAS\NUEVO SOFTWARE RUTAS\raul\nueva\RoutingGenerationProgram\ArchivosLargeBore\GramosCromoCeramicoCT2.xls")
166         oHoja = oBook.Sheets("DatosEntrada")
    
```

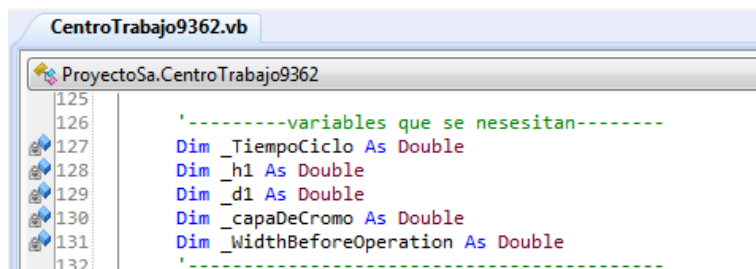
Figura 12 Nuevas líneas de código.

La función que realizan estas líneas de código es la de abrir el archivo y la hoja donde se encuentran las formulas, las cuales están dentro de la siguiente tabla:

CALCULO DE GRAMOS DE CROMO CERAMICO DEPOSITADO			
DATOS DE ENTRADA			
VARIABLE	DESCRIPCION DE LAS VARIABLES	ORIGEN	ENTER DATA
DiaNom=	Diametro nominal:	4.2	Pulg
EspCr=	Espesor de cromo	0.003	Pulg
WidthNom=	Width nominal	0.094	Pulg
LargoM=	largo del mandril	15.7	Pulg
Densidad=	Densidad de corriente	68	Amp/dm ²
Tiempo	Tiempo	#N/A	hrs
RESULTADOS			
AniM=	anillos por mandril	146	Anillos por mandril
I=	Amperaje de depositación	909	Amp
M =	(cromo metálico depositado en el anillo)	#N/A	Kgs de Cr metálico
CrM _(mandril)	Cr Metálico (mandril)	#N/A	Kgs
CrM _(anillo)	Cr Metálico (anillo)	#N/A	Kgs
401285	ACIDO CROMICO ESCAMAS	0.001	KG
401287	HEEF 236 MS	0.000	KG
401286	UNICHROME HCR 842 A	0.000	KG
0401226A	ACIDO SULFURICO	0.000	KG
0401226A	ACIDO SULFURICO	0.000	L
04012412A	ALUMINA	0.023	Lbs

Tabla 1. Tabla para calcular la cantidad de materia prima.

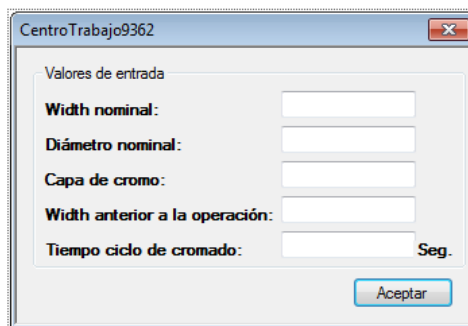
Al igual que en el anterior centro de trabajo se deben de declarar las variables necesarias que guarden los valores de entrada que necesita el centro de trabajo para llevar a cabo los cálculos y son las siguientes:



```
125  
126  
127 '-----variables que se necesitan-----  
128 Dim _TiempoCiclo As Double  
129 Dim _h1 As Double  
130 Dim _d1 As Double  
131 Dim _capaDeCromo As Double  
132 Dim _widthBeforeOperation As Double  
133 '-----
```

Figura 13 Variables del CT 9362.

Las cuales almacenan los valores del diámetro, width, tiempo ciclo y la capa de cromo que el usuario inserta en el formulario del centro de trabajo.



CentroTrabajo9362

Valores de entrada

Width nominal:

Diámetro nominal:

Capa de cromo:

Width anterior a la operación:

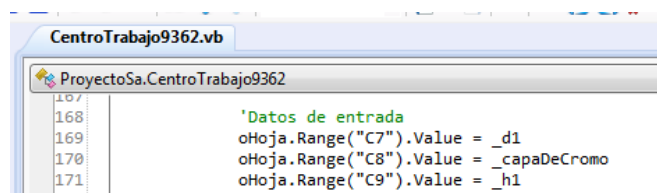
Tiempo ciclo de cromado: Seg.

Aceptar

Figura 14 Formulario del CT 9362.

Nota: se recomienda no eliminar la variable `_widthBeforeoperation` ya que es utilizada por otras clases, lo que sí se puede eliminar es su respectiva etiqueta y el campo de texto correspondiente o también lo que se puede hacer es ocultar estos elementos del formulario.

Ya que se declararon las variables deben de con su respectiva celda en el archivo de Excel como se muestra en las siguientes líneas de código.



```
167  
168 'Datos de entrada  
169 oHoja.Range("C7").Value = _d1  
170 oHoja.Range("C8").Value = _capaDeCromo  
171 oHoja.Range("C9").Value = _h1
```

Figura 15 Código para igualar una variable con una celda de Excel.

Lo que realiza esta parte de código es que manda los valores que el usuario ingreso al formulario a su celda correspondiente, por ejemplo a la celda C7 de la hoja DatosEntrada le corresponde el valor del diámetro nominal, entonces el software guarda el valor en la variable `_d1` y lo manda a dicha celda. Todo esto sucede con los demás valores también.

Posteriormente ahora si ya se puede realizar todo el procedimiento que se vio anteriormente para agregar la materia prima que le corresponde al centro de trabajo

```

CentroTrabajo9362.vb
ProyectoSa.CentroTrabajo9362
calculador()
173 'datos de salida
174 Dim mpAcidoCromicoEscamas As New MateriaPrimaGenerica
175 With mpAcidoCromicoEscamas
176 .Cantidad = math.Round(CDec(oHoja.Range("C23").Value)*10000,3)
177 .CodigoMateriaPrima = "0401285"
178 .Descripcion = "ACIDO CROMICO ESCAMAS"
179 .Measurement = "KG"
180 End With
181 ListaMateriaPrima.Add(mpAcidoCromicoEscamas)
182
183 Dim mpHeef As New MateriaPrimaGenerica
184 With mpHeef
185 .Cantidad = math.Round(CDec(oHoja.Range("C24").Value)*10000,3)
186 .CodigoMateriaPrima = "0401287"
187 .Descripcion = "HEEF 236 MS"
188 .Measurement = "KG"
189 End With
190 ListaMateriaPrima.Add(mpHeef)
191
192 Dim mpUnichrome As New MateriaPrimaGenerica
193 With mpUnichrome
194 'Se cambio a KG y el codigo tambien cambi6 (de 0401241 a 04012412A), esto por un correo de J.H.R el dia 27 de Nov. 2014
195 .Cantidad = math.Round(CDec(oHoja.Range("C25").Value)*10000,3)
196 .CodigoMateriaPrima = "0401286"
197 .Descripcion = "UNICHROME HCR 842 A"
198 .Measurement = "KG"
199 End With
200 ListaMateriaPrima.Add(mpUnichrome)
201
202 Dim mpAcidoSulfurico As New MateriaPrimaGenerica
203 With mpAcidoSulfurico
204 .Cantidad=math.Round(CDec(oHoja.Range("C27").Value)*10000,3)
205 .CodigoMateriaPrima="0401226A"
206 .Descripcion="ACIDO SULFURICO"
207 .Measurement="L"
208 End With
209 ListaMateriaPrima.Add(mpAcidoSulfurico)
110

```

Figura 16 Variables tipo MateriaPrimaGenerica del CT 9362.

Todo el procedimiento es similar a lo que se realizó en los dos centros de trabajo anteriores, lo único que cambia es que el valor de la cantidad se extrae de las celdas de la tabla resultados las cuales contienen la fórmula para calcular la cantidad de materia prima a usar. También contiene el código de la materia prima, así como también la descripción y el measurement de la misma.

	A	B	C	D
14				
15	RESULTADOS			
16				
17				
18	AniM=	anillos por mandril	146	Anillos por mandril
19	I=	Amperaje de deposición	909	Amp
20	M =	(cromo metálico depositado en el anillo)	#N/A	Kgs de Cr metálico
21	CrM _(mandril)	Cr Metálico (mandril)	#N/A	Kgs
22	CrM _(anillo)	Cr Metálico (anillo)	#N/A	Kgs
23	401285	ACIDO CROMICO ESCAMAS	0.001	KG
24	401287	HEEF 236 MS	0.000	KG
25	401286	UNICHROME HCR 842 A	0.000	KG
26	0401226A	ACIDO SULFURICO	0.000	KG
27	0401226A	ACIDO SULFURICO	0.000	L
28	04012412A	ALUMINA	0.023	Lbs

Tabla 2. Tabla de resultados de la cantidad de materia prima a usar.

Por ultimo solo se debe agregar la variable de la materia prima a la ListaMateriaPrima para que se pueda visualizar su contenido al momento de crear

cotizaciones. Si se llegaran a actualizar las formulas, se tendrán que cambiar las formulas directamente en el archivo de Excel para que surtan efecto a la hora de hacer la cotización.

6.2 Actualización de tiempos estándar – Anillos

6.2.1 CentroTrabajo110.vb – FIRST ROUGH GRIND

A este centro de trabajo se le modifico la fórmula que calcula el tiempo estándar de operación, para ello primero se tiene que revisar el archivo de Excel que contiene los centros de trabajo a los cuales se les tiene que modificar la fórmula para calcular el tiempo estándar.

Primero se recomienda hacer el cálculo de tiempos en el software RGP para verificar si la formula cambio, si el resultado es igual al que está en el Excel no será necesario hacerle ninguna modificación al centro de trabajo, sin embargo si el resultado no es el mismo al del Excel abra que hacer una serie de modificaciones al centro de trabajo.

Lo primero será revisar si los valores de entrada son los correctos para llevar a cabo el cálculo si en el archivo de Excel se requieren más valores de entrada se deberán agregar al formulario. De lo contrario no se tendrá que modificar nada al formulario. En este caso no será necesario agregar nuevos valores de entrada ya que para este centro de trabajo lo único que cambio fue su fórmula.

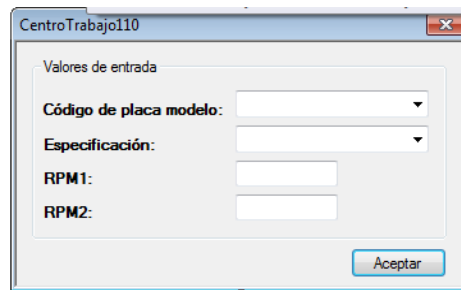


Figura 17 Formulario del CT 110.

Lo siguiente a realizar es cambiar la formula, para realizar este cambio debemos dirigirnos al método calcular de la clase CentroTrabajo110.vb ya estando dentro del método se realiza un if antes de hacer el cálculo de la formula, este if lo que hace es que dependiendo del material tienen valores de tiempo ciclo diferentes y estos se obtiene a través de una operación matemática. Y es la siguiente:


```

213 Public Sub calcular() Implements ICentroTrabajo.calcular
214 Dim dia_cas As Double
215 Dim width_cas As Double
216 buscar_septup()
217 dia_cas = saca_diametrocastingMayor(Es_Casting.Substring(0,8))
218 width_cas = saca_widthcasting(Es_Casting.Substring(0,8))
219 Dim t_ciclo2,t_ciclo1, ciclo_carga As Double
220 If banElementoFijo = False Then
221     Tipo = Module1.sacar_tipo_material(_material)
222 End If
223 If Tipo = "HIERRO GRIS" Or Tipo = "HIERRO GRIS CENTRIFUGADO" Then
224     t_ciclo1 = Math.Round((0.36*9)/_rmp1,3)
225     t_ciclo2 = Math.Round((0.3*12)/_rmp2,3)
226 ElseIf Tipo = "HIERRO GRIS INTERMEDIO"
227     t_ciclo1 = Math.Round((0.49*7)/_rmp1,3)
228     t_ciclo2 = Math.Round((0.36*11)/_rmp2,3)
229 ElseIf Tipo = "HIERRO GRIS ALTO MODULO"
230     t_ciclo1 = 0.65
231     t_ciclo2 = 0.65
232 Else
233     msgbox("Material " & Tipo & " no disponible para el C.C 110, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
234 End If

```

Figura 18 if que obtiene los tiempos ciclo.

Como se muestra en la figura 19 los materiales hierro gris y hierro gris intermedio sus tiempos ciclos se obtienen de una operación matemática, excepto el material hierro gris alto modulo ya que para este material los valores ya estaban definidos. Las operaciones se obtienen como se muestra en la siguiente figura.

MAHLE COMPONENTES PARA MOTOR DE MEXICO				
Fórmula para Cálculo de Tiempos Estándar				
First rough grind				
RPM 1		11	in	
RPM 2		19	in	
Carga		1	anillos	
Tipo de Material		HIERRO GRIS ALTO MOD		
Tiempo de Ciclo 1		0.6500	segundos	
Tiempo de Ciclo 2		0.6500	segundos	SAP
Actividades ciclicas		0.0748	segundos	3.8930
Actividades no ciclicas		0.0267	segundos	
Ciclo por carga		1.4015	segundos	
Horas / 100 anillos		0.0389	horas / 100 anillos	
Piezas / hora		2568	anillos / hora	
SIMPLIFICADA:		(((Tiempo ciclo 1 de tipo de material * RPM De Muestra) / RPM 1 + (Tiempo ciclo 2 de tipo de material * RPM de Muestra) / RPM 2) + 0.1015) / 36		
TIPO DE MATERIAL	TC 1	RPM DE MUESTRA	TC 2	RPM DE MUESTRA
ACERO	N/D	N/D	N/D	N/D
HIERRO DUCTIL	N/D	N/D	N/D	N/D
HIERRO GRIS	0.2945	9.0	0.1895	12.0
HIERRO GRIS ALTO MOD	0.6500	7.2	0.6500	7.0
HIERRO GRIS INTERMEDIO	0.3118	7.0	0.2084	11.0

Figura 19 Obteniendo la operación para sacar los tiempos ciclo.

Como se muestra en la figura anterior al seleccionar la celda TC 1 en la barra de funciones se muestra la operación que se realiza para obtener el valor de la misma, igual pasa con los demás valores de los tiempos ciclo de los diferentes materiales.

Ahora que tenemos los tiempos ciclos, procedemos a cambiar la fórmula para obtener el tiempo estándar. Para ello primero debemos declarar una variable en la cual almacenaremos el valor del ciclo por carga, que posteriormente se usara para calcular la formula. El valor de la variable se obtiene mediante una sumatoria donde involucra los tiempos ciclos uno y dos, además de otros dos valores ya establecidos.

```
236 | ciclo_carga = (t_ciclo1 + t_ciclo2 + 0.0748 + 0.0267)
```

Figura 20 Variable ciclo por carga.

Ya que se estableció la variable ciclo carga, se muestra la formula actualizada en la figura 21.

```
237 | tiempo_maquina = Math.Round((100 * ((ciclo_carga)/3600)/1)*100,3)
```

Figura 21 Formula actualizada para calcular el tiempo estándar del CT 110.

Y con esto la fórmula para calcular el tiempo estándar ya estaría actualizado.

6.2.2 CentroTrabajo112.vb – SECOND ROUGH GRIND

Al igual que en el CT 110 no habrá necesidad de agregar nuevos valores de entrada ya que con los que cuenta son más que necesarios para realizar el cálculo del tiempo estándar. El formulario de este centro de trabajo es el siguiente:

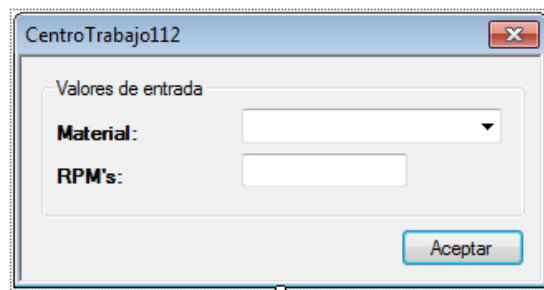


Figura 22 Formulario del CT 112.

Ya que se revisó que no es necesario agregar nuevos valores de entrada, lo que sigue es establecer el tiempo ciclo de cada material mediante una condición if en el método calcular de la clase CentroTrabajo112.vb y se realiza de la siguiente manera:

```

CentroTrabajo112.vb
ProyectoSa.CentroTrabajo112
159 Public Sub calcular() implements ICentroTrabajo.calcular
160     buscar_septup()
161     If banElementoFijo = False Then
162         tipo = Module1.sacar_tipo_material(_material)
163     End If
164     Dim t_ciclo, c_carga As Double
165     If tipo = "HIERRO DUCTIL" Then
166         t_ciclo = Math.Round((0.46*9.5)/_rpm,3)
167     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
168         t_ciclo = Math.Round((0.45*8)/_rpm,3)
169     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
170         t_ciclo = Math.Round((0.57*6)/_rpm,3)
171     ElseIf tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO"
172         t_ciclo = Math.Round((0.41*9)/_rpm,3)
173     Else
174         msgbox("Material " & tipo & " no disponible para el C.C 112, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
175     Exit Sub
176 End If

```

Figura 23 Condicional para obtener el tiempo ciclo.

El valor del tiempo ciclo de cada material se obtiene con una misma operación matemática lo único que cambia son los valores dependiendo del material. Esos valores se obtienen de la tabla de materiales como se muestra a continuación:

	A	B	C	D	E	F	G	H	I
24		TIPO DE MATERIAL		TIEMPO CICLO		RPM DE MUESTRA			
25		ACERO		N/D		N/D			
26		HIERRO DUCTIL		0.55		9.5			
27		HIERRO GRIS		0.46		9			
28		HIERRO GRIS ALTO MOD		0.43		6			
29		HIERRO GRIS INTERMEDIO		0.45		8			

Tabla 3. Tabla de materiales

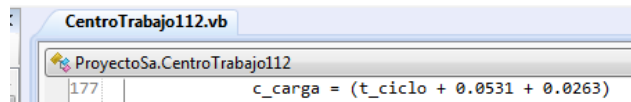
Al seleccionar el tiempo ciclo de cualquier material aparecerá la operación en la barra de funciones de cómo se obtiene el valor de la celda, entonces esa operación que aparece es la que se debe poner en la condicional de su respectivo material.

Después de que se realice el paso anterior se debe declarar una variable del tipo doublé en la cual se almacenara el valor del ciclo por carga. Este valor se obtiene de la suma de los valores que se muestran en la figura 24.

	A	B	C	D	E	F	G	H	I
1									
2		MAHLE COMPONENTES PARA MOTOR DE MEXICO							
3		<i>Fórmula para Cálculo de Tiempos Estándar</i>							
4		SECOND ROUGH GRIND							
5									
6									
7		RPM				8	rpm		
8		Width nominal				0.04685	in		
9		Cantidad de anillos				1	anillos		
10		Tipo de Material				HIERRO GRIS INTERMEDIO			
11									
12		→	Tiempo de Ciclo			0.45	segundos		
13		→	Actividades ciclicas			0.0531	segundos		
14		→	Actividades no ciclicas			0.0263	segundos		SAP
15			Ciclo por carga			==SUM(F12:F14)	segundos		1.4706

Figura 24 Valores para obtener el ciclo por carga.

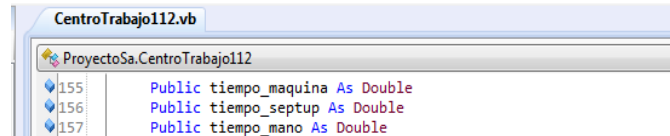
Obviamente el tiempo ciclo va a cambiar dependiendo del material. Entonces la variable quedaría de la siguiente manera:



```
CentroTrabajo112.vb
ProyectoSa.CentroTrabajo112
177 | c_carga = (t_ciclo + 0.0531 + 0.0263)
```

Figura 25 Variable ciclo por carga.

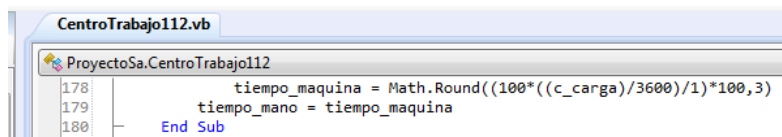
La última actividad a realizar es declarar una variable del tipo double en donde se almacenara la fórmula para calcular el tiempo estándar del centro de trabajo.



```
CentroTrabajo112.vb
ProyectoSa.CentroTrabajo112
155 | Public tiempo_maquina As Double
156 | Public tiempo_septup As Double
157 | Public tiempo_mano As Double
```

Figura 26 Declaración de variables.

Ya declarada la variable y realizado todos los procedimientos anteriores, solo resta asignarle a la variable tiempo_maquina la formula actualizada para calcular el tiempo estándar correspondiente al centro de trabajo.



```
CentroTrabajo112.vb
ProyectoSa.CentroTrabajo112
178 | tiempo_maquina = Math.Round((100*((c_carga)/3600)/1)*100,3)
179 | tiempo_mano = tiempo_maquina
180 | End Sub
```

Figura 27 Formula actualizada.

6.2.3 CentroTrabajo132.vb – SPLITTER CUFFS

En este centro de trabajo si se le tuvieron que agregar más valores de entrada ya que antes de la actualización solo necesitaba un valor de entrada, sin embargo con la actualización de la formula se requieren más valores de entrada para poder llevar a cabo el cálculo del tiempo estándar. Este es el nuevo formulario.

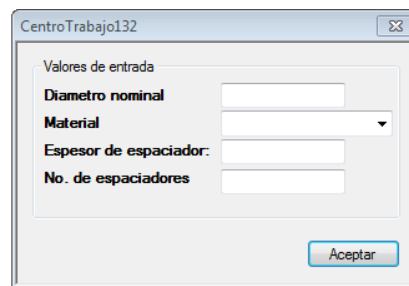


Figura 28 Formulario correspondiente al CT132.

Como se puede ver en la figura 3.1 se le agregaron nuevos valores de entrada al formulario del centro de trabajo ya que anteriormente solo requería el espesor de espaciador, pero ahora requiere tres valores más. Para agregarlos solo hay que

abrir la clase en modo de diseño y agregarle sus respectivos campos de texto o combo box, además de etiqueta a cada valor de entrada.

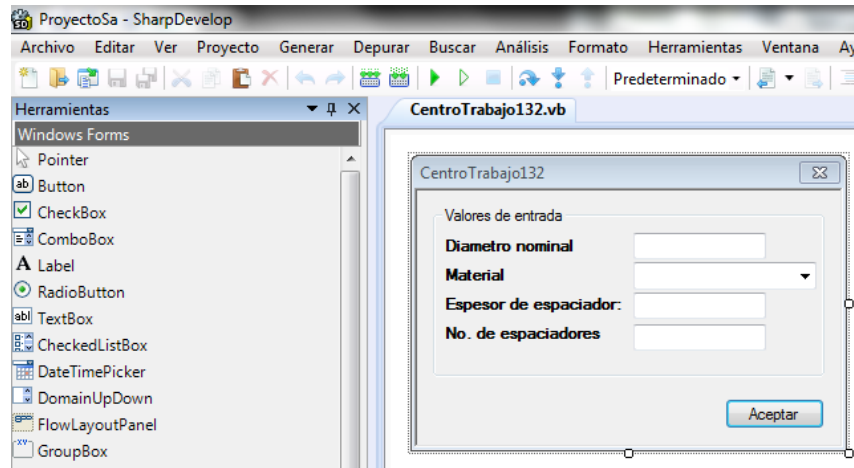


Figura 29 Creando el formulario de la clase.

Una vez que se ha creado el diseño del formulario, se deben de declarar varias variables en las cuales se almacenaran los valores introducidos por el usuario, además serán las que se usan en toda la clase, a estas se les llaman variables necesarias y son las siguientes:

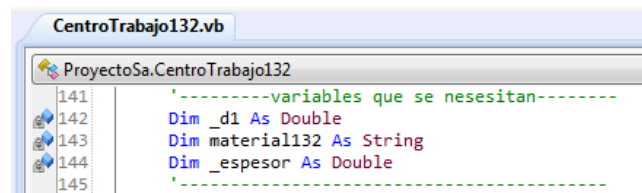


Figura 30 Variables donde se almacenaran los diferentes valores.

Ahora hay que igualar las variables con los parámetros de la clase, esto se hace como se muestra a continuación:

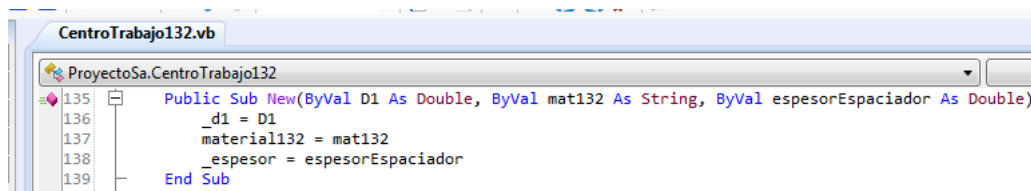


Figura 31 Igualando variables con parámetros.

Después de igualar se deben seguir los siguientes pasos que son obligatorios ya que si no se hacen el software no se ejecutara. El primer paso consta de declarar un número de variables de tipo objeto igual a los valores de entrada como se muestra en la siguiente imagen:

```

CentroTrabajo132.vb
ProyectoSa.CentroTrabajo132
120 Public Sub New()
121     Me.InitializeComponent()
122     Dim diametroNominal132 As New Dato("diametroNominal132","Distance","diametro nominal de la operación SPLITTER CUFFS",Nothing,"diametro nominal de splitter")
123     listaDatosRequeridos.Add(diametroNominal132)
124
125     Dim material_ As New DatoString("MATERIAL","Especificación de materia prima (MF012-S,SPR-128,MF025-L,etc...)", "Material:")
126     listaDatosStringRequeridos.Add(material_)
127
128     Dim espesorEspaciador132 As New Dato("espesorEspaciador132","Distance","Espesor de espaciador de operación SPLITTER CUFFS",Nothing,"Width espaciador de splitter")
129     listaDatosRequeridos.Add(espesorEspaciador132)
130
131     Dim numeroEspaciadores As New Dato("numeroEspaciadores","Cantidad","Numero de espaciadores de operación SPLITTER CUFFS",Nothing,"numero de espaciadores de splitter")
132     listaDatosRequeridos.Add(numeroEspaciadores)
133 End Sub

```

Figura 32 Variables que se agregan a la lista de datos requeridos.

El segundo paso es agregar las variables necesarias al método pasarValoresDeListas e igualarlas con los métodos obtenerValorDatoDouble u obtenerValorDatoString dependiendo del tipo de dato que se le asignó a la variable, estos métodos pertenecen al module1 del proyecto. Este paso se ve de la siguiente manera:

```

ProyectoSa - SharpDevelop
chivo Editar Ver Refactorizar Proyecto Generar Depurar Buscar Análisis Herramientas Ventana Ayuda
CentroTrabajo132.vb*
ProyectoSa.CentroTrabajo132
pasarValoresDeListas(IDatos As List(Of Dato), IDatosBoo
1111 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),
1112     ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
1113     If banElementoFijo = False Then
1114         material132 = Module1.obtenerValorDatoString(lDatosString, "MATERIAL")
1115     End If
1116     _d1 = Module1.obtenerValorDatoDouble(lDatos, "diametroNominal132")
1117     _espesor = module1.obtenerValorDatoDouble(lDatos, "espesorEspaciador132")
1118     numespaciadores = Module1.obtenerValorDatoDouble(lDatos, "numeroEspaciadores")
1119 End Sub

```

Figura 33 Pasando valores de listas.

Una vez que se realizaron estos dos pasos habrá que extraer los materiales de la base de datos y se realiza de la siguiente forma:

```

CentroTrabajo132.vb*
ProyectoSa.CentroTrabajo132
Test(IDatos As List(Of Dato), IDatosBool As List(Of DatoBoolean), IDatosString As List(Of DatoString)
86 Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),
87     ByVal lDatosString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
88     material132 = Module1.obtenerValorDatoString(lDatosString, "MATERIAL")
89     tipo = Module1.sacar_tipo_material(material132)
90
91     If tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO GRIS" Or tipo = "HIERRO DUCTIL" Then
92     Else
93     MsgBox("Material " & tipo & " no disponible para el C.C 112 (" & 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 DUCTIL")
98     listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
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

```

Figura 34 Extrayendo los materiales de la base de datos.

Lo único que se debe de agregar o en su caso quitar son los elementos de la listaOpcionales de acuerdo a los materiales que se usan en este centro de trabajo.

Cabe resaltar que hace falta declara estas dos variables para que no haya ningún problema:

```

CentroTrabajo132.vb*
ProyectoSa.CentroTrabajo132
109 Dim tipo As String
110 Dim banElementoFijo As Boolean = False

```

Figura 35 Variables que se deben de declarar antes del método TEST.

Una vez realizado esto se debe crear un método dentro de la clase llamado llenar_material y se realiza como se muestra a continuación:

```

CentroTrabajo132.vb*
ProyectoSa.CentroTrabajo132
207
208 Private Sub llenar_material()
209     Dim l As New List(Of String)
210     l.Add("HIERRO GRIS")
211     l.Add("HIERRO GRIS CENTRIFUGADO")
212     l.Add("HIERRO DUCTIL")
213     For Each m In obtenerEspecTipoMaterial(l)
214         comboBox1.Items.Add(m)
215     Next
216 End Sub

```

Figura 36 Método llenar material.

Como se muestra en la figura 16 es la lista donde se van a agregar los materiales necesarios para este centro de trabajo. Además se debe cambiar la línea de código específicamente la palabra comboBox1 por el nombre que se le haya asignado al combo box que se agregó al formulario.

Una vez realizado todo lo anterior solo resta cambiar la formula por la que ya está actualizada. Lo primero que se debe de hacer es una condicional en donde dependiendo del material y el diámetro que introduzca el usuario es el valor que va a tomar el tiempo ciclo. Para que sea más claro observe la siguiente tabla:

	Tiempo ciclo según diámetro	Diámetro por tipo de Material	Tiempo ciclo según diámetro y tipo de material	Tiempo ciclo según diámetro	Diámetro por tipo de Material	Tiempo ciclo según diámetro y tipo de material
TIPO DE MATERIAL	$\Theta > 3.7$	Θ	TC $\Theta > 3.7$	$\Theta < 3.7$	Θ	TC $\Theta < 3.7$
HIERRO DUCTIL	283.5	4.44500	205.02	191.64	2.51730	244.72
HIERRO GRIS	90	4.56200	63.42	60	2.75000	70.13
HIERRO GRIS ALTO MOD	N/A	N/A	N/A	N/A	N/A	N/A
HIERRO GRIS INTERMEDIO	N/A	N/A	N/A	N/A	N/A	N/A

Tabla 4. Tabla de datos correspondiente a los materiales.

Cabe resaltar que el valor del tiempo ciclo según diámetro se obtiene mediante una operación matemática y no es un valor fijo como se muestra en la tabla.

Todo lo que se mencionó se ve de esta forma ya en código:

```

CentroTrabajo132.vb*
ProyectoSa.CentroTrabajo132
New(D1 As Double, mat132 As String, espesorEspaciador As Double)
165 Public Sub calcular() implements ICentroTrabajo.calcular
166 tiempo_septup=buscar_setupin(centro)
167 If banElementoFijo = False Then
168     tipo = Module1.sacar_tipo_material(material132)
169 End If
170 Dim tcsd,diatipomat,tcsdtm As Double
171 If tipo = "HIERRO GRIS" And _d1>3.7 Or tipo = "HIERRO GRIS CENTRIFUGADO" And _d1>3.7 Then
172     tcsd=90
173     diatipomat=4.562
174     tcsdtm=(_d1 * tcsd)/diatipomat
175 ElseIf tipo = "HIERRO DUCTIL" And _d1>3.7
176     tcsd=283.5
177     diatipomat=4.445
178     tcsdtm=(_d1 * tcsd)/diatipomat
179 ElseIf tipo = "HIERRO GRIS" And _d1<3.7 Or tipo = "HIERRO GRIS CENTRIFUGADO" And _d1<3.7
180     tcsd=60
181     diatipomat=2.75
182     tcsdtm=(_d1 * tcsd)/diatipomat
183 ElseIf tipo = "HIERRO DUCTIL" And _d1<3.7
184     tcsd=191.64
185     diatipomat=2.51730
186     tcsdtm=(_d1 * tcsd)/diatipomat
187 Else
188     msgbox("Material " & Tipo & " no disponible para el C.C 110, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
189 End If
190 Dim carga As Integer
191 Dim ciclo_por_carga As Double
192 carga = numespaciadores
193 ciclo_por_carga = (tcsdtm + 9.89 +6.79)

```

Figura 37 Método calcular del CT 132.

Como se puede ver en la figura 37 dentro del método se declararon las variables tcsd y diatipomat que almacenan los valores de la tabla que se mostró atrás, sin embargo como se mencionó anteriormente el valor de la variable tcsdtm se obtiene de una operación matemática. También se puede observar que se declaró la variable ciclo por carga que obtiene su valor de una sumatoria.

Ahora que ya se tienen todos elementos necesarios para poder realizar el cálculo del tiempo estándar del CT 132, solo resta agregar la formula actualizada al método y queda de la siguiente manera:

```

194 tiempo_maquina = Math.Round((100*((ciclo_por_carga)/3600)/carga)*100,3)

```

Figura 38 Formula actualizada para calcular el tiempo estándar.

Y ya con esto queda actualizado el tiempo estándar del centro de trabajo 132.

6.2.4 CentroTrabajo140.vb – DEGREASE RINGS

Para este centro de trabajo no hubo la necesidad de modificar su interfaz gráfica ya que con los elementos con los que cuenta son suficientes y útiles para realizar el cálculo del tiempo estándar.

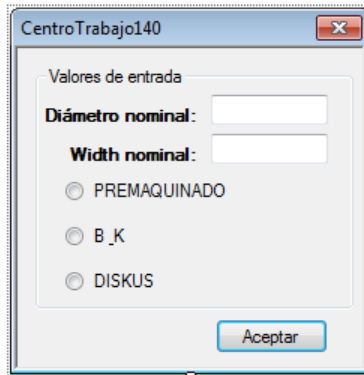


Figura 39 Interfaz gráfica del centro de trabajo.

Después de verificar que no hay necesidad de cambiar la interfaz gráfica ahora podemos realizar la siguiente actividad que consta de verificar el número de rieles que se van a usar de acuerdo al diámetro que el usuario ingrese en el formulario. Esta es la tabla donde se muestra el número de rieles a usar de acuerdo al rango en que se encuentre el diámetro.

Relación Diámetro-Rieles	Diámetro	N° de Rieles
1	0.7 - 4.2	12
2	4.3 - 5.59	5
3	5.6 - 6.93	3

Tabla 5. Tabla relación diámetro-rieles

De acuerdo a la tabla anterior solo existe un determinado rango para el diámetro que va desde 0.7 hasta 6.93 si el diámetro es inferior o superior a este rango no va a tener número de rieles y se mostrara un mensaje de error como el siguiente:

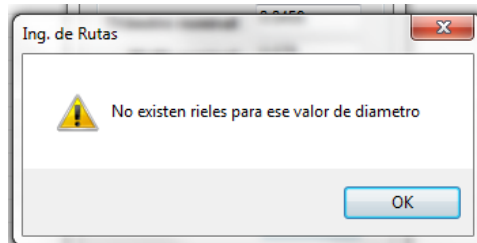
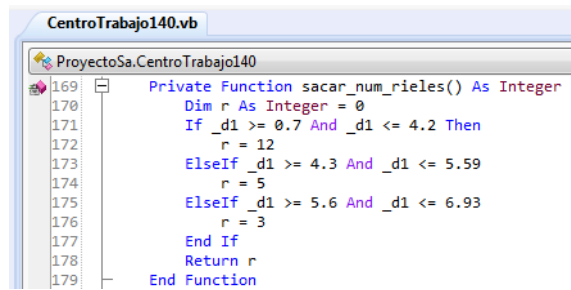


Figura 40 Error al insertar un valor de diámetro fuera de rango.

Todo lo mencionado anteriormente se ve en código de la siguiente manera:



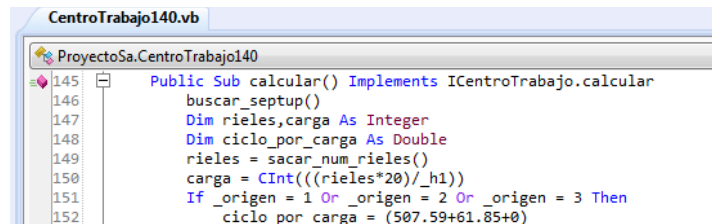
```

169 Private Function sacar_num_rieles() As Integer
170     Dim r As Integer = 0
171     If _d1 >= 0.7 And _d1 <= 4.2 Then
172         r = 12
173     ElseIf _d1 >= 4.3 And _d1 <= 5.59
174         r = 5
175     ElseIf _d1 >= 5.6 And _d1 <= 6.93
176         r = 3
177     End If
178     Return r
179 End Function

```

Figura 41 Función que obtiene sacar el número de rieles de acuerdo al valor del diámetro.

Como se muestra en la figura anterior se debe crear una función en la cual obtendremos el valor del número de rieles de acuerdo al valor que el usuario ingrese en el formulario. Una que creada la función se procede a realizar las siguientes operaciones y son las siguientes:



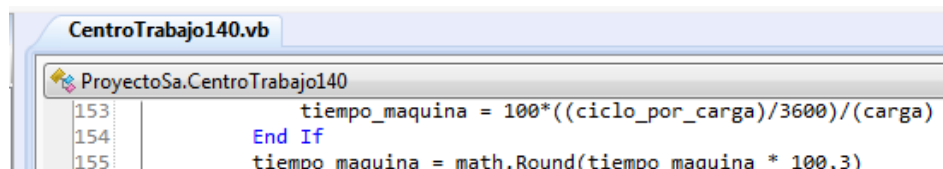
```

145 Public Sub calcular() Implements ICentroTrabajo.calcular
146     buscar_septup()
147     Dim rieles,carga As Integer
148     Dim ciclo_por_carga As Double
149     rieles = sacar_num_rieles()
150     carga = CInt(((rieles*20)/_h1))
151     If _origen = 1 Or _origen = 2 Or _origen = 3 Then
152         ciclo_por_carga = (507.59+61.85+0)

```

Figura 42 Método calcular del CT 140.

Dentro del método calcular se declararon 3 variables en la cuales se almacenan los valores necesarios para obtener el tiempo máquina del centro de trabajo. El valor de la variable rieles se obtiene de la función sacar_num_rieles() que es la función que se mencionó anteriormente, la variable carga obtiene su valor de una operación matemática, aquí se debe mencionar que se debe convertir el resultado de la operación a un número entero ya que si no se realiza la conversión se obtendría un número decimal y el resultado no sería el esperado. Por último el valor de a variable ciclo por carga se obtiene de una sumatoria. Ya que se tienen los valores de todas las variables solo resta agregar la formula actualizada.



```

153         tiempo_maquina = 100*((ciclo_por_carga)/3600)/(carga)
154     End If
155     tiempo maquina = math.Round(tiempo maquina * 100,3)

```

Figura 43 Formula actualizada del CT 140.

Una vez que se hizo todo lo anterior se debe realizar un último paso y es el siguiente:

```
CentroTrabajo140.vb*
ProyectoSa.CentroTrabajo140
200 Sub Button1Click(sender As Object, e As EventArgs)
201     If isnumeric(textBox1.Text) And isNumeric(Me.textBox2.Text) Then
202         If RB_BK.Checked = True Then
203             _origen = 1
204         ElseIf RB_DISKUS.Checked = True Then
205             _origen = 2
206         Else
207             _origen = 3
208         End If
209         Me._d1 = CDec(textBox1.Text)
210         Me._h1 = CDec(Me.textBox2.Text)
211         If _d1 >= 0.7 And _d1 <= 6.93 Then
212             calcular()
213             Close()
214         Else
215             msgbox("No existen rieles para ese valor de diametro",MsgBoxStyle.Exclamation,"Ing. de Rutas")
216         End If
217     Else
218         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
219     End If
220 End Sub
```

Figura 44 Evento clic del botón del formulario.

Como se muestra en la figura anterior se realiza una condicional en la cual si el valor de diámetro que ingrese el usuario no se encuentra en el rango establecido no se podrá realizar el cálculo y se mostrara un mensaje de error. Esto es lo que se mencionó al principio de la modificación de este centro. Una vez terminado este paso ya quedaría actualizado el centro de trabajo.

6.2.5 CentroTrabajo150.vb – FIN. GRD.

En este centro de trabajo no existió la necesidad de cambiar la interfaz gráfica del formulario ya que con los elementos que cuenta son más que necesarios para poder llevar a cabo el cálculo del tiempo estándar. El formulario es el siguiente:

Figura 45 Formulario correspondiente al CT 150.

Ahora que se aseguró que no es necesario cambiar la interfaz gráfica del formulario se debe realizar lo siguiente:

```

CentroTrabajo150.vb
ProyectoSa.CentroTrabajo150
167 Public Sub calcular() implements ICentroTrabajo.calcular
168     buscar_septup()
169     Dim t_ciclo2,t_ciclo1 As Double
170     If banElementoFijo = False Then
171         tipo = Module1.sacar_tipo_material(_material)
172     End If
173     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
174         t_ciclo1 = (0.3*12)/_rpm1
175         t_ciclo2 = (0.24*15)/_rpm2
176     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
177         t_ciclo1 = (0.52*4.15)/_rpm1
178         t_ciclo2 = (0.45*7.5)/_rpm2
179     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
180         t_ciclo1 = (0.49*7)/_rpm1
181         t_ciclo2 = (0.4*9)/_rpm2
182     ElseIf tipo = "HIERRO DUCTIL"
183         t_ciclo1 = (0.52*5)/_rpm1
184         t_ciclo2 = (0.44*8)/_rpm2
185     Else
186         msgbox("Material " & tipo & " no disponible para el C.C 150, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
187     Exit Sub
188 End If

```

Figura 46 Condicional donde se obtiene el tiempo ciclo de acuerdo al material.

Como se puede ver en la figura anterior se debe crear un if en el cual dependiendo el tipo de material, los valores del tiempo ciclo 1 y tiempo ciclo 2 son diferentes para cada material. Además como se puede observar los valores del tiempo ciclo 1 y del tiempo ciclo 2 se obtienen de una operación matemática. Estas operaciones se obtienen de la siguiente manera:

TIPO DE MATERIAL	TC 1	PM DE MUESTRA	TC 2	RPM DE MUESTRA
ACERO	N/D	N/D	N/D	N/D
HIERRO DUCTIL	0.52	5	0.44	8
HIERRO GRIS	0.72	12	0.45	15
HIERRO GRIS ALTO MOD	0.69	7	0.45	9
HIERRO GRIS INTERMEDIO	0.43	4.15	0.42	7.5

Figura 47 Forma de obtener la operación para calcular el tiempo ciclo 1 y 2.

Como lo muestra la figura anterior se debe seleccionar la celda del tiempo ciclo 1 o la celda del tiempo ciclo 2 correspondientes al material para ver la operación con la cual obtener su valor.

Una vez realizado esto solo hay que agregar la formula actualizada para calcular el tiempo estándar correspondiente a este centro de trabajo. La fórmula es la siguiente:

```

CentroTrabajo150.vb
ProyectoSa.CentroTrabajo150
189 tiempo_maquina = ( 0.01069 + ( ( t_ciclo1 / _rpm1 ) * 19 ) + ( t_ciclo2 / _rpm2 ) * 19 ) / 1332
190 tiempo_maquina = math.Round(tiempo_maquina * 100,3)

```

Figura 48 Formula actualizada para el cálculo del tiempo estándar correspondiente al CT 150.

6.2.6 CentroTrabajo230.vb – ROUGH CAM TURN

En este centro de trabajo se modificó se tuvo que modificar la interfaz gráfica debido a que el valor de entrada diámetro ya no es necesario para realizar el cálculo del tiempo estándar y por ello se eliminó. A continuación se muestra el formulario anterior y el actual.

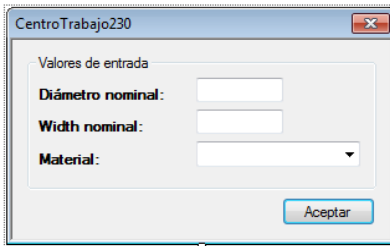


Figura 49 Formulario antes de la actualización.

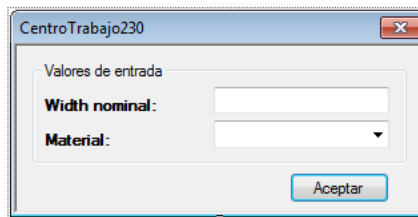


Figura 50 Formulario actual.

Como en este caso se eliminó el diámetro del formulario, entonces se tuvo que eliminar todo el código relacionado a este valor.

Primero se eliminó de la lista llamada listaDatosRequeridos, la cual se encuentra dentro de uno de los constructores de la clase. Además al momento de eliminar de esta lista también se eliminó del método pasarValoresDeListas(), quedando el código de la siguiente manera.

```
126 Public Sub New()  
127     Me.InitializeComponent()  
128  
129     Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "width nominal del anillo (Plano)", Nothing, "width nominal de anillo:")  
130     listaDatosRequeridos.Add(widthNominalAnillo)  
131  
132     Dim material_ As New DatoString("MATERIAL", "Especificación de materia prima (MF012-S, SPR-128, MF025-L, etc.)", "Material:")  
133     listaDatosStringRequeridos.Add(material_)  
134 End Sub
```

Figura 51 Constructor donde se agregan los valores a las listas.

```
119 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas  
120     _width = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")  
121     If banElementoFijo = False Then  
122         _material = module1.obtenerValorDatoString(lDatosString, "MATERIAL")  
123     End If  
124 End Sub
```

Figura 52 Método pasarValoreDeListas() del CT 230.

Los pasos anteriores se realizan con la finalidad de que al momento de que el usuario realiza la cotización de una lista de centros de trabajo aparezcan los valores de entrada para realizar el cálculo del tiempo estándar. A continuación se

muestra como se ven los valores de entrada cuando se agrega el centro de trabajo a la lista.

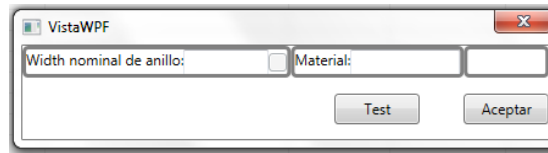


Figura 53 Valores de entrada agregados a las listas de datos requeridos.

Una vez que se realizó todo lo anterior también se tuvo que eliminar el parámetro del otro constructor de la clase, así como su variable donde se almacenaba el valor que el usuario ingresaba al formulario.

```

136 Public Sub New(ByVal H1 As Double, ByVal material_ As String)
137     _material = material_
138     _width = H1
139 End Sub

```

Figura 54 Constructor y parámetros actuales de la clase.

```

141 #Region "VARIABLES"
142 Dim _material As String
143 Dim _width As Double
144 #End Region

```

Figura 55 Variables donde se almacenan los valores de entrada.

Después de haber realizado todo lo anterior se procedió a modificar el método calcular en el cual se encuentra la fórmula para calcular el tiempo estándar. El código del método es el siguiente:

```

160 Public Sub calcular() Implements ICentroTrabajo.calcular
161     buscar_septup()
162     Dim t_ciclo As Double
163     If banElementoFijo = False Then
164         tipo = Module1.sacar_tipo_material(_material)
165     End If
166     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
167         t_ciclo = 60
168     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
169         t_ciclo = 60
170     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
171         t_ciclo = 210
172     ElseIf tipo = "HIERRO DUCTIL"
173         t_ciclo = 120
174     Else
175         msgbox("Material " & tipo & " no disponible para el C.C 230, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
176     End If
177     tiempo_maquina = Math.Round(((_width * (t_ciclo + 34.87324)) / 356.4) * 100,3)
178     tiempo_mano = tiempo_maquina
179 End Sub

```

Figura 56 Método calcular() del CT 230.

Como se puede apreciar en la imagen anterior lo primero que se debe de obtener es el tiempo ciclo, el cual se obtiene mediante una condición if en donde el valor del tiempo ciclo va a variar dependiendo del material que el usuario elija, y una vez obtenido el valor de este se procede a ejecutar la fórmula del tiempo estándar, la cual se almacena en la variable tiempo_maquina.

Y se finalizó con la modificación del evento clic del botón del formulario, eliminando la igualdad de la variable diámetro la cual ya se había eliminado con el TextBox que le correspondía a este valor.

```

237 Sub Button1Click(sender As Object, e As EventArgs)
238
239 If isnumeric(textBox1.Text) and isnumeric(textBox2.Text) Then
240     diametro = textBox2.Text
241     _width = textBox1.Text
242     _material = comboBox1.Text
243     calcular()
244     Close()
245 Else
246     msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
247 End If
End Sub

```

Figura 57 Evento clic del botón del formulario.

6.2.7 CentroTrabajo232.vb – ROUGH CAM TURN (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 se debe de buscar la carpeta llamada “Centrodetrabajo”, 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.

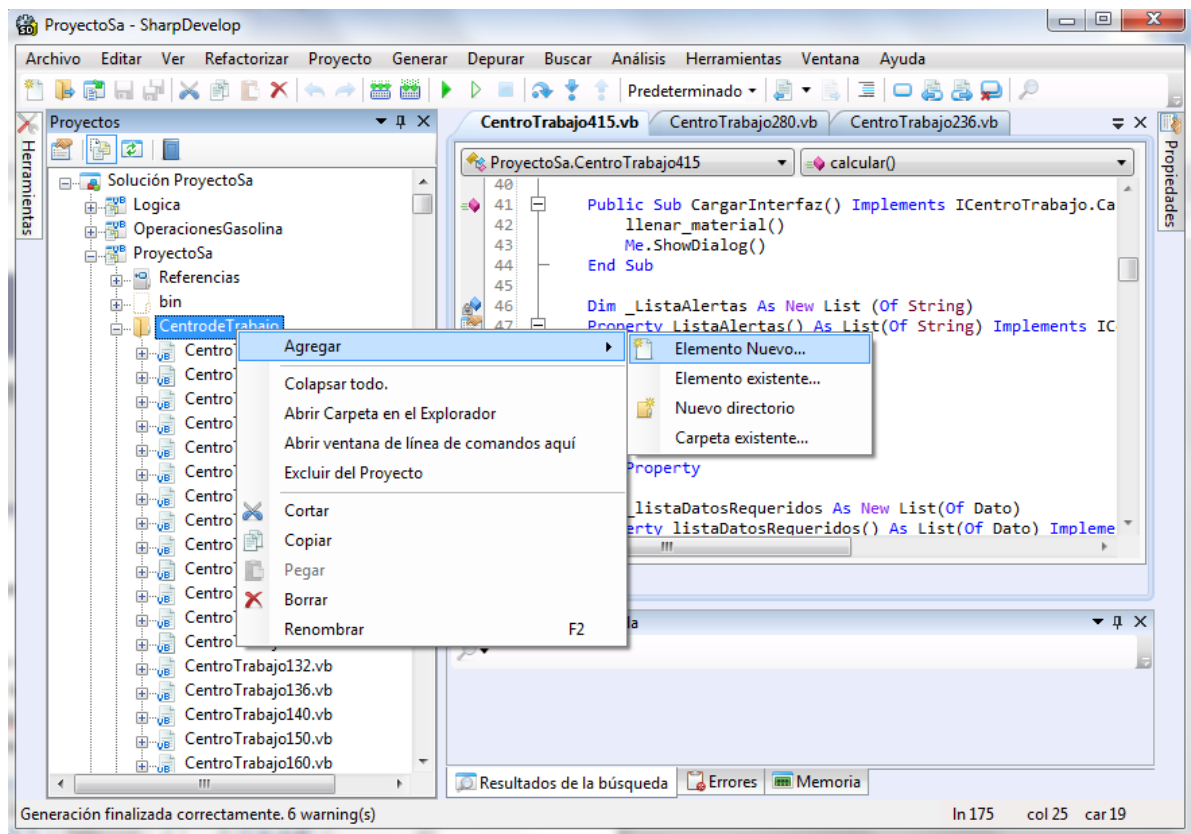


Figura 58 Agregar Nuevo Centro de Trabajo. - CT-232.

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 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 “CentroTrabajo232.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.

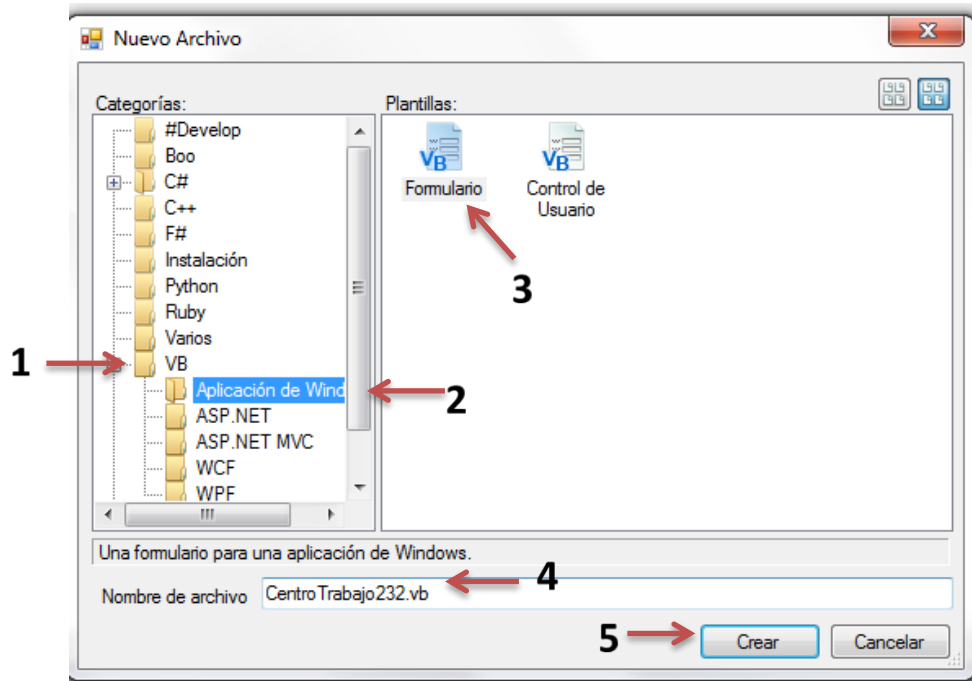


Figura 59 Agregar Centro de Trabajo. - CT-232.

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.

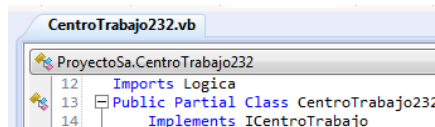


Figura 60 Imports Logica.

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).

Para la interfaz gráfica se agregaron dos valores de entrada que son necesarios para llevar a cabo el cálculo del tiempo estándar los cuales son Width nominal y material. El formulario quedó de la siguiente manera:

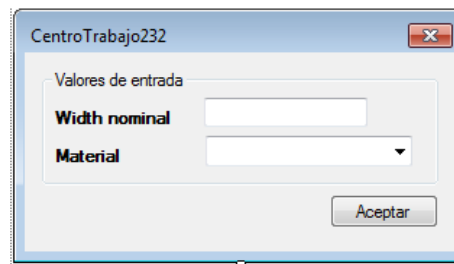


Figura 61 Formulario del CT 232.

Una vez que se realizó el formulario, se debe de agregar las propiedades de la interface ICentroTrabajo del proyecto lógica.

```

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     llenar_material()
46     Me.ShowDialog()
47 End Sub
48
49 Dim _ListaAlertas As New List(Of String)
50 Property ListaAlertas() As List(Of String) Implements ICentroTrabajo.ListaAlertas
51 Get
52     Return _ListaAlertas
53 End Get
54 Set
55     _ListaAlertas = Value
56 End Set
57 End Property
58
59 Dim _listaDatosRequeridos As New List(Of Dato)
60 Property listaDatosRequeridos() As List(Of Dato) Implements ICentroTrabajo.listaDatosRequeridos
61 Get
62     Return _listaDatosRequeridos
63 End Get
64 Set
65     _listaDatosRequeridos = value
66 End Set
67 End Property

```

Figura 62 Propiedades de la interface ICentroTrabajo.

Nota: en la imagen anterior no se muestran todas las propiedades, así que faltan más.

Después de haber realizado lo anterior se tienen que declarar las variables en las que se van a almacenar los valores de entrada y son las siguientes:

```

139 #Region "VARIABLES"
140 Dim _width As Double
141 Dim _material As String
142 #End Region

```

Figura 63 Variables donde se almacenan los valores de entrada.

Al finalizar la declaración de las variables se tienen que agregar los materiales que se van a usar en la función test, pero primero se tienen que declarar las siguientes variables:

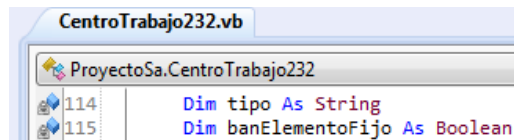


Figura 64 Variables necesarias para la función Test.

Después de haber declarado las variables se deben de agregar los materiales que se van a utilizar en este centro de trabajo se realiza de la siguiente manera:

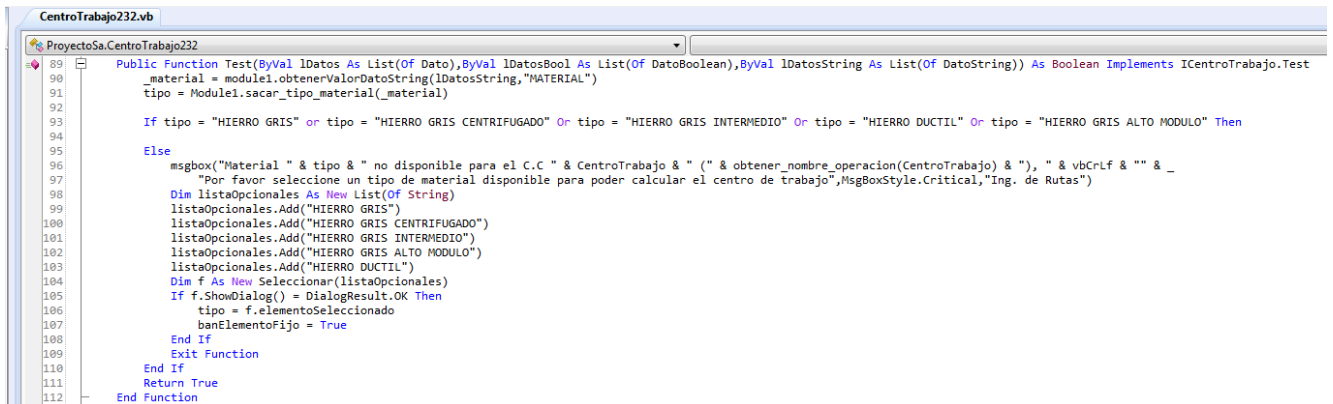


Figura 65 Función Test del CT 232.

Lo único que se debe de cambiar son los materiales que se van a usar.

Ahora se debe crear un método en el cual se van a agregar los materiales al ComboBox del formulario y se realiza de la siguiente manera:

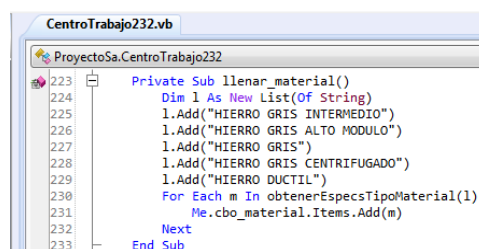


Figura 66 Método llenar_material().

Ahora se debe de crear el método calcular() en el cual van a estar todas las operaciones necesarias para poder realizar el cálculo del tiempo estándar.

```

CentroTrabajo232.vb
ProyectoSa.CentroTrabajo232
calcular()
158 Public Sub calcular() Implements ICentroTrabajo.calcular
159     buscar_septup()
160     If banElementoFijo = False Then
161         tipo = Module1.sacar_tipo_material(_material)
162     End If
163     Dim t_ciclo As Double
164     Dim carga As Integer
165     carga = CInt(6/_width)
166     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
167         t_ciclo = 180.68
168     ElseIf tipo = "HIERRO DUCTIL"
169         t_ciclo = 634
170     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
171         t_ciclo = 254.56
172     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
173         t_ciclo = 488.25
174     Else
175         msgbox("Material " & tipo & " no disponible para el C.C 232, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
176     End If
177     Dim t_total As Double
178     t_total = (t_ciclo + 30.31 + 29.25)
179     tiempo_maquina = Math.Round((100*(t_total/3600)/carga)*100,3)
180     tiempo_mano = tiempo_maquina
181 End Sub

```

Figura 67 Método calcular del CT 232 con su fórmula actualizada.

Lo que realiza el código anterior es una condición if en la cual el valor del tiempo ciclo va a variar dependiendo del material que se seleccione, y después de hacer la condición y obtener el valor, ahora si procede a realizar el cálculo del tiempo estándar.

También se debe de agregar el método buscar_septup() para extraer el tiempo septup de la base de datos y almacenar el valor en una variable del centro de trabajo.

```

CentroTrabajo232.vb
ProyectoSa.CentroTrabajo232
193 Private Sub buscar_septup()
194     Dim lector As SqlDataReader
195     Dim query As String
196     Dim cmd As New SqlCommand
197     query = "Select TiempoSetup " & _
198           "From CentroTrabajo " & _
199           "where CentroTrabajo = '" & centro & "'"
200     cmd = conecta(query)
201     Try
202         lector = cmd.ExecuteReader
203         While lector.Read
204             tiempo_septup = lector("TiempoSetup").ToString
205         End While
206     Catch er As Exception
207         module1.mandarError("Clase: " & Me.GetType().Name & vbCrLf & er.Message)
208     End Try
209     cn.Close()
210 End Sub

```

Figura 68 Método buscar_septup().

Ya lo último que se debe de hacer es agregarle el evento clic al botón del formulario. Este es el código:

```

CentroTrabajo232.vb
ProyectoSa.CentroTrabajo232
factorLabor
212 Sub Button1Click(sender As Object, e As EventArgs)
213     If IsNumeric(textBox1.Text) And cbo_material.Text <> "" Then
214         _width = textBox1.Text
215         _material = cbo_material.Text
216         calcular()
217         Close()
218     Else
219         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
220     End If
221 End Sub

```

Figura 69 Evento clic del botón del formulario.

Lo primero que se realiza en el código anterior es una condición if en la que si el usuario ingrese un valor no numérico o deja sin seleccionar un material se le va a mandar un mensaje de error y no se realizara el cálculo, sin embargo si hace todo lo contrario se realizará el cálculo sin problema.

6.2.8 CentroTrabajo235.vb – CAM TURN BORE MILL (CTB - 170 MACHINE)

En este centro de trabajo si fue necesario actualizar la interfaz gráfica del formulario correspondiente al centro de trabajo ya que son necesarios más elementos para poder efectuar el cálculo del tiempo estándar.

Figura 70 Formulario actualizado del CT 235.

Figura 71 Formulario antes de la actualización.

Una vez que se haya cambiado la interfaz gráfica del formulario se deben agregar las variables necesarias para que la clase funcione de manera correcta. Esto es lo que se debe realizar:

```

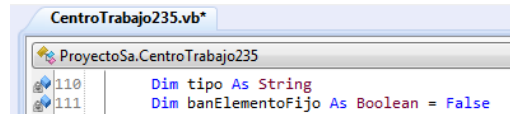
CentroTrabajo235.vb
95 Public Function Test(ByVal IDatos As List(Of Dato), ByVal IDatosBool As List(Of DatoBoolean), ByVal IDatosString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
96     material235 = Module1.obtenerValorDatoString(IDatosString, "MATERIAL")
97     tipo = module1.sacar_tipo_material(material235)
98     If tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS ALTO MODULO" Or tipo = "HIERRO GRIS INTERMEDIO" Or tipo = "HIERRO DUCTIL" Then
99         Else
100             MsgBox("Material " & tipo & " no disponible para el C.C 110 (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
101                 "Por favor seleccione un tipo de material disponible para poder calcular el centro de trabajo", MsgBoxStyle.Critical, "Ing. de Rutas")
102             Dim listaOpcionales As New List(Of String)
103             listaOpcionales.Add("HIERRO GRIS")
104             listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
105             listaOpcionales.Add("HIERRO GRIS ALTO MODULO")
106             listaOpcionales.Add("HIERRO DUCTIL")
107             listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
108             Dim f As New Seleccionar(listaOpcionales)
109             If f.ShowDialog() = DialogResult.OK Then
110                 tipo = f.elementoSeleccionado
111                 banElementoFijo = True
112             End If
113             Exit Function
114         End If
115     Return True
116 End Function

```

Figura 72 Función Test para agregar los materiales al centro de trabajo.

Este paso es repetitivo para los centros de trabajo que utilizan materiales y lo único que cambia son los materiales a usar en el centro de trabajo. Solo basta con agregar una línea similar a `listaOpcionales.add(“”)` y dentro de las comillas el nombre del material, esto para agregar un nuevo material. Para eliminar un material solo se debe eliminar dicha línea con el nombre del material que se quiere eliminar.

Antes de esto se deben de declarar las siguientes variables:



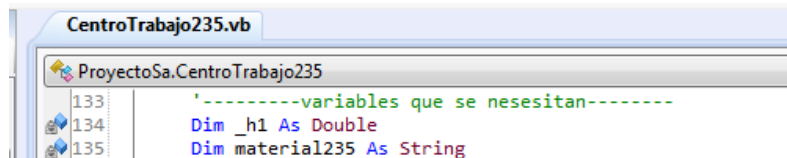
```
CentroTrabajo235.vb*
ProyectoSa.CentroTrabajo235
110 Dim tipo As String
111 Dim banElementoFijo As Boolean = False
```

Figura 73 Variables necesarias para la función Test.

Una vez realizado el paso anterior se deben realizar los siguientes cuatro pasos:

Paso 1.

Agregar una variable del tipo String la cual almacenara el material con el que se va a realizar el cálculo.

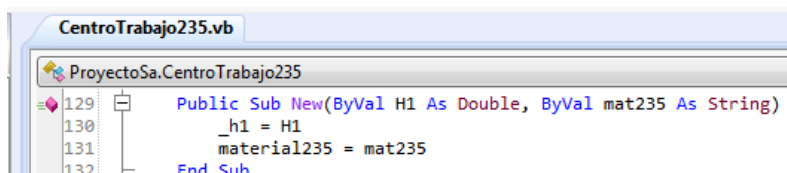


```
CentroTrabajo235.vb
ProyectoSa.CentroTrabajo235
133 '-----variables que se nesesitan-----
134 Dim _h1 As Double
135 Dim material235 As String
```

Figura 74 Variables necesarias para almacenar los valores que el usuario ingrese al formulario.

Paso 2.

Agregar un nuevo parámetro al constructor de la clase llamado mat235 e igualarlo con la variable material235.

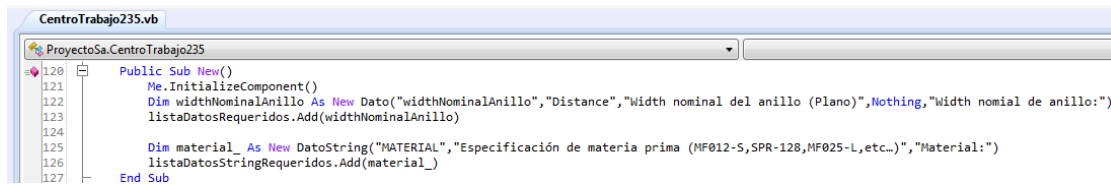


```
CentroTrabajo235.vb
ProyectoSa.CentroTrabajo235
129 Public Sub New(ByVal H1 As Double, ByVal mat235 As String)
130     _h1 = H1
131     material235 = mat235
132 End Sub
```

Figura 75 Agregando un nuevo parámetro al constructor de la clase.

Paso 3.

Agregar el elemento material a la listaDatosStringRequeridos y se realiza de la siguiente manera:



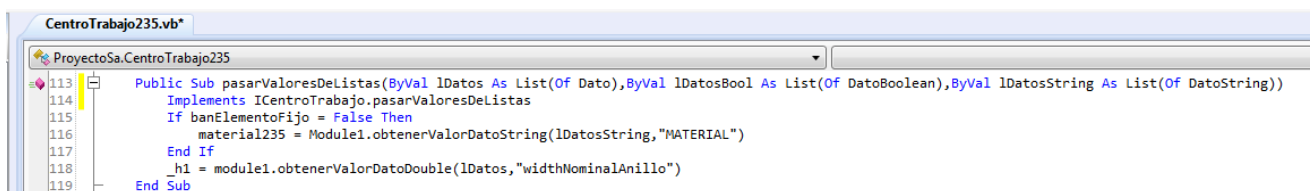
```
120 Public Sub New()  
121     Me.InitializeComponent()  
122     Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "Width nominal del anillo (Plano)", Nothing, "Width nominal de anillo:")  
123     listaDatosRequeridos.Add(widthNominalAnillo)  
124  
125     Dim material_ As New DatoString("MATERIAL", "Especificación de materia prima (MF012-S, SPR-128, MF025-L, etc...)", "Material:")  
126     listaDatosStringRequeridos.Add(material_)  
127 End Sub
```

Figura 76 Agregando el elemento material a la ListaDatosStringRequeridos.

En la figura anterior se muestra como se agrega un nuevo elemento a la lista mencionada, el nombre de la variable puede ser cualquiera.

Paso 4.

El último paso a realizar es lo que se muestra a continuación:

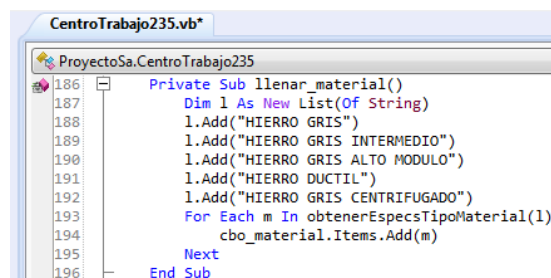


```
113 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString))  
114     Implements ICentroTrabajo.pasarValoresDeListas  
115     If banElementoFijo = False Then  
116         material235 = Module1.obtenerValorDatoString(lDatosString, "MATERIAL")  
117     End If  
118     _h1 = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")  
119 End Sub
```

Figura 77 Método publico pasarValoresDeListas.

Aquí lo que se realiza es igualar las variables que se declararon en el paso 1 con un método del modulo1 del ProyectoSa, el cual obtiene los valores que se agregaron a la listaDatosStringRequeridos.

Después de haber realizado los pasos anteriores se debe realizar esta otra actividad. En esta actividad que también es repetitiva para cualquier centro de trabajo que utilice materiales. Se debe crear un método llamado llenar_material el cual crea una lista en la cual se almacenan los materiales que se van a utilizar en este centro de trabajo. El código se muestra a continuación:



```
186 Private Sub llenar_material()  
187     Dim l As New List(Of String)  
188     l.Add("HIERRO GRIS")  
189     l.Add("HIERRO GRIS INTERMEDIO")  
190     l.Add("HIERRO GRIS ALTO MODULO")  
191     l.Add("HIERRO DUCTIL")  
192     l.Add("HIERRO GRIS CENTRIFUGADO")  
193     For Each m In obtenerEspecTipoMaterial(l)  
194         cbo_material.Items.Add(m)  
195     Next  
196 End Sub
```

Figura 78 Método llenar material.

Una vez que se haya realizado esta actividad lo siguiente es mandar llamar este método por medio de un objeto, agregando dicho objeto al método CargarInterfaz como se muestra a continuación:

```

CentroTrabajo235.vb*
ProyectoSa.CentroTrabajo235
41 Public Sub CargarInterfaz() Implements ICentroTrabajo.CargarInterfaz
42     llenar_material()
43     Me.ShowDialog()
44 End Sub

```

Figura 79 Método CargarInterfaz en el cual se manda a llamar el método llenar_material.

Ahora sí que ya se tiene todo lo necesario, es hora de realizar las operaciones necesarias para obtener el valor del tiempo estándar.

Ya estando en el método calcular de la clase, se crea un condición if en donde el valor del tiempo ciclo varía dependiendo del material que se vaya a utilizar.

La condición se muestra en la siguiente imagen:

```

CentroTrabajo235.vb
ProyectoSa.CentroTrabajo235
llenar_material()
153 Public Sub calcular() implements ICentroTrabajo.calcular
154     tiempo_septup=buscar_setupin(centro)
155     If banElementoFijo = False Then
156         tipo = Module1.sacar_tipo_material(material235)
157     End If
158     Dim tcm As Double
159     If tipo = "HIERRO DUCTIL" Then
160         tcm = 362.35
161     ElseIf tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO"
162         tcm = 117.28
163     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
164         tcm = 514.35
165     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
166         tcm = 189.67
167     Else
168         MsgBox("Material " & tipo & " no disponible para el C.C 110, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
169     End If

```

Figura 80 Condición if para obtener el valor del tiempo ciclo.

Una vez realizado el if ya tenemos todas las variantes para poder realizar el cálculo del tiempo estándar, y lo único que se tendría que hacer es agregar la formula actualizada para calcular el tiempo estándar y es la siguiente:

```

CentroTrabajo235.vb
ProyectoSa.CentroTrabajo235
170     tiempo_maquina = (_h1*(tcm+47.32))/216
171     tiempo_maquina = (tiempo_maquina*100)

```

Figura 81 Formula actualizada para calcular el tiempo estándar del CT 235.

Obviamente la segunda fórmula que aparece en la imagen es la misma solo que se tiene que multiplicar por cien por cuestiones de la empresa, además esto que se comentó se debe de realizar para todos los centros de trabajo sin excepción alguna.

6.2.9 CentroTrabajo285.vb – BATES BORE (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 se debe de buscar la carpeta llamada "CentroTrabajo", 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.

Lo siguiente es hacer lo mismo que se hizo en otros centros de trabajo que se agregaron desde cero. Por ejemplo CentroTrabajo232.vb – ROUGH CAM TURN (HIGH RUNNERS), hasta la parte donde se crea la interfaz gráfica.

Y como se menciona arriba se debe crear la interfaz gráfica del centro de trabajo de acuerdo con los datos que requiera la fórmula para llevar a cabo el cálculo del tiempo estándar. Entonces así es como queda la interfaz gráfica:

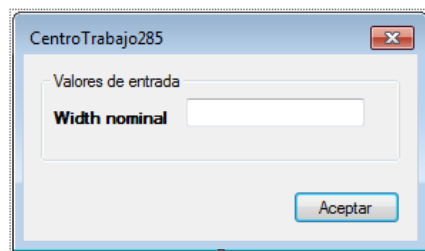


Figura 82 Formulario del CT 285.

Como se puede apreciar en la imagen anterior el único valor de entrada que se ocupa es el Width.

Al finalizar de crear la interfaz gráfica, dentro de la codificación se tiene que declarar una variable en la cual se almacenara el valor introducido por el usuario y se muestra en la siguiente imagen.

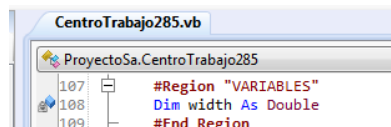


Figura 83 Variable donde se almacena el valor de entrada.

Ahora hay que igualar las variables con los parámetros de la clase, esto se hace como se muestra a continuación:

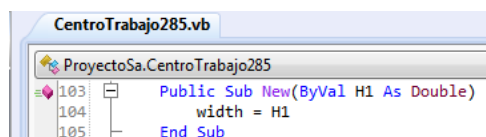


Figura 84 Igualando variables con parámetros.

Después de igualar se deben seguir los siguientes pasos que son obligatorios ya que si no se hacen el software no se ejecutara. El primer paso consta de declarar un número de variables de tipo objeto igual a los valores de entrada como se muestra en la siguiente imagen:

```
196 Public Sub New()  
197     Me.InitializeComponent()  
198  
199     Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "width nominal del anillo", Nothing, "width del anillo:")  
200     listaDatosRequeridos.Add(widthNominalAnillo)  
201 End Sub
```

Figura 85 Variables que se agregan a la lista de datos requeridos.

El segundo paso es agregar las variables necesarias al método pasarValoresDeListas e igualarlas con los métodos obtenerValorDatoDouble u obtenerValorDatoString dependiendo del tipo de dato que se le asignó a la variable, estos métodos pertenecen al module1 del proyecto. Este paso se ve de la siguiente manera:

```
91  
92 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString))  
93     Implements ICentroTrabajo.pasarValoresDeListas  
94  
95     width = Module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")  
96 End Sub
```

Figura 86 Pasando valores de listas.

Una vez que se realizó todo lo anterior se procede a crear el método calcular en donde se encontraran todas las operaciones necesarias para poder llevar a cabo el cálculo del tiempo estándar.

```
127 Public Sub calcular() Implements ICentroTrabajo.calcular  
128     tiempo_septup = buscar_setupin(centro)  
129     tiempo_maquina = Math.Round( ( ( width * 24.32 ) / 36 ) / 2 ) * 100,3)  
130     tiempo_mano = tiempo_maquina  
131 End Sub
```

Figura 87 Método calcular del CT 285.

Como se puede ver en la imagen anterior el tiempo septup se obtiene del método buscar_setupin() del module1 del proyecto. Y la fórmula del tiempo estándar es la variable tiempo_maquina la cual obtiene su valor de una operación matemática.

Ya lo último que queda por hacer es agregar el evento clic para que cuando el usuario le dé clic al botón realice la operación.

```
145 Sub Button1Click(sender As Object, e As EventArgs)  
146     If IsNumeric(textBox1.Text) Then  
147         width = textBox1.Text  
148         calcular()  
149         Close()  
150     Else  
151         msgbox("Revisar valores", MsgBoxStyle.Exclamation, "Ing. de Rutas")  
152     End If  
153 End Sub
```

Figura 88 Evento clic del botón del formulario.

Lo que realiza el código anterior es una condición if en la que si el usuario ingresa un valor numérico realiza la operación, sino no la realiza y manda un mensaje de error.

6.2.10 CentroTrabajo421.vb – CNC AUTO. FIN TURN

Para la actualización de este centro de trabajo se tuvo que eliminar su interfaz gráfica ya que no es necesaria, porque este centro de trabajo utiliza un solo material, además de que el tiempo ciclo no es variable y siempre es el mismo. Por eso es mejor realizar el cálculo de manera directa.

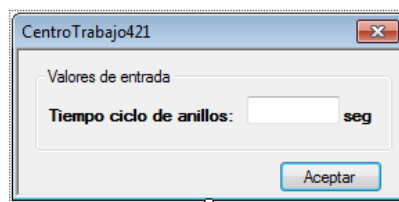


Figura 89 Formulario antes de la eliminación de sus elementos.

Una vez que se eliminaron los elementos del formulario se le asignó el valor a la variable tc_anillo y se actualizó la fórmula como se muestra en las siguientes imágenes:

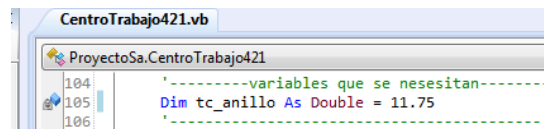


Figura 90 Asignando valor al tiempo ciclo.

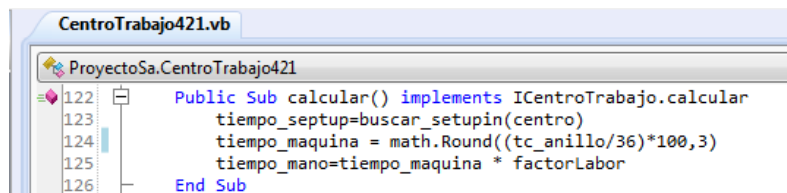


Figura 91 Fórmula actualizada del CT 421.

Una vez realizado todo lo anterior este centro de trabajo queda actualizado.

6.2.11 CentroTrabajo423.vb – KATAOKA (ANILLOS THM)

A este centro de trabajo se le tuvo que crear una interfaz gráfica ya que anteriormente el cálculo se realizaba directamente al cargar el centro de trabajo pero con la actualización fue necesario agregar ciertos elementos a los valores de entrada para poder llevar a cabo el cálculo del tiempo estándar, el nuevo aspecto del formulario quedó de la siguiente manera:

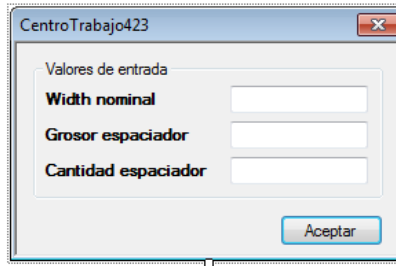


Figura 92 Formulario actual del CT 423.

Después de crear la interfaz del centro de trabajo, ahora hay que asignarles una variable a cada valor de entrada, además estas variables son muy útiles en toda la clase. Estas son las variables:

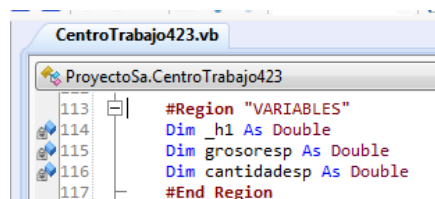


Figura 93 Variables correspondientes a cada valor de entrada.

Una vez que se declararon las variables se tiene que agregar las líneas de código que se mencionaron en otros centros de trabajo las cuales son necesarias para el buen funcionamiento de la clase y son las siguientes:

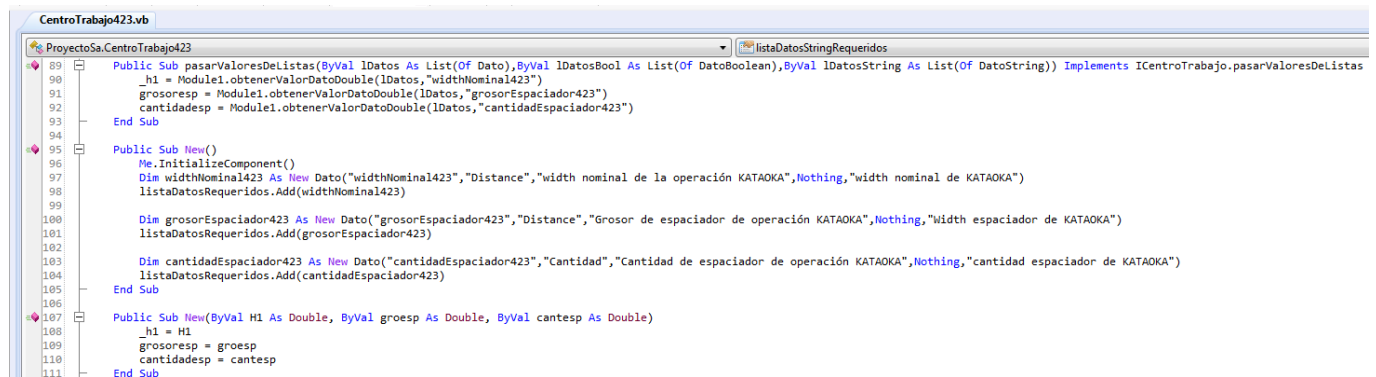


Figura 94 Pasando valores a la interface ICentroTrabajo del proyecto Logica.

Se debe mencionar que lo que se realizó se tiene que hacer siempre que se agreguen nuevos valores de entrada al formulario del centro de trabajo.

Después de haber realizado todo lo anterior ya se tienen todos los elementos para llevar a cabo el cálculo del tiempo estándar. Para ello se debe de dirigir al método calcular() donde se encuentra la formula. El código es el siguiente:

```

CentroTrabajo423.vb
ProyectoSa.CentroTrabajo423
Public Sub calcular() implements ICentroTrabajo.calcular
135     tiempo_septup = buscar_setupin(centro)
136     tiempo_maquina = (150.88 * _h1) / (36 * (3.5087 - (grosoresp * cantidadesp)))
137     tiempo_maquina = Math.Round(tiempo_maquina *100,3)
138     tiempo_mano = tiempo_maquina
139 End Sub

```

Figura 95 Método calcular() donde se lleva a cabo la operación para obtener el tiempo estándar.

Ahora que ya se tiene todo listo en el método calcular solo resta agregar unas instrucciones al evento clic correspondiente al botón del formulario y esto se hace como se muestra a continuación:

```

CentroTrabajo423.vb
ProyectoSa.CentroTrabajo423
151 Sub Button1Click(sender As Object, e As EventArgs)
152     If IsNumeric(textBox1.Text) Or IsNumeric(textBox2.Text) Or IsNumeric(textBox3.Text) Then
153         _h1 = textBox1.Text
154         grosoresp = textBox2.Text
155         cantidadesp = textBox3.Text
156         calcular()
157         Me.Close()
158     Else
159         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
160     End If
161 End Sub

```

Figura 96 Evento clic del botón del formulario.

Lo que realiza el código anterior, es almacenar los valores que introduce el usuario a las variables que se declararon desde el principio y además se realiza una condición if en la que si el usuario introduce un valor que no es numérico no deja realizar el cálculo y manda un mensaje de error pidiendo que se revisen los valores introducidos.

6.2.12 CentroTrabajo433.vb – SIMPLEX

A este centro de trabajo no se le hizo ninguna modificación ni en su interfaz gráfica, ni en su código, por lo tanto esta actualizado. Esta es su interfaz gráfica y la su fórmula.

Figura 97 Formulario del CT 433.

```

CentroTrabajo433.vb
ProyectoSa.CentroTrabajo433
Public Sub calcular() implements ICentroTrabajo.calcular
122     tiempo_septup=buscar_setupin(centro)
123     tiempo_maquina = math.Round((( 18.32 + ( diametro * 135 ) / 4.75 ) / 144 ) * 100,3)
124     tiempo_mano = tiempo_maquina
125 End Sub

```

Figura 98 Método calcular y formula del tiempo estándar del CT 433.

6.2.13 CentroTrabajo435.vb – GRIND CHANNEL (NORTON)

En este centro de trabajo se tuvo que cambiar su interfaz gráfica ya que la anterior no contaba con los valores de entrada necesarios para poder llevar a cabo el cálculo del tiempo estándar. Entonces quedo de la siguiente manera:

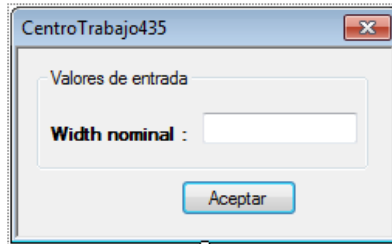


Figura 99 Formulario antes de la actualización.

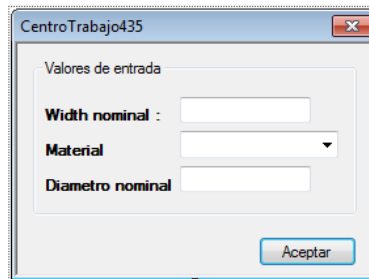


Figura 100 Formulario actualizado.

Después de que se modifica la interfaz del formulario se procede a declarar las variables donde se almacenaran los nuevos valores de entrada y se hace de la siguiente forma:

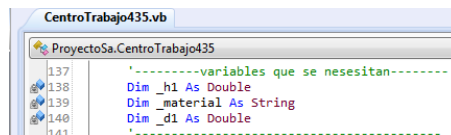


Figura 101 Variables donde se almacenan los diferentes valores.

Ahora se deben de agregar dos nuevos parámetros al constructor de la clase ya que se agregaron dos nuevos valores. También hay que igualarlos con las variables que se declararon.

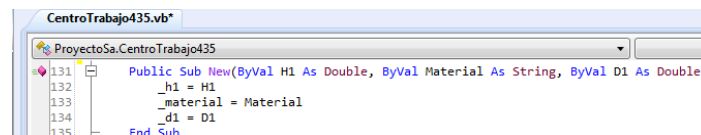


Figura 102 Igualando variables con parámetros.

El primer paso consta de declarar un número de variables de tipo objeto igual a los valores de entrada como se muestra en la siguiente imagen:

```

CentroTrabajo435.vb*
ProyectoSa.CentroTrabajo435
119 Public Sub New()
120     Me.InitializeComponent()
121     Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "Width nominal del anillo (Plano)", Nothing, "width nominal de anillo:")
122     listaDatosRequeridos.Add(widthNominalAnillo)
123
124     Dim material_ As New DatoString("MATERIAL", "MATERIAL Componente", "Material:")
125     listaDatosStringRequeridos.Add(material_)
126
127     Dim diametroNominalAnillo As New Dato("diametroNominalAnillo", "Distance", "Diametro nominal del anillo", Nothing, "diametro nominal del anillo")
128     listaDatosRequeridos.Add(diametroNominalAnillo)
129 End Sub

```

Figura 103 Variables que se agregan a la lista de datos requeridos.

El segundo paso es agregar las variables necesarias al método pasarValoresDeListas e igualarlas con los métodos obtenerValorDatoDouble u obtenerValorDatoString dependiendo del tipo de dato que se le asignó a la variable, estos métodos pertenecen al module1 del proyecto. Este paso se ve de la siguiente manera:

```

CentroTrabajo435.vb*
ProyectoSa.CentroTrabajo435
pasarValoresDeListas(IDatos As List(Of Dato), IDatosBool As List(Of DatoBoolean), IDatosString As List(Of DatoString))
111 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString))
112     Implements ICentroTrabajo.pasarValoresDeListas
113     If banElementoFijo = False Then
114         _material = module1.obtenerValorDatoString(lDatosString, "MATERIAL")
115     End If
116     _h1 = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")
117     _d1 = Module1.obtenerValorDatoDouble(lDatos, "diametroNominalAnillo")
118 End Sub

```

Figura 104 Pasando valores de listas.

Después de haber realizado todo lo anterior se deben de agregar los materiales que se van a utilizar a la función Test, sin embargo antes de realizar esto se tienen que declarar las siguientes variables:

```

CentroTrabajo435.vb*
ProyectoSa.CentroTrabajo435
108 Dim tipo As String
109 Dim banElementoFijo As Boolean = False

```

Figura 105 Variables que se deben de declarar antes del método Test.

Después de declarar las variables ahora si se agregan los materiales a la función Test como se muestra a continuación:

```

CentroTrabajo435.vb*
ProyectoSa.CentroTrabajo435
pasarValoresDeListas(IDatos As List(Of Dato), IDatosBool As List(Of DatoBoolean), IDatosString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
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 ICentroTrabajo.Test
87     _material = Module1.obtenerValorDatoString(lDatosString, "MATERIAL")
88     tipo = Module1.sacar_tipo_material(_material)
89
90     If tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO GRIS" Or tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Then
91     Else
92         msgbox("Material " & tipo & " no disponible para el C.C 110 (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
93         "Por favor seleccione un tipo de material disponible para poder calcular el centro de trabajo", MsgBoxStyle.Critical, "Ing. de Rutas")
94         Dim listaOpcionales As New List(Of String)
95         listaOpcionales.Add("HIERRO GRIS")
96         listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
97         listaOpcionales.Add("ACERO INOXIDABLE")
98         listaOpcionales.Add("ACERO AL CARBON")
99         Dim f As New Selectionar(listaOpcionales)
100         If f.ShowDialog() = DialogResult.OK Then
101             tipo = f.elementoSeleccionado
102             banElementoFijo = True
103         End If
104     End If
105     Return True
106 End Function

```

Figura 106 Agregando materiales a la función Test.

Ahora se debe agregar el método llamado `llenar_material()` donde se crea una lista con los materiales que se van a utilizar en el centro de trabajo y los elementos de esta se agregan al `comboBox` correspondiente al material. El código es el siguiente.

```

187 Private Sub llenar_material()
188     Dim l As New List(Of String)
189     l.Add("HIERRO GRIS")
190     l.Add("HIERRO GRIS CENTRIFUGADO")
191     l.Add("ACERO INOXIDABLE")
192     l.Add("ACERO AL CARBON")
193     For Each m In obtenerEspecTipoMaterial(1)
194         comboBox1.Items.Add(m)
195     Next
196 End Sub
    
```

Figura 107 Método `llenar_material()`.

Ahora se debe agregar el objeto `llenar_material()` al método `CargarInterfaz()` como se muestra en la siguiente imagen.

```

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

Figura 108 Método `CargarInterfaz()` del CT 435.

Ahora se tiene que modificar el método `calcular()` con las operaciones necesarias para llevar a cabo el cálculo del tiempo estándar. A continuación se muestra el método.

```

158 Public Sub calcular() Implements ICentroTrabajo.calcular
159     tiempo_septup=buscar_setupin(centro)
160     If banElementoFijo = False Then
161         tipo = Module1.sacar_tipo_material(_material)
162     End If
163     Dim t_ciclo_muestra As Double
164     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
165         t_ciclo_muestra = 65.02
166     ElseIf tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON"
167         t_ciclo_muestra = 75.59
168     Else
169         MsgBox("Material " & tipo & " no disponible para el C.C 435, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
170     Exit Sub
171     End If
172     tiempo_maquina = (_h1*((t_ciclo_muestra * _d1)/3.8588)+140.75))/207
173     tiempo_maquina = Math.Round(tiempo_maquina * 100,3)
174     tiempo_mano=tiempo_maquina
175 End Sub
    
```

Figura 109 Método `calcular` donde se encuentra la fórmula para el cálculo del tiempo estándar.

Lo que realiza el código anterior es que hace una condición `if` en la que el valor del tiempo ciclo varía dependiendo del tipo de material que seleccione el usuario. Después de que se realizó la condición y se obtuvo el valor del tiempo ciclo ahora si se realiza el cálculo ya que tiene todos los valores para llevarse a cabo.

Ya por último se debe de modificar el evento clic del botón del formulario ya que se agregaron nuevos valores de entrada. Este es el código:


```
CentroTrabajo435.vb
ProyectoSa.CentroTrabajo435
197 Sub Button1Click(sender As Object, e As EventArgs)
198     If isnumeric(Me.textBox1.Text) And IsNumeric(Me.textBox2.Text) And comboBox1.Text <> "" Then
199         Me._h1 = CDec(Me.textBox1.Text)
200         Me._d1 = CDec(Me.textBox2.Text)
201         Me._material = comboBox1.Text
202         Me.calcular()
203         Me.Close()
204     Else
205         msgbox("Revisar valores")
206     End If
207 End Sub
```

Figura 110 Evento clic del botón del formulario del CT 435.

Lo que realiza el código anterior es que cuando el usuario inserta valores numéricos y selecciona un material se realiza el cálculo, además los valores que inserta el usuario los almacena en su respectiva variable, sin embargo sino se insertan valores no numéricos y no se selecciona un material no se realiza el cálculo.

6.2.14 CentroTrabajo440.vb – TAPER SIDE

A este centro de trabajo se le tuvo que modificar su interfaz gráfica ya que anteriormente se solicitaba el diámetro y el material para poder obtener el tiempo estándar, sin embargo con la actualización, ahora solo se le solicita al usuario que seleccione el material para obtener el tiempo estándar. En las siguientes figuras se muestra el formulario anterior y el actual.

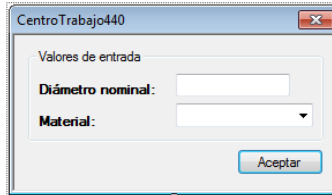


Figura 111 Formulario antes de la actualización.

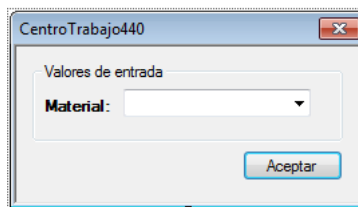


Figura 112 Formulario actual.

Como se eliminó un valor de entrada que es diámetro, entonces se tuvo que eliminar todo el código referente a este. Lo primero que se removió fue la variable principal y el parámetro del constructor que le correspondía al diámetro.

```

CentroTrabajo440.vb
ProyectoSa.CentroTrabajo440
126 Public Sub New(ByVal Material As String)
127     _material = Material
128 End Sub
129
130 '-----variables que se necesitan-----
131 Dim _material As String
132 '-----

```

Figura 113 Constructor de la clase actual.

Posteriormente se eliminó el diámetro de la listaDatosRequeridos y después se eliminó del método pasarValoresDeListas.

```

CentroTrabajo440.vb
ProyectoSa.CentroTrabajo440
114 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
115     If banElementoFijo = False Then
116         _material = module1.obtenerValorDatoString(lDatosString, "MATERIAL")
117     End If
118 End Sub
119
120 Public Sub New()
121     Me.InitializeComponent()
122     Dim material_ As New DatoString("MATERIAL", "MATERIAL Componente", "Material:")
123     listaDatosStringRequeridos.Add(material_)
124 End Sub

```

Figura 114 Método pasarValoresDeListas() y listaDatosRequeridos actuales.

Después se procedió a modificar el método calcular() que es donde se encuentran todas las operaciones necesarias para obtener el valor del tiempo estándar. A continuación se muestra el código completo.

```

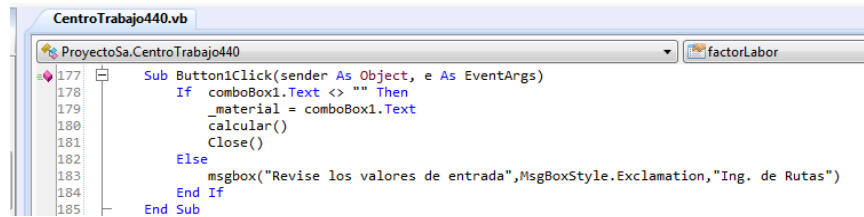
CentroTrabajo440.vb
ProyectoSa.CentroTrabajo440
147 Public Sub calcular() implements ICentroTrabajo.calcular
148     tiempo_septup = buscar_setupin(centro)
149     Dim t_ciclo As Double
150     If banElementoFijo = False Then
151         tipo = Module1.sacar_tipo_material(_material)
152     End If
153     If tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Then
154         t_ciclo = 84.98
155     ElseIf tipo = "HIERRO DUCTIL"
156         t_ciclo = 62.11
157     ElseIf tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO"
158         t_ciclo = 61.296
159     Else
160         MsgBox("Material " & tipo & " no disponible para el C.C 440, Imposible calcular", MsgBoxStyle.Critical, "Ing. de Rutas")
161         Exit Sub
162     End If
163     tiempo_maquina = Math.Round(((t_ciclo + 19.987042) / 36) * 100, 3)
164     tiempo_mano=tiempo_maquina
165 End Sub

```

Figura 115 Método calcular() del CT 440.

Como en todos los centros de trabajo lo primero que se realiza en este método es obtener el tiempo setup el cual se obtiene de la tabla CentroTrabajo que pertenece a la base de datos RGP2, después se obtienen los materiales también de la base de datos. A continuación se realiza una condición if en la que el valor de la variable t_ciclo va a variar dependiendo del material que el usuario elija, y una vez que se obtiene el valor de la variable antes mencionada se procede a realizar el cálculo de la fórmula del tiempo estándar, la cual está almacenada en la variable tiempo máquina, y para finalizar el tiempo mano se iguala al tiempo máquina.

Para finalizar la actualización de este centro de trabajo se modificó el evento clic del botón del formulario donde lo único que se realizó fue eliminar todo lo referente al diámetro quedando el código de la siguiente manera.



```
177 Sub Button1Click(sender As Object, e As EventArgs)
178     If comboBox1.Text <> "" Then
179         _material = comboBox1.Text
180         calcular()
181         Close()
182     Else
183         msgbox("Revise los valores de entrada",MsgBoxStyle.Exclamation,"Ing. de Rutas")
184     End If
185 End Sub
```

Figura 116 Evento clic del botón del formulario.

6.2.15 CentroTrabajo460.vb – WASH

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

Se debe seguir todo el protocolo que se ha hecho en centros de trabajo anteriores por ejemplo CentroTrabajo232.vb – ROUGH CAM TURN (HIGH RUNNERS), hasta la parte del diseño de la interfaz gráfica.

A esta interfaz gráfica se agregaron dos valores de entrada los cuales son Width nominal y diámetro nominal los cuales son necesarios para realizar el cálculo del tiempo estándar, entonces el formulario quedo de la siguiente manera:

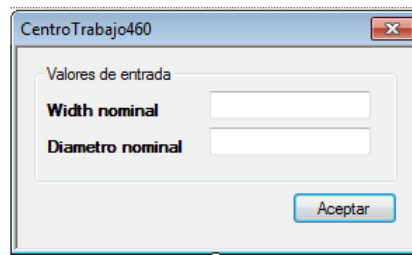
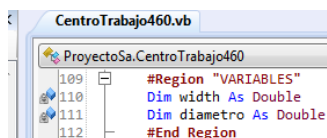


Figura 117 Formulario correspondiente al CT 460.

Una vez creado el formulario se deben de procesar los datos de entrada mediante variables, las cuales se muestran a continuación.



```
109 #Region "VARIABLES"
110 Dim width As Double
111 Dim diametro As Double
112 #End Region
```

Figura 118 Variables donde se almacenaran los valores de entrada.

Ahora se deben de agregar dos nuevos parámetros al constructor de la clase ya que se agregaron dos nuevos valores. También hay que igualarlos con las variables que se declararon.

```

CentroTrabajo460.vb
ProyectoSa.CentroTrabajo460
104 Public Sub New(Byval H1 As Double, Byval D1 As Double)
105     width = H1
106     diametro = D1
107 End Sub

```

Figura 119 Igualando parámetros con variables.

El primer paso consta de declarar un número de variables de tipo objeto igual a los valores de entrada como se muestra en la siguiente imagen:

```

CentroTrabajo460.vb
ProyectoSa.CentroTrabajo460
94 Public Sub New()
95     Me.InitializeComponent()
96
97     Dim widthNominalAnillo As New Dato("widthNominalAnillo","Distance","Width nominal del anillo",Nothing,"width del anillo")
98     listaDatosRequeridos.Add(widthNominalAnillo)
99
100    Dim diametroNominalAnillo As New Dato("diametroNominalAnillo","Distance","Diametro nominal del anillo",Nothing,"Diametro del anillo")
101    listaDatosRequeridos.Add(diametroNominalAnillo)
102 End Sub

```

Figura 120 Variables que se agregan a la lista de datos requeridos.

El segundo paso es agregar las variables necesarias al método pasarValoresDeListas e igualarlas con los métodos obtenerValorDatoDouble u obtenerValorDatoString dependiendo del tipo de dato que se le asignó a la variable, estos métodos pertenecen al module1 del proyecto. Este paso se ve de la siguiente manera:

```

CentroTrabajo460.vb
ProyectoSa.CentroTrabajo460
89 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
90     width = Module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")
91     diametro = Module1.obtenerValorDatoDouble(lDatos,"diametroNominalAnillo")
92 End Sub

```

Figura 121 Pasando valores de listas.

Ahora se debe de crear un método en el cual se obtendrá el número de rieles a usar dependiendo del diámetro que el usuario ingrese dentro del rango establecido.

```

CentroTrabajo460.vb
ProyectoSa.CentroTrabajo460
149 Private Function sacar_num_rieles() As Integer
150     Dim r As Integer = 0
151     If diametro >= 0.7 And diametro <= 4.2 Then
152         r = 12
153     ElseIf diametro >= 4.3 And diametro <= 5.59
154         r = 5
155     ElseIf diametro >= 5.6 And diametro <= 6.93
156         r = 3
157     End If
158     Return r
159 End Function

```

Figura 122 Función sacar_num_rieles().

Después de haber realizado todo lo anterior, ahora se debe de crear el método calcular en el cual se encuentran todas las operaciones necesarias para llevar a cabo el cálculo del tiempo estándar. El método se muestra en la siguiente imagen.

```

CentroTrabajo460.vb
ProyectoSa.CentroTrabajo460
128 Public Sub calcular() Implements ICentroTrabajo.calcular
129     tiempo_septup = buscar_setupin(centro)
130     Dim rieles,carga As Integer
131     Dim ciclo_por_carga As Double
132     rieles = sacar_num_rieles()
133     carga = Cint((rieles * 20) / width)
134     ciclo_por_carga = (480.06 + 103.92)
135     tiempo_maquina = Math.Round((100 * (ciclo_por_carga / 3600) / carga)*100,3)
136     tiempo_mano = tiempo_maquina
137 End Sub

```

Figura 123 Método calcular donde se encuentra la fórmula del tiempo estándar.

Lo que primero que se realiza dentro de este es obtener el tiempo setup a través de otro método llamado buscar_setupin() que se encuentra en el module1 del proyecto, después se declaran una serie de variables en las cuales se almacenaran los valores obtenidos en las operaciones, luego se extrae el valor que se obtuvo de la función sacar_num_rieles() y se almacena en una variable. Y una vez que se tienen todos los valores necesarios se realiza el cálculo.

Posteriormente se tiene que agregar el evento clic al botón del formulario, y se le agrega el siguiente código:

```

CentroTrabajo460.vb
ProyectoSa.CentroTrabajo460
161 Sub ButtonClick(sender As Object, e As EventArgs)
162     If IsNumeric(txt_width.Text) And IsNumeric(txt_diametro.Text) Then
163         width = txt_width.Text
164         diametro = txt_diametro.Text
165         If diametro >= 0.7 And diametro <= 6.93 Then
166             calcular()
167             Close()
168         Else
169             msgbox("No existen rieles para ese valor de diametro",MsgBoxStyle.Exclamation,"Ing. de Rutas")
170         End If
171     Else
172         msgbox("Revisar los valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
173     End If
174 End Sub

```

Figura 124 Evento clic del botón del formulario.

Lo primero que se realiza en el código anterior es una condición if en la cual si los valores que ingresa el usuario son numéricos se almacenan dichos valores en las variables que se declararon al principio, sin embargo si el usuario ingresa valores no numéricos no realizara nada y se mostrara un mensaje de error, luego se realiza otra condición en donde si el valor del diámetro está dentro del rango establecido se realiza el cálculo y se cierra el formulario, de lo contrario se mostrara un mensaje de error.

6.2.16 CentroTrabajo491.vb – O.D. SHEFFIELD GASOLINE

Para este centro de trabajo se le tuvieron que agregar dos nuevos valores de entrada los cuales son Width nominal y Width de espaciadores, por esa razón se tuvo que cambiar la interfaz del centro de trabajo, la cual quedo de la siguiente forma.

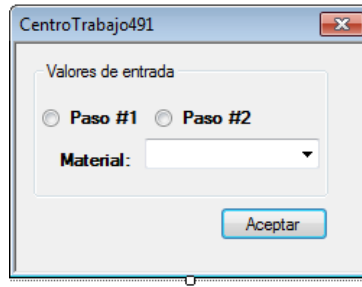


Figura 125 Formulario antes de la actualización.

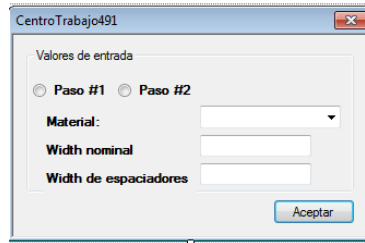


Figura 126 Formulario actualizado.

Una vez actualizada la interfaz gráfica se procede a declarar las variables correspondientes a los nuevos valores de entrada y son las siguientes.

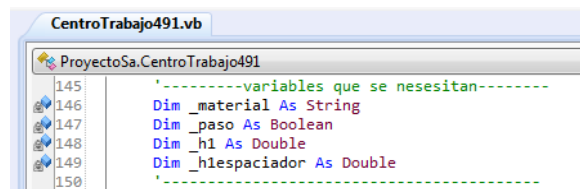


Figura 127 Variables donde se almacenan los valores de entrada.

Como se ha hecho en centros de trabajo anteriores se debe agregar nuevos parámetros al constructor de la clase, agregar variables a la lista de datos requeridos y pasar valores de listas.

Después de que se realizó todo lo anterior se debe actualizar el método calcular, donde se agregaron nuevas operaciones para poder realizar el cálculo, y quedo de la siguiente manera.

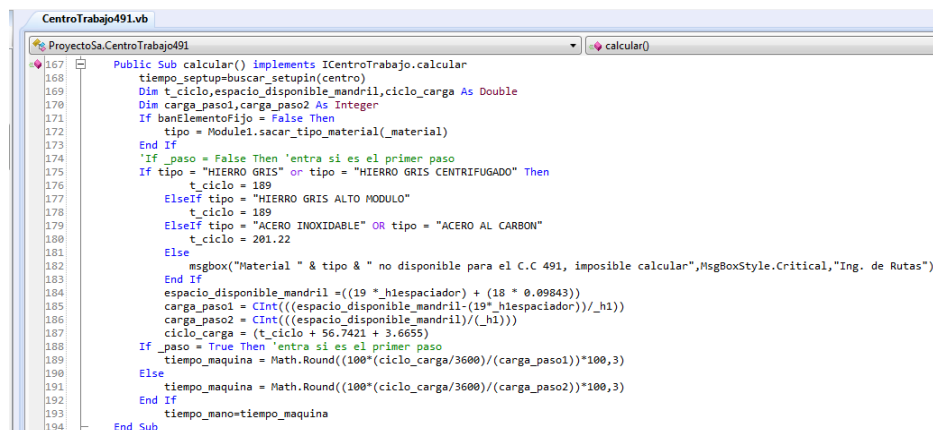
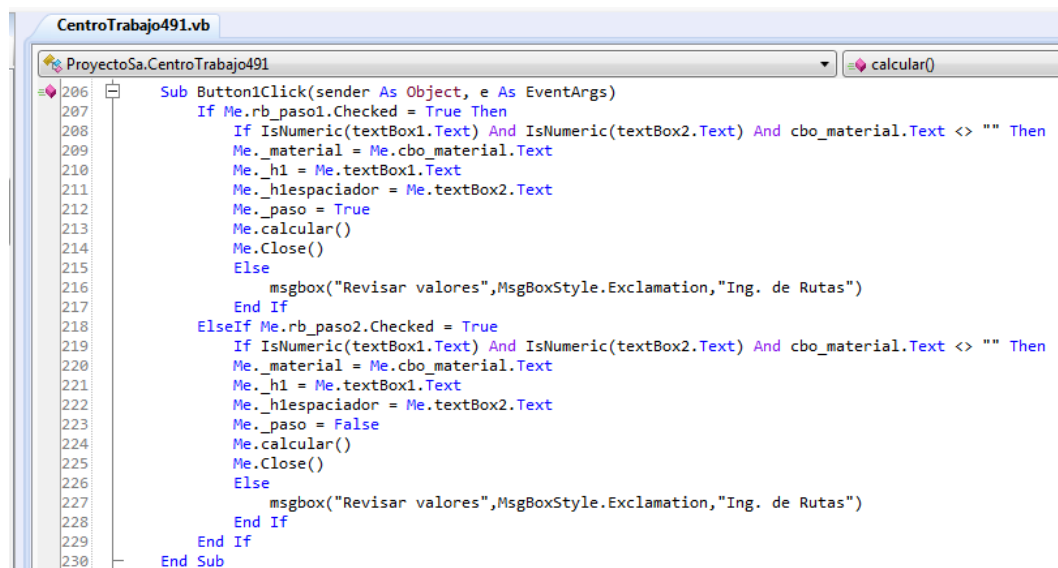


Figura 128 Método calcular() del CT 491.

Como se ha mencionado en centros de trabajo anteriores lo primero que se hace dentro de este método es obtener el tiempo setup, luego se declaran algunas variables donde se almacenan los valores obtenidos de las operaciones, después se realiza una condición en la cual el tiempo ciclo va a variar dependiendo del material que el usuario seleccione. Y luego se realizan las operaciones para obtener los valores necesarios para llevar a cabo el cálculo, una vez que realizo las operaciones se realiza otra condición if en donde si el usuario selecciona el paso 1 se realizara el cálculo con la fórmula del paso 1, sino se realizara el cálculo con la fórmula del paso 2.

Ahora solo resta agregar el siguiente código al evento de clic del botón del formulario tal como se muestra en la siguiente imagen:



```
206 Sub Button1Click(sender As Object, e As EventArgs)
207     If Me.rb_paso1.Checked = True Then
208         If IsNumeric(textBox1.Text) And IsNumeric(textBox2.Text) And cbo_material.Text <> "" Then
209             Me._material = Me.cbo_material.Text
210             Me._h1 = Me.textBox1.Text
211             Me._h1espaciador = Me.textBox2.Text
212             Me._paso = True
213             Me.calcular()
214             Me.Close()
215         Else
216             msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
217         End If
218     ElseIf Me.rb_paso2.Checked = True
219         If IsNumeric(textBox1.Text) And IsNumeric(textBox2.Text) And cbo_material.Text <> "" Then
220             Me._material = Me.cbo_material.Text
221             Me._h1 = Me.textBox1.Text
222             Me._h1espaciador = Me.textBox2.Text
223             Me._paso = False
224             Me.calcular()
225             Me.Close()
226         Else
227             msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
228         End If
229     End If
230 End Sub
```

Figura 129 Evento clic del botón del formulario.

Lo que realiza el código anterior es que primero se hace condición if en la que si el usuario selecciona el paso 1 se va a realizar otra condición en la que si el usuario ingresa valores numéricos y selecciona un material el formulario va a proceder a almacenar los variables de entrada en las variables que se declararon al principio y luego la variable _paso va a pasar a ser true, y luego realizara el cálculo y cerrara el formulario, sino se ingresan valores numéricos y no se selecciona ningún material entonces aparecerá un mensaje de error y no se realizara. Sino si el usuario selecciona el paso 2 realizo lo mismo que cuando se selecciona el paso 1 con la diferencia de que la variable _paso pasa a ser false. Y ya con esto queda actualizado este centro de trabajo.

6.2.17 CentroTrabajo496.vb – STRESS RELIEF FOR STEEL COMP. RINGS

A centro de trabajo se le agrego un valor de entrada a la interfaz gráfica el cual es diámetro nominal. Este es el formulario:

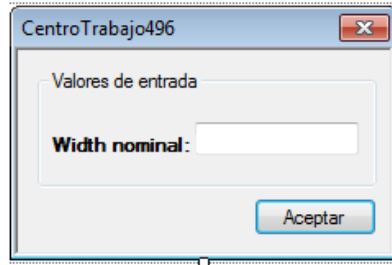


Figura 130 Formulario antes de la actualización.

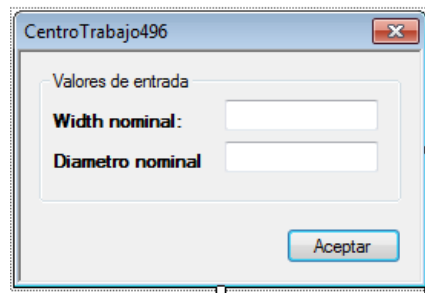


Figura 131 Formulario actualizado.

Una vez que se modificó la interfaz gráfica se procede a declarar las variables correspondientes a los nuevos valores de entrada y es la siguiente:

```
CentroTrabajo496.vb
ProyectoSa.CentroTrabajo496
108 '-----variables que se necesitan-----
109 'Friend Shadows width As Double
110 Dim _h1 As Double
111 Dim _d1 As Double
112 '-----
```

Figura 132 Variables donde se almacenan los valores de entrada.

Como se ha hecho en centros de trabajo anteriores se debe agregar nuevos parámetros al constructor de la clase, agregar variables a la lista de datos requeridos y pasar valores de listas.

Después se debe de crear un método en el cual se obtiene el tamaño de carga por pino dependiendo del valor de diámetro que el usuario ingrese y este dentro del rango. El código es el siguiente:

```
CentroTrabajo496.vb
ProyectoSa.CentroTrabajo496 tiempoSet
146 Private Sub sacar_tamaño_de_carga_por_pino(ByVal diametro As Double)
147     If diametro <= 4.3307 Then
148         tam_carga_pino = 12.1
149     Else
150         tam_carga_pino = 11.92
151     End If
152 End Sub
```

Figura 133 Método sacar_tamaño_de_carga_por_pino().

Después se debe de modificar el método calcular() del centro de trabajo en donde se encuentran las operaciones necesarias para poder realizar el cálculo del tiempo estándar. A continuación se muestra el código completo.

```

128 | Public Sub calcular() implements ICentroTrabajo.calcular
129 |     tiempo_septup=buscar_setupin(centro)
130 |     sacar_tamaño_de_carga_por_pino(_d1)
131 |     Dim cant_pinos_por_nivel,cant_pinos_totales As Integer
132 |     If tam_carga_pino = 11.92 Then
133 |         cant_pinos_por_nivel = 35
134 |     Else
135 |         cant_pinos_por_nivel = 69
136 |     End If
137 |     cant_pinos_totales = (cant_pinos_por_nivel * 2)
138 |     tiempo_maquina = Math.Round(((35026 * _h1) / ((tam_carga_pino * cant_pinos_totales) * 36))*100,3)
139 |     tiempo_mano= Math.Round(tiempo_maquina * factorLabor,3)
140 | End Sub

```

Figura 134 Método calcular del CT 496.

Lo primero que se realiza en este método es obtener el tiempo setup de la base de datos mediante el método buscar_setupin() del module1 del proyecto, después se obtiene el valor del tamaño de carga por pino mediante el método sacar_tamaño_de_carga_por_pino() que se creó en esta clase. Luego se declaran algunas variables donde se almacenaran los valores obtenidos de las operaciones. A continuación se crea una condición if en la que se obtiene el valor de la variable cant_pinos_por_nivel, después se realiza una operación donde se obtiene el valor de la variable cant_pinos_totales. Una vez que se tienen todos los valores, finalmente se realiza el cálculo del tiempo estándar.

Finalmente se agrega el evento clic al botón del formulario y se le agrega el siguiente código.

```

164 | Sub Button1Click(sender As Object, e As EventArgs)
165 |     If isnumeric(Me.textBox1.Text) Then
166 |         Me._h1 = CDec(Me.textBox1.Text)
167 |         me._d1 = CDec(Me.textBox2.Text)
168 |         Me.calcular()
169 |         Me.Close()
170 |     Else
171 |         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
172 |     End If
173 | End Sub

```

Figura 135 Evento clic del botón del formulario

La función que realiza el código anterior es que si el usuario ingresa valores numéricos al formulario se realiza el cálculo y se cierra el formulario. De lo contrario si el usuario ingresa valores no numéricos no se realizara el cálculo y aparecerá un error.

6.2.18 CentroTrabajo715.vb – O.D. CHROME PLATE RINGS

A este centro de trabajo se le agregaron nuevos valores entrada pero estos no afectan el cálculo del tiempo estándar, entonces los valores de entrada que ya existían antes de que se modificara la interfaz gráfica son suficientes para poder llevar a cabo el cálculo del tiempo estándar.

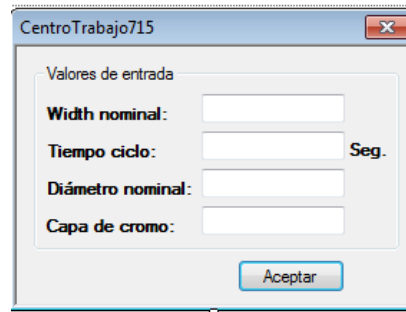


Figura 136 Formulario antes de la actualización.

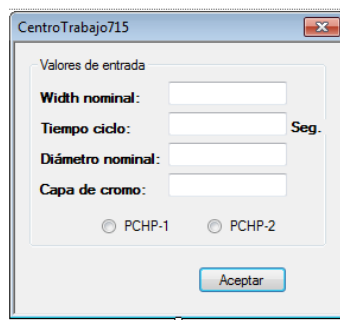


Figura 137 Formulario actualizado.

Cabe mencionar que este centro de trabajo existen dos métodos calcular uno es cuando se selecciona el PCHP-1 y el otro es cuando se selecciona el PCHP-2. Sin embargo la fórmula es la misma para los dos métodos, así que solo se cambió la nueva por la que ya se encontraba obsoleta. A continuación se muestra el código completo.

```
CentroTrabajo715.vb
ProyectoSa.CentroTrabajo715
Public Sub calcular() Implements ICentroTrabajo.calcular
136 Dim ace, unichr, acsul, acsulfl As Double
137 tiempo_septup = buscar_setupin(centro)
138 'actualizado el dia 11 de Diciembre de 2013, correo enviado por J.L.Y ese mismo dia.
139 tiempo_maquina = math.Round((_h1*(t_ciclo+107.72))/4809.6,3)*100
140 tiempo_mano = tiempo_maquina
141
```

Figura 138 Formula del tiempo estándar del método calcular() (es la misma para el método calcular2()).

Una vez que se hizo esto queda actualizado este centro de trabajo.

6.2.19 CentroTrabajo750.vb – PHOSPHATE GASOLINE

A este centro de trabajo no hubo la necesidad de cambiar su interfaz gráfica ya que con los valores de entrada que tiene son suficientes para realizar el cálculo del tiempo estándar. A continuación se muestra el formulario.

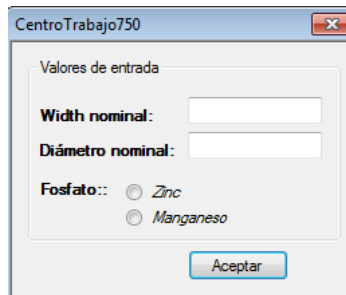


Figura 139 Formulario del centro de trabajo 750.

Ahora se debe de crear un método en cual se va a obtener el valor de los bastidores a usar, dependiendo del valor del diámetro que el usuario inserte y este dentro del rango establecido.

```
CentroTrabajo750.vb
ProyectoSa.CentroTrabajo750
167 Private Sub cuantos_bastidores(ByVal diametro1 As Double)
168     If diametro1>6.26 Then
169         factorbastidores = 0
170         Exit Sub
171     Else If diametro1>=5.71 Then
172         factorbastidores = 5
173         Exit Sub
174     Else If diametro1>=4.6 Then
175         factorbastidores = 9
176         Exit Sub
177     Else If diametro1>=4.1 Then
178         factorbastidores = 11
179         Exit Sub
180     ElseIf diametro1>=1.75
181         factorbastidores = 15
182     ElseIf diametro1>=0.8268
183         factorbastidores = 29
184     Else
185         factorbastidores = 0
186         Exit Sub
187     End If
188 End Sub
```

Figura 140 Método cuantos_bastidores().

Después de crear el método ahora si se procede a modificar el método calcular. A continuación se muestra el código de dicho método.

```

CentroTrabajo750.vb
ProyectoSa.CentroTrabajo750
tiem
141 Public Sub calcular() implements ICentroTrabajo.calcular
142 tiempo_septup=buscar_setupin(centro)
143 cuantos_bastidores(diametro)
144 Dim t_ciclo,ciclo_por_carga As Double
145 Dim carga As Integer
146 If fosfato = True Then
147     t_ciclo = 990.92
148 Else
149     t_ciclo = 990.92
150 End If
151 carga = CInt(((factorbastidores * 20) / _h1))
152 ciclo_por_carga = (t_ciclo + 181.35 + 92.56)
153 tiempo_maquina = Math.Round((100 * (ciclo_por_carga/3600) / carga)*100,3)
154 tiempo_mano = tiempo_maquina
155 End Sub

```

Figura 141 Método calcular del CT 750.

Lo primero que se hace siempre en el método calcular es obtener el tiempo setup de la base de datos, después se obtiene el número de bastidores a usar mediante el método cuantos_bastidores() que mencionamos como se creó. Luego se hace una condición if en donde el valor del tiempo ciclo es igual si variable fosfato es true o false. Y ya por último se realizan las operaciones donde se realiza el cálculo del tiempo estándar.

La última actividad a realizar es agregar una condición if al evento clic del botón del formulario, en donde si el valor que ingreso el usuario está dentro del rango establecido se realizara el cálculo y se cerrara el formulario. De lo contrario no procederá el cálculo y se mostrara un mensaje de error.

```

CentroTrabajo750.vb
ProyectoSa.CentroTrabajo750
tiempo_septup
190 Sub Button1Click(sender As Object, e As EventArgs)
191 If IsNumeric(textBox1.Text) And IsNumeric(textBox2.Text) And radioButton1.Checked Or radioButton2.Checked Then
192     Me._h1 = Me.textBox1.Text
193     Me.diametro = Me.textBox2.Text
194     If Me.radioButton1.Checked Then
195         Me.fosfato = True
196     ElseIf Me.radioButton2.Checked
197         Me.fosfato = False
198     End If
199     If diametro >=0.8268 And diametro <= 6.26 Then
200         calcular()
201         Close()
202     Else
203         msgbox("No existen bastidores para ese valor de diametro",MsgBoxStyle.Exclamation,"Ing. de Rutas")
204         textBox2.Text = ""
205         textBox2.Focus
206     End If
207 Else
208     msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
209 End If
210 End Sub

```

Figura 142 Evento clic del botón del formulario.

6.2.20 CentroTrabajo755.vb – FERROX FILL GASOLINE

A este centro no se le hizo ningún cambio a su interfaz gráfica ya que con los valores de entrada con los que cuenta son necesarios para poder realizar el cálculo. El formulario es el siguiente:

Figura 143 Formulario del CT 755.

Tampoco no se hizo ninguna modificación al método calcular ni a la fórmula que obtiene el tiempo estándar. El método es el siguiente:

```

CentroTrabajo755.vb
ProyectoSa.CentroTrabajo755
127 Public Sub calcular() implements ICentroTrabajo.calcular
128     tiempo_septup=buscar_setupin(centro)
129     tiempo_maquina = math.Round((( ( 6186.2 + ( ( _diametro * 403.96 ) / 4.75 ) )*_width ) / 524.16) * 100,3)
130     tiempo_mano=tiempo_maquina * factorLabor
131 End Sub

```

Figura 144 Método calcular() del CT 755.

Entonces a este centro de trabajo no se le hizo ninguna modificación y quedo de igual forma.

6.2.21 CentroTrabajo757.vb – HEAD TREAT GASOLINE

A este centro de trabajo no se le modifíco su interfaz gráfica ya que con los valores de entrada con los que cuenta son suficientes para realizar el cálculo del tiempo estándar. El formulario es el siguiente:

Figura 145 Formulario del CT 757.

Lo único que se modificó en este centro de trabajo fue el método calcular(). El código quedo de la siguiente manera.

```

CentroTrabajo757.vb
ProyectoSa.CentroTrabajo757
Public Sub calcular() implements ICentroTrabajo.calcular
158 tiempo_setup=buscar_setupin(centro)
159 If banElementoFijo = False Then
160 Tipo = Module1.sacar_tipo_material(material)
161 End If
162 Dim t_ciclo_prep, t_compesacion, t_ciclo_enfra As Double
163 If Tipo="HIERRO GRIS ALTO MODULO" Then
164 t_ciclo_prep = 4380
165 t_compesacion = 0
166 t_ciclo_enfra = 8392
167 ElseIf Tipo = "HIERRO GRIS" Or Tipo = "HIERRO GRIS CENTRIFUGADO"
168 t_ciclo_prep = 2400
169 t_compesacion = 900
170 t_ciclo_enfra = 4380
171 ElseIf Tipo = "HIERRO GRIS INTERMEDIO"
172 t_ciclo_prep = 2700
173 t_compesacion = 900
174 t_ciclo_enfra =4380
175 ElseIf Tipo = "HIERRO DUCTIL"
176 t_ciclo_prep = 0
177 t_compesacion = 0
178 t_ciclo_enfra =900
179 Else
180 msgbox("Material " & Tipo & " no disponible para el C.C 757, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
181 Exit Sub
182 End If
183 tiempo_maquina = Math.Round(((t_ciclo_prep + t_compesacion + tpo_horneado + t_ciclo_enfra) +1007.88)/(11088)*100,3)
184 tiempo_mano = tiempo_maquina
185 End Sub

```

Figura 146 Método calcular() del CT 757.

Lo que realiza el código anterior es que los valores de los tiempos van a variar dependiendo del material que se vaya a usar. Y una vez que se obtienen los valores se procede a realizar el cálculo del tiempo estándar.

Y con esta modificación a este método quedo actualizado este centro de trabajo.

6.3 Actualización de tiempos estándar – Large Bore

6.3.1 CentroTrabajo9226.vb – I.D. GROOVE L.B.

Para este centro de trabajo se le tuvo que crear una interfaz gráfica ya que anteriormente realizaba el cálculo de forma directa, sin embargo ahora es necesario que el usuario ingrese varios valores de entrada. Entonces el formulario es el siguiente:

Figura 147 Formulario del CT 9226.

Una vez que se la interfaz de este centro de trabajo se procedió a almacenar los valores de entrada en sus respectivas variables y son las siguientes:

```

136 '-----variables que se nesesitan-----
137 Dim material19226 As String
138 Dim h1 As Double
139

```

Figura 148 Variables necesarias para la clase.

Ahora que se declararon las variables se deben de agregar un número igual de parámetros a las variables declaradas como se muestra en la siguiente imagen:

```

131 Public Sub New(ByVal _material As String, ByVal H1 As Double)
132     material19226 = _material
133     h1 = H1
134 End Sub

```

Figura 149 Agregando nuevos parámetros al constructor de la clase.

Como se puede ver en la imagen anterior se tiene que igualar las variables con los parámetros de la clase. Después de haber realizado la igualdad se procede a agregar los valores de entrada a su respectiva lista como se muestra a continuación.

```

122 Public Sub New()
123     Me.InitializeComponent()
124     Dim material_ As New DatoString("MATERIAL", "Especificación de materia prima (MF012-S,SPR-128,MF025-L,etc...)", "Material:")
125     listaDatosStringRequeridos.Add(material_)
126
127     Dim width_ As New Dato("width_", "Distance", "Width nominal del anillo", Nothing, "width del anillo")
128     listaDatosRequeridos.Add(width_)
129 End Sub

```

Figura 150 Pasando valores de entrada a las listas de datos requeridos.

Para el caso del material como su valor de entrada es una cadena de texto se debe de agregar a la listaDatosStringRequeridos, para el caso de los valores numéricos se deben de agregar a la listaDatosRequeridos.

Una vez que se realizó lo anterior ahora se deben modificar el método pasarValoresDeListas con los valores de entrada que se agregaron a este centro de trabajo, esto se realiza de la siguiente forma.

```

115 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
116     If banElementoFijo = False Then
117         material19226 = Module1.obtenerValorDatoString(lDatosString, "MATERIAL")
118     End If
119     h1 = Module1.obtenerValorDatoDouble(lDatos, "width_")
120 End Sub

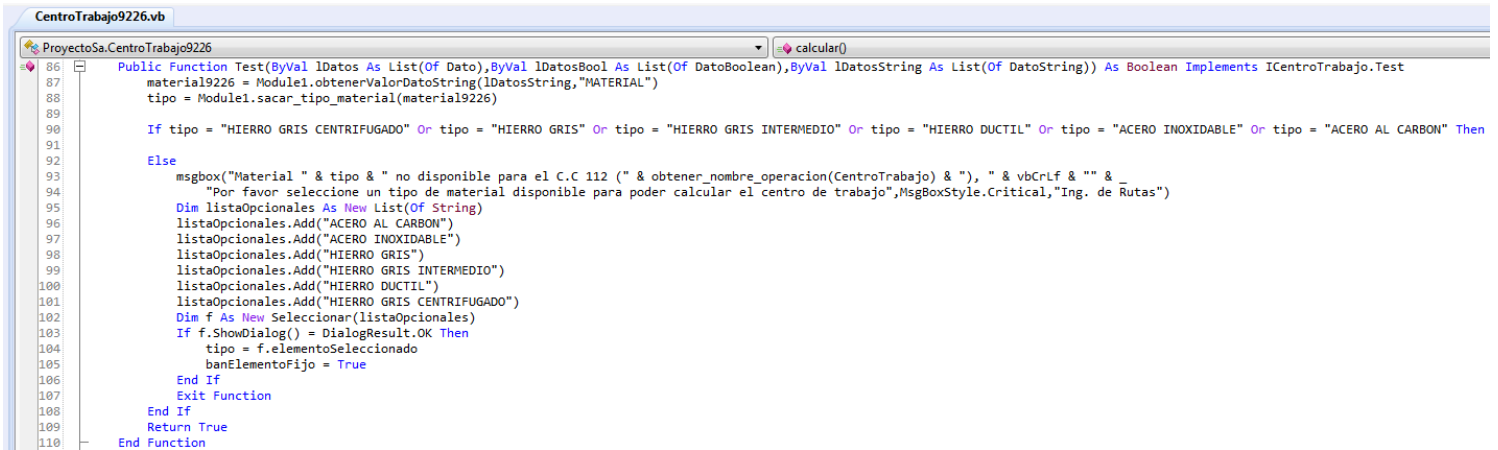
```

Figura 151 Pasando valores de listas.

Para poder pasar el valor del material primero se debe de realizar una condición en la cual si el valor de la variable banElementoFijo que es de tipo boolean es false realiza el procedimiento, y para el caso del valor double no se realiza ninguna condición.

NOTA: todos los pasos anteriores se realizan cada vez que se agrega un nuevo valor de entrada al centro de trabajo.

Ahora se debe de agregar las siguientes líneas de código al método Test de la clase.

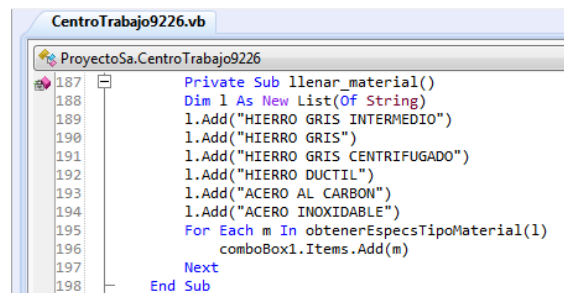


```
86 Public Function Test(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
87     material9226 = Module1.obtenerValorDatoString(lDatosString,"MATERIAL")
88     tipo = Module1.sacar_tipo_material(material9226)
89
90     If tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS INTERMEDIO" Or tipo = "HIERRO DUCTIL" Or tipo = "ACERO INOXIDABLE" Or tipo = "ACERO AL CARBON" Then
91
92     Else
93         MsgBox("Material " & tipo & " no disponible para el C.C 112 (" & 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("ACERO AL CARBON")
97         listaOpcionales.Add("ACERO INOXIDABLE")
98         listaOpcionales.Add("HIERRO GRIS")
99         listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
100        listaOpcionales.Add("HIERRO DUCTIL")
101        listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
102        Dim f As New Seleccionar(listaOpcionales)
103        If f.ShowDialog() = DialogResult.OK Then
104            tipo = f.elementoSeleccionado
105            banElementoFijo = True
106        End If
107        Exit Function
108    End If
109    Return True
110 End Function
```

Figura 152 Agregando lista de materiales al método Test de la clase.

Este paso se realiza siempre que se agrega el valor material al centro de trabajo.

Paso siguiente es crear un método en la clase del centro de trabajo en el cual se obtiene la lista de los materiales de la base de datos del proyecto, el código es el siguiente.



```
187 Private Sub llenar_material()
188     Dim l As New List(Of String)
189     l.Add("HIERRO GRIS INTERMEDIO")
190     l.Add("HIERRO GRIS")
191     l.Add("HIERRO GRIS CENTRIFUGADO")
192     l.Add("HIERRO DUCTIL")
193     l.Add("ACERO AL CARBON")
194     l.Add("ACERO INOXIDABLE")
195     For Each m In obtenerEspecesTipoMaterial(1)
196         comboBox1.Items.Add(m)
197     Next
198 End Sub
```

Figura 153 Método llenar_material().

Ahora que ya se tienen todos los elementos anteriores se procede a modificar el método calcular() de la clase como se muestra a continuación.


```

155 Public Sub calcular() implements ICentroTrabajo.calcular
156 tiempo_septup=buscar_setupin(centro)
157 If banElementoFijo = False Then
158     tipo = Module1.sacar_tipo_material(material9226)
159 End If
160 Dim t_ciclo As Double
161 If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
162     t_ciclo = 199.7
163 ElseIf tipo = "ACERO AL CARBON" Or tipo = "ACERO INOXIDABLE"
164     t_ciclo = 331.6
165 ElseIf tipo = "HIERRO DUCTIL"
166     t_ciclo = 240.3
167 ElseIf tipo = "HIERRO GRIS INTERMEDIO"
168     t_ciclo = 122.8
169 Else
170     msgbox("Material " & tipo & " no disponible para el C.C 9226, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
171 Exit Sub
172 End If
173 tiempo_maquina = Math.Round((((t_ciclo + 103.95) * (h1)) / 101.7792)*100,3)
174 tiempo_mano=tiempo_maquina
175 End Sub

```

Figura 154 Método calcular del CT 9226.

Como se puede apreciar en la imagen anterior lo primero que se realiza es una condición if en donde el valor del tiempo ciclo va a variar dependiendo del material que el usuario seleccione. Una vez obtenido tiempo ciclo se realiza el cálculo del tiempo estándar mediante la variable tiempo_maquina, ya que esta variable contiene la formula con la cual se obtiene el tiempo estándar.

Ahora se debe de crear el evento clic al botón del formulario como se muestra a continuación.

```

199 Sub Button1Click(sender As Object, e As EventArgs)
200 If IsNumeric(textBox1.Text) And comboBox1.Text <> "" Then
201     material9226 = comboBox1.Text
202     h1 = textBox1.Text
203     calcular()
204     Close()
205 Else
206     MsgBox("Revisar valores",MsgBoxStyle.Critical,"Ing. de Rutas")
207 End If
208 End Sub
209

```

Figura 155 Evento clic del botón del formulario.

Como se puede apreciar en la imagen anterior primero se realiza una condición if en la que si el valor que ingresa el usuario es numérico, además que si el usuario selecciono un material del ComboBox, se realiza el almacenamiento de los valores que el usuario ingreso a las variables y una vez que se tienen los valores se procede a realiza el cálculo del tiempo estándar mediante el método calcular() y ya por último se cierra el formulario y listo se obtendrá el valor del tiempo estándar.

6.3.2 CentroTrabajo9227.vb – GROOVE FERROX L.B.

Para este centro de trabajo no hubo la necesidad de cambiar su interfaz gráfica ya que con los elementos que contiene son suficientes y necesarios para llevar a cabo el cálculo del tiempo estándar. A continuación se muestra el formulario.

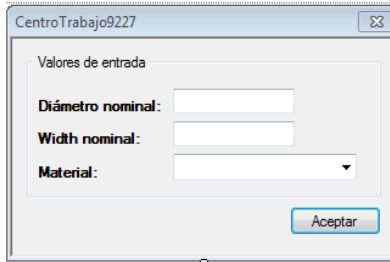


Figura 156 Formulario del CT 9227.

Como a este centro de trabajo no se le realizó ninguna modificación a su interfaz gráfica, entonces solo se modificó el método calcular() el cual contiene la fórmula para calcular el tiempo estándar. El método es el siguiente.

```

CentroTrabajo9227.vb*
ProyectoSa.CentroTrabajo9227
calculador()
159 Public Sub calcular() implements ICentroTrabajo.calcular
160 tiempo_septup=buscar_setupin(centro)
161 Dim t_ciclo,tam_carga_Pulg As Double
162 If banElementoFijo = False Then
163 tipo = Module1.sacar_tipo_material(_material)
164 End If
165 If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
166 tam_carga_Pulg = 2.805
167 t_ciclo = Math.Round(((995.17 * _diametro)/8.125),3)
168 ElseIf tipo = "HIERRO GRIS INTERMEDIO"
169 tam_carga_Pulg = 2.985
170 t_ciclo = Math.Round(((463.79 * _diametro)/8.5),3)
171 ElseIf tipo = "HIERRO DUCTIL"
172 tam_carga_Pulg = 2.815
173 t_ciclo = Math.Round(((1889.11 * _diametro)/9.062),3)
174 Else
175 msgbox("Material " & tipo & " no disponible para el C.C 9227, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
176 Exit Sub
177 End If
178 Dim tam_de_carga As Integer
179 Dim ciclo_por_carga As Double
180 tam_de_carga = CInt(tam_carga_Pulg/h1)
181 ciclo_por_carga = (t_ciclo + 129.30)
182 tiempo_maquina = Math.Round((100 * (ciclo_por_carga/3600) / tam_de_carga) * 100,3)
183 tiempo_mano=tiempo_maquina
184 End Sub

```

Figura 157 Método calcular() del CT 9227.

El cambio que se realizó fue a la condición if donde anteriormente solo se obtenía el valor que se le asignaba al tiempo ciclo, sin embargo con la modificación lo que se obtienen son dos valores los cuales varían dependiendo del material que el usuario seleccione, cabe mencionar que ahora el valor del tiempo ciclo se obtiene de una operación matemática, ya que anteriormente el valor del tiempo ciclo ya está asignado. También se agregaron dos variables más las cuales obtienen su valor dependiendo de los valores que arroja la condición y pues ya por último se modificó la variable tiempo_maquina la cual contiene la fórmula para obtener el valor del tiempo estándar.

6.3.3 CentroTrabajo9243.vb – LAP. O.D. L.B.

A este centro de trabajo se le tuvo que modificar su interfaz gráfica ya que los elementos con los que contaba la anterior interfaz no eran los suficientes para

poder llevar a cabo el cálculo del tiempo estándar. Entonces se muestra el antes y el después del formulario.

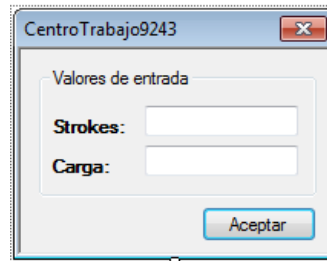


Figura 158 Formulario antes de la modificación.

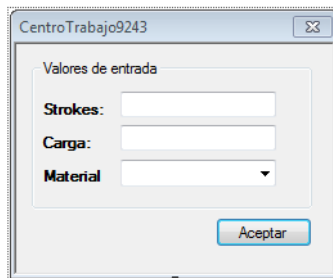


Figura 159 Formulario después de la actualización.

Una vez que se realizó el cambio de la interfaz gráfica se procede a declarar la variable correspondiente al nuevo valor de entrada.

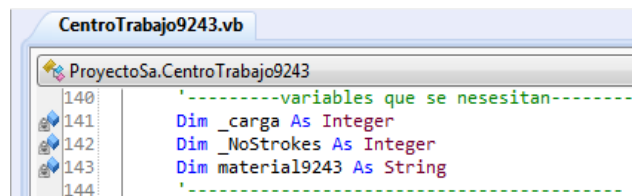


Figura 160 Variables correspondientes a los valores de entrada.

Como se agregó un nuevo valor de entrada se tiene que agregar un nuevo parámetro al constructor, después se tienen que pasar los valores de entrada a la lista de datos requeridos y ya por último pasar los valores de listas, como se explicó en el CentroTrabajo9226.vb – I.D. GROOVE L.B.

También como se agregó el valor de entrada material se debe de agregar la lista de materiales al método Test de la clase, además también agregar el método llenar_material(), con los materiales que se van a usar en este centro de trabajo. Todo esto también se explica en el CentroTrabajo9226.vb – I.D. GROOVE L.B.

Después de que se haya realizado todo lo anterior solo resta modificar el método calcular con las operaciones necesarias para poder llevar a cabo el cálculo del tiempo estándar. A continuación se muestra el código de dicho método

```

160 Public Sub calcular() implements ICentroTrabajo.calcular
161     tiempo_septup=buscar_setupin(centro)
162     If banElementoFijo = False Then
163         tipo = Module1.sacar_tipo_material(material9243)
164     End If
165     Dim t_ciclo As Double
166     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
167         t_ciclo = 0.961666667
168     ElseIf tipo = "ACERO AL CARBON" Or tipo = "ACERO INOXIDABLE"
169         t_ciclo = 0.778285154
170     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
171         t_ciclo = 0.703718182
172     ElseIf tipo = "HIERRO DUCTIL"
173         t_ciclo = 0.703083929
174     Else
175         msgbox("Material " & tipo & " no disponible para el C.C 9243, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
176     End If
177     tiempo_maquina = Math.Round(((80.07 + (_NoStrokes * t_ciclo)) / (_carga * 36)) * 100,3)
178     tiempo_mano = math.Round(tiempo_maquina * .166,3)
179 End Sub

```

Figura 161 Método calcular correspondiente al CT 9243.

En este método el único cambio que se realizó fue el de crear una condición if en la que el valor del tiempo ciclo va a variar dependiendo del material que el usuario seleccione. Y una vez que se obtiene el valor del tiempo ciclo solo resta cambiar la operación que almacena la variable tiempo_maquina, dicha operación es la fórmula para obtener el tiempo estándar.

6.3.4 CentroTrabajo9244.vb – DEGREASE RINGS

Para este centro de trabajo no fue necesario cambiar su interfaz gráfica, así que solo se modificó su código, específicamente el método calcular(), que es donde se encuentra la fórmula para obtener el tiempo estándar. El código del método es el siguiente.

```

121 Public Sub calcular() implements ICentroTrabajo.calcular
122     Try
123         tiempo_septup=buscar_setupin(centro)
124         tiempo_maquina = math.Round(((752.11 * _h1) / (1836))*100,3)
125         tiempo_mano=tiempo_maquina
126     Catch er As Exception
127         Me.ListaAlertas.Add(Name & ": " & er.Message)
128     End Try
129 End Sub

```

Figura 162 Método calcular() del CT 9244.

En el código anterior lo único que se hizo fue cambiar la fórmula obsoleta por la fórmula actualizada, dicha fórmula está almacenada en la variable tiempo_maquina. Y con ese simple cambio se actualizó este centro de trabajo.

6.3.5 CentroTrabajo9261.vb – CROP SAW L.B.

Al igual que en el centro de trabajo anterior no se modificó la interfaz gráfica de este centro de trabajo, quedando el formulario de la siguiente manera:

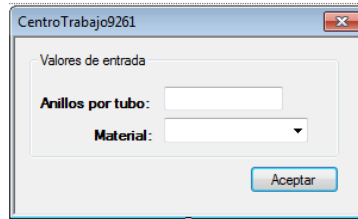


Figura 163 Formulario del CT 9261.

Así que los únicos cambios se realizaron dentro del método calcular(). A continuación se muestra todo el código de dicho método.

```
CentroTrabajo9261.vb
ProyectoSa.CentroTrabajo9261
153 Public Sub calcular() Implements ICentroTrabajo.calcular
154     tiempo_septup=buscar_setupin(centro)
155     If banElementoFijo = False Then
156         tipo = Module1.sacar_tipo_material(_material)
157     End If
158     Dim cantidad_cuff As Integer
159     Dim t_ciclo_cuff,t_ciclo_total_cuff,act_ciclicas As Double
160     If tipo="HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
161         cantidad_cuff = 3
162         t_ciclo_cuff = 341.8
163         t_ciclo_total_cuff = 1367.2
164         act_ciclicas = 277.267067
165     ElseIf tipo="HIERRO DUCTIL"
166         cantidad_cuff = 4
167         t_ciclo_cuff = 354.45
168         t_ciclo_total_cuff = 1772.25
169         act_ciclicas = 288.8004
170     Else
171         msgbox("El material " & tipo & " no esta disponible para el calculo,varificar con Ing. Industrial",MsgBoxStyle.Exclamation)
172         me.ListaAlertas.Add(Name & ": " & "El material " & tipo & " no esta disponible para el calculo,varificar con Ing. Industrial")
173     End If
174     Dim ciclo_por_carga As Double
175     ciclo_por_carga = (t_ciclo_total_cuff + act_ciclicas + 295.2992)
176     tiempo_maquina = Math.Round((100*(ciclo_por_carga/3600))/(cantidad_cuff*_anillosTubo))*100,3)
177     tiempo_mano=tiempo_maquina + factorLabor
178 End Sub
```

Figura 164 Método calcular() del CT 9261.

Como se puede ver en la imagen anterior a la condición if se le agregaron más variables, ya que anteriormente solo se obtenía el valor de la variable cantidad_cuff, sin embargo ahora con la actualización se tienen que obtener tres valores los cuales son necesarios para realizar los cálculos, lo que no cambio fueron los materiales que se usan en este centro de trabajo.

6.3.6 CentroTrabajo9262.vb – CNC CUFF TO RING & INS L.B.

Para este centro de trabajo se puede decir que se divide en tres operaciones, entonces para ello se tuvo que modificar la interfaz gráfica de tal manera que se pueda realizar el cálculo de las tres operaciones en un mismo formulario, a continuación se muestra la interfaz.

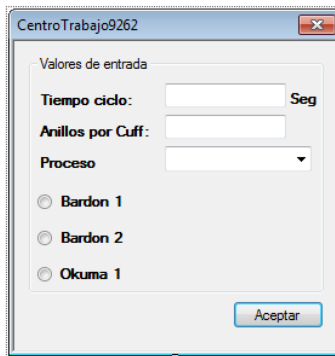


Figura 165 Formulario del CT 9262.

Operación Bardon 1

Para esta operación solo son necesarios los valores de entrada tiempo ciclo y anillos por cuff, al momento que se selecciona esta operación se desactiva el ComboBox de proceso ya que este valor de entrada no es necesario para realizar el cálculo del tiempo estándar de esta operación. En la siguiente imagen se muestra como se ve el formulario al seleccionar esta operación.

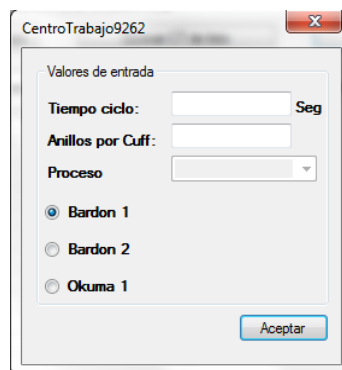


Figura 166 Interfaz del formulario al seleccionar la operación Bardon 1 y Okuma 1.

Para desactivar el ComboBox al seleccionar la operación Bardon 1 u Okuma 2 se les agrego el evento CheckedChange a los radiobuttons correspondientes a las operaciones mencionadas y se les agrego el siguiente código.

```

CentroTrabajo9262.vb
ProyectoSa.CentroTrabajo9262
228 Sub Bardon1CheckedChanged(sender As Object, e As EventArgs)
229     cbo_proceso.Enabled = False
230 End Sub
231
232 Sub Bardon2CheckedChanged(sender As Object, e As EventArgs)
233     cbo_proceso.Enabled = True
234 End Sub
235
236 Sub RadioButton1CheckedChanged(sender As Object, e As EventArgs)
237     cbo_proceso.Enabled = False
238 End Sub

```

Figura 167 Evento CheckedChanged de los radiobuttons del formulario.

Después de agregar los eventos a los radiobuttons, se declararon dos variables de tipo booleano como a continuación se muestra.

```

111 '-----variables que se necesitan-----
112 Dim _tc As Double
113 Dim _pzas_cuff As Double
114 Dim procesob2 As String
115 Dim bardon As Boolean
116 Dim okuma As Boolean
117

```

Figura 168 Variables necesarias para almacenar los valores de entrada.

A las variables mencionadas se le va a asignar su valor al momento que se seleccione una operación y se le dé clic al botón del formulario. A continuación se muestra el código de evento clic del botón del formulario.

```

193 Sub ButtonClick(sender As Object, e As EventArgs)
194     If bardon1.Checked = True Then
195         If IsNumeric(Me.textBox1.Text) And IsNumeric(Me.textBox2.Text) Then
196             Me._tc=Me.textBox1.Text
197             Me._pzas_cuff=Me.textBox2.Text
198             Me.bardon = True
199             Me.calcular()
200             Me.Close()
201         Else
202             MsgBox("Revisar valores", MsgBoxStyle.Exclamation,"Ing. de Rutas")
203         End If
204     ElseIf bardon2.Checked = True
205         If IsNumeric(Me.textBox1.Text) And IsNumeric(Me.textBox2.Text) And cbo_proceso.Text <> "" Then
206             Me._tc=Me.textBox1.Text
207             Me._pzas_cuff=Me.textBox2.Text
208             Me.procesob2 = Me.cbo_proceso.Text
209             Me.bardon = False
210             Me.calcular()
211             Me.Close()
212         Else
213             MsgBox("Revisar valores", MsgBoxStyle.Exclamation,"Ing. de Rutas")
214         End If
215     ElseIf okuma1.Checked = True
216         If IsNumeric(Me.textBox1.Text) And IsNumeric(Me.textBox2.Text) Then
217             Me._tc=Me.textBox1.Text
218             Me._pzas_cuff=Me.textBox2.Text
219             Me.okuma = True
220             Me.calcular()
221             Me.Close()
222         Else
223             MsgBox("Revisar valores", MsgBoxStyle.Exclamation,"Ing. de Rutas")
224         End If
225     End If
226 End Sub

```

Figura 169 Código del evento clic del botón del formulario.

Como se puede apreciar en la imagen anterior dependiendo de la operación que se seleccione se van a almacenar en las variables únicamente los valores de entrada necesarios para la operación. Además de que las variables de tipo boolean se les va a asignar un valor verdadero o falso.

Una vez que se realizó todo lo anterior ya solo resta cambiar el método calcular() en donde se almacenan las formulas correspondientes para cada operación del centro de trabajo. La fórmula para la operación Bardon 1 es la siguiente:

```

137     If bardon = True Then
138         tiempo_maquina = Math.Round(((135.02 + _tc) / (36 * _pzas_cuff))*100,3)

```

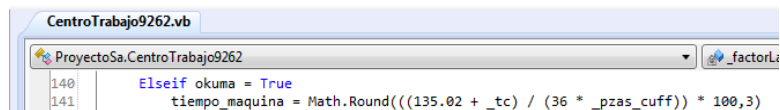
Figura 170 Formula correspondiente a la operación Bardon 1 del CT 9262.

Lo primero que se hace es una condición if en la que si el valor de la variable bardon es true, como se mencionó atrás el valor de la variable va ser true siempre y cuando se seleccione la operación Bardon 1, y vez que se insertaron los valores

necesarios al dar clic en el botón se calculara el tiempo estándar de esta operación.

Operación Okuma 1

Siguiendo con el método calcular del centro de trabajo si se selecciona la operación Okuma 1 el valor de la variable okuma va a ser true, y una vez que se ingresaron los valore de entrada al formulario al momento de dar clic al botón del formulario se llevara a cabo el cálculo del tiempo estándar correspondiente a esta operación. El código de esta operación es el siguiente.

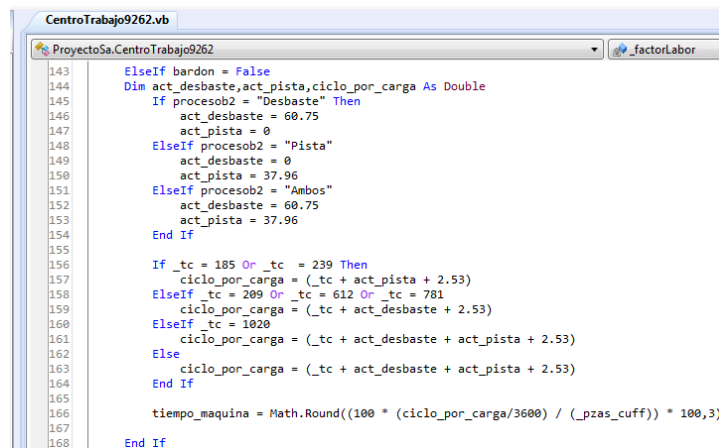


```
CentroTrabajo9262.vb
ProyectoSa.CentroTrabajo9262
140 Elseif okuma = True
141 tiempo_maquina = Math.Round(((135.02 + _tc) / (36 * _pzas_cuff)) * 100,3)
```

Figura 171 Formula correspondiente a la operación Okuma 1 del CT 9262.

Operación Bardon 2

Y si se selecciona la operación Bardon 2 entonces el valor de la variable bardon va a ser false, y después se va a realizar una condición if en la que el valor de las variables act_debaste y act_pista va a variar dependiendo del proceso que se elija, después se realiza otra condición if en la que el valor de la variable ciclo_por_carga va a variar dependiendo del valor de tiempo ciclo que el usuario ingrese. Y una vez que se ingresaron los valores de entrada y se obtuvieron los valores de las condiciones se procede a realizar el cálculo del tiempo estándar.



```
CentroTrabajo9262.vb
ProyectoSa.CentroTrabajo9262
143 Elseif bardon = False
144 Dim act_debaste,act_pista,ciclo_por_carga As Double
145 If procesob2 = "Desbaste" Then
146 act_debaste = 60.75
147 act_pista = 0
148 Elseif procesob2 = "Pista"
149 act_debaste = 0
150 act_pista = 37.96
151 Elseif procesob2 = "Ambos"
152 act_debaste = 60.75
153 act_pista = 37.96
154 End If
155
156 If _tc = 185 Or _tc = 239 Then
157 ciclo_por_carga = (_tc + act_pista + 2.53)
158 Elseif _tc = 209 Or _tc = 612 Or _tc = 781
159 ciclo_por_carga = (_tc + act_debaste + 2.53)
160 Elseif _tc = 1020
161 ciclo_por_carga = (_tc + act_debaste + act_pista + 2.53)
162 Else
163 ciclo_por_carga = (_tc + act_debaste + act_pista + 2.53)
164 End If
165
166 tiempo_maquina = Math.Round((100 * (ciclo_por_carga/3600) / (_pzas_cuff)) * 100,3)
167
168 End If
```

Figura 172 Código correspondiente a la operación Bardon 2 del CT 9262.

Además se creó el siguiente método el cual agrega los elementos que va a contener el ComboBox.


```

CentroTrabajo9262.vb
ProyectoSa.CentroTrabajo9262
182 Private Sub llenar_proceso()
183     Dim proceso As New List(Of String)
184     proceso.Add("Desbaste")
185     proceso.Add("Pista")
186     proceso.Add("Ambos")
187     For Each m In proceso
188         cbo_proceso.Items.Add(m)
189     Next
190 End Sub

```

Figura 173 Método llenar_proceso().

6.3.7 CentroTrabajo9263.vb – CNC C'BORE & PRETTERN L.B.

Este centro de trabajo se divide en dos operaciones la cuales son Okuma 2 y Cincinnati. Como se divide en dos operaciones se tuvo que modificar la interfaz gráfica de tal manera que se pueda calcular el tiempo estándar de las dos operaciones en un mismo formulario. A continuación se muestra el antes y el después del formulario.

Figura 174 Formulario antes de la actualización.

Figura 175 Formulario actualizado.

Como en el centro de trabajo anterior cada operación utiliza diferentes valores de entrada, por lo tanto se deben de desactivar algunos, en este caso cuando se selecciona la operación Cincinnati se debe de desactivar el valor de entrada proceso y si se selecciona la operación Okuma 2 se debe de activar, para ello se agregó el evento CheckedChanged a los dos radiobutton del formulario y se agregó el siguiente código.

```

CentroTrabajo9263.vb
ProyectoSa.CentroTrabajo9263
198 Sub Okuma2CheckedChanged(sender As Object, e As EventArgs)
199     cbo_proceso.Enabled = True
200 End Sub
201
202 Sub Cincinnati1CheckedChanged(sender As Object, e As EventArgs)
203     cbo_proceso.Enabled = False
204 End Sub

```

Figura 176 Evento CheckedChanged correspondiente a los radiobutton.

Después se agregó una variable de tipo boolean llamada okuma la cual es necesaria para poder realizar el cálculo del tiempo estándar de la operación con el mismo nombre.

```

109 '-----variables que se necesitan-----
110 Dim anillos_xcuff As Double
111 Dim t_ciclo As Double
112 Dim proceso As String
113 Dim okuma As Boolean
114 '-----

```

Figura 177 Variables necesarias para almacenar los valores de entrada.

Una vez que se declararon las variables se deben de procesar los datos de entrada, almacenándolos en sus respectivas variables. Para ello se agregó el siguiente código al evento clic del botón del formulario.

```

173 Sub Button1Click(sender As Object, e As EventArgs)
174 If okuma2.Checked = True Then
175     If IsNumeric(textBox2.Text) And IsNumeric(textBox1.Text) And cbo_proceso.Text <> "" Then
176         anillos_xcuff = CDec(textBox2.Text)
177         t_ciclo = CDec(textBox1.Text)
178         proceso = cbo_proceso.Text
179         okuma = True
180         calcular()
181         Close()
182     Else
183         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
184     End If
185 ElseIf cincinnati1.Checked = True
186     If IsNumeric(textBox2.Text) And IsNumeric(textBox1.Text) Then
187         anillos_xcuff = CDec(textBox2.Text)
188         t_ciclo = CDec(textBox1.Text)
189         okuma = False
190         calcular()
191         Close()
192     Else
193         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
194     End If
195 End If
196 End Sub

```

Figura 178 Código del evento clic del botón del formulario.

Como se muestra en la imagen anterior si se selecciona la operación Okuma 2 entonces el valor de la variable okuma va ser true, y si se selecciona la operación Cincinnati entonces el valor de la variable okuma va ser false.

Ahora se debe de crear un método el cual va a agregar los elementos que va a contener el ComboBox del proceso.

```

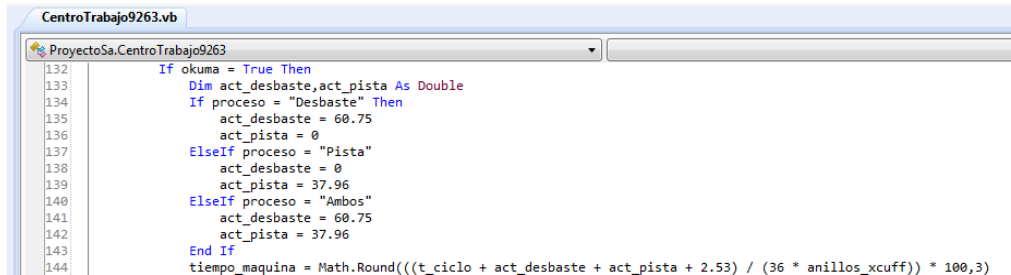
163 Private Sub llenar_proceso()
164     Dim proceso As New List(Of String)
165     proceso.Add("Desbaste")
166     proceso.Add("Pista")
167     proceso.Add("Ambos")
168     For Each m In proceso
169         cbo_proceso.Items.Add(m)
170     Next
171 End Sub

```

Figura 179 Método llenar_proceso().

Operación Okuma 2

Después de que se realizó todo lo anterior se debe de modificar el método calcular() del centro de trabajo, de tal manera que obtenga el valor del tiempo estándar dependiendo de la operación que se seleccione. Para ello se crea una condición if en donde si el valor de la variable okuma es true entonces va a realizar las operaciones correspondientes a la operación Okuma 2. El código de esta operación es el siguiente:



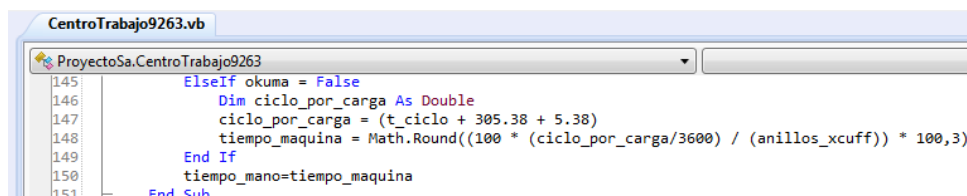
```
132     If okuma = True Then
133         Dim act_desbaste,act_pista As Double
134         If proceso = "Desbaste" Then
135             act_desbaste = 60.75
136             act_pista = 0
137         ElseIf proceso = "Pista"
138             act_desbaste = 0
139             act_pista = 37.96
140         ElseIf proceso = "Ambos"
141             act_desbaste = 60.75
142             act_pista = 37.96
143         End If
144         tiempo_maquina = Math.Round(((t_ciclo + act_desbaste + act_pista + 2.53) / (36 * anillos_xcuff)) * 100,3)
```

Figura 180 Código de la operación Okuma 2.

Como se puede apreciar en la imagen anterior primero se realiza una condición if en la que el valor de las variables act_desbaste y act_pista va a variar dependiendo del proceso que el usuario seleccione y una vez que se obtiene el valor de las variables mencionadas, se procede a realizar el cálculo del tiempo estándar.

Operación Cincinnati

Si se selecciona la operación Cincinnati entonces el valor de la variable okuma es false y con ello se lleva a cabo el cálculo del tiempo estándar correspondiente. El código de esta operación es el siguiente:



```
145     ElseIf okuma = False
146         Dim ciclo_por_carga As Double
147         ciclo_por_carga = (t_ciclo + 305.38 + 5.38)
148         tiempo_maquina = Math.Round((100 * (ciclo_por_carga/3600) / (anillos_xcuff)) * 100,3)
149     End If
150     tiempo_mano=tiempo_maquina
151 End Sub
```

Figura 181 Código de la operación Cincinnati.

6.3.8 CentroTrabajo9266.vb – TURN O.D., I.D. CUFF L.B.

Este centro de trabajo se divide en dos operaciones las cuales son puma y puma 240. Como se divide en dos operaciones se tuvo que modificar la interfaz gráfica de tal manera que se pueda calcular el tiempo estándar de las dos operaciones en

un mismo formulario. A continuación se muestra el antes y el después del formulario.

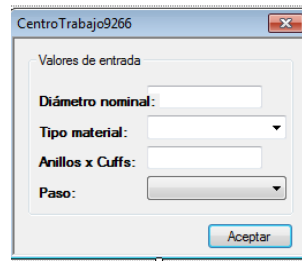


Figura 182 Formulario antes de la actualización.

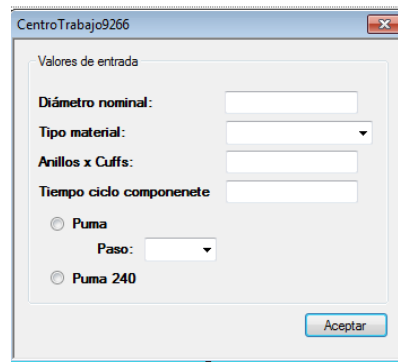


Figura 183 Formulario actualizado.

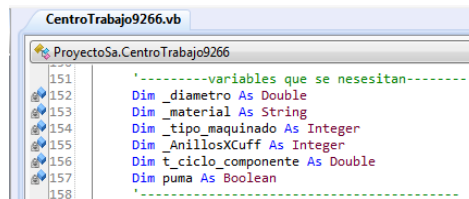
Al igual que en otros centros de trabajo que tienen varias operaciones, cada operación utiliza sus valores de entrada, por lo tanto se activaran o desactivaran dichos valores de entrada dependiendo de la operación que se seleccione. Para hacer esto posible se les agrego a los radiobutton el evento `CheckedChanged` y dentro del evento se agregó el siguiente código:

```
CentroTrabajo9266.vb
ProyectoSa.CentroTrabajo9266
258 Sub Rb_pumaCheckedChanged(sender As Object, e As EventArgs)
259     txt_diametro.Enabled = True
260     cbo_material.Enabled = True
261     cbo_paso.Enabled = True
262     txt_tcc.Enabled = False
263     txt_tcc.Text = ""
264 End Sub
265
266 Sub Rb_puma240CheckedChanged(sender As Object, e As EventArgs)
267     txt_diametro.Enabled = False
268     txt_diametro.Text = ""
269     cbo_material.Enabled = False
270     cbo_material.Text = ""
271     cbo_paso.Enabled = False
272     cbo_paso.Text = ""
273     txt_tcc.Enabled = True
274 End Sub
```

Figura 184 Evento `CheckedChanged` de los radiobuttons del formulario.

Como se mencionó arriba dependiendo de la operación que se elija, se van a activar únicamente los valores de entrada necesarios para realizar el cálculo del tiempo estándar de la operación elegida, y pues obviamente los valores de entrada que no son necesarios se desactivaran.

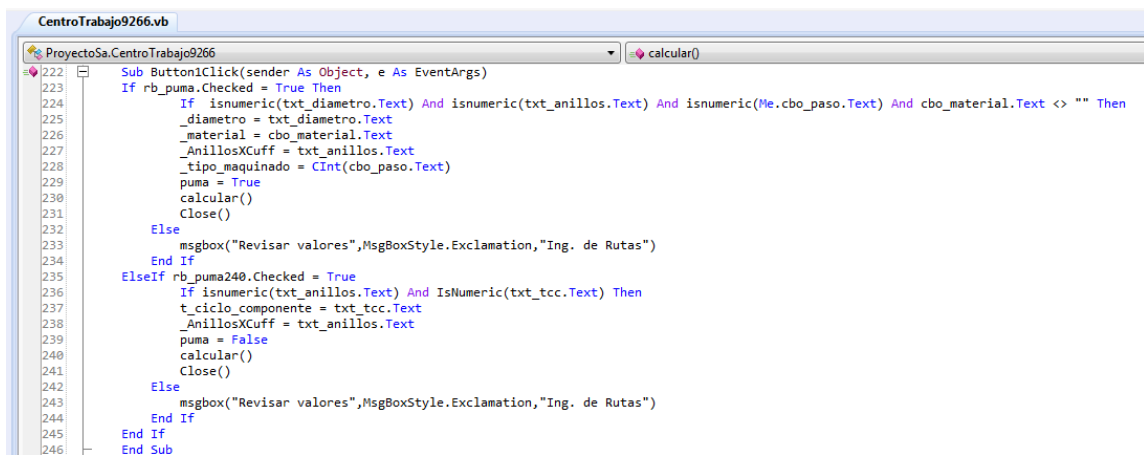
Una vez que se realizó lo anterior se declaró una variable de tipo boolean llamada “puma” y su valor va ser true si el usuario elige la operación puma, y si el usuario elige la operación puma 240 entonces el valor va ser false.



```
151 '-----variables que se necesitan-----
152 Dim _diametro As Double
153 Dim _material As String
154 Dim _tipo_maquinado As Integer
155 Dim _AnillosXCuff As Integer
156 Dim t_ciclo_componente As Double
157 Dim puma As Boolean
158
```

Figura 185 Variables que se necesitan para almacenar los valores de entrada.

Para almacenar los valores de entrada en las variables que declaramos hay que agregar el siguiente código al evento clic del botón del formulario.



```
222 Sub Button1Click(sender As Object, e As EventArgs)
223     If rb_puma.Checked = True Then
224         If IsNumeric(txt_diametro.Text) And IsNumeric(txt_anillos.Text) And IsNumeric(Me.cbo_paso.Text) And cbo_material.Text <> "" Then
225             _diametro = txt_diametro.Text
226             _material = cbo_material.Text
227             _AnillosXCuff = txt_anillos.Text
228             _tipo_maquinado = CInt(cbo_paso.Text)
229             puma = True
230             calcular()
231             Close()
232         Else
233             MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
234         End If
235     ElseIf rb_puma240.Checked = True
236         If IsNumeric(txt_anillos.Text) And IsNumeric(txt_tcc.Text) Then
237             t_ciclo_componente = txt_tcc.Text
238             _AnillosXCuff = txt_anillos.Text
239             puma = False
240             calcular()
241             Close()
242         Else
243             MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
244         End If
245     End If
246 End Sub
```

Figura 186 Evento clic del botón del formulario.

Lo primero que realiza el código anterior es un condición if en la que si el usuario elige la operación puma va a almacenar los valores de entrada necesarios para esta operación en sus respectivas variables, y como se mencionó atrás el valor de la variable “puma” es true, y si el usuario elige la operación puma240 realiza lo mismo que la operación anterior con la diferencia de que el valor de la variable “puma” es false.

Después de haber realizado lo anterior se debe de modificar el método calcular() del centro de trabajo ya que ahí es donde se agregaron las fórmulas de las dos operaciones. A continuación se muestra el código de las dos operaciones.

Operación Puma

```
CentroTrabajo9266.vb
ProyectoSa.CentroTrabajo9266 calcular()
179 If puma = True Then
180   If _tipo_maquinado = 1 Then
181     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
182       t_ciclo = 263.44
183     ElseIf tipo = "HIERRO DUCTIL"
184       t_ciclo = 329.3
185     Else
186       MsgBox("Material " & tipo & " no disponible para el C.C 9266, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
187       Me.ListaAlertas.Add(Name & ": El material " & tipo & " no se encuentra disponible para el calculo, verificar con Ing. Industrial")
188       Exit Sub
189     End If
190     tiempo_maquina = Math.Round(((7.42271632143 + ((_diametro * t_ciclo) / 326.232)) / (_AnillosXCuff)) * 100,3)
191   ElseIf _tipo_maquinado = 2
192     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
193       t_ciclo = 381.60
194     ElseIf tipo = "HIERRO DUCTIL"
195       t_ciclo = 447.00
196     Else
197       MsgBox("Material " & tipo & " no disponible para el C.C 9266, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
198       Me.ListaAlertas.Add(Name & ": El material " & tipo & " no se encuentra disponible para el calculo, verificar con Ing. Industrial")
199       Exit Sub
200     End If
201     tiempo_maquina = Math.Round(((7.57602083333 + ((_diametro * t_ciclo) / 326.232)) / (_AnillosXCuff)) * 100,3)
202   End If
203 End If
```

Figura 187 Código de la operación Puma del CT 9266.

Como se puede apreciar en la imagen anterior lo primero que se hace es una condición if en la que dependiendo del paso que el usuario seleccione, se realizara otra condición if en la que el valor del tiempo ciclo va a variar dependiendo del material que el usuario seleccione y una vez que se obtiene el valor del tiempo ciclo se procede a realizar el cálculo de la formula correspondiente a esta operación.

Operación Puma 240

```
CentroTrabajo9266.vb
ProyectoSa.CentroTrabajo9266 calcular()
204 ElseIf puma = False
205   Dim ciclo_por_carga As Double
206   ciclo_por_carga = (t_ciclo_componente + 305.38 + 5.38)
207   tiempo_maquina = Math.Round((100 * (ciclo_por_carga/3600)) / (_AnillosXCuff)) * 100,3)
208 End If
209 tiempo_mano=tiempo_maquina * factorLabor
210 End Sub
```

Figura 188 Código de la operación Puma 240 del CT 9266.

Para esta operación solo se debe de obtener el ciclo por carga y para obtener dicho valor solo se requiere que el usuario ingrese un valor de tiempo ciclo en el formulario. Y una vez que se obtiene el valor, se procede a realizar el cálculo de la fórmula de esta operación.

6.3.9 CentroTrabajo9285.vb – ABRASIVE WHEEL L.B.

Para este centro de trabajo se tuvo que crear una interfaz gráfica ya que antes de la actualización el cálculo se realizaba de manera directa y no se necesitaba que el usuario ingresara ningún valor de entrada, sin embargo con la actualización el usuario debe seleccionar un material para poder obtener el valor del tiempo estándar. A continuación se muestra el formulario.

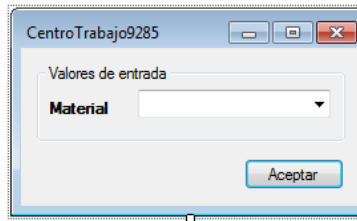


Figura 189 Formulario del CT 9285.

Despues de crear la interfaz grafica se deben de agregar los materiales a la funcion Test de la clase como se ha hecho en centros de trabajo anteriores, asi como tambien agregar el parametro al constructor de la clase e igualarlo con la variable donde se almacenara el valor que el usuario seleccione, despues se debe de agregar el valor de entrada a la listaDatosStringRequeridos y por ultimo pasar los valores de listas. Como por ejemplo se ha hecho en el CentroTrabajo9226.vb – I.D. GROOVE L.B.

Una vez que se realice lo anterior se debe de modificar el metodo calcular() del centro de trabajo agregando las nuevas operaciones. A continuacion se muestra el codigo del metodo.

```

CentroTrabajo9285.vb
ProyectoSa.CentroTrabajo9285
factorLabor
152 Public Sub calcular() Implements ICentroTrabajo.calcular
153 tiempo_septup=buscar_setupin(centro)
154 If banElementoFijo = False Then
155     tipo = Module1.sacar_tipo_material(material9285)
156 End If
157 Dim t_ciclo As Double
158 If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
159     t_ciclo = 1.66
160 ElseIf tipo = "HIERRO GRIS INTERMEDIO"
161     t_ciclo = 2.39
162 ElseIf tipo = "ACERO AL CARBON" Or tipo = "ACERO INOXIDABLE"
163     t_ciclo = 1.79
164 ElseIf tipo = "HIERRO GRIS ALTO MODULO"
165     t_ciclo = 1.79
166 ElseIf tipo = "HIERRO DUCTIL"
167     t_ciclo = 1.66
168 Else
169     MsgBox("Material " & tipo & " no disponible para el C.C 9266, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
170 End If
171 tiempo_maquina = Math.Round(((4.296789 + t_ciclo) / 36) * 100,3)
172 tiempo_mano=tiempo_maquina
173 End Sub

```

Figura 190 Método calcular() del CT 9285.

Lo primero que se realiza en este metodo es un condicion if en donde el valor de la variable t_ciclo va a variar dependiendo del material que el usuario seleccione. y una vez que se obtiene el valor de la variable se procede a realizar el calculo del tiempo estandar.

Por ultimo se debe de agregar el evento clic al boton del formulario y dentro del metodo del evento agregar el siguiente codigo.

```
CentroTrabajo9285.vb
ProyectoSa.CentroTrabajo9285
199 Sub Button1Click(sender As Object, e As EventArgs)
200     If cbo_material.Text <> "" Then
201         material19285 = cbo_material.Text
202         calcular()
203         Close()
204     End If
205 End Sub
```

Figura 191 Método del evento clic del botón del formulario.

Lo que se realiza en el código anterior es que si el ComboBox no está vacío en el momento en que el usuario da clic en el botón del formulario entonces el valor de entrada que se seleccione se almacenará en la variable que se le asignó, después se realizarán las operaciones del método calcular y por último se cerrará el formulario. Sin embargo si el ComboBox está vacío al momento de dar clic en el botón, le aparecerá un mensaje de error al usuario, además de que no se realizará el cálculo del tiempo estándar.

6.3.10 CentroTrabajo9301.vb – FINISH TURN CELL #1 L.B.

En este centro de trabajo se manejan dos operaciones las cuales son finish turn cell #1 y super finish, por lo tanto se tuvo que modificar su interfaz gráfica de este centro de trabajo de tal manera que permita realizar el cálculo de las dos operaciones en un mismo formulario. A continuación se muestra el formulario.

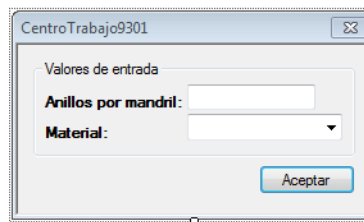
The image shows a Windows form titled 'CentroTrabajo9301'. Under the heading 'Valores de entrada', there are two labels: 'Anillos por mandril:' followed by a text input field, and 'Material:' followed by a dropdown menu. At the bottom right of the form is a button labeled 'Aceptar'.

Figura 192 Formulario antes de la actualización.

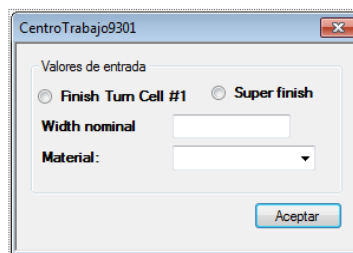
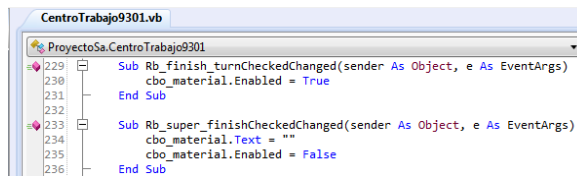
The image shows the same 'CentroTrabajo9301' form after updates. The 'Anillos por mandril' field has been replaced by two radio buttons: 'Finish Turn Cell #1' and 'Super finish'. Below them is a 'Width nominal' text input field. The 'Material' dropdown menu remains. The 'Aceptar' button is still present at the bottom right.

Figura 193 Formulario actualizado.

Como se puede apreciar en la imagen anterior se tuvieron que cambiar el valor de entrada anillos por mandril por Width nominal. Además se agregaron dos opciones las cuales son las operaciones existentes en este centro de trabajo.

Después de modificar la interfaz del formulario se les agrego el evento ChekedChanged a los radiobuttons y dentro del método del evento se agregó el siguiente código.

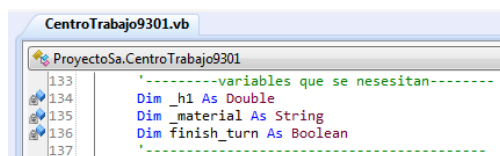


```
CentroTrabajo9301.vb
ProyectoSa.CentroTrabajo9301
229 Sub Rb_finish_turnCheckedChanged(sender As Object, e As EventArgs)
230     cbo_material.Enabled = True
231 End Sub
232
233 Sub Rb_super_finishCheckedChanged(sender As Object, e As EventArgs)
234     cbo_material.Text = ""
235     cbo_material.Enabled = False
236 End Sub
```

Figura 194 Evento ChekedChanged de los radiobuttons del formulario.

La función que realiza el código anterior es que cuando se selecciona la operación finish turn se activa el ComboBox de material, en cambio sí se selecciona la operación super finish se desactiva dicho elemento del formulario.

Una vez que se agregó el código anterior se procede a declarar las variables donde se almacenan los valores de entrada, las variables son las siguientes:

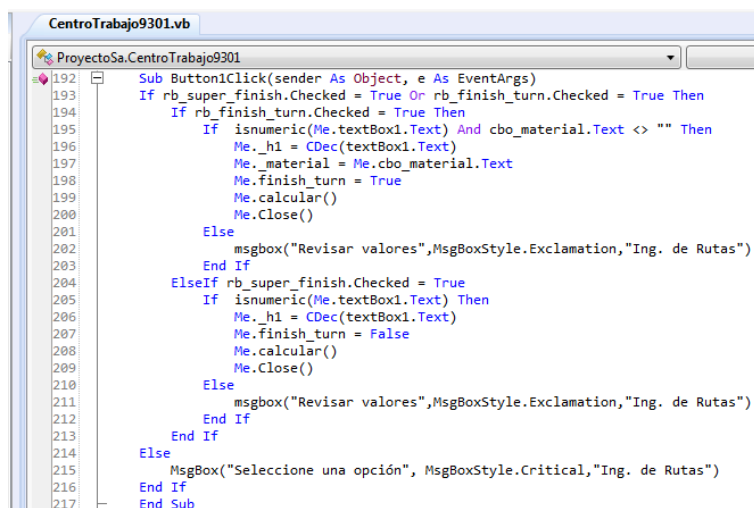


```
CentroTrabajo9301.vb
ProyectoSa.CentroTrabajo9301
133 '-----variables que se neesitan-----
134 Dim _h1 As Double
135 Dim _material As String
136 Dim finish_turn As Boolean
137
```

Figura 195 Variables donde se almacenan los valores de entrada.

Como se muestra en la imagen se declaró una variable de tipo boolean llamada finish_turn la cual será de utilidad al momento de agregar la formula y las operaciones correspondientes a las operaciones del centro de trabajo.

Una vez que se declararon las variables se agregó el siguiente código al evento clic del botón del formulario.



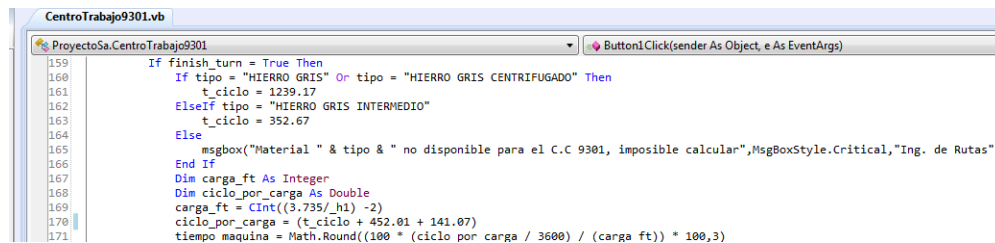
```
CentroTrabajo9301.vb
ProyectoSa.CentroTrabajo9301
192 Sub Button1Click(sender As Object, e As EventArgs)
193     If rb_super_finish.Checked = True Or rb_finish_turn.Checked = True Then
194         If rb_finish_turn.Checked = True Then
195             If isnumeric(Me.textBox1.Text) And cbo_material.Text <> "" Then
196                 Me._h1 = CDec(textBox1.Text)
197                 Me._material = Me.cbo_material.Text
198                 Me.finish_turn = True
199                 Me.calcular()
200                 Me.Close()
201             Else
202                 msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
203             End If
204         ElseIf rb_super_finish.Checked = True
205             If isnumeric(Me.textBox1.Text) Then
206                 Me._h1 = CDec(textBox1.Text)
207                 Me.finish_turn = False
208                 Me.calcular()
209                 Me.Close()
210             Else
211                 msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
212             End If
213         End If
214     Else
215         MsgBox("Seleccione una opción", MsgBoxStyle.Critical,"Ing. de Rutas")
216     End If
217 End Sub
```

Figura 196 Código del evento clic del botón del formulario.

La función que realiza el código anterior es que si el usuario elige la operación finish turn entonces se van a almacenar los valores de entrada necesarios en sus respectivas variables, además de que el valor de la variable finish_turn de tipo boolean será true, pero si el usuario elige la operación super finish se realiza lo mismo que con la otra operación, con la diferencia de que el valor de la variable finish_turn pasa a ser false.

Una vez realizado lo anterior se debe de agregar las formulas correspondientes a cada operación dentro del método calcular().

Operación Finish Turn Cell #1

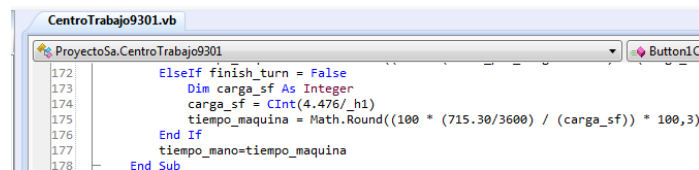


```
159     If finish_turn = True Then
160     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
161         t_ciclo = 1239.17
162     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
163         t_ciclo = 352.67
164     Else
165         MsgBox("Material " & tipo & " no disponible para el C.C 9301, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
166     End If
167     Dim carga_ft As Integer
168     Dim ciclo_por_carga As Double
169     carga_ft = CInt((3.735/h1) - 2)
170     ciclo_por_carga = (t_ciclo + 452.01 + 141.07)
171     tiempo_maquina = Math.Round((100 * (ciclo por carga / 3600) / (carga ft)) * 100,3)
```

Figura 197 Código para obtener el tiempo estándar de la operación Finish Turn Cell #1.

Como se puede ver en la imagen anterior lo primero que se realiza es una condición if en donde el valor de la variable t_ciclo va a variar dependiendo del material que el usuario seleccione, después se declaran dos variables las cuales almacenan una operación matemática la cual depende de los valores que el usuario ingrese al formulario. Y ya que se tienen los valores de las variables se procede a realizar el cálculo de la fórmula del tiempo estándar correspondiente a esta operación.

Operación Super Finish



```
172     ElseIf finish_turn = False
173     Dim carga_sf As Integer
174     carga_sf = CInt(4.476/h1)
175     tiempo_maquina = Math.Round((100 * (715.30/3600) / (carga_sf)) * 100,3)
176     End If
177     tiempo_mano=tiempo_maquina
178 End Sub
```

Figura 198 Código para obtener el tiempo estándar de la operación Super Finish.

Para esta operación solo se necesita obtener un valor para poder realizar el cálculo, el valor es la carga de la operación, este valor se almacena en la variable carga_sf y depende del valor de Width que el usuario ingrese en el formulario. Una vez obtenido el valor de la variable mencionada se procede a realizar el cálculo del tiempo estándar de la operación.

6.3.11 CentroTrabajo9302.vb – Finish Turn Cell #2

Para este centro de trabajo se tuvo que cambiar la interfaz gráfica del formulario ya que con los elementos que contenía dicho formulario no eran suficientes para poder llevar a cabo el cálculo del tiempo estándar y por esa razón se tuvo que modificar su interfaz. A continuación se muestra el antes y el después del formulario.

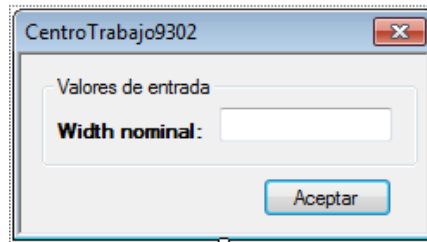


Figura 199 Formulario antes de la actualización.

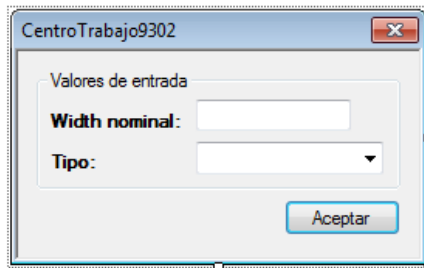


Figura 200 Formulario actualizado.

Como se puede apreciar en la imagen lo único que se agregó el valor de entrada tipo.

Después de que se modificó el formulario se agrega una variable llamada tipo que es donde se almacena el elemento del ComboBox que el usuario selecciono.

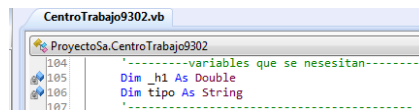


Figura 201 Variables donde se almacenan los diferentes valores de entrada.

Después de haber realizado lo anterior se modificó el método calcular() del centro de trabajo. a continuación se muestra el código de dicho método.

```
CentroTrabajo9302.vb
ProyectoSa.CentroTrabajo9302
calculador()
Public Sub calcular() implements ICentroTrabajo.calcular
122 tiempo_septup=buscar_setupin(centro)
123
124 Dim carga As Integer
125 Dim t_ciclo As Double
126 If tipo = "BSG10" Then
127     t_ciclo = 588.01
128     carga = 25
129 ElseIf tipo = "OTROS"
130     t_ciclo = 427.35
131     carga = 14
132 Else
133     MsgBox("Tipo no disponible para el C.C 9302, imposible calcular", MsgBoxStyle.Critical,"Ing. de Rutas")
134 End If
135 tiempo_maquina = Math.Round(((t_ciclo + 452.54) / (36 * carga)) * 100,3)
136 tiempo_mano=tiempo_maquina
137 End Sub
```

Figura 202 Método calcular() del CT 9302.

Lo primero que se realiza en este método es una condición if en la el valor de las variables t_ciclo y carga va a variar dependiendo del tipo que seleccione el usuario. Y una vez que se obtienen los valores de esas variables se procede a realizar el cálculo del tiempo estándar.

6.3.12 CentroTrabajo9305.vb – GAP NOTCH IDENTIFICATION L.B.

A este centro de trabajo se le tuvo que crear una interfaz gráfica ya que anteriormente no se necesitaban valores de entrada para poder realizar el cálculo y este se realizaba de manera directa, sin embargo ahora con la actualización, el usuario debe agregar algunos valores de entrada al formulario de este centro de trabajo para poder llevar a cabo el cálculo. A continuación se muestra la apariencia del formulario.

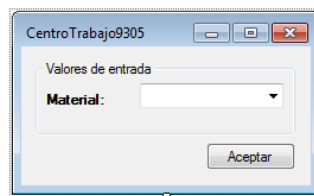


Figura 203 Formulario del CT 9305.

Como se puede ver en la imagen anterior el único valor de entrada que se agrego fue el de material.

Una vez creada la interfaz gráfica, se debe de agregar una nueva variable de tipo String en la cual se almacenara el material que el usuario seleccione. Después de declarar dicha variable se debe de agregar el siguiente código a la función Test de la clase.

```

CentroTrabajo9305.vb*
ProyectoSa.CentroTrabajo9305
Test(IDatos As List(Of Dato), IDatosBool As List(Of DatoBoolean), IDatosString As List(Of DatoString))
186 Public Function Test(ByVal IDatos As List(Of Dato), ByVal IDatosBool As List(Of DatoBoolean), ByVal IDatosString As List(Of DatoString)) As Boolean Implements ICentroTrabajo.Test
187     material9305 = Module1.obtenerValorDatoString(IDatosString, "MATERIAL")
188     tipo = Module1.sacar_tipo_material(material9305)
189
190     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Or tipo = "HIERRO GRIS INTERMEDIO" Or tipo = "HIERRO DUCTIL" Or tipo = "ACERO AL CARBON" Or tipo = "ACERO INOXIDABLE" Then
191
192     Else
193         MsgBox("Material " & tipo & " no disponible para el C.C " & CentroTrabajo & " (" & obtener_nombre_operacion(CentroTrabajo) & "), " & vbCrLf & "" & _
194             "Por favor seleccione un tipo de material disponible para poder calcular el centro de trabajo", MsgBoxStyle.Critical, "Ing. de Rutas")
195         Dim listaOpcionales As New List(Of String)
196         listaOpcionales.Add("HIERRO GRIS")
197         listaOpcionales.Add("HIERRO GRIS CENTRIFUGADO")
198         listaOpcionales.Add("HIERRO GRIS INTERMEDIO")
199         listaOpcionales.Add("HIERRO DUCTIL")
200         listaOpcionales.Add("ACERO AL CARBON")
201         listaOpcionales.Add("ACERO INOXIDABLE")
202         Dim f As New Seleccionar(listaOpcionales)
203         If f.ShowDialog() = DialogResult.OK Then
204             tipo = f.elementoSeleccionado
205             banElementoFijo = True
206         End If
207     End Function
208 End If
209 Return True
210 End Function

```

Figura 204 Función Test del CT 9305.

En el código anterior se definen los materiales que se van a usar en este centro de trabajo.

Como se agregó un nuevo valor de entrada se debe de agregar un nuevo parámetro al constructor de la clase como se muestra a continuación.

```

CentroTrabajo9305.vb*
ProyectoSa.CentroTrabajo9305
128 Public Sub New(ByVal material_ As String)
129     material9305 = material_
130 End Sub

```

Figura 205 Parámetros del constructor de la clase.

Además ese parámetro se iguala con la variable que almacena el valor de entrada. Luego el valor de entrada se debe de agregar a listaDatosStringRequeridos como en la siguiente imagen.

```

CentroTrabajo9305.vb*
ProyectoSa.CentroTrabajo9305
tiempo_septup
121 Public Sub New()
122     Me.InitializeComponent()
123
124     Dim _material As New DatoString("MATERIAL", "Especificación de materia prima (MF012-S, SPR-128, MF025-L, etc...)", "MATERIAL:")
125     listaDatosStringRequeridos.Add(_material)
126 End Sub

```

Figura 206 Agregando valores de entrada a la listaDatosStringRequeridos.

Después se deben de pasar valores de listas como en la siguiente imagen.

```

CentroTrabajo9305.vb*
ProyectoSa.CentroTrabajo9305
tiempo_septup
115 Public Sub pasarValoresDeListas(ByVal IDatos As List(Of Dato), ByVal IDatosBool As List(Of DatoBoolean), ByVal IDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
116     If banElementoFijo = False Then
117         material9305 = Module1.obtenerValorDatoString(IDatosString, "MATERIAL")
118     End If
119 End Sub

```

Figura 207 Pasando valores de listas.

Ahora se debe de agregar el evento clic al botón del formulario, y dentro del método del evento se agregó el siguiente código.

```

CentroTrabajo9305.vb
ProyectoSa.CentroTrabajo9305
195 Sub Button1Click(sender As Object, e As EventArgs)
196     If cbo_material.Text <> "" Then
197         material9305 = cbo_material.Text
198         calcular()
199         Close()
200     Else
201         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
202     End If
203 End Sub

```

Figura 208 Método del evento clic del botón del formulario.

Lo que se realiza en el metodo anterior es una condicion en donde si el ComboBox no esta vacio al momento que el usuario da clic en el boton del formulario entonces el material que el usuario elija se almacenara en la variable que se asigno, despues se realiza el metodo calcular y por ultimo se cerrara el formulario. Pero si al momento de dar clic el ComboBox esta vacio se mostrara un mensaje de error informando que se revisen los valores.

Después de haber realizado todo lo anterior se modificó el método calcular() del centro de trabajo. A continuación se muestra el código de dicho método.

```

CentroTrabajo9305.vb
ProyectoSa.CentroTrabajo9305 tiempo_septup
150 Public Sub calcular() implements ICentroTrabajo.calcular
151     tiempo_septup=buscar_setupin(centro)
152     If banElementoFijo = False Then
153         tipo = Module1.sacar_tipo_material(material9305)
154     End If
155     Dim t_ciclo As Double
156     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
157         t_ciclo = 0.49
158     ElseIf tipo = "ACERO AL CARBON" Or tipo = "ACERO INOXIDABLE"
159         t_ciclo = 0.49
160     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
161         t_ciclo = 0.64
162     ElseIf tipo = "HIERRO DUCTIL"
163         t_ciclo = 0.64
164     Else
165         msgbox("Material " & tipo & " no disponible para el C.C 9305, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
166     End If
167     tiempo_maquina = Math.Round(((4.87 + t_ciclo) / 36 ) * 100,3)
168     tiempo_manos=tiempo_maquina
169 End Sub

```

Figura 209 Método calcular del CT 9305.

Lo primero que se realiza dentro de este método es una condición if en donde el valor de la variable t_ciclo va a variar dependiendo del material que el usuario elija. Y una vez que se obtiene el valor de la variable se procede al realizar el cálculo del tiempo estándar.

6.3.13 CentroTrabajo9329.vb – LASER MARK L.B.

A este centro de trabajo se le realizo una pequeña modificación a su interfaz gráfica. A continuación se muestra el antes y el después del formulario.

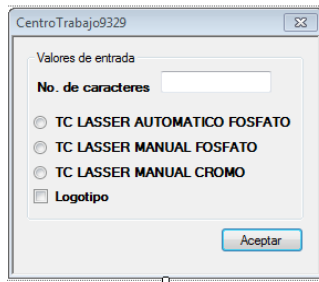


Figura 210 Formulario antes de la actualización.

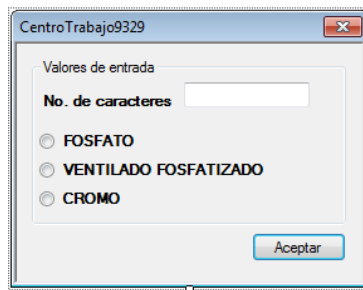


Figura 211 Formulario actualizado.

Como se puede apreciar en la imagen anterior solo se cambió el texto de los radiobuttons, además de que se eliminó la casilla de logotipo.

En este caso no hubo la necesidad de agregar nuevas variables, ya que con las variables que contaba la clase son suficientes para almacenar los valores de entrada del formulario.

Después de haber realizado la pequeña modificación al formulario, se procedió a modificar código del evento clic del botón del formulario como se muestra en la siguiente imagen.

```

CentroTrabajo9329.vb
ProyectoSa.CentroTrabajo9329
194 Sub Button1Click(sender As Object, e As EventArgs)
195     If IsNumeric(Me.textBox1.Text) And rb_fosfato.Checked = True Or rb_ventilado_fosfato.Checked = True Or rb_cromo.Checked = True Then
196         Me._nocaracteres = Me.textBox1.Text
197         If Me.rb_fosfato.Checked = True Then
198             Me._tipo = "F"
199         ElseIf Me.rb_ventilado_fosfato.Checked
200             Me._tipo = "VF"
201         Else
202             Me._tipo = "C"
203         End If
204         calcular()
205         close()
206     Else
207         MsgBox("Revisar valores", MsgBoxStyle.Exclamation, "Ing. de Rutas")
208     End If
209 End Sub

```

Figura 212 Método del evento clic del formulario.

La función que realiza el código anterior es que si el usuario ingresa un valor numérico en el campo de texto correspondiente al número de caracteres, además de que selecciono una opción, entonces va a proceder a almacenar el valor del número de caracteres en la variable que se le asigno, posteriormente se realizara un condición if en la que el valor de la variable tipo a ser igual a las iniciales del tipo de material que el usuario elija.

Después de modificar el evento clic solo resta modificar el código del método calcular() con las operaciones y formulas actualizadas. A continuación se muestra el código del método.

```

CentroTrabajo9329.vb*
ProyectoSa.CentroTrabajo9329
Public Sub calcular() implements ICentroTrabajo.calcular
Try
    tiempo_septup=buscar_setupin(centro)
    Dim t_ciclo_caracter As Double
    If _tipo = "F" Then
        t_ciclo_caracter = 0.0530
    ElseIf _tipo = "VF"
        t_ciclo_caracter = 0.0830
    ElseIf _tipo = "C"
        t_ciclo_caracter = 0.2983
    End If
    'tiempo_maquina = ((3.62111+((_nocaracteres*t_ciclo_marcado)+(t_ciclo_logotipo)))/36)
    'tiempo_maquina = math.Round(tiempo_maquina * 100,3)
    tiempo_maquina = Math.Round((((t_ciclo_caracter * _nocaracteres) + 13.745206) / 36) * 100,3)
    tiempo_mano = tiempo_maquina
Catch er As Exception
Me.ListaAlertas.Add(Name & ": " & er.Message)
End Try
End Sub

```

Figura 213 Método calcular del CT 9329.

Como se había mencionado en el evento clic la variable tipo va a tomar el valor de las iniciales del tipo de material que el usuario elija, entonces eso nos ayuda dentro de este método ya que el valor de la variable t_ciclo_caracter va a variar dependiendo del valor que haya tomado la variable tipo, y una vez obtenido el valor de t_ciclo_caracter se procede al realizar el cálculo de la fórmula del tiempo estándar, la cual se almacena en la variable tiempo_maquina.

6.3.14 CentroTrabajo9341.vb – SIDE PLATE L.B.

A este centro de trabajo se le hicieron algunas modificaciones a su interfaz gráfica ya algunos elementos con los que contaba dicho formulario ya no eran de utilidad para poder llevar a cabo los cálculos de este centro de trabajo. Por este motivo se tuvo que modificar el formulario, quedando de la siguiente manera.

Figura 214 Formulario antes de la actualización.

Figura 215 Formulario actualizado.

Después de haber realizado la modificación del formulario, solo se modificó el método calcular() del centro de trabajo. A continuación se muestra el código del método.

```

CentroTrabajo9341.vb
ProyectoSa.CentroTrabajo9341
calculador()
Public Sub calcular() implements ICentroTrabajo.calcular
166:
167:
168:   tiempo_septup=buscar_setupin(centro)
169:   Dim ciclo_por_carga,act_ciclica_tipo_proceso As Double
170:   Dim tam_de_carga As Integer
171:   If tipo_anillo = "Gasolina" Then
172:     act_ciclica_tipo_proceso = 387.49
173:   Else
174:     act_ciclica_tipo_proceso = 109.39
175:   End If
176:   If tipo_anillo = "Gasolina" And _noLadosACromar = 2 Then
177:     tam_de_carga = (28 * _anillosXRaqueta * 1 * _noLadosACromar)
178:   Else
179:     tam_de_carga = (28 * _anillosXRaqueta * 1)
180:   End If
181:   ciclo_por_carga = (t_ciclo_cromado + act_ciclica_tipo_proceso + 27.58)
182:   tiempo_maquina = Math.Round((100 * (ciclo_por_carga / 3600) / (tam_de_carga)) * 100, 3)

```

Figura 216 Cálculos para obtener el tiempo estándar del CT 9341.

Lo primero que se realiza es una condición if en la que dependiendo del tipo de anillo que seleccione el usuario, la variable act_ciclicas_tipo_proceso va a tomar diferentes valores. Después se realiza otra condición if en donde si el usuario selecciona el tipo de anillo "Gasolina" y además elige 2 lados a cromar entonces esta condición va a tener una operación matemática única para obtener el valor del tamaño de carga, sino se cumple la condición se usara otra operación. Una vez que se obtienen todos los valores se procede a realizar el cálculo de la fórmula del tiempo estándar de este centro de trabajo.

6.3.15 CentroTrabajo9361.vb – CHROME PLATE I.D. & O.D. L.B.

Este centro de trabajo cuenta con dos operaciones las cuales son I.D. y O.D., anteriormente solo se manejaba una sola operación, pero como ahora son dos se tuvo que cambiar la interfaz gráfica del formulario de tal forma que sea posible calcular el tiempo estándar de las dos operaciones en un mismo formulario.

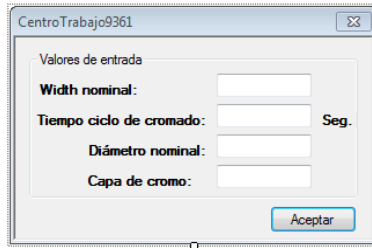


Figura 217 Formulario antes de la actualización.

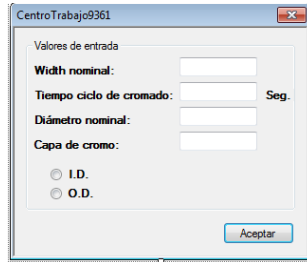


Figura 218 Formulario actualizado.

Una vez que se actualizó la interfaz gráfica del formulario se declaró una variable de tipo boolean llamada `dimension_id` dentro de las variables necesarias, esta variable sirve para separar los procedimientos de cada operación dentro del método `calcular()`.

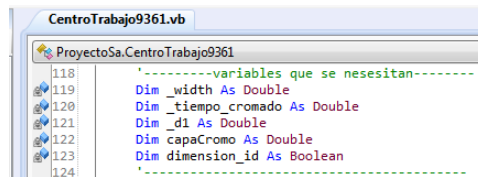


Figura 219 Variables donde se almacenan los valores de entrada.

Después de declarar la variable se agregó el evento `CheckedChanged` a los radiobuttons del formulario y dentro del método del evento se agregó el siguiente código.

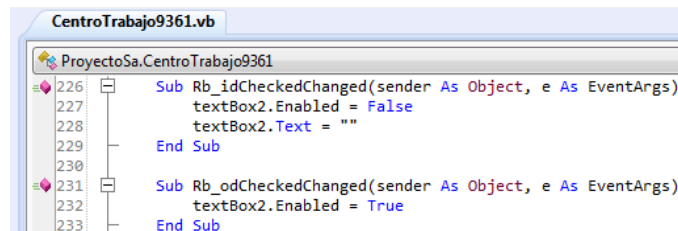


Figura 220 Evento `CheckedChanged` de los radiobuttons.

Lo que realiza el código anterior es que cuando se elige la operación `id` se desactiva el campo de texto del tiempo ciclo, además se vacía en caso de contener información, caso contrario si el usuario elige la operación `od`.

Una vez que se finalizó la codificación anterior, ahora se procede a cambiar el código que está dentro del evento clic del botón del formulario. A continuación se muestra dicho código.

```

194 Sub Button1Click(sender As Object, e As EventArgs)
195     If rb_id.Checked = True Or rb_od.Checked = True Then
196         If rb_id.Checked = True Then
197             If isnumeric(textBox1.Text) And isnumeric(Me.txt_capa_cromo.Text) And isnumeric(Me.txt_d1.Text) Then
198                 _width = textBox1.Text
199                 Me._d1 = Me.txt_d1.Text
200                 Me.capaCromo = Me.txt_capa_cromo.Text
201                 dimension_id = True
202                 Calcular()
203                 Close()
204             Else
205                 MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
206             End If
207         ElseIf rb_od.Checked = True Then
208             If isnumeric(textBox1.Text) And isnumeric(Me.txt_capa_cromo.Text) And isnumeric(Me.txt_d1.Text) And IsNumeric(textBox2.Text) Then
209                 _width = textBox1.Text
210                 _tiempo_cromado = textBox2.Text
211                 Me._d1 = Me.txt_d1.Text
212                 Me.capaCromo = Me.txt_capa_cromo.Text
213                 dimension_id = False
214                 Calcular()
215                 Close()
216             Else
217                 MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
218             End If
219         End If
220     Else
221         MsgBox("Por favor seleccione una opción", MsgBoxStyle.Exclamation,"Ing. de Rutas")
222     End If
223 End Sub

```

Figura 221 Código del evento clic del botón del formulario.

En la imagen anterior se puede ver que si el usuario elige cualquiera de las dos opciones se van a almacenar los valores de entrada en sus respectivas variables, lo único que cambia es el valor de la variable dimension_id ya que si el usuario elige la operación id entonces el valor de la variable es true, pero si el usuario elige la operación od entonces el valor de la variable será false.

Una vez que se realizó todo lo anterior se procede a modificar el código del método calcular() del centro de trabajo. El código del método es el siguiente:

```

139 Public Sub calcular() Implements ICentroTrabajo.calcular
140     Dim accres,acsulk,acsulf1 As Double
141     tiempo_septup = buscar_setupin(centro)
142     Dim carga,carga_total,tinas As Integer
143     Dim tam_de_carga,ciclo_por_carga As Double
144
145     If dimension_id = True Then
146         tam_de_carga = 17.5
147         tinas = 2
148         ciclo_por_carga = 5574.55
149     ElseIf dimension_id = False
150         tam_de_carga = 17.1
151         tinas = 10
152         ciclo_por_carga = (_tiempo_cromado + 919.85 + 661.48)
153     End If
154     carga = CInt(tam_de_carga / _width)
155     carga_total = (carga * tinas)
156     tiempo_maquina = Math.Round((100 * (ciclo_por_carga / 3600) / (carga_total)) * 100,3)

```

Figura 222 Método calcular del CT 9361.

Lo que se realiza es este método es que a las variables tam_de_carga, tinas, ciclo_por_carga, tienen un valor diferente para cada operación, y esos valores se obtienen dependiendo de la operación que el usuario elija. Una vez que se obtienen los valores de las variables mencionadas, se procede a calcular el valor de las variables carga y carga_total, estas variables dependen de los valores de las primeras variables. Una vez que se tienen los valores de las variables ciclo_por_carga y carga_total, se ejecuta la fórmula del tiempo estándar, la cual se

almacena en la variable tiempo_maquina. Y con estas modificaciones se terminó de actualizar este centro de trabajo.

6.3.16 CentroTrabajo9363.vb – FERROX FILL L.B.

A este centro de trabajo prácticamente no se le modificó nada, sin embargo a continuación se muestra el formulario y el método calcular() de este centro de trabajo.

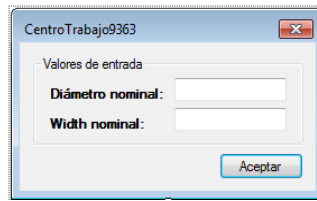


Figura 223 Formulario del CT 9363.

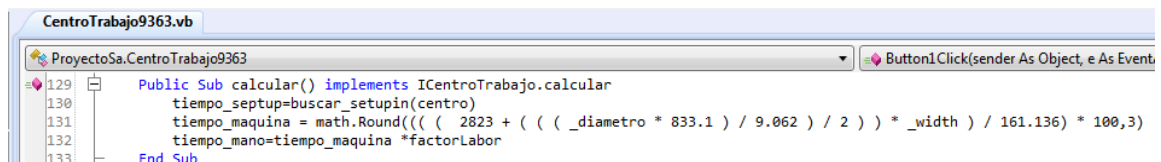


Figura 224 Método calcular() del CT 9363.

6.3.17 CentroTrabajo9364.vb – FERROX FURN L.B.

Este centro de trabajo realizaba el cálculo del tiempo de manera directa y por lo tanto no necesitaba de valores de entrada, pero ahora con la actualización es necesario que el usuario ingrese un valor para que el cálculo se pueda llevar a cabo. Entonces se creó la siguiente interfaz gráfica.

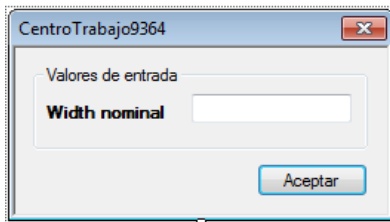


Figura 225 Formulario del CT 9364.

Después de haber creado la interfaz gráfica se declaró la variable correspondiente al valor de entrada. Luego se agregó un nuevo parámetro al constructor de la clase, para luego igualarlo con la variable que almacena el valor de entrada.

```

CentroTrabajo9364.vb
ProyectoSa.CentroTrabajo9364
100 Public Sub New(ByVal H1 As Double)
101     _width = H1
102 End Sub
103
104 '-----variables que se necesitan-----
105 Dim _width As Double
106 '-----

```

Figura 226 Parámetros que contiene el constructor de la clase.

Después el valor de entrada se agrega a la listaDatosRequeridos y por último se pasan los valores de listas, como se muestra en la siguiente imagen.

```

CentroTrabajo9364.vb
ProyectoSa.CentroTrabajo9364 listaDatosStringRequeridos
93 Public Sub New()
94     Me.InitializeComponent()
95
96     Dim widthNominalAnillo As New Dato("widthNominalLB", "Distance", "width nominal del anillo", Nothing, "width del anillo")
97     listaDatosRequeridos.Add(widthNominalAnillo)
98 End Sub

```

Figura 227 Agregando valores de entrada a ListaDatosRequeridos.

```

CentroTrabajo9364.vb
ProyectoSa.CentroTrabajo9364 listaDatosStringRequeridos
89 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
90     _width = Module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")
91 End Sub

```

Figura 228 Pasar valores de listas.

Una vez que se realizó todo el procedimiento anterior se actualizo el método calcular() perteneciente a este centro de trabajo, quedando de la siguiente forma.

```

CentroTrabajo9364.vb*
ProyectoSa.CentroTrabajo9364
122 Public Sub calcular() implements ICentroTrabajo.calcular
123     tiempo_septup=buscar_setupin(centro)
124     Dim carga_por_canastilla As Double
125     carga_por_canastilla = Math.Round(((12*5) / _width),2)
126     tiempo_maquina = Math.Round((100 * (5349.94 / 3600) / (carga_por_canastilla)) * 100,3)
127     tiempo_mano=tiempo_maquina * factorLabor
128 End Sub

```

Figura 229 Método calcular() del CT 9364.

Como se puede apreciar solo es necesario obtener el valor de la variable carga_por_canastilla, el valor de esta variable depende del valor de width que ingrese el usuario. Una vez que se tiene el valor de la variable, es posible realizar el cálculo del tiempo estándar.

Posteriormente se le agrego el evento clic al botón del formulario y dentro del método del evento se agregó el siguiente código.

```

CentroTrabajo9364.vb
ProyectoSa.CentroTrabajo9364
140 Sub Button1Click(sender As Object, e As EventArgs)
141     If IsNumeric(textBox1.Text) Then
142         _width = CDec(textBox1.Text)
143         calcular()
144         Close()
145     Else
146         MsgBox("Revisar valores", MsgBoxStyle.Critical, "Ing. de Rutas")
147     End If
148 End Sub

```

Figura 230 Evento clic del botón del formulario.

Lo que realiza el código anterior es una condición if en donde si el valor que ingresa el usuario es numérico entonces realiza el cálculo y se cierra el formulario, sin embargo si el valor no es numérico se mostrara un mensaje de error y no se realizara el cálculo.

6.3.18 CentroTrabajo9381.vb – MANGANESE PHOSPHATE L.B.

Ah este centro de trabajo no se le realizo ninguna modificación a su interfaz gráfica debido a que con los valores de entrada que cuenta el formulario son suficientes para llevar a cabo el cálculo del tiempo estándar. A continuación se muestra el formulario.

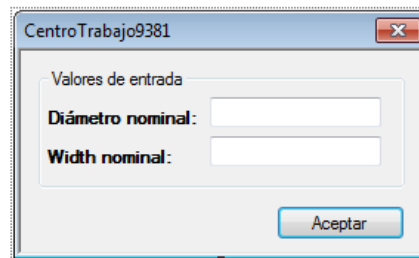


Figura 231 Formulario del CT 9381.

Como no se le hizo ninguna modificación a la interfaz gráfica, entonces no se agregaron, ni se eliminaron variables, parámetros, listaDatosRequeridos, etc. Quedando el código de la misma forma.

```
CentroTrabajo9381.vb
ProyectoSa.CentroTrabajo9381
105 Public Sub New(ByVal _width As Double, ByVal _diametro As Double)
106     widths= _width
107     diametro =_diametro
108 End Sub
109
110 Dim widths As Double
111 Dim diametro As Double
112
```

Figura 232 Constructor y parametros del CT 9381.

```
CentroTrabajo9381.vb
ProyectoSa.CentroTrabajo9381
89 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
90     diametro = module1.obtenerValorDatoDouble(lDatos,"diaNominalAnillo")
91     widths = module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")
92 End Sub
93
94 Public Sub New()
95     Me.InitializeComponent()
96
97     Dim diaNominalAnillo As New Dato("diaNominalAnillo", "Distance", "Diámetro nominal del anillo (Plano)",Nothing, "Diámetro nominal del anillo:")
98     listaDatosRequeridos.Add(diaNominalAnillo)
99
100     Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "Width nominal del anillo (Plano)",Nothing, "width nominal de anillo:")
101     listaDatosRequeridos.Add(widthNominalAnillo)
102 End Sub
103
```

Figura 233 Método pasarValoresDeListas() y listaDatosRequeridos.

Entonces lo único que se realizo fue modificar el método calcular(), dentro de este método se encuentra las operaciones matemáticas y la fórmula para obtener el

tiempo estándar. En la siguiente imagen se muestra el código completo de dicho método.

```
CentroTrabajo9381.vb
ProyectoSa.CentroTrabajo9381
calculador()

127 Public Sub calcular() implements ICentroTrabajo.calcular
128     Try
129         tiempo_septup=buscar_setupin(centro)
130         Dim bastidores,carga As Integer
131         If diametro >= 10 Then
132             bastidores = 5
133         ElseIf diametro > 7
134             bastidores = 5
135         ElseIf diametro > 5.75
136             bastidores = 7
137         Else
138             msgbox("Diametro no disponible,imposible calcular",MsgBoxStyle.Exclamation,"Ing. de Rutas")
139             ListaAlertas.Add(Name & ": El Diametro " & diametro & " no se encuentra disponible para el calculo, verificar con Ing. Industrial")
140             Exit Sub
141         End If
142         carga = CInt((bastidores * 20) / widths)
143         tiempo_maquina = Math.Round((100 * (1264.82915/3600) / (carga)) * 100,3)
144         tiempo_mano=tiempo_maquina
145         Catch er As Exception
146             Me.ListaAlertas.Add(Name & ": " & er.Message)
147         End Try
148     End Sub
```

Figura 234 Método calcular() del CT 9381

Lo primero que se realiza en el código anterior es obtener el valor del tiempo setup, después se realiza una condición if en para cierto rango de diametro que ingrese el usuario se le tiene asignado un valor para la variable bastidores, una vez que se obtiene el valor de la variable se procede a calcular el valor de la variable carga y finalmente cuando se obtiene el valor de esta última se procede a calcular la fórmula del tiempo estándar la cual está almacenada en la variable tiempo_maquina. Y para finalizar se obtiene el valor del tiempo mano el cual es igual al tiempo maquina.

6.4 Actualización de tiempos estándar – Segmentos

6.4.1 CentroTrabajo456.vb – Lapping segments

A este centro de trabajo no se le realizó ninguna modificación a su interfaz gráfica ya que con los elementos que contaba el formulario eran suficientes para llevar a cabo el cálculo del tiempo estándar.

Figura 235 Formulario del CT 456.

Como no se agregó ningún otro valor de entrada, entonces no fue necesario agregar nuevas variables ni nuevos parámetros.

Donde sí se realizaron algunas modificaciones fue en el método calcular donde se modificó la condición if y los valores que devolvía la variable tct. Quedando el código de la siguiente forma:

```
137 Public Sub calcular() implements ICentroTrabajo.calcular
138     buscar_septup()
139     If _programa = 1 Then
140         tct = 5
141     ElseIf _programa = 2 Or _programa = 4
142         tct = 20
143     ElseIf _programa = 3
144         tct = 15
145     ElseIf _programa = 5
146         tct = 25
147     ElseIf _programa = 6
148         tct = 30
149     ElseIf _programa = 7
150         tct = 35
151     ElseIf _programa = 8
152         tct = 40
153     ElseIf _programa = 9
154         tct = 45
155     ElseIf _programa = 10
156         tct = 50
157     End If
158     Dim carga_por_ciclo,ciclo_por_carga As Double
159     carga_por_ciclo = (3.5 / _width)
160     ciclo_por_carga = (tct + 43.0728262579484 + 23.2685100589874)
161     tiempo_maquina = Math.Round((100 * (ciclo_por_carga / 3600) / (carga_por_ciclo)) * 100,3)
162     tiempo_mano = tiempo_maquina
163 End Sub
```

Figura 236 Método calcular() del CT 456.

Después de realizar el cambio en la condición se declararon dos variables llamadas carga_por_ciclo y ciclo_por_carga las cuales obtienen su valor de una operación que depende de los valores que ingrese el usuario al formulario. Lo último que se realizó fue agregar la nueva fórmula para calcular el tiempo estándar, la cual se almacena en la variable tiempo_maquina, además de que depende del valor de las dos variables anteriores.

6.4.2 CentroTrabajo466.vb – Scotchbrite Segments

Para la actualización de este centro de trabajo no fue necesario modificar su interfaz gráfica ya que con valores de entrada que contaba el formulario eran necesarios para poder realizar el cálculo del tiempo estándar. A continuación se muestra el formulario.

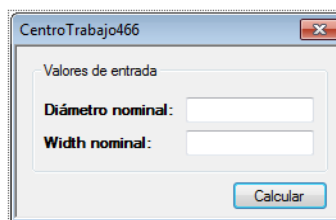
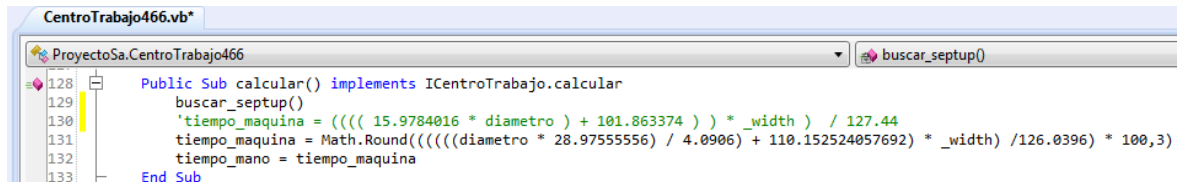


Figura 237 Formulario del CT 466.

Como no se agregaron nuevos valores de entrada, no fue necesario agregar nuevas variables, ni nuevos parámetros al constructor.

Por lo tanto lo único que se modificó fue el método calcular() del centro de trabajo, específicamente la fórmula para calcular el tiempo estándar, la cual se almacena en la variable tiempo_maquina. A continuación se muestra el código completo de dicho método.



```
128 Public Sub calcular() implements ICentroTrabajo.calcular
129     buscar_septup()
130     'tiempo_maquina = ((( ( 15.9784016 * diametro ) + 101.863374 ) * _width ) / 127.44
131     tiempo_maquina = Math.Round((((diámetro * 28.97555556) / 4.0906) + 110.152524057692) * _width) / 126.0396) * 100,3)
132     tiempo_mano = tiempo_maquina
133 End Sub
```

Figura 238 Método calcular() del CT 466.

Como se puede apreciar en la imagen anterior la variable tiempo_maquina que se encuentra en color verde era la anterior fórmula para calcular el tiempo estándar, y pues obviamente la que no está en color verde es la que actualmente está en uso

6.4.3 CentroTrabajo497.vb – Nitrurado de segmentos

Para este centro de trabajo si se tuvo que modificar la interfaz gráfica, ya que actualmente se solicitan más valores de entrada de los que se solicitaban anteriormente, a continuación se muestra el antes y el después del formulario de este centro de trabajo.

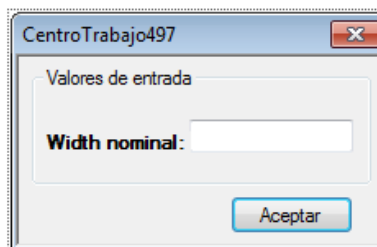


Figura 239 Formulario antes de la actualización.

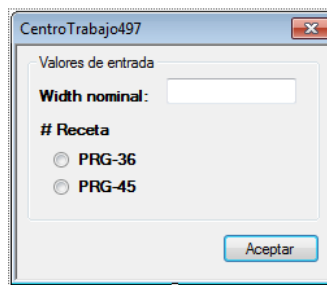


Figura 240 Formulario actual.

Como se agregaron nuevos valores de entrada, entonces se tuvieron que agregar nuevas variables y nuevos parámetros para poder llevar a cabo el cálculo del tiempo estándar. A continuación se muestran las líneas de código que se agregaron.

```

106 Public Sub New(ByVal h1 As Double,Byval _receta As Boolean)
107     _h1 = h1
108     recetaprg36 = _receta
109 End Sub
110
111 '-----variables que se nesesitan-----
112 Public _h1 As Double
113 Dim recetaprg36 As Boolean
114 '-----

```

Figura 241 Variables y parámetros del CT 497.

Como se puede apreciar en la imagen anterior solamente se agregó una sola variable de tipo booleana, esta variable se agregó debido a que se agregaron valores de entrada tipo RadioButton y estos solo devuelven valores True o False. Además también se agregó a la listaDatosBooleanosRequeridos, así como al método pasarValoresDeListas() a continuación se muestran las líneas de código que se agregaron.

```

97 Public Sub New()
98     Me.InitializeComponent()
99     Dim widthNominalAnillo As New Dato("widthNominalAnillo","Distance","Width nominal del anillo (Plano)",Nothing,"Width nominal de anillo:")
100     listaDatosRequeridos.Add(widthNominalAnillo)
101
102     Dim receta497 As New DatoBoolean("receta497","Si se activa se elige la receta PRG-36, si se desactiva se elige la receta PRG-45","PRG-36")
103     listaDatosBooleanosRequeridos.Add(receta497)
104 End Sub

```

Figura 242 listaDatosBooleanosRequeridos y listaDatosRequeridos.

```

92 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),Byval lDatosBool As List(Of DatoBoolean),Byval lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
93     _h1 = module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")
94     recetaprg36 = Module1.obtenerValorDatoBoolean(lDatosBool,"receta497")
95 End Sub

```

Figura 243 Método pasarValoresDeListas().

Lo anterior se realiza con el fin que cuando en usuario ingresa una lista de centros de trabajo para su cotización aparezcan los valores de entrada necesarios para llevar a cabo el cálculo del tiempo estándar.

Posteriormente se cambió el método calcular() con las formulas y operaciones necesarias para obtener el valor del tiempo estándar. En la siguiente imagen se muestra el código de dicho método.

```
CentroTrabajo497.vb*
ProyectoSa.CentroTrabajo497
129 Public Sub calcular() implements ICentroTrabajo.calcular
130     buscar_septup()
131     Dim t_ciclo As Integer
132     If recetaprg36 = True Then
133         t_ciclo = 47160
134     Else
135         t_ciclo = 37800
136     End If
137     tiempo_maquina = (factor1 * _h1)
138     tiempo_maquina = Math.Round((((t_ciclo + 1520) * _h1) / 56844) * 100,3)
139     tiempo_mano = tiempo_maquina
140 End Sub
```

Figura 244 Método calcular() del CT 497.

Como se puede apreciar en la imagen anterior lo primero que se realiza es una condición if en donde el valor del tiempo ciclo va a tener un valor diferente para cada receta. Además la variable tiempo_maquina en color verde, corresponde a la fórmula que anteriormente se usaba, y pues la que no está en color verde es la actual.

La última modificación que se realizó para que este centro pasara a estar actualizado, fue modificar el evento clic del botón de formulario, donde se agregó el siguiente código.

```
CentroTrabajo497.vb*
ProyectoSa.CentroTrabajo497
172 Sub Button1Click(sender As Object, e As EventArgs)
173     If IsNumeric(Me.textBox1.Text) And rb_prg36.Checked = True Then
174         _h1=Me.textBox1.Text
175         recetaprg36 = True
176         calcular()
177         Close()
178     ElseIf IsNumeric(Me.textBox1.Text) And rb_prg45.Checked = True
179         _h1 = Me.textBox1.Text
180         recetaprg36 = False
181         calcular()
182         Close()
183     Else
184         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
185     End If
186 End Sub
```

Figura 245 Evento clic del botón del formulario.

Como se puede apreciar en la imagen anterior se realiza una condición if en la dependiendo de la receta que se elija, el valor de la variable recetaprg36 va a ser false o true. Y con esto quedo actualizado el centro de trabajo.

6.4.4 CentroTrabajo498.vb – Barrell grade segments

Para este centro de trabajo no fue necesario modificar su interfaz gráfica ya que con el valor de entrada que contaba el formulario es más que suficiente para poder realizar el cálculo. En la siguiente imagen se muestra el formulario.

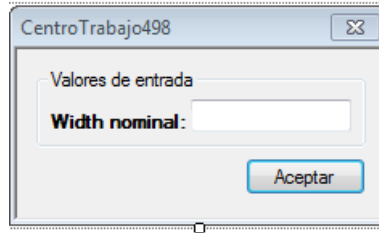


Figura 246 Formulario del CT 498.

Como no se agregó ningún valor de entrada, entonces no se tuvo que agregar ninguna variable, ni tampoco ningún parámetro al constructor.

El único cambio que se realizó fue en el método `calcular()`, en donde únicamente se agregó la fórmula para obtener el tiempo estándar actual. A continuación se muestra el código de dicho método.

```
CentroTrabajo498.vb*
ProyectoSa.CentroTrabajo498
124 Public Sub calcular() implements ICentroTrabajo.calcular
125     buscar_septup()
126     'tiempo_maquina = (factor1*espesor)/factor2
127     tiempo_maquina = Math.Round((((3.9173 * 8.6266) / 3.9173) + 33.7816) * espesor) / 160.5168) * 100,3)
128     tiempo_mano = tiempo_maquina
129 End Sub
```

Figura 247 Método `calcular()` del CT 498.

En la imagen anterior se muestra la fórmula anterior y la actual, la que se encuentra comentada es la anterior, y pues obviamente la otra es la actual. Y con esta pequeña modificación se actualizó este centro de trabajo.

6.4.5 CentroTrabajo510.vb – Bobinado de segmentos

Al igual que el centro de trabajo anterior a este tampoco se le cambió su interfaz gráfica, ya que con los valores de entrada con los que cuenta son suficientes para realizar el cálculo del tiempo estándar.

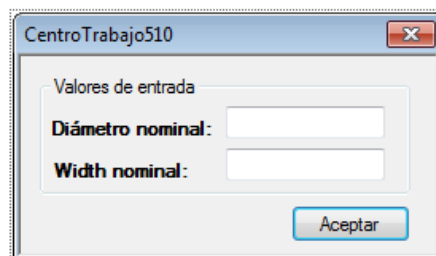


Figura 248 Formulario del CT 510.

Como no se le realizó ninguna modificación al formulario entonces no se agregaron nuevas variables, ni tampoco nuevos parámetros, así que el código quedó de igual manera.

La única modificación que se realizó se llevó a cabo en el método calcular(), específicamente en la variable tiempo_maquina donde se cambiaron las operaciones que almacenaba dicha variable. En la siguiente imagen se muestra el código de dicho método.

Figura 249 Método calcular() del CT 510.

En la imagen anterior se muestra la formula anterior y la formula actual, en donde la anterior es de color verde y pues obviamente la otra es la que actualmente está en uso. Y con este pequeño cambio quedo actualizado este centro de trabajo.

6.4.6 CentroTrabajo514.vb – Desengrase bobinado segmentos

Para este centro de trabajo no hubo necesidad de cambiar su interfaz gráfica, ya que los valores que contiene el formulario son suficientes para poder realizar el cálculo del tiempo estándar. En la siguiente imagen se muestra el formulario.

Figura 250 Formulario del CT 514.

Como no se realizó ninguna modificación, entonces el único cambio que se realizó se llevó a cabo dentro del método calcular de este centro de trabajo, fue comentar la formula anterior y después agregar la variable tiempo_maquina con las nuevas operaciones. En la siguiente imagen se muestra el código actual del método.

Figura 251 Método calcular() del CT 514.

Y con esa modificación se actualizo este centro de trabajo.

6.4.7 CentroTrabajo517.vb – Degrease segments

Al formulario de este centro de trabajo solo se le hizo una modificación, la cual fue agregar un checkbox, esto debido a que este centro de trabajo cuenta con otra operación y por este motivo se agregó dicho control. A continuación se muestra el formulario anterior y el actual.

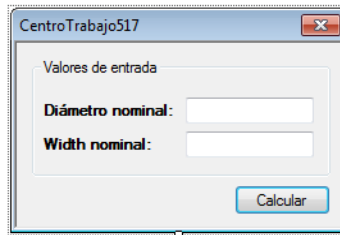


Figura 252 Formulario antes de la actualización.

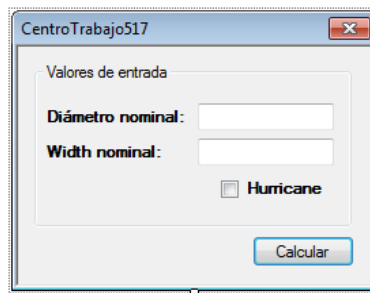


Figura 253 Formulario actual.

Como solo se agregó el checkbox entonces se le asignó una variable de tipo booleana, así como el parámetro de tipo booleano al constructor de la clase. Como se muestra a continuación.

```
CentroTrabajo517.vb
ProyectoSa.CentroTrabajo517
110 Public Sub New(ByVal h1 As Double, ByVal d1 As Double, ByVal des_seg_hurricane As Boolean)
111     _width = h1
112     _diametro = d1
113     des_hurricane = des_seg_hurricane
114 End Sub
115 '-----variables que se necesitan-----
116 Dim _width As Double
117 Dim _diametro As Double
118 Dim des_hurricane As Boolean
119 '-----'
```

Figura 254 Variables necesarias y parámetros del constructor del CT 517.

También se agregó una variable tipo DatoBoolean llamada “ope_hurricane” a la listaDatosBooleanosRequeridos y posteriormente se pasan los valores de la listas mediante el método pasarValoresDeListas.

```

CentroTrabajo517.vb
ProyectoSa.CentroTrabajo517
98 Public Sub New()
99 Me.InitializeComponent()
100 Dim diaNominalAnillo As New Dato("diaNominalAnillo", "Distance", "Diámetro nominal del anillo (Plano)", Nothing, "Diámetro nominal del anillo:")
101 listaDatosRequeridos.Add(diaNominalAnillo)
102
103 Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "Width nominal del anillo (Plano)", Nothing, "Width nominal de anillo:")
104 listaDatosRequeridos.Add(widthNominalAnillo)
105
106 Dim ope_hurricane As New DatoBoolean("ope_hurricane", "Activar para realizar operacion Hurricane", "Hurricane")
107 listaDatosBooleanosRequeridos.Add(ope_hurricane)
108 End Sub

```

Figura 255 Variables Tipo Dato y DatoBoolean.

```

CentroTrabajo517.vb
ProyectoSa.CentroTrabajo517
92 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
93 _diametro = module1.obtenerValorDatoDouble(lDatos, "diaNominalAnillo")
94 _width = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")
95 des_hurricane = Module1.obtenerValorDatoBoolean(lDatosBool, "ope_hurricane")
96 End Sub

```

Figura 256 Método pasarValoresDeListas().

Una vez que se realizó lo anterior se procedió a crear el método sacar_rieles_hurricane(), dicho método lo que realiza es obtener el valor de la variable riel dependiendo del rango en que se encuentre el diámetro que el usuario ingrese.

```

CentroTrabajo517.vb
ProyectoSa.CentroTrabajo517
171 Private Function sacar_rieles_hurricane() As Integer
172 Dim riel As Integer
173 If _diametro <= 1.8749 Then
174 riel = 0
175 ElseIf _diametro >= 1.875 And _diametro <= 4.74
176 riel = 12
177 ElseIf _diametro >= 4.75 And _diametro <= 5.5
178 riel = 5
179 ElseIf _diametro >= 5.51 And _diametro <= 6.125
180 riel = 3
181 Else
182 riel = 0
183 End If
184 Return riel
185 End Function

```

Figura 257 Método sacar_rieles_hurricane().

Una vez que se creó el método anterior, se procedió a modificar el método calcular() el cual se condiciono para obtener el tiempo estándar de las dos operaciones con las que cuenta este centro de trabajo. En la siguiente imagen se muestra el código del método.

```

CentroTrabajo517.vb
ProyectoSa.CentroTrabajo517
135 Public Sub calcular() implements ICentroTrabajo.calcular
136 tiempo_septup=buscar_setupin(centro)
137 If des_hurricane = True Then
138 riele = sacar_rieles_hurricane()
139 tiempo_maquina = Math.Round(((1440.8321 * _width) / (575.964 * riele)) * 100,3)
140 ElseIf des_hurricane = False Then
141 riele = sacar_rieles()
142 tiempo_maquina = Math.Round(((1266.27 + (riele * 7.8698)) * _width) / ((riele * 15) * 36)) * 100,3)
143 End If
144 tiempo_mano=tiempo_maquina
145 End Sub

```

Figura 258 Método calcular() del CT 517.

Lo que se realiza en el código anterior es que si el usuario elige la operación hurricane, para obtener el valor de la variable riele se utilizara el método sacar_rieles_hurricane(), además que se calculara la formula correspondiente a esta operación. Si no se selecciona se realizara la operación normal.

Después de que se modificó el método calcular(), se procedió a modificar el evento clic del botón del formulario, donde se agregaron las siguientes líneas de código.

```
CentroTrabajo517.vb
ProyectoSa.CentroTrabajo517
187 Sub Button1Click(sender As Object, e As EventArgs)
188     If chk_hurricane.Checked = True Then
189         If isnumeric(textBox1.Text) AND isnumeric(textBox2.Text) Then
190             _width =textBox2.Text
191             _diametro=textBox1.Text
192             des_hurricane = True
193             calcular()
194             Close()
195         Else
196             msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
197         End If
198     Else
199         If isnumeric(textBox1.Text) AND isnumeric(textBox2.Text) Then
200             _width =textBox2.Text
201             _diametro=textBox1.Text
202             des_hurricane = False
203             calcular()
204             Close()
205         Else
206             msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
207         End If
208     End If
209 End Sub
```

Figura 259 Evento clic del botón del formulario.

Las operaciones que se realizan en el código anterior son similares para las dos operaciones como asignarles el valor que se obtiene de los campos de texto a las variables `_width` y `_diametro` y realizar el método `calcular()` después de obtener los valores de las variables, sin embargo lo único que cambia es el valor de la variable `des_hurricane` la cual es `true` cuando se selecciona la operación `hurricane` y `false` si no se chequea la operación `hurricane`.

6.4.8 CentroTrabajo520.vb – Normalizado

A este centro de trabajo no se le agregaron, ni quitaron valores de entrada por lo tanto queda de igual manera. En la siguiente imagen se muestra dicho formulario.

Figura 260 Formulario del CT 520.

Lo que si se modifico fue la función `buscar_tiempo` donde se cambiaron los valores del tiempo ciclo de acuerdo al programa, en la siguiente imagen se muestra la función anterior y la función actual.


```

149 Private Function buscar_tiempo() As Double
150     Dim t As Double
151     If _programa = 1 Then
152         t = 3300
153     ElseIf _programa = 2
154         t = 4200
155     ElseIf _programa = 3
156         t = 5100
157     ElseIf _programa = 4
158         t = 4200
159     ElseIf _programa = 5
160         t = 3300
161     ElseIf _programa = 6
162         t = 4200
163     ElseIf _programa = 7
164         t = 3300
165     ElseIf _programa = 8
166         t = 3300
167     End If
168     Return t
169 End Function

```

Figura 261 Función buscar_tiempo() anterior.

```

149 Private Function buscar_tiempo() As Double
150     Dim t As Double
151     If _programa = 1 Then
152         t = 1920
153     ElseIf _programa = 2
154         t = 2520
155     ElseIf _programa = 3
156         t = 2520
157     ElseIf _programa = 4
158         t = 3900
159     ElseIf _programa = 5
160         t = 4200
161     ElseIf _programa = 6
162         t = 4800
163     ElseIf _programa = 7
164         t = 3900
165     ElseIf _programa = 8
166         t = 4800
167     End If
168     Return t
169 End Function

```

Figura 262 Función buscar_tiempo() actual.

Después de que se realizó la modificación a la función anterior, quedó actualizado este centro de trabajo ya que las operaciones del método calcular son iguales a las actuales, por tal motivo no se realizó modificación alguna a este método. De todos modos a continuación se muestra el código del método calcular.

```

131 Public Sub calcular() implements ICentroTrabajo.calcular
132     buscar_septup()
133     Dim t_ciclo As Double
134     t_ciclo = buscar_tiempo()
135     tiempo_maquina = math.Round(((( 5165.6 + t_ciclo ) * _h1 ) / ( 15798.24 ))*100,3)
136     tiempo_mano = tiempo_maquina
137 End Sub

```

Figura 263 Método calcular() del CT 520.

6.4.9 CentroTrabajo550.vb – Cut off vulcan

A este centro de trabajo no se le realizó ninguna modificación a su interfaz gráfica ya que con los valores de entrada que cuenta el formulario son suficientes para realizar el cálculo del tiempo estándar. A continuación se muestra el formulario de este centro de trabajo.

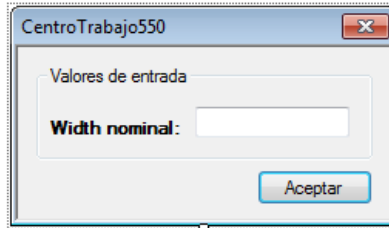


Figura 264 Formulario del CT 550.

Como no se le realizó ninguna modificación al formulario, entonces no es necesario agregar nuevas variables, ni nuevos parámetros.

La única modificación que se realizó dentro del método calcular donde lo que se realizó fue sustituir la fórmula anterior por la fórmula actual.

```
CentroTrabajo550.vb
ProyectoSa.CentroTrabajo550
126 Public Sub calcular() implements ICentroTrabajo.calcular
127     buscar_septup()
128     'tiempo_maquina= math.Round((_width*factor1)*100,3)
129     tiempo_maquina = Math.Round(((141.8959 * _width) / (113.4)) * 100,3)
130     tiempo_mano = tiempo_maquina
131 End Sub
```

Figura 265 Método calcular() del CT 550.

En la imagen anterior se muestra la fórmula anterior y la fórmula actual, donde la anterior es la variable tiempo_maquina en color verde. Y obviamente la actual es la que está en color negro.

6.4.10 CentroTrabajo555.vb – Cut off Thompson cromado

Al formulario de este centro de trabajo se le agregaron nuevos valores de entrada, los cuales son controles RadioButton y CheckBox, a continuación se muestra el formulario anterior y el actual.

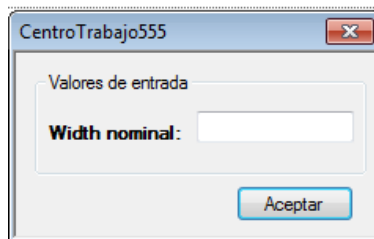


Figura 266 Formulario antes de la actualización.

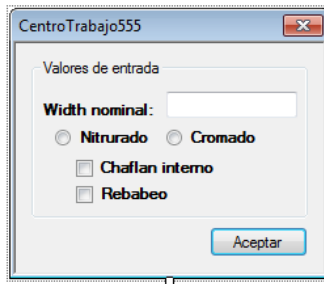


Figura 267 Formulario actual.

Como se agregaron nuevos valores de entrada, entonces se agregaron nuevas variables y nuevos parámetros. A continuación se muestran las variables y parámetros que se agregaron.

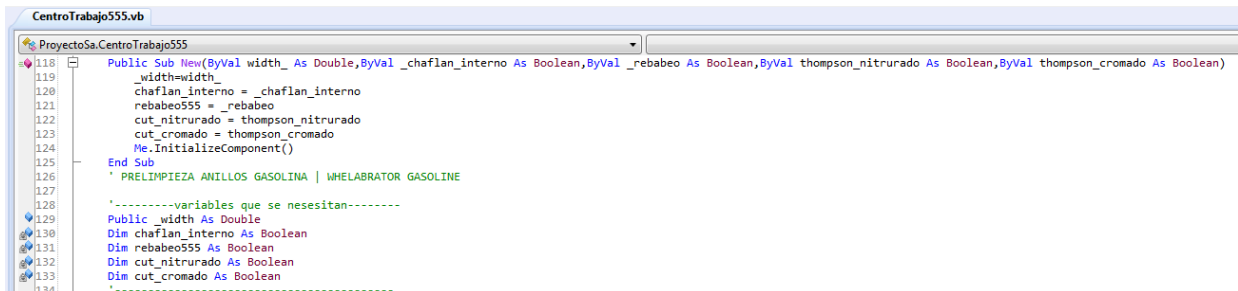


Figura 268 Variables y parámetros necesarios para calcular el tiempo estándar.

Como se puede apreciar en la imagen anterior a cada nuevo valor de entrada se le asignó una variable de tipo boolean, debido a que los controles RadioButton y CheckBox, solo devuelven dos valores que son true y false.

Después de que se realizó lo anterior, se procedió a agregar las variables tipo DatoBoolean correspondientes a cada valor de entrada y una vez declaradas se agregaron a la lista llamada listaDatosBooleanosRequeridos. En la siguiente imagen se muestra el proceso que se realizó.

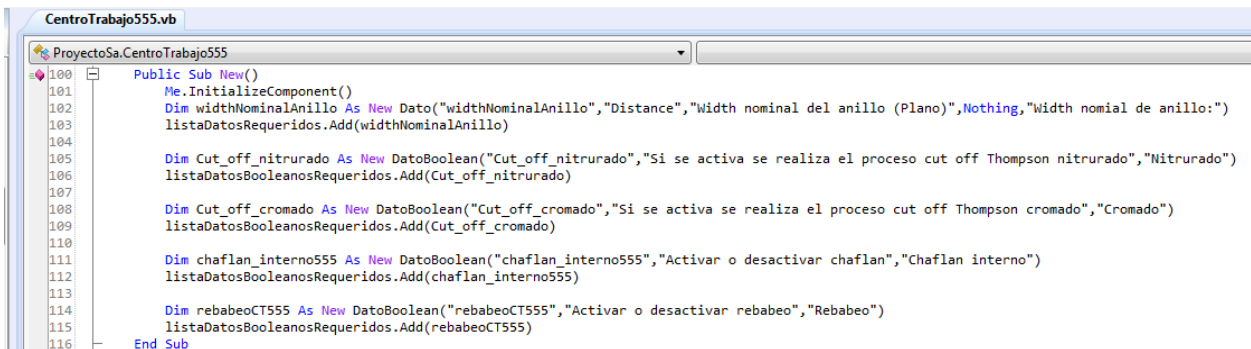


Figura 269 ListaDatosBooleanosRequeridos y ListaDatosRequeridos.

Luego se procede a pasar el valor de las listas mediante el método pasarValoresDeListas, como se muestra a continuación.

```

CentroTrabajo555.vb
ProyectoSa.CentroTrabajo555
Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
92     _width = module1.obtenerValorDatoDouble(lDatos,"widthNominalAnillo")
93     chaflan_interno = Module1.obtenerValorDatoBoolean(lDatosBool,"chaflan_interno555")
94     rebabeo555 = Module1.obtenerValorDatoBoolean(lDatosBool,"rebabeoCT555")
95     cut_nitrurado = Module1.obtenerValorDatoBoolean(lDatosBool,"Cut_off_nitrurado")
96     cut_cromado = Module1.obtenerValorDatoBoolean(lDatosBool,"Cut_off_cromado")
97
98 End Sub

```

Figura 270 Método pasarValoresDeListas().

Una vez realizado todo lo anterior se procedió a modificar el método calcular() con las formulas y operaciones necesarias para obtener el valor del tiempo estándar. A continuación se muestra el código completo de dicho método.

```

CentroTrabajo555.vb
ProyectoSa.CentroTrabajo555
Public Sub calcular() implements ICentroTrabajo.calcular
149     buscar_septup()
150     Dim tc_chaflan,tc_rebabeo As Double
151     If cut_nitrurado = True Then
152         If chaflan_interno = True And rebabeo555 = True Then
153             tc_chaflan = 4.911818
154             tc_rebabeo = 3.17
155         ElseIf chaflan_interno = True And rebabeo555 = False
156             tc_chaflan = 4.911818
157             tc_rebabeo = 0
158         ElseIf chaflan_interno = false And rebabeo555 = True
159             tc_chaflan = 0
160             tc_rebabeo = 3.17
161         ElseIf chaflan_interno = False And rebabeo555 = False
162             tc_chaflan = 0
163             tc_rebabeo = 0
164         End If
165         tiempo_maquina = Math.Round((((211.23 + 43.87 + (tc_chaflan + tc_rebabeo)) * _width) / 226.8) * 100,3)
166     ElseIf cut_cromado = True
167         If chaflan_interno = True And rebabeo555 = True Then
168             tc_chaflan = 4.911818
169             tc_rebabeo = 3.172491
170         ElseIf chaflan_interno = True And rebabeo555 = False
171             tc_chaflan = 4.911818
172             tc_rebabeo = 0
173         ElseIf chaflan_interno = false And rebabeo555 = True
174             tc_chaflan = 0
175             tc_rebabeo = 3.172491
176         ElseIf chaflan_interno = False And rebabeo555 = False
177             tc_chaflan = 0
178             tc_rebabeo = 0
179         End If
180         tiempo_maquina = Math.Round((((226.6923 + 34.27 + (tc_chaflan + tc_rebabeo)) * _width) / 226.6452) * 100,3)
181     End If
182     tiempo_mano = tiempo_maquina
183
184 End Sub

```

Figura 271 Método calcular() del CT 555.

Lo que se realiza en el código anterior es una condición en donde dependiendo de la operación que se elija se realizara el cálculo del tiempo estándar de la que se eligió.

6.4.11 CentroTrabajo558.vb – Gap side brusher

Este centro de trabajo tiene el tiempo estándar ya definido, por ello no se necesitan de valores de entrada para obtener el valor, por esta razón el cálculo se realiza de forma directa y no se necesita mostrar un formulario.

El único cambio que se realizó se llevó a cabo dentro del método calcular(), donde prácticamente se reemplazó la formula anterior por la formula actualizada. A continuación se muestra el código del método calcular().

```
CentroTrabajo558.vb
ProyectoSa.CentroTrabajo558
114 Public Sub calcular() implements ICentroTrabajo.calcular
115     tiempo_septup=buscar_setupin(centro)
116     'tiempo_maquina = (0.1980*100)
117     tiempo_maquina = Math.Round(0.197976 * 100,3)
118     tiempo_mano=tiempo_maquina
119 End Sub
```

Figura 272 Método calcular() del CT 558.

Como se puede apreciar en la imagen anterior la variable que está en color verde es la fórmula anterior y pues como se puede ver la fórmula actual es casi igual a la anterior solo se agregaron más decimales, y con esta pequeña modificación quedo actualizado este centro de trabajo.

6.4.12 CentroTrabajo716.vb – OD Chrome plate segments

Al formulario de este centro de trabajo se le agregaron dos radioButton debido a que este centro de trabajo cuenta con dos operaciones. A continuación se muestra el formulario anterior y el actual.

Figura 273 Formulario antes de la actualización.

Figura 274 Formulario actual.

Como se agregaron nuevos valores de entrada, entonces se tuvieron que declarar nuevas variables, así como también se agregaron nuevos parámetros al constructor de la clase. A continuación se muestran las variables y el constructor de la clase.

```

CentroTrabajo716.vb
ProyectoSa.CentroTrabajo716
CentroTrabajo
Public Sub New (ByVal d1 As Double, ByVal h1 As Double, ByVal espesorCromo As Double, ByVal linea_auto716 As Boolean)
    diametro = d1
    _h1 = h1
    espesorCrom = espesorCromo
    linea_auto = linea_auto716
End Sub
'CROMADO DE SEGMENTOS
'-----variables que se necesitan-----
Dim diametro As Double
Dim _h1 As Double
Dim espesorCrom As Double
Dim linea_auto As Boolean

```

Figura 275 Variables y constructor de la clase.

Después de realizar lo anterior se procedió a declarar una variable de tipo DatoBoolean correspondiente a la operación línea automática, una vez declarada se agregó a la listaDatosBooleanosRequeridos.

```

CentroTrabajo716.vb
ProyectoSa.CentroTrabajo716
New()
Public Sub New()
    Me.InitializeComponent()
    Dim diaNominalAnillo As New Dato("diaNominalAnillo", "Distance", "Diámetro nominal del anillo (Plano)", Nothing, "Diámetro nominal del anillo:")
    listaDatosRequeridos.Add(diaNominalAnillo)
    Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "Width nominal del anillo (Plano)", Nothing, "width nomial de anillo:")
    listaDatosRequeridos.Add(widthNominalAnillo)
    Dim capaCromoOD As New Dato("capaCromoOD", "Distance", "Capa de cromo en el diámetro exterior del componente", Nothing, "Capa de cromo:")
    listaDatosRequeridos.Add(capaCromoOD)
    Dim linea_automatica716 As New DatoBoolean("linea_automatica716", "Activar para realizar el calculo de la linea automatica,desactivar para realizar el calculo de la linea manual", "Línea automática")
    listaDatosBooleanosRequeridos.Add(linea_automatica716)
End Sub

```

Figura 276 ListaDatosBooleanosRequeridos.

Luego se procede a pasar el valor de las listas mediante el método llamado pasarValoresDeListas, como se muestra en la siguiente imagen.

```

CentroTrabajo716.vb
ProyectoSa.CentroTrabajo716
pasarValoresDeListas
Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
    diametro = module1.obtenerValorDatoDouble(lDatos, "diaNominalAnillo")
    _h1 = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")
    espesorCrom = MODULE1.obtenerValorDatoDouble(lDatos, "capaCromoOD")
    linea_auto = Module1.obtenerValorDatoBoolean(lDatosBool, "linea_automatica716")
End Sub

```

Figura 277 Método pasarValoresDeListas.

Después de haber realizado todo lo anterior se procedió a modificar el método calcular() agregando las nuevas operaciones y cálculos para obtener el valor del tiempo estándar.

```

CentroTrabajo716.vb
ProyectoSa.CentroTrabajo716
calcular
Public Sub calcular() Implements ICentroTrabajo.calcular
    Dim aces, unichrs, acsul, acsulfls As Double
    tiempo_septup=buscar_setupin(centro)
    buscarCiclo()
    If linea_auto = True Then
        tiempo_maquina = Math.Round((((948.8039 + t_ciclo) * _h1) / 38320.118) * 100,3)
    Else
        tiempo_maquina = Math.Round((((742.9466 + t_ciclo) * _h1) / 5474.304) * 100,3)
    End If
    tiempo_mano = tiempo_maquina

```

Figura 278 Método calcular() del CT 716.

Lo que se realiza en el código anterior es una condición en donde la fórmula del tiempo estándar va a cambiar dependiendo de la operación que el usuario elija.

Y ya por ultimo lo que se realizo fue modificar el evento clic del botón del formulario. Como se muestra a continuación.

```

CentroTrabajo716.vb
ProyectoSa.CentroTrabajo716
New()
271 Sub Button1Click(sender As Object, e As EventArgs)
272 If isnumeric(Me.textBox1.Text) and isnumeric(Me.textBox2.Text) and isnumeric(Me.textBox3.Text) And rb_linea_auto.Checked = True Or rb_linea_manual.Checked = True Then
273     Me._h1 = CDec(Me.textBox2.Text)
274     Me.diametro=CDec(Me.textBox1.Text)
275     Me.espesor Crom=CDec(Me.textBox3.Text)
276     Me.linea_auto = rb_linea_auto.Checked
277     Me.calcular()
278     Me.Close()
279 Else
280     MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
281 End If

```

Figura 279 Evento clic del botón del formulario.

Lo primero que se realiza es una condición en la que si el usuario ingresa valores numéricos a los campos de texto, además de que si elige una de las dos operaciones, entonces se almacenaran los valores que el usuario ingreso al formulario en las variables correspondientes, y después se realiza el método calcular y por último se cierra el formulario. Y con esto queda actualizado este centro de trabajo.

6.4.13 CentroTrabajo717.vb – ID Chrome plate segments

A este centro de trabajo se le tuvieron que agregar nuevos valores de entrada, debido a que con los valores de entrada anteriores ya no son suficientes para poder obtener el valor del tiempo estándar. A continuación se muestra el formulario anterior y el formulario actual.

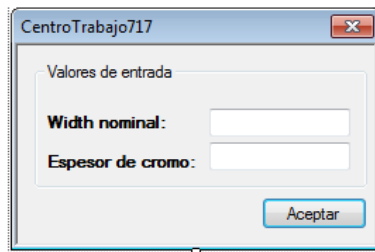


Figura 280 Formulario antes de la actualización.

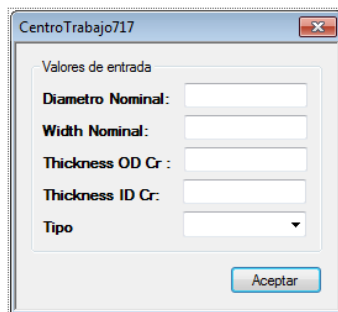


Figura 281 Formulario actual.

Después de realizar la modificación del formulario de este centro de trabajo, se les asigno una variable a cada nuevo valor de entrada, así como también se agregaron los parámetros al constructor de la clase. A continuación se muestran las variables y los parámetros.

```

120 Public Sub New (ByVal h1 As Double, ByVal espesorCromo As Double, ByVal d1 As Double, ByVal thick0D As Double, ByVal componente As String)
121     _h1 = h1
122     _espesorcrom = espesorCromo
123     _DiamN = d1
124     thickness0D = thick0D
125     com_tipo = componente
126 End Sub
127
128 'CROMADO DE SEGMENTOS
129 -----variables que se necesitan-----
130 dim _h1 As Double
131 Dim _espesorcrom As Double
132 -----Actualizado 13/07/16 J.Chavez-----
133 Dim thickness0D As Double
134 Dim _DiamN As Double
135 Dim com_tipo As String
136

```

Figura 282 Variables y parámetros del CT 717.

Posteriormente se agregaron las variables de tipo Dato y DatoString correspondientes a los valores de entrada, una vez declaradas se agregaron a la lista correspondiente. Las variables tipo DatoString se agregan a la listaDatosStringRequeridos y las tipo Dato a la listaDatosRequeridos.

Luego se pasan los valores de las listas mediante el método pasarValoresDeListas. A continuación se muestra el código del método.

```

92 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
93     _h1 = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")
94     _espesorcrom = module1.obtenerValorDatoDouble(lDatos, "capaCromoID")
95     -----actualizado-----
96     thickness0D = module1.obtenerValorDatoDouble(lDatos, "thick0D")
97     _DiamN = module1.obtenerValorDatoDouble(lDatos, "DiametroAnillo")
98     com_tipo = Module1.obtenerValorDatoString(lDatosString, "tipo717")
99
100 End Sub

```

Figura 283 Metodo pasarValoresDeListas().

Una vez que se realizó todo lo anterior se procedió a modificar el método calcular() en donde se agregaron las nuevas fórmulas para obtener el valor del tiempo estándar. En la siguiente imagen se muestra el código del método calcular().

```

156 Public Sub calcular() Implements ICentroTrabajo.calcular
157     Dim accresc, acsulss, acsulflss As Double
158     tiempo_septup=buscar_setupin(centro)
159     'busca()
160     Dim t_ciclo As Double
161     If com_tipo = "BBM" Then
162         t_ciclo =1505.88
163     ElseIf com_tipo = "MBJ"
164         t_ciclo = 4011
165     End If
166     'tiempo_maquina = (factor1+TCicloTina)/((factor2*(factor3*(factor4/_h1))))
167     'tiempo_maquina = tiempo_maquina * 100
168     tiempo_maquina = Math.Round(((t_ciclo + 1681.54800727789) / (36 * 1536)) * 100,3)
169     tiempo_mano=tiempo_maquina

```

Figura 284 Método calcular() del CT 717.

Lo primero que se realiza en el código anterior es una condición en la que el valor de la variable t_ciclo varía dependiendo del tipo que el usuario elija. Una vez que se obtiene el valor de dicha variable, se procede a calcular la fórmula del tiempo estándar la cual esta almacena en la variable tiempo_maquina.

Nota: la parte que se encuentra en color verde son las formulas anteriores.

Y por último se modificó el evento clic del botón del formulario, asignándole a cada variable el control del formulario que le corresponde. En la siguiente imagen se muestra el código de dicho evento.

```

CentroTrabajo717.vb
ProyectoSa.CentroTrabajo717 Button1Click(sender As Object, e As EventArgs)
225 Sub Button1Click(sender As Object, e As EventArgs)
226 If isnumeric(Me.textBox1.Text) And isnumeric(Me.textBox3.Text) And IsNumeric(Me.textBox2.Text) And IsNumeric(Me.textBox4.Text) And cbo_componente.Text <> "" Then
227     Me._h1=Me.textBox1.Text
228     Me._espesorcrom=Me.textBox3.Text
229     Me._Diam=Me.textBox2.Text
230     Me.thickness00=Me.textBox4.Text
231     Me.com_tipo = cbo_componente.Text
232     Me.calcular
233     Me.Close
234 Else
235     msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
236 End If
237 End Sub

```

Figura 285 Evento clic del botón del formulario.

6.4.14 CentroTrabajo720.vb – Blackening

Al formulario de este centro de trabajo se le agregaron nuevos valores de entrada, porque con los valores que contaba no eran suficientes para poder llevar a cabo el cálculo del tiempo estándar. En la siguiente imagen se muestra el formulario anterior y el formulario actual.

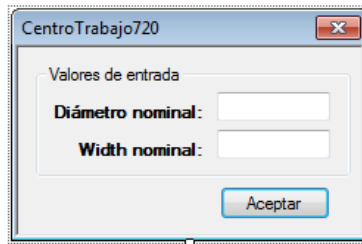


Figura 286 Formulario antes de la actualización.

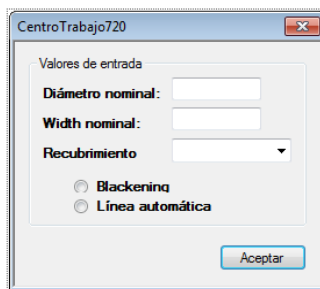


Figura 287 Formulario actual.

Después de que se modificó el formulario, se agregaron las variables correspondientes a cada valor de entrada, así como también se agregaron los nuevos parámetros al constructor de la clase. A continuación se muestran todas las variables y el constructor de la clase con todos sus parámetros.

```

112 Public Sub New(ByVal h1 As Double, ByVal d1 As Double, ByVal recubrimiento720 As String, ByVal blackening720 As Boolean)
113     _h1 = h1
114     _d1 = d1
115     recubrimiento = recubrimiento720
116     blackening = blackening720
117 End Sub
118 '-----variables que se nesecitan-----
119 Dim _h1 As Double
120 Dim _d1 As Double
121 Dim recubrimiento As String
122 Dim blackening As Boolean
123 '-----

```

Figura 288 Variables y parámetros del CT 720.

Después lo que se realizó fue agregar las variables tipo DatoString y DatoBoolean correspondientes a cada valor de entrada, y una vez que se declararon las variables se agregan a la lista que le corresponde, las tipo DatoString se agregan a la listaDatosStringRequeridos y las que son DatoBoolean se agregan a la listaDatosBooleanosRequeridos.

```

97 Public Sub New()
98     Me.InitializeComponent()
99     Dim diaNominalAnillo As New Dato("diaNominalAnillo", "Distance", "Diámetro nominal del anillo (Plano)", Nothing, "Diámetro nominal del anillo:")
100     listaDatosRequeridos.Add(diaNominalAnillo)
101
102     Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "Width nominal del anillo (Plano)", Nothing, "Width nominal de anillo:")
103     listaDatosRequeridos.Add(widthNominalAnillo)
104
105     Dim recubrimientoblack720 As New DatoString("recubrimientoblack720", "tipo de recubrimiento a usar (Cromo o Nitruado)", "Recubrimiento")
106     listaDatosStringRequeridos.Add(recubrimientoblack720)
107
108     Dim opeblackening720 As New DatoBoolean("blackening720", "Si se activa se realiza la operación blackening, si se desactiva se realiza la operación línea automática", "Blackening")
109     listaDatosBooleanosRequeridos.Add(opeblackening720)
110 End Sub

```

Figura 289 Listas de datos requeridos.

Posteriormente se pasan los valores de las listas mediante el método pasarValoresDeListas() como se muestra en la siguiente imagen.

```

91
92 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
93     _d1 = module1.obtenerValorDatoDouble(lDatos, "diaNominalAnillo")
94     _h1 = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")
95     recubrimiento = Module1.obtenerValorDatoString(lDatosString, "recubrimientoblack720")
96     blackening = Module1.obtenerValorDatoBoolean(lDatosBool, "blackening720")
97 End Sub

```

Figura 290 Método pasarValoresDeListas del CT 720.

Una vez que se realizó todo lo anterior se creó una función llamada sacar_rieles() lo que se realiza en esta función es una condición en la que dependiendo del rango en que se encuentre el valor de diámetro que ingrese el usuario, la variable r va a tener un valor específico para cada rango.

```

CentroTrabajo720.vb
ProyectoSa.CentroTrabajo720
196 Private Function sacar_rieles() As Integer
197     Dim r As Integer
198     If _d1 <= 1.874 Then
199         r = 0
200     ElseIf _d1 >= 1.875 And _d1 <= 4.25
201         r = 12
202     ElseIf _d1 >= 4.26 And _d1 <= 5.25
203         r = 6
204     ElseIf _d1 >= 5.26 And _d1 <= 6.125
205         r = 4
206     Else
207         r = 0
208     End If
209     Return r
210 End Function

```

Figura 291 Función sacar_rieles().

Después de que se creó la función se procedió a modificar el método calcular() se agregaron las dos fórmulas de las operaciones de este centro de trabajo. A continuación se muestra el código del método.

```

CentroTrabajo720.vb
ProyectoSa.CentroTrabajo720
141 Public Sub calcular() implements ICentroTrabajo.calcular
142     buscar_septup()
143     Dim rack,rieles As Integer
144     Dim tc,ac As Double
145     If blackening = True Then
146         If _d1 <= 4.5 Then
147             rack = 12
148         Else
149             rack = 6
150         End If
151
152         If recubrimiento = "Cromo" Then
153             tc = 510.25
154             ac = 510.25
155         ElseIf recubrimiento = "Nitrurado"
156             tc = 471.56
157             ac = 471.46
158         End If
159         tiempo_maquina = Math.Round(((600.5064 + tc + ac) * _h1) / (432 * rack)) * 100,3)
160     Else
161         rieles = sacar_rieles()
162         tiempo_maquina = Math.Round(((734.22 * _h1) / ((36 * (6.33 * rieles)))) * 100,3)
163     End If
164     tiempo_mano = tiempo_maquina
165 End Sub

```

Figura 292 Método calcular() del CT 720.

Lo primero que se realiza dentro del método es una condición en la cual si el diámetro es menor a 4.5 entonces el valor de la variable rack es 12, si no se cumple esa condición el valor de rack es 6, después se realiza otra condición en la que el valor de las variables tc y ac varía dependiendo del recubrimiento que elija el usuario, cabe mencionar estas dos condiciones anteriores se realizan únicamente cuando se selecciona la operación blackening.

Cuando el usuario elige la operación línea automática lo único que se debe obtener es la cantidad de rieles a usar y dicho valor se obtiene de la función sacar_rieles que mencione anteriormente, una vez obtenido el valor se ejecuta la fórmula del tiempo estándar correspondiente a esta operación.

Lo último que se realizó fue la modificación del evento clic del botón del formulario, donde se agregaron las siguientes líneas de código.

```

CentroTrabajo720.vb
ProyectoSa.CentroTrabajo720
212 Sub Button1Click(sender As Object, e As EventArgs)
213     If rb_black.Checked Or rb_linea.Checked Then
214         If rb_black.Checked Then
215             If isnumeric(textBox1.Text) and isnumeric(textBox2.Text) And comboBox1.Text <> "" Then
216                 _h1 = textBox1.Text
217                 _d1 = Textbox2.Text
218                 recubrimiento = comboBox1.Text
219                 blackening = True
220                 calcular()
221                 close()
222             Else
223                 msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
224             End If
225         ElseIf rb_linea.Checked
226             If isnumeric(textBox1.Text) and isnumeric(textBox2.Text) Then
227                 _h1 = textBox1.Text
228                 _d1 = Textbox2.Text
229                 blackening = False
230                 calcular()
231                 close()
232             Else
233                 msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
234             End If
235         End If
236     Else
237         MsgBox("Elija una de las dos opciones", MsgBoxStyle.Exclamation,"Ing. de Rutas")
238     End If
239 End Sub

```

Figura 293 Evento clic del botón del formulario.

Lo que se realiza en el código anterior, son varias condiciones las cuales ayudan a que el usuario no cometa errores al momento que ingresa los valores al formulario. Y con esto queda actualizado el centro de trabajo.

6.4.15 CentroTrabajo831.vb – Final insp. Segments

A este centro de trabajo no se le realizo ninguna modificación a su interfaz gráfica, esto debido a que el formulario contiene los valores de entrada necesarios para llevar a cabo el cálculo del tiempo estándar. A continuación se muestra el formulario de este centro de trabajo.

Figura 294 Formulario del CT 831.

Entonces lo único que se modifico fue el método calcular(), donde se agregaron nuevas variables y nuevas condiciones. En la siguiente imagen se muestra el código completo del método calcular().

```

CentroTrabajo831.vb
ProyectoSa.CentroTrabajo831
New()
142 Public Sub calcular() implements ICentroTrabajo.calcular
143 tiempo_septup=buscar_setupin(centro)
144 Dim t_ciclo_aceite,t_ciclo_pintura As Double
145 If _ban_aceite = True Then
146     t_ciclo_aceite = 0.509586538461538
147 End If
148 If _ban_pintura = True Then
149     t_ciclo_pintura = 4.843699791666667
150 End If
151 tiempo_mano = math.Round(((( 55.27 + t_ciclo_aceite + ( 9 * _noFranjas ) ) / 36 ) / ( 4 / espesor ))*100,3)
152 tiempo_maquina = Math.Round(((63.12 + (t_ciclo_pintura * _noFranjas) + (t_ciclo_aceite)) * (espesor) / (144)) * 100,3)
153 tiempo_mano = tiempo_maquina
154 End Sub

```

Figura 295 Método calcular() del CT 831.

Como se puede apreciar en la imagen anterior se declararon dos variables las cuales son t_ciclo_aceite y t_ciclo_pintura las cuales sirven para almacenar los tiempos ciclo de aceite y pintura, despues se realizan dos condiciones en donde si el usuario elige aceite el valor de la variable t_ciclo_aceite será de 0.509586538461538, en cambio si el usuario no elige aceite el valor de la variable pasara a ser cero. Esto ocurre de igual forma con pintura la única diferencia es que el valor de la variable t_ciclo_pintura será de 4.843699791666667. Una vez obtenidos los valores de esa variables se procede a ejecutar la fórmula del tiempo estándar, la cual esta almacenada en la variable tiempo_maquina. Cabe mencionar que la variable tiempo_mano que está en color verde es la fórmula anterior.

Al evento clic tampoco se le hicieron modificaciones, porque normalmente se modifica cuando se agregan nuevos valores de entrada, y en este caso no se realizó ninguna modificación al formulario de este centro de trabajo.

6.5 Actualización de tiempos estándar – FRANKLIN

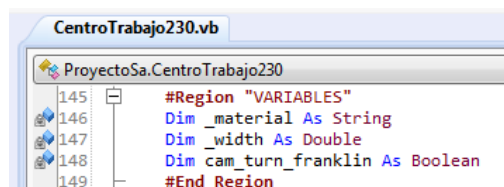
6.5.1 CentroTrabajo230.vb – ROUGH CAM TURN (FRANKLIN)

Este centro de trabajo pertenece a anillos, sin embargo cuenta con la operación Franklin también, los valores de entrada son los mismos, pero se agregó un CheckBox para poder seleccionar la operación que corresponde a Franklin. En la siguiente imagen se muestra el formulario actual.

The image shows a Windows form titled 'CentroTrabajo230'. It has a 'Valores de entrada' section with two text boxes: 'Width nominal:' and 'Material:'. Below these is a checkbox labeled 'Rough cam turn (Franklin)'. At the bottom right is an 'Aceptar' button.

Figura 296 Formulario actual del CT 230.

Como únicamente se agregó el CheckBox, entonces solo se agregó una variable de tipo boolean la cual almacenara el valor que devuelva el CheckBox ya se true o false.



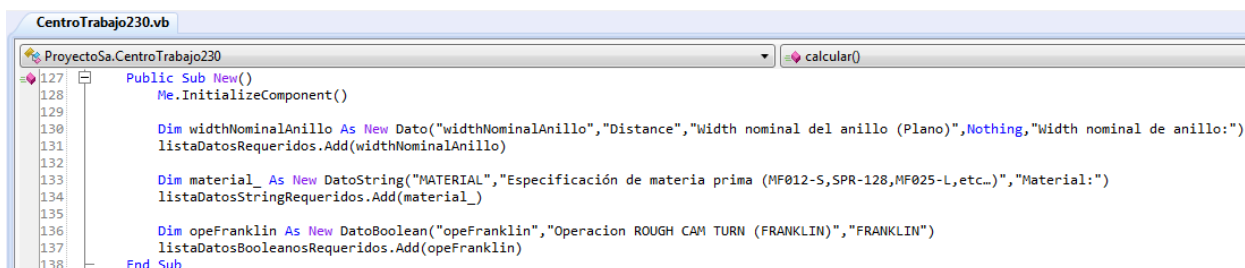
```

145 #Region "VARIABLES"
146 Dim _material As String
147 Dim _width As Double
148 Dim cam_turn_franklin As Boolean
149 #End Region

```

Figura 297 Variables del CT 230.

Una vez agregada la variable, se agregó otra variable pero esta de tipo DatoBoolean, llamada opeFranklin, una vez declarada se agregó a la listaDatosBooleanos Requeridos.



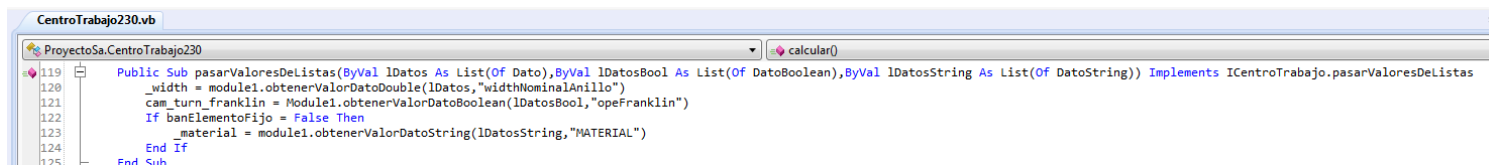
```

127 Public Sub New()
128     Me.InitializeComponent()
129
130     Dim widthNominalAnillo As New Dato("widthNominalAnillo", "Distance", "Width nominal del anillo (Plano)", Nothing, "width nominal de anillo:")
131     listaDatosRequeridos.Add(widthNominalAnillo)
132
133     Dim material_ As New DatoString("MATERIAL", "Especificación de materia prima (MF012-S, SPR-128, MF025-L, etc...)", "Material:")
134     listaDatosStringRequeridos.Add(material_)
135
136     Dim opeFranklin As New DatoBoolean("opeFranklin", "Operacion ROUGH CAM TURN (FRANKLIN)", "FRANKLIN")
137     listaDatosBooleanosRequeridos.Add(opeFranklin)
138 End Sub

```

Figura 298 Variables tipo Dato, DatoBoolean y DatoString.

Luego se debe de pasar el valor de las listas mediante el método pasarValoresDeListas(), en la siguiente imagen se muestra el procedimiento.



```

119 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato), ByVal lDatosBool As List(Of DatoBoolean), ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
120     _width = module1.obtenerValorDatoDouble(lDatos, "widthNominalAnillo")
121     cam_turn_franklin = Module1.obtenerValorDatoBoolean(lDatosBool, "opeFranklin")
122     If banElementoFijo = False Then
123         _material = module1.obtenerValorDatoString(lDatosString, "MATERIAL")
124     End If
125 End Sub

```

Figura 299 Método pasarValoresDeListas().

Después de haber realizado todo lo anterior se procedió a modificar el método calcular(), donde únicamente se agregó una condición la cual consiste en que si el usuario elige la operación franklin se realizara el cálculo del tiempo estándar de dicha operación, pero si el usuario no la elige se realizara el cálculo correspondiente a la operación de anillos. A continuación se muestra el código del método calcular().

```

CentroTrabajo230.vb
ProyectoSa.CentroTrabajo230
calculador()
165 Public Sub calcular() Implements ICentroTrabajo.calcular
166     buscar_septup()
167     Dim t_ciclo As Double
168     If banElementoFijo = False Then
169         tipo = Module1.sacar_tipo_material(_material)
170     End If
171     If tipo = "HIERRO GRIS" Or tipo = "HIERRO GRIS CENTRIFUGADO" Then
172         t_ciclo = 60
173     ElseIf tipo = "HIERRO GRIS INTERMEDIO"
174         t_ciclo = 60
175     ElseIf tipo = "HIERRO GRIS ALTO MODULO"
176         t_ciclo = 210
177     ElseIf tipo = "HIERRO DUCTIL"
178         t_ciclo = 120
179     Else
180         msgbox("Material " & tipo & " no disponible para el C.C 230, imposible calcular",MsgBoxStyle.Critical,"Ing. de Rutas")
181     End If
182     If cam_turn_franklin = True Then
183         tiempo_maquina = Math.Round(((width * (t_ciclo + 52.9971181818182)) / 356.4) * 100,3)
184     Else
185         tiempo_maquina = Math.Round(((width * (t_ciclo + 34.87324)) / 356.4) * 100,3)
186     End If
187     tiempo_mano = tiempo_maquina
188 End Sub

```

Figura 300 Método calcular() del CT 230.

Y finalmente se modificó el evento clic del botón del formulario donde únicamente se asignó la variable cam_turn_franklin al CheckBox del formulario, esto para almacenar los valores que devuelve el CheckBox. Y con esto se agregó la operación Franklin al centro de trabajo 230.

6.5.2 CentroTrabajo2000.vb – FIRST ROUGH GRIND (FRANKLIN)

En este centro de trabajo si hubo un cambio en los valores de entrada, ya que anteriormente se solicitaba el diámetro para poder realizar el cálculo, pero ahora con la actualización se solicita al usuario que ingrese un número de RPM. En la siguiente imagen se muestra el formulario anterior y el formulario actual.

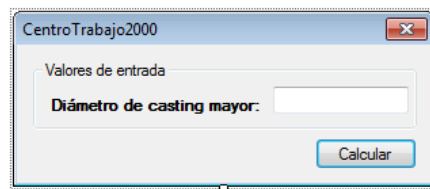


Figura 301 Formulario antes de la actualización.

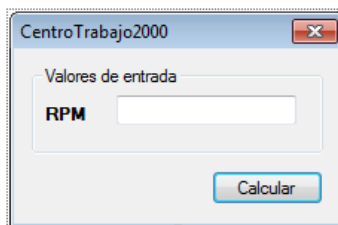


Figura 302 Formulario actual.

Como se cambió el valor de entrada del formulario, entonces se tuvieron que agregar nuevas variables, así como nuevos parámetros al constructor de la clase.

```

CentroTrabajo2000.vb
ProyectoSa.CentroTrabajo2000
108 Public Sub New(ByVal RPM As Double)
109     _rpm = RPM
110 End Sub
111
112 '-----Valores que se requieren-----
113 Dim _rpm As Double
114

```

Figura 303 Variables y parámetros del CT 2000.

Después se agregó una variable de tipo Dato llamada rpm2000 y una vez que se declaró se agregó a la listaDatosRequeridos.

```

CentroTrabajo2000.vb
ProyectoSa.CentroTrabajo2000
93 Public Sub New()
94     Me.InitializeComponent()
95
96     Dim rpm2000 As New Dato("rpm2000","Cantidad","RPM en la operación FIRST ROUGH GRIND FRANKLIN",Nothing,"RPM")
97     listaDatosRequeridos.Add(rpm2000)
98 End Sub

```

Figura 304 Variable tipo Dato y listaDatosRequeridos.

Posteriormente se pasaron los valores de las listas mediante el método pasarValoresDeListas(). A continuación se muestra el código de este método.

```

CentroTrabajo2000.vb
ProyectoSa.CentroTrabajo2000
89 Public Sub pasarValoresDeListas(ByVal lDatos As List(Of Dato),ByVal lDatosBool As List(Of DatoBoolean),ByVal lDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
90     _rpm = Module1.obtenerValorDatoDouble(lDatos,"rpm2000")
91 End Sub

```

Figura 305 Método pasarValoresDeListas del CT 2000.

Y una vez que se realizó todo lo anterior se procedió a modificar el método calcular(), que es donde se almacena la fórmula para calcular el tiempo estándar. En la siguiente imagen se muestra el código completo del método calcular.

```

CentroTrabajo2000.vb
ProyectoSa.CentroTrabajo2000
136 Public Sub calcular() implements ICentroTrabajo.calcular
137     tiempo_septup = buscar_setupin(centro)
138     tiempo_maquina = math.Round((0.7975 / ( ( ( 37 ) / _diaCastingMayor ) * 3.78 )) * 100,3)
139     tiempo_maquina = Math.Round(((3.42 / _rpm) + 0.0828769426336375) / 36) * 100,3)
140     tiempo_mano = tiempoMaquina * factorLabor
141 End Sub

```

Figura 306 Método calcular() del CT 2000.

Como se puede ver en la imagen anterior el unico cambio que se realizo fue en las operaciones que almacena la variable tiempo_maquina donde se remplazaron las anteriores por la nuevas. La variable tiempo_maquina en color verde era la formula anterior.

La actualizacion de este centro de trabajo se finalizo con la modificacion del evento clic del boton del formulario, donde se agregaron las siguientes lineas de codigo.


```
CentroTrabajo2000.vb
ProyectoSa.CentroTrabajo2000
126 Sub Btn_calcularClick(sender As Object, e As EventArgs)
127     If IsNumeric(Me.textBox1.Text) Then
128         _rpm = CDec(textBox1.Text)
129         calcular()
130         Close()
131     Else
132         MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
133     End If
134 End Sub
```

Figura 307 Evento clic del botón del formulario.

Lo que realiza el código anterior es que si el usuario ingresa un valor numérico al campo de texto el valor se almacena en la variable `_rpm` y se ejecutara el método `calcular()`, y por último se cerrara el formulario.

6.5.3 CentroTrabajo2060.vb – SPLITTER CUFF (FRANKLIN)

Al formulario de este centro de trabajo se le agregaron nuevos valores de entrada, debido a que el formulario solo contaba con un valor de entrada, pero ahora son necesarios más para poder llevar a cabo el cálculo del tiempo estándar. En la siguiente imagen se muestra el formulario anterior y el formulario actual.

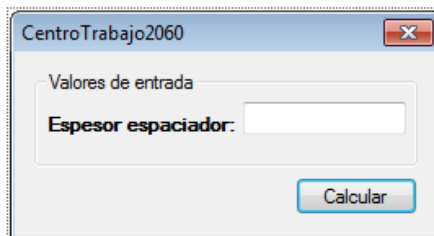


Figura 308 Formulario antes de la actualización.

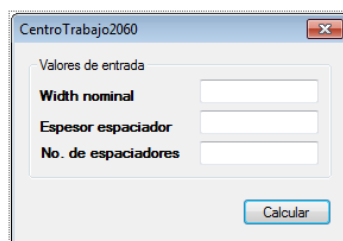


Figura 309 Formulario actual.

Como se agregaron nuevos valores de entrada, entonces se tienen que agregar nuevas variables y nuevos parámetros al constructor de la clase.

```

CentroTrabajo2060.vb
ProyectoSa.CentroTrabajo2060
Public Sub New(ByVal espesorEspaciador As Double, ByVal H1 As Double, ByVal Num_espaciador As Integer)
    _espesorEspaciador = espesorEspaciador
    _width = H1
    No_espaciadores = Num_espaciador
End Sub
'-----Valores que se requieren-----
Dim _width As Double
Dim _espesorEspaciador As Double
Dim No_espaciadores As Integer
'-----

```

Figura 310 Variables y parámetros necesarios del CT 2060.

Además de agregar las variables y los parámetros se agregaron dos variables de tipo Dato las cuales almacenaran los valores que el usuario ingrese al formulario wpf, este formulario aparece cuando el usuario ingresa el número de centro de trabajo a la lista de centros de trabajo. Una vez declarada se agrega a la lista que le corresponde la cual es listaDatosRequeridos.

```

CentroTrabajo2060.vb
ProyectoSa.CentroTrabajo2060
Public Sub New()
    Me.InitializeComponent()
    Dim espesorEspaciador2060 As New Dato("espesorEspaciador2060", "Distance", "Espesor espaciador en operación SPLITTER CUFFS (FRANKLIN)", Nothing, "Width espaciador splitter:")
    listaDatosRequeridos.Add(espesorEspaciador2060)
    Dim widthNominal2060 As New Dato("widthNominal2060", "Distance", "Width nominal del anillo(Franklin)", Nothing, "Width nominal")
    listaDatosRequeridos.Add(widthNominal2060)
    Dim noEspaciadores2060 As New Dato("noEspaciadores2060", "Cantidad", "Numero de espaciadores en operación SPLITTER CUFFS (FRANKLIN)", Nothing, "No. de espaciadores")
    listaDatosRequeridos.Add(noEspaciadores2060)
End Sub

```

Figura 311 Constructor donde están las variables tipo Dato.

Posteriormente se pasan los valores de la listaDatosRequeridos, mediante el método pasarValoresDeListas, como se muestra a continuación.

Una vez que se realizó todo lo anterior se procedió a modificar el método calcular() donde únicamente se cambió la formula anterior por la formula actual, dicha fórmula se almacena en la variable tiempo_maquina. Como se puede ver en la siguiente imagen la variable que se encuentra en color verde es la fórmula anterior y pues obviamente la otra es la que actualmente está en funcionamiento.

```

CentroTrabajo2060.vb
ProyectoSa.CentroTrabajo2060
Public Sub calcular() implements ICentroTrabajo.calcular
    tiempo_septup = buscar_setupin(centro)
    'tiempo_maquina = math.Round(( 4.37694 / ( ( 7.548 / ( _espesorEspaciador + 0.031 ) ) - 3 ) ) * 100,3)
    tiempo_maquina = Math.Round(((( _width ) * ( 99.8379971681416 )) / (36 * (( _espesorEspaciador * No_espaciadores) + 0.465))) * 100,3)
    tiempo_mano = tiempoMaquina
End Sub

```

Figura 312 Método calcular() del CT 2060.

La actualización de este centro de trabajo se finalizó con la modificación del evento clic del botón del formulario, donde se agregaron las siguientes líneas de código.

```

137 Sub Btn_calcularClick(sender As Object, e As EventArgs)
138     If IsNumeric(textBox1.Text) And IsNumeric(txt_h1.Text) And IsNumeric(txt_no_spacers.Text) Then
139         _width = CDec(txt_h1.Text)
140         _espesorEspaciador = CDec(textBox1.Text)
141         No_espaciadores = CInt(txt_no_spacers.Text)
142         calcular()
143         Close()
144     Else
145         MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
146     End If
147 End Sub

```

Figura 313 Evento clic del botón del formulario.

Únicamente lo único que se realizó fue asignarle a las variables `_width` y `No_espaciadores` su correspondiente campo de texto. Y con esto se actualizó este centro de trabajo.

6.5.4 CentroTrabajo2195.vb – FINISH MILL (FRANKLIN)

Para la actualización de este centro de trabajo no fue necesario realizar ningún cambio al formulario de este centro de trabajo, debido a que dicho formulario cuenta con los valores de entrada necesarios para realizar el cálculo del tiempo estándar. Sin embargo se muestra el formulario en la siguiente imagen.

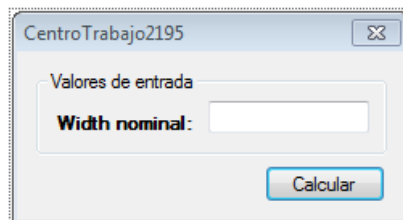


Figura 314 Formulario del CT 2195.

Como no se realizó ningún cambio al formulario entonces no fue necesario agregar nuevas variables ni nuevos parámetros. El único cambio que se realizó fue en el método `calcular()` donde únicamente se reemplazó la fórmula anterior por la actual. Como se puede ver en la siguiente imagen la variable `tiempo_maquina` que se encuentra en color verde es la fórmula anterior.

```

136 Public Sub calcular() implements ICentroTrabajo.calcular
137     tiempo_septup = buscar_setupin(centro)
138     'tiempo_maquina = math.Round((( ( 61.45 ) / 36 ) / ( 5.5 / _h1 ))*100,3)
139     tiempo_maquina = Math.Round(((34.8739897944732 * _h1) / 207) * 100,3)
140     tiempo_mano = tiempoMaquina
141 End Sub

```

Figura 315 Método calcular() del CT 2195.

6.5.5 CentroTrabajo2230.vb – DUPLEX (FRANKLIN)

Para este centro de trabajo no fue necesario realizar ninguna modificación debido a que los valores de entrada que contiene el formulario son suficientes para poder realizar el cálculo del tiempo estándar. En la siguiente imagen se muestra el formulario de este centro de trabajo.

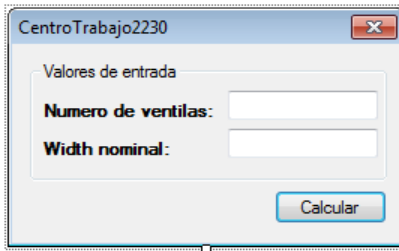


Figura 316 Formulario del CT 2230.

La única modificación que se realizó se llevó a cabo en el método calcular(), donde se sustituyó la formula anterior por la actual. Como se puede ver en la siguiente imagen la formula anterior corresponde a la variable tiempo_maquina en color verde.

```
CentroTrabajo2230.vb
ProyectoSa.CentroTrabajo2230
public Sub calcular() implements ICentroTrabajo.calcular
142: tiempo_septup = buscar_setupin(centro)
143: 'tiempo_maquina = math.Round((( 46.52 + (( 40.4 * _NoVentilas ) / 8 ) ) / ( 36 * (( 2.1 / _h1 ) - 2 ) ) ) * 100,3)
144: tiempo_maquina = Math.Round(((45.5086293886231 + ((40.4 * _NoVentilas) / 8)) / (36 * (2 * ((2.1 / _h1) - 2)))) * 100,3)]
145: tiempo_mano = tiempoMaquina
146:
147: End Sub
```

Figura 317 Método calcular() del CT 2230.

6.5.6 CentroTrabajo2240.vb – PICK (FRANKLIN)

Al formulario de este centro de trabajo se le agrego un nuevo valor de entrada, debido a que es necesario para obtener el valor del tiempo estándar. En la siguiente imagen se muestra el formulario anterior y el actual.

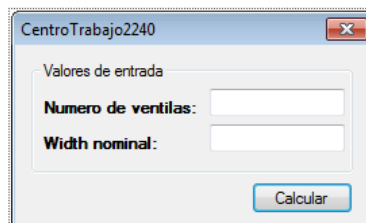


Figura 318 Formulario antes de la actualización.

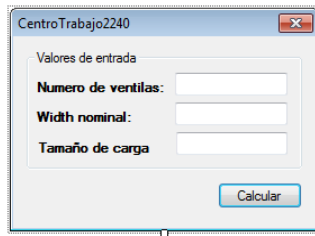


Figura 319 Formulario actual.

Como se agregó un nuevo valor de entrada, entonces se tuvieron que agregar nuevas variables y nuevos parámetros a la clase.

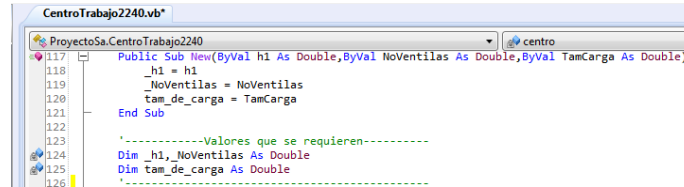


Figura 320 Variables y parámetros del CT 2240.

También se agregó una variable tipo Dato y después agregarlo a la listaDatosRequeridos, como se muestra a continuación.

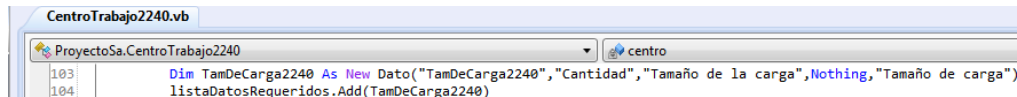


Figura 321 Variable tipo Dato.

Posteriormente se pasa el valor de la lista mediante el método pasarValoresDeListas, como se muestra en la siguiente imagen.

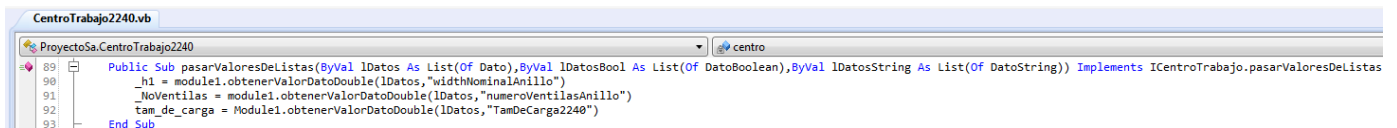


Figura 322 Método pasarValoresDeListas().

Una vez que se realizó todo lo anterior se procedió a modificar el método calcular. En la siguiente imagen se muestra el código de dicho método.

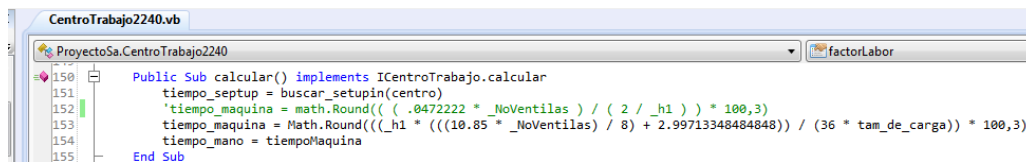


Figura 323 Método calcular() del CT 2240.

El único cambio que se realizó fue la sustitución de la fórmula anterior que es la variable tiempo_maquina que está en color verde, por la nueva fórmula.

Y se finalizó la actualización de este centro de trabajo con la modificación del evento click del botón del formulario donde únicamente se le asignó un campo de texto a la variable tam_de_carga.

```

CentroTrabajo2240.vb
ProyectoSa.CentroTrabajo2240
138 Sub Btn_calcularClick(sender As Object, e As EventArgs)
139     If IsNumeric(textBox1.Text) And IsNumeric(textBox2.Text) And IsNumeric(textBox3.Text) Then
140         NoVentilas = CDec(textBox1.Text)
141         _h1 = CDec(textBox2.Text)
142         tam_de_carga = CDec(textBox3.Text)
143         calcular()
144         Close()
145     Else
146         MsgBox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
147     End If
148 End Sub

```

Figura 324 Evento clic del botón del formulario.

6.5.7 CentroTrabajo2250.vb – I.D. BRUSH (FRANKLIN)

Al formulario de este centro de trabajo no se le realizó ninguna modificación, así que quedó de igual forma. A continuación se muestra el formulario del centro de trabajo.

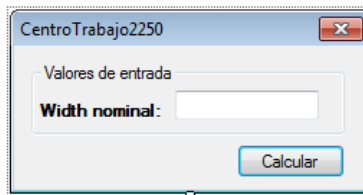


Figura 325 Formulario del CT 2250.

La única modificación que se realizó se llevó a cabo dentro del método calcular(). En la siguiente imagen se muestra el código de dicho método.

```

CentroTrabajo2250.vb
ProyectoSa.CentroTrabajo2250
136 Public Sub calcular() implements ICentroTrabajo.calcular
137     tiempo_septup = buscar_setupin(centro)
138     tiempo_maquina = Math.Round((( 2.3975933 ) / ( 5 / _h1 ) ) * 100,3)
139     tiempo_maquina = Math.Round(((61.0566663348416 * _h1) / 226.8) * 100,3)
140     tiempo_mano = tiempoMaquina
141 End Sub

```

Figura 326 Método calcular() del CT 2250.

6.5.8 CentroTrabajo2255.vb – SCOTCHBRITE RINGS (FRANKLIN)

Al formulario de este centro de trabajo no se le realizó ninguna modificación debido a que los valores de entrada con los que cuenta son suficientes para realizar el cálculo del tiempo estándar. En la siguiente imagen se muestra el formulario del centro de trabajo.

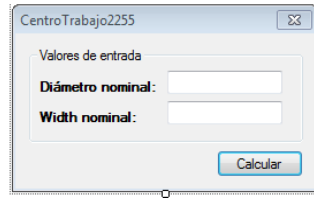


Figura 327 Formulario del CT 2255.

La única modificación que se realizó se hizo dentro del método calcular() de la clase. A continuación se muestra el código del método.

```

CentroTrabajo2255.vb
ProyectoSa.CentroTrabajo2255
144 Public Sub calcular() implements ICentroTrabajo.calcular
145     tiempo_septup = buscar_setupin(centro)
146     'tiempo_maquina = math.Round(((( 61.43 + ( ( 18.06 * _d1 ) / 2.49456 ) ) * _h1 ) / 135) * 100,3)
147     tiempo_maquina = Math.Round(((((_d1 * 20.7711688311688) / 2.3617) + 56.3468092057218) * _h1) / 134.928) * 100,3)
148     tiempo_mano = tiempoMaquina
149 End Sub

```

Figura 328 Método calcular() del CT 2255.

Lo único que se realizó fue reemplazar la fórmula anterior que es la variable tiempo_maquina comentada, por la nueva fórmula que por obvias razones es la negra.

6.5.9 CentroTrabajo2290.vb – PHOSPHATE GASOLINE

Al formulario de este centro de trabajo se le eliminó un valor de entrada debido a que la nueva fórmula para obtener el tiempo estándar solo requiere de dos valores de entrada. En la siguiente imagen se muestra el formulario anterior y el formulario actual.

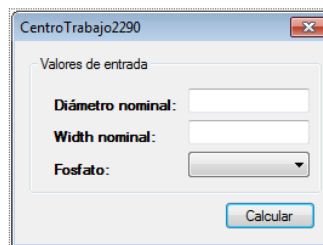


Figura 329 Formulario antes de la actualización.

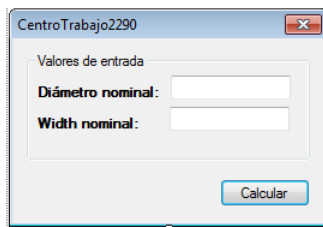


Figura 330 Formulario actual.

Como se eliminó un valor de entrada, entonces también se tuvo que eliminar su respectiva variable y también su parámetro. A continuación se muestra el constructor y las variables de la clase.

```

107 Public Sub New(ByVal d1 As Double, ByVal h1 As Double)
108     _d1 = d1
109     _h1 = h1
110 End Sub
111
112 '-----Valores que se requieren-----
113 Dim _d1 As Double
114 Dim _h1 As Double
115

```

Figura 331 Constructor y variables de la clase.

También se eliminó su variable tipo Dato y se eliminó de la listaDatosRequeridos quedando de la siguiente manera.

```

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 End Sub

```

Figura 332 Constructor donde se encuentra la listaDatosRequeridos.

También se eliminó su valor del método pasarValoresDeListas quedando el método de la siguiente forma.

```

89 Public Sub pasarValoresDeListas(ByVal IDatos As List(Of Dato), ByVal IDatosBool As List(Of DatoBoolean), ByVal IDatosString As List(Of DatoString)) Implements ICentroTrabajo.pasarValoresDeListas
90     _d1 = module1.obtenerValorDatoDouble(IDatos, "diaNominalAnillo")
91     _h1 = module1.obtenerValorDatoDouble(IDatos, "widthNominalAnillo")
92 End Sub

```

Figura 333 Método pasarValoreDeListas.

Una vez que se eliminaron todas las variables relacionadas al valor de entrada eliminado, se creó una función llamada buscar_bastidor(), el código del método es el siguiente.

```

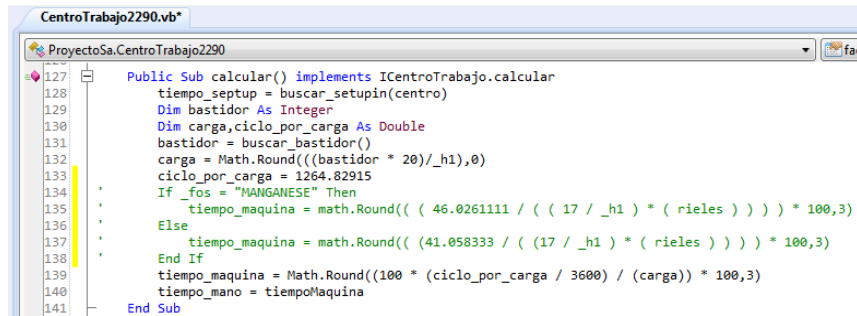
164 Private Function buscar_bastidor() As Integer
165     Dim b As Integer
166     If _d1 >= 5.71 Then
167         b = 5
168     ElseIf _d1 >= 4.6
169         b = 9
170     ElseIf _d1 >= 4.1
171         b = 11
172     ElseIf _d1 >= 1.75
173         b = 15
174     ElseIf _d1 >= 0.8268
175         b = 29
176     End If
177     Return b
178 End Function

```

Figura 334 buscar_bastidor().

Lo que se realiza en la función anterior es que se obtiene el valor de la variable b de acuerdo al valor de diámetro que el usuario inserte.

Después se modificó el método calcular(), donde se agregó el siguiente código.

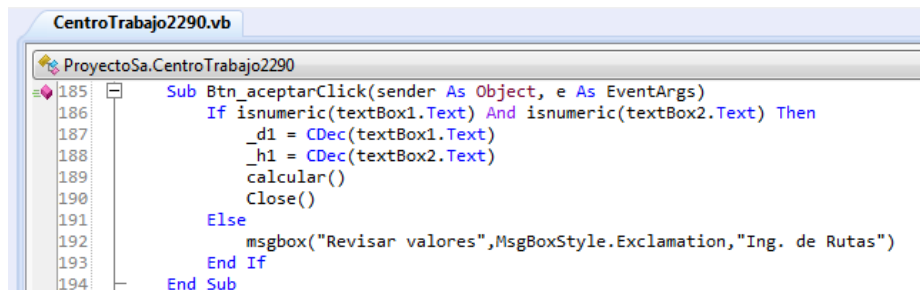


```
127 Public Sub calcular() implements ICentroTrabajo.calcular
128     tiempo_septup = buscar_setupin(centro)
129     Dim bastidor As Integer
130     Dim carga,ciclo_por_carga As Double
131     bastidor = buscar_bastidor()
132     carga = Math.Round(((bastidor * 20)/_h1),0)
133     ciclo_por_carga = 1264.82915
134     If _fos = "MANGANESE" Then
135         tiempo_maquina = math.Round(( ( 46.0261111 / ( ( 17 / _h1 ) * ( rieles ) ) ) ) * 100,3)
136     Else
137         tiempo_maquina = math.Round(( ( 41.058333 / ( ( 17 / _h1 ) * ( rieles ) ) ) ) * 100,3)
138     End If
139     tiempo_maquina = Math.Round((100 * (ciclo_por_carga / 3600) / (carga)) * 100,3)
140     tiempo_mano = tiempoMaquina
141 End Sub
```

Figura 335 Método calcular() del CT 2290.

Dentro de este método se agregaron las variables carga, ciclo_por_carga y bastidor, esta última obtiene su valor de la función buscar_bastidor(), la variable carga obtiene su valor de una operación matemática la cual depende del width que ingrese el usuario al formulario, y por ultimo a la variable ciclo_por_carga se le asignó un valor fijo. Una vez que se tiene el valor de las tres variables se procede a realizar el cálculo del tiempo estándar. Como se puede apreciar en la imagen anterior la parte que se encuentra comentada pertenece a las formulas anteriores.

Y para finalizar la actualización de este centro de trabajo se modificó el evento clic del botón donde únicamente se eliminó la igualdad de la variable correspondiente al valor de entrada con el comboBox que se le había asignado, quedando el código como se muestra en la siguiente imagen.



```
185 Sub Btn_aceptarClick(sender As Object, e As EventArgs)
186     If isnumeric(textBox1.Text) And isnumeric(textBox2.Text) Then
187         _d1 = CDec(textBox1.Text)
188         _h1 = CDec(textBox2.Text)
189         calcular()
190         Close()
191     Else
192         msgbox("Revisar valores",MsgBoxStyle.Exclamation,"Ing. de Rutas")
193     End If
194 End Sub
```

Figura 336 Evento clic del botón del formulario.

6.5.10 CentroTrabajo2295.vb – VISUAL INSPECTION (FRANKLIN)

Al formulario de este centro de trabajo no se le realizó ninguna modificación, porque con el valor de entrada que contiene es suficiente para realizar el cálculo del tiempo estándar. A continuación se muestra el formulario.

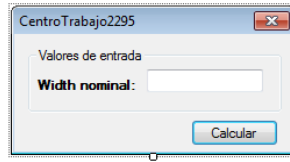


Figura 337 Formulario del CT 2295.

Entonces la única modificación se realizó en el método calcular(). En la siguiente imagen se muestra el código del método calcular().

```
CentroTrabajo2295.vb
ProyectoSa.CentroTrabajo2295
137 Public Sub calcular() implements ICentroTrabajo.calcular
138     tiempo_septup = buscar_setupin(centro)
139     'tiempo_maquina = math.Round((( 161.16 * _h1 ) / 648)*100,3)
140     tiempo_maquina = Math.Round(((169.4712575 * _h1) / 1422) * 100,3)
141     tiempo_mano = tiempoMaquina
142 End Sub
```

Figura 338 Método calcular del CT 2295.

Lo único que se realizó fue sustituir la fórmula anterior que es la variable tiempo_maquina en color verde, por la nueva fórmula que es la que está en negro.

7. Resultados

En mi participación en la planta, apoyé para el logro de los objetivos del departamento de rutas, pudiendo mencionar los siguientes.

- Se agregaron las nuevas materias primas a los centros de trabajo 715, 9361 y 9362.
- Se actualizaron los métodos, cálculos, interfaces gráficas a los centros de trabajo, con el fin de obtener los nuevos valores del tiempo estándar.
- También se dieron de alta en el sistema RGP los siguientes centros de trabajo 232 y 285.
- La información proporcionada por el software RGP estará actualizada y lista para darse de alta en el ERP de la empresa, que en este caso se llama SAP.
- Se repararon varios errores que el personal del departamento de rutas ya tenía identificados.

8. Conclusiones y recomendaciones

8.1 Conclusiones

Cumpliendo con los objetivos que se plantearon en este proyecto se concluye que las operaciones y formulas con los que se obtiene el tiempo estándar de los centros de trabajo están actualizadas y en funcionamiento, también se encuentra actualizada la lista de materia prima de algunos centros de trabajo. Esto garantiza el buen funcionamiento del software RGP, y mientras dicho software funcione de manera correcta, ayuda de forma significativa al departamento de rutas, porque de este software se obtiene la información necesaria para fabricar los componentes.

8.2 Recomendaciones

Es muy importante que el personal que lleva a cabo estas actividades tenga conocimientos en programación orientada a objetos, ya que el software RGP está basado en ese tipo de programación.

Además dicho personal debe de tener conocimientos de manipulación de base de datos ya que el software almacena toda la información en una base de datos montada en un servidor remoto.

Dentro de esta empresa se recomienda que el programador domine el lenguaje de programación Visual Basic .NET ya que la mayoría de los softwares que utiliza la empresa están codificados bajo ese lenguaje, esto debido que el principal sistema operativo que se utiliza en esta empresa es Windows. También se recomienda que el programador sepa utilizar el gestor de base de datos SQL SERVER en cualquiera de sus versiones.

9. Referencias

Díaz, A., Gonzales, J. C., & Ruiz, M. E. (2005). *IMPLANTACIÓN DE UN SISTEMA ERP EN UNA ORGANIZACIÓN*. FISI-UNMSM.

Hernández Muñoz, J. A. (1999). *Así es SAP R/3*. Madrid: Osborne/McGraw-Hill.

Juárez López, F. J. (2011). *MIGRACIÓN ERP-SYTELINE / ERP-SAP*. Instituto Tecnológico de Aguascalientes, Aguascalientes, Ags. México.

Muro Márquez, J. N. (2011). *Implementación y seguimiento de acciones correctivas y preventivas puestas en piso*. Universidad Politécnica de Aguascalientes, Aguascalientes, Ags. México.

Sancho Cancela, J. A. (2002). *ERP Funcionalidades*.