



**UNIVERSIDAD MICHOCANA DE SAN
NICOLÁS DE HIDALGO**

FACULTAD DE INGENIERÍA EN TECNOLOGÍA DE LA MADERA

TESIS

**“SOFTWARE PARA LA CUBICACIÓN DE MATERIA PRIMAS Y PRODUCTOS
FORESTALES MADERABLES”**

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN TECNOLOGÍA DE LA MADERA

PRESENTA:

GABRIEL PÉREZ JUÁREZ

ASESOR:

M. C. MARCO ANTONIO HERRERA FERREYRA

MARZO DEL 2012



ÍNDICE.

	Pág.
INDICE DE CUADROS.....	IV
INDICE DE FIGURAS.....	V
1. INTRODUCCION.....	1
2. ANTECEDENTES.....	2
3. OBJETIVO GENERAL.....	4
3.1. Objetivos específicos.....	4
4. MARCO DE REFERENCIA.....	5
4.1. Panorama general del sector forestal en México.....	5
4.2. Definición de Materias Primas y Productos Forestales Maderables.....	8
4.3. Principales Materias Primas y Productos Forestales Maderables obtenidos del aprovechamiento de un árbol.....	8
4.4. Introducción a la cubicación.....	9
4.4.1. Concepto de Cubicación.....	10
4.4.2. Importancia de la Cubicación.....	10
4.5. Unidades de medición utilizadas en la cubicación.....	10
4.6. Cubicación de Materias Primas y Productos Forestales Maderables.....	12
4.6.1. Árboles completos.....	12
4.6.1.1. Volumen del fuste.....	12
4.6.1.2. Volumen de la punta.....	12
4.6.1.3. Volumen del tocón.....	13
4.6.2. Madera en rollo.....	15
4.6.2.1. Fórmula de Huber.....	15
4.6.2.2. Fórmula de Smalian.....	16
4.6.2.3. Fórmula de Newton.....	18
4.6.2.4. Equivalencias empleadas en cubicación de madera en rollo....	19
4.6.3. Cubicación por apilamiento.....	19
4.6.3.1. Madera en rollo cortas dimensiones (trocito o bolo).....	19
4.6.3.2. Leña en raja.....	20

4.6.3.3. Brazuelo.....	21
4.6.3.4. Astillas para celulosa.....	23
4.6.4. Madera aserrada o con escuadría.....	23
4.6.4.1. Métodos de cubicación.....	25
4.6.4.2. Equivalencias empleadas en cubicación de madera aserrada.....	27
4.6.5. Embalajes de madera.....	28
4.6.5.1. Método de cubicación.....	29
4.6.6. Tableros contrachapados o Triplay.....	30
4.6.6.1. Método de cubicación.....	30
5. LENGUAJE DE PROGRAMACION C.....	33
5.1. Definición de Software.....	33
5.2. Características del lenguaje C.....	33
5.3. Estructura de un programa en C.....	33
5.4. Tipos de datos.....	34
5.5. Operadores aritméticos y de asignación.....	35
5.6. Salida / Entrada.....	36
5.7. Operadores relacionales.....	36
5.8. Sentencias condicionales.....	37
5.9. Operadores lógicos.....	37
5.10. Bucles.....	38
5.11. Funciones.....	38
5.12. Estructuras.....	39
6. METODOLOGÍA.....	40
6.1. Modelo incremental.....	40
6.1.1. Análisis.....	41
6.1.2. Diseño.....	41
6.1.3. Programación o codificación.....	42
6.1.3.1. Descarga e instalación del programa Turbo++ 3.0.....	42
6.1.3.2. Creación del archivo para el Software.....	43
6.1.3.3. Desarrollo del código fuente del Software.....	43
6.1.4. Prueba.....	44
7. RESULTADOS.....	45
7.1. Entrada al programa.....	45

7.2. Forma de uso del Software.....	45
7.2.1. Presentación del programa.....	45
7.2.2. Presentación de productos forestales maderables.....	46
7.2.3. Cubicación de árboles completos.....	47
7.2.4. Cubicación de madera en rollo.....	49
7.2.5. Cubicación por apilamiento.....	52
7.2.6. Cubicación de embalajes de madera.....	53
7.2.7. Cubicación de madera aserrada o con escuadría.....	56
7.2.8. Cubicación de triplay.....	58
8. CONCLUSIONES.....	59
9. RECOMENDACIONES.....	60
10. BIBLIOGRAFÍA.....	61
11. ANEXOS.....	63
11.1. Código fuente del Software.....	63

ÍNDICE DE CUADROS.

Cuadro	Pág.
1. Existencias de vegetación en México.....	5
2. Producción forestal maderable por Estado 2007.....	6
3. Producción forestal maderable por producto.....	7
4. Producción forestal maderable por género.....	7
5. Equivalencias del Sistema Métrico Decimal.....	11
6. Equivalencias del Sistema Ingles.....	11
7. Equivalencias empleadas en la cubicación de madera en rollo.....	19
8. Dimensiones comerciales de tablas y tablonés.....	23
9. Dimensiones comerciales de polines.....	24
10. Dimensiones comerciales de durmientes.....	24
11. Dimensiones comerciales de vigas.....	25
12. Equivalencias empleadas para la cubicación de madera aserrada.....	28
13. Tipos de datos.....	34
14. Secuencias de escape.....	35
15. Operadores aritméticos.....	35
16. Operadores de asignación.....	36
17. Prioridad de los operadores.....	36
18. Operadores relacionales.....	37
19. Operadores lógicos.....	37
20. Nombre de la función asignada a cada producto forestal.....	44

ÍNDICE DE FIGURAS.

Figura		Pág.
1.	Principales productos obtenidos del aprovechamiento del árbol.....	9
2.	Árbol completo que se desea cubicar.....	13
3.	Madera en rollo que se desea cubicar empleando la fórmula de Huber.....	16
4.	Madera en rollo que se desea cubicar empleando la fórmula de Smalian.....	17
5.	Madera en rollo que se desea cubicar empleando la fórmula de Newton.....	18
6.	Apilamiento de trocito con corteza que se desea cubicar.....	20
7.	Apilamiento de leña en raja con corteza que se desea cubicar.....	21
8.	Apilamiento de Brazuelo con corteza que se desea cubicar.....	22
9.	Tablón que se desea cubicar empleando pies tabla.....	26
10.	Tabla que se desea cubicar empleando metros cúbicos.....	27
11.	Tarima industrial que se desea cubicar empleando pies tabla.....	29
12.	Hoja de triplay que se desea cubicar empleando m^2 base 5.2 mm.....	31
13.	Flujograma del Modelo Incremental.....	40
14.	Diseño del programa.....	41
15.	Entrada al software.....	45
16.	Ventana de presentación del software.....	46
17.	Ventana del menú de productos forestales maderables.....	47
18.	Ventana del menú para cubicar árboles completos.....	48
19.	Ventana para cubicar árboles completos.....	49
20.	Ventana para cubicar madera en rollo.....	51
21.	Ventana del menú para selección de producto.....	52
22.	Ventana para la cubicación por apilamiento.....	53
23.	Ventana para la selección de embalaje.....	54
24.	Ventana para el despiece de tarimas.....	55
25.	Ventana para la cubicación de tarimas.....	55
26.	Ventana para la cubicación de cajas para empaque.....	56
27.	Ventana para la cubicación de madera aserrada.....	57
28.	Ventana para la cubicación de madera aserrada por tipo de producto.....	57
29.	Ventana para la cubicación de triplay.....	58

1. INTRODUCCIÓN.

México cuenta con una superficie total de 196.8 millones de hectáreas (ha.), de las cuales el 72% (141.7 millones de ha.) se encuentra dedicada a los distintos usos forestales, teniendo como principales ecosistemas; Bosques de clima templado frío (30.4 millones de ha.), Selvas (26.4 millones de ha.) y Vegetación de zonas áridas (58.5 millones de ha.).

No obstante de contar con un bajo nivel de aprovechamiento de los bosques y selvas, ya que de las 21 millones de ha. con potencial comercial, solo se encuentran bajo el manejo de técnicas silvícolas, poco más de la cuarta parte; cabe destacar la relevancia del Sector Silvícola en la economía del país, ya que anualmente se generan un gran número de empleos y una derrama económica significativa.

En cuanto a la producción forestal maderable, tras una importante caída a finales de los años ochenta, se ha estabilizado, manteniéndose en aproximadamente 6.3 millones de metros cúbicos rollo anual (m^3r); aunque su potencial productivo está estimado en 30.5 millones de m^3r por año (Gestiopolis, 2010).

Debido a la importancia que tiene la actividad forestal en los aspectos técnicos, económicos, ecológicos, sociales y políticos de nuestro país, resulta de gran relevancia contar con una evaluación precisa de los recursos forestales para el manejo sostenido del bosque; asimismo, la aplicación correcta de las técnicas de medición forestal trae consigo diversas ventajas dentro del proceso productivo, como por ejemplo: se dan bases para conocer las existencias reales totales del bosque y extraer, los volúmenes que sustenten los incrementos respectivos, el pago correcto en los trabajos derivados del aprovechamiento (derribo, elaboración de productos largas y cortas dimensiones, así como puntas y ramas); permite hacer coincidir el volumen de madera que se transporta con la documentación que la ampara; en la industria (proceso de transformación), es muy importante para poder realizar diagnósticos sobre la capacidad y rendimientos de producción, y finalmente para la comercialización de los diferentes productos.

No obstante se puede establecer que una de las principales razones de las deficiencias en la medición de los productos forestales maderables, es la falta de divulgación y homogenización de las diferentes técnicas y metodologías existentes; por tal motivo, surgió la idea de hacer este **Software o programa computacional** en donde se aproveche un medio tan importante como es la computadora, para poder desarrollar esta herramienta de apoyo para determinar el volumen y llevar un control de las principales materias primas y productos forestales maderables de una manera didáctica, exacta y de fácil uso.

2. ANTECEDENTES.

Los productos maderables que del bosque se obtienen, inicialmente se utilizaron de manera doméstica, es decir, en la construcción de casas habitación y como material combustible, sobre todo para la preparación de alimentos y la transacción se realizaba a través de trueque; se puede establecer que en nuestro país la madera adquiere un mayor valor comercial, con la introducción del ferrocarril a principios del siglo XVIII, por lo cual se hizo necesario conocer las cantidades de madera para la elaboración del durmiente utilizado en la construcción de las vías ferreas, posteriormente con la introducción de la luz eléctrica la fabricación del poste de madera fue de primera necesidad en el establecimiento de las líneas eléctricas. Cabe destacar que de manera doméstica paralelo a lo anterior se daba ya la fabricación de muebles de manera doméstica y algunos productos de primera necesidad como cajas, bancos, mesas sillas etc.

Se tienen antecedentes que a finales del siglo XVIII, se establecieron algunas fábricas para la elaboración de celulosa y papel, cuya materia prima se constituía primordialmente de madera, por tal razón se hizo necesario realizar el aprovechamiento de los bosques de manera ordenada, bajo la aplicación de técnicas silvícolas. Posteriormente a inicios del siglo XIX, prolifera la instalación de aserraderos, fabricas de tableros (Aglomerados y contrachapados), fábricas de cajas para empaque, talleres artesanales, etc.

Por tal razón, se hizo necesario conocer la estimación de los volúmenes maderables con la mayor precisión posible, en las diferentes fases del proceso productivo, como son: la elaboración de los inventarios forestales para contar con información numérica del potencial de los bosques y selvas, los estudios dasonómicos que sustentan el aprovechamiento sostenible de los mismos, los trabajos de extracción, transporte, industrialización y comercialización, donde en cada una de ellas, se requiere de realizar una serie de mediciones y cálculos que permitan su ejecución y desarrollo dentro de las normas técnicas y legales establecida.

Cabe destacar que tradicionalmente en México, la cuantificación y/o cubicación de la madera en rollo y aserrada se ha venido realizando a través del sistema inglés, lo cual ha perdurado hasta la actualidad; sin embargo la Ley General de Desarrollo Forestal Sustentable de 1986, establece como instrumento de medición oficial de los productos maderables el Sistema Métrico Decimal.

De acuerdo con lo anterior para fines prácticos en nuestro país, se adoptaron diversas formas de medición y/o cubicación de los productos maderables que en su mayoría eran imprecisas y favorecían generalmente al comprador de la madera en rollo largas dimensiones como la Regla Doyle, en el caso de la madera en rollo cortas dimensiones y leñas, las famosas cuerdas que básicamente se trataba de pilamientos con dimensiones preestablecidas donde se daban coeficientes de

intersección para bajar las existencias volumétricas. Por otra parte los transportistas generalmente incurrían en infracciones debido a las diferencias de existencias volumétricas ocasionadas por los diferentes sistemas de cubicación (Sistema Inglés y Métrico decimal); asimismo en las industrias establecidas para la transformación de la materia prima (madera en rollo), es frecuente que al realizar las cuantificaciones y cubicación de lo que entra y sale, simple y sencillamente no concuerdan los números.

En lo referente a Software y/o programas computacionales o aplicaciones de sistemas de computo para el sector forestal y en concreto relacionados con el tema de cubicación, se tienen pocas referencias, por citar un ejemplo, está el programa CUBICA v1.2 que fue desarrollado en Visual Basic 6.0 y está basado en modelos del perfil del árbol y permite calcular mediante el uso de PC (Con sistema operativo Windows 95 o superior) el volumen de las existencias de una masa forestal a partir de los datos de un inventario forestal. Además permite simular distintos despieces de acuerdo con criterios establecidos por el usuario y proporciona los datos en cada producto. La aplicación se desarrolla bajo un sistema de menús y formularios en los que el usuario introduce los datos necesarios para la simulación (Rodríguez y Broto, 2002).

Por otro lado Becerril (2002), desarrolló e implementó un sistema computacional en programación DELPHI, que permite la construcción de programas de secado para diferentes especies, el cual también permite graficar las variables del programa de secado e informar aspectos generales sobre el secado de la madera. También se tiene conocimiento de trabajos como el de Barraza (1990) en donde llevó a cabo un control estadístico de calidad computarizado para un aserradero y menciona que el uso de la computadora para realizar cálculos aritméticos es imprescindible y bastante confiable ya que de otra manera, el realizarlas manualmente es bastante laborioso y tiene un alto riesgo de error además de los tiempos que se necesitan. Así mismo Murguía (1996), realizó un modelado de control de producción de un aserradero utilizando la hoja electrónica Microsoft Excel versión 5.0 puntualizando que en esta propuesta de modelado de control de producción se combinó la industria, la computación y la administración con la finalidad de obtener resultados favorables para la industria del aserrío.

Como se puede observar aún cuando se han realizado serios esfuerzos para precisar la cubicación de las diferentes materias primas y productos forestales maderables y efectuado el desarrollo de Software o programas computacionales para la aplicación en el sector forestal, aun falta mucho por hacer, sobre todo con éste último, ya que existe un gran potencial con las herramientas actuales de computación (programación y/o desarrollo de software). Por tal motivo este trabajo se puede considerar un trabajo sencillo pero que a su vez es de gran utilidad sobre todo para aquellas personas involucradas directa o indirectamente en el aprovechamiento, transformación y comercialización de materias primas y productos forestales maderables.

3. OBJETIVO GENERAL.

Desarrollar un Software en Lenguaje C, que permita determinar el volumen mediante la cubicación, de materias primas y productos forestales maderables de manera sencilla, práctica y de fácil uso.

3.1 Objetivos específicos.

1. Impulsar e incentivar el desarrollo de Software y/o programas computacionales enfocados al sector forestal para coadyuvar a su desarrollo.
2. Resaltar la importancia y el potencial que tiene el campo de la programación, para realizar aplicaciones en el sector forestal.
3. Describir y detallar las fases o etapas (ciclo de vida) que comprende el desarrollo de un software y/o programa computacional, así como la forma en que se estructura el código fuente del mismo.
4. Describir las características y usos de las materias primas y productos forestales maderables.
5. Describir los sistemas de cubicación y unidades de medición empleadas en la práctica, para determinar el volumen de materias primas y productos forestales maderables.

4. MARCO DE REFERENCIA.

4.1. Panorama general del sector forestal en México.

Actualmente al sector forestal en México, no se le ha dado la importancia que se le debería dar, los bosques además de ser importantes para el hábitat del ser humano, es un recurso económico importante, ya que de él se extraen diferentes productos que participan como materias primas o como productos finales.

Por su situación geográfica y características de clima y relieve, México cuenta con diferentes tipos de recursos forestales:

- Bosques tropicales, en donde se encuentran maderas preciosas como: cedro rojo, caoba, ébano, palo de rosa y otras.
- Bosques mixtos, donde existen maderas duras y maderas suaves.
- Bosques de coníferas, que también se conocen como maderas blandas, que en especial sirve a la industria de la celulosa y el papel
- Bosques espinosos o Chaparrales, en las zonas semidesérticas, se explota: cera de candelilla, ixtle, guayule, jojoba, nopal, etc.

Del total de la superficie territorial de México, el 72% (141,7 millones de hectáreas) se encuentra dedicada a los distintos usos forestales (Cuadro 1). El país cuenta con varios ecosistemas forestales naturales, siendo los principales tipos:

- Bosques de clima templado frío (31.8 millones de ha.).
- Selvas (23.5 millones de ha.).
- Vegetación de zonas áridas (58.5 millones de ha.).

Cuadro 1. Existencias de vegetación en México.

Tipo de vegetación	Superficie (ha)	Porcentaje
Bosque templado	31,8	22,4
Selvas	23,5	16,6
Manglares	0,7	0,5
Vegetación hidrófila y halófila	4,2	3,0
Vegetación de zonas áridas	58,5	41,3
Otras asociaciones	0,8	0,6
Áreas forestales perturbadas	22,2	15,7
Superficie total forestal	141,7	100,0

Fuente: Gestipolis, 2010.

En lo que respecta a la producción forestal maderable durante el período 1998-2007, ha variado de 8.3 millones de metros cúbicos rollo (m³r) en 1998 a 9.4 millones de m³r en 2000, representando un aumento del 13.2 %. A partir del 2001 se presentó una disminución, con excepción del año 2003, en el cual la producción alcanzó un volumen de 7.0 millones de m³r, lo que representó un aumento del 5.0 % con respecto al año anterior. En los años 2004 y 2005 hubo una disminución, reportándose 6.7 y 6.4 millones de m³r respectivamente y para 2006 y 2007 se presentó un ligero aumento, pasando a 6.5 y 7.0 millones de m³r, que representó un incremento del 0.9% y 7.8 respectivamente, en relación al año anterior (Cuadro 2).

Los principales estados productores en 2007 fueron: Durango (25.4%), Chihuahua (22.4%), Michoacán (9.9%), Jalisco (8.5%) y Oaxaca (7.7%) que contribuyeron con el 74.0% de la producción total, equivalente a 5.2 millones de m³r. Cabe resaltar que los dos estados con mayor producción fueron Durango y Chihuahua, con una participación conjunta del 47.8% de la producción forestal maderable total.

Los estados de Aguascalientes, Baja California Sur, Colima, Chiapas, Chihuahua, Durango, Guerrero, Hidalgo, Jalisco, México, Michoacán, Morelos, Nayarit, Nuevo León, Oaxaca, Puebla, Querétaro, Sinaloa, Sonora, Tlaxcala, Veracruz y Zacatecas registraron un aumento en la producción, dando un incremento total de 644,371 m³r (10.6%) en relación al año anterior.

Por el contrario, los estados de Baja California, Campeche, Coahuila, Distrito Federal, Guanajuato, Quintana Roo, San Luis Potosí, Tabasco, Tamaulipas y Yucatán, registraron una disminución en conjunto de 137,078 m³r (34.6%) en comparación a 2006.

Por otro lado, el 66.8% de la producción de 2007 se destinó a madera para aserrío (4.7 millones de m³r), el 12.6% a productos celulósicos (882 mil m³r) y el restante 20.6% (1.4 millones de m³r) a tableros, postes, pilotes y morillos y combustibles (Cuadro 2). El estado de Durango es el principal productor de escuadría, Celulósicos, chapa, postes, durmientes y combustible, siguiéndole el estado de Chihuahua, en escuadría, chapa y celulósicos.

Cuadro 2. Producción forestal maderable por Estado 2007 (m³r).

Estado	Producción maderable			Variación %	Destino de la producción				
	2006	2007	Participación		Aserrio	Celulosa	Chapa y triplay	Postes	Combustibles
Durango	1,757,638	1,775,224	25.4%	1	1,272,575	155,152	155,892	130,524	61,081
Chihuahua	1,446,503	1,568,189	22.4%	7	1,001,514	196,768	356,331	3,584	9,992
Michoacán	585,069	694,170	9.9	19	488,555	145,059	20,600	7,800	32,156
Jalisco	579,335	593,368	8.5	2	220,143	248,808	0	84	124,334
Oaxaca	374,200	539,371	7.7	44	416,197	99,383	0	686	23,104
Otros	1,718,423	1,818,139	26.0	6	1,270,243	37,047	1,340	70,506	439,003
Total	6,481,168	6,988,461	100	8	4,669,226	882,217	534,163	213,185	689,670

Fuente: CONAFOR, 2010.

En la producción maderable de 2007 destacan los aumentos en los volúmenes de Aserrío (100 mil m³r) celulosa (222 mil m³r), tableros (225 mil m³r), y leña (73 mil m³r), que representaron un incremento en la producción con relación a 2006, del 2.2%, 33.6%, 72.8% y 25.0%, respectivamente. Por otro lado, la producción de madera para postes disminuyó (40 mil m³r), y carbón (72 mil m³r), equivalentes a decrementos del 15.8% y 18.1%, respectivamente, en relación con el año anterior (Cuadro 3).

Cuadro 3. Producción forestal maderable por producto (m³r).

Producto	2006	2007	Variación %
Aserrío	4,569,188	4,669,226	2.2
Celulosa	660,225	882,217	33.6
Chapa y triplay	309,046	534,163	72.8
Postes	253,149	213,185	-15.8
Combustibles, leña	290,296	362,844	25
Carbón	399,264	326,826	-18.1
Total	6,481	6,988,461	7.8

Fuente: CONAFOR, 2010.

Los principales géneros y/o grupos aprovechados durante el año 2007 fueron: el pino con 5.7 millones de m³r (80.9%) y el encino 0.6 millones de m³r (8.0%), los restantes 0.8 millones de m³r (11.1%) corresponden a los otros géneros y/o grupos.

De la producción forestal nacional maderable por género/grupo, las coníferas contribuyeron con el 83.1%, las latifoliadas con el 10.2% y las tropicales con el 6.7% (Cuadro 4).

Cuadro 4: Producción forestal maderable por género (m³r).

Género y/o grupo	2006	2007	Participación %
Pino	4,922,913	5,655,928	80.9
Oyamel	112,418	116,860	1.7
Otras coníferas	73,610	36,066	0.5
Encino	777,047	561,069	8.0
Otras latifoliadas	100,029	153,109	2.2
Preciosas	37,683	21,166	0.3
Comunes tropicales	457,467	444,263	6.4
Total	6,481,168	6,988,461	100.0

Fuente: CONAFOR, 2010.

En la producción de madera de coníferas destacan los estados de Durango, Chihuahua y Michoacán, los cuales representan el 64.5%. Para las latifoliadas, los estados de mayor producción son: Durango, Sonora y Veracruz los cuales aportan el 53.7%. En cuanto al grupo de las tropicales los principales estados son: Veracruz, Tamaulipas y Campeche, con el 62.1% de la producción.

Respecto al valor de la producción, en total fue de 7,803'448,375 pesos, siendo cinco estados los que reportan el 81.2% de dicho valor: Chihuahua (37.5%), Durango (22.5%), Michoacán (8.1%), Jalisco (7.8%) y Oaxaca (5.3%).

Por grupo de productos, el 83.7% del valor la producción lo representan: la escuadría (79.6%), postes, pilotes y morillos (2.7%), así como los durmientes (1.4%).

4.2. Definición de Materias primas y productos forestales maderables.

1. Materia prima forestal se define como:

“Los productos del aprovechamiento de los recursos forestales que no han sufrido procesos de transformación hasta el segundo grado” (SEMARNAT, 2003), como por ejemplo la madera en rollo, el brazuelo, etc.

2. Producto forestal maderable se define como:

“El bien obtenido del resultado de un proceso de transformación de materias primas forestales, con otra denominación, nuevas características y un uso final distinto” (SEMARNAT, 2003), como por ejemplo la madera aserrada, el triplay, etc.

4.3. Principales materias primas y productos forestales maderables obtenidos del aprovechamiento de un árbol.

Las materias primas y productos forestales maderables que se pueden obtener de un árbol dependen básicamente de:

- La forma y dimensiones del fuste de cada árbol (ramificaciones, bifurcaciones, recto o torcido, cilíndrico u ovalado, longitud, etc.).
- La demanda del mercado por determinado producto, ya que cuando el mercado demanda un producto más que otro, se producen materias primas en mayor cantidad del producto requerido. Las principales materias primas y productos forestales maderables que se obtienen del aprovechamiento de los árboles se muestran en la Figura 1, cabe mencionar que su ilustran únicamente, los de mayor importancia comercial en México.

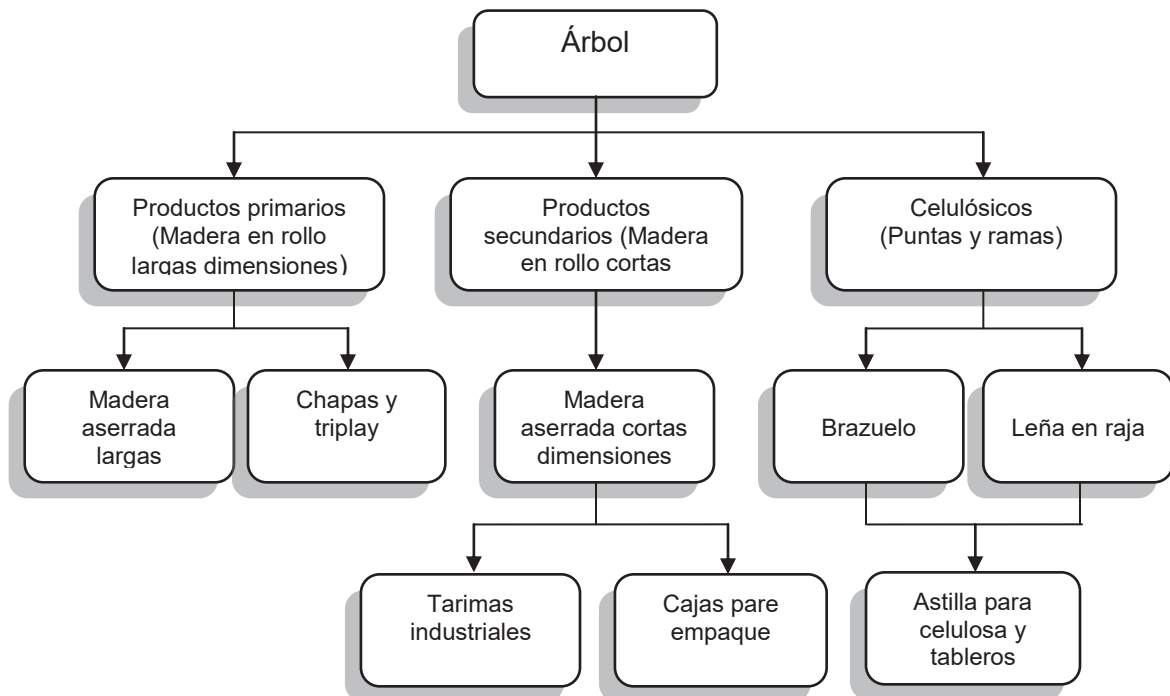


Figura 1. Principales productos obtenidos del aprovechamiento del árbol (PROBOSQUE, 1998).

4.4. Introducción a la Cubicación.

La correcta administración de los recursos forestales requiere, para su planeación y ejecución, de información tanto cuantitativa como cualitativa obtenida a través de lo que se denomina inventario forestal que demanda necesariamente el conocimiento teórico y práctico de métodos y técnicas de medición forestal. En la medición forestal se identifican dos etapas que en la práctica generalmente se dan en forma paralela, la determinación de las dimensiones de árboles y de bosques. Por tal motivo es importante mencionar los siguientes conceptos de los cuales se deriva la cubicación de las materias primas y productos forestales maderables:

La Dasometría es la rama de la Dasonomía que se ocupa de la medición de los árboles, de la determinación del volumen de los bosques y de los crecimientos de los árboles y bosques. Podemos definir 3 divisiones de la dasometría:

a) La Dendrometría, que es el objeto de nuestro estudio y que trata de la medición de las dimensiones del árbol, del estudio de su forma y, en último término, de la determinación de su volumen.

b) La Dasometría o Estereometría de la masa, que se ocupa de las cuestiones relacionadas con la estimación métrica y cubicación de la masa forestal, entendida como conjunto de árboles que conviven en un espacio común.

c) La Epidometría, que estudia las técnicas de medición y las leyes que regulan el crecimiento y producción de los árboles (Facultad de Ciencias del medio Ambiente UCML, 2006).

4.4.1. Concepto de cubicación.

Es la determinación del volumen de madera que contiene una materia prima o producto forestal maderable en cualquiera de sus formas (trozo, leña en raja brazuelo, etc.), dado en metros cúbicos (m^3) u otra medida adecuada de volumen (PROBOSQUE, 1998).

4.4.2. Importancia de la cubicación.

La cubicación de las materias primas y productos forestales maderables es de suma importancia, ya que con el conocimiento y la aplicación de las técnicas de cubicación de una manera adecuada podemos calcular: los rendimientos que se obtienen del aprovechamiento del arbolado que se derriba y se dimensiona en trozos, brazuelo y leñas; calcular cuánto volumen de madera transportan los camiones; calcular que volumen de madera entrega el vendedor al comprador; llevar controles de producción de maderas y existencias en el bosque, en brecha, en patios de concentración, en patios de las industrias de transformación, además de conocer los rendimientos que se obtienen al transformar un producto en otro (PROBOSQUE, 1998).

4.5. Unidades de medición utilizadas en la cubicación.

Para la cubicación de materias primas y productos forestales maderables se emplean básicamente 2 sistemas de unidades de medición: el Sistema Métrico Decimal, cuya unidad básica es el metro (m) y el Sistema Ingles cuyas unidades básicas son las pulgadas (pulg.) y el pie (ft). En los Cuadros 5 y 6 se muestran las principales equivalencias entre estos sistemas.

Cuadro 5. Equivalencias del Sistema Métrico Decimal.

Unidades de longitud			
1 m	10 dm	100 cm	1000 mm
1 dm	10 cm	100 mm	
1 cm	10 mm		
1 Km	1000 m		
1 Hm	100 m		
1 Dm	10 m		
Unidades de superficie			
1 Km ²		1000000 m ²	
1 Ha		10000 m ²	
1 Hm		10000 m ²	
1 Dm ²		100 m ²	
Unidades de volumen			
1 m ³	1000 dm ³	1000000 cm ³	1000000000 mm ³
1 dm ³	1000 cm ³	1000000 mm ³	
1 cm ³	1000 mm ³		

Fuente: PROBOSQUE, 1998.

Cuadro 6. Equivalencias del Sistema Inglés.

Unidades de longitud		
1 Pie	12 Pulgadas	0.3048 m
1 Pie	0.3048 m	30.48 cm
1 Pulgada	.0254 m	2.54 cm
Unidades de superficie		
1 Pie ²	144 Pulgada ²	0.0929 m ²
1 Pie ²	0.0929 m ²	629 cm ²
1 Pulgada ²	6.4516 cm ²	
Unidades de volumen		
1 Pie ³	1728Pulgadas ³	0.028317 m ³
1 Pie ³	0.028317 m ³	28.317 dm ³
1 Pulgada ³	16.387 cm ³	

Fuente: PROBOSQUE, 1998.

4.6. Cubicación de Materias Primas y Productos Forestales Maderables.

4.6.1. Árboles completos.

Para obtener el volumen de los árboles completos es importante diferenciar que para el caso que nos ocupa será el arbolado derribado, ya que para el arbolado en pie se utilizan generalmente tablas de volumen de 2 entradas (Diámetro y Altura) y generalmente son por regiones, géneros o bien incluso por grupos de especies. Al momento de derribar el árbol podemos calcular la totalidad del volumen maderable aprovechable diferenciando el fuste, tocón y punta para posteriormente cubicarlos, sumarlos y así obtener el volumen total (Donal y Schumacher, 1986).

4.6.1.1. Volumen del fuste.

El volumen de árboles completos puede obtenerse calculando el volumen de cortas secciones del mismo y sumando los resultados. Sin embargo, esos cálculos pueden abreviarse si todas las trozas son del mismo largo. Si ese largo uniforme es de L metros y las superficies de las secciones transversales sucesivas, comenzando desde la base del árbol, se designan como $A_1, A_2, A_3 \dots A_{n-1}, A_n$, el volumen de todos los troncos será:

$$V = L/2(A_1 + 2A_2 + 2A_3 \dots + 2A_{n-1} + A_n)$$

Donde:

V = Volumen del fuste (m^3).

A = Áreas de las secciones a distancias iguales del fuste (m^2).

L = Longitud del fuste (m).

4.6.1.2. Volumen de la punta.

A veces la punta del árbol, más arriba de la última troza, se asemeja a una paraboloides. El volumen de ese sólido puede calcularse con la fórmula:

$$V = \frac{1}{2} AL$$

En donde A es la superficie de la base. En algunos árboles, la forma de la punta se aproxima más a la de un cono, en tal caso el volumen se calculará con la siguiente fórmula:

$$V = 1/3 AL$$

Donde:

V = Volumen de la punta (m^3).

A = Área de la base de la punta (m^2).

L = Longitud de la punta (m).

4.6.1.3. Volumen del tocón.

Es prácticamente imposible medir con precisión el volumen del tocón. A menudo la sección transversal a nivel del suelo es tan irregular, que no puede determinarse exactamente la superficie. Por consiguiente se acostumbra considerar el tocón como si fuera un cilindro de sección transversal equivalente a su superficie superior. Para el cálculo de su volumen se emplea la fórmula siguiente:

$$V = AL$$

Donde:

V = Volumen del tocón (m^3).

A = Área de la superficie superior del tocón (m^2).

L = Longitud del tocón (m).

Ejemplo: Se desea calcular el volumen del árbol que se muestra en la Figura 2.

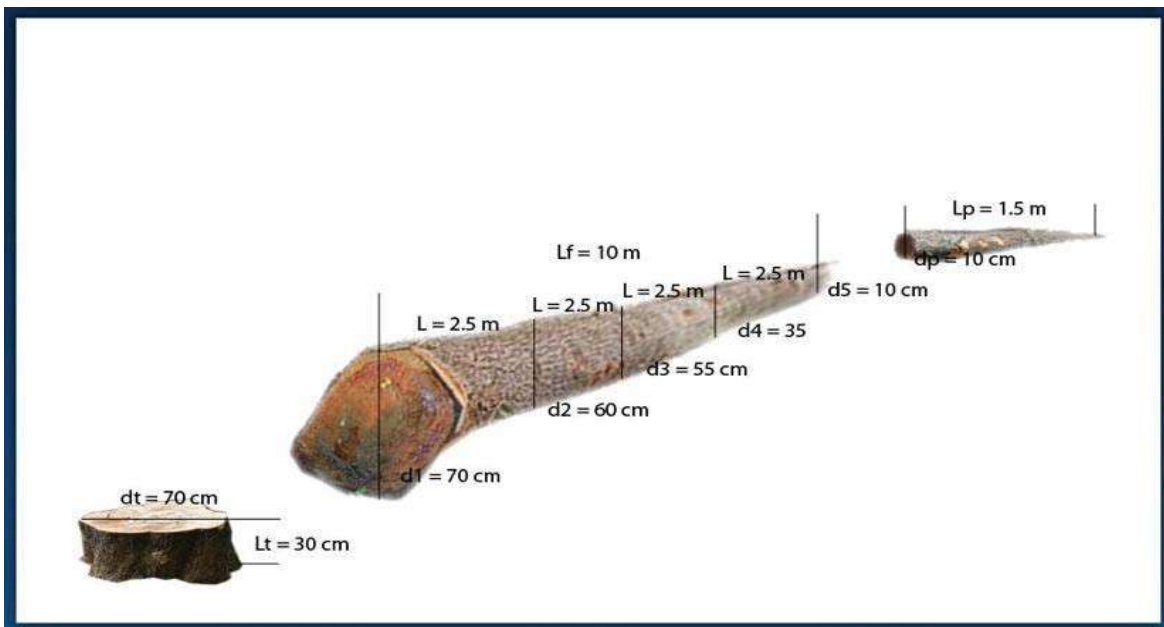


Figura 2. Árbol completo que se desea cubicar.

1. Datos y cubicación del fuste:

Longitud del fuste = 10 m

Distancia entre secciones = 2.5 m, por lo tanto son 5 las áreas de secciones del fuste que se valorarán (ver figura 2):

$$d_1 = 0.70 \text{ m, donde } A_1 = 0.7854 (d_1)^2 = 0.384 \text{ m}^2$$

$$d_2 = 0.60 \text{ m, donde } A_2 = 0.7854 (d_2)^2 = 0.283 \text{ m}^2$$

$$d_3 = 0.55 \text{ m, donde } A_3 = 0.7854 (d_3)^2 = 0.237 \text{ m}^2$$

$$d_4 = 0.35 \text{ m, donde } A_4 = 0.7854 (d_4)^2 = 0.096 \text{ m}^2$$

$$d_5 = 0.10 \text{ m, donde } A_5 = 0.7854 (d_5)^2 = 0.008 \text{ m}^2$$

Sustituyendo en la fórmula:

$$V = L/2(A_1 + 2A_2 + 2A_3 + 2A_4 + A_5)$$

$$V = 2.5 \text{ m} / 2 (0.384 \text{ m}^2) + 2(0.283 \text{ m}^2) + 2(0.237 \text{ m}^2) + 2(0.096 \text{ m}^2) + 0.008 \text{ m}^2$$

$$V = 1.25 \text{ m} (0.384 \text{ m}^2) + 0.566 \text{ m}^2 + 0.474 \text{ m}^2 + 0.192 \text{ m}^2 + 0.008 \text{ m}^2$$

$$V = 1.25 \text{ m} (1.624 \text{ m}^2) = 2.030 \text{ m}^3$$

El volumen del fuste = **2.030 m³**

2. Datos y cubicación del extremo o punta:

$$d = 0.10 \text{ m, donde } A = 0.7854 (d)^2 \Rightarrow A = 0.7854 (0.10 \text{ m})^2 = 0.008 \text{ m}^2$$

$$L = 1.5 \text{ m}$$

Sustituyendo en la fórmula:

$$V = \frac{1}{2} AL \Rightarrow V = \frac{1}{2} (0.008 \text{ m}^2 \times 1.5 \text{ m}) = 0.006 \text{ m}^3$$

El volumen de la punta = **0.006 m³**

3. Datos y cubicación del tocón:

$$d = 0.70 \text{ m, donde } A = 0.7854 (d)^2 \Rightarrow A = 0.7854 (0.70 \text{ m})^2 = 0.384 \text{ m}^2$$

$$L = 30 \text{ cm} = 0.30 \text{ m}$$

Sustituyendo en la fórmula:

$$V = AL \Rightarrow V = (0.384 \text{ m}^2 \times 0.30 \text{ m}) = 0.115 \text{ m}^3$$

El volumen del tocón = **0.115 m³**

Por lo tanto:

4. Volumen del árbol completo = Vol. Fuste + Vol. Punta + Vol. Tocón

Volumen del árbol completo = 2.030 m³ + 0.006 m³ + 0.115 m³ = 2.151 m³

Volumen del árbol completo = **2.151 m³**

4.6.2. Madera en rollo.

La madera en rollo se define como troncos de árboles derribados o seccionados con diámetro mayor a diez centímetros en cualquiera de sus extremos, sin incluir la corteza y sin importar su longitud (SEMARNAT, 2005).

La madera en rollo se clasifica de la siguiente manera:

1. Largas dimensiones: 8, 10, 12, 14, 16, 18 o hasta 20 pies de largo.
2. Cortas dimensiones: 2, 4 y 6 pies de largo.

La madera en rollo de largas dimensiones se utiliza para el abastecimiento de Fábricas de triplay, Fábricas de muebles, Fábricas de papel y aserraderos; por otra parte la madera en rollo de cortas dimensiones (trocito o bolo) se utiliza para el abastecimiento de Fábricas de papel, Fábricas de cajas de empaque, Fábricas de elaboración de tarimas, entre otros.

Existen 3 fórmulas para cubicar o determinar el volumen de la madera en rollo que son la de Huber, Smalian y Newton; de las cuales, la más puesta en práctica actualmente en México es la de Smalian (Elguero, 1994).

4.6.2.1. Fórmula de Huber.

Sirve para calcular el volumen de piezas de madera en rollo tomando como base el área de la sección media y la longitud de la misma:

$$V = A_m \times L$$

Donde:

V = Volumen (m³).

A_m = Área de la sección media de la troza (m²).

L = Longitud de la troza (m).

Ejemplo: Se desea cubicar la madera en rollo que se muestra en la figura 3.

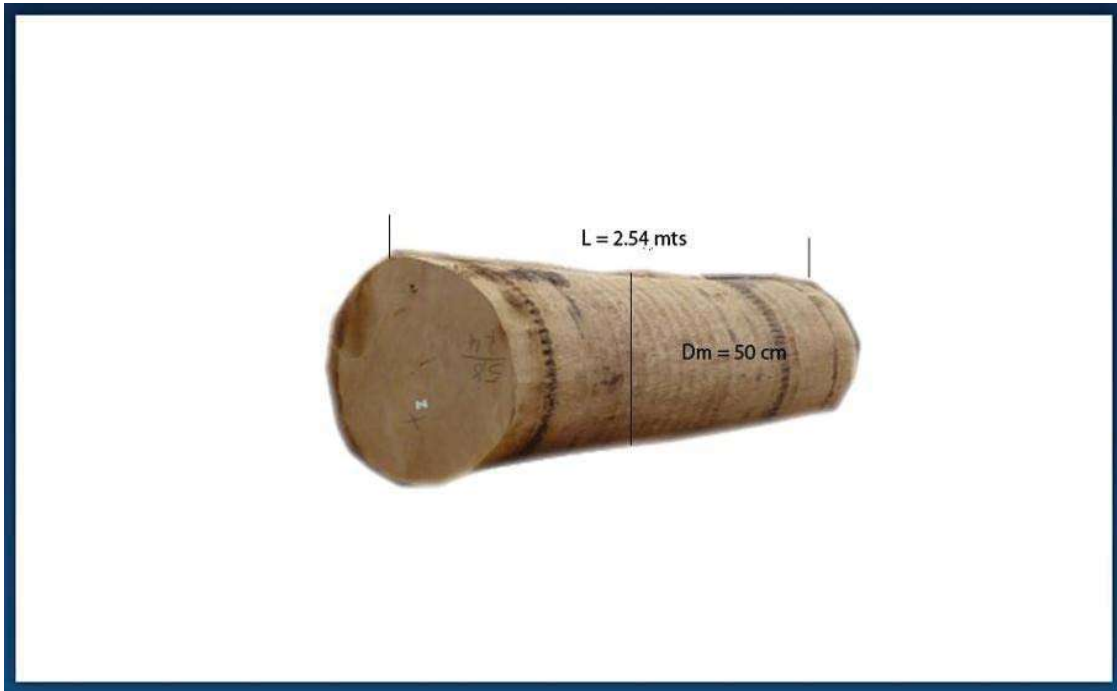


Figura 3. Madera en rollo que se desea cubicar empleando la fórmula de Huber.

$D_m = 50 \text{ cm} = 0.50 \text{ m}$, en donde $A = 0.7854 (D_m)^2$
 $L = 2.54 \text{ m}$

Sustituyendo en la fórmula:

$$V = A_m \times L \Rightarrow V = 0.7854 (0.5 \text{ m})^2 \times 2.54 = 0.491 \text{ m}^3$$

El volumen = **0.499 m³**

4.6.2.2. Fórmula de Smalian.

Sirve para calcular el volumen de piezas de madera en rollo, tomando como base el promedio de las superficies o áreas de sus extremos y su longitud:

$$V = \left(\frac{A + a}{2} \right) \times L$$

Donde:

V = Volumen (m^3).

A = Área del extremo mayor de la troza (m^2).

a = Área del extremo menor de la troza (m^2).

L = Longitud de la troza (m).

Ejemplo: Se desea cubicar la madera en rollo que se muestra en la figura 4.

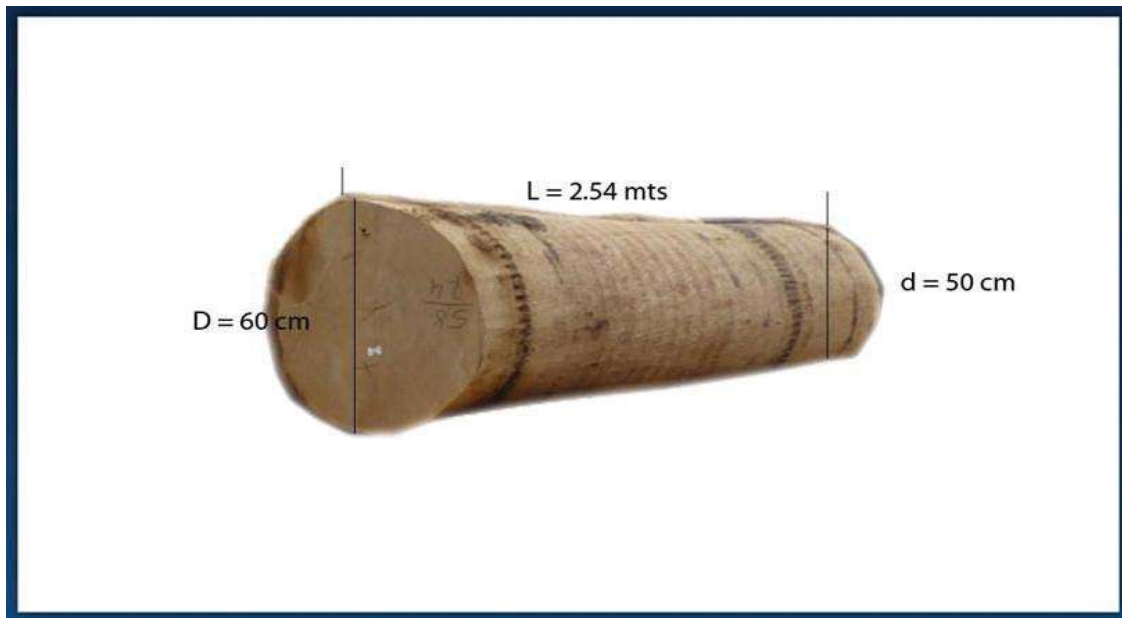


Figura 4. Madera en rollo que se desea cubicar empleando la fórmula de Smalian.

$$D = 60 \text{ cm} = 0.60 \text{ m} \text{ donde } A = 0.7854D^2 \Rightarrow A = 0.7854 (0.60)^2 = 0.283 \text{ m}^2$$

$$d = 50 \text{ cm} = 0.50 \text{ m} \text{ donde } a = 0.7854d^2 \Rightarrow a = 0.7854 (0.50)^2 = 0.196 \text{ m}^2$$

$$L = 2.54 \text{ m}$$

Sustituyendo en la fórmula:

$$V = \left[\frac{A + a}{2} \right] \times L \Rightarrow V = \left[\frac{0.283 \text{ m}^2 + 0.196 \text{ m}^2}{2} \right] \times 2.54 \text{ m} = 0.608 \text{ m}^3$$

El volumen = **0.608 m³**

4.6.2.3. Fórmula de Newton.

Sirve para calcular el volumen de piezas de madera en rollo, tomando como base las áreas de sus extremos, el área de la sección media y su longitud. Esta fórmula es más exacta que la de Huber y Smalian, sin embargo, es la que menos se utiliza en la práctica:

$$V = \left(\frac{A + 4 A_m + a}{6} \right) \times L$$

Donde:

V = Volumen (m^3).

A = Área del extremo mayor de la troza (m^2).

A_m = Área de la sección media de la troza (m^2).

a = Área del extremo menor de la troza (m^2).

L = Longitud de la troza (m).

Ejemplo: Se desea cubicar la madera en rollo que se muestra en la figura 5.

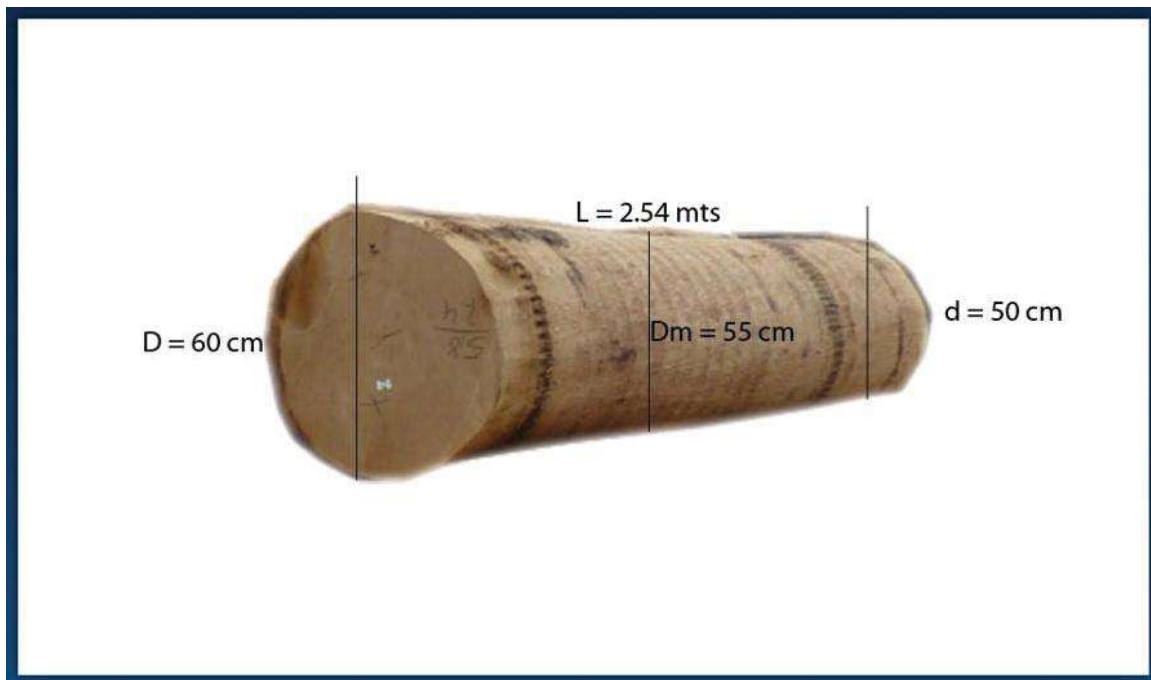


Figura 5. Madera en rollo que se desea cubicar empleando la fórmula de Newton.

$D = 60 \text{ cm} = 0.60 \text{ m}$ donde $A = 0.7854 (D)^2 \Rightarrow A = 0.7854 (0.60)^2 = 0.283 \text{ m}^2$
 $D_m = 55 \text{ cm} = 0.50 \text{ m}$ donde $A_m = 0.7854 (D_m)^2 \Rightarrow A_m = 0.7854(0.55)^2 = 0.237 \text{ m}^2$
 $d = 50 \text{ cm} = 0.50 \text{ m}$ donde $a = 0.7854 (d)^2 \Rightarrow a = 0.7854 (0.50)^2 = 0.196 \text{ m}^2$
 $L = 2.54 \text{ m}$

Sustituyendo en la fórmula:

$$V = \left(\frac{A + 4 A_m + a}{6} \right) \times L \Rightarrow V = \left(\frac{0.283 \text{ m}^2 + 4 (0.237 \text{ m}^2) + 0.196 \text{ m}^2}{6} \right) \times 2.54 \text{ m}$$

$$= 0.604 \text{ m}^3$$

El volumen = **0.604 m³**

4.6.2.4. Equivalencias empleadas en la cubicación de la madera en rollo.

Al cubicar la madera en rollo es importante contar con equivalencias que nos permitan convertir de un sistema de unidades a otro, ya sea del sistema métrico decimal al sistema ingles o viceversa, según convenga al momento de la transformación, transporte o comercialización (Cuadro 7).

Cuadro 7. Equivalencias empleadas en la cubicación de la madera en rollo.

1 m ³	⇒	211.86 pies rollo
1 millar pies rollo	⇒	4.71 m ³

Fuente: Centro de investigación del Pacífico Centro Michoacán, 1993.

4.6.3. Cubicación por apilamiento.

4.6.3.1. Madera en rollo cortas dimensiones (Trocito o bolo).

Además de cubicarse pieza por pieza con las fórmulas de Huber, Smalian y Newton, también se acostumbra hacer apilamientos de trozas de cortas dimensiones con o sin corteza, con diámetro mínimo de 15 cm. Para su cubicación, se obtiene primero el volumen total del apilamiento (volumen aparente), para lo cual se multiplican las medidas de **ancho** por el **largo** por el **alto** del mismo. Al valor resultante se le aplica un factor o **coeficiente de apilamiento** que usualmente varía desde **0.65**, si el trozo esta apilado con corteza, hasta **0.75**, si el trozo esta apilado

sin corteza, obteniéndose el volumen real de madera que se encuentra apilada (PROBOSQUE, 1998).

Ejemplo: Se desea cubicar el apilamiento de trocito que se muestra en la figura 6.

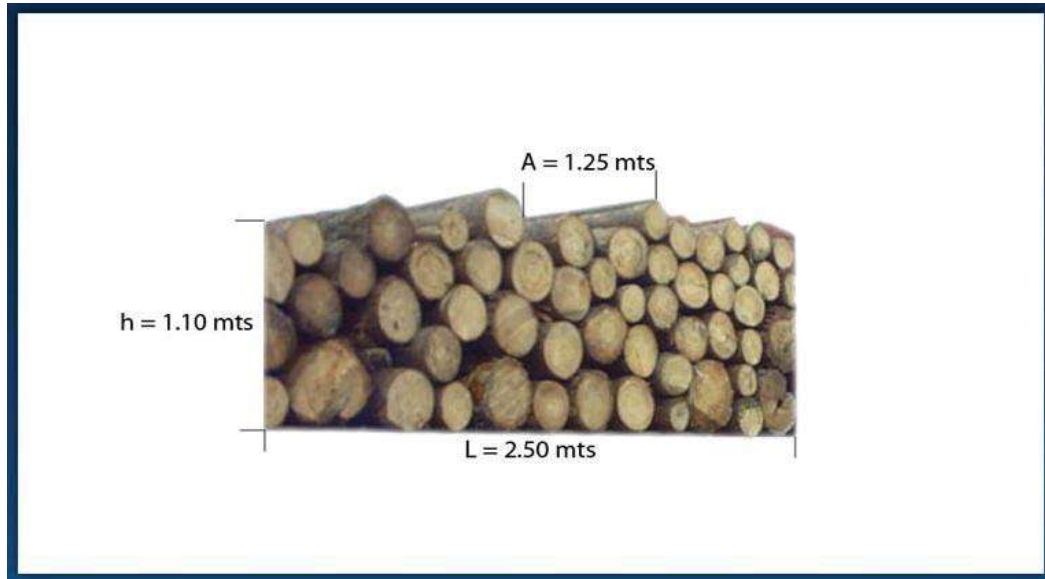


Figura 6. Apilamiento de trocito con corteza que se desea cubicar.

Producto: Trocito con corteza

Dimensiones del apilamiento:

Ancho = 1.25 m

Alto (h) = 1.10 m

Largo = 2.50 m

Sustituyendo en la fórmula:

$$V = \text{Ancho} \times \text{Alto} \times \text{Largo} \times \text{Factor} \Rightarrow V = 1.25 \text{ m} \times 1.1 \text{ m} \times 2.50 \text{ m} \times 0.65 = 2.234 \text{ m}^3$$

El volumen real del apilamiento de trocito con corteza = **2.234 m³**

4.6.3.2. Leña en raja.

Las leñas en raja de la misma longitud, se acomodan en apilamientos de diferentes largos y usualmente la misma altura. La longitud más común de las rajadas es de 61 cm o de 1.22 m. Para obtener el volumen real de la madera contenida en el apilamiento, se procede en la misma forma que el caso anterior, multiplicando el **ancho** por el **largo** por el **alto** del apilamiento, al resultado se le aplica un coeficiente

de apilamiento de **0.7** si la leña se encuentra sin corteza y de **0.6** si se encuentra con corteza, dando como resultado el volumen real de madera apilada (PROBOSQUE, 1998).

Ejemplo: Se desea cubicar el apilado de leña en raja que se muestra en la figura 7.



Figura 7. Apilamiento de leña en raja con corteza que se desea cubicar.

Producto: Leña en raja con corteza

Dimensiones del apilamiento:

Ancho = 1.22 m

Alto (h) = 1.50 m

Largo = 2.60 m

Sustituyendo en la fórmula:

$$V = \text{Ancho} \times \text{Alto} \times \text{Largo} \times \text{Factor} \Rightarrow V = 1.22 \text{ m} \times 1.50 \text{ m} \times 2.60 \text{ m} \times 0.6 = 2.885 \text{ m}^3$$

El volumen real del apilamiento de la leña en raja con corteza = **2.855 m³**

4.6.3.3. Brazuelo.

Los brazuelos son piezas de madera obtenidos de las ramas de los árboles, las cuales tienen diámetros mínimos de 5 cm y longitudes comprendidas entre 1.00 y 1.22 m, siendo esta última medida la más usual. Para calcular el volumen de

madera de un apilamiento de brazuelo, se multiplica el **ancho** por el **largo** por la **altura** del mismo. El volumen aparente obtenido, se multiplica por el **coeficiente de apilamiento** de **0.5** cuando se trata de brazuelo con corteza y por **0.6** para el brazuelo sin corteza, dando como resultado el volumen real de madera apilada (PROBOSQUE, 1998).

Ejemplo: Se desea cubicar el apilamiento de brazuelo que se muestra en la figura 8.

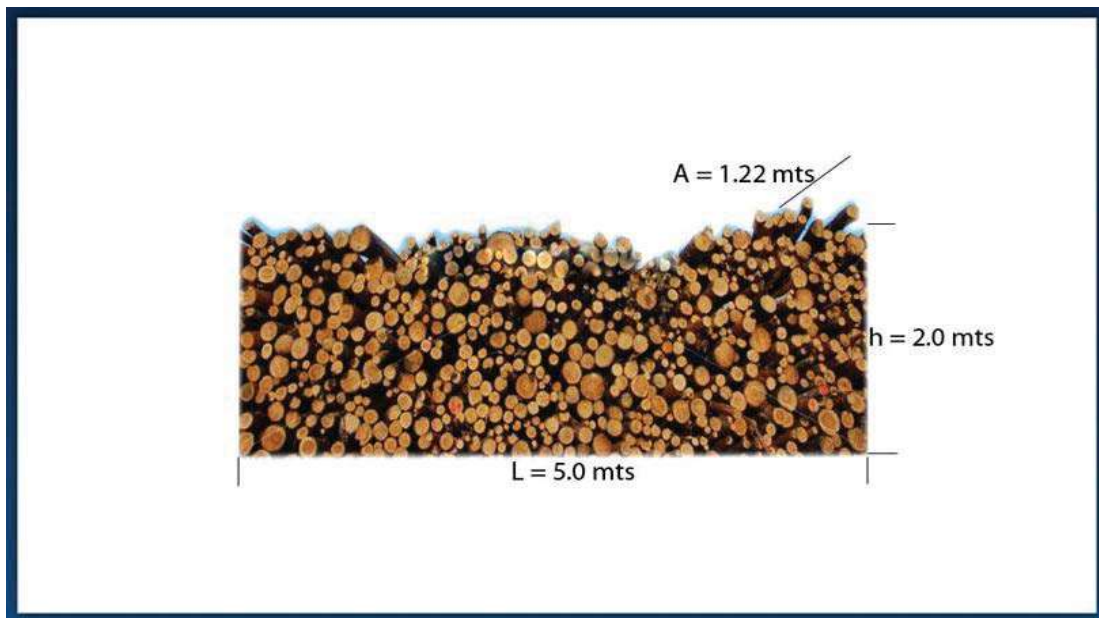


Figura 8. Apilamiento de Brazuelo con corteza que se desea cubicar.

Producto: Brazuelo con corteza

Dimensiones del apilamiento:

Ancho = 1.22 m

Alto (h) = 2.0 m

Largo = 5.0 m

Sustituyendo en la fórmula:

$$V = \text{Ancho} \times \text{Alto} \times \text{Largo} \times \text{Factor} \Rightarrow V = 1.22 \text{ m} \times 2.0 \text{ m} \times 5.0 \text{ m} \times 0.50 = 6.100 \text{ m}^3$$

El volumen real del apilamiento de brazuelo con corteza = **6.100 m³**

4.6.3.4. Astillas para celulosa.

Las astillas para celulosa se obtienen generalmente del material celulósico que se obtiene del árbol (puntas y ramas), las dimensiones las especifican los clientes que generalmente son fabricantes de celulosa y papel así como fabricantes de tableros aglomerados.

Para calcular el volumen de madera de un apilamiento de astillas, se desarrolla la misma metodología que en los productos mencionados anteriormente, multiplicando el **ancho** por el **largo** por la **altura** del lote. El volumen aparente obtenido, se multiplica por el **coeficiente de apilamiento** de **0.7** dando como resultado el volumen real de madera apilada (PROBOSQUE, 1998).

4.6.4. Madera aserrada o con escuadría.

La madera aserrada o madera con escuadría se obtiene de la madera en rollo mediante el proceso de aserrío y se define como una materia prima en cortes angulares proveniente de la vegetación forestal maderable, en cuya elaboración se utilizan equipos mecánicos (SEMARNAT, 2005).

Los tipos de productos obtenidos de la industria del aserrío son los siguientes:

A. Tablas y tablonés.

Las dimensiones comerciales de las tablas y tablonés en cuanto a espesor son de $\frac{1}{2}$ hasta 1 pulgada para las tablas y de $1 \frac{1}{2}$ hasta $2 \frac{1}{2}$ pulgadas para los tablonés, en lo que se refiere al ancho van de 4 hasta 12 pulgadas y en lo referente a las longitudes van de 4 hasta 6 pies en cortas dimensiones y de 8 hasta 18 pies en largas dimensiones (Cuadro 8).

Cuadro 8. Dimensiones comerciales de tablas y tablonés.

Clasificación	Espesor (pulg.)	Ancho (pulg.)	Largo (pies)
Largas dimensiones	$\frac{1}{2}$, $\frac{3}{4}$, 1, $1 \frac{1}{2}$, 2, $2 \frac{1}{2}$	4, 6, 8, 10, 12	8, 10, 12, 14, 16, 18
Cortas dimensiones	$\frac{1}{2}$, $\frac{3}{4}$, 1, $1 \frac{1}{2}$, 2, $2 \frac{1}{2}$	4, 6, 8, 10, 12	4, 6

Fuente: Maderería La Huerta, 2009.

B. Polines.

Las dimensiones comerciales de los polines en cuanto a espesor y ancho van de 3 hasta 4 pulgadas y en lo que respecta a la longitud generalmente son de 8 pies (Cuadro 9).

Cuadro 9. Dimensiones comerciales de polines.

Espesor (pulgadas)	Ancho (pulgadas)	Largo (pies)
3	3	8
3 ¼	3 ¼	8
3 ½	3 ½	8
4	4	8

Fuente: Maderería La Huerta, 2009.

C. Durmientes.

Los durmientes se clasifican en **durmiente aserrado** los cuales tienen espesores de 7 pulgadas, anchos de 8 a 9 pulgadas y longitudes que van de 8 hasta 8 ½ pies y los **juegos de cambios** que tienen espesores de 7 pulgadas, anchos de 9 a 10 pulgadas y longitudes que van de 9 hasta 16 pulgadas (Cuadro 10).

Cuadro 10. Dimensiones comerciales de durmientes.

Clasificación	Espesor (pulg.)	Ancho (pulg.)	Largo (pies)
Durmiente aserrado	7	8	8, 8 ½
	7	9	8 ½
Juegos de cambios	7	9	9 hasta 16
	7	10	12 hasta 14

Fuente: Comité de Medición Forestal, 1999.

D. Vigas.

Para las Vigas sus dimensiones comerciales en cuanto a espesor van de 3 ½ hasta 12 pulgadas, en lo referente al ancho van de 4 hasta 12 pulgadas y sus longitudes generalmente son de 10 hasta 20 pies (Cuadro 11).

Cuadro 11. Dimensiones comerciales de vigas.

Clasificación	Espesor (pulg.)	Ancho (pulg.)	Largo (pies)
Vigueta	3 ½	4, 6, 8, 10, 12	10, 12, 14, 16, 18, 20
Viga	4, 6	4, 6, 8, 10, 12	10, 12, 14, 16, 18, 20
Gualdra	8, 10, 12	4, 6, 8, 10, 12	10, 12, 14, 16, 18, 20

Fuente: Maderería La Huerta, 2009.

4.6.4.1. Métodos de Cubicación.

Existen 2 fórmulas para la cubicación de la madera aserrada o con escuadría, en uno se maneja como unidad el metro cúbico (m³) y en el otro los pies tabla (pt).

Por tradición los centros de almacenamiento y/o transformación (aserraderos, madererías, etc.) utilizan con mayor frecuencia como unidad de medida el pie tabla, sin embargo en la legislación vigente, se establece como unidad de medida el metro cúbico (SEMARNAT, 2005).

1. Fórmula para calcular el volumen en pies tabla (pt):

$$V = \frac{e \times A \times L}{12}$$

Donde:

e = Espesor de la pieza (pulgadas).

A = Ancho de la pieza (pulgadas).

L = Longitud de la pieza (pies).

V = Volumen de la pieza (pt).

Ejemplo: Se desea cubicar el tablón que se muestra en la figura 9.

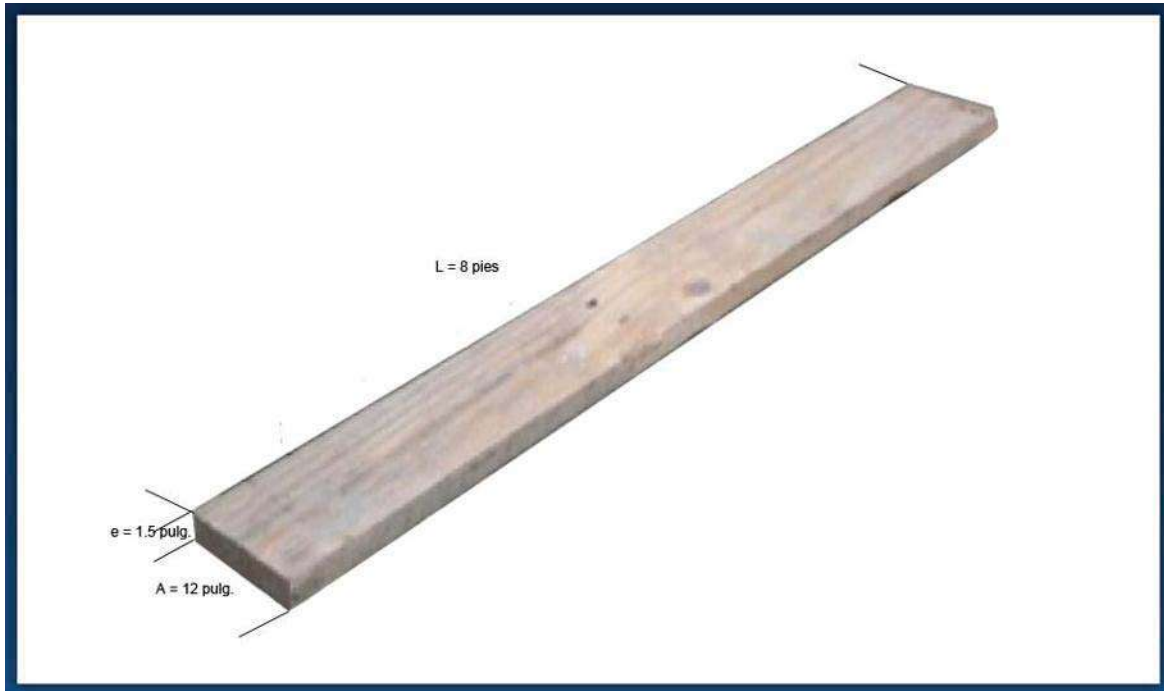


Figura 9. Tablón que se desea cubicar empleando los pies tabla.

Dimensiones de la pieza:

$e = 1.5$ pulg.

$A = 12$ pulg.

$L = 8$ pies

Sustituyendo en la fórmula:

$$V = \frac{e \times A \times L}{12} \Rightarrow V = \frac{(1.5 \text{ pulg.}) \times (12 \text{ pulg.}) \times (8 \text{ pies})}{12} = 12 \text{ pt}$$

El volumen de la pieza = **12 pt**

2. Fórmula para calcular el volumen en metros cúbicos (m^3):

$$V = e \times A \times L$$

Donde:

e = Espesor de la pieza (metros).

A = Ancho de la pieza (metros).

L = Longitud de la pieza (metros).

V = Volumen de la pieza (m^3).

Ejemplo: Se desea cubicar el tablón que se muestra en la figura 10.

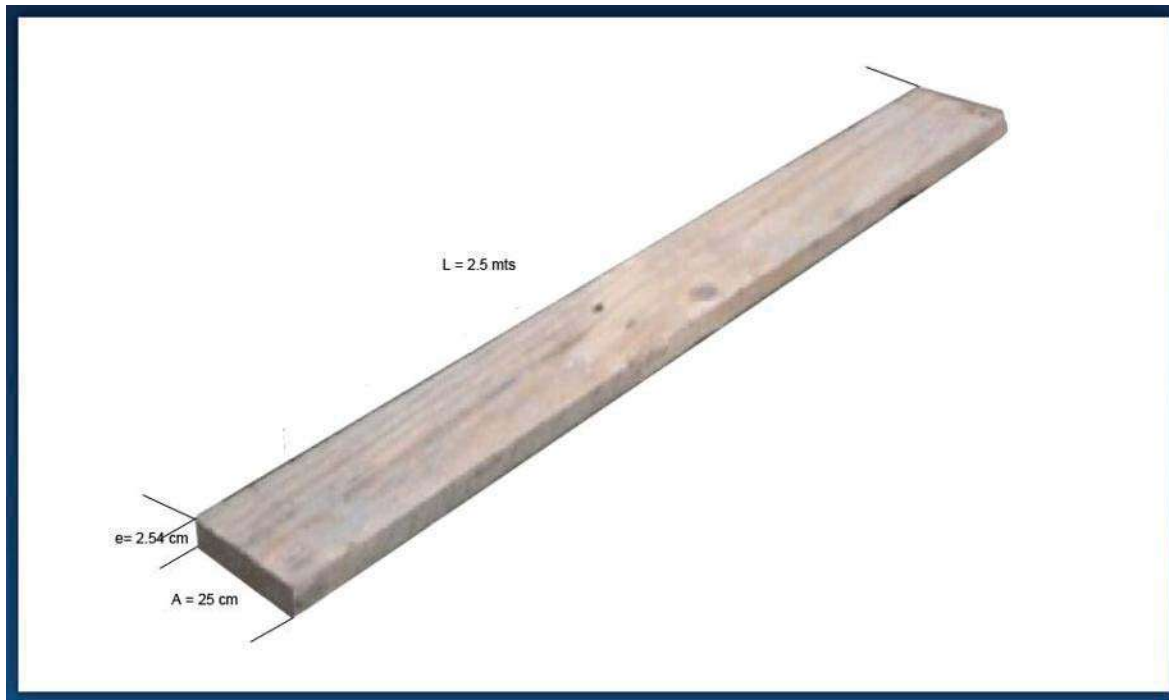


Figura 10. Tabla que se desea cubicar empleando metros cúbicos.

Dimensiones de la pieza:

$$e = 2.54 \text{ cm} = 0.0254 \text{ m}$$

$$A = 25 \text{ cm} = 0.25 \text{ m}$$

$$L = 2.5 \text{ m}$$

Sustituyendo en la fórmula:

$$V = e \times A \times L \Rightarrow V = (0.0254 \text{ m}) \times (0.25 \text{ m}) \times (2.5 \text{ m}) = 0.016 \text{ m}^3$$

El volumen de la pieza = **0.016 m³**

4.6.4.2. Equivalencias empleadas en la cubicación de la madera aserrada.

Al igual que en la madera en rollo, en la madera aserrada o con escuadria existen 2 equivalencias que son de gran utilidad en cualquier etapa del proceso productivo (transformación, comercialización, etc); las cuales nos permiten convertir ya sea de pie tabla (pt) a metros cúbicos (m³) o viceversa (Cuadro 10).

Cuadro 10. Equivalencias empleadas para la cubicación de la madera aserrada.

1 m ³	⇒	424 pie tabla
1 pie tabla	⇒	0.00236 m ³

Fuente: Ramirez y Treviño, 1994.

4.6.5. Embalajes de madera.

Un embalaje de madera se define como: madera o productos de madera, utilizados para sujetar, contener, proteger o transportar bienes, excluidos aquellos que sean de papel (SEMARNAT, 2005).

Los embalajes de madera más comunes son los siguientes:

A. Tarima industrial.

Las tarimas industriales son bases de madera compuestas por un conjunto de piezas de madera aserrada de diferentes dimensiones de acuerdo a las necesidades y requerimientos del cliente (ancho, largo y espesor) que sirven para almacenar y contener productos y artículos varios, con la finalidad de poderlos transportar de un lugar a otro con gran facilidad. La madera empleada para su elaboración debe de estar libre de rajaduras, torceduras, nudos flojos; admite pocos nudos macizos y bolsas de resina (Nava, 1992).

B. Cajas de madera para empaque.

- Cajas de madera alambradas.

Son cajas hechas de madera, alambre y grapas, el hecho de reforzar las cajas con alambre permite reducir el espesor de las tablillas empleadas ahorrando hasta un 20 % en la madera utilizada.

Estas cajas se entregan sin montar lo cual permite almacenarse y transportarse fácilmente, estas soportan de 10 hasta 200 kg, son destinadas para el empaque de frutas y legumbres, para el transporte de pollos y para los llamados empaques industriales, preferentemente de muebles para el hogar y acumuladores (Enríquez, 1980).

- Cajas de madera clavada.

Son cajas compuestas por madera y clavos, éstas se caracterizan por que son de uso múltiple y a la vez reutilizable, su uso principal es en el empaque de la producción hortofrutícola de consumo nacional. Ofrecen resistencia al agujeramiento, aplastamiento o magulado por las esquinas, son más pesadas, pieza por pieza, que las cajas alambradas. Estos productos tienen la propiedad de mantener fresco el producto envasado manteniéndolo en condiciones óptimas (Cacarí, 1993).

4.6.5.1. Método de cubicación.

Este se lleva a cabo mediante el **despiece**, el cual consiste en sumar todas y cada una de las piezas que conforman el embalaje de madera (tarimas o cajas), empleándose la misma metodología y unidades que se utilizan en la cubicación de madera aserrada o con escuadría (Cacarí, 1993).

Ejemplo: Se pretende determinar el volumen de la tarima industrial que se muestra en la figura 11.



Figura 11. Tarima industrial que se desea cubicar empleando pies tabla.

Despiece:

Clave	Descripción	No. De piezas	Dimensiones e x A x L (pulgadas)	Volumen (pt)
A	Van al centro de la tarima	3	1 3/4" *3 3/4" *48"	6.563
B	Van en la cara inferior	4	3/4" *5" *40"	4.163
C	Van en la cubierta	7	3/4" *4" *40"	5.828
Total		14		16.554

4.6.6. Tableros contrachapados o Triplay.

El tablero contrachapado o triplay, son tableros conformados de un número impar de chapas de madera, estando la dirección de la fibra de cada chapa en ángulos perpendiculares a las adyacentes, la vista y la trascara y todas las capas numeradas en non, generalmente están orientadas con la dirección de la fibra paralela a la dimensión larga del tablero, las chapas son unidas bajo presión por un adhesivo para crear un tablero con una unión adhesiva tan fuerte o más que la madera. La forma alternada de la dirección de la fibra de cada chapa contigua y el número impar de chapas equilibra las deformaciones, minimiza contracciones, pandeos y rajaduras del tablero. Se utilizan mucho en la elaboración de muebles de todo tipo, así como en la construcción ya que es de gran utilidad debido a sus dimensiones y estabilidad dimensional (Triplay de Michoacán, 2003).

Las dimensiones comerciales de los tableros son:

- 1.22 m de ancho x 2.44 m de largo

En menor escala:

- 0.91 m de ancho x 2.44 m de largo
- 0.91 de ancho x 2.14 de largo.

Los espesores comerciales son: 3 mm, 6 mm, 9mm, 12 mm, 16mm y 19mm.

4.6.6.1. Método de cubicación.

Para el caso de los tableros contrachapados o triplay en la práctica se utiliza el metro cúbico (m^3) como unidad de medida de volumen, aplicándose la misma metodología que con la madera aserrada. Cabe mencionar, que derivado de una experiencia profesional desarrollada en la Fábrica "Triplay de Michoacán S. de R. L.

de C. V.” y basado en sus políticas internas, se puso en práctica una metodología que utilizan para llevar el control de la producción del triplay, a lo que denominan como **m² base 5.2 mm** (metros cuadrados base 5.2 mm).

Considerando que para la medición de la producción es muy difícil e incorrecto sumar piezas u hojas de triplay, ya que no es lo mismo producir una hoja de 3 mm a una hoja de 19 mm; por ésta razón se estableció una unidad de medición común y equivalente para poder sumar la producción; estableciéndose como base el triplay de 5.2 mm (6 mm comercial) de espesor (Triplay de Michoacán, 2003). Dando como resultado la fórmula que se describe a continuación:

$$V = (e \times A \times L) / 5.2 \text{ mm}$$

Donde:

e = Espesor de la pieza (milímetros).

A = Ancho de la pieza (metros).

L = Longitud de la pieza (metros).

V = Volumen de la pieza (m² base 5.2 mm).

Ejemplo: Se desea determinar el volumen de la hoja de triplay que se muestra en la figura 12.



Figura 12. Hoja de triplay que se desea cubicar empleando m² base 5.2 mm.

$$\begin{aligned}e &= 3 \text{ mm} \\A &= 1.22 \text{ m} \\L &= 2.44 \text{ m}\end{aligned}$$

Sustituyendo en la formula:

$$V = \frac{(e \times A \times L)}{5.2} \Rightarrow V = \frac{(3 \text{ mm} \times 1.22 \text{ m} \times 2.44 \text{ m})}{5.2 \text{ mm}} = 1.717 \text{ m}^2 \text{ base } 5.2 \text{ mm}.$$

El volumen de la hoja de triplay = **1.717 m² base 5.2 mm.**

5. LENGUAJE DE PROGRAMACION C.

5.1. Definición de Software.

Conjunto de instrucciones y datos codificados para ser leídas e interpretadas por una computadora. Estas instrucciones y datos son concebidos para el procesamiento electrónico de datos (Wiktionary, 2010).

5.2. Características de Lenguaje C.

El lenguaje C se conoce como un lenguaje compilado. Existen dos tipos de lenguaje: interpretados y compilados. Los interpretados son aquellos que necesitan del código fuente para funcionar (P.ej: Basic). Los compilados convierten el código fuente en un fichero objeto y éste en un fichero ejecutable. Este es el caso del lenguaje C.

Podemos decir que el lenguaje C es un lenguaje de nivel medio, ya que combina elementos de lenguaje de alto nivel con la funcionalidad del lenguaje ensamblador. Es un lenguaje estructurado, ya que permite crear procedimientos en bloques dentro de otros procedimientos. Hay que destacar que el C es un lenguaje portable, ya que permite utilizar el mismo código en diferentes equipos y sistemas informáticos: el lenguaje es independiente de la arquitectura de cualquier máquina (UNAM, 2007).

5.3. Estructura de un programa en C.

Estructura.

Todo programa en C consta de una o más funciones, una de las cuales se llama **main**. El programa comienza en la función main, desde la cual es posible llamar a otras funciones. Cada función estará formada por la cabecera de la función, compuesta por el nombre de la misma y la lista de argumentos (si los hubiese), la declaración de las variables a utilizar y la secuencia de sentencias a ejecutar.

Comentarios.

A la hora de programar es conveniente añadir comentarios (cuantos más mejor) para poder saber qué función tiene cada parte del código, en caso de que no lo utilicemos durante algún tiempo. Además se facilitará el trabajo a otros programadores que puedan utilizar el archivo fuente. Para poner comentarios en un programa escrito en **C** usamos los símbolos */** y **/*.

Identificadores.

Un identificador es el nombre que damos a las variables y funciones. Está formado por una secuencia de letras y dígitos, aunque también acepta el carácter de subrayado `_`. Por el contrario no acepta los acentos ni la ñ/Ñ (UNAM, 2007).

5.4. Tipos de datos.

Tipos.

En 'C' existen básicamente cuatro tipos de datos, aunque como se verá después, podremos definir nuestros propios tipos de datos a partir de estos cuatro. A continuación se detalla su nombre, el tamaño que ocupa en memoria y el rango de sus posibles valores (Cuadro 13).

Cuadro 13. Tipo de datos.

Tipo	Tamaño (byte)	Rango de valores
Char	1	-128 a 127
Int	2	-32768 a 32767
Float	4	3,4 E-38 a 3,4 E+38
Double	8	1,7 E-308 a 1,7 E+308

Fuente: Monografías, 2007.

Las variables.

Una variable es un tipo de dato, referenciado mediante un identificador (que es el nombre de la variable). Su contenido podrá ser modificado a lo largo del programa. Una variable sólo puede pertenecer a un tipo de dato. Para poder utilizar una variable, primero tiene que ser declarada (Monografías, 2007).

Sintaxis: `<tipo_de_dato> <variable> [= <expresión>];`

```
int numero = 35;
```

Constantes.

Al contrario que las variables, las constantes mantienen su valor a lo largo de todo el programa. Para indicar al compilador que se trata de una constante, usaremos la directiva **#define** (Monografías, 2007).

Sintaxis: **#define** <constante> <secuencia_de_caracteres>

```
#define PI 3.141592
```


Secuencias de escape.

Ciertos caracteres no representados gráficamente se pueden representar mediante lo que se conoce como secuencia de escape (Cuadro 14).

Cuadro 14. Secuencias de escape.

<code>\n</code>	salto de línea
<code>\b</code>	Retroceso
<code>\t</code>	tabulación horizontal
<code>\v</code>	tabulación vertical
<code>\\</code>	Contrabarra
<code>\f</code>	salto de página
<code>\'</code>	Apóstrofe
<code>\"</code>	comillas dobles
<code>\0</code>	fin de una cadena de caracteres

Fuente: Monografías, 2007.

5.5. Operadores aritméticos y de asignación.

Los datos se manipulan mediante expresiones, que sirven para calcular valores. En C hay varios operadores para construir expresiones (Cuadro 15).

Cuadro 15. Operadores aritméticos.

Binarios	
<code>+</code>	Suma
<code>-</code>	Resta
<code>*</code>	Multiplicación
<code>/</code>	División
<code>%</code>	Módulo (resto)
Unitarios	
<code>++</code>	Incremento (suma 1)
<code>--</code>	Decremento (resta 1)
<code>-</code>	Cambio de signo

Fuente: Monografías, 2007.

La mayoría de los operadores aritméticos binarios explicados anteriormente tienen su correspondiente operador de asignación (Cuadro 16).

Cuadro 16. Operadores de asignación.

=	Asignación simple
+=	Suma
-=	Resta
*=	Multiplicación
/=	División
%=	Módulo (resto)

Fuente: Monografías, 2007.

Será importante tener en cuenta la prioridad o importancia de los operadores a la hora de trabajar con ellos (Cuadro 17).

Cuadro 17. Prioridad de los operadores

()	Mayor prioridad
++, --	
*, /, %	
+, -	Menor prioridad

Fuente: Monografías, 2007.

5.6. Salida / Entrada.

Sentencia printf(); La sentencia printf permite la aparición de valores numéricos, caracteres y cadenas de texto por pantalla.

Sentencia scanf(); La sentencia scanf permite asignar a una o más variables, uno o más valores (datos) recibidos desde la entrada estándar o teclado (Santos, 2007).

5.7. Operadores relacionales.

Los operadores relacionales se utilizan para comparar el contenido de dos variables. En C existen seis operadores relacionales básicos (Cuadro 18).

Cuadro 18. Operadores relacionales.

>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
==	Igual que
!=	Distinto que

Fuente: Monografías, 2007.

5.8. Sentencias condicionales.

Este tipo de sentencias permiten variar el flujo del programa en base a unas determinadas condiciones (Santos, 2007).

Existen estructuras diferentes, a continuación se citan algunas:

- **Estructura IF...ELSE:**

Se trata de una instrucción alternativa doble (o simplemente alternativa doble); permite seleccionar, por medio de una condición, el siguiente bloque de instrucciones a ejecutar, de entre dos posibles.

- **Estructura SWITCH:**

Se trata de una instrucción alternativa múltiple (o simplemente alternativa múltiple) permite seleccionar, por medio de una expresión, el siguiente bloque de instrucciones a ejecutar de entre varios posibles.

5.9. Operadores lógicos.

Para elaborar condiciones complejas, existen los llamados operadores lógicos (Cuadro 19).

Cuadro 19. Operadores lógicos.

&&	AND
	OR
!	NOT (El valor contrario)

Fuente: Monografías, 2007.

5.10. Bucles.

Los bucles son estructuras que permiten ejecutar partes del código de forma repetida mientras se cumpla una condición. Esta condición puede ser simple o compuesta de otras condiciones unidas por operadores lógicos (Santos, 2007)

A continuación se mencionan algunos:

- **Sentencia WHILE**

Una instrucción repetitiva **while** permite ejecutar, repetidamente, (cero o más veces) un bloque de instrucciones, mientras que, una determinada condición sea verdadera.

- **Sentencia DO... WHILE**

Una instrucción repetitiva **do...while** permite ejecutar repetidamente (una o más veces) un bloque de instrucciones, mientras que, una determinada condición sea verdadera.

- **Sentencia FOR**

El bucle **for** es ideal usarlo cuando, de antemano, ya se sabe el número de veces (iteraciones) que tiene que ejecutarse un determinado bloque de instrucciones.

5.11. Funciones.

Las funciones son bloques de código utilizados para dividir un programa en partes más pequeñas, cada una de las cuáles tendrá una tarea determinada.

Sintaxis:

```
tipo _ función nombre_ función (tipo y nombre de argumentos)
      {
      bloque de sentencias      }
```

Tipo_ función: puede ser de cualquier tipo de los que ya se conocen. El valor devuelto por la función será de este tipo. Por defecto, es decir, si no indicamos

el tipo, la función devolverá un valor de tipo entero (int). Si no queremos que retorne ningún valor se deberá indicar el tipo vacío (void).

Nombre_función: es el nombre que le daremos a la función.

Tipo y nombre de argumentos: son los parámetros que recibe la función. Los argumentos de una función no son más que variables locales que reciben un valor. Este valor se lo enviamos al hacer la llamada a la función. Pueden existir funciones que no reciban argumentos.

Bloque de sentencias: es el conjunto de sentencias que serán ejecutadas cuando se realice la llamada a la función (Monografías, 2007)

5.12. Estructuras.

Una estructura es un conjunto de una o más variables, de distinto tipo, agrupadas bajo un mismo nombre para que su manejo sea más sencillo. Su utilización más habitual es para la programación de bases de datos, ya que están especialmente indicadas para el trabajo con registros o fichas.

Sintaxis:

```
struct tipo_estructura
{
    tipo_variable nombre_variable1;
    tipo_variable nombre_variable2;
    tipo_variable nombre_variable3;
};
```

Donde **tipo_estructura** es el nombre del nuevo tipo de dato que hemos creado. Por último, **tipo_variable** y **nombre_variable** son las variables que forman parte de la estructura (Monografías, 2007).

6. METODOLOGÍA.

6.1. Modelo incremental.

La metodología que se tomó como referencia para realizar el programa se llama “**Modelo Incremental**” (Un modelo de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas). En la Figura 13 se representa de una forma gráfica este ciclo (Universidad de Concepción, 2006).

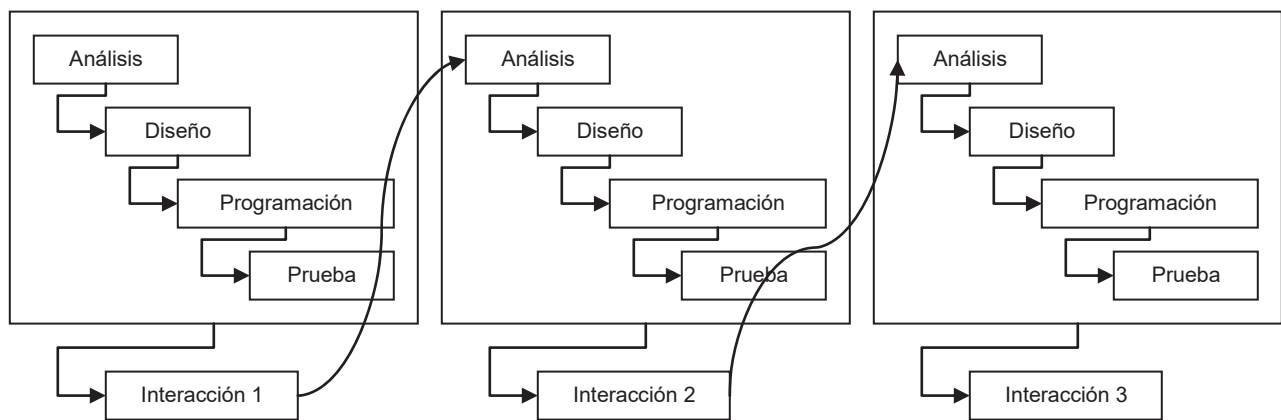


Figura 13. Flujograma del Modelo Incremental.

Es importante tener una idea de cómo será el programa lo antes posible. De este modo, y a fin de disminuir las expectativas como cliente o usuario, se pueden ir desarrollando módulos con sus funcionalidades, para que se puedan ir probando. Esto permite ir aumentando gradualmente la capacidad del software, también permite hacer las sugerencias de cambios en etapas lo más tempranas posibles.

Por otra parte, fue elegido por que es pertinente saber cuanto antes si el producto es tal cómo se interpreta, si está satisfaciendo las necesidades y consideraciones. Así, si no se puede dar una idea acabada de lo que se desea, en cada módulo realizado, permite efectuar una revisión de los requerimientos y un refinamiento de los mismos a fin de aproximarse al producto final con un acercamiento de lo que se tiene en mente.

Este modelo consta de cuatro etapas las cuales son: **análisis, diseño, codificación o programación y prueba**. A continuación se describe cada una de estas etapas:

6.1.1. Análisis.

Extraer los requisitos de un producto de software fue la primera etapa para crearlo. En esta parte se reconocen todos los objetivos que debe cubrir y los requisitos incompletos, ambiguos o contradictorios, para evitar conflictos en una etapa posterior. Una vez identificadas las necesidades del producto, se realiza la parte medular del programa que es la cubicación de las materias primas y productos forestales maderables con las fórmulas comerciales aplicadas para cada uno de estos, realizando los cálculos de una manera correcta y sistemática además de tener contemplados los formatos de los cuadros de menú requeridos. Se utilizó una programación estructurada para su mejor funcionamiento y si tiene algún error este pueda ser detectado y corregido de forma sencilla.

6.1.2. Diseño.

Primero se determina como funcionará el programa de forma general sin entrar en detalles. En esta etapa se "diseñó" el formato, determinando los módulos que lo componen y su apariencia, de acuerdo a una jerarquía apropiada, así como cuál es la función principal del programa y como van a estar relacionadas las demás funciones con la principal (Figura 14).

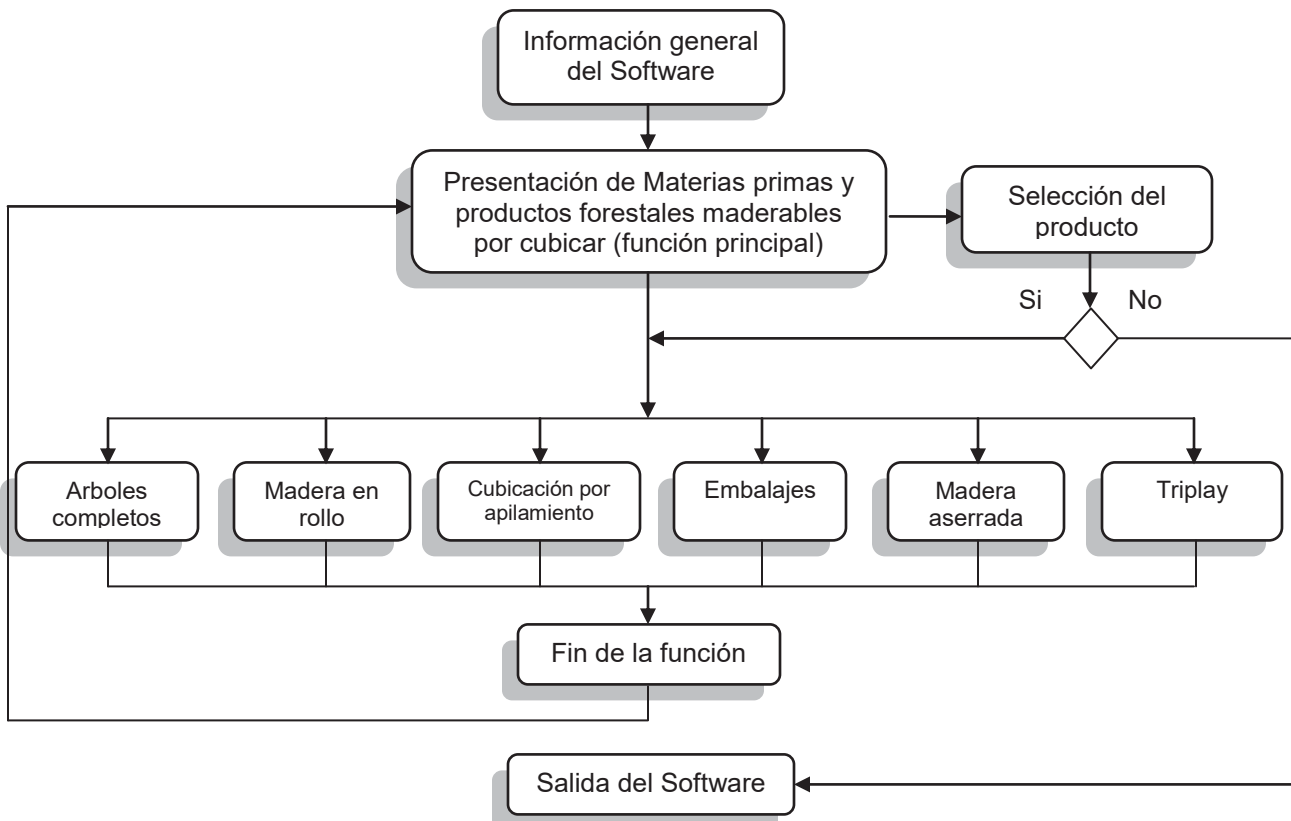


Figura 14. Diseño del programa

6.1.3. Programación o Codificación.

Reducir un diseño a código puede ser la parte más obvia del trabajo, pero no es necesariamente la porción más larga. En esta parte se traduce el comportamiento que se especificó en la etapa de diseño al código de programación.

El tipo de programación utilizada es la programación estructurada: la orientación de esta programación, se dirige hacia los procesos que intervienen en el sistema a desarrollar, es decir, cada función a realizar por el sistema se descompone en pequeños módulos individuales. Esto se debe a que es más fácil resolver problemas pequeños y luego unir cada una de las soluciones que aborda un problema grande. Y este tipo de programación se adapta de muy buena manera, ya que en el análisis se planificó hacer la descomposición del problema en módulos. A continuación se explica de una manera breve cual fue el desarrollo al momento de hacer la codificación del programa.

Primero se descargó e instaló el programa **Turbo C++ 3.0**, donde se desarrolló el Software para después crear el archivo **CUBICA 1**, llevándose a acabo de igual forma que en otros programas como Word, Excel, entre otros; para posteriormente desarrollar el código de programación, realizándose los siguientes pasos:

6.1.3.1. Descarga e instalación del programa Turbo C++ 3.0.

- Este programa se descargó de manera gratuita de la página web www.top4download.com.
- Ya descargado, se descomprime el archivo en la ruta: **C:/TC**
- Se ejecuta el programa: **INSTALL.EXE**, al aparecer la ventana correspondiente se presiona la tecla **Intro**, apareciendo otra ventana.
- Para continuar se escribe la letra de la unidad donde se encuentran los ficheros de instalación. Si se encuentra en la unidad C, se pulsa **C**, seguido de presionar **Intro**.
- A continuación se muestra otra ventana en la que aparece por defecto el directorio donde se encuentran los archivos de instalación, seguido de presionar **Intro** para continuar.
- La siguiente ventana nos indica en primer lugar el directorio donde se instalará Turbo C++ 3.0, que siguiendo el ejemplo será: C: \TC, con los cursores nos

desplazamos hasta la opción **Start Installation** y se presiona **Intro**, iniciando la instalación.

- Mientras se instala, en la parte inferior de la ventana aparecerán los ficheros que se están copiando.
- Una vez copiados todos los archivos saldrá un cuadro que nos informa de los últimos pasos de la instalación. Presionando **Intro** para continuar.
- Tras pulsar Intro, aparece el contenido del fichero “**leeme**” de Turbo C++ 3.0. Para finalizar presionamos **Escape**, concluyendo nuestra instalación.

6.1.3.2. Creación del archivo para el programa.

- Entrar a Turbo C++ 3.0 que se encuentra instalado en nuestro equipo de cómputo.
- Entrar en la opción **FILE** y presionar **New** (nuevo).
- Después de haber trabajado se entra a la opción **FILE** nuevamente y se guarda presionando **Save as**.
- Al presionar la opción **Save as** se deberá poner el nombre del archivo que en este caso se llamó “**CUBICA 1**”.
- Cuando se vuelva a trabajar en el mismo archivo y se quiera guardar lo que se hizo se debe entrar a **FILE** y presionar **Save** o presionar la tecla **F2**.
- Cuando se quiera salir de Turbo C++ 3.0 se deberá entrar en la opción **FILE** y posteriormente presionar **Quit**. Para salir también se puede presionar la tecla **Alt** y simultáneamente la tecla **X**.

6.1.3.3. Desarrollo del código fuente del Software.

- Se declaran las librerías al principio o a la cabeza del programa por la sintaxis del lenguaje, mismas que nos facilitan el trabajo en la creación de un programa. Después se hizo la declaración de las estructuras usadas en el mismo.
- A continuación se realizó la declaración y desarrollo de las funciones. La función “**void main(void)**”, es la función principal del programa, la cual presenta los datos generales de presentación del programa. Esta llama a la función “**presentación**”, al presionar cualquier tecla.

- La función “**void presentacion(void)**”, muestra todos los tipos de materias primas y productos forestales maderables que va a cubicar el programa mediante un menú, los cuales son: Árboles completos, Madera en rollo, cubicación por apilamiento, Embalajes de madera, Madera con escuadría y Triplay. Cada uno de estos cálculos es hecho en diferentes funciones (Cuadro 20), que son llamadas por medio del número asignado en el menú; y se hace una serie de comparaciones, las cuales consisten en comparar el número seleccionado con los números que corresponden a los productos forestales mencionados anteriormente.
- Para realizar la cubicación de cada producto forestal maderable primeramente se le asignó a cada módulo un nombre de función que representa cada producto.

Cuadro 20. Nombre de la función asignada a cada producto forestal.

Producto forestal maderable	Nombre de función asignada
Árboles completos	arbol()
Madera en rollo	rollo()
Cubicación por apilamiento	lotes()
Embalaje de madera	embalaje()
Madera con escuadría	madera()
Ttriplay	triplay()

- Después se desarrolló la estructura de cada una de las funciones que representan a un producto forestal maderable para realizar las operaciones de cubicación bajo los requerimientos y criterios del mismo, verificando que su funcionamiento sea correcto. Ya que se tuvieron todas las funciones, se integraron en el programa para ver su funcionamiento (ver anexo 1).

6.1.4. Prueba.

Consistió en comprobar que el software realiza correctamente las tareas indicadas (cubicación de las materias primas y productos forestales maderables, así como el flujo del sistema de acuerdo al diseño). La técnica de prueba que se utilizó fue, probar por separado cada módulo del software según se fue implementando, y luego probarlo de forma integral para ver si ya estaba bien el funcionamiento general del programa. Al estar completamente satisfechos con el funcionamiento del Software o programa, se procede a generar el archivo ejecutable, presionando la tecla **F9 (Make)**, en este caso se le denominó **CUBICA 1.exe**, ubicándolo en **Disco local (C:)**, en la Carpeta **Escritorio**. Habiendo generado el archivo ejecutable lo podemos copiar a una memoria USB o a un Disco Compacto para disponer del Software en el momento y lugar que se desee.

7. RESULTADOS.

7.1. Entrada al programa.

Para entrar al Software para la cubicación de materias primas y productos forestales maderables. Para el caso de Windows, primeramente se presiona **Inicio**, posteriormente se selecciona la opción de Mi PC.

Después de haber seleccionado Mi PC, se dirige a la unidad en donde se tenga guardado el software; para este caso se tiene en el **Disco local (C:)**, en la Carpeta **Escritorio**, en donde aparece el archivo ejecutable **CUBICA 1.exe**; en caso de tenerlo en una memoria USB o en un Disco Compacto, se accede a la unidad respectiva. Para ejecutarlo se hace doble click en el Icono del archivo, para así acceder al Software para la cubicación de materias primas y productos forestales maderables (Figura 15).

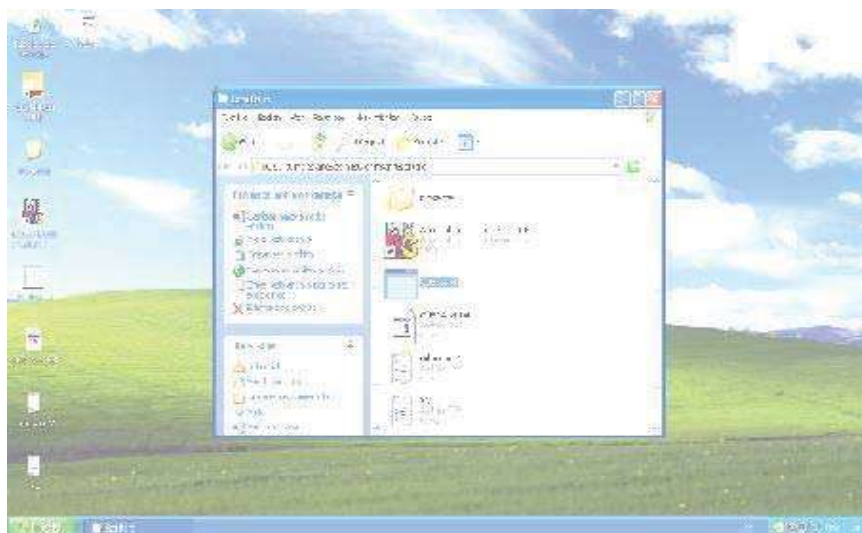


Figura 15. Entrada al Software.

7.2. Forma de uso del Software.

7.2.1. Presentación del Software.

En este cuadro de dialogo se muestran los datos generales del Software o Programa como: Universidad y Facultad donde fue elaborado, el nombre del programa y quien lo elaboró, principalmente (Figura 16).

Para continuar se presiona cualquier tecla del equipo de computo donde se está ejecutando el Software.

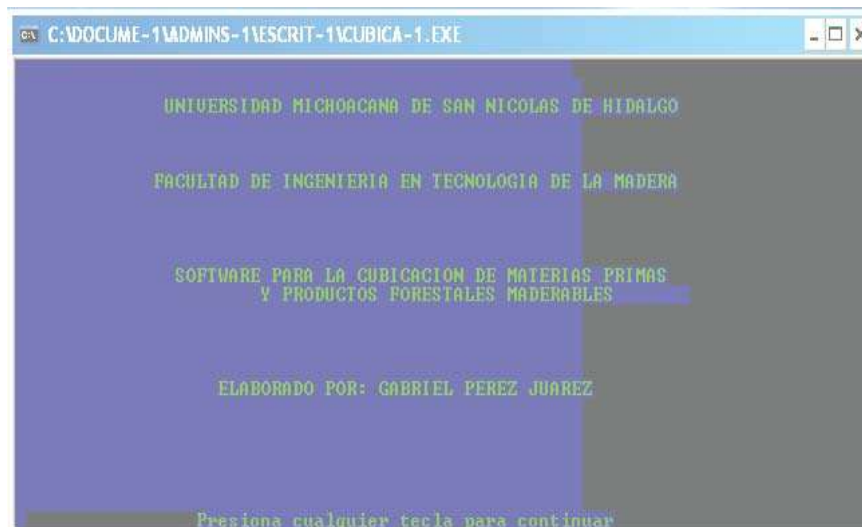


Figura 16. Ventana de presentación del Software.

7.2.2. Presentación de productos forestales maderables.

En esta ventana del cuadro de menú se enumeran las materias primas y productos forestales maderables que cubica el Software (Figura 17).

Para continuar, se pone el número correspondiente del producto que se desea cubicar de las 7 posibles opciones al lado del enunciado que se encuentra en la parte inferior del cuadro de dialogo **“Dame el número del producto que deseas cubicar”** seguido de presionar la tecla enter.

Si se desea salir del Software se debe poner el número 8 seguido de presionar la tecla enter y posteriormente hacer click en el icono en forma de cruz que se encuentra en la parte superior derecha de la ventana.

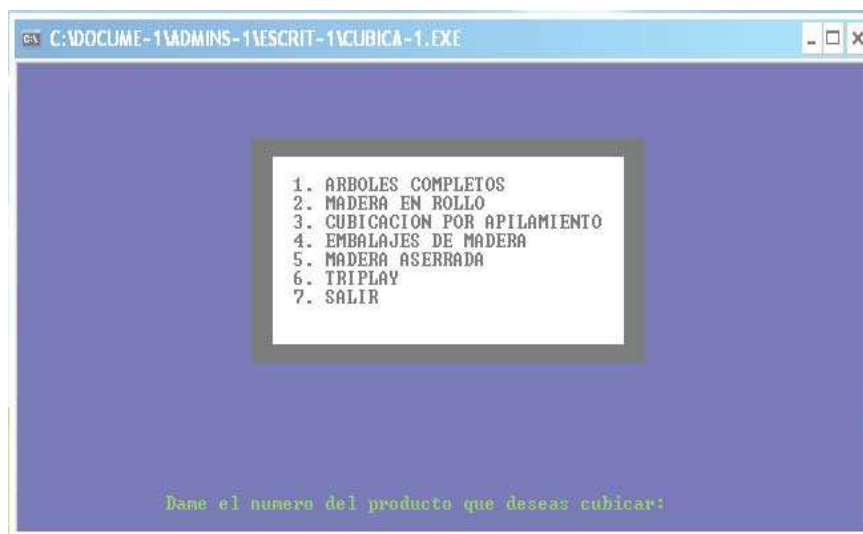


Figura 17. Ventana del menú de productos forestales maderables.

7.2.3. Cubicación de árboles completos.

En este cuadro de menú se muestran las posibles opciones que se tienen al escoger árboles completos, las cuales son cinco: Tocón, Puntas, fustes, Árboles completos y Salir (Figura 18).

Se elaboró de esta manera debido a que la suma de Tocón, Fuste y Punta nos da el volumen de un árbol completo (ver marco de referencia); dándose así la ventaja de calcular el volumen del árbol completo o bien de manera independiente cualquiera de las partes que conforman el mismo, escogiendo la opción que se quiera.

La opción cinco es para salir del cuadro de menú de cubicación de árboles completos mandando a la ventana principal de productos forestales maderables.

Para continuar se pone el número de la opción que se escogió al lado del enunciado “**Dame el número de producto que deseas**” seguido de presionar la tecla enter.



Figura 18. Ventana del menú para cubicar árboles completos.

En la figura 18 se muestra la ventana para cubicar los árboles completos o en su defecto las partes que lo conforman, en donde los datos de entrada cambiarán de acuerdo a la opción que se escoja como se explica a continuación:

Si se escoge la opción número 2 “**Tocón**” lo que se tiene que hacer es introducir dos datos de entrada que son el **diámetro del tocón (D) en metros** seguido de presionar la tecla enter posteriormente se pone la **longitud del tocón (L) en metros** seguido de presionar la tecla enter, dando como resultado el volumen del tocón en metros cúbicos (Figura 19).

Al escoger la opción 3 “**Fuste**” aparecerá el enunciado “**dame el número de diámetros del fuste**” poniéndose el número de diámetros en que fue seccionado el fuste por cubicar, tomando en cuenta que la distancia entre cada sección debe ser igual, seguido de presionar la tecla enter (ver marco de referencia). Posteriormente aparecerá el cursor debajo de la letra **(D)** que se encuentra debajo de la palabra fuste del formato en donde se irán poniendo el **diámetro (D) en metros** de cada sección seguido de presionar la tecla enter en cada caso. Posteriormente se pondrá la **longitud (L) en metros** que separa cada sección siendo igual entre una y otra, seguido de presionar la tecla enter (Figura 19).

Si se escoge la opción número cuatro “**Punta**” lo que se hace es introducir dos datos de entrada: el **diámetro del extremo de la punta (D) en metros** seguido de presionar la tecla enter y posteriormente la **longitud (L) en metros** de la misma seguido de presionar la tecla enter, dando así el volumen de la punta en metros cúbicos.

Si se escoge la opción 1 “**Árboles completos**” se hace exactamente lo mismo que se hizo en las tres opciones anteriores y en el mismo orden obteniéndose así el volumen del árbol completo en metros cúbicos (Figura 19). Si se escoge la opción cinco el programa mandará a la ventana principal de productos forestales maderables.

Después de introducir los datos de entrada correspondientes, dependiendo de la opción que se elija, aparecerá el enunciado “**deseas hacer otro cálculo? s/n**” en la parte inferior de la ventana; si se acepta se debe de **presionar la tecla s** siguiendo con la cubicación de otra pieza de la opción seleccionada, si se rechaza se **presiona la tecla n** en donde saldrá el total de las piezas cubicadas según con su respectivo volumen total según sea el caso, volviendo a presionar la tecla n se saldrá de esta ventana mandando el Software al cuadro de menú de opciones para la cubicación de arboles completos (Figura 18).

ARBOLES COMPLETOS										
No Arbol	Tocon D	L	Fuste D	L	Punta D	L	Uolumen Tocon M3	Uolumen Fuste M3	Uolumen Punta M3	Uolumen Total M3
1	.6	.3	.2	3	.2	1.5	0.005	0.048	0.024	0.957

Deseas hacer otro cálculo? (s/n)

Figura 19. Ventana para cubicar árboles completos.

7.2.4. Cubicación de madera en rollo.

Al seleccionar esta función se muestra un cuadro de menú con las fórmulas que se pueden utilizar para la cubicación de la madera en rollo (Huber, Smalian y Newton) las cuales son comúnmente aplicadas (ver marco de referencia).

Para continuar, se pone al lado del enunciado que aparece en la ventana “**dame la opción**” el número de una de las tres posibles opciones seguido de presionar la tecla enter.

Acto seguido aparece una ventana con un cuadro de dialogo que muestra el enunciado **“desea manejar una longitud comercial? s/n”** si se acepta se **presiona la tecla s** mandando al cuadro de dialogo que a continuación se mencionará, si se rechaza se **presiona la tecla n** donde el programa mandará a un cuadro de diálogo donde pedirá la longitud que se desea utilizar, esta opción se elaboró con la finalidad de ubicar con la longitud que se desee que no sea comercial o si hay alguna discrepancia debido a algún refuerzo, etc.

Posteriormente aparece un cuadro de menú que muestra dos posibles opciones que sirven como referencia en la comercialización de acuerdo con la longitud de la madera en rollo. Largas y cortas dimensiones. Debajo de estas dos opciones se encuentra el enunciado **“dame la opción que desees”** poniendo el numero de una de las dos opciones seguido de presionar la tecla enter.

Acto seguido aparece un cuadro de menú que muestra las medidas comerciales de la madera en rollo bajo la denominación de largas dimensiones, lo mismo se hará cuando se seleccione la opción dos en el formato anterior.

Para continuar, lo único que se hace es **poner la medida seleccionada** al lado del enunciado **“dame la medida”** seguido de presionar la tecla enter.

El cuadro de diálogo que posteriormente aparece es para agregarle a la medida comercial antes seleccionada el refuerzo empleado comercialmente ya que en las medidas comerciales dadas anteriormente se refiere a medidas nominales, se manejó por separado debido a que este es variable en algunas ocasiones. Luego aparece la pregunta **“deseas agregar refuerzo? s/n”**, si se desea agregarlo, se **presiona la tecla s** seguido de presionar enter, si no se desea agregarlo se **presiona la tecla n** seguido de presionar enter.

Si se presionó la tecla s, a continuación aparecerá el enunciado **“dame el refuerzo en metros”** poniéndose el valor del refuerzo en metros seguido de presionar enter, mandando el programa a la ventana de cubicación de la madera en rollo.

La siguiente ventana es la que sirve para la cubicación de la madera en rollo como tal, desde luego teniendo definida tanto la fórmula como la longitud por emplear, Dependiendo de la fórmula que se escoja será el número de datos de entrada, como se explica a continuación:

Si se selecciona la opción 1 **“Huber”**, el dato de entrada será solamente uno que corresponde al diámetro medio de la madera en rollo representado en la ventana como **dm**, el cual se debe **introducir en metros** seguido de presionar enter, inmediatamente nos dará el volumen en metros cúbicos y en pies rollo.

Si se selecciona la opción 2 “**Smalian**”, los datos de entrada serán dos, correspondientes al diámetro del extremo 1 y el diámetro del extremo 2 de la madera en rollo, representados en la ventana como **D1** y **d2**, respectivamente, se deben de **introducir en metros** seguido de presionar la tecla enter, en cada uno de los casos inmediatamente se dará el volumen en metros cúbicos y pies rollo (Figura 20).

Si se selecciona la opción 3 “**Newton**” los datos de entrada serán tres, que corresponden al diámetro del extremo 1, al diámetro medio y al diámetro del extremo 2 de la madera en rollo, representados en la ventana como **D1**, **dm** y **d2** respectivamente, los cuales se deben de **introducir en metros** seguido de presionar enter, en cada uno de los casos, dando así el volumen metros cúbicos y pies rollo.

Después de introducir los datos y conocer el volumen aparecerá el enunciado “**deseas hacer otro cálculo? s/n**” en la parte inferior de la ventana; si se acepta se **presiona la tecla s** y se continúa cubicando otra pieza de madera en rollo, si se rechaza se **presiona la tecla n**, dando el total de piezas cubicadas con su respectivo volumen total de las mismas.

Para continuar, se vuelve a presionar la tecla n mandando el Software al menú principal de productos.

MADERA EN ROLLO							
# Trozo	D1 Mts	dm Mts	d2 Mts	Long Mts	Prom Mts	Volumen M3	Volumen Pie rollo
1	.6		.5	2.55	0.55	0.606	128.35

Deseas hacer otro cálculo? (s/n)

Figura 20. Ventana para cubicar madera en rollo.

7.2.5. Cubicación por apilamiento.

Este cuadro de menú nos muestra los principales productos forestales maderables que son sometidos a algún tipo de control mediante la cubicación por apilamiento.

Para continuar se introduce el número del producto que se desea cubicar de las ocho opciones posibles y se pone al lado de la pregunta que aparece en la parte inferior de la ventana **“Dame el número del producto que deseas cubicar”** seguido de presionar enter (Figura 21).

Si se desea salir de este cuadro de menú, se pone la opción 9 y se presiona enter, el Software mandará al cuadro de menú principal (Figura 17).

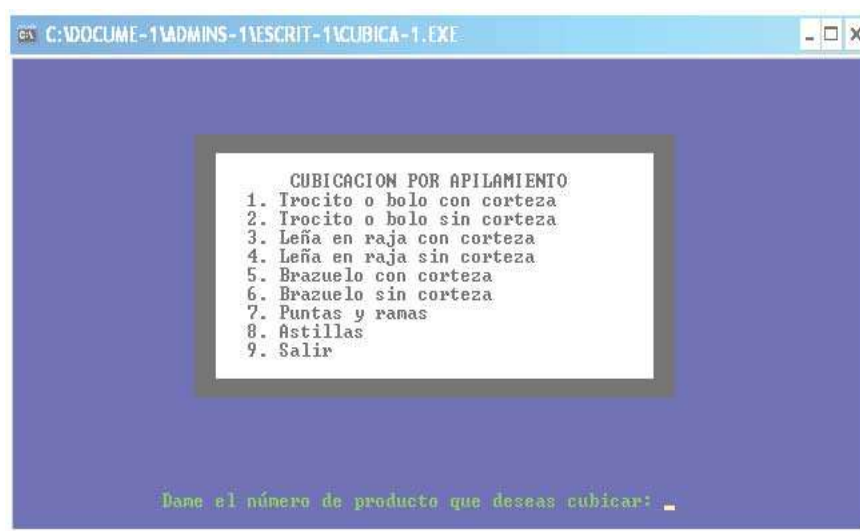


Figura 21. Ventana del menú para selección de producto.

Después de tener perfectamente bien definido el producto que se desea cubicar, se pasa al formato o ventana de cubicación por apilamiento, en donde se introducen tres datos de entrada (Figura 22): **altura** del lote en metros, **ancho** del lote en metros y **largo** del lote en metros, seguido de presionar la tecla enter en cada uno de los casos. De esta manera aparecerá tanto el factor de apilamiento propuesto como el resultado del volumen del lote en metros cúbicos y pies rollo.

Al tener el volumen aparecerá en la parte inferior de la ventana **“deseas hacer otro cálculo? s/n”**, si se acepta se **presiona la tecla s** y se continua cubicando otro apilamiento del mismo tipo, si se rechaza se **presiona la tecla n**, en donde aparecerá el total de lotes cubicados además de aparecer otro enunciado “

deseas hacer otro cálculo? s/n”, si se acepta se **presiona la tecla s**, en donde el Software mandará a la ventana anterior para escoger otro producto para cubicar por apilamiento, si se rechaza, se **presiona la tecla n** mandándonos al cuadro de menú principal de materias primas y productos forestales maderables (Figura 17).

Dimensiones del lote			Leña en raja con corteza		
Alto Mts	Ancho Mts	Largo Mts	Factor Propuesto	Vol.Total M3	Vol. Total Pie rollo
2	1.5	3.5	0.60	6.300	1334.72

Deseas hacer otro cálculo? <s/n>

Figura 22. Ventana para la cubicación por apilamiento.

7.2.6. Cubicación de embalajes de madera.

En la siguiente ventana se muestra un cuadro de menú donde se mencionan los dos principales tipos de embalajes que circulan en el mercado nacional que son las tarimas y las cajas para empaque (Figura 23).

Posteriormente se pondrá únicamente el numero del producto por cubicar al lado de la pregunta que aparece en la parte inferior de la ventana **“Dame el número del producto que deseas cubicar”**, para así pasar a la ventana del producto que se seleccionó.

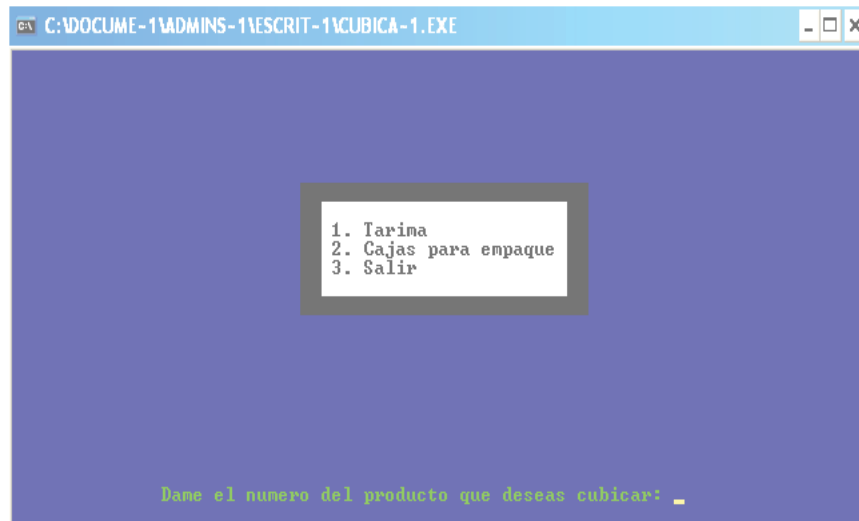


Figura 23. Ventana para la selección del embalaje.

A. Cubicación de tarimas.

Esta ventana muestra la forma en que se pueden cubicar las tarimas de madera apareciendo primeramente un recuadro llamado “**recuadro de despiece**”, el cual servirá para cubicar la tarima de manera individual (Figura 24).

Primeramente se pone el **nombre de la tarima** seguido de presionar enter, después se deberán poner el número de piezas de igual dimensiones que contiene la tarima; **espesor en pulgadas, ancho en pulgadas y largo en pies**, seguido de presionar enter para cada caso, dando el volumen en pies tabla y metros cúbicos del total de las piezas.

Después aparecerá en la parte inferior de la ventana el enunciado “**tienes más piezas? s/n**”, si se acepta se **presiona la tecla s** y se repite el paso anterior dando el número de piezas y sus dimensiones, si se rechaza se **presiona la tecla n** en donde el Software indicará que ha finalizado el despiece de la tarima, continuando a la siguiente ventana.

C:\DOCUME~1\ADMIN~1\ESCRIT~1\CUBICA-1.EXE					
TARIMAS					
Tipo	U.Unitario P.T.	#Piezas	Volumen M3	Volumen P.T.	
Despiece					
Tipo:Exportacion	Dimensiones				
No	Espesor	Ancho	Largo	Volumen	Volumen:
Piezas	Pulgada	Pulgada	Pies	M3	P.T.
1	1.75	3.75	4	0.015	6.562
2	.75	6	3.3	0.006	2.425
3	.75	5	3.3	0.005	2.062
4	.75	4	3.3	0.016	6.600

Tienes mas piezas? (s/n)

Figura 24. Ventana para el despiece de tarimas.

Después de dar por finalizado nuestro despiece se continúa con la siguiente ventana, la cual muestra el volumen unitario en pies tabla de la tarima que se cubicó anteriormente así como el nombre que se le designó. Para continuar se introducen los datos de entrada que son el número de tarimas por cubicar seguido de presionar enter, dando el volumen del total de tarimas por cubicar, tanto en metros cúbicos como en pies tabla (Figura 25). Después aparecerá en la parte inferior de la ventana “**tienes otro tipo? s/n**”; si se acepta se **presiona la tecla s** y se continua con otro despiece, si se rechaza se **presiona la tecla n** mandando el Software a la ventana principal de productos forestales maderables.

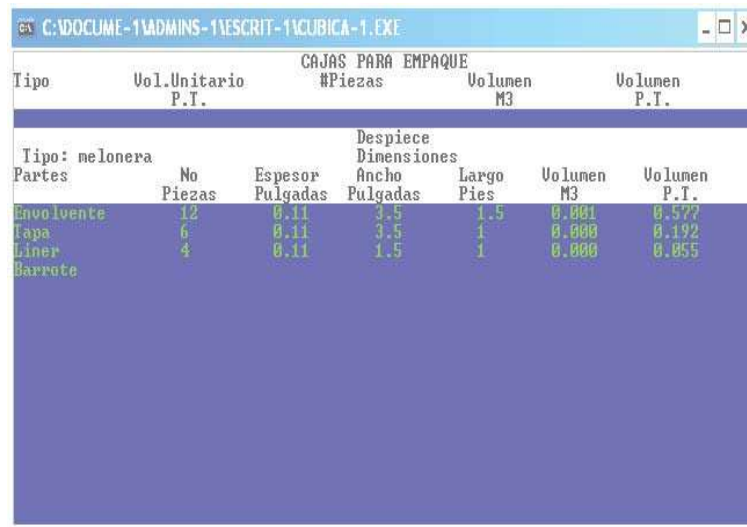
C:\DOCUME~1\ADMIN~1\ESCRIT~1\CUBICA-1.EXE					
TARIMAS					
Tipo	U.Unitario P.T.	#Piezas	Volumen M3	Volumen P.T.	
Exportacion	17.70	1000	41.772	17700.00	

Tienes otro tipo? (s/n)

Figura 25. Ventana para la cubicación de tarimas.

B. Cubicación de cajas para empaque.

Después de haber seleccionado a este producto, la mecánica de cubicación es prácticamente la misma que en la cubicación de tarimas, en donde se realiza un despiece; asignación de nombre del producto y cálculo del volumen unitario y la posterior anotación del número total de piezas, dando el Software la respectiva cubicación tanto en metros cúbicos como en pies tabla (Figura 26).



Tipo	Vol.Unitario P.I.	#Piezas	Volumen M3	Volumen P.I.		
Despiece						
Dimensiones						
Partes	No Piezas	Espesor Pulgadas	Ancho Pulgadas	Largo Pies	Volumen M3	Volumen P.I.
Envolvente	12	0.11	3.5	1.5	0.001	0.577
Tapa	6	0.11	3.5	1	0.000	0.192
Liner	4	0.11	1.5	1	0.000	0.055
Barrate						

Figura 26. Ventana para la cubicación de cajas para empaque.

7.2.7. Cubicación de madera aserrada o con escuadría.

En esta ventana se lleva a cabo la cubicación de la madera aserrada, introduciendo los datos de entrada que son las dimensiones de la pieza; **espesor (E) en pulgadas, ancho (A) en pulgadas y largo (L) en pies**, en caso de haber seleccionado como unidad de cubicación los pies tabla y los mismos datos de entrada pero **en metros** en caso de haber seleccionado como unidad de cubicación los metros cúbicos en el cuadro de menú previamente visto, seguido de presionar enter en cada caso, además de poner las piezas por cubicar ya sea por calidad o mill room en el caso de los polines por citar un ejemplo seguido de presionar enter. Inmediatamente después de dar los datos de entrada el Software reconocerá las dimensiones dadas, mostrando en la ventana el nombre del producto, si no las reconoce se pondrá en pantalla N. C. (no comercial), pero de igual forma el Software realizará la cubicación; también dará el total de piezas así como el volumen total en metros cúbicos y pies tabla, al mismo tiempo aparecerá el enunciado “**deseas hacer otro cálculo? s/n**”; si se acepta se **presiona la tecla s**, se continúa cubriendo, si se rechaza se **presiona la tecla n** (Figura 27).

Producto	Dimensiones			M.R.	Calidad			Total Piezas	Vol. M3	Vol. P.T.
	E * A * L				1a	2a	3a			
	Pulg.	Pulg.	Pie							
Polin	3.5	3.5	8	10			10	0.193	81.67	
Tabla	.75	12	8	10	10	1	1	22	0.312	132.00
Tabla	.75	10	8	100	0	0	0	100	1.100	500.00
Vigueta	3.5	4	10	100				100	2.753	1166.67
Total:				120	10	1	1	132	1.684	713.67

Deseas hacer otro calculo? <s/n>

Figura 27. Ventana para la cubicación de madera aserrada

Al momento de **presionar la tecla n** el Software dará el total de piezas, así como el volumen total de todas las piezas cubicadas en metros cúbicos y pies tabla. Al mismo tiempo saldrá el enunciado “**deseas el volumen total por tipo de producto? s/n**”, en la parte inferior de la ventana, si se acepta se **presiona la tecla s** en donde se desglosará el volumen total por tipo de producto (Figura 28), si se rechaza se **presiona la tecla n** mandando el programa al cuadro de menú de productos forestales maderables (Figura 17).

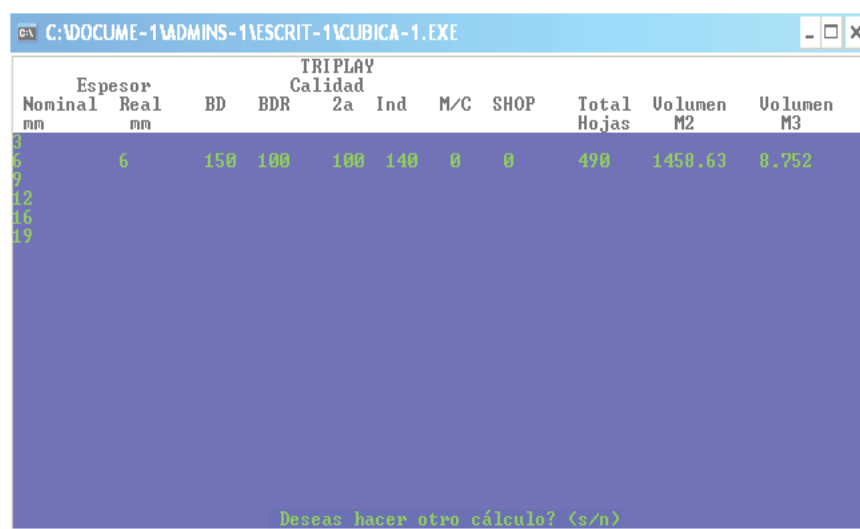
Producto	Dimensiones			M.R.	Calidad			Total Piezas	Vol. M3	Vol. P.T.
	E * A * L				1a	2a	3a			
	Pulg.	Pulg.	Pie							
Tabla							100	1.416	600.00	
Polin							100	1.927	816.67	
Total:				200	0	0	0	200	3.343	1416.67

Figura 28. Ventana para la cubicación de madera aserrada por tipo de producto.

7.2.8. Cubicación de triplay.

Primeramente aparecerá un cuadro de menú mostrando las dimensiones comerciales del triplay, poniéndose el número de una de las tres opciones disponibles al lado del enunciado **“dame la opción que deseas”**, seguido de presionar enter. Teniendo definida la dimensión comercial, la ventana que a continuación se describe muestra la forma de cubicar el triplay, mostrando primeramente los espesores nominales comerciales, lo que se debe de hacer es poner el **espesor real en milímetros** seguido de presionar enter, así como el **número de hojas** por calidad seguido de presionar enter en cada caso, de esta forma el Software calculará tanto de total de hojas por espesor así como el volumen tanto en metros cúbicos como en metros cuadrados base 5.2 mm para este caso (Figura 29).

Al poner estos dos datos de entrada (espesor y número de hojas) por espesor saldrá el enunciado **“deseas hacer otro cálculo? s/n”**, si se acepta se **presiona la tecla s** y aparecerá el enunciado **“que espesor deseas cubicar?”**, poniéndose el espesor comercial que se desea seguir cubriendo seguido de presionar enter, si se rechaza se **presiona la tecla n** y el Software dará el total de hojas tanto por calidad como por espesor ; al mismo tiempo saldrá el enunciado **“deseas cubicar con otra dimensión? s/n”** en la parte inferior de la ventana, si se acepta se **presiona la tecla s** mandando el Software al cuadro de menú de selección de dimensiones siguiendo con la misma mecánica, si se rechaza se **presiona la tecla n** mandando el Software al cuadro de menú de productos forestales maderables (Figura 17).



Espesor		TRIPLAY						Total	Uolumen	Uolumen
Nominal	Real	Calidad				M/C	SHOP			
mm	mm	BD	BDR	2a	Ind					
9	6	150	100	100	140	0	0	490	1450.63	8.752
6										
9										
12										
16										
19										

Deseas hacer otro cálculo? (s/n)

Figura 29. Ventana para la cubicación de triplay.

8. CONCLUSIONES.

- Con el desarrollo de este Software, se dará un aporte en la facilidad para poder llevar a cabo de forma práctica y sencilla, la determinación del volumen de materias primas y productos forestales maderables.
- Este Software de cubicación de materias primas y productos forestales maderables puede representar una gran herramienta de apoyo por su portabilidad, por la variedad de productos que se pueden cubicar, por la orientación que nos brinda en cuanto a la comercialización de los mismos y por que se consideran las unidades de cubicación que se emplean en la práctica para cada caso en particular.
- El Software de cubicación de materias primas y productos forestales maderables nos ayudará a reducir el margen de error por la facilidad de uso, así como por los fundamentos técnicos de cubicación aplicados en su desarrollo y que son los empleados en la práctica en todo el proceso productivo del aprovechamiento del bosque (extracción, transporte, almacenamiento, transformación y comercialización).
- Como herramienta de apoyo en la determinación de volumen de materias primas y productos forestales maderables, este Software puede ir adecuándose a los cambios o mejoras que se puedan dar en las técnicas de medición o cubicación existentes.
- La elaboración de este Software desarrollado en el lenguaje de programación C, sirve como pauta para la creación de otros programas enfocados al sector forestal así como para incursionar, debido a su similitud, a otros lenguajes de programación mas nuevos, complejos y visualmente más atractivos, ya que se aprendió sobre el diseño, la estructuración y la aplicación de comandos básicos que comprende un Software o programa computacional en general.

9. RECOMENDACIONES.

- Para contar con una buena exactitud en la determinación del volumen de las diferentes materias primas y productos forestales maderables por medio del Software, es importante introducir información y/o datos de entrada precisos y confiables.
- Para darle un uso adecuado al Software se tiene que leer cuidadosamente de este trabajo tanto el “marco de referencia” para tener conocimiento de las características, fórmulas de cubicación y unidades de medida empleadas para cada producto en particular, así como la “forma de uso”, para saber perfectamente el flujo de operación, los datos de entrada y unidades de medida que se deben dar, para evitar cualquier confusión o conflicto que pueda afectar durante el uso del Software.
- Para el uso del Software existen algunas ventajas por su portabilidad, ya que se puede tener incluido en una computadora, grabado en un disco o inclusive en una memoria USB, no teniendo problemas referente a la disponibilidad de un equipo de cómputo sobre todo para el caso de usarlo en centros de almacenamiento y/o transformación, transporte y comercios; donde se complica un poco es al momento de quererlo usar en la extracción de las materias primas forestales ya que se complica contar con un equipo de cómputo en campo, aunque en la actualidad existen dispositivos como el “iPad” e inclusive algunos “teléfonos celulares” en los cuales se pueden ejecutar software y/o programas computacionales.
- Debido a la rapidez con la que evoluciona la tecnología en la actualidad, y más aún con todo lo relacionado a la informática y equipos de cómputo, sobre todo con relación a los procesadores y sistemas operativos de los mismos; para este caso en particular se tiene una limitante para su uso, ya que se puede ejecutar solamente en equipos de cómputo que contienen un sistema operativo de hasta 32 bits.
- Con este trabajo se sientan las bases para poder incursionar en un futuro a otros lenguajes de programación más modernos y visualmente más atractivos, ya que se aprendió la metodología básica para desarrollar un software, desde su análisis, diseño, programación (estructura, declaración de funciones, etc) y prueba; ya sea mejorando trabajos de esta índole o desarrollando nuevas aplicaciones inherentes al sector forestal principalmente.

10. BIBLIOGRAFIA.

Barraza I., S. F. 1990. Control estadístico de calidad computarizado para un aserradero. Tesina profesional. Facultad de Ingeniería en Tecnología de la Madera. Universidad Michoacana de San Nicolás de Hidalgo. Morelia, México.

Becerril M., V. 2002. Creación de un paquete de software que genere las secuelas de secado. Tesis profesional. Facultad de Ingeniería en Tecnología de la Madera . Universidad Michoacana de San Nicolás de Hidalgo. Morelia, México.

Cacari A., T. 1993. Aspectos técnicos para la implementación de una fabrica de cajas de madera con especies tropicales. Tesis profesional. Facultad de Ingeniería en Tecnología de la Madera. Universidad Michoacana de San Nicolás de Hidalgo. Morelia, México.

Centro de investigación del Pacífico centro Michoacán. 1993. Medición forestal. Guía técnica No. 4. S.A.R.H. México.

Comité de Medición Forestal. 1999. Tablas de cubicación de productos forestales. Boletín Técnico. Consejo Técnico consultivo Forestal y de suelos del Estado de Chihuahua. México.

CONAFOR (Comisión Nacional Forestal). 2010. Sistema Nacional de Información Forestal. [Internet]. Mexico. Disponible en: <http://148.223.105.188:2222/gif/snif_portal/> [Consulta: Febrero 2010].

Donal, B. y Schumacher, F. 1986. Medición Forestal. Editorial Herrero S. A. México.

Elguero P., M. G. 1994. Análisis de la cubicación de la madera en rollo. Tesina profesional. Facultad de Ingeniería en Tecnología de la Madera. Universidad Michoacana de San Nicolás de Hidalgo. Morelia, México.

Enríquez Q., M.. 1980. La industria de las cajas de madera alambrada en el Estado de Durango. Tesis profesional. Universidad Autónoma de Chapingo. Estado de México, México.

Facultad de Ciencias del Medio Ambiente UCML. 2006. Dasometría. [Internet]. España. Disponible en: < <http://www.agronomos.agr-ab.uclm.es/Guias/Guiasfores/Segundo/Dasometria.htm> > [Consulta: Diciembre 2006].

Gestiopolis, 2010. Sector Forestal en México. [Internet]. Mexico < <http://www.gestiopolis.com/recursos/.../secformx.htm> > [Consulta: Noviembre 2010].

Maderería La Huerta. 2009. Lista de dimensiones comerciales y precios de la madera aserrada. Nota divulgativa. Morelia, México.

Monografías, 2007. Lenguaje C. [Internet]. México. Disponible en: < <http://www.monografias.com/trabajo4/lenguajec/lenguajecshtml>> [Consulta: Abril2007].

Murguía G., J. L. 1996. Modelado del control de la producción de un aserradero, utilizando la hoja electrónica Microsoft Excel para Windows versión 5.0. Tesis profesional. Facultad de Ingeniería en Tecnología de la Madera. Universidad Michoacana de San Nicolás de Hidalgo. Morelia, México.

Nava R., A. 1992. Proceso de fabricación de tarimas industriales. Tesis profesional. Facultad de Ingeniería en Tecnología de la Madera. Universidad Michoacana de San Nicolás de Hidalgo. Morelia, México.

PROBOSQUE (Protectora de Bosques).1998. Manual de cubicación de productos forestales. Metepec, Estado de México. México.

Ramírez, H. y Treviño, J. L. 1994. Dendrometria, Primera Edición. Universidad Autónoma de Chapingo. Estado de México. México.

Rodríguez., P. F. y Broto C., M. 2002. Guía para el usuario del programa CUBICA v 1.2 España. <http://web.udl.es> [Consulta: Noviembre 2006].

Santos E., J. M. 2007. Introducción al Lenguaje C. [Internet]. España. Disponible en: < <http://sopa.dis.ulpgc.es/so/cpp/intro-c/> > [Consulta: Mayo 2007].

SEMARNAT (Secretaria del Medio Ambiente y Recursos Naturales). 2003. Ley General de Desarrollo Forestal Sustentable. Ley. México.

SEMARNAT (Secretaria del Medio Ambiente y Recursos Naturales). 2005. Reglamento de la Ley General de Desarrollo Forestal Sustentable. Reglamento. México.

Triplay de Michoacán. 2003. Instructivo técnico para el supervisor del área de prensado, dimensionado y pulido. Boletín técnico. Indaparapeo, Mich., México.

UNAM (Universidad Nacional Autónoma de México). 2007. El Lenguaje C. [Internet]. México. Disponible en: < <http://dgenp.unam.mx/planteles/p4/p4lenguajec/1int-c.htm#intro>> [Consulta: Julio 2007].

Universidad de Concepción, Departamento de Ingeniería Informática y Ciencias de la computación, Facultad de Ingeniería. 2006. Gestión de proyectos de desarrollo de Software. [Internet]. Chile. Disponible en: <<http://www.inf.udeccl/mvaras/gpis/apuntesGPDS.pdf>> [Consulta: Enero 2007].

Wiktionary, 2010. Definición de Software. [Internet]. Mexico <<http://es.wiktionary.org/wiki/software>> [Consulta: Noviembre 2010]

11. ANEXOS.

11.1. Código fuente del Software.

```
#include<dos.h>
#include<ctype.h>
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>
#include<math.h>

struct cadena{
    char tar[20];
    float pie;
}cad[18];
struct caden{
    char tar[20];
    float ban;
}pro[24];
struct producto{
    char prod[10];
    int tp,ban;
    float tpm;
    float ttp;
}p[10];
////////////////////////////////////
struct caden producto(float es,float an,float lo,int c,int x){
    struct caden aux;
    int ban=0;
    if(x==1){/////pie tabla
        if((es==0.5 || es==0.75 || es==1) && (an==4 || an==6 || an==8 || an==10 || an==12) &&
(lo==8 || lo==10 || lo==12 || lo==14 || lo==16 || lo==18)){
            gotoxy(1,c);printf("Tabla");
            strcpy(aux.tar,"Tabla");
            aux.ban=0;
        }
        else
            if((es==0.5 || es==1 || es==0.75 || es==1.5 || es==2 || es==2.5) && (an==4 ||
an==6 || an==8 || an==10 || an==12) && (lo==2 || lo==4 || lo==6)){
                gotoxy(1,c);printf("C.D.");
                strcpy(aux.tar, "C.D.");
                aux.ban=1;
            }
        else
            if((es==1.5 || es==2 || es==2.5) && (an==4 || an==6 || an==8 || an==10 ||
an==12) && (lo==8 || lo==10 || lo==12 || lo==14 || lo==16 || lo==18)){
                gotoxy(1,c);printf("Tablon");
                strcpy(aux.tar,"Tablon");
                aux.ban=2;
            }
        else
            if((es==7) && (an==8 || an==9) && (lo==8 || lo==8.5)){
```

```

        gotoxy(1,c);printf("Durmiente");
        strcpy(aux.tar,"Durmiente");
        aux.ban=3;
    }
    else
        if((es==7) && (an==9 || an==10) && (lo>=9 && lo<=16)){
            gotoxy(1,c);printf("Cambio-via");
            strcpy(aux.tar,"Cambio- via");
            aux.ban=4;
        }
    else
        if((es==3.25 || es==3 || es==3.5 || es==4) && (an==3.25 || an==3.5 || an==3 ||
an==4) && (lo==8)){
            gotoxy(1,c);printf("Polin");
            strcpy(aux.tar,"Polin");
            aux.ban=5;
        }
    else
        if((es==3.5) && (an==4 || an==6 || an==8 || an==12 || an==10) && (lo==18 ||
lo==20 || lo==10 || lo==12 || lo==14 || lo==16)){
            gotoxy(1,c);printf("Vigueta");
            strcpy(aux.tar,"Vigueta");
            aux.ban=6;
        }
    else
        if((es==4 || es==6) && (an==4 || an==10 || an==6 || an==8 || an==12) &&
(lo==18 || lo==20 || lo==10 || lo==12 || lo==14 || lo==16)){
            gotoxy(1,c);printf("Viga");
            strcpy(aux.tar,"Viga");
            aux.ban=7;
        }
    else
        if((es==8 || es==10 || es==12) && (an==4 || an==10 || an==6 || an==8 || an==12)
&& (lo==18 || lo==20 || lo==10 || lo==12 || lo==14 || lo==16)){
            gotoxy(1,c);printf("Gualdra");
            strcpy(aux.tar,"Gualdra");
            aux.ban=8;
        }
    else{
        gotoxy(1,c);printf("N.C.");
        strcpy(aux.tar,"N.C.");
        aux.ban=9;
    }
}
} //fin if x==1
else{ ///if x==2 metros cubicos
    if(((es>=0.0125 && es<=0.0175) || (es>=0.018 && es<=0.024) || (es>=0.025 &&
es<=0.03) && ((an>=0.1 && an<=0.113) || (an>=0.15 && an<=0.163) || (an>=0.2 && an<=0.213) ||
(an>=0.25 && an<=0.263) || (an>=0.3 && an<=0.313)) && ((lo>=2.438 && lo<=2.513) || (lo>=3.048 &&
lo<=3.123) || (lo>=3.658 && lo<=3.732) || (lo>=4.267 && lo<=4.342) || (lo>=4.877 && lo<=4.952) ||
(lo>=5.486 && lo<=5.561)))){
        gotoxy(1,c);printf("Tabla");
        strcpy(aux.tar,"Tabla");
        aux.ban=0;
    }
}
}

```

```

        ban=1;
    }
    if(((es>=0.0125 && es<=0.0175) || (es>=0.0188 && es<=0.024) || (es>=0.025 &&
es<=0.03) || (es>=0.037 && es<=0.043) || (es>=0.050 && es<=0.055) || (es>=0.062 && es<=0.068))
&& ((an>=0.1 && an<=0.113) || (an>=0.15 && an<=0.163) || (an>=0.2 && an<=0.213) || (an>=0.25 &&
an<=0.263) || (an>=0.3 && an<=0.313)) && ((lo>=0.610 && lo<=0.660) || (lo>=1.219 && lo<=1.269) ||
(lo>=1.829 && lo<=1.879))){
        gotoxy(1,c);printf("C.D.");
        strcpy(aux.tar,"C.D.");
        aux.ban=1;
        ban=1;
    }
    if(((es>=0.037 && es<=0.043) || (es>=0.050 && es<=0.055) || (es>=0.062 &&
es<=0.068)) && ((an>=0.1 && an<=0.113) || (an>=0.15 && an<=0.163) || (an>=0.2 && an<=0.213) ||
(an>=0.25 && an<=0.263) || (an>=0.3 && an<=0.313)) && ((lo>=2.438 && lo<=2.513) || (lo>=3.048 &&
lo<=3.123) || (lo>=3.658 && lo<=3.732) || (lo>=4.267 && lo<=4.342) || (lo>=4.877 && lo<=4.952) ||
(lo>=5.486 && lo<=5.561))){
        gotoxy(1,c);printf("Tablon");
        strcpy(aux.tar,"Tablon");
        aux.ban=2;
        ban=1;
    }
    if(es>=0.174 && es<=0.178)
        if((an>=0.19 && an<=0.21) || (an>=0.224 && an<=0.234))
            if((lo>=2.43 && lo<=2.45) || (lo>=2.589 && lo<=2.591)){
                gotoxy(1,c);printf("Durmiente");
                strcpy(aux.tar,"Durmiente");
                aux.ban=3;
                ban=1;
            }
    if(es>=0.174 && es<=0.178)
        if((an>=0.224 && an<=0.234) || (an>=0.249 && an<=0.259))
            if((lo>=2.743 && lo<=4.877) || (lo>=3.657 && lo<=4.268)){
                gotoxy(1,c);printf("Cambio via");
                strcpy(aux.tar,"Cambio via");
                aux.ban=4;
                ban=1;
            }
    if(es>=0.075 && es<=0.11)
        if(an>=0.075 && an<=0.11)
            if(lo>=2.44 && lo<=2.54){
                gotoxy(1,c);printf("Polin");
                strcpy(aux.tar,"Polin");
                aux.ban=5;
                ban=1;
            }
    if((es>=0.087 && es<=0.093) && ((an>=0.1 && an<=0.113) || (an>=0.15 &&
an<=0.163) || (an>=0.2 && an<=0.213) || (an>=0.25 && an<=0.263) || (an>=0.3 && an<=0.313)) &&
((lo>=6.096 && lo<=6.171) || (lo>=3.048 && lo<=3.123) || (lo>=3.658 && lo<=3.732) || (lo>=4.267 &&
lo<=4.342) || (lo>=4.877 && lo<=4.952) || (lo>=5.486 && lo<=5.561))){
        gotoxy(1,c);printf("Vigueta");
        strcpy(aux.tar,"Vigueta");
        aux.ban=6;
        ban=1;
    }
}

```

```

        if(((es>=0.10 && es<=0.105) || (es>=0.15 && es<=0.157)) && ((an>=0.1 &&
an<=0.113) || (an>=0.15 && an<=0.163) || (an>=0.2 && an<=0.213) || (an>=0.25 && an<=0.263) ||
(an>=0.3 && an<=0.313)) && ((lo>=6.096 && lo<=6.171) || (lo>=3.048 && lo<=3.123) || (lo>=3.658 &&
lo<=3.732) || (lo>=4.267 && lo<=4.342) || (lo>=4.877 && lo<=4.952) || (lo>=5.486 && lo<=5.561)))){
            gotoxy(1,c);printf("Viga");
            strcpy(aux.tar,"Viga");
            aux.ban=7;
            ban=1;
        }
        if(((es>=0.20 && es<=0.208) || (es>=0.25 && es<=0.259) || (es>=0.3 && es<=0.309))
&& ((an>=0.1 && an<=0.113) || (an>=0.15 && an<=0.163) || (an>=0.2 && an<=0.213) || (an>=0.25 &&
an<=0.263) || (an>=0.3 && an<=0.313)) && ((lo>=6.096 && lo<=6.171) || (lo>=3.048 && lo<=3.123) ||
(lo>=3.658 && lo<=3.732) || (lo>=4.267 && lo<=4.342) || (lo>=4.877 && lo<=4.952) || (lo>=5.486 &&
lo<=5.561)))){
            gotoxy(1,c);printf("Gualdra");
            strcpy(aux.tar,"Gualdra");
            aux.ban=8;
            ban=1;
        }
        if(ban==0){
            gotoxy(1,c);printf("N.C.");
            strcpy(aux.tar,"N.C.");
            aux.ban=9;
        }
    }
}/////////else
return aux;
}

////////////////////////////////////// MADERA EN ROLLO//////////////////////////////////////
void rollo(){
    char res='s',r,re='s',resp='s';
    int trozo[24],form,cont,c,dim,tro,ban;
    float prom[24],d1[24],d2[24],dm[24],ing=0,volm[24],volp[24],cte=0.7854,refu;
    float log[24],p,dd1,dd2,dm2,pprom=0,pvolp=0,pvolm=0;
    while(re=='s' || re=='S'){
        clrscr();
        window(1,1,80,3);
        textcolor(BLACK);
        textbackground(LIGHTGRAY);
        cont=4,tro=1;
        c=0,pprom=0,pvolp=0,pvolm=0;
        while(c<2){
            cprintf("                                \r\n");
            c++;
        }
        gotoxy(1,1);cprintf("                                MADERA EN ROLLO                                ");
        gotoxy(1,2);cprintf("                                                                ");
        gotoxy(1,3);cprintf("                                                                ");
        gotoxy(1,2);cprintf("\rFórmulas:    1. Huber    2. Smalian    3. Newton ");
        gotoxy(1,3);cprintf("Que fórmula deseas?: ");scanf("%d",&form);
        gotoxy(1,2);cprintf("                                                                ");
        gotoxy(1,3);cprintf("                                                                ");
        gotoxy(1,2);cprintf("\rDeseas manejar una longitud comercial? (s/n) ");r=getch();
        gotoxy(1,2);cprintf("                                                                ");
        if(r=='s' || r=='S'){

```



```

gotoxy(1,2);cprintf("\r      1. Largas dimensiones          2. Cortas dimensiones");
gotoxy(1,3);cprintf("\rQue opcin deseas?: ");scanf("%d",&dim);
gotoxy(1,2);cprintf("\r                                     ");
gotoxy(1,3);cprintf("\r                                     ");
if(dim==1){
gotoxy(1,2);cprintf("\r 2.44 mts  3.05 mts  3.66 mts  4.27 mts  4.88 mts  5.49 mts
6.10 mts");
gotoxy(1,3);cprintf("\r Dame la medida que deseas.");scanf("%f",&ing);
}
if(dim==2){
gotoxy(1,2);cprintf("\r 0,61 mts          1.22 mts          1.83 mts");
gotoxy(1,3);cprintf("\r Dame la medida que deseas: ");scanf("%f",&ing);
}
gotoxy(1,2);cprintf("                                     ");
gotoxy(1,3);cprintf("                                     ");
gotoxy(1,2);cprintf("\r Deseas agregar refuerzo? (s/n) ");resp=getch();
if(resp=='s' || resp=='S'){
gotoxy(1,2);cprintf("\r Dame el refuerzo en metros:      ");
gotoxy(1,3);cprintf("\r Refuerzo: ");scanf("%f",&refu);
}
}
if(r=='n' || r=='N'){
gotoxy(1,3);cprintf("                                     ");
gotoxy(1,3);cprintf("\r Deseas manejar una sola longitud? (s/n) "); r=getch();
gotoxy(1,3);cprintf("                                     ");
if(r=='s' || r=='S'){
gotoxy(1,3);cprintf("\r Dame la longitud en metros: ");scanf("%f",&ing);
}
}
gotoxy(1,1);cprintf("                                MADERA EN ROLLO                                ");
gotoxy(1,2);cprintf("                                     ");
gotoxy(1,3);cprintf("                                     ");
gotoxy(1,2);cprintf("                                     ");
gotoxy(1,3);cprintf("                                     ");
res='s';
gotoxy(1,1);cprintf("                                MADERA EN ROLLO                                ");
gotoxy(1,2);cprintf("\r #      d1      dm      d2      Long      Prom      Volumen      Volumen
\n");
gotoxy(1,3);cprintf("\r Trozo      Mts      Mts      Mts      Mts      Mts      M3      Pie rollo");
window(1,1,80,25);
textcolor(LIGHTGREEN);
textbackground(BLUE);
ing=ing+refu;
while(res=='s' || res=='S'){
if(cont==24){
c=4;
while(c<cont){
if(ban!='\0'){
gotoxy(1,c);cprintf("
");
trozo[c]=trozo[c+1];
volm[c]=volm[c+1];
volp[c]=volp[c+1];
log[c]=log[c+1];
prom[c]=prom[c+1];//////////

```

```

        d1[c]=d1[c+1];
        d2[c]=d2[c+1];
        dm[c]=dm[c+1];
        gotoxy(2,c);cprintf("%d",trozo[c]);
        gotoxy(36,c);cprintf("%.2f",log[c]);
        gotoxy(54,c);cprintf("%.3f",volm[c]);
        gotoxy(66,c);cprintf("%.2f",volp[c]);
        if(form==1){
            gotoxy(20,c);cprintf("%.2f",dm[c]);
        }
        if(form==2){
            gotoxy(12,c);cprintf("%.2f",d1[c]);
            gotoxy(28,c);cprintf("%.2f",d2[c]);
            //gotoxy(45,c);cprintf("%.2f",prom[c]);
        }
        if(form==3){
            gotoxy(12,c);cprintf("%.2f",d1[c]);
            gotoxy(20,c);cprintf("%.2f",dm[c]);
            gotoxy(28,c);cprintf("%.2f",d2[c]);
        }
        gotoxy(45,c);cprintf("%.2f",prom[c]);/////
    }
    c++;
    if(c==23){
        ban='\0';
        cont=23;
    }
}
ban=1;
if(cont<=23){
    gotoxy(1,cont);cprintf("
trozo[cont]=tro;
gotoxy(2,cont);cprintf("%d",trozo[cont]);
if(r=='s' || r=='S'){
    gotoxy(36,cont);cprintf("%.2f",ing);
}
else{
    gotoxy(36,cont);scanf("%f",&ing);
}
if(form==1){
    gotoxy(20,cont);scanf("%f",&dm[cont]);
    gotoxy(45,cont);cprintf("%.2f",dm[cont]);
    prom[cont]=dm[cont];
    pprom=pprom+dm[cont];
    dm2=dm[cont];
    volm[cont]=0.7854*(dm2*dm2)*ing;
}
if(form==2){
    gotoxy(12,cont);scanf("%f",&d1[cont]);
    gotoxy(28,cont);scanf("%f",&d2[cont]);
    dd1=d1[cont];
    dd2=d2[cont];
    prom[cont]=((dd1+dd2)/2);
    pprom=pprom+prom[cont];
");

```

```

        gotoxy(45,cont);cprintf("%.2f",prom[cont]);
        p=prom[cont]* prom[cont];
        volm[cont]=p*ing*cte;
    }
    if(form==3){
        gotoxy(12,cont);scanf("%f",&d1[cont]);
        gotoxy(20,cont);scanf("%f",&dm[cont]);
        gotoxy(28,cont);scanf("%f",&d2[cont]);
        prom[cont]=((d1[cont]+dm[cont]+d2[cont])/3);////////
        gotoxy(45,cont);cprintf("%.2f",prom[cont]);
        ///prom[cont]=dm[cont];//////////
        pprom=pprom+prom[cont];//////////dm[cont];
        dd1=d1[cont];
        dm2=dm[cont];
        dd2=d2[cont];
        dd1=dd1*dd1;
        dm2=4*(dm2*dm2);
        dd2=dd2*dd2;
        volm[cont]=(0.1309*(dd1+dm2+dd2))*ing;
    }
    log[cont]=ing;
    volp[cont]=211.86*volm[cont];
    gotoxy(54,cont);cprintf("%.3f",volm[cont]);
    gotoxy(66,cont);cprintf("%.2f",volp[cont]);
    pvolm=pvolm+volm[cont];
    pvolp=pvolp+volp[cont];
    ///tro++;
    if(cont<=23)
        cont+=1;
}////////if cont
textcolor(458);
textbackground(BLUE);
gotoxy(25,25);cprintf("Deseas hacer otro c lculo? (s/n)");
res=getch();
gotoxy(25,25);cprintf("                ");
textcolor(BLACK);
textbackground(7);
gotoxy(1,24);cprintf("                ");
gotoxy(1,24);cprintf("Total:");
gotoxy(45,24);cprintf("%.2f",pprom/tro);
gotoxy(54,24);cprintf("%.3f",pvolm);
gotoxy(66,24);cprintf("%.2f",pvolp);
textcolor(LIGHTGREEN);
textbackground(BLUE);
tro++;////////
}/////while res
textcolor(458);
textbackground(BLUE);
gotoxy(25,25);cprintf("                ");
gotoxy(20,25);cprintf("Deseas cubicar con otra longitud? (s/n)");
re=getch();
gotoxy(25,25);cprintf("                ");
textcolor(LIGHTGREEN);
textbackground(BLUE);
}///fin del primer while re

```

```

} ///fin rollo
//////////////////////////////////////ARBOLES COMPLETOS//////////////////////////////////////
void arboles(int op){
    int ban,nf,c,arb[24],cord=5,nd;
    float td[24],tl[24],fl[24],fuste=0,pd[24],pl[24],volt[24],volf[24],volp[24],voltot[24];
    float vt=0,vf=0,vp=0,vtt=0;
    float fd[150];
    char res='s';
    clrscr();
    window(1,1,80,4);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    c=0;
    while(c<3){
        cprintf("                                \r\n ");
        c++;
    }
    gotoxy(1,1);cprintf("                                ARBOLES COMPLETOS                                ");
    cprintf("\r No Tocon   Fuste   Punta   Volumen   Volumen   Volumen   Volumen \r\n");
    cprintf("\r Arbol í L í L í L Tocon   fuste   Punta   Total \r\n");
    cprintf("\r      Mts   Mts   Mts   M3   M3   M3   M3");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    if(op==1){
        while(res=='s' || res=='S'){
            ban=1;
            fuste=0;
            if(cord<=22){
                arb[cord]=cord-4;
                gotoxy(2,cord);cprintf("%d",arb[cord]);
                gotoxy(9,cord);scanf("%f",&td[cord]);
                gotoxy(14,cord);scanf("%f",&tl[cord]);
                gotoxy(5,cord+1);cprintf("Dame el numero de diametros del
fuste:");scanf("%d",&nf);

                nd=0;
                textbackground(LIGHTGRAY);
                gotoxy(20,3);cprintf(" ");
                while(nd<nf){
                    textbackground(BLUE);
                    gotoxy(19,cord);cprintf("                                ");
                    textbackground(LIGHTGRAY);
                    gotoxy(20,3);cprintf("%d",nd);
                    textbackground(BLUE);
                    gotoxy(19,cord);scanf("%f",&fd[nd]);
                    nd+=1;
                }
                textbackground(LIGHTGRAY);
                gotoxy(20,3);cprintf(" ");
                textbackground(BLUE);
                nf=0;
                while(nf<nd){
                    if(nf==0){
                        fuste=fuste+fd[nf]*fd[nf];
                    }
                }
            }
        }
    }
}

```

```

else
    if((nf+1)==nd){
        fuste=fuste+fd[nf]*fd[nf];
    }
    else{
        fuste=fuste+(2*(fd[nf]*fd[nf]));
    }
    nf+=1;
}
gotoxy(5, cord+1); cprintf(" ");
gotoxy(24, cord); scanf("%f", &fl[cord]);
volf[cord]=0.3927*fuste* fl[cord];
vf+=volf[cord];
gotoxy(30, cord); scanf("%f", &pd[cord]);
gotoxy(35, cord); scanf("%f", &pl[cord]);
volt[cord]=0.7854*(td[cord]*td[cord])*tl[cord];
gotoxy(40, cord); printf("%.3f", volt[cord]);
vt+=volt[cord];
gotoxy(50, cord); cprintf("%.3f", volt[cord]);
volp[cord]=0.3927*(pd[cord]*pd[cord])*pl[cord];
vp+=volp[cord];
gotoxy(60, cord); cprintf("%.3f", volp[cord]);
voltot[cord]=volt[cord]+volf[cord]+volp[cord];
gotoxy(70, cord); cprintf("%.3f", voltot[cord]);
vtt+=voltot[cord];
textcolor(458);
textbackground(BLUE);
gotoxy(25, 25); cprintf("Deseas hacer otro c lculo? (s/n)");
res=getch();
gotoxy(25, 25); cprintf(" ");
cord+=1;
textcolor(BLACK);
textbackground(7);
gotoxy(1, 24); cprintf("

gotoxy(1, 24); cprintf("Total: ");
gotoxy(40, 24); cprintf("%.3f", vt);
gotoxy(60, 24); cprintf("%.3f", vp);
gotoxy(50, 24); cprintf("%.3f", vf);
gotoxy(70, 24); cprintf("%.3f", vtt);
textcolor(LIGHTGREEN);
textbackground(BLUE);
}///fin if
if(cord==23){//
    c=5;
    while(c<cord){
        if(ban!='\0'){
            gotoxy(1, c); cprintf("

            arb[c]=arb[c+1];
            td[c]=td[c+1];
            tl[c]=tl[c+1];
            volt[c]=volt[c+1];
            fl[c]=fl[c+1];
            volf[c]=volf[c+1];

```

```

        pl[c]=pl[c+1];
        pd[c]=pd[c+1];
        volp[c]=volp[c+1];
        voltot[c]=voltot[c+1];
        gotoxy(2,c);cprintf("%d",arb[c]);
        gotoxy(9,c);cprintf("%.1f",td[c]);
        gotoxy(14,c);cprintf("%.1f",tl[c]);
        gotoxy(24,c);cprintf("%.1f",fl[c]);
        gotoxy(30,c);cprintf("%.1f",pd[c]);
        gotoxy(35,c);cprintf("%.1f",pl[c]);
        gotoxy(40,c);cprintf("%.3f",volt[c]);
        gotoxy(50,c);cprintf("%.3f",volp[c]);
        gotoxy(60,c);cprintf("%.3f",volp[c]);
        gotoxy(70,c);cprintf("%.3f",voltot[c]);
    }//if
    c++;
    if(c==22)
        ban='\0';
}//while c
if(ban=='\0'){
    arb[22]+=1;
    gotoxy(1,22);cprintf("
");
    gotoxy(2,22);cprintf("%d",arb[22]);
    gotoxy(9,22);scanf("%f",&td[22]);
    gotoxy(14,22);scanf("%f",&tl[22]);
    gotoxy(5,23); cprintf("Dame el nmero de di metros del
fuste:");scanf("%d",&nf);

    nd=0;
    textbackground(LIGHTGRAY);
    gotoxy(20,3);cprintf("
");
    while(nd<nf){
        textbackground(BLUE);
        gotoxy(19,22);cprintf("
");
        textbackground(LIGHTGRAY);
        gotoxy(20,3);cprintf("%d",nd);
        textbackground(BLUE);
        gotoxy(19,22);scanf("%f",&fd[nd]);
        nd+=1;
    }
    textbackground(LIGHTGRAY);
    gotoxy(20,3);cprintf("
");
    textbackground(BLUE);
    nf=0;
    while(nf<nd){
        if(nf==0){
            fuste=fuste+fd[nf]*fd[nf];
        }
        else
            if((nf+1)==nd){
                fuste=fuste+fd[nf]*fd[nf];
            }
        else{
            fuste=fuste+(2*(fd[nf]*fd[nf]));
        }
    }
}

```

```

        nf+=1;
    }
    gotoxy(5,23);cprintf("
    gotoxy(24,22);scanf("%f",&f[22]);
    vf+=vol[22];
    gotoxy(30,22);scanf("%f",&pd[22]);
    gotoxy(35,22);scanf("%f",&p[22]);
    volt[22]=0.7854*(td[22]*td[22])*t[22];
    gotoxy(40,22);cprintf("%.3f",volt[22]);
    vt+=volt[22];
    gotoxy(50,22);cprintf("%.3f",vol[22]);
    volp[22]=0.3927*(pd[22]*pd[22])*p[22];
    vp+=volp[22];
    gotoxy(60,22);cprintf("%.3f",volp[22]);
    voltot[22]=volt[22]+vol[22]+volp[22];
    gotoxy(70,22);cprintf("%.3f",voltot[22]);
    vtt+=voltot[22];
    textcolor(458);
    textbackground(BLUE);
    gotoxy(25,25);cprintf("Deseas hacer otro cálculo? (s/n)");
    res=getch();
    gotoxy(25,25);cprintf("
    textcolor(LIGHTGREEN);
    textbackground(7);
    gotoxy(1,24);cprintf("
    gotoxy(1,24);cprintf("Total:");
    gotoxy(40,24);cprintf("%.3f",vt);
    gotoxy(60,24);cprintf("%.3f",vp);
    gotoxy(50,24);cprintf("%.3f",vf);
    gotoxy(70,24);cprintf("%.3f",vtt);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
}////////if ban
}////////if coord 23
}//// fin while res
getch();
}///fin op 1
if(op==2){
    while(res=='s' || res=='S'){
        ban=1;
        if(coord<=23){
            arb[coord]=coord-4;
            gotoxy(2,coord);cprintf("%d",arb[coord]);
            gotoxy(9,coord);scanf("%f",&td[coord]);
            gotoxy(14,coord);scanf("%f",&t[coord]);
            volt[coord]=0.7854*(td[coord]*td[coord])*t[coord];
            gotoxy(40,coord);cprintf("%.3f",volt[coord]);
            vt+=volt[coord];
            textcolor(458);
            textbackground(BLUE);
            gotoxy(25,25);cprintf("Deseas hacer otro cálculo? (s/n)");
            res=getch();
            gotoxy(25,25);cprintf("
            coord+=1;
            textcolor(BLACK);

```

```

        textbackground(7);
        gotoxy(1,24);cprintf("
");

        gotoxy(1,24); cprintf("Total: ");
        gotoxy(40,24);cprintf("%.3f",vt);
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
}///fin if cord 23
if(cord==24){
    c=5;
    while(c<cord){
        if(ban=='\0'){
            arb[23]+=1;
            gotoxy(1,23);cprintf("
");

            gotoxy(2,23);cprintf("%d",arb[23]);
            gotoxy(9,23);scanf("%f",&td[23]);
            gotoxy(14,23);scanf("%f",&tl[23]);
            volt[23]=0.7854*(td[23]*td[23])*tl[23];
            gotoxy(40,23);cprintf("%.3f",volt[23]);
            vt+=volt[23];
            textcolor(458);
            textbackground(BLUE);
            gotoxy(25,25);cprintf("Deseas hacer otro calculo? (s/n
");

            res=getch();
            gotoxy(25,25);cprintf("
");
            textcolor(LIGHTGREEN);
            textbackground(7);
            gotoxy(1,24);cprintf("
");

            gotoxy(1,24);cprintf("Total: ");
            gotoxy(40,24);cprintf("%.3f",vt);
            textcolor(LIGHTGREEN);
            textbackground(BLUE);
}//////////

if(ban!='\0'){
    gotoxy(1,c);cprintf("
");
    arb[c]=arb[c+1];
    td[c]=td[c+1];
    tl[c]=tl[c+1];
    volt[c]=volt[c+1];
    gotoxy(2,c);cprintf("%d",arb[c]);
    gotoxy(9,c);cprintf("%.1f",td[c]);
    gotoxy(14,c);cprintf("%.1f",tl[c]);
    gotoxy(40,c);cprintf("%.3f",volt[c]);
}
c++;
if(c==23){
    ban='\0';
}
/*
if(ban=='\0'){
    arb[23]+=1;

```



```

gotoxy(1,23);cprintf("
");
gotoxy(2,23);cprintf("%d",arb[23]);
gotoxy(9,23);scanf("%f",&td[23]);
gotoxy(14,23);scanf("%f",&tl[23]);
volt[23]=0.7854*(td[23]*td[23])*tl[23];
gotoxy(40,23);cprintf("%.3f",volt[23]);
vt+=volt[23];
textcolor(458);
textbackground(BLUE);
gotoxy(25,25);cprintf("Deseas hacer otro calculo? (s/n
");
res=getch();
gotoxy(25,25);cprintf("
");
textcolor(LIGHTGREEN);
textbackground(7);
gotoxy(1,24);cprintf("
");
gotoxy(1,24);cprintf("Total: ");
gotoxy(40,24);cprintf("%.3f",vt);
textcolor(LIGHTGREEN);
textbackground(BLUE);
}*/ /////if ban
//c++;
}////while c
}////if coord 24
}////while
getch();
}///fin op 2
if(op==3){
while(res=='s' || res=='S') {
ban=1;
fuste=0;
if(cord<=22){
arb[cord]=cord-4;
gotoxy(2,cord);cprintf("%d",arb[cord]);
gotoxy(5,cord+1);cprintf("Dame el numero de diametros del
fuste:");scanf("%d",&nf);
nd=0;
textbackground(LIGHTGRAY);
gotoxy(20,3);cprintf(" ");
while(nd<nf){
textbackground(BLUE);
gotoxy(19,cord);cprintf("
");
textbackground(LIGHTGRAY);
gotoxy(20,3);cprintf("%d",nd);
textbackground(BLUE);
gotoxy(19,cord);scanf("%f",&fd[nd]);
nd+=1;
}
textbackground(LIGHTGRAY);
gotoxy(20,3);cprintf(" ");
textbackground(BLUE);
nf=0;
while(nf<nd){

```

```

        if(nf==0){
            fuste=fuste+fd[nf]*fd[nf];
        }
        else
            if((nf+1)==nd){
                fuste=fuste+fd[nf]*fd[nf];
            }
            else{
                fuste=fuste+(2*(fd[nf]*fd[nf]));
            }
        nf+=1;
    }
    gotoxy(5,cord+1);cprintf("
");
    gotoxy(24,cord);scanf("%f",&fl[cord]);
    volf[cord]=0.3927*fuste*fl[cord];
    vf+=volf[cord];
    gotoxy(50,cord);cprintf("%.3f",volf[cord]);
    textcolor(458);
    textbackground(BLUE);
    gotoxy(25,25);cprintf("Deseas hacer otro c lculo? (s/n)");
    res=getch();
    gotoxy(25,25); cprintf("
");
    cord+=1;
    textcolor(BLACK);
    textbackground(7);
    gotoxy(1,24);cprintf("
");

    gotoxy(1,24);cprintf("Total: ");
    gotoxy(50,24);cprintf("%.3f",vf);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
} //fin ifde 23
if(cord==23){
    c=5;
    while(c<cord){
        if(ban!='\0'){
            gotoxy(1,c);cprintf("
");

            arb[c]=arb[c+1];
            fl[c]=fl[c+1];
            volf[c]=volf[c+1];
            gotoxy(2,c);cprintf("%d",arb[c]);
            gotoxy(24,c);cprintf("%.1f",fl[c]);
            gotoxy(50,c);cprintf("%.3f",volf[c]); //vol fuste
        }
        c++;
        if(c==22)
            ban='\0';
    }
    if(ban=='\0'){
        arb[22]+=1;
        gotoxy(1,22);cprintf("
");
        gotoxy(2,22);cprintf("%d",arb[22]);
        gotoxy(5,23);cprintf("Dame el numero de diametros del fuste:
");scanf("%d",&nf);

```

```

nd=0;
textbackground(LIGHTGRAY);
gotoxy(20,3);cprintf(" ");
while(nd<nf){
    textbackground(BLUE);
    gotoxy(19,22);cprintf("                ");
    textbackground(LIGHTGRAY);
    gotoxy(20,3); cprintf("%d",nd);
    textbackground(BLUE);
    gotoxy(19,22);scanf("%f",&fd[nd]);
    nd+=1;
}
textbackground(LIGHTGRAY);
gotoxy(20,3);cprintf(" ");
nf=0;
textbackground(BLUE);
while(nf<nd){
    if(nf==0){
        fuste=fuste+fd[nf]*fd[nf];
    }
    else
    if((nf+1)==nd){
        fuste=fuste+fd[nf]*fd[nf];
    }
    else{
        fuste=fuste+(2*(fd[nf]*fd[nf]));
    }
    nf+=1;
}
gotoxy(5,23);cprintf("                ");
gotoxy(24,22); scanf("%f",&fl[22]);
volff[22]=0.3927*fuste*fl[22];
vf+=volff[22];
gotoxy(50,22);cprintf("%.3f",volff[22]);
textcolor(458);
textbackground(BLUE);
gotoxy(25,25);cprintf("Deseas hacer otro calculo? (s/n) ");
res=getch();
gotoxy(25,25);cprintf("                ");
textcolor(BLACK);
textbackground(7);
gotoxy(1,24);cprintf("                ");
gotoxy(1,24);cprintf("Total: ");
gotoxy(50,24);cprintf("%.3f",vf);
textcolor(LIGHTGREEN);
textbackground(BLUE);
}
}
}
}
}
getch();
}///fin op 3
if(op==4){
    while(res=='s' || res=='S'){
        ban=1;
        if(cord<=23){

```

```

        arb[cord]=cord-4;
        gotoxy(2,cord);cprintf("%d",arb[cord]);
        textbackground(LIGHTGRAY);
        textbackground(LIGHTGRAY);
        textbackground(BLUE);
        gotoxy(30,cord);scanf("%f",&pd[cord]);
        gotoxy(35,cord);scanf("%f",&pl[cord]);
        volp[cord]=0.3927*(pd[cord]*pd[cord])*pl[cord];
        vp+=volp[cord];
        gotoxy(60,cord);cprintf("%.3f",volp[cord]);
        textcolor(458);
        textbackground(BLUE);
        gotoxy(25,25);cprintf("Deseas hacer otro c lculo? (s/n)");
        res=getch();
        gotoxy(25,25); cprintf("
        cord+=1;
        textcolor(BLACK);
        textbackground(7);
        gotoxy(1,24);cprintf("
");

        gotoxy(1,24);cprintf("Total: ");
        gotoxy(60,24);cprintf("%.3f",vp);
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
} //fin del if
if(cord==24){
    c=5;
    while(c<cord){
        if(ban!='\0'){
            gotoxy(1,c);cprintf("
            arb[c]=arb[c+1];
            pl[c]=pl[c+1];
            pd[c]=pd[c+1];
            volp[c]=volp[c+1];
            gotoxy(2,c);cprintf("%d",arb[c]);
            gotoxy(30,c);cprintf("%.1f",pd[c]);
            gotoxy(35,c);cprintf("%.1f",pl[c]);
            gotoxy(60,c);cprintf("%.3f",volp[c]);
        }
        c++;
        if(c==23)
            ban='\0';
    }
    if(ban=='\0'){
        arb[23]+=1;
        gotoxy(1,23);cprintf("
");

        gotoxy(2,23);cprintf("%d",arb[23]);
        gotoxy(30,23);scanf("%f",&pd[23]);
        gotoxy(35,23);scanf("%f", &pl[23]);
        volp[23]=(0.3927*(pd[23]*pd[23])*pl[23]);
        gotoxy(60,23);cprintf("%.3f",volp[23]);
        vp+=volp[23];
        textcolor(458);

```

```

        textbackground(BLUE);
        gotoxy(25,25);cprintf("Deseas hacer otro cálculo? (s/n)");
        res=getch();
        gotoxy(25,25);cprintf("                ");
        textcolor(BLACK);
        textbackground(7);
        gotoxy(1,24);cprintf("                ");
        gotoxy(1,24);cprintf("Total: ");
        gotoxy(60,24);cprintf("%.3f",vp);
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
    }
}
}
getch();
}////fin op4
}
////77
void arbol(){
    int op=0,c=0,x;
    while(op!=5){
        clrscr();
        window(1,1,80,25);
        textcolor(LIGHTGREEN);
        while(c<24){
            cprintf("                \r\n");
            c++;
        }
        window(27,8,52,16);
        textbackground(8);
        c=0;
        while(c<6){
            cprintf("                \r\n");
            c++;
        }
        window(29,9,50,16);
        textcolor(0);
        textbackground(7);
        c=0;
        while(c<6){
            cprintf("                \r\n");
            c++;
        }
        textbackground(BLACK);
        gotoxy(1,8);cprintf("                ");
        textbackground(LIGHTGRAY);
        textcolor(WHITE);
        gotoxy(1,1);cprintf("É");
        gotoxy(1,7);cprintf("É");
        gotoxy(22,1);cprintf("»");
        gotoxy(22,7);cprintf("¼");
        c=2;
        while(c<7){
            textcolor(WHITE);
            gotoxy(1,c);cprintf("o");

```

```

        gotoxy(22,c);cprintf("o");
        c++;
    }
    x=2;
    while(x<22){
        textcolor(WHITE);
        gotoxy(x,1);cprintf("I");
        gotoxy(x,7);cprintf("I");
        x++;
    }
    textcolor(BLACK);
    gotoxy(2,2);cprintf("1. Arboles completos");
    gotoxy(2,3);cprintf("2. Tocones");
    gotoxy(2,4);cprintf("3. Fustes");
    gotoxy(2,5);cprintf("4. Puntas");
    gotoxy(2,6);cprintf("5. Salir");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    gotoxy(20,24);cprintf("Dame el número de producto que deseas.");scanf("%d",&op);
    arboles(op);
}
}
////////////////////////////////////CUBICACION POR APILAMIENTO////////////////////////////////////
int prelotes(){
    int op,c,x;
    clrscr();
    c=0;
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    while(c<24){
        cprintf("                \r\n");
        c++;
    }
    c=0;
    window(18,5,62,18);
    textbackground(8);
    while(c<20){
        cprintf("                \r\n");
        c++;
    }
    c=0;
    window(20,6,60,18);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    while(c<18){
        cprintf("                \r\n");
        c++;
    }
    textbackground(BLACK);
    gotoxy(1,18);cprintf("                ");
    textbackground(LIGHTGRAY);
    textcolor(WHITE);
    gotoxy(1,1);cprintf("É");
}

```

```

gotoxy(1,12);cprintf("É");
gotoxy(41,1);cprintf("»");
gotoxy(41,12);cprintf("¼");
c=2;
while(c<12){
    textcolor(WHITE);
    gotoxy(1,c);cprintf("o");
    gotoxy(41,c);cprintf("o");
    c++;
}
x=2;
while(x<41){
    textcolor(WHITE);
    gotoxy(x,1);cprintf("I");
    gotoxy(x,12);cprintf("I");
    x++;
}
textcolor(BLACK);
gotoxy(8,2);cprintf("CUBICACION POR APILAMIENTO");
gotoxy(4,3);cprintf("1. Trocito o bolo con corteza");
gotoxy(4,4);cprintf("2. Trocito o bolo sin corteza");
gotoxy(4,5); cprintf("3. Leña en raja con corteza");
gotoxy(4,6); cprintf("4. Leña en raja sin corteza");
gotoxy(4,7); cprintf("5. Brazuelo con corteza");
gotoxy(4,8); cprintf("6. Brazuelo sin corteza");
gotoxy(4,9); cprintf("7. Puntas y ramas");
gotoxy(4,10); cprintf("8. Astillas");
gotoxy(4,11); cprintf("9. Salir");
window(1,1,80,25);
textcolor(LIGHTGREEN);
textbackground(BLUE);
gotoxy(15,24);cprintf("Dame el número de producto que deseas cubicar: ");scanf("%d",&op);
return op;
}

void lotes(){
    int ban,c,op,cor,lote,tlote=0;
    float factor,esp[24],an[24],lar[24],vm3[24],vpr[24];
    float tesp=0,tan=0,tlar=0,tvm3=0,tvpr=0;
    char res,re='s',r;
    clrscr();
    while(re=='s' || r=='S'){
        c=0;
        window(1,1,80,25);
        textbackground(BLUE);
        while(c<24){
            cprintf("                                \r\n");
            c++;
        }
        op=prelotes();
        clrscr();
        tesp=0;tan=0;tlar=0,tvm3=0,tvpr=0;tlote=0;
        window(1,1,80,5);
        textcolor(BLACK);
        textbackground(LIGHTGRAY);

```

```

c=0;
while(c<3){
    cprintf("                \r\n");
    c++;
}
if(op==1){
gotoxy(1,1);cprintf("                Trocito o bolo con corteza                ");
factor=0.65;
}
if(op==2){
gotoxy(1,1);cprintf("                Trocito o bolo sin corteza                ");
factor=0.75;
}
if(op==3){
gotoxy(1,1);cprintf("                Leña en raja con corteza                ");
factor=0.6;
}
if(op==4){
gotoxy(1,1);cprintf("                Leña en raja sin corteza                ");
factor=0.7;
}
if(op==5){
gotoxy(1,1);cprintf("                Brazuelo con corteza                ");
factor=0.5;
}
if(op==6){
gotoxy(1,1);cprintf("                Brazuelo sin corteza                ");
factor=0.6;
}
if(op==7){
gotoxy(1,1);cprintf("                Puntas y ramas                ");
factor=0.5;
}
if(op==8){
gotoxy(1,1);cprintf("                Astillas                ");
factor=0.7;
}
if(op==9)
    re='n';
if(op!=9 && op<10){//////////77777
gotoxy(1,2);cprintf(" Dimensiones del lote                ");
gotoxy(1,3);cprintf("\rEspesor  Ancho  Largo  Factor                Vol.Total  Vol.
Total ");
gotoxy(1,4);cprintf(" Mts  Mts  Mts  Propuesto  M3  Pie rollo ");
if(op!=9){//////==
///cprintf("\r Mts  Mts  Mts  Propuesto M3  cuerdas ");
cor=5;
res='s';
window(1,1,80,25);
textcolor(LIGHTGREEN);
textbackground(BLUE);
while(res=='s' || r=='S'){
    if(cor==24){
        c=5;
        while(c<cor){

```



```

");
        if(ban!="\0"){
            gotoxy(1,c);cprintf("

            esp[c]=esp[c+1];
            an[c]=an[c+1];
            lar[c]=lar[c+1];
            vm3[c]=vm3[c+1];
            vpr[c]=vpr[c+1];
            gotoxy(1,c);cprintf("%.1f",esp[c]);
            gotoxy(14,c);cprintf("%.1f",an[c]);
            gotoxy(25,c);cprintf("%.1f",lar[c]);
            gotoxy(38,c);cprintf("%.2f",factor);
            gotoxy(52,c);cprintf("%.3f",vm3[c]);
            gotoxy(67,c);cprintf("%.2f",vpr[c]);

        }
        c++;
        if(c==23){
            ban="\0";
            cor=23;

        }
    }//while c
}////fin if cord==24
ban=1;
if(cor<=23){
    gotoxy(1,cor);cprintf("

    gotoxy(1,cor);scanf("%f",&esp[cor]);
    gotoxy(14,cor);scanf("%f",&an[cor]);
    gotoxy(25,cor);scanf("%f",&lar[cor]);
    gotoxy(38,cor);cprintf("%.2f",factor);
    tesp+=esp[cor];
    tan+=an[cor];
    tlar+=lar[cor];
    tlote+=lote;
    vm3[cor]=esp[cor]*an[cor]*lar[cor]*factor;
    tvm3+=vm3[cor];
    if(op!=9)
        vpr[cor]=211.86*vm3[cor];
    else
        vpr[cor]=(lar[cor]*an[cor]*esp[cor])/3.6317;
    tvpr+=vpr[cor];
    gotoxy(68,cor);cprintf("%.2f",vpr[cor]);
    gotoxy(52,cor);cprintf("%.3f",vm3[cor]);
    if(cor<=23){
        cor+=1;
    }
    textcolor(458);
    textbackground(BLUE);
    gotoxy(25,25);cprintf("Deseas hacer otro calculo?

(s/n)");

    res=getch();
    gotoxy(25,25);cprintf("

    textcolor(BLACK);
    textbackground(7);

```

```

        gotoxy(1,24);cprintf("Total
");
        gotoxy(38,24);cprintf("%.2f",factor);
        gotoxy(52,24);cprintf("%.2f",tvm3);
        gotoxy(68,24);cprintf("%.2f",tvpr);
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        }////if cord 23
        }////fin del 2 while
        textcolor(458);
        gotoxy(20,25);cprintf("Deseas hacer otro calculo? (s/n)");
        res=getch();
        }////7
        }//////if op 9 op menor 10
        /*else{
            textcolor(LIGHTGREEN);
            textbackground(BLUE);
            gotoxy(15,24);cprintf("Escribe una opción valida ");
            delay(10000);
        }*/
        }////fin while
    }
    ////////////////////////////////////TARIMA////////////////////////////////////
    struct cadena otra(int cor){
        int q,pieza,c,c1;
        float pt,m3,esp,an,lar,to;
        char res='s',tar[20];
        struct cadena cade;
        cade.pie=0;
        c1=c=cor;
        textcolor(BLACK);
        textbackground(LIGHTGRAY);
        q=0;
        while(q<=3){
            gotoxy(1,cor+q);cprintf("
");
            q++;
        }
        gotoxy(1,cor);cprintf("
Despiece
");
        cor++;
        gotoxy(1,cor);cprintf("Tipo:");scanf("%20s",&cade.tar);
        gotoxy(28,cor);cprintf("Dimensiones");
        cor++;
        gotoxy(1,cor);cprintf(" No      Espesor      Ancho      Largo      Volumen      Volumen: ");
        gotoxy(1,cor+1);cprintf("Piezas      Pulgada      Pulgada      Pies      M3      P.T. ");
        cor+=2;
        window(1,1,80,25);
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        while(res=='s' || res=='S'){
            gotoxy(1,cor);cprintf("
");
            gotoxy(1,cor);scanf("%d",&pieza);
            gotoxy(17,cor);scanf("%f",&esp);
            gotoxy(31,cor);scanf("%f",&an);
            gotoxy(43,cor);scanf("%f",&lar);

```

```

        pt=to=((esp*an*lar)/12)*pieza;
        m3=pt*0.00236;
        cade.pie+=to;
        gotoxy(54,cor);cprintf("%.3f",m3);
        gotoxy(69,cor);cprintf("%.3f",pt);
        textcolor(458);
        gotoxy(25,25);cprintf("Tienes mas piezas? (s/n)");
        res=getch();
        gotoxy(25,25);cprintf("                ");
        textcolor(LIGHTGREEN);
        if(cor<23)
            cor+=1;
    }
    gotoxy(20,25);cprintf("                ");
    for(c;c<=cor;c++){
        gotoxy(1,c);cprintf("                ");
    }
    gotoxy(1,c1-1);cprintf("%s",cade.tar);
    return cade;/////total
}

void tarima(){
    float vtm3=0,vtp=0,pp[18],vm[18],vp[18],pt[18];
    int c,ban,np[18],cor=4,totpie=0;
    char res='s';
    clrscr();
    window(1,1,80,4);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    c=0;
    while(c<2){
        cprintf("                \r\n");
        c++;
    }
    gotoxy(1,1);cprintf("                TARIMAS                ");
    gotoxy(1,2);cprintf("Tipo          V.Unitario      #Piezas          Volumen          Volumen          ");
    gotoxy(1,3);cprintf("                P.T.                M3                P.T.                ");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    while(res=='s' || res=='S'){
        if(cor==16){
            c=4;
            while(c<cor){
                if(ban!='0'){
                    gotoxy(1,c);cprintf("                ");
                    strcpy(cad[c].tar,cad[c+1].tar);
                    cad[c].pie=cad[c+1].pie;
                    np[c]=np[c+1];
                    pp[c]=pp[c+1];
                    vm[c]=vm[c+1];
                    vp[c]=vp[c+1];
                    pt[c]=pt[c+1];
                    gotoxy(1,c);cprintf("%s",cad[c].tar);
                    gotoxy(18,c);cprintf("%.2f",cad[c].pie);
                }
            }
        }
    }
}

```

```

        gotoxy(36,c);cprintf("%d",np[c]);
        gotoxy(51,c);cprintf("%.3f",vm[c]);
        gotoxy(67,c);cprintf("%.2f",vp[c]);
    }
    c++;//////////chechar
    if(c==16){
        ban='\0';
        cor=15;
    }
}
ban=1;
if(cor<=15){
    gotoxy(20,25);cprintf(" ");
    gotoxy(1,cor);cprintf(" ");
    cad[cor]=otra(cor+1);
    gotoxy(18,cor);cprintf("%.2f",cad[cor].pie);
    textcolor(LIGHTGREEN);
    gotoxy(36,cor);scanf("%d",&np[cor]);
    totpie+=np[cor];
    vp[cor]=np[cor]*cad[cor].pie;
    vm[cor]=0.00236*vp[cor];
    gotoxy(51,cor);cprintf("%.3f",vm[cor]);
    gotoxy(67,cor);cprintf("%.2f",vp[cor]);
    vtm3+=vm[cor];
    vtp+=vp[cor];
    if(cor<=15){
        cor+=1;
    }
    textcolor(458);
    textbackground(BLUE);
    gotoxy(25,25);cprintf("Tienes otro tipo? (s/n) ");
    res=getch();
    gotoxy(25,25);cprintf(" ");
    textcolor(BLACK);
    textbackground(7);
    gotoxy(1,24);cprintf(" ");
    gotoxy(1,24);cprintf("Total: ");
    gotoxy(36,24);cprintf("%d",totpie);
    gotoxy(51,24);cprintf("%.2f",vtm3);
    gotoxy(67,24);cprintf("%.2f",vtp);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
}
}
getch();
}
//////////TRIPLAY//////////

void triplay(){
    char res,re='s';
    int tbd=0,tbdr=0,ban,tse=0,tind=0,tmc=0,tshop=0,total=0,a;
    int op,bd,bdr,se,ind,con1,mc,shop,to,con=4,c,con2,op2,op3,ban1=0,ban2;
    float e[6],m2[6];
    float tm2=0,tm3=0,m3,toes=0;

```

```

while(re=='S' || re=='s'){
    a:
    clrscr();
    window(1,1,80,4);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    op2=op3=c=0;
    while(c<6){
        m2[c]=0;
        c++;
    }
    con=5,ban2=ban=0,con1=0,c=0,tm2=0,tm3=0,toes=0;
    tbd=0,tbdr=0,tse=0,tind=0,tmc=0,tshop=0,total=0;
    res='s';
    while(c<3){
        cprintf("
        c++;
        \r\n");
    }
    gotoxy(1,1);cprintf("
    Medidas comerciales
    gotoxy(1,2);cprintf("\r 1. A=1.22*2.44 MTS
    2. A=091*L= 2.44 MTS
    3.
    A=0.91*L=2.14 mts");
    gotoxy(1,3);cprintf("\r Dame la opcion que deseas: ");scanf("%d",&op);
    if(op>0 && op<4){
        gotoxy(1,1);cprintf("
        TRIPLAY
        gotoxy(1,2);cprintf("
        Espesor
        Calidad
        ");
        gotoxy(1,3);cprintf(" Nominal Real BD BDR 2a Ind M/C SHOP Total
        Volumen Volumen");
        gotoxy(1,4);cprintf(" mm mm
        Hojas M2 M3
        ");
    }
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    if(op==1){
        gotoxy(1,5);cprintf("3");
        gotoxy(1,6);cprintf("6");
        gotoxy(1,7);cprintf("9");
        gotoxy(1,8);cprintf("12");
        gotoxy(1,9);cprintf("16");
        gotoxy(1,10);cprintf("19");
    }else
    if(op==2){
        gotoxy(1,5);cprintf("3");
        gotoxy(1,6);cprintf("6");
    }else
    if(op==3){
        gotoxy(1,5);cprintf("3");
        gotoxy(1,6);cprintf("6");
    }
    else{
        gotoxy(25,23);cprintf(" Teclee una opcion valida ");
        ///ban1=1;
        res='n';
        delay(1500);
    }
}

```

```

        goto a;
    }
    while(res=='s' || res=='S'){
        if(con==5 && ban==0){
            con=6;
            con1=1;
            ban2=1;
        }
        ban1=0;
        if(ban2==0){
            textcolor(458);
            textbackground(BLUE);
            gotoxy(25,23);cprintf(" Que espesor deseas cubicar? ");
            gotoxy(54,23);scanf("%d",&ban);
            gotoxy(25,23);cprintf(" ");
            textcolor(LIGHTGREEN);
            if(ban==3 && op==1){
                con=5;
                gotoxy(6,con);cprintf(" ");
            }
            else
            if(ban==6 && op==1){
                con=6;
                gotoxy(6,con);cprintf(" ");
            }
            else
            if(ban==9 && op==1){
                con=7;
                gotoxy(6,con);cprintf(" ");
            }
            else
            if(ban==12 && op==1){
                con=8;
                gotoxy(6,con);cprintf(" ");
            }
            else
            if(ban==16 && op==1){
                con=9;
                gotoxy(6,con);cprintf(" ");
            }
            else
            if(ban==19 && op==1){
                con=10;
                gotoxy(6,con);cprintf(" ");
            }
            else
            if(ban==3 && op==2){
                con=5;
                gotoxy(6,con);cprintf(" ");
            }
            else
            if(ban==6 && op==2){
                con=6;
                gotoxy(6,con);cprintf(" ");
            }
        }
    }
}

```

```

else
if(ban==3 && op==3){
con=5;
gotoxy(6,con);cprintf("
");
}
else
if(ban==6 && op==3){
con=6;
gotoxy(6,con);cprintf("
");
}
else{
gotoxy(25,23);cprintf(" Dame el espesor comercial que deseas: ");
ban1=1;
res='s';
delay(1500);
gotoxy(25,23);cprintf("
");
}
}
if(ban1==0){
textcolor(LIGHTGREEN);
textbackground(BLUE);
gotoxy(11,con);scanf("%f",&e[con1]);
gotoxy(19,con);scanf("%d",&bd);
gotoxy(24,con);scanf("%d",&bdr);
gotoxy(31,con);scanf("%d",&se);
gotoxy(36,con);scanf("%d",&ind);
gotoxy(42,con);scanf("%d",&mc);
gotoxy(47,con);scanf("%d",&shop);
tbd=tbd+bd;
tbdr=tbdr+bdr;
tse=tse+se;
tind=tind+ind;
tmc=tmc+mc;
tshop=tshop+shop;
to=bd+bdr+se+ind+mc+shop;
total+=to;
gotoxy(54,con);cprintf("%d",to);
if(op==1){
m2[con1]=(1.22*2.44*e[con1])/e[1];
m3=(e[con1]/1000)*1.22*2.44*to;
}
if(op==2){
m2[con1]=(0.91*2.44*e[con1])/e[1];
m3=(e[con1]/1000)*0.91*2.44*to;
op2++;
}
if(op==3){
m2[con1]=(0.91*2.14*e[con1])/e[1];
m3=(e[con1]/1000)*0.91*2.14*to;
op3++;
}
m2[con1]= m2[con1]*to;
toes=toes+m2[con1];
gotoxy(61,con);cprintf("%.2f",m2[con1]);
gotoxy(71,con);cprintf("%.3f",m3);

```

```

        tm2+=m2[con1];
        tm3+=m3;
        con+=1;
        con1+=1;
        textcolor(458);
        textbackground(BLUE);
        gotoxy(25,25);cprintf(" Deseas hacer otro calculo? (s/n)");
        res=getch();
        gotoxy(25,25);cprintf("                ");
        textcolor(BLACK);
        textbackground(7);
        gotoxy(1,24);cprintf(" Total:

");

        gotoxy(19,24);cprintf("%d",tbd);
        gotoxy(24,24);cprintf("%d",tldr);
        gotoxy(31,24);cprintf("%d",tse);
        gotoxy(36,24);cprintf("%d",tind);
        gotoxy(42,24);cprintf("%d",tmc);
        gotoxy(47,24);cprintf("%d",tshop);
        gotoxy(54,24);cprintf("%d",total);
        gotoxy(61,24);cprintf("%.2f",tm2);
        gotoxy(71,24);cprintf("%.3f",tm3);
        textbackground(BLUE);
        ban2=0;
    }
}
textcolor(458);
textbackground(BLUE);
gotoxy(20,25);cprintf(" Deseas cubicar con otra medida comercial? (s/n) ");
re=getch();
gotoxy(20,25);cprintf("                ");
textcolor(LIGHTGREEN);
}
}

////////////////////////////////////Madera aserrada////////////////////////////////////
void madera(){
    float toes=0,pmroom,pprim,pseg,pter,e[24],a[24],l[24],volm[24],volp[24],voltm=0,voltp=0,volcal;
    int cont=5,prim[24],seg[24],ter[24],prod[24],mroom[24],tpmr=0,tpp=0,tps=0,tpt=0,tprod=0;
    int c,con2,ba,y,op;
    char res='s';
    clrscr();
    window(1,1,80,4);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    c=0;
    while(c<10){
        p[c].tpm=0;
        p[c].tpp=0;
        c++;
    }
    c=0;
    while(c<3){
        cprintf("                \r\n");
        c++;
    }
}

```



```

}
printf("          1. Pie Tabla          2. Metros cubicos          \n\n");
printf(" Que unidades deseas utilizar? ");scanf("%d",&op);
if(op<3){
    printf("          MADERA ASERRADA          \n");
    printf(" \r          Dimensiones          Calidad          \n");
    printf(" \rProducto    E * A * L    M.R. 1a 2a 3a Total Vol. Vol.\n");
    if(op==1)
        printf(" \r          Pulg. Pulg. Pie          Piezas M3 P.T.");
    if(op==2)
        printf(" \r          Mts. Mts. Mts.          Piezas M3 P.T.");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    while(res=='s' || res=='S'){
        if(cont==24){
            c=5;
            while(c<cont){
                if(ba!='\0'){
                    gotoxy(1,c);printf("
");
                    strcpy(pro[c].tar,pro[c+1].tar);
                    e[c]=e[c+1];
                    a[c]=a[c+1];
                    l[c]=l[c+1];
                    mroom[c]=mroom[c+1];
                    prim[c]=prim[c+1];
                    seg[c]=seg[c+1];
                    ter[c]=ter[c+1];
                    prod[c]=prod[c+1];
                    volm[c]=volm[c+1];
                    volp[c]=volp[c+1];
                    gotoxy(1,c);printf("%s",pro[c].tar);
                    gotoxy(12,c);printf("%.2f",e[c]);
                    gotoxy(19,c);printf("%.2f",a[c]);
                    gotoxy(26,c);printf("%.2f",l[c]);
                    gotoxy(33,c);printf("%d",mroom[c]);
                    gotoxy(38,c);printf("%d",prim[c]);
                    gotoxy(43,c);printf("%d",seg[c]);
                    gotoxy(48,c);printf("%d",ter[c]);
                    gotoxy(55,c);printf("%d",prod[c]);
                    gotoxy(62,c);printf("%.3f",volm[c]);
                    gotoxy(71,c);printf("%.2f",volp[c]);
                }
                c++;
                if(c==23){
                    ba='\0';
                }
            }
            cont=23;
        }
        ba=1;
        if(cont<=23){
            gotoxy(1,cont);printf("
");

```

```

gotoxy(12,cont);scanf("%f",&e[cont]);
gotoxy(19,cont);scanf("%f",&a[cont]);
gotoxy(26,cont);scanf("%f",&l[cont]);
pro[cont]=producto(e[cont],a[cont],l[cont],cont,op);
gotoxy(33,cont);scanf("%d",&mroom[cont]);
if(pro[cont].ban==1 || pro[cont].ban==3 || pro[cont].ban==4 ||
pro[cont].ban==5 || pro[cont].ban==6 || pro[cont].ban==7 || pro[cont].ban==8){
    prim[cont]=0,seg[cont]=0,ter[cont]=0;
}
else{
    gotoxy(38,cont);scanf("%d",&prim[cont]);
    gotoxy(43,cont);scanf("%d",&seg[cont]);
    gotoxy(48,cont);scanf("%d",&ter[cont]);
}
if(mroom>0)
    tpmr=tpmr+mroom[cont];
if(prim>0)
    tpp=tpp+prim[cont];
if(seg>0)
    tps=tps+seg[cont];
if(ter>0)
    tpt=tpt+ter[cont];
if(op==1){
    prod[cont]=prim[cont]+seg[cont]+ter[cont]+mroom[cont];
    gotoxy(55,cont);cprintf("%d",prod[cont]);
    tprod+=prod[cont];
    volp[cont]=((e[cont]*a[cont]*l[cont])/12)*prod[cont];
    volm[cont]=volp[cont]*0.00236;
    toes=toes+volm[cont];
    gotoxy(62,cont);cprintf("%.3f",volm[cont]);
    gotoxy(71,cont);cprintf("%.2f",volp[cont]);
}
else{
    prod[cont]=prim[cont]+seg[cont]+ter[cont]+mroom[cont];
    gotoxy(55,cont);cprintf("%d",prod[cont]);
    tprod+=prod[cont];
    volm[cont]=e[cont]*a[cont]*l[cont]*prod[cont];
    volp[cont]=volm[cont]/0.00236;
    toes=toes+volm[cont];
    gotoxy(62,cont);cprintf("%.3f",volm[cont]);
    gotoxy(71,cont);cprintf("%.2f",volp[cont]);
}
if(strcmp(pro[cont].tar,"Tabla")==0){
    strcpy(p[0].prod,"Tabla");
    p[0].tp+=prod[cont];
    p[0].tpm+=volm[cont];
    p[0].ttp+=volp[cont];
    p[0].ban=1;
}
if(strcmp(pro[cont].tar,"C.D.")==0){
    strcpy(p[1].prod,"C.D.");
    p[1].tp+=prod[cont];
    p[1].tpm+=volm[cont];
    p[1].ttp+=volp[cont];
    p[1].ban=1;
}

```

```

}
if(strcmp(pro[cont].tar,"Tablon")==0){
    strcpy(p[2].prod,"Tablon");
    p[2].tp+=prod[cont];
    p[2].tpm+=volm[cont];
    p[2].ttp+=volp[cont];
    p[2].ban=1;
}
if(strcmp(pro[cont].tar,"Durmiente")==0){
    strcpy(p[3].prod,"Durmiente");
    p[3].tp+=prod[cont];
    p[3].tpm+=volm[cont];
    p[3].ttp+=volp[cont];
    p[3].ban=1;
}
if(strcmp(pro[cont].tar,"Cambio-Via")==0){
    strcpy(p[4].prod,"Cambio-Via");
    p[4].tp+=prod[cont];
    p[4].tpm+=volm[cont];
    p[4].ttp+=volp[cont];
    p[4].ban=1;
}
if(strcmp(pro[cont].tar,"Polin")==0){
    strcpy(p[5].prod,"Polin");
    p[5].tp+=prod[cont];
    p[5].tpm+=volm[cont];
    p[5].ttp+=volp[cont];
    p[5].ban=1;
}
if(strcmp(pro[cont].tar,"Vigueta")==0){
    strcpy(p[6].prod,"Vigueta");
    p[6].tp+=prod[cont];
    p[6].tpm+=volm[cont];
    p[6].ttp+=volp[cont];
    p[6].ban=1;
}
if(strcmp(pro[cont].tar,"Viga")==0){
    strcpy(p[7].prod,"Viga");
    p[7].tp+=prod[cont];
    p[7].tpm+=volm[cont];
    p[7].ttp+=volp[cont];
    p[7].ban=1;
}
if(strcmp(pro[cont].tar,"Gualdra")==0){
    strcpy(p[8].prod,"Gualdra");
    p[8].tp+=prod[cont];
    p[8].tpm+=volm[cont];
    p[8].ttp+=volp[cont];
    p[8].ban=1;
}
if(strcmp(pro[cont].tar,"N.C.")==0){
    strcpy(p[9].prod,"N.C.");
    p[9].tp+=prod[cont];
    p[9].tpm+=volm[cont];
    p[9].ttp+=volp[cont];
}

```

```

        p[9].ban=1;
    }
    voltm+=volm[cont];
    voltp+=volp[cont];
    textcolor(458);
    textbackground(BLUE);
    gotoxy(25,25);cprintf(" Deseas hacer otro calculo? (s/n) ");
    res=getch();
    gotoxy(25,25);cprintf(" ");
    textcolor(BLACK);
    textbackground(7);
    gotoxy(1,24);cprintf("

");

    gotoxy(1,24);cprintf(" Total: ");
    gotoxy(33,24);cprintf("%d",tprm);
    gotoxy(38,24);cprintf("%d",tpp);
    gotoxy(43,24);cprintf("%d",tps);
    gotoxy(48,24);cprintf("%d",tpt);
    gotoxy(55,24);cprintf("%d",tprod);
    gotoxy(62,24);cprintf("%.3f",voltm);
    gotoxy(71,24);cprintf("%.2f",voltp);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    if(cont<=23)
        cont+=1;
    }
}
else{
    cprintf("Seleccione una opciøn valida \n");
    delay(1000);
    madera();
}
textcolor(458);
textbackground(BLUE);
gotoxy(22,25);cprintf(" Deseas el volumen total por producto? (s/n) ");
res=getch();
gotoxy(22,25);cprintf(" ");
if(res=='s' || res=='S'){
    clrscr();
    window(1,1,80,4);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    c=0;
    while(c<3){
        cprintf(" \n ");
        c++;
    }
    cprintf("          MADERA ASERRADA          \n");
    cprintf(" \r          Dimensiones          Calidad          \n");
    cprintf(" \rProducto  E * A * L    M.R. 1a 2a 3a Total  Vol.  Vol. ");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    c=0;

```

```

        y=5;
        while(c<10){
            if(p[c].ban==1){
                gotoxy(1,y);cprintf("%s",p[c].prod);
                gotoxy(54,y);cprintf("%d",p[c].tp);
                gotoxy(61,y);cprintf("%.3f",p[c].tpm);
                gotoxy(70,y);cprintf("%.2f",p[c].tpp);
                y++;
            }
            c++;
        }
        textcolor(BLACK);
        textbackground(7);
        gotoxy(1,24);cprintf("
");
        gotoxy(1,24);cprintf(" Total: ");
        gotoxy(31,24);cprintf("%d",tpmr);
        gotoxy(37,24);cprintf("%d",tpp);
        gotoxy(42,24);cprintf("%d",tps);
        gotoxy(47,24);cprintf("%d",tpt);
        gotoxy(54,24);cprintf("%d",tprod);
        gotoxy(61,24);cprintf("%.3f",voltm);
        gotoxy(70,24);cprintf("%.2f",voltp);
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        getch();
    }
}
//////////////////////////////////// CAJAS //////////////////////////////////////
struct cadena caja(int cor){
    int q,pieza,c,c1,con=0;
    float pt,m3,esp,an,lar,to;
    //char tar [20];
    struct cadena ar;
    c=c1=cor;
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    q=0;
    while(q<=3){
        gotoxy(1,cor+q);cprintf("
");
        q++;
    }
    gotoxy(1,cor);cprintf("
Despiece
");
    cor++;
    gotoxy(1,cor);cprintf(" Tipo: ");scanf("%20s",&ar.tar);
    gotoxy(37,cor);cprintf(" Dimensiones ");
    cor++;
    gotoxy(1,cor);cprintf("Partes      No  Espesor  Ancho  Largo  Volumen  Volumen");
    gotoxy(1,cor+1);cprintf("      Piezas  Pulgadas  Pulgadas  Pies  M3  P.T. ");
    cor+=2;
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    gotoxy(1,cor);cprintf("Envolvente");
    gotoxy(1,cor+1);cprintf("Tapa");
    gotoxy(1,cor+2);cprintf("Liner");
}

```

```

gotoxy(1,cor+3);cprintf("Barrote");
while(con<4){
    gotoxy(19,cor);scanf("%d",&pieza);
    if(pieza==0){
        esp=an=lar=0;
    }
    else{
        gotoxy(29,cor);scanf("%f",&esp);
        gotoxy(40,cor);scanf("%f",&an);
        gotoxy(51,cor);scanf("%f",&lar);
    }
    pt=to=((esp*an*lar)/12)*pieza;
    m3=pt*0.00236;
    ar.pie+=to;
    gotoxy(59,cor);cprintf("%.3f",m3);
    gotoxy(70,cor);cprintf("%.3f",pt);
    cor+=1;
    con++;
}
delay(1000);
for(c;c<cor;c++){
    gotoxy(1,c);cprintf("
");
}
gotoxy(1,c1-1);cprintf("%s",ar.tar);
return ar;
}
////////////////////////////////////
void cajas(){
    float vtm3=0,vtp=0,pp[18],vm[18],vp[18],pt[18];
    int c,ban,np[18],cor=4,totpie=0;
    char res='s';
    clrscr();
    window(1,1,80,4);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    c=0;
    while(c<2){
        cprintf("
");
        c++;
    }
    gotoxy(1,1);cprintf("
CAJAS PARA EMPAQUE
");
    gotoxy(1,2);cprintf("Tipo Vol.Unitario #Piezas Volumen Volumen
");
    gotoxy(1,3);cprintf(" P.T. M3 P.T.
");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    while(res=='s' || res=='S'){
        if(cor==16){
            c=4;
            while(c<cor){
                if(ban!='0'){
                    gotoxy(1,c);cprintf("
");
                    strcpy(cad[c].tar,cad[c+1].tar);
                    cad[c].pie=cad[c+1].pie;
                    np[c]=np[c+1];
                }
            }
        }
    }
}

```

```

        pp[c]=pp[c+1];
        vm[c]=vm[c+1];
        vp[c]=vp[c+1];
        pt[c]=pt[c+1];
        gotoxy(1,c);cprintf("%s",cad[c].tar);
        gotoxy(18,c);cprintf("%.2f",cad[c].pie);
        gotoxy(36,c);cprintf("%d",np[c]);
        gotoxy(51,c);cprintf("%.3f",vm[c]);
        gotoxy(67,c); cprintf("%.2f",vp[c]);
    }
    c++;
    if(c==15){
        ban='\0';
        cor=15;
    }
}
ban=1;
if(cor<=15){
    gotoxy(1,cor);cprintf("
cad[cor]=caja(cor+1);
gotoxy(18,cor);cprintf("%.2f",cad[cor].pie);
textcolor(LIGHTGREEN);
gotoxy(36,cor);scanf("%d",&np[cor]);
totpie+=np[cor];
vp[cor]=np[cor]*cad[cor].pie;
vm[cor]=0.00236*vp[cor];
gotoxy(51,cor);cprintf("%.3f",vm[cor]);
gotoxy(67,cor);cprintf("%.2f",vp[cor]);
vtm3+=vm[cor];
vtp+=vp[cor];
textcolor(458);
textbackground(BLUE);
gotoxy(25,25);cprintf("Tienes otro tipo de caja? (s/n)");
if(cor<=15){
    cor+=1;
}
res=getch();
gotoxy(25,25);cprintf("
textcolor(BLACK);
textbackground(7);
gotoxy(1,24);cprintf("
gotoxy(1,24);cprintf("Total: ");
gotoxy(36,24);cprintf("%d",totpie);
gotoxy(51,24);cprintf("%.2f",vtm3);
gotoxy(67,24);cprintf("%.2f",vtp);
textcolor(LIGHTGREEN);
textbackground(BLUE);
}
}
getch();
}
////////////////////////////////EMBALAJE////////////////////////////////
void embalaje(){
    int res=0,c,x;

```

```

while(res!=3){
    clrscr();
    c=0;
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    while(c<24){
        printf("                \r\n ");
        c++;
    }
    c=0;
    window(28,8,54,14);
    textbackground(8);
    while(c<9){
        printf("                \r\n ");
        c++;
    }
    c=0;
    window(30,9,52,14);
    textcolor(BLACK);
    textbackground(LIGHTGRAY);
    while(c<5){
        printf("                \n ");
        c++;
    }
    textbackground(BLACK);
    gotoxy(1,6);printf("                ");
    textbackground(LIGHTGRAY);
    textcolor(WHITE);
    gotoxy(1,1);printf("É");
    gotoxy(1,5);printf("É");
    gotoxy(23,1);printf("»"); ///////////////ojo
    gotoxy(23,5);printf("¼");
    c=2;
    while(c<5){
        textcolor(WHITE);
        gotoxy(1,c);printf("o");
        gotoxy(23,c);printf("o");
        c++;
    }
    x=2;
    while(x<23){
        textcolor(WHITE);
        gotoxy(x,1);printf("Í");
        gotoxy(x,5);printf("Í");
        x++;
    }
    textcolor(BLACK);
    gotoxy(2,2);printf("1. Tarima \n");
    gotoxy(2,3);printf("2. Cajas para empaque");
    gotoxy(2,4);printf("3. Salir \n");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);

```



```

        gotoxy(15,24);cprintf("Dame el numero del producto que deseas cubicar:
");scanf("%d",&res);
        if(res<=3)
            switch(res){
                case 1: tarima();break;
                case 2: cajas();break;
                case 3: break;
            }
        else{
            gotoxy(10,24);cprintf("
");
            gotoxy(20,24);cprintf("Tecleo una opcion no valida, trate de nuevo");
            delay(1500);
        }
    }
}
/**/
////////////////////////////////////PRESENTACION////////////////////////////////////
void presentacion(void){
    int producto,c,x,q;
    do{
        clrscr();
        c=0;
        window(1,1,80,25);
        textcolor(LIGHTGREEN);
        textbackground(BLUE);
        while(c<24){
            cprintf("
");
            c++;
        }
        c=0;
        window(23,5,59,16);
        textbackground(8);
        while(c<16){
            cprintf("
");
            c++;
        }
        c=0;
        window(25,6,57,16);
        textcolor(BLACK);
        textbackground(LIGHTGRAY);
        while(c<6){
            cprintf("
");
            c++;
        }
        textbackground(BLACK);
        gotoxy(1,11);cprintf("
");
        textbackground(LIGHTGRAY);
        textcolor(WHITE);
        gotoxy(1,1);cprintf("É");
        gotoxy(1,10);cprintf("É");
        gotoxy(33,1);cprintf("»");////////////////////////////////////ojo
        gotoxy(33,10);cprintf("¼");
        c=2;
        while(c<10){
            textcolor(WHITE);

```

```

        gotoxy(1,c);cprintf("o");
        gotoxy(33,c);cprintf("o");
        c++;
    }
    x=2;
    while(x<33){
        textcolor(WHITE);
        gotoxy(x,1);cprintf("I");
        gotoxy(x,10);cprintf("I");
        x++;
    }
    textcolor(BLACK);
    gotoxy(3,2);cprintf("1. ARBOLES COMPLETOS \n");
    gotoxy(3,3);cprintf("2. MADERA EN ROLLO \n");
    gotoxy(3,4);cprintf("3. CUBICACION POR APILAMIENTO");
    //Gotoxy(3,5);cprintf("4. POSTES \n");
    gotoxy(3,5);cprintf("4. EMBALAJES DE MADERA \n");
    gotoxy(3,6);cprintf("5. MADERA ASERRADA");
    gotoxy(3,7);cprintf("6. TRIPLAY \n");
    gotoxy(3,8);cprintf("7. SALIR");
    window(1,1,80,25);
    textcolor(LIGHTGREEN);
    textbackground(BLUE);
    q:
    gotoxy(15,24);cprintf("Dame el numero del producto que deseas
cubicar:");scanf("%d",&producto);
    if(producto>0 && producto<8){
        switch(producto){
            case 1: arbol();break;
            case 2: rollo();break;
            case 3: lotes();break;
            //case 4: postes();break;
            case 4: embalaje();break;
            case 5: madera();break;
            case 6: triplay();break;
            case 7: break;
        }
    }
    else{
        gotoxy(10,24);cprintf("
");
        gotoxy(20,24);cprintf("Tecleo una opcion no valida, trate de nuevo");
        delay(1500);
        flushall;
        goto q;
    }
}
}
}

////fabs
}
}while(producto!=7);
}

////////////////////////////////////main principal////////////////////////////////////
void main(void){
    int c=0;
    clrscr();
    window(1,1,80,25);
    textcolor(LIGHTGREEN);

```

```

textbackground(BLUE);
while(c<24){
    cprintf("                \r\n ");
    c++;
}
gotoxy(15,3);cprintf("UNIVERSIDAD MICHOACANA DE SAN NICOLAS DE HIDALGO");
gotoxy(14,7);cprintf("FACULTAD DE INGENIERIA EN TECNOLOGIA DE LA MADERA");
gotoxy(16,12);cprintf("SOFTWARE PARA LA CUBICACION DE MATERIAS PRIMAS");
gotoxy(19,13);cprintf(" Y PRODUCTOS FORESTALES MADERABLES ");
gotoxy(20,18);cprintf("ELABORADO POR: ");
cprintf("GABRIEL PEREZ JUAREZ");
gotoxy(18,25);cprintf("Presiona cualquier letra para continuar");
getch();
presentacion();
}

```