



TECNOLOGICO NACIONAL DE MEXICO

Instituto Tecnológico de la Laguna

Ingeniería en Sistemas Computacionales

TOPICOS AVANZADOS DE PROGRAMACION

PERIODO: Ene - Jun / 2020

GRUPO: "B" 17 – 18 Hrs

PRACTICA No. U1P03

COMPOSICION DE OBJETOS—PRISMAS

ALUMNO:

18131209 José Misael Adame Sandoval

PROFESOR:

Ing. Luis Fernando Gil Vázquez

Torreón, Coah. a 24 de Febrero de 2020

Situación Didáctica

Ingeniería en Estructuras S.A. es una empresa que diseña y construye tanques de acero para la industria mexicana. Los modelos de tanques típicos tienen 4 formas distintas: prisma regular con base rectangular, prisma regular con base en forma de triángulo rectángulo y cilindro. Cuando se les requiere un nuevo diseño los ingenieros de la empresa se basan en una serie de dimensiones que describen a estos tipos de tanques, dichas dimensiones son cálculos que permiten conocer el área de la base, el área lateral, área total y volumen.

La empresa ha pensado que sus ingenieros cuenten con una aplicación de escritorio con interfaz visual que les permita introducir las medidas de la forma del tanque y que puedan obtener rápidamente las dimensiones mencionadas para que puedan dar una estimación de tiempo y costo al cliente.

La aplicación debe mostrar una pantalla de bienvenida (Splash) durante unos segundos, dicha pantalla debe incluir una barra o indicador de progreso.

Después deberá pasar a la pantalla principal donde el usuario podrá seleccionar uno de los modelos de tanque a calcular. Al seleccionar cada modelo de tanque se debe mostrar la figura del tanque y que el usuario pueda poner las medidas correspondientes en campos localizados cerca a su lugar en la figura.

Adicionalmente la app debe contar con diálogo ACERCA DE a la que se pueda acceder desde la pantalla principal en una opción de menú, el diálogo Acerca de debe mostrar logo de la institución, nombre de la aplicación, versión y autor(es).

Para el diseño de clases se reutilizarán las clases de figuras básicas Circulo, Rectangulo y TrianguloRect de manera que los prismas se diseñaran mediante composición de objetos.

Análisis

Se utilizarán distintas fórmulas para la obtención del área de la base, área lateral, área total y volumen de un cilindro, prisma rectangular y prisma triangular

Las fórmulas para el área de la base, área lateral, área total y volumen de un CILINDRO son:

$$\text{Área de la base} = \pi * (\text{radio})^2$$

$$\text{Díametro} = 2r$$

$$\text{Circunferencia} = \pi * \text{Díametro}$$

$$\text{Área Lateral} = \text{Circunferencia} * \text{altura}$$

$$\text{Área Total} = \text{Área Lateral} + (2 * \text{Área de la base})$$

$$\text{Volumen} = \text{Área de la base} * \text{altura}$$

```
radio ≠ ""
radio > 0
```

Las fórmulas para el área de la base, área lateral, área total y volumen de un PRISMA RECTANGULAR son:

$$\text{Área de la base} = \text{largo} * \text{ancho}$$

$$\text{Perímetro} = (2 * \text{largo}) + (2 * \text{ancho})$$

$$\text{Área Lateral} = \text{Perímetro} * \text{altura}$$

$$\text{Área Total} = \text{Área Lateral} + (2 * \text{Área de la base})$$

$$\text{Volumen} = \text{Área de la base} * \text{altura}$$

```
Largo && Ancho ≠ ""
```

```
Largo && Ancho > 0
```

Las fórmulas para el área de la base, área lateral, área total y volumen de un PRISMA TRIANGULAR son:

$$\text{Área de la base} = \frac{\text{cateto_a} * \text{cateto_b}}{2}$$

$$\text{Hipotenusa} = \sqrt{(\text{cateto_a})^2 + (\text{cateto_b})^2}$$

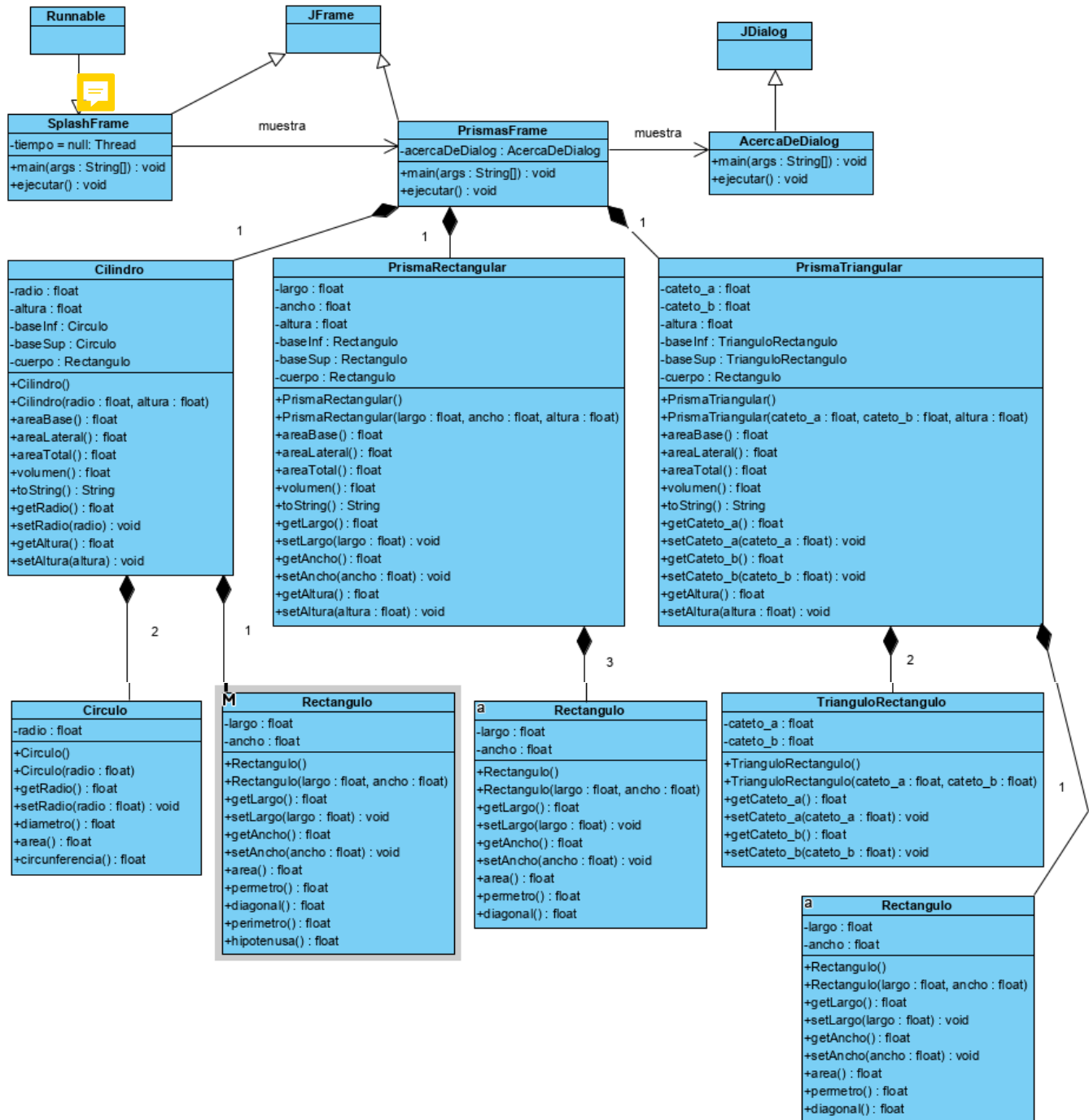
$$\text{Área Lateral} = \text{cateto_a} + \text{cateto_b} + \text{hipotenusa}$$

$$\text{Área Total} = \text{Área Lateral} + (2 * \text{Área de la base})$$

$$\text{Volumen} = \text{Área de la base} * \text{altura}$$

```
cateto_a && cateto_b != ""  
cateto_a && cateto_b > 0
```

Diseño



Código

AcercaDeDialog.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Aplicación GUI de un Acerca de..
*:
*: Archivo      : AcercaDeDialog.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 20/Feb/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación que muestra información sobre el autor de la
*:                la aplicación PrismasApp
*:
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:=====
*: 23/feb/2020 Misael Adame        Agregar prólogo.
*:-----*/

package prismas;

import javax.swing.Icon;

public class AcercaDeDialog extends javax.swing.JDialog {

    //-----

    public AcercaDeDialog ( java.awt.Frame parent, boolean modal ) {
        super ( parent, true );
        initComponents();
        Icon ITL = Imagenes.escalarImagen ( jlblImagenITL.getIcon (),
                                           jlblImagenITL.getWidth (),
                                           jlblImagenITL.getHeight ());

        Icon TECNM = Imagenes.escalarImagen ( jlblImagenTECNM.getIcon (),
                                              jlblImagenTECNM.getWidth (),
                                              jlblImagenTECNM.getHeight ());

        jlblImagenITL.setIcon ( ITL );
        jlblImagenTECNM.setIcon ( TECNM );
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    private void jButtonCerrarActionPerformed(java.awt.event.ActionEvent evt) {
        dispose ();
    }

    //-----

    public static void main(String args[]) {
        Look and feel setting code (optional)

        ejecutar ();
    }

    //-----

```

```

public static void ejecutar () {
    java.awt.EventQueue.invokeLater (new Runnable () {
        public void run () {
            AcercaDeDialog dialog = new AcercaDeDialog ( new javax.swing.JFrame (), true);
            dialog.addWindowListener ( new java.awt.event.WindowAdapter () {
                @Override
                public void windowClosing( java.awt.event.WindowEvent e ) {
                    System.exit ( 0 );
                }
            });
            dialog.setVisible ( true );
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JButton btnCerrar;
private javax.swing.JLabel lblImagenITL;
private javax.swing.JLabel lblImagenTECNM;
// End of variables declaration
}

```

Cilindro.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase que realiza cálculos de un Cilindro
*:
*: Archivo      : Cilindro.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 20/Feb/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Clase que contiene métodos de cálculo para realizar
*:                las siguientes operaciones para un cilindro
*:                1. Área Base
*:                2. Área Lateral
*:                3. Área Total
*:                4. Volumen
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 23/feb/2020 Misael Adame      Agregar prólogo.
*:-----*/

package prismas;

public class Cilindro {

    //-----

    private float radio, altura;
    private Circulo baseInf, baseSup;
    private Rectangulo cuerpo;

    //-----

    public Cilindro(){
        radio = 0;

```

```
        altura = 0;
        baseInf = new Circulo ();
        baseSup = new Circulo ();
        cuerpo = new Rectangulo ();
    }

    //-----

    public Cilindro ( float radio, float altura ) {
        this.radio = radio;
        this.altura = altura;
        baseInf = new Circulo ( radio );
        baseSup = new Circulo ( radio );
        cuerpo = new Rectangulo ( baseInf.circunferencia (), altura );
    }

    //-----

    public float areaBase(){
        return baseInf.area();
    }

    //-----

    public float areaLateral(){
        return cuerpo.area();
    }

    //-----

    public float areaTotal(){
        return baseInf.area() + baseSup.area() + cuerpo.area();
    }

    //-----

    public float volumen(){
        return baseInf.area() * altura;
    }

    //-----

    public String toString(){
        return "Cilindro de radio = " +radio +" altura = " +altura;
    }

    //-----

    public float getRadio(){
        return radio;
    }

    //-----

    public void setRadio ( float radio ) {
        this.radio = radio;
        baseInf.setRadio ( radio );
        baseSup.setRadio ( radio );
        cuerpo.setLargo ( baseInf.circunferencia () );
    }

    //-----

    public float getAltura(){
        return altura;
    }

    //-----

    public void setAltura ( float altura ) {
        this.altura = altura;
        cuerpo.setAncho ( altura );
    }
}
```

Circulo.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase que realiza cálculos de un círculo
*:
*: Archivo      : Circulo.java
*: Autor       : José Misael Adame Sandoval      18131209
*: Fecha       : 20/Feb/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Clase que contiene métodos de cálculo para realizar
*:              las siguientes fórmulas y operaciones para un círculo
*:              1. Diámetro
*:              2. Área
*:              3. Circunferencia
*:
*: Última modif:
*: Fecha       Modificó           Motivo
*:=====
*: 23/feb/2020 Misael Adame      Agregar prólogo.
*:-----*/

package prismas;

public class Circulo {

    //-----

    private float radio;

    //-----

    public Circulo () {
        radio = 0;
    }

    //-----

    public Circulo ( float radio ) {
        this.radio = radio;
    }

    //-----

    public float getRadio () {
        return radio;
    }

    //-----

    public void setRadio ( float radio ) {
        this.radio = radio;
    }

    //-----

    public float diametro () {
        return 2 * radio;
    }

    //-----

    public float area () {
        return (float) (Math.PI * Math.pow(radio, 2));
    }

    //-----

    public float circunferencia () {

```



```

        return (float) (Math.PI * diametro ( ) );
    }
}

```

Imágenes.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TÓPICOS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase con método que permite ajustar el ancho y alto de una imagen
*:
*:
*: Archivo      : Imágenes.java
*: Autor       : Ing. Fernando Gil
*: Fecha       : 29/08/2018
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : El método estático escalarImagen () es el más adecuado de usar para
*:              ajustar el ancho y alto de la imagen de un objeto Icon.
*:              El método recibe 3 argumentos: el objeto Icon y el ancho y alto al que
*:              se desea ajustar. El método devuelve un objeto Icon con la imagen ya
*:              redimensionada.
*:
*: Última modif:
*: Fecha      Modific?          Motivo
*:=====
*: 18/02/2020 F.Gil             -Corregir información del prólogo.
*:-----*/

package prismas;

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.image.BufferedImage;
import javax.swing.Icon;
import javax.swing.ImageIcon;

public class Imágenes {

    //-----

    public static Image getScaledImage ( Image srcImg, int w, int h ) {
        BufferedImage resizedImg = new BufferedImage(w, h, BufferedImage.TYPE_INT_RGB);
        Graphics2D g2 = resizedImg.createGraphics();
        g2.setRenderingHint(RenderingHints.KEY_INTERPOLATION, RenderingHints.VALUE_INTERPOLATION_BILINEAR);
        g2.drawImage(srcImg, 0, 0, w, h, null);
        g2.dispose();
        return resizedImg;
    }

    //-----

    public static Icon escalarImagen ( Icon srcImg, int w, int h ) {
        Image img = ( (ImageIcon) srcImg ).getImage ();
        img = getScaledImage ( img, w, h );
        return new ImageIcon ( img );
    }

    //-----
}

```

PrismaRectangular.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TÓPICOS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS

```

```

:*
:*          Clase que realiza cálculos de un Prisma Rectangular
:*
:* Archivo      : Cilindro.java
:* Autor       : José Misael Adame Sandoval      18131209
:* Fecha       : 20/Feb/2020
:* Compilador  : JAVA J2SE v1.8.0
:* Descripción : Clase que contiene metodos de calculo para realizar
:*              las siguientes operaciones para un Prisma Rectangular
:*              1. Area Base
:*              2. Area Lateral
:*              3. Area Total
:*              4. Volumen
:*
:* Ultima modif:
:* Fecha      Modificó          Motivo
:*=====
:* 23/feb/2020 Misael Adame      Agregar prólogo.
:*-----*/

```

```
package prismas;
```

```

public class PrismaRectangular {

    //-----

    private float largo, ancho, altura;
    private Rectangulo baseInf, baseSup, cuerpo;

    //-----

    public PrismaRectangular () {
        largo = 0;
        ancho = 0;
        altura = 0;
        baseInf = new Rectangulo ();
        baseSup = new Rectangulo ();
        cuerpo = new Rectangulo ();
    }

    //-----

    public PrismaRectangular ( float largo, float ancho, float altura ) {
        this.largo = largo;
        this.ancho = ancho;
        this.altura = altura;
        baseInf = new Rectangulo ( largo, ancho );
        baseSup = new Rectangulo ( largo, ancho );
        cuerpo = new Rectangulo ( baseInf.perimetro(), altura );
    }

    //-----

    public float areaBase () {
        return baseInf.area();
    }

    //-----

    public float areaLateral() {
        return cuerpo.area();
    }

    //-----

    public float areaTotal () {
        return baseInf.area() + baseSup.area() + cuerpo.area();
    }

    //-----

    public float volumen () {
        return baseInf.area() * altura;
    }
}

```

```

//-----

public String toString () {
    return "Prisma rectangular de largo = " +largo +" ancho = " +ancho
        +" altura = " +altura;
}

//-----

public float getLargo () {
    return largo;
}

//-----

public void setLargo ( float largo ) {
    this.largo = largo;
    baseInf.setLargo( largo);
    baseSup.setLargo( largo );
    cuerpo.setLargo( baseInf.perimetro() );
}

//-----

public float getAncho () {
    return ancho;
}

//-----

public void setAncho ( float ancho ) {
    this.ancho = ancho;
    baseInf.setAncho ( ancho );
    baseSup.setAncho( ancho );
}

//-----

public float getAltura () {
    return altura;
}

//-----

public void setAltura ( float altura ) {
    this.altura = altura;
    cuerpo.setAncho ( altura );
}
}

```

PrismasFrame.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Aplicación GUI para probar la clase Conversiones
*:
*: Archivo      : PrismasFrame.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 20/Feb/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación que accede a las clases Cilindro, PrismaRectangular
*:                y PrismaTriangular e invoca a cada uno de sus métodos para
*:                comprobar su funcionamiento
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:=====
*: 24/feb/2020 Misael Adame      Agregar prólogo y realizar validaciones.
*:-----*/

```

```
package prismas;

import javax.swing.Icon;
import javax.swing.JOptionPane;

public class PrismasFrame extends javax.swing.JFrame {

    //-----

    private AcercaDeDialog acercaDeDialog;

    //-----

    public PrismasFrame() {
        initComponents();

        // Ajustar el tamaño de las imágenes
        Icon cilindro = Imagenes.escalarImagen ( jlblImagenCilindro.getIcon(),
                                                jlblImagenCilindro.getWidth(),
                                                jlblImagenCilindro.getHeight());

        Icon p_rect = Imagenes.escalarImagen ( jlblImagenPrismaRectangular.getIcon(),
                                                jlblImagenPrismaRectangular.getWidth(),
                                                jlblImagenPrismaRectangular.getHeight());

        Icon p_triangu = Imagenes.escalarImagen ( jlblImagenPrismaTriangular.getIcon(),
                                                jlblImagenPrismaTriangular.getWidth(),
                                                jlblImagenPrismaTriangular.getHeight());

        jlblImagenCilindro.setIcon ( cilindro );
        jlblImagenPrismaRectangular.setIcon ( p_rect );
        jlblImagenPrismaTriangular.setIcon ( p_triangu );
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    private void jmnEdicionLimpiarActionPerformed(java.awt.event.ActionEvent evt) {

        jtxfCilindroRadio.setText ( null );
        jtxfCilindroAltura.setText ( null );

        jtxfPrismRectLargo.setText ( null );
        jtxfPrismRectAncho.setText ( null );
        jtxfPrismRectAltura.setText ( null );

        jtxfPrismTriangCatetoA.setText ( null );
        jtxfPrismTriangCatetoB.setText ( null );
        jtxfPrismTriangAltura.setText ( null );

        jlblCilindroAreaBase.setText ( null );
        jlblCilindroAreaLateral.setText ( null );
        jlblCilindroAreaTotal.setText ( null );
        jlblCilindroVolumen.setText ( null );

        jlblPrismRectAreaBase.setText ( null );
        jlblPrismRectAreaLateral.setText ( null );
        jlblPrismRectAreaTotal.setText ( null );
        jlblPrismRectVolumen.setText ( null );

        jlblPrismTriangAreaBase.setText ( null );
        jlblPrismTriangAreaLateral.setText ( null );
        jlblPrismTriangAreaTotal.setText ( null );
        jlblPrismTriangVolumen.setText ( null );
    }

    //-----

    private void jmnAyudaAcercaDeActionPerformed(java.awt.event.ActionEvent evt) {
```

```

        acercaDeDialog = new AcercaDeDialog( this, true );
        acercaDeDialog.setVisible( true );
    }

    //-----

    private void jButtonPrismRectCalcularActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            float largo = Float.parseFloat ( jtxfPrismRectLargo.getText() );
            float ancho = Float.parseFloat ( jtxfPrismRectAncho.getText() );
            float altura = Float.parseFloat ( jtxfPrismRectAltura.getText() );
            PrismaRectangular pr = new PrismaRectangular ( largo, ancho, altura );
            if ( largo < 0 || ancho < 0 || altura < 0 )
                JOptionPane.showMessageDialog (
                    this,
                    "Debe proporcionar un valor numérico positivo en todos los campos",
                    "Error",
                    JOptionPane.ERROR_MESSAGE
                );
            else {
                jLabelPrismRectAreaBase.setText ( pr.areaBase() + " " );
                jLabelPrismRectAreaLateral.setText ( pr.areaLateral() + " " );
                jLabelPrismRectAreaTotal.setText ( pr.areaTotal() + " " );
                jLabelPrismRectVolumen.setText ( pr.volumen() + " " );
            }
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog (
                this,
                "Debe proporcionar un valor numérico positivo en todos los campos",
                "Error",
                JOptionPane.ERROR_MESSAGE
            );
        }
    }

    //-----

    private void jButtonCilindroCalcularActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            float radio = Float.parseFloat ( jtxfCilindroRadio.getText() );
            float altura = Float.parseFloat ( jtxfCilindroAltura.getText() );
            Cilindro cil = new Cilindro ( radio, altura );
            if ( radio < 0 || altura < 0 )
                JOptionPane.showMessageDialog (
                    this,
                    "Debe proporcionar un valor numérico positivo en todos los campos",
                    "Error",
                    JOptionPane.ERROR_MESSAGE
                );
            else {
                jLabelCilindroAreaBase.setText ( cil.areaBase() + " " );
                jLabelCilindroAreaLateral.setText ( cil.areaLateral() + " " );
                jLabelCilindroAreaTotal.setText ( cil.areaTotal() + " " );
                jLabelCilindroVolumen.setText ( cil.volumen() + " " );
            }
        } catch ( NumberFormatException exc ) {
            JOptionPane.showMessageDialog (
                this,
                "Debe proporcionar un valor numérico positivo en todos los campos",
                "Error",
                JOptionPane.ERROR_MESSAGE
            );
        }
    }

    //-----

    private void jButtonPrismTriangCalcularActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            float cateto_a = Float.parseFloat ( jtxfPrismTriangCatetoA.getText() );
            float cateto_b = Float.parseFloat ( jtxfPrismTriangCatetoB.getText() );
            float altura = Float.parseFloat ( jtxfPrismTriangAltura.getText() );
            PrismaTriangular pt = new PrismaTriangular ( cateto_a, cateto_b, altura );

```

```

        if ( cateto_a < 0 || cateto_b < 0 || altura < 0 )
            JOptionPane.showMessageDialog (
                this,
                "Debe proporcionar un valor numérico positivo en todos los campos",
                "Error",
                JOptionPane.ERROR_MESSAGE
            );
        else {
            lblPrismTriangAreaBase.setText ( pt.areaBase() + " " );
            lblPrismTriangAreaLateral.setText ( pt.areaLateral() + " " );
            lblPrismTriangAreaTotal.setText ( pt.areaTotal() + " " );
            lblPrismTriangVolumen.setText ( pt.volumen() + " " );
        }
    } catch ( NumberFormatException exce ) {
        JOptionPane.showMessageDialog (
            this,
            "Debe proporcionar un valor numérico positivo en todos los campos",
            "Error",
            JOptionPane.ERROR_MESSAGE
        );
    }
}

//-----

private void jmnArchivoSalirActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

//-----

public static void main ( String args [] ) {
    ejecutar ();
}

//-----

public static void ejecutar() {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new PrismasFrame().setVisible(true);
        }
    });
}
}

```

PrismaTriangular.java

```

/*-----
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase que realiza cálculos de un Prisma Triangular
*:
*: Archivo      : Cilindro.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 20/Feb/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Clase que contiene métodos de cálculo para realizar
*:                las siguientes operaciones para un prisma triangular
*:                1. Área Base
*:                2. Área Lateral
*:                3. Área Total
*:                4. Volumen
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:=====
*: 23/feb/2020 Misael Adame      Agregar prólogo.
*:-----*/

```

```
package prismas;

public class PrismaTriangular {

    //-----

    private float cateto_a, cateto_b, altura;
    private TrianguloRectangulo baseInf, baseSup;
    private Rectangulo cuerpo;

    //-----

    public PrismaTriangular () {
        cateto_a = 0;
        cateto_b = 0;
        altura = 0;
        baseInf = new TrianguloRectangulo ();
        baseSup = new TrianguloRectangulo ();
        cuerpo = new Rectangulo ();
    }

    //-----

    public PrismaTriangular ( float cateto_a, float cateto_b, float altura ) {
        this.cateto_a = cateto_a;
        this.cateto_b = cateto_b;
        this.altura = altura;
        baseInf = new TrianguloRectangulo ( cateto_a, cateto_b );
        baseSup = new TrianguloRectangulo ( cateto_a, cateto_b );
        cuerpo = new Rectangulo ( baseInf.perimetro(), altura );
    }

    //-----

    public float areaBase () {
        return baseInf.area();
    }

    //-----

    public float areaLateral() {
        return cuerpo.area();
    }

    //-----

    public float areaTotal () {
        return baseInf.area() + baseSup.area() + cuerpo.area();
    }

    //-----

    public float volumen () {
        return baseInf.area() * altura;
    }

    //-----

    public String toString () {
        return "Prisma triangular de cateto a = " +cateto_a +" cateto b = " +cateto_b
            +" altura = " +altura;
    }

    //-----

    public float getCateto_a () {
        return cateto_a;
    }

    //-----

    public void setCateto_a ( float cateto_a ) {
        this.cateto_a = cateto_a;
    }
}
```

```

        baseInf.setCateto_a( cateto_a );
        baseSup.setCateto_a( cateto_a );
        cuerpo.setLargo( baseInf.perimetro() );
    }

    public float getCateto_b () {
        return cateto_b;
    }

    //-----

    public void setCateto_b ( float cateto_b ) {
        this.cateto_b = cateto_b;
        baseInf.setCateto_b ( cateto_b );
        baseSup.setCateto_b ( cateto_b );
    }

    //-----

    public float getAltura () {
        return altura;
    }

    //-----

    public void setAltura ( float altura ) {
        this.altura = altura;
        cuerpo.setAncho ( altura );
    }
}

```

Rectangulo.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TÓPICOS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020    HORA: 17-18 HRS
*:
*:          Clase que realiza cálculos de un Rectángulo
*:
*: Archivo      : Rectangulo.java
*: Autor       : José Misael Adame Sandoval    18131209
*: Fecha      : 20/Feb/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Clase que contiene métodos de cálculo para realizar
*:              las siguientes fórmulas y operaciones para un rectángulo
*:              1. Área
*:              2. Perímetro
*:              3. Diagonal
*:
*: Última modif:
*: Fecha      Modificó          Motivo
*:=====
*: 23/feb/2020 Misael Adame      Agregar prólogo.
*:-----*/

package prismas;

public class Rectangulo {

    //-----

    private float largo, ancho;

    //-----

    public Rectangulo () {
        largo = 0;
        ancho = 0;
    }

    //-----

```



```

public Rectangulo ( float largo, float ancho ) {
    this.largo = largo;
    this.ancho = ancho;
}

//-----

public float getLargo() {
    return largo;
}

//-----

public void setLargo(float largo) {
    this.largo = largo;
}

//-----

public float getAncho() {
    return ancho;
}

//-----

public void setAncho(float ancho) {
    this.ancho = ancho;
}

//-----

public float area () {
    return largo * ancho;
}

//-----

public float perimetro () {
    return ( 2 * largo ) + ( 2 * ancho );
}

//-----

public float diagonal () {
    return ( float ) Math.sqrt ( Math.pow ( largo , 2 ) + Math.pow ( ancho , 2 ) );
}
}

```

TrianguloRectangulo.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TÓPICOS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase que realiza cálculos de un triángulo rectángulo
*:
*: Archivo      : TrianguloRectangulo.java
*: Autor       : José Misael Adame Sandoval      18131209
*: Fecha      : 20/Feb/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Clase que contiene métodos de cálculo para realizar
*:              las siguientes fórmulas y operaciones para un triángulo
*:              rectángulo.
*:              1. Área
*:              2. Hipotenusa
*:              3. Circunferencia
*:
*: Última modif:
*: Fecha      Modificó      Motivo

```

```
:*=====
:* 23/feb/2020 Misael Adame      Agregar prólogo.
:*-----*/

package prismas;

public class TrianguloRectangulo {

    //-----

    private float cateto_a, cateto_b;

    //-----

    public TrianguloRectangulo() {
        cateto_a = 0;
        cateto_b = 0;
    }

    //-----

    public TrianguloRectangulo(float cateto_a, float cateto_b) {
        this.cateto_a = cateto_a;
        this.cateto_b = cateto_b;
    }

    //-----

    public float getCateto_a() {
        return cateto_a;
    }

    //-----

    public void setCateto_a(float cateto_a) {
        this.cateto_a = cateto_a;
    }

    //-----

    public float getCateto_b() {
        return cateto_b;
    }

    //-----

    public void setCateto_b(float cateto_b) {
        this.cateto_b = cateto_b;
    }

    //-----

    public float area() {
        return ( cateto_a * cateto_b ) / 2;
    }

    //-----

    public float hipotenusa () {
        return (float) Math.sqrt(Math.pow(cateto_a, 2) + Math.pow(cateto_b, 2) );
    }

    //-----

    public float perimetro() {
        return cateto_a + cateto_b + hipotenusa();
    }
}
```

SplashFrame.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Aplicación GUI Splash
*:
*: Archivo      : SplashFrame.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 23/Feb/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación que hace de Splash (antes de iniciar el
*:                PrismasFrame.java)
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:=====
*: 24/feb/2020 Misael Adame        Agregar prólogo.
*:-----*/

package prismas;

import com.sun.awt.AWTUtilities;
import java.util.logging.Level;
import java.util.logging.Logger;

public class SplashFrame extends javax.swing.JFrame implements Runnable {

    //-----

    private Thread tiempo = null;

    //-----

    public SplashFrame() {
        initComponents();
        this.setLocationRelativeTo(null);
        AWTUtilities.setWindowOpaque(this, false);
        tiempo = new Thread ( this );
        tiempo.start();
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    public static void main ( String args [] ) {
        ejecutar ();
    }

    //-----

    public static void ejecutar() {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new PrismasFrame().setVisible(true);
            }
        });
    }

    //-----

    @Override
    public void run() {

        while ( tiempo != null){
            try {

```

```

        for ( int i = 0; i <= 100; i++) {
            Thread.sleep(70);
            this.jlblPorcBarraProgreso.setText(Integer.toString(i)+" %");
            this.jProgressBarSplash.setValue(i);
            if( i == 100 ) {

                this.setVisible(false);
                Thread.sleep(800);
                new PrismasFrame().setVisible(true);

            }

        }
        break;
    } catch (InterruptedException ex) {
        Logger.getLogger(SplashFrame.class.getName()).log(Level.SEVERE, null, ex);
    }
}
tiempo = null;
}
}

```

PrismasTest.java

```

/*-----
/*-----
:*
:*          INSTITUTO TECNOLÓGICO DE LA LAGUNA
:*          INGENIERÍA EN SISTEMAS COMPUTACIONALES
:*          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
:*
:*          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
:*
:*          Aplicación JUnit que realiza prueba de los métodos
:*
:* Archivo      : PrismasTest.java
:* Autor        : José Misael Adame Sandoval      18131209
:* Fecha        : 23/Feb/2020
:* Compilador   : JAVA J2SE v1.8.0
:* Descripción  : Clase que contiene métodos de prueba (testing) e invoca
:*                a cada uno de los métodos de las clases Cilindro,
:*                Prisma Rectangular, Prisma Triangular para
:*                comprobar su funcionamiento asignando un valor y su
:*                resultado esperado.
:*
:* Última modif:
:* Fecha      Modificó      Motivo
:*=====
:* 24/feb/2020 Misael Adame      Agregar prólogo.
:*-----*/

package pruebas;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;
import prismas.Cilindro;
import prismas.PrismaRectangular;
import prismas.PrismaTriangular;

public class PrismasTest {

    //-----

    public PrismasTest() {
    }

    //-----

    @BeforeClass
    public static void setUpClass() {

```

```
}

//-----

@AfterClass
public static void tearDownClass() {
}

//-----

@Before
public void setUp() {
}

//-----

@After
public void tearDown() {
}

//-----

@Test
public void cilindroTest () {
    // Prueba del constructor de default
    Cilindro c = new Cilindro ();
    assertEquals ( "Cilindro Area de Base", 0.0, c.areaBase(), 0.1 );
    assertEquals ( "Cilindro Area Lateral", 0.0, c.areaLateral(), 0.1 );
    assertEquals ( "Cilindro Area Total", 0.0, c.areaTotal(), 0.1 );
    assertEquals ( "Cilindro volumen", 0.0, c.volumen(), 0.1 );

    // Prueba de cilindro con dimensiones
    c = new Cilindro ( 5.1f, 12.82f );
    assertEquals ( "Cilindro Area de Base", 81.7128, c.areaBase(), 0.0001 );
    assertEquals ( "Cilindro Area Lateral", 410.8072, c.areaLateral(), 0.0001 );
    assertEquals ( "Cilindro Area Total", 574.2328, c.areaTotal(), 0.0001 );
    assertEquals ( "Cilindro volumen", 1047.5583, c.volumen(), 0.0001 );

    // Prueba estableciendo las dimensiones con los metodos setter
    c = new Cilindro();
    c.setRadio( 5.1f );
    c.setAltura ( 12.82f );
    assertEquals ( "Cilindro Area de Base", 81.7128, c.areaBase(), 0.0001 );
    assertEquals ( "Cilindro Area Lateral", 410.8072, c.areaLateral(), 0.0001 );
    assertEquals ( "Cilindro Area Total", 574.2328, c.areaTotal(), 0.0001 );
    assertEquals ( "Cilindro volumen", 1047.5583, c.volumen(), 0.0001 );
}

//-----

@Test
public void prismaRectangularTest () {
    // Prueba del constructor de default
    PrismaRectangular pr = new PrismaRectangular ();
    assertEquals ( "Prisma Rectangular Area de Base", 0.0, pr.areaBase(), 0.1 );
    assertEquals ( "Prisma Rectangular Area Lateral", 0.0, pr.areaLateral(), 0.1 );
    assertEquals ( "Prisma Rectangular Area Total", 0.0, pr.areaTotal(), 0.1 );
    assertEquals ( "Prisma Rectangular volumen", 0.0, pr.volumen(), 0.1 );

    // Prueba con dimensiones
    pr = new PrismaRectangular( 8.0f, 4.0f, 7.0f );
    assertEquals ( "Prisma Rectangular Area de Base", 32.0, pr.areaBase(), 0.0001 );
    assertEquals ( "Prisma Rectangular Area Lateral", 168.0, pr.areaLateral(), 0.0001 );
    assertEquals ( "Prisma Rectangular Area Total", 232.0, pr.areaTotal(), 0.0001 );
    assertEquals ( "Prisma Rectangular volumen", 224.0, pr.volumen(), 0.0001 );
}

//-----

@Test
public void prismaTriangularTest () {
    // Prueba del constructor de default
    PrismaTriangular pt = new PrismaTriangular ();
    assertEquals ( "Prisma Triangular Area de Base", 0.0, pt.areaBase(), 0.1 );
}
```

```
assertEquals ( "Prisma Triangular Area Lateral", 0.0, pt.areaLateral(), 0.1 );
assertEquals ( "Prisma Triangular Area Total", 0.0, pt.areaTotal(), 0.1 );
assertEquals ( "Prisma Triangular volumen", 0.0, pt.volumen(), 0.1 );

// Prueba con dimensiones
pt = new PrismaTriangular( 5.5f, 4.5f, 6.2f );
assertEquals ( "Prisma Triangular Area de Base", 12.375, pt.areaBase(), 0.001 );
assertEquals ( "Prisma Triangular Area Lateral", 106.059, pt.areaLateral(), 0.001 );
assertEquals ( "Prisma Triangular Area Total", 130.809, pt.areaTotal(), 0.001 );
assertEquals ( "Prisma Triangular volumen", 76.725, pt.volumen(), 0.001 );
}
```

Prueba de Ejecución

Al ejecutar la aplicación es necesario capturar al menos un valor numérico

Este es el splash

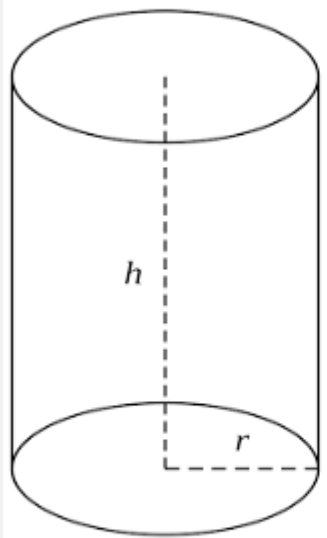


Esta es la interfaz inicial

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



Datos

Radio (r) :

Altura (h) :

Calcular

Resultados

Área de la base:

Área lateral:

Área Total:

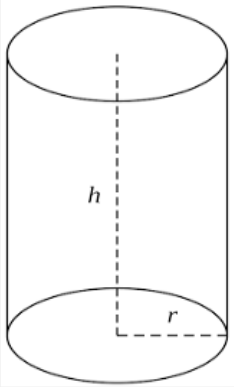
Volumen:

Ejecutándose con Cilindro

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



Datos

Radio (r) : 5.1

Altura (h) : 12.82

Calcular

Resultados

Área de la base: 81.71282

Área lateral: 410.8072

Área Total: 574.23285

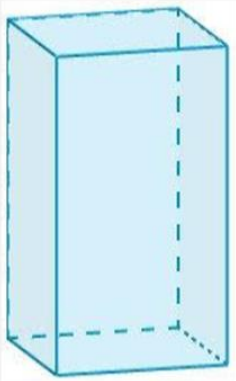
Volumen: 1047.5583

Ejecutándose con Prisma Rectangular

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



Datos

Largo (l) : 8

Ancho (a) : 4

Altura (h) : 7

Calcular

Resultados

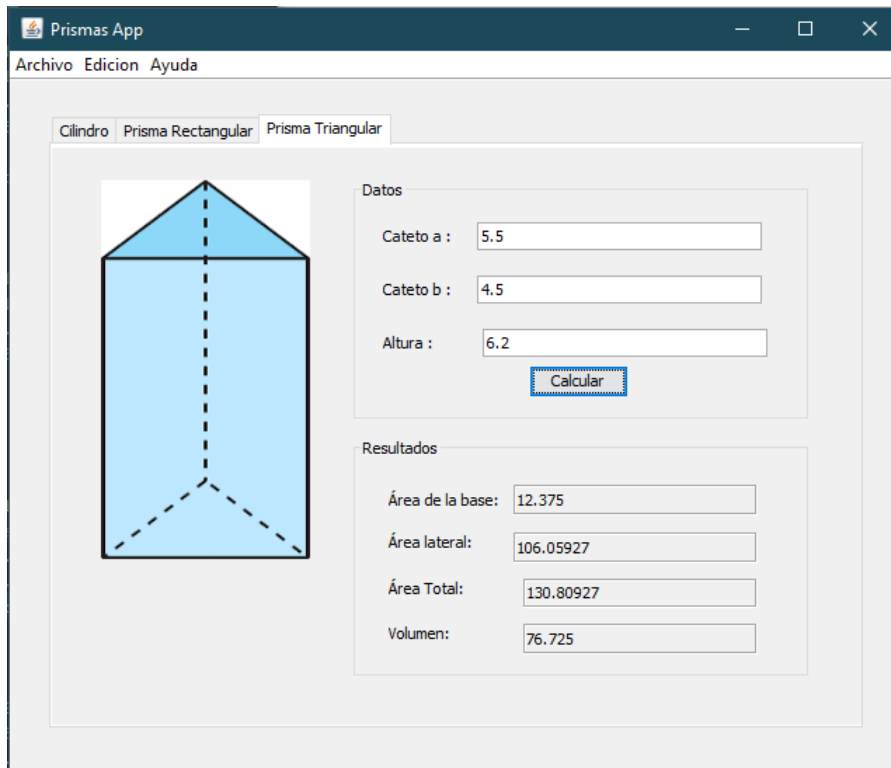
Área de la base: 32.0

Área lateral: 168.0

Área Total: 232.0

Volumen: 224.0

Ejecutándose con Prisma Triangular



Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular

Datos

Cateto a : 5.5

Cateto b : 4.5

Altura : 6.2

Calcular

Resultados

Área de la base: 12.375

Área lateral: 106.05927

Área Total: 130.80927

Volumen: 76.725

Al ejecutar la clase PrismasTest fue exitoso en todos los casos de prueba

