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 tipicos tienen 4 formas distintas: prisma regular con base rectangular, prisma regular con base en forma de triangulo 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 calculos que permiten conocer el area de la base, el area 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 rapidamente 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 la medidas correspondientes en campos localizados cerca a su lugar en la figura.

Adicionalmente la app debe contar con dialogo 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, version 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 del área de la base, área lateral, área total y volumen de un CILINDRO son:

**Area de la base** = π \* (radio)2

Diametro = 2r

Circunferencia = π \* Diametro

**Area Lateral** = Circunferencia \* altura

**Area Total =** Area Lateral + (2 \* Area de la base)

**Volumen =** Area de la base \* altura

radio ≠ “”

radio > 0

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

**Area de la base** = largo \* ancho

Perimetro = ( 2 \* largo ) + ( 2 \* ancho )

**Area Lateral** = Perimetro\* altura

**Area Total =** Area Lateral + (2 \* Area de la base)

**Volumen =** Area de la base \* altura

Largo && Ancho ≠ “”

Largo && Ancho > 0

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

cateto\_a \* cateto\_b

**Area de la base** = -----------------------------

2

Hipotenusa =

**Area Lateral** = cateto\_a + cateto\_b + hipotenusa

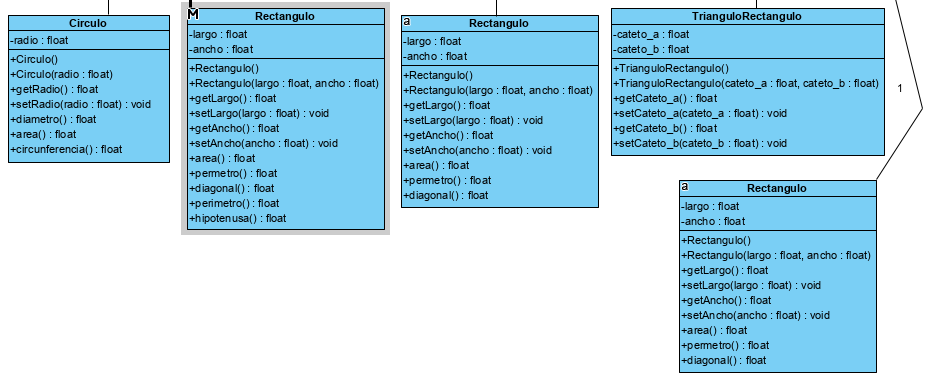
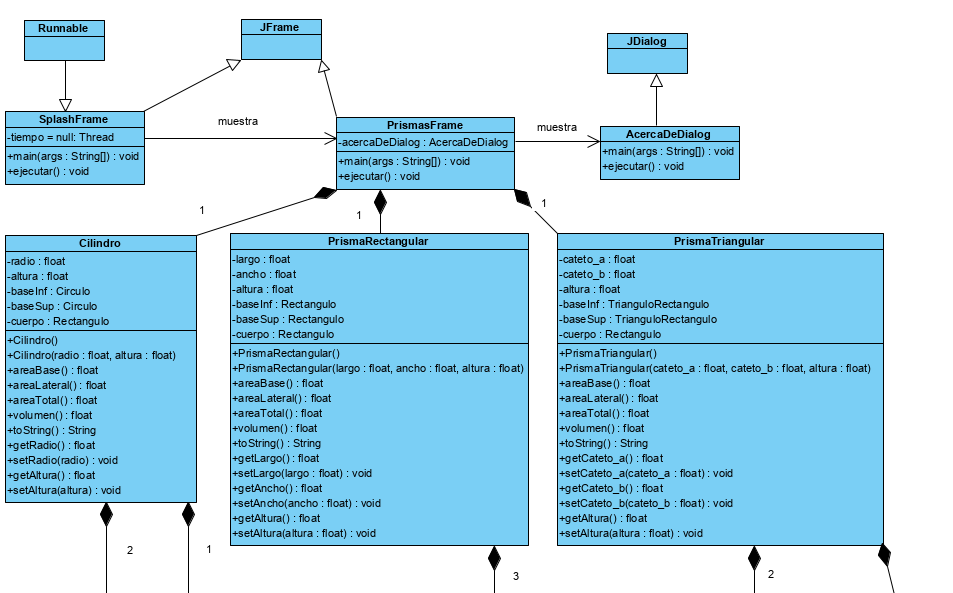
**Area Total =** Area Lateral + (2 \* Area de la base)

**Volumen =** Area de la base \* altura

cateto\_a && cateto\_b ≠ “”

cateto\_a && cateto\_b > 0

**Diseño**

**

**Código**

**AcercaDeDialog.java**

/\*------------------------------------------------------------------------------------------

:\* INSTITUTO TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "B"

:\*

:\* SEMESTRE: ENE-JUN/2020 HORA: 17-18 HRS

:\*

:\* Aplicacion 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

:\*

:\*

:\* Ultima 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 jbtnCerrarActionPerformed(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 jbtnCerrar;

private javax.swing.JLabel jlblImagenITL;

private javax.swing.JLabel jlblImagenTECNM;

// End of variables declaration

}

**Cilindro.java**

/\*------------------------------------------------------------------------------------------

:\* INSTITUTO TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "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 metodos de calculo para realizar

:\* las siguientes operaciones para un cilindro

:\* 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 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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "B"

:\*

:\* SEMESTRE: ENE-JUN/2020 HORA: 17-18 HRS

:\*

:\* Clase que realiza cálculos de un circulo

:\*

:\* Archivo : Circulo.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 formulas y operaciones para un circulo

:\* 1. Diametro

:\* 2. Area

:\* 3. Circunferencia

:\*

:\* Ultima 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 () );

}

}

**Imagenes.java**

/\*------------------------------------------------------------------------------------------

:\* INSTITUTO TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "B"

:\*

:\* SEMESTRE: ENE-JUN/2020 HORA: 17-18 HRS

:\*

:\* Clase con metodo que permite ajustar el ancho y alto de una imagen

:\*

:\*

:\* Archivo : Imagenes.java

:\* Autor : Ing. Fernando Gil

:\* Fecha : 29/08/2018

:\* Compilador : JAVA J2SE v1.8.0

:\* Descripción : El metodo estatico escalarImagen () es el más adecuado de usar para

:\* ajustar el ancho y alto de la imagen de un objeto Icon.

:\* El metodo recibe 3 argumentos: el objeto Icon y el ancho y alto al que

:\* se desea ajustar. El metodo devuelve un objeto Icon con la imagen ya

:\* redimensionada.

:\*

:\* Ultima modif:

:\* Fecha Modific? Motivo

:\*==========================================================================================

:\* 18/02/2020 F.Gil -Corregir informacion del prologo.

:\*------------------------------------------------------------------------------------------\*/

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 Imagenes {

//-----------------------------------------------------------------------------------------------------

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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "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

:\*

:\* Ultima 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 imagenes

Icon cilindro = Imagenes.escalarImagen ( jlblImagenCilindro.getIcon(),

jlblImagenCilindro.getWidth(),

jlblImagenCilindro.getHeight());

Icon p\_rect = Imagenes.escalarImagen ( jlblImagenPrismaRectangular.getIcon(),

jlblImagenPrismaRectangular.getWidth(),

jlblImagenPrismaRectangular.getHeight());

Icon p\_triang = Imagenes.escalarImagen ( jlblImagenPrismaTriangular.getIcon(),

jlblImagenPrismaTriangular.getWidth(),

jlblImagenPrismaTriangular.getHeight());

jlblImagenCilindro.setIcon ( cilindro );

jlblImagenPrismaRectangular.setIcon ( p\_rect );

jlblImagenPrismaTriangular.setIcon ( p\_triang );

}

//--------------------------------------------------------------------------

@SuppressWarnings("unchecked")

Generated Code

//--------------------------------------------------------------------------

private void jmniEdicionLimpiarActionPerformed(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 );

jlblClilindroAreaLateral.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 jmniAyudaAcercaDeActionPerformed(java.awt.event.ActionEvent evt) {

acercaDeDialog = new AcercaDeDialog( this, true );

acercaDeDialog.setVisible( true );

}

//--------------------------------------------------------------------------

private void jbtnPrisRectCalcularActionPerformed(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 {

jlblPrismRectAreaBase.setText ( pr.areaBase() + "" );

jlblPrismRectAreaLateral.setText ( pr.areaLateral()+ "" );

jlblPrismRectAreaTotal.setText ( pr.areaTotal() + "" );

jlblPrismRectVolumen.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 jbtnCilindroCalcularActionPerformed(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 {

jlblCilindroAreaBase.setText ( cil.areaBase() + "" );

jlblClilindroAreaLateral.setText ( cil.areaLateral() + "" );

jlblCilindroAreaTotal.setText ( cil.areaTotal() + "" );

jlblCilindroVolumen.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 jbtnPrismTriangCalcularActionPerformed(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 {

jlblPrismTriangAreaBase.setText ( pt.areaBase() + "" );

jlblPrismTriangAreaLateral.setText ( pt.areaLateral() + "" );

jlblPrismTriangAreaTotal.setText ( pt.areaTotal() + "" );

jlblPrismTriangVolumen.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 jmniArchivoSalirActionPerformed(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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "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 metodos de calculo para realizar

:\* las siguientes operaciones para un prisma triangular

:\* 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 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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "B"

:\*

:\* SEMESTRE: ENE-JUN/2020 HORA: 17-18 HRS

:\*

:\* Clase que realiza cálculos de un Rectangulo

:\*

:\* Archivo : Rectangulo.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 formulas y operaciones para un rectangulo

:\* 1. Area

:\* 2. Perimetro

:\* 3. Diagonal

:\*

:\* Ultima 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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "B"

:\*

:\* SEMESTRE: ENE-JUN/2020 HORA: 17-18 HRS

:\*

:\* Clase que realiza cálculos de un triangulo rectangulo

:\*

:\* Archivo : TrianguloRectangulo.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 formulas y operaciones para un triangulo

:\* rectángulo.

:\* 1. Area

:\* 2. Hipotenusa

:\* 3. Circunferencia

:\*

:\* Ultima 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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "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)

:\*

:\* Ultima 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 TECNOLOGICO DE LA LAGUNA

:\* INGENIERIA EN SISTEMAS COMPUTACIONALES

:\* TOPICOS AVANZADOS DE PROGRAMACION "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.

:\*

:\* Ultima 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 );

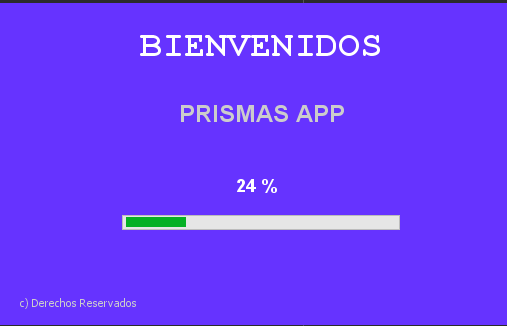
}

}

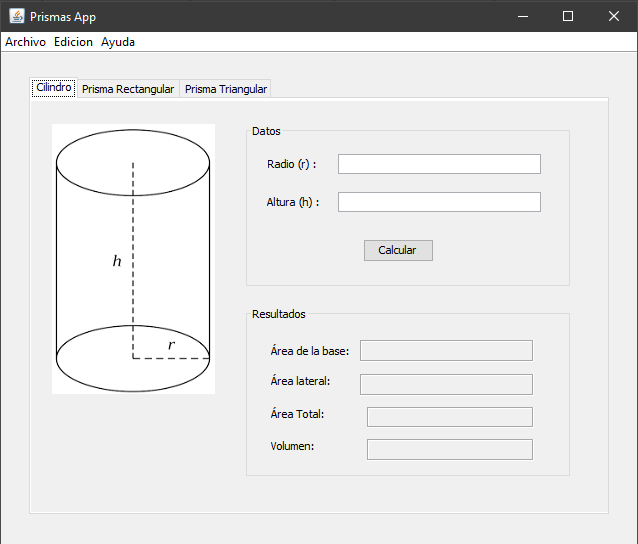
**Prueba de Ejecución**

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

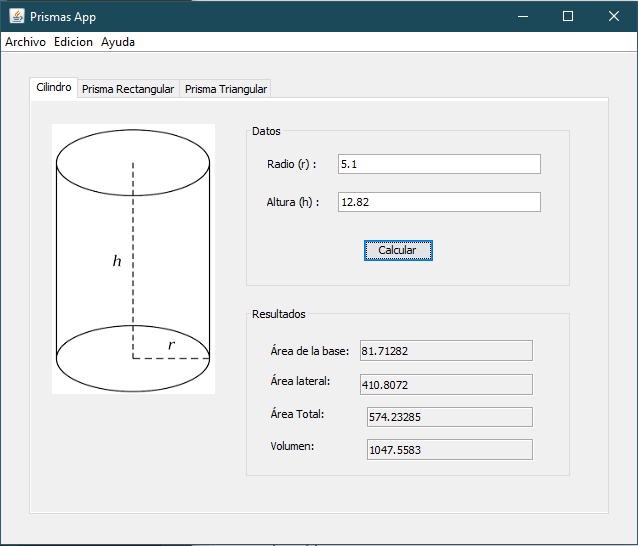
Este es el splash



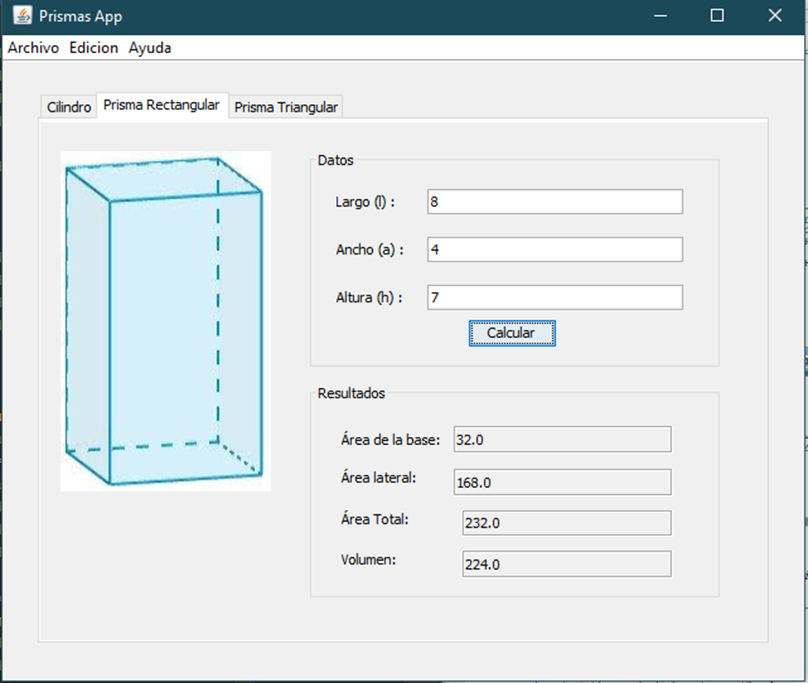
Esta es la interfaz inicial



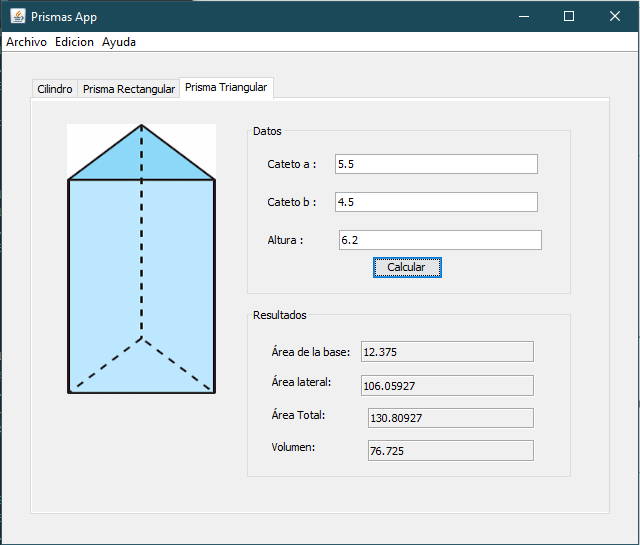
Ejecutándose con Cilindro



Ejecutandose con Prisma Rectangular



Ejecutandose con Prisma Triangular



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

