



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. U2P03

# **APLICACIÓN PRISMASAPP USANDO BEANS PROPIOS**

ALUMNO:

18131209 José Misael Adame Sandoval

PROFESOR:

Ing. Luis Fernando Gil Vázquez

Torreón, Coah. a 7 de Abril de 2020

## Ejercicio 1

Se requiere diseñar e implementar como JavaBeans 3 diferentes prismas: Cilindro, Prisma Rectangular y Prisma Triangular.

Los beans debe ser del tipo visible es decir deben tener una interfaz visual de usuario. Cada bean debe contar con propiedades editables en tiempo de diseño que permitan especificar las dimensiones que lo definen tales como radio, altura, largo y ancho de la base, etc dependiendo del tipo de prisma. Además de una propiedad que permita cambiar la imagen del prisma.

En su implementación el bean debe encapsular un objeto de la clase base del prisma, por ejemplo: el bean JCilindro debe componerse de un objeto de la clase Cilindro que previamente ya se ha desarrollado.

Cada bean debe proporcionar métodos para conocer el área de su base, área lateral, área total y volumen.

Incorporar al bean un menú contextual que proporcione una opción EDITAR VALORES que presente un diálogo donde se puedan capturar las dimensiones del prisma y una segunda opción llamada ACERCA DE..., esta función debe mostrar un mensaje de diálogo con el nombre del bean, versión y los nombre del autor.

Los beans deben ser capaces de disparar un evento al completar la captura de datos en el diálogo EDITAR VALORES, dicho evento debe enviar como información los valores antiguos y los nuevos valores a sus listeners.

Se debe usar la característica del BeanInfo para que las propiedades del bean aparezcan en primer lugar en la paleta de propiedades, cada propiedad debe desplegar una leyenda legible en el nombre de la propiedad y su correspondiente descripción. Además el bean debe contar con un icono representativo en la paleta de componentes de Netbeans.

Adicional a los 3 beans para los prismas se solicita diseñar e implementar un bean Acerca De que pueda ser reutilizado en posteriores proyectos. El bean Acerca De debe permitir personalizar en tiempo de diseño mediante la paleta de propiedades todas las leyendas del diálogo acerca de, así como los dos logos incluidos. Este bean no genera eventos.

Todos los componentes que se incluyan en el código del bean **deben tener un nombre de variable adecuado no el nombre de default.**

Los beans debe incluirse en la librería del curso en el package **mx.edu.itl.beans** y empacarse en un archivo JAR.

Posteriormente ya que se tengan listos los beans de los 3 prismas se deberá crear una nueva versión basada en la aplicación PrismasApp en la cual se reutilizarán los 3 beans diseñados. El Frame principal de PrismasApp debe ser listener de los eventos que generan los prismas, de tal manera que si uno captura los valores del prisma por medio del diálogo que trae incorporado el bean el Frame recibirá el evento y deberá mostrar dichos valores en el formulario para leer los datos del prisma. De igual forma si los datos del prisma se capturan a través del formulario del Frame estos nuevos valores se deben reflejar en el bean. En la siguiente página se anexa un esquema que ilustra esta funcionalidad.

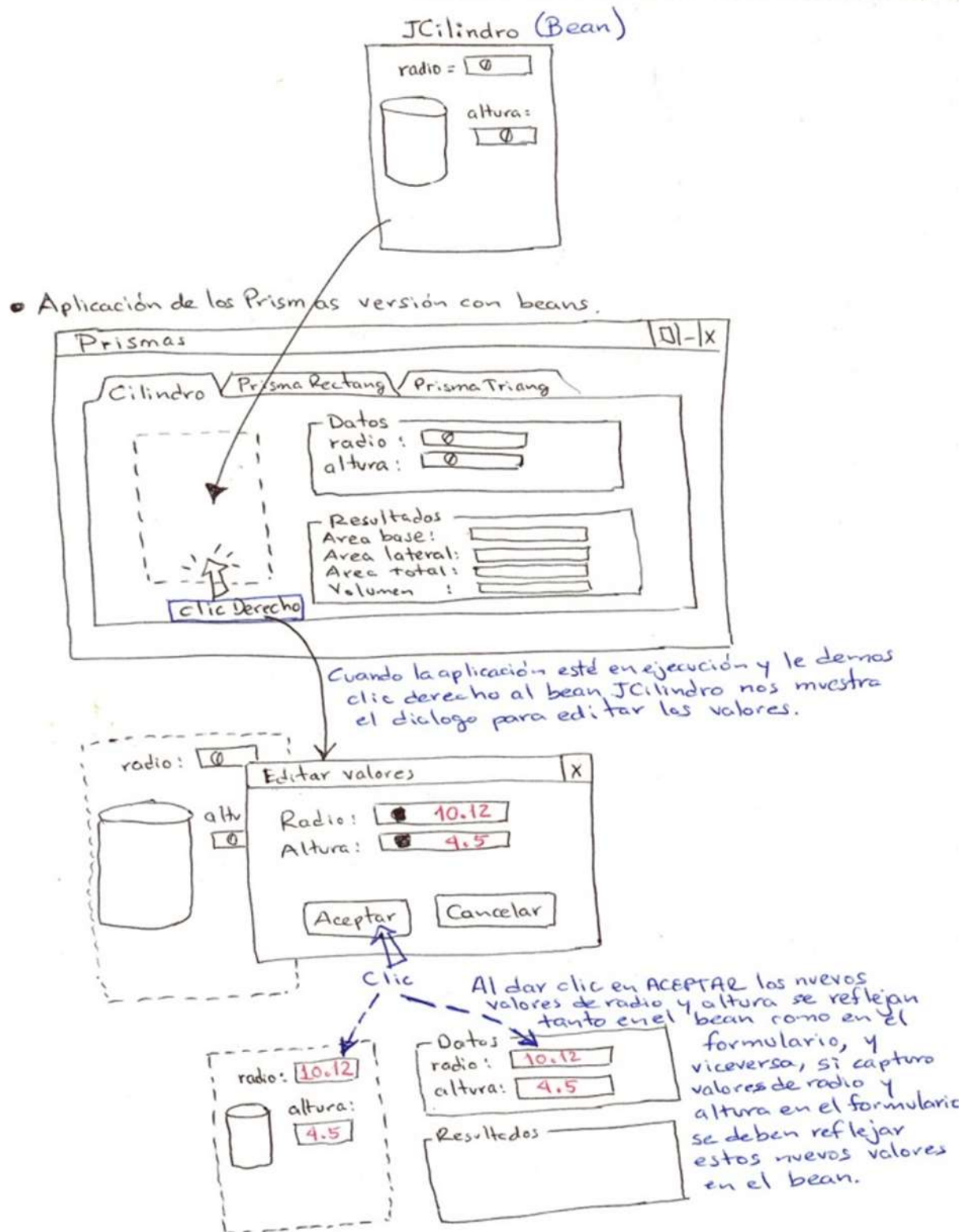
La opción Archivo -> Acerca de debe hacer uso del bean Acerca De.

En la sección de ANALISIS describir cómo será diseñado cada bean, es decir, cuál será el componente visual base del bean, qué otros componentes lo integrarán y qué propiedades tendrá cada uno.

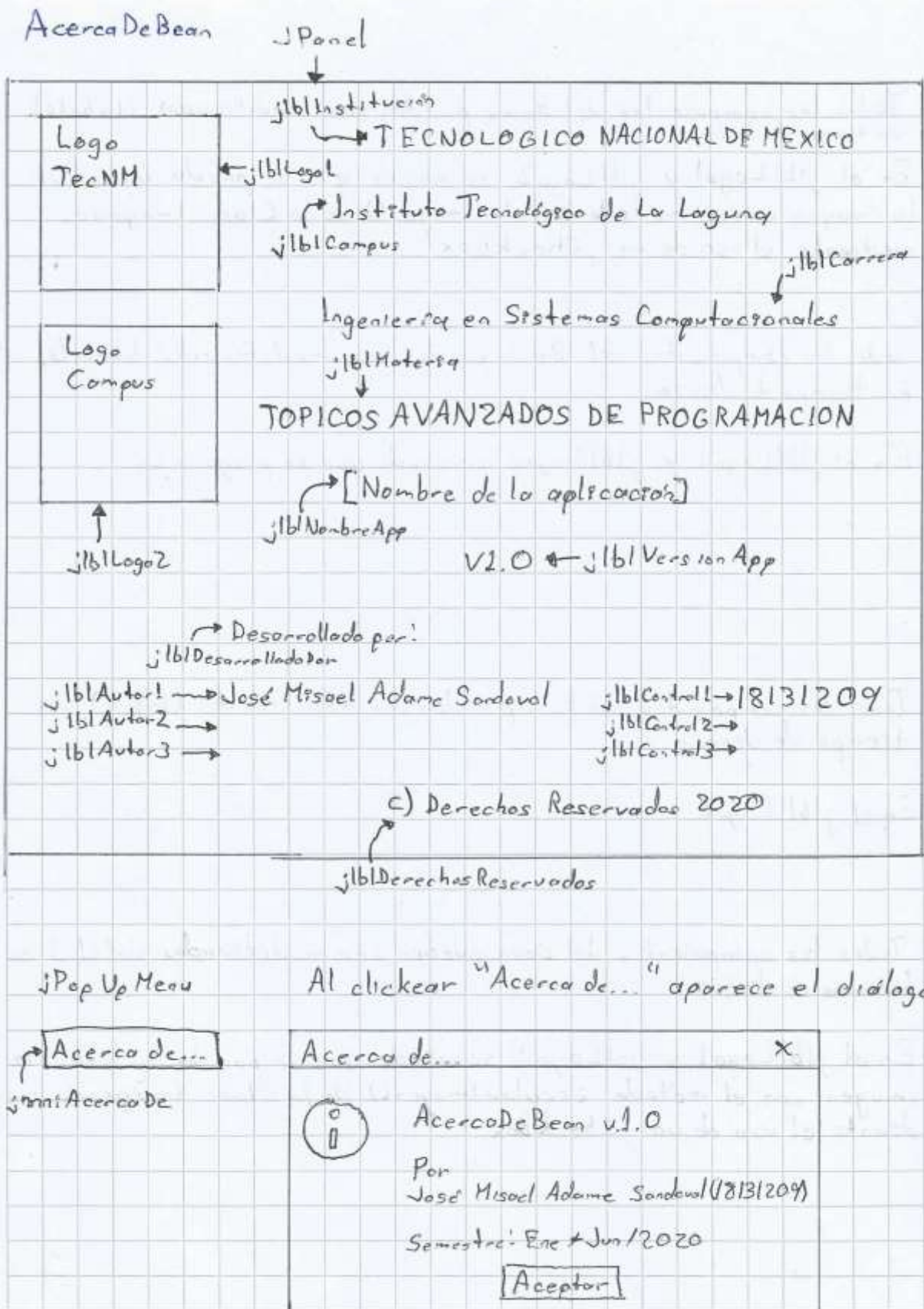
En la sección de DISEÑO va el diagrama de clases UML de cada bean.

En la sección de CODIGO va el código de todas las clases JAVA con la calidad establecida.

En la sección de PRUEBA DE EJECUCION incluir una captura de la paleta de componentes de Netbeans donde se aprecien los iconos de los 4 beans diseñados ( JCilindro, JPrismaRectangular, JPrismaTriangular, y AcercaDeBean ). Incluir capturas de pantallas de la paleta de propiedades de cada bean donde se aprecien las propiedades que definen las dimensiones del prisma. Incluir capturas de pantallas de la aplicación PrismasBeansApp en ejecución.

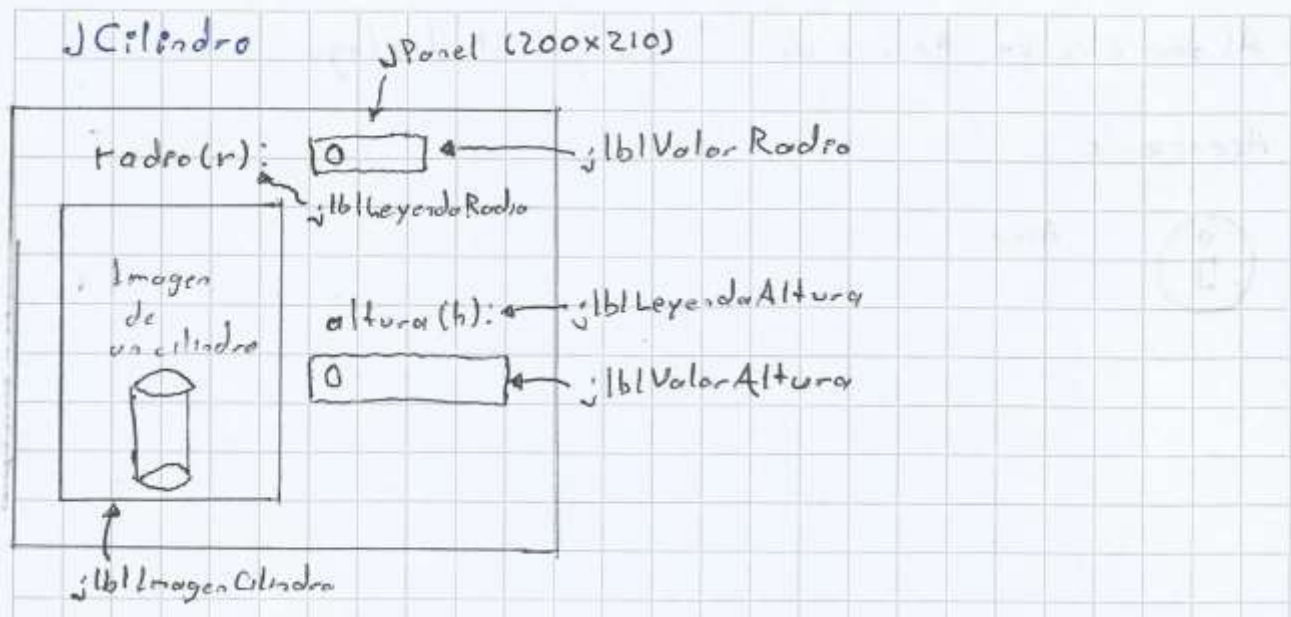


## Análisis



Todos los componentes del Bean pueden ser modificados (labels) en tiempos de diseño.

En el `JblLogo1` y `JblLogo2` se añade que se pueda ajustar la imagen con el método `escalarImagen()` de la Clase `Imagenes` mediante el uso de un `CheckBox`.

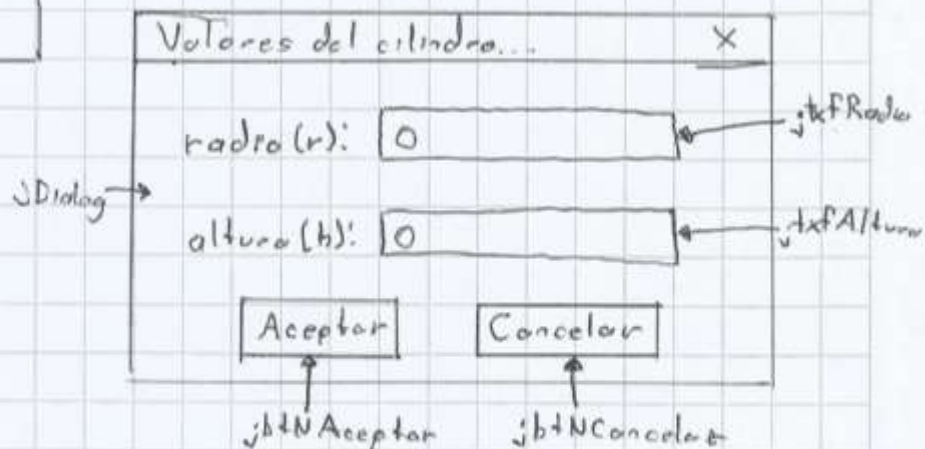




¡PopUpMenu

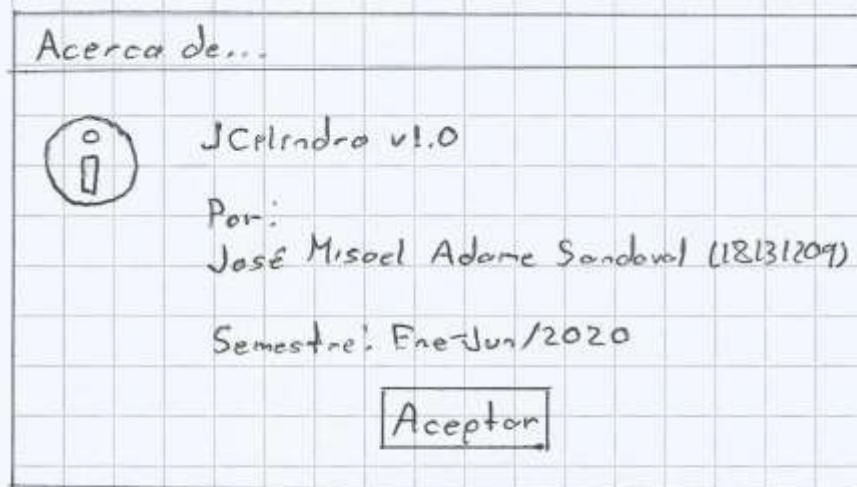
Valores del prisma...  
Acerca de...

Al dar clic en "Valores del Prisma..." aparece el siguiente JDialog Cilindro

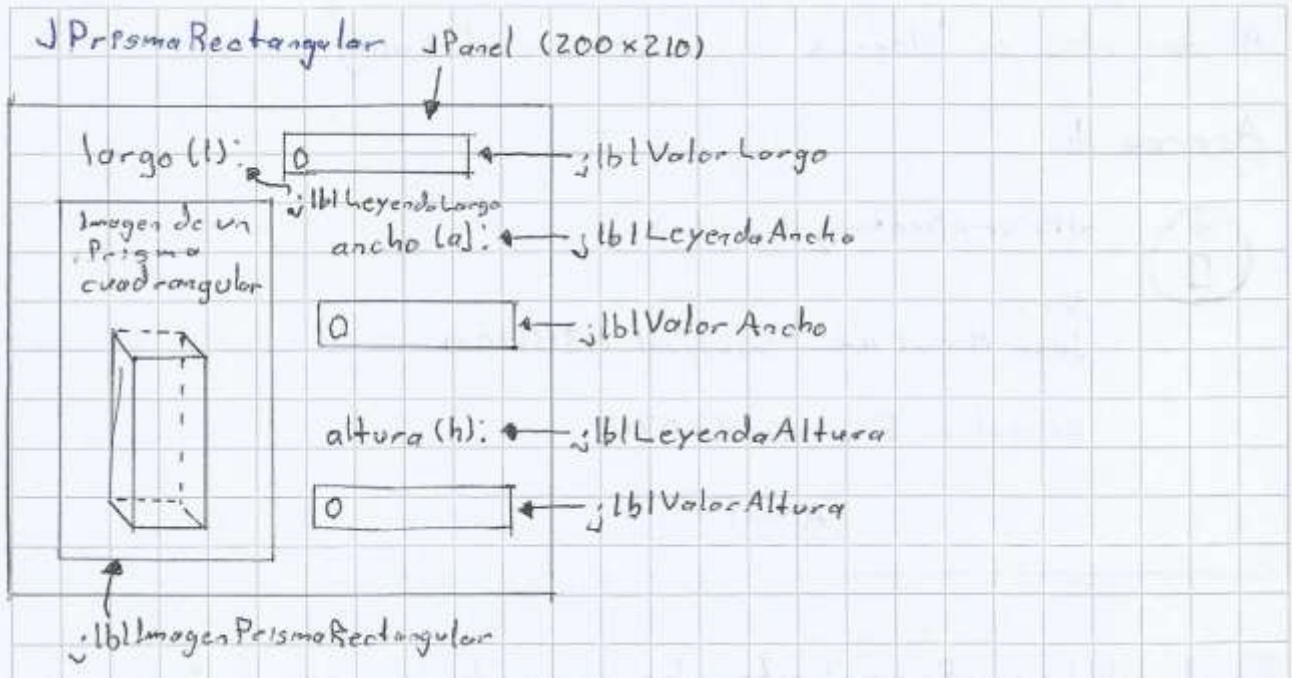


Todas las componentes del JCilindro pueden ser modificadas (labels) en tiempo de diseño

Al dar clic en "Acerca de..." aparece el diálogo



En el ¡lblImagenCilindro se añade una propiedad para que se pueda ajustar la imagen con el método `escalarImagen()` de la Clase `Imagenes` mediante el uso de un ¡CheckBox.

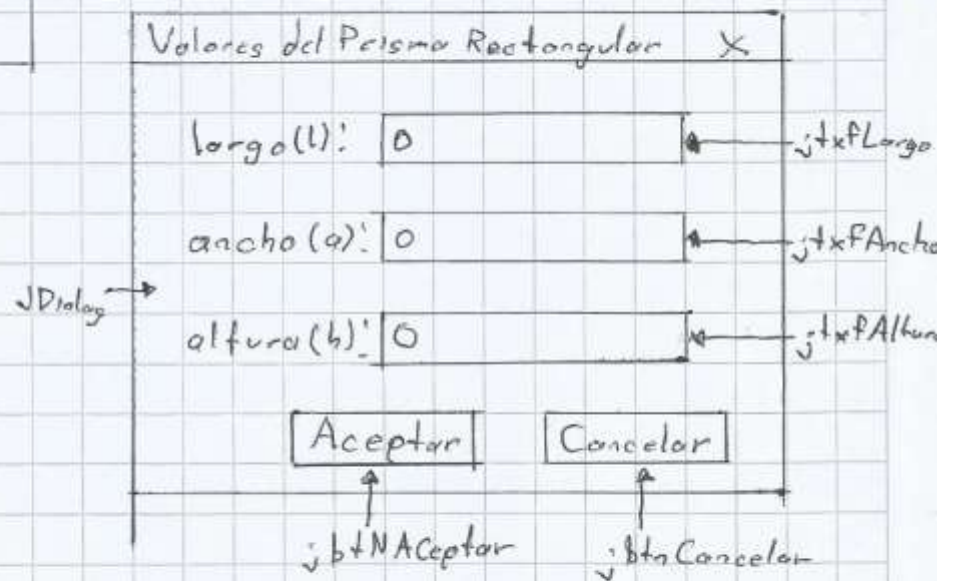


**jPopupMenu**

Valores del prisma...

Acercar de...


Al dar clic en "Valores del Prisma..." aparece el siguiente **JDialogPrismaRectangular**



Todos los componentes del **JPrismaRectangular** pueden ser modificados (labels) en tiempo de diseño

Al dar clic en "Acercar de..." aparece el diálogo

Acercar de...



JPismaRectangular v1.0

Por:  
José Misael Adame Sandoval (18B1209)

Semestre: Ene-Jun/2020

Aceptar

En el `jpgImagenPismaRectangular` se añade una propiedad para que se pueda ajustar la imagen con el método `escalarImagen()` de la clase `Imagenes` mediante el uso de un `CheckBox`

JPismaTriangular

`jpgImagenPismaRectangular` → `JPanl (200x210)`

`jpgLegendaCatetoA` → cateto (a):  `jpgValorCatetoA`

`cateto (b):` `jpgLegendaCatetoB`  `jpgValorCatetoB`

`altura (h):` `jpgLegendaAltura`  `jpgValorAltura`


`jpgImagenPismaTriangular` → 

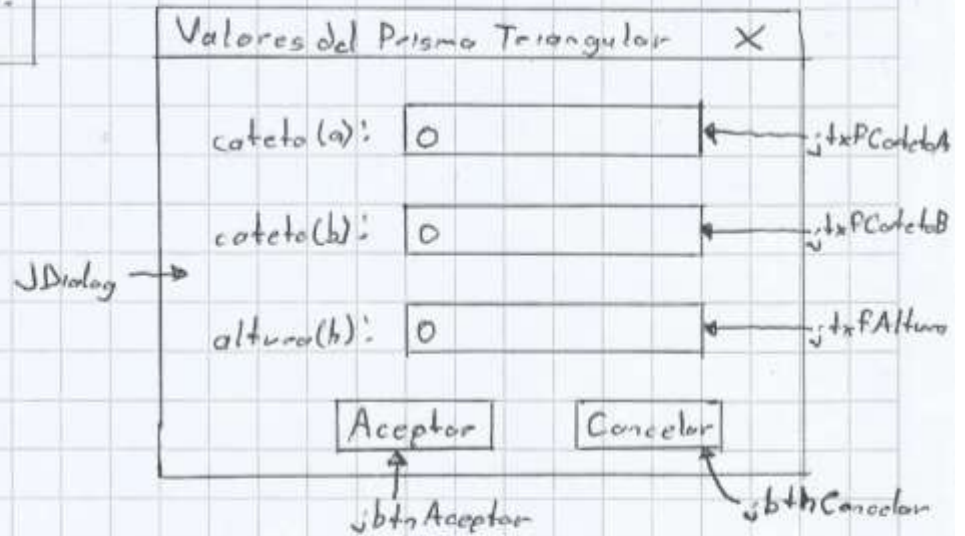
Imagen de un Pisma Rectangular



¡PopUpMenu

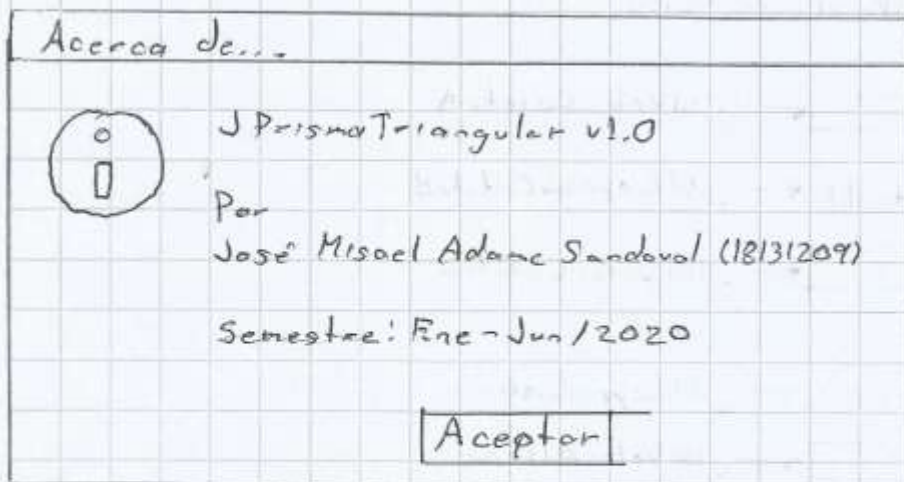
Valores del prisma...  
Acerca de...

Al dar clic en "Valores del Prisma..." aparece el siguiente JDialogPrismaTriangular



Todos los componentes del JPrismaTriangular pueden ser modificados (labels) en tiempo de diseño

Al dar clic en "Acerca de..." aparece el diálogo



En el JbblImagenPrismaTriangular se añade una propiedad para que se pueda ajustar la imagen con el método `escalarImagen()` de la clase `Imagenes` mediante el uso de un `CheckBox`.



## Código

### 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 mx.edu.itl.figuras;

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 () );
}
}
```

## 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 mx.edu.itl.figuras;

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
:*          TEMAS 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 vl.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 mx.edu.itl.figuras;

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 ();
}
}

```

## 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 mx.edu.itl.prismas;
import mx.edu.itl.figuras.Circulo;

```

```
import mx.edu.itl.figuras.Rectangulo;

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 );
}

}
```

## PrismaRectangular.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 Rectangular
*:
*: Archivo : PrismaRectangular.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 Rectangular
*: 1. Área Base
*: 2. Área Lateral
*: 3. Área Total
*: 4. Volumen
*:
*: Última modif:
*: Fecha Modificó Motivo
*:=====
*: 23/feb/2020 Misael Adame Agregar prólogo.
*:-----*/

package mx.edu.itl.prismas;

import mx.edu.itl.figuras.Rectangulo;

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 );  
}  
}
```

**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      : PrismaTriangular.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 mx.edu.itl.prismas;

import mx.edu.itl.figuras.Rectangulo;
import mx.edu.itl.figuras.TrianguloRectangulo;

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 );
}

}
```

**AcercaDeBean.java**

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Bean GUI de un Acerca de..
*:
*: Archivo      : AcercaDeBean.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 01/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Bean que muestra información sobre el autor de la
*:                aplicación que está realizando
*:
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.beans;

import javax.swing.Icon;
import javax.swing.JOptionPane;
import mx.edu.itl.util.Imagenes;

public class AcercaDeBean extends javax.swing.JPanel {

    private boolean bolLogo1Ajustado = false;
    private boolean bolLogo2Ajustado = false;

    //-----

    public AcercaDeBean () {
        initComponents ();
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    private void jMenuItemAcercaDeActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog (
            this,
            "AcercaDeBean v1.0 \n\n" +
            "Por: \n" +
            "Jose Misael Adame Sandoval (18131209) \n\n" +
            "Semestre: Ene - Jun/2020",
            "Acerca de...",
            JOptionPane.INFORMATION_MESSAGE
        );
    }

    //-----

    public String getInstitucion () {
        return jlblInstitucion.getText();
    }

    //-----

    public void setInstitucion ( String institucion ) {
        jlblInstitucion.setText ( institucion );
    }

```



```
//-----  
  
public String getCampus () {  
    return lblCampus.getText();  
}  
  
//-----  
  
public void setCampus ( String campus ) {  
    lblCampus.setText ( campus );  
}  
  
//-----  
  
public String getCarrera () {  
    return lblCarrera.getText();  
}  
  
//-----  
  
public void setCarrera ( String carrera ) {  
    lblCarrera.setText ( carrera );  
}  
  
//-----  
  
public String getMateria () {  
    return lblMateria.getText ();  
}  
  
//-----  
  
public void setMateria ( String materia ) {  
    lblMateria.setText ( materia );  
}  
  
//-----  
  
public String getNombreApp () {  
    return lblNombreApp.getText ();  
}  
  
//-----  
  
public void setNombreApp ( String nombre_app ) {  
    lblNombreApp.setText ( nombre_app );  
}  
  
//-----  
  
public String getVersionApp () {  
    return lblVersionApp.getText ();  
}  
  
//-----  
  
public void setVersionApp ( String version_app ) {  
    lblVersionApp.setText ( version_app );  
}  
  
//-----  
  
public String getDesarrolladoPor () {  
    return lblDesarrolladoPor.getText ();  
}  
  
//-----  
  
public void setDesarrolladoPor ( String desarrollado_por ) {  
    lblDesarrolladoPor.setText ( desarrollado_por );  
}  
  
//-----  
  
public String getAutor1 () {
```

```
        return jlblAutor1.getText ();
    }

    //-----

    public void setAutor1 ( String autor_1 ) {
        jlblAutor1.setText ( autor_1 );
    }

    //-----

    public String getAutor2 () {
        return jlblAutor2.getText ();
    }

    //-----

    public void setAutor2 ( String autor_2 ) {
        jlblAutor2.setText ( autor_2 );
    }

    //-----

    public String getAutor3 () {
        return jlblAutor3.getText ();
    }

    //-----

    public void setAutor3 ( String autor_3 ) {
        jlblAutor3.setText ( autor_3 );
    }

    //-----

    public String getControl1 () {
        return jlblControl1.getText ();
    }

    //-----

    public void setControl1 ( String control_1 ) {
        jlblControl1.setText ( control_1 );
    }

    //-----

    public String getControl2 () {
        return jlblControl2.getText ();
    }

    //-----

    public void setControl2 ( String control_2 ) {
        jlblControl2.setText ( control_2 );
    }

    //-----

    public String getControl3 () {
        return jlblControl2.getText ();
    }

    //-----

    public void setControl3 ( String control_3 ) {
        jlblControl3.setText ( control_3 );
    }

    //-----

    public Icon getLogo1 () {
        return jlblLogo1.getIcon ();
    }
}
```

```
//-----  
public void setLogo1 ( Icon logo_1 ) {  
    lblLogo1.setIcon ( logo_1 );  
}  
  
//-----  
public Icon getLogo2 () {  
    return lblLogo2.getIcon ();  
}  
  
//-----  
public void setLogo2 ( Icon logo_2 ) {  
    lblLogo2.setIcon ( logo_2 );  
}  
  
//-----  
public String getDerechosReservados () {  
    return lblDerechosReservados.getText ();  
}  
  
//-----  
public void setDerechosReservados ( String derechos_reservados ) {  
    lblDerechosReservados.setText ( derechos_reservados );  
}  
  
//-----  
public boolean isLogo1Ajustado () {  
    return bolLogo1Ajustado;  
}  
  
//-----  
public void setLogo1Ajustado ( boolean valor ) {  
    bolLogo1Ajustado = valor;  
  
    if ( bolLogo1Ajustado == true ) {  
        // No funciona correctamente lo que nos proporcionó para ajustar las imágenes  
        lblLogo1.setIcon ( Imagenes.escalarImagen ( lblLogo1.getIcon (),  
                                                    lblLogo1.getWidth (),  
                                                    lblLogo1.getHeight () ) );  
    }  
}  
  
//-----  
public boolean isLogo2Ajustado () {  
    return bolLogo2Ajustado;  
}  
  
//-----  
public void setLogo2Ajustado ( boolean valor ) {  
    bolLogo2Ajustado = valor;  
    if ( bolLogo2Ajustado == true ) {  
        // No funciona correctamente lo que nos proporcionó para ajustar las imágenes  
        lblLogo2.setIcon ( Imagenes.escalarImagen ( lblLogo2.getIcon (),  
                                                    lblLogo2.getWidth (),  
                                                    lblLogo2.getHeight () ) );  
    }  
}  
}
```

**IPrisma.java**

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase Interface del Prisma
*:
*: Archivo      : IPrisma.java
*: Autor       : José Misael Adame Sandoval      18131209
*: Fecha       : 06/Abr/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Clase Interface que es en donde se basarán los Prismas
*:
*:
*: Última modif:
*: Fecha       Modificó           Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.beans;

public interface IPrisma {

    public float areaBase ();

    //-----

    public float areaLateral ();

    //-----

    public float areaTotal ();

    //-----

    public float volumen ();

}

```

**JCilindro.java**

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Bean GUI de un JCilindro
*:
*: Archivo      : JCilindro.java
*: Autor       : José Misael Adame Sandoval      18131209
*: Fecha       : 01/Abr/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Bean que muestra información sobre el radio y altura
*:              de un cilindro, junto con su imagen.
*:
*:
*: Última modif:
*: Fecha       Modificó           Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.beans;

import java.util.ArrayList;
import javax.swing.Icon;
import javax.swing.JOptionPane;

```



```
import mx.edu.itl.eventos.JCilindroEvent;  
import mx.edu.itl.prismas.Cilindro;  
import mx.edu.itl.eventos.JValoresPrismasListener;  
import mx.edu.itl.util.Imagenes;  
  
public class JCilindro extends javax.swing.JPanel implements IPrisma {  
  
    //-----  
  
    private Cilindro cilindro = new Cilindro ();  
    private boolean bolImagenCilindroAjustado = false;  
    public JDialogCilindro JCilindroValores;  
    private ArrayList<JValoresPrismasListener> listeners = new ArrayList<> ();  
  
    //-----  
  
    public JCilindro() {  
        initComponents ();  
  
        cilindro = new Cilindro ();  
    }  
  
    //-----  
  
    public JCilindro ( float radio, float altura ) {  
        cilindro = new Cilindro ( radio, altura );  
    }  
  
    //-----  
  
    public String getLeyendaRadio () {  
        return lblLeyendaRadio.getText();  
    }  
  
    //-----  
  
    public void setLeyendaRadio ( String leyenda_radio ) {  
        lblLeyendaRadio.setText ( leyenda_radio );  
    }  
  
    //-----  
  
    public float getRadio () {  
        return cilindro.getRadio ();  
    }  
  
    //-----  
  
    public void setRadio ( float radio ) {  
        cilindro.setRadio ( radio );  
        lblValorRadio.setText ( Float.toString ( radio ) );  
    }  
  
    //-----  
  
    public String getLeyendaAltura () {  
        return lblLeyendaAltura.getText();  
    }  
  
    //-----  
  
    public void setLeyendaAltura ( String leyenda_altura ) {  
        lblLeyendaAltura.setText ( leyenda_altura );  
    }  
  
    //-----  
  
    public float getAltura () {  
        return cilindro.getAltura ();  
    }  
  
    //-----  
  
    public void setAltura ( float altura ) {
```

```

        cilindro.setAltura ( altura );
        lblValorAltura.setText ( Float.toString ( altura ) );
    }

//-----

public Icon getCilindro () {
    return lblImagenCilindro.getIcon ();
}

//-----

public void setCilindro ( Icon cilindro ) {
    lblImagenCilindro.setIcon ( cilindro );
}

public boolean isImagenCilindroAjustado () {
    return bolImagenCilindroAjustado;
}

//-----

public void setImagenCilindroAjustado ( boolean valor ) {
    bolImagenCilindroAjustado = valor;

    if ( bolImagenCilindroAjustado == true ) {
        // No funciona correctamente lo que nos proporcionó para ajustar las imágenes
        lblImagenCilindro.setIcon ( Imagenes.escalarImagen ( lblImagenCilindro.getIcon (),
                                                                lblImagenCilindro.getWidth (),
                                                                lblImagenCilindro.getHeight () ) );
    }
}

//-----

@SuppressWarnings("unchecked")
Generated Code

private void jmnAcercaDeCilindroActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog (
        this,
        "JCilindro v1.0 \n\n" +
        "Por: \n" +
        "Jose Misael Adame Sandoval (18131209) \n\n" +
        "Semestre: Ene - Jun/2020",
        "Acerca de...",
        JOptionPane.INFORMATION_MESSAGE
    );
}

//-----

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

    // JDialogCilindro.ejecutar();

    JCilindroValores = new JDialogCilindro ( this, true );
    JCilindroValores.setVisible ( true );
}

//-----

public void update () {

    JCilindroEvent evento = new JCilindroEvent ( cilindro.getRadio (),
                                                cilindro.getAltura (),
                                                JCilindroValores.anteriorRadio,
                                                JCilindroValores.anteriorAltura );

    fireJCilindroEvent ( evento );
}

//-----

```

```

public void addJCilindroListener ( JValoresPrismasListener listener ) {
    listeners.add ( listener );
}

public void removeJCilindroListener ( JValoresPrismasListener listener ) {
    listeners.remove ( listener );
}

//-----

public void fireJCilindroEvent ( JCilindroEvent evento ) {
    for ( JValoresPrismasListener listener : listeners ) {
        listener.valoresCilindro ( evento );
    }
}
}

```

## JDialoCilindro.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 Cilindro
*:
*: Archivo      : JDialoCilindro.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 06/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación JDialo que permite ingresar los valores del radio y altura
*:               del JCilindro
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.beans;

import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class JDialoCilindro extends javax.swing.JDialog {

    //-----

    private JCilindro jCilindro;
    public float anteriorAltura;
    public float anteriorRadio;

    //-----

    public JDialoCilindro ( JPanel parent, boolean modal ) {

        initComponents ();

        jCilindro = ( JCilindro ) parent;

        anteriorRadio = jCilindro.getRadio ();
        anteriorAltura = jCilindro.getAltura ();

        txtfRadio.setText ( anteriorRadio + " " );
        txtfAltura.setText ( anteriorAltura + " " );
    }

    //-----

```

@SuppressWarnings("unchecked")  
Generated Code

```
//-----

private void jButtonAceptarActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        float nuevoRadio = Float.parseFloat ( jTextFieldRadio.getText () );
        float nuevaAltura = Float.parseFloat ( jTextFieldAltura.getText () );
        if ( nuevoRadio < 0 || nuevaAltura < 0 )
            JOptionPane.showMessageDialog (
                this,
                "Debe proporcionar un valor numérico positivo en todos los campos",
                "Error",
                JOptionPane.ERROR_MESSAGE
            );
        else {
            jCilindro.setRadio ( nuevoRadio );
            jCilindro.setAltura ( nuevaAltura );
            jCilindro.update ();
            dispose ();
        }
    } catch ( NumberFormatException ex ) {
        JOptionPane.showMessageDialog (
            this,
            "Debe proporcionar un valor numérico positivo en todos los campos",
            "Error",
            JOptionPane.ERROR_MESSAGE
        );
    }
}

//-----

private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible ( false );
}

//-----

public static void ejecutar() {
    /* Set the Windows look and feel */

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            JDialogCilindro dialog = new JDialogCilindro(new JPanel(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

//-----

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

**JDialogPrismaRectangular.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 Prisma Rectangular
*:
*: Archivo      : JDialogPrismaRectangular.java
*: Autor       : José Misael Adame Sandoval      18131209
*: Fecha       : 06/Abr/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Aplicación JDialog que permite ingresar los valores del largo, ancho
*:              y altura del JPrismaRectangular
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:-----
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.beans;

import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class JDialogPrismaRectangular extends javax.swing.JDialog {

    //-----

    private JPrismaRectangular jPrismaRectangular;
    public float anteriorAltura;
    public float anteriorLargo;
    public float anteriorAncho;

    //-----

    public JDialogPrismaRectangular( JPanel parent, boolean modal ) {

        initComponents ();

        jPrismaRectangular = ( JPrismaRectangular ) parent;

        anteriorLargo = jPrismaRectangular.getLargo ();
        anteriorAncho = jPrismaRectangular.getAncho ();
        anteriorAltura = jPrismaRectangular.getAltura ();
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    private void jButtonAceptarActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            float nuevoLargo = Float.parseFloat ( jtxfLargo.getText () );
            float nuevoAncho = Float.parseFloat ( jtxfAncho.getText () );
            float nuevaAltura = Float.parseFloat ( jtxfAltura.getText () );
            if ( nuevoLargo < 0 || nuevoAncho < 0 || nuevaAltura < 0 )
                JOptionPane.showMessageDialog (
                    this,
                    "Debe proporcionar un valor numérico positivo en todos los campos",
                    "Error",
                    JOptionPane.ERROR_MESSAGE
                );
            else {
                jPrismaRectangular.setLargo ( nuevoLargo );
                jPrismaRectangular.setAncho ( nuevoAncho );
                jPrismaRectangular.setAltura ( nuevaAltura );
                jPrismaRectangular.update ();
            }
        }
    }
}

```

```

        dispose ();
    }
} catch ( NumberFormatException ex ) {
    JOptionPane.showMessageDialog (
        this,
        "Debe proporcionar un valor numérico positivo en todos los campos",
        "Error",
        JOptionPane.ERROR_MESSAGE
    );
}

}

//-----

private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible ( false );
}

//-----

public static void ejecutar() {
    /* Set the Windows look and feel */
    Look and feel setting code (optional)

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            JDialogPrismaRectangular dialog = new JDialogPrismaRectangular(new JPanel(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

//-----

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

```

### JDialogPrismaTriangular.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 Prisma Triangular
*:
*: Archivo      : JDialogPrismaTriangular.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 06/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación JDialog que permite ingresar los valores del cateto A, cateto B
*:               y altura del JPrismaTriangular
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.beans;

```

```

import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class JDialogPrismaTriangular extends javax.swing.JDialog {

    //-----

    private JPrismaTriangular jPrismaTriangular;
    public float anteriorAltura;
    public float anteriorCatetoA;
    public float anteriorCatetoB;

    //-----

    public JDialogPrismaTriangular( JPanel parent, boolean modal ) {

        initComponents ();

        jPrismaTriangular = ( JPrismaTriangular ) parent;

        anteriorCatetoA = jPrismaTriangular.getCateto_a ();
        anteriorCatetoB = jPrismaTriangular.getCateto_b ();
        anteriorAltura = jPrismaTriangular.getAltura ();
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    private void jButtonAceptarActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            float nuevoCatetoA = Float.parseFloat ( jTextFieldCatetoA.getText () );
            float nuevoCatetoB = Float.parseFloat ( jTextFieldCatetoB.getText () );
            float nuevaAltura = Float.parseFloat ( jTextFieldAltura.getText () );
            if ( nuevoCatetoA < 0 || nuevoCatetoB < 0 || nuevaAltura < 0 )
                JOptionPane.showMessageDialog (
                    this,
                    "Debe proporcionar un valor numérico positivo en todos los campos",
                    "Error",
                    JOptionPane.ERROR_MESSAGE
                );
            else {
                jPrismaTriangular.setCateto_a ( nuevoCatetoA );
                jPrismaTriangular.setCateto_b ( nuevoCatetoB );
                jPrismaTriangular.setAltura ( nuevaAltura );
                jPrismaTriangular.update ();
                dispose ();
            }
        } catch ( NumberFormatException ex ) {
            JOptionPane.showMessageDialog (
                this,
                "Debe proporcionar un valor numérico positivo en todos los campos",
                "Error",
                JOptionPane.ERROR_MESSAGE
            );
        }
    }

    //-----

    private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible ( false );
    }

    //-----

    public static void ejecutar() {
        /* Set the Windows look and feel */

        /* Create and display the dialog */
        java.awt.EventQueue.invokeLater(new Runnable() {

```



```

        public void run() {
            JDialogPrismaTriangular dialog = new JDialogPrismaTriangular(new JPanel(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

//-----

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

```

## JPrismaRectangular.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Bean GUI de un JPrismaRectangular
*:
*: Archivo      : JPrismaRectangular.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 01/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Bean que muestra información sobre el largo, ancho y altura
*:                de un Prisma Rectangular, junto con su imagen.
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.beans;

import java.util.ArrayList;
import javax.swing.Icon;
import javax.swing.JOptionPane;
import mx.edu.itl.eventos.JPrismaRectangularEvent;
import mx.edu.itl.prismas.PrismaRectangular;
import mx.edu.itl.eventos.JValoresPrismasListener;
import mx.edu.itl.util.Imagenes;

public class JPrismaRectangular extends javax.swing.JPanel implements IPrisma {

    //-----

    private PrismaRectangular prismaRectangular = new PrismaRectangular ();
    private boolean bolImagenPrismaRectangularAjustado = false;
    public JDialogPrismaRectangular JPrismaRectangularValores;
    private ArrayList<JValoresPrismasListener> listeners = new ArrayList ();

    //-----

    public JPrismaRectangular() {
        initComponents();

        prismaRectangular = new PrismaRectangular();
    }

    //-----

```

```
public JPrismaRectangular ( float largo, float ancho , float altura ) {
    prismaRectangular = new PrismaRectangular ( largo, ancho, altura );
}

//-----

public String getLeyendaLargo () {
    return jlblLeyendaLargo.getText();
}

//-----

public void setLeyendaLargo ( String leyenda_largo ) {
    jlblLeyendaLargo.setText ( leyenda_largo );
}

//-----

public float getLargo() {
    return prismaRectangular.getLargo();
}

//-----

public void setLargo ( float largo ) {
    prismaRectangular.setLargo ( largo );
    jlblValorLargo.setText ( Float.toString ( largo ) );
}

//-----

public String getLeyendaAncho () {
    return jlblLeyendaAncho.getText();
}

//-----

public void setLeyendaAncho ( String leyenda_ancho ) {
    jlblLeyendaAncho.setText ( leyenda_ancho );
}

//-----

public float getAncho () {
    return prismaRectangular.getAncho ();
}

//-----

public void setAncho ( float ancho ) {
    prismaRectangular.setAncho ( ancho );
    jlblValorAncho.setText ( Float.toString ( ancho ) );
}

//-----

public String getLeyendaAltura () {
    return jlblLeyendaAltura.getText();
}

//-----

public void setLeyendaAltura ( String leyenda_altura ) {
    jlblLeyendaAltura.setText ( leyenda_altura );
}

//-----

public float getAltura () {
    return prismaRectangular.getAltura ();
}

//-----

public void setAltura ( float altura ) {
```

```

        prismaRectangular.setAltura ( altura );
        lblValorAltura.setText ( Float.toString ( altura ) );
    }

    //-----

    public Icon getPrismaRectangular () {
        return lblImagenPrismaRectangular.getIcon ();
    }

    //-----

    public void setImagenPrismaRectangular ( Icon prismaRectangular ) {
        lblImagenPrismaRectangular.setIcon ( prismaRectangular );
    }

    //-----

    public boolean isImagenPrismaRectangularAjustado () {
        return bolImagenPrismaRectangularAjustado;
    }

    //-----

    public void setImagenPrismaRectangularAjustado ( boolean valor ) {
        bolImagenPrismaRectangularAjustado = valor;

        if ( bolImagenPrismaRectangularAjustado == true ) {
            // No funciona correctamente lo que nos proporcionó para ajustar las imágenes
            lblImagenPrismaRectangular.setIcon (
                Imagenes.escalarImagen (
                    lblImagenPrismaRectangular.getIcon (),
                    lblImagenPrismaRectangular.getWidth (),
                    lblImagenPrismaRectangular.getHeight () )
            );
        }
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    private void jmnAcercaDePrismaRectangularActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog (
            this,
            "JPrismaRectangular v1.0 \n\n" +
            "Por: \n" +
            "Jose Misael Adame Sandoval (18131209) \n\n" +
            "Semestre: Ene - Jun/2020",
            "Acerca de...",
            JOptionPane.INFORMATION_MESSAGE
        );
    }

    //-----

    private void jmnValoresPrismaRectangularActionPerformed(java.awt.event.ActionEvent evt) {
        // JDialogCilindro.ejecutar();

        JPrismaRectangularValores = new JDialogPrismaRectangular ( this, true );
        JPrismaRectangularValores.setVisible ( true );
    }

    //-----

    public void update () {
        JPrismaRectangularEvent evento = new JPrismaRectangularEvent ( prismaRectangular.getLargo (),
                                                                           prismaRectangular.getAncho(),
                                                                           prismaRectangular.getAltura (),

```

```

JPrismaRectangularValores.anteriorLargo,
JPrismaRectangularValores.anteriorAncho,
JPrismaRectangularValores.anteriorAltura );

        fireJPrismaRectangularEvent ( evento );

    }

//-----

public void addJPrismaRectangularListener ( JValoresPrismasListener listener ) {
    listeners.add ( listener );
}

//-----

public void removeJPrismaRectangularListener ( JValoresPrismasListener listener ) {
    listeners.remove ( listener );
}

//-----

public void fireJPrismaRectangularEvent ( JPrismaRectangularEvent evento ) {
    for ( JValoresPrismasListener listener : listeners ) {
        listener.valoresPrismaRectangular ( evento );
    }
}

@Override
public float areaBase() {
    return prismaRectangular.areaBase ();
}

@Override
public float areaLateral() {
    return prismaRectangular.areaLateral ();
}

@Override
public float areaTotal() {
    return prismaRectangular.areaTotal ();
}

@Override
public float volumen() {
    return prismaRectangular.volumen ();
}
}

```

## JPrismaTriangular.java

```

/*-----
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Bean GUI de un JPrismaTriangular
*:
*: Archivo      : JPrismaRectangular.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 01/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Bean que muestra información sobre el cateto A, Cateto B y altura
*:                de un Prisma Triangular, junto con su imagen.
*:
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:=====

```

```
/* 06/Abr/2020 Misael Adame          Agregar prólogo.
: *-----*/

package mx.edu.itl.beans;

import java.util.ArrayList;
import javax.swing.Icon;
import javax.swing.JOptionPane;
import mx.edu.itl.eventos.JPrismaTriangularEvent;
import mx.edu.itl.eventos.JValoresPrismasListener;
import mx.edu.itl.prismas.PrismaTriangular;
import mx.edu.itl.util.Imagenes;

public class JPrismaTriangular extends javax.swing.JPanel implements IPrisma {

    //-----

    private PrismaTriangular prismaTriangular = new PrismaTriangular ();
    public JDialogPrismaTriangular JPrismaTriangularValores;
    private boolean bolImagenPrismaTriangularAjustado = false;
    private ArrayList<JValoresPrismasListener> listeners = new ArrayList ();

    //-----

    public JPrismaTriangular() {
        initComponents();

        prismaTriangular = new PrismaTriangular();
    }

    //-----

    public JPrismaTriangular ( float cateto_a, float cateto_b , float altura ) {
        prismaTriangular = new PrismaTriangular ( cateto_a, cateto_b, altura );
    }

    //-----

    public String getCatetoA () {
        return lblLeyendaCatetoA.getText();
    }

    //-----

    public void setCatetoA ( String leyenda_cateto_a ) {
        lblLeyendaCatetoA.setText ( leyenda_cateto_a );
    }

    //-----

    public float getCateto_a() {
        return prismaTriangular.getCateto_a();
    }

    //-----

    public void setCateto_a ( float cateto_a ) {
        prismaTriangular.setCateto_a ( cateto_a );
        lblValorCatetoA.setText ( Float.toString ( cateto_a ) );
    }

    //-----

    public String getCatetoB () {
        return lblLeyendaCatetoB.getText();
    }

    //-----

    public void setCatetoB ( String leyenda_cateto_b ) {
        lblLeyendaCatetoB.setText ( leyenda_cateto_b );
    }

    //-----
}
```

```
public float getCateto_b () {
    return prismaTriangular.getCateto_b ();
}

//-----

public void setCateto_b ( float cateto_b ) {
    prismaTriangular.setCateto_b ( cateto_b );
    lblValorCatetoB.setText ( Float.toString ( cateto_b ) );
}

//-----

public String getLeyendaAltura () {
    return lblLeyendaAltura.getText();
}

//-----

public void setLeyendaAltura ( String leyenda_altura ) {
    lblLeyendaAltura.setText ( leyenda_altura );
}

//-----

public float getAltura () {
    return prismaTriangular.getAltura ();
}

//-----

public void setAltura ( float altura ) {
    prismaTriangular.setAltura ( altura );
    lblValorAltura.setText ( Float.toString ( altura ) );
}

//-----

public Icon getPrismaTriangular () {
    return lblImagenPrismaTriangular.getIcon ();
}

//-----

public void setPrismaTriangular ( Icon prismaTriangular ) {
    lblImagenPrismaTriangular.setIcon ( prismaTriangular );
}

//-----

public boolean isImagenPrismaTriangularAjustado () {
    return bolImagenPrismaTriangularAjustado;
}

//-----

public void setImagenPrismaTriangularAjustado ( boolean valor ) {
    bolImagenPrismaTriangularAjustado = valor;

    if ( bolImagenPrismaTriangularAjustado == true ) {
        // No funciona correctamente lo que nos proporcionó para ajustar las imágenes
        lblImagenPrismaTriangular.setIcon ( Imagenes.escalarImagen
            ( lblImagenPrismaTriangular.getIcon (),
              lblImagenPrismaTriangular.getWidth (),
              lblImagenPrismaTriangular.getHeight () ) );
    }
}

//-----
@SuppressWarnings("unchecked")
Generated Code

//-----
```

```

private void jmniAcercaDePrismaTriangularActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog (
        this,
        "JPismaRectangular v1.0 \n\n" +
        "Por: \n" +
        "Jose Misael Adame Sandoval (18131209) \n\n" +
        "Semestre: Ene - Jun/2020",
        "Acerca de...",
        JOptionPane.INFORMATION_MESSAGE
    );
}

//-----

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

    // JDialogCilindro.ejecutar();

    JPismaTriangularValores = new JDialogPrismaTriangular ( this, true );
    JPismaTriangularValores.setVisible ( true );
}

//-----

public void update () {

    JPismaTriangularEvent evento = new JPismaTriangularEvent ( prismaTriangular.getCateto_a (),
                                                                prismaTriangular.getCateto_b(),
                                                                prismaTriangular.getAltura (),

JPismaTriangularValores.anteriorCatetoA,
JPismaTriangularValores.anteriorCatetoB,
JPismaTriangularValores.anteriorAltura );

    fireJPismaTriangularEvent ( evento );

}

//-----

public void addJPismaTriangularListener ( JValoresPrismasListener listener ) {
    listeners.add ( listener );
}

public void removeJPismaTriangularListener ( JValoresPrismasListener listener ) {
    listeners.remove ( listener );
}

//-----

public void fireJPismaTriangularEvent ( JPismaTriangularEvent evento ) {
    for ( JValoresPrismasListener listener : listeners ) {
        listener.valoresPrismaTriangular ( evento );
    }
}

//-----

@Override
public float areaBase() {
    return prismaTriangular.areaBase ();
}

//-----

@Override
public float areaLateral() {
    return prismaTriangular.areaLateral ();
}

//-----

@Override

```

```

public float areaTotal() {
    return prismaTriangular.areaTotal ();
}

//-----

@Override
public float volumen() {
    return prismaTriangular.volumen ();
}
}

```

## JCilindroEvent.java

```

/*-----
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020    HORA: 17-18 HRS
*:
*:          Clase del evento del JCilindro
*:
*: Archivo      : JCilindroEvent.java
*: Autor        : José Misael Adame Sandoval    18131209
*: Fecha        : 06/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Clase en donde se llama al evento desde el JCilindro
*:                en donde se realizan modificaciones al mismo Bean
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 06/Abr/2020 Misael Adame    Agregar prólogo.
*:-----*/

package mx.edu.itl.eventos;

public class JCilindroEvent {

    //-----

    private float nuevoRadio, nuevaAltura, anteriorRadio, anteriorAltura;

    //-----

    public JCilindroEvent () {
        this.nuevoRadio = 0;
        this.nuevaAltura = 0;
        this.anteriorRadio = 0;
        this.anteriorAltura = 0;
    }

    //-----

    public JCilindroEvent ( float nuevoRadio, float nuevaAltura, float anteriorRadio,
                           float anteriorAltura ) {
        this.nuevoRadio = nuevoRadio;
        this.nuevaAltura = nuevaAltura;
        this.anteriorRadio = anteriorRadio;
        this.anteriorAltura = anteriorAltura;
    }

    //-----

    public float getNuevoRadio () {
        return nuevoRadio;
    }

    //-----

    public void setNuevoRadio ( float nuevoRadio ) {
        this.nuevoRadio = nuevoRadio;
    }
}

```



```

}

//-----

public float getNuevaAltura () {
    return nuevaAltura;
}

//-----

public void setNuevaAltura ( float nuevaAltura ) {
    this.nuevaAltura = nuevaAltura;
}

//-----

public float getAnteriorRadio () {
    return anteriorRadio;
}

//-----

public void setAnteriorRadio ( float anteriorRadio ) {
    this.anteriorRadio = anteriorRadio;
}

//-----

public float getAnteriorAltura() {
    return anteriorAltura;
}

//-----

public void setAnteriorAltura(float anteriorAltura) {
    this.anteriorAltura = anteriorAltura;
}
}

```

### JPrismaRectangularEvent.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase del evento del JPrismaRectangular
*:
*: Archivo      : JPrismaRectangularEvent.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 06/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Clase en donde se llama al evento desde el JPrismaRectangular
*:               en donde se realizan modificaciones al mismo Bean
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.eventos;

public class JPrismaRectangularEvent {

    //-----

    private float nuevoLargo, nuevoAncho, nuevaAltura;
    private float anteriorLargo, anteriorAncho, anteriorAltura;

    //-----

```

```
public JPrismaRectangularEvent() {
    this.nuevoLargo = 0;
    this.nuevoAncho = 0;
    this.nuevaAltura = 0;
    this.anteriorLargo = 0;
    this.anteriorAncho = 0;
    this.anteriorAltura = 0;
}

//-----

public JPrismaRectangularEvent (float nuevoLargo, float nuevoAncho, float nuevaAltura,
                                float anteriorLargo, float anteriorAncho, float anteriorAltura) {
    this.nuevoLargo = nuevoLargo;
    this.nuevoAncho = nuevoAncho;
    this.nuevaAltura = nuevaAltura;
    this.anteriorLargo = anteriorLargo;
    this.anteriorAncho = anteriorAncho;
    this.anteriorAltura = anteriorAltura;
}

//-----

public float getNuevoLargo() {
    return nuevoLargo;
}

//-----

public void setNuevoLargo(float nuevoLargo) {
    this.nuevoLargo = nuevoLargo;
}

//-----

public float getNuevoAncho() {
    return nuevoAncho;
}

//-----

public void setNuevoAncho(float nuevoAncho) {
    this.nuevoAncho = nuevoAncho;
}

//-----

public float getNuevaAltura() {
    return nuevaAltura;
}

//-----

public void setNuevaAltura(float nuevaAltura) {
    this.nuevaAltura = nuevaAltura;
}

//-----

public float getAnteriorLargo() {
    return anteriorLargo;
}

//-----

public void setAnteriorLargo(float anteriorLargo) {
    this.anteriorLargo = anteriorLargo;
}

//-----

public float getAnteriorAncho() {
    return anteriorAncho;
}
}
```

```

//-----

public void setAnteriorAncho(float anteriorAncho) {
    this.anteriorAncho = anteriorAncho;
}

//-----

public float getAnteriorAltura() {
    return anteriorAltura;
}

//-----

public void setAnteriorAltura(float anteriorAltura) {
    this.anteriorAltura = anteriorAltura;
}
}

```

## JPrismaTriangularEvent.java

```

/*-----
:*                               INSTITUTO TECNOLÓGICO DE LA LAGUNA
:*                               INGENIERÍA EN SISTEMAS COMPUTACIONALES
:*                               TEMAS AVANZADOS DE PROGRAMACIÓN "B"
:*
:*                               SEMESTRE: ENE-JUN/2020    HORA: 17-18 HRS
:*
:*                               Clase del evento del JPrismaTriangular
:*
:* Archivo      : JPrismaTriangularEvent.java
:* Autor       : José Misael Adame Sandoval      18131209
:* Fecha      : 06/Abr/2020
:* Compilador  : JAVA J2SE v1.8.0
:* Descripción : Clase en donde se llama al evento desde el JPrismaTriangular
:*              en donde se realizan modificaciones al mismo Bean
:*
:*
:* Última modif:
:* Fecha      Modificó      Motivo
:*=====
:* 06/Abr/2020 Misael Adame    Agregar prólogo.
:*-----*/

package mx.edu.itl.eventos;

public class JPrismaTriangularEvent {

    //-----

    private float nuevoCatetoA, nuevoCatetoB, nuevaAltura;
    private float anteriorCatetoA, anteriorCatetoB, anteriorAltura;

    //-----

    public JPrismaTriangularEvent() {
        this.nuevoCatetoA = 0;
        this.nuevoCatetoB = 0;
        this.nuevaAltura = 0;
        this.anteriorCatetoA = 0;
        this.anteriorCatetoB = 0;
        this.anteriorAltura = 0;
    }

    //-----

    public JPrismaTriangularEvent(float nuevoCatetoA, float nuevoCatetoB, float nuevaAltura,
                                   float anteriorCatetoA, float anteriorCatetoB, float anteriorAltura) {
        this.nuevoCatetoA = nuevoCatetoA;
        this.nuevoCatetoB = nuevoCatetoB;
        this.nuevaAltura = nuevaAltura;
        this.anteriorCatetoA = anteriorCatetoA;
        this.anteriorCatetoB = anteriorCatetoB;
    }
}

```

```
        this.anteriorAltura = anteriorAltura;
    }

    //-----

    public float getNuevoCatetoA() {
        return nuevoCatetoA;
    }

    //-----

    public void setNuevoCatetoA(float nuevoCatetoA) {
        this.nuevoCatetoA = nuevoCatetoA;
    }

    //-----

    public float getNuevoCatetoB() {
        return nuevoCatetoB;
    }

    //-----

    public void setNuevoCatetoB(float nuevoCatetoB) {
        this.nuevoCatetoB = nuevoCatetoB;
    }

    //-----

    public float getNuevaAltura() {
        return nuevaAltura;
    }

    //-----

    public void setNuevaAltura(float nuevaAltura) {
        this.nuevaAltura = nuevaAltura;
    }

    //-----

    public float getAnteriorCatetoA() {
        return anteriorCatetoA;
    }

    //-----

    public void setAnteriorCatetoA(float anteriorCatetoA) {
        this.anteriorCatetoA = anteriorCatetoA;
    }

    //-----

    public float getAnteriorCatetoB() {
        return anteriorCatetoB;
    }

    //-----

    public void setAnteriorCatetoB(float anteriorCatetoB) {
        this.anteriorCatetoB = anteriorCatetoB;
    }

    //-----

    public float getAnteriorAltura() {
        return anteriorAltura;
    }

    //-----

    public void setAnteriorAltura(float anteriorAltura) {
        this.anteriorAltura = anteriorAltura;
    }
}
```

**JValoresPrismasListener.java**

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TÓPICOS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase Interface del JValoresPrismasListener
*:
*: Archivo      : JValoresPrismasListener.java
*: Autor       : José Misael Adame Sandoval      18131209
*: Fecha       : 06/Abr/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Clase Interface que es en donde se basarán los eventos
*:              de los prismas (Listener)
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package mx.edu.itl.eventos;

public interface JValoresPrismasListener {

    public void valoresCilindro( JCilindroEvent evento );

    //-----

    public void valoresPrismaRectangular( JPrismaRectangularEvent evento );

    //-----

    public void valoresPrismaTriangular ( JPrismaTriangularEvent evento );

    //-----
}

```

**AcercaDeDialog.java**

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TÓPICOS 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       : 06/Abr/2020
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Aplicación que muestra información sobre el autor de la
*:              la aplicación PrismasApp con Beans (es un Bean implementado)
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 06/Abr/2020 Misael Adame      Agregar prólogo.
*:-----*/

package prismas;

public class AcercaDeDialog extends javax.swing.JDialog {

    //-----
}

```

```

public AcercaDeDialog ( java.awt.Frame parent, boolean modal ) {
    super ( parent, modal );
    initComponents ();
}

//-----

@SuppressWarnings("unchecked")
Generated Code

//-----

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

//-----

public static void ejecutar () {
    /* Set the Windows look and feel */
    Look and feel setting code (optional)

    /* Create and display the dialog */
    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 );
        }
    });
}
}

```

## 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 los Beans
*:
*: Archivo      : PrismasFrame.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 3/Abr/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación que accede a las Beans JCilindro, JPrismaRectangular
*:                y JPrismaTriangular e invoca a cada uno de los métodos y
*:                funcionalidades para realizar los cálculos correspondientes
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:-----
*: 6/Abr/2020 Misael Adame      Agregar prólogo
*:-----*/

package prismas;

import javax.swing.JOptionPane;
import mx.edu.itl.eventos.JCilindroEvent;
import mx.edu.itl.eventos.JPrismaRectangularEvent;
import mx.edu.itl.eventos.JPrismaTriangularEvent;
import mx.edu.itl.eventos.JValoresPrismasListener;

public class PrismasFrame extends javax.swing.JFrame implements JValoresPrismasListener {

```

```

//-----

private AcercaDeDialog acercaDeDialog;

//-----

public PrismasFrame () {
    initComponents ();

    jCilindro.addJCilindroListener ( this );
    jPrismaRectangular.addJPrismaRectangularListener ( this );
    jPrismaTriangular.addJPrismaTriangularListener ( this );

    jtxfCilindroRadio.setText ( jCilindro.getRadio () + " " );
    jtxfCilindroAltura.setText ( jCilindro.getAltura () + " " );

    jtxfPrismRectLargo.setText ( jPrismaRectangular.getLargo () + " " );
    jtxfPrismRectAncho.setText ( jPrismaRectangular.getAncho () + " " );
    jtxfPrismRectAltura.setText ( jPrismaRectangular.getAltura () + " " );

    jtxfPrismTriangCatetoA.setText ( jPrismaTriangular.getCateto_a () + " " );
    jtxfPrismTriangCatetoB.setText ( jPrismaTriangular.getCateto_b () + " " );
    jtxfPrismTriangAltura.setText ( jPrismaTriangular.getAltura () + " " );

}

//-----

@SuppressWarnings ( "unchecked" )
Generated Code

//-----

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

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

    jCilindro.setRadio ( 0 );
    jCilindro.setAltura ( 0 );

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

    jPrismaRectangular.setLargo ( 0 );
    jPrismaRectangular.setAncho ( 0 );
    jPrismaRectangular.setAltura ( 0 );

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

    jPrismaTriangular.setCateto_a ( 0 );
    jPrismaTriangular.setCateto_b ( 0 );
    jPrismaTriangular.setAltura ( 0 );

    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 jbtnPrismRectCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        float largo = Float.parseFloat ( jtxfPrismRectLargo.getText () );
        float ancho = Float.parseFloat ( jtxfPrismRectAncho.getText () );
        float altura = Float.parseFloat ( jtxfPrismRectAltura.getText () );
        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 ( jPrismaRectangular.areaBase () + " " );
            jlblPrismRectAreaLateral.setText ( jPrismaRectangular.areaLateral () + " " );
            jlblPrismRectAreaTotal.setText ( jPrismaRectangular.areaTotal () + " " );
            jlblPrismRectVolumen.setText ( jPrismaRectangular.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 () );
        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 ( jCilindro.areaBase () + " " );
            jlblCilindroAreaLateral.setText ( jCilindro.areaLateral () + " " );
            jlblCilindroAreaTotal.setText ( jCilindro.areaTotal () + " " );
            jlblCilindroVolumen.setText ( jCilindro.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 () );
    }
}

```



```

        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 ( jPrismaTriangular.areaBase () + " " );
            lblPrismTriangAreaLateral.setText ( jPrismaTriangular.areaLateral () + " " );
            lblPrismTriangAreaTotal.setText ( jPrismaTriangular.areaTotal () + " " );
            lblPrismTriangVolumen.setText ( jPrismaTriangular.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 ();
}

private void jtxfCilindroRadioKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float radio = Float.parseFloat ( jtxfCilindroRadio.getText () );
        jCilindro.setRadio ( radio );
    } catch ( NumberFormatException ex ) {
        jCilindro.setRadio ( 0 );
    }
}

//-----

private void jtxfCilindroAlturaKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float altura = Float.parseFloat ( jtxfCilindroAltura.getText () );
        jCilindro.setAltura ( altura );
    } catch ( NumberFormatException ex ) {
        jCilindro.setAltura ( 0 );
    }
}

//-----

private void jtxfPrismRectLargoKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float largo = Float.parseFloat ( jtxfPrismRectLargo.getText () );
        jPrismaRectangular.setLargo ( largo );
    } catch ( NumberFormatException ex ) {
        jPrismaRectangular.setLargo ( 0 );
    }
}

//-----

private void jtxfPrismRectAnchoKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float ancho = Float.parseFloat ( jtxfPrismRectAncho.getText () );
        jPrismaRectangular.setAncho ( ancho );
    } catch ( NumberFormatException ex ) {
        jPrismaRectangular.setAncho ( 0 );
    }
}

//-----

```

```

private void jtxfPrismRectAlturaKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float altura = Float.parseFloat (jtxfPrismRectAltura.getText() );
        jPrismaRectangular.setAltura ( altura );
    } catch ( NumberFormatException ex ) {
        jPrismaRectangular.setAltura ( 0 );
    }
}

//-----

private void jtxfPrismTriangCatetoBKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float cateto_b = Float.parseFloat ( jtxfPrismTriangCatetoB.getText() );
        jPrismaTriangular.setCateto_b ( cateto_b );
    } catch ( NumberFormatException ex ) {
        jPrismaTriangular.setCateto_b ( 0 );
    }
}

//-----

private void jtxfPrismTriangAlturaKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float altura = Float.parseFloat ( jtxfPrismTriangAltura.getText() );
        jPrismaTriangular.setAltura(altura);
    } catch ( NumberFormatException ex ) {
        jPrismaTriangular.setAltura ( 0 );
    }
}

//-----

private void jtxfPrismTriangCatetoAKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        float cateto_a = Float.parseFloat ( jtxfPrismTriangCatetoA.getText() );
        jPrismaTriangular.setCateto_a ( cateto_a );
    } catch ( NumberFormatException ex ) {
        jPrismaTriangular.setCateto_a ( 0 );
    }
}

//-----

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

//-----

public static void ejecutar () {

    Look and feel stting code(optional)

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

//-----

@Override
public void valoresCilindro ( JCilindroEvent evento ) {
    float radio = evento.getNuevoRadio ();
    float altura = evento.getNuevaAltura ();
    jtxfCilindroRadio.setText ( Float.toString ( radio ) );
    jtxfCilindroAltura.setText ( Float.toString ( altura ) );
}

//-----

```

```

@Override
public void valoresPrismaRectangular(JPrismaRectangularEvent evento) {
    float largo = evento.getNuevoLargo ();
    float ancho = evento.getNuevoAncho ();
    float altura = evento.getNuevaAltura ();
    jtxfPrismRectLargo.setText ( Float.toString ( largo ) );
    jtxfPrismRectAncho.setText ( Float.toString ( ancho ) );
    jtxfPrismRectAltura.setText ( Float.toString ( altura ) );
}

//-----

@Override
public void valoresPrismaTriangular(JPrismaTriangularEvent evento) {
    float cateto_a = evento.getNuevoCatetoA ();
    float cateto_b = evento.getNuevoCatetoB ();
    float altura = evento.getNuevaAltura ();
    jtxfPrismTriangCatetoA.setText ( Float.toString ( cateto_a ) );
    jtxfPrismTriangCatetoB.setText ( Float.toString ( cateto_b ) );
    jtxfPrismTriangAltura.setText ( Float.toString ( altura ) );
}
}

```

## SplashPrismasFrame.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 : SplashPrismasFrame.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 SplashPrismasFrame extends javax.swing.JFrame implements Runnable {

    //-----

    private Thread tiempo = null;

    //-----

    public SplashPrismasFrame() {
        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() {

    /* Set the Windows look and feel */
    Look and feel setting code (optional)

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new SplashPrismasFrame().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(400);
                    new PrismasFrame().setVisible(true);

                }

            }
            break;
        } catch (InterruptedException ex) {
            Logger.getLogger(SplashPrismasFrame.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 mx.edu.itl.prismas.Cilindro;
import mx.edu.itl.prismas.PrismaRectangular;
import mx.edu.itl.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 );

}

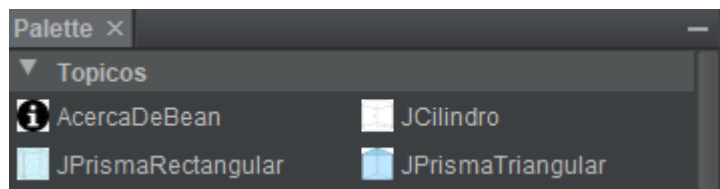
//-----

@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 );

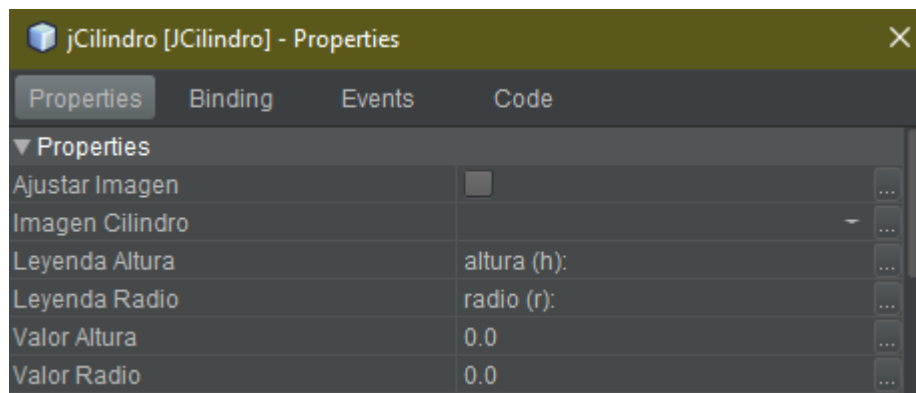
    // Prueba con dimensiones
    pt = new PrismaTriangular( 5.5f, 4.5f, 6.2f );
    assertEquals ( "Prisma Triangular Area de Base", 12.375, pt.areaBase(), 0.001 );
    assertEquals ( "Prisma Triangular Area Lateral", 106.059, pt.areaLateral(), 0.001 );
    assertEquals ( "Prisma Triangular Area Total",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

## Prueba de Ejecución

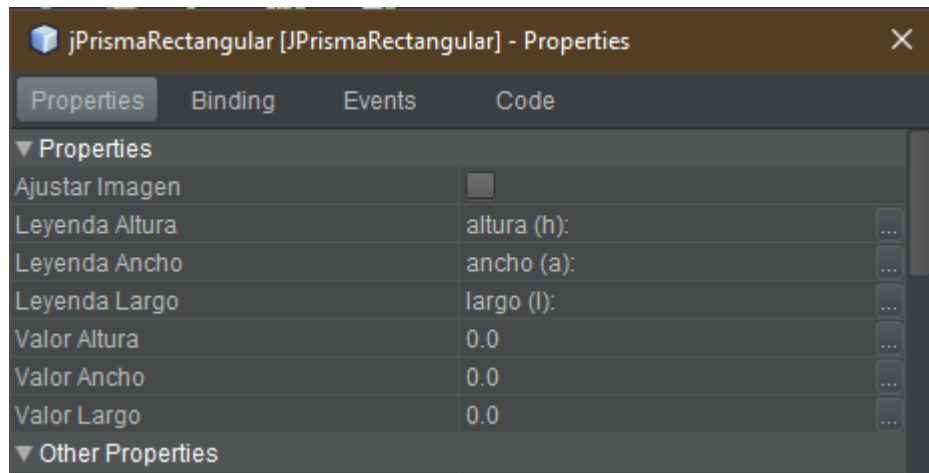
Paleta de NetBeans con los 4 beans diseñados



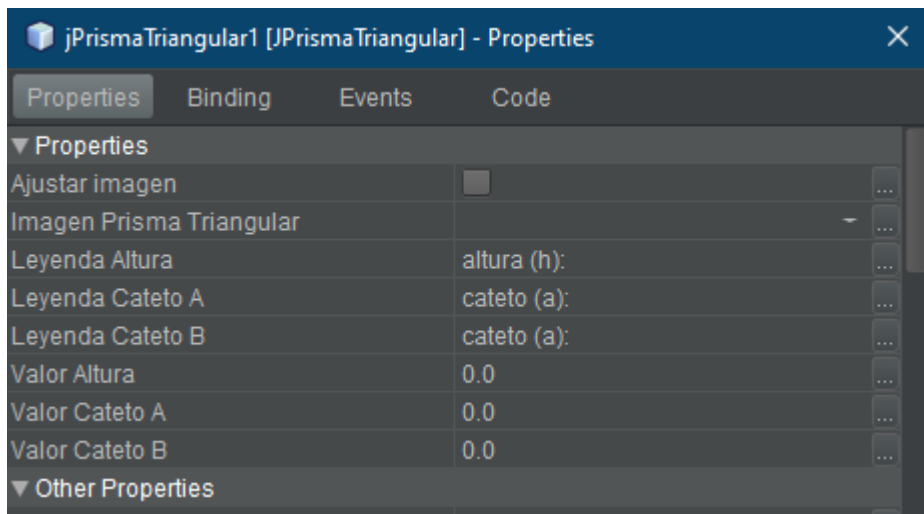
Propiedades del bean JCilindro



Propiedades del bean JPrismaRectangular



Propiedades del bean JPrismaTriangular



Este es el Splash al ejecutar la aplicación



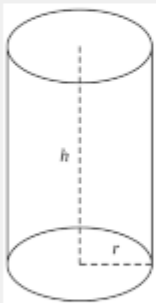


Esta es la interfaz inicial

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



radio (r):

altura (h):

Datos

Radio (r) :

Altura (h) :

Calcular

Resultados

Área de la base:

Área lateral:

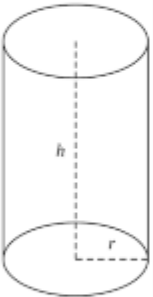
Área Total:

Volumen:

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



radio (r):

altura (h):

Datos

Radio (r) :

Altura (h) :

Calcular

Resultados

Área de la base:

Área lateral:

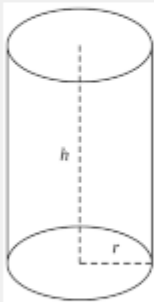
Área Total:

Volumen:

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



radio (r): 9.0

altura (h): 4.0

Datos

Radio (r) : 9

Altura (h) : 4

Calcular

Resultados

Área de la base: 254.46901

Área lateral: 226.19467

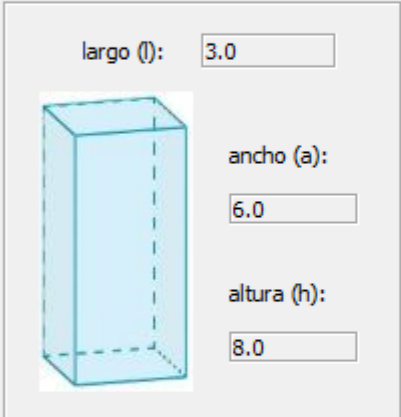
Área Total: 735.1327

Volumen: 1017.87604

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



largo (l): 3.0

ancho (a): 6.0

altura (h): 8.0

Datos

Largo (l) : 3

Ancho (a) : 6

Altura (h) : 8

Calcular

Resultados

Área de la base:

Área lateral:

Área Total:

Volumen:

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular

**Datos**

largo (l):

ancho (a):

altura (h):

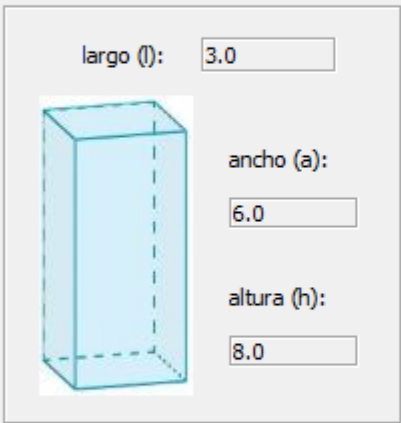
**Resultados**

Área de la base:

Área lateral:

Área Total:

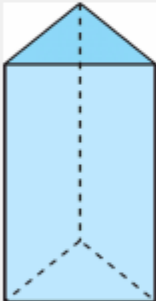
Volumen:



Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular



cateto (a):

cateto (b):

altura (h):

Datos

Cateto (a) :

Cateto (b) :

Altura (h) :

Resultados

Área de la base:

Área lateral:

Área Total:

Volumen:

Prismas App

Archivo Edición Ayuda

Cilindro Prisma Rectangular Prisma Triangular

**Datos**

Cateto (a) : 6

Cateto (b) : 3

Altura (h) : 4

Calcular

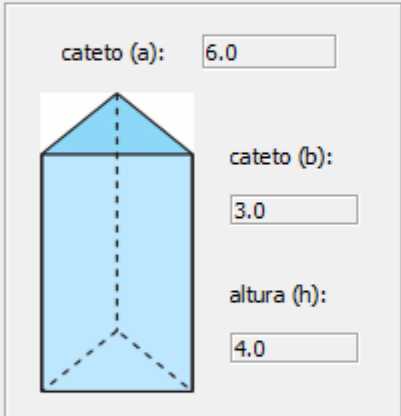
**Resultados**

Área de la base: 9.0

Área lateral: 48.0

Área Total: 66.0

Volumen: 36.0



Al hacer clic Derecho

radio (r) : 9.0

Altura (h) : 4

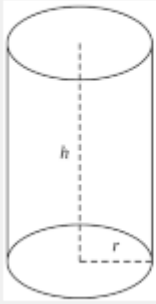
altura (h) : 4.0

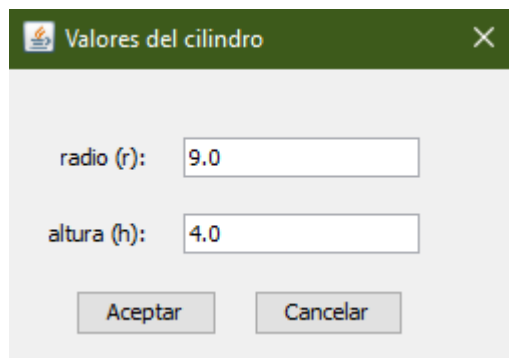
Valores del prisma...  
Acerca de...

**Resultados**

Área de la base

Área lateral:





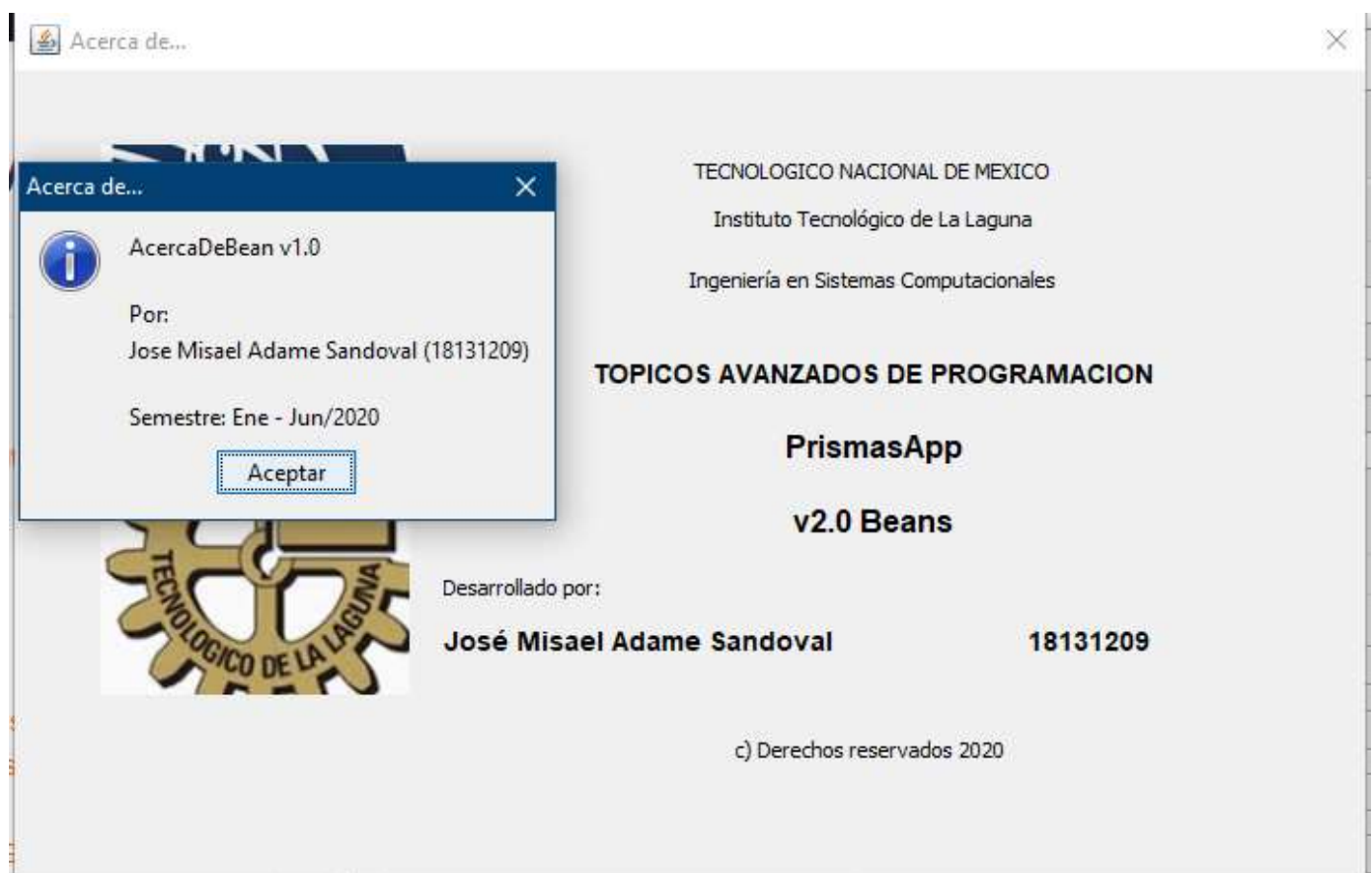
Valores del cilindro

radio (r): 9.0

altura (h): 4.0

Aceptar Cancelar

Acerca De Prismas





Clic derecho en “Acerca de...”

