



INSTITUTO POLITECNICO NACIONAL

UNIDAD PROFESIONAL

**UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERIA Y
CIENCIAS SOCIALES Y ADMINISTRATIVAS**

INGENIERIA EN INFORMATICA

Programación Orientada a Objetos

SECUENCIA: 2NM42

Equipo 07

Porcentaje de Participación

De
Santiago Landeros Edwin Uriel

Reveles Ramírez Jesús Emilio

Villegas Nolasco Elide Yolotzin

Herrera Cano Edwin Rubén

Palmieri Mondragón Gerardo

PROFESOR. Oviedo Galdeano Mario

Primer Departamental: Proyecto Convertidor de Unidades



Fecha de entrega:

Tabla de contenidos

Planteamiento de problemas.....
Descripción de solución.....
Diagrama de clases.....
Diagramas de casos de uso.....
Pseudocódigo de métodos.....
Listado de módulos.....
Imagen del navegador de proyectos.....
Copias de las ventanas.....
Comentarios finales.....

Planteamiento del problema

Se desea realizar una aplicación en el lenguaje de programación Java que permita la conversión de valores de una unidad a su equivalente en otra. Las unidades de medida que se podrán convertir son: Longitud, Volumen, Superficie, Peso, Temperatura y Velocidad, dependiendo del dato ingresado por el usuario. Este problema deberá contar con una interfaz gráfica con la que trabajará el Usuario (GUI).

La aplicación permitirá las siguientes conversiones:

Longitud:	Superficie:	Volumen:	Peso:	Velocidad:	Temperatura:
• kilómetros	• kilómetros cuadrados	• metro cúbico	• Tonelada	• Km/s	• Celsius
• metros	• hectáreas	• litro	• kilogramo	• M/s	• Fahrenheit
• centímetros	• pulgadas cuadradas	• centímetro cúbico	• gramo	• Km/hr	• kelvin
• milímetros	• metros cuadrados	• mililitro	• tonelada corta	• Milla/s	
• millas	• millas cuadradas	• barril	• libra	• Milla/hr	
• yardas	• acres	• galón	• onza	• pies/seg	
• pies	• pies cuadrados	• pinta		• Nudo	
• pulgadas		• onza líquida			

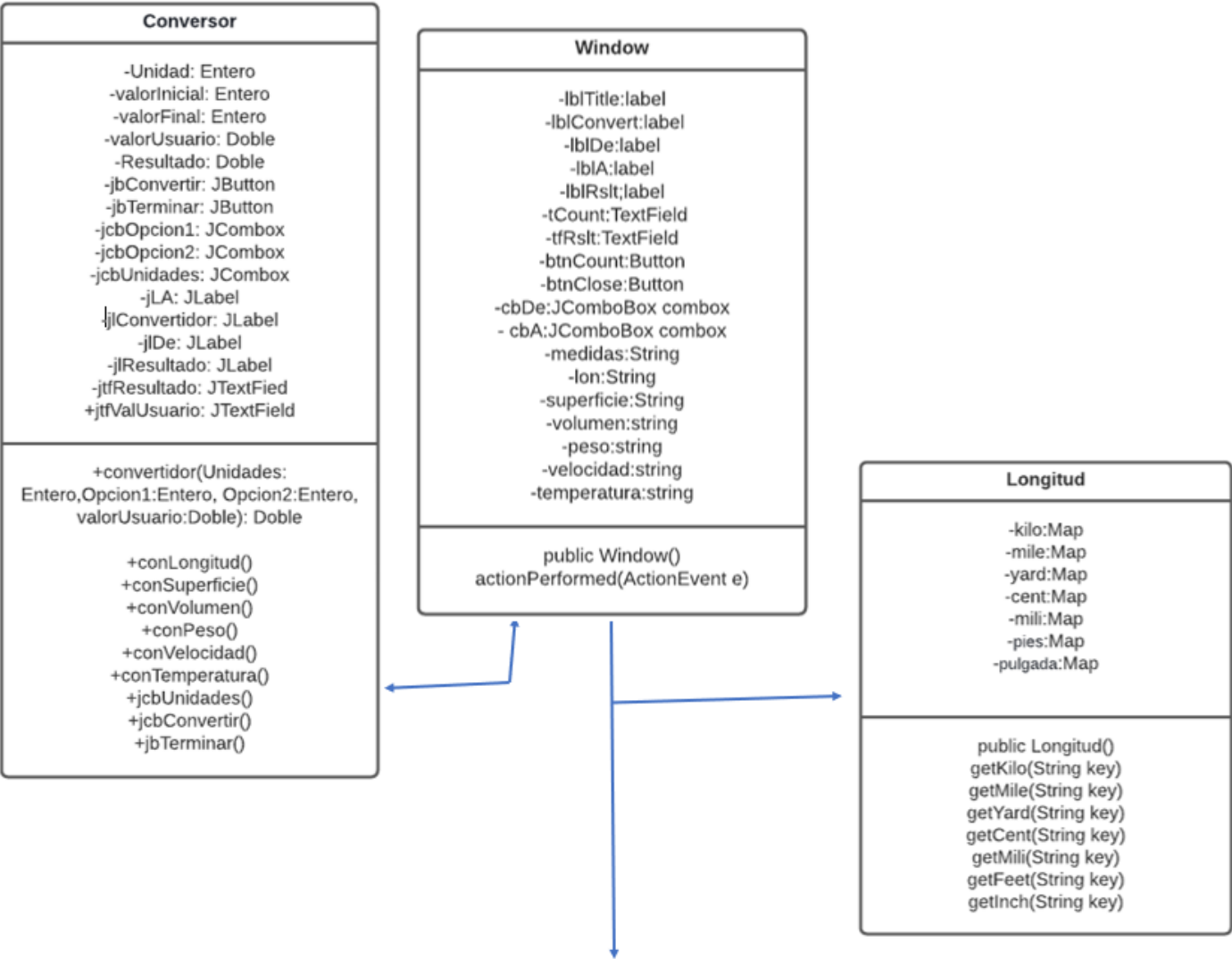
Descripción de solución

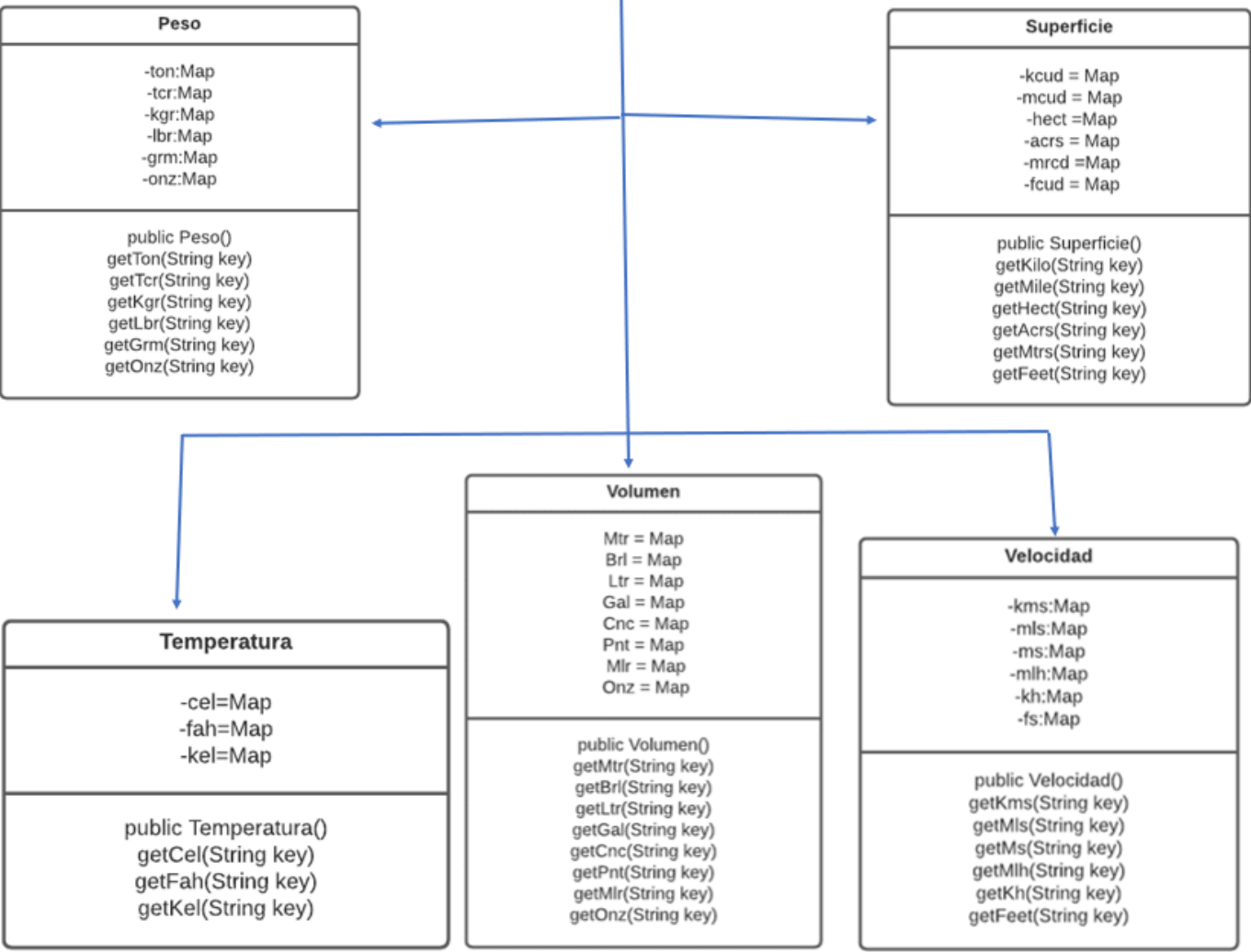
Lo principal será identificar las clases con las que vamos a trabajar para poder abordar el problema y con ello trabajar en la aplicación para poder construir el sistema. Estas clases, a su vez, tendrán las herramientas gráficas (GUI) para que se pueda interactuar con la aplicación.

- Clase Portada: Contendrá los nombres de los integrantes del equipo, así como el número de este, además tendrá un botón para continuar a la ventana de las conversiones.
- Clase Principal: Ejecutará el Main y también tendrá la capacidad de cerrar el programa cuando se requiera.
- Clase Conversor: Donde tendremos los campos con los que trabajaremos el sistema, estos campos podrían ocupar la selección de las unidades (dependiendo el tipo de unidad). También deberá permitir introducir al usuario los datos con los que va a trabajar.

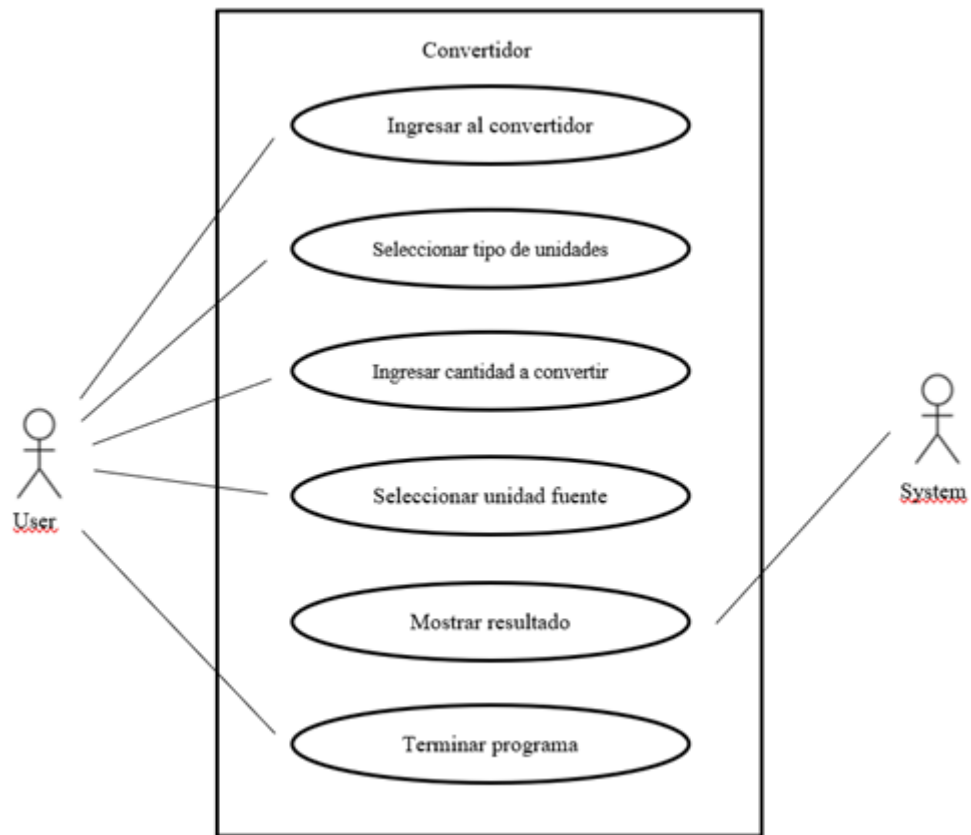
Por ejemplo, si quisiera convertir 1 km a Metros, seleccionaría el tipo de dato, introduciría el valor de 1, seleccionaría el dato deseado (en este caso metros) e iniciaría la conversión. En este campo tendremos las operaciones matemáticas y lógicas que nos van a permitir realizar las conversiones. Esta clase contendrá los tipos de datos que se trabajaran, en relación con los datos introducidos por el usuario podrá realizar los cálculos matemáticos necesarios para la conversión.

Diagramas de Clase





Diagramas de casos de uso



Pseudocódigo de métodos

Listado de modulos

Modulo Window.java

```
11 public class Window extends JFrame implements ActionListener{
12     private Label lblTitle, lblConvert, lblDe, lblA, lblRslt; // Declare a Label component
13     private TextField tfCount, tfRslt; // Declare a TextField component
14     private Button btnCount, btnClose; // Declare a Button component
15     private JComboBox combobox, cbDe, cbA;
16     private String[] medidas = {"Longitud", "Superficie", "Volumen", "Peso", "Velocidad", "Temperatura"},
17         lon = {"Kilometros", "Millas", "Yardas", "Centimetros", "Milimetros", "Pies", "Pulgadas"},
18         superficie = {"K. cuadrados", "M. cuadradas", "Hectareas", "Acres", "Mtr. cuadrados", "Pies cuadrados"},
19         volumen = {"Metro cubico", "Barril", "Litro", "Galon", "C. cubico", "Finta", "Mililitro", "Onza liquida"},
20         peso = {"Tonelada", "T. corta", "Kilogramo", "Libra", "Gramo", "Onza"},
21         velocidad = {"Km/s", "Milla/s", "m/s", "Milla/hr", "Km/hr", "Pie/s"},
22         temperatura = {"Celsius", "Fahrenheit", "Kelvin"};
23 }
```

Evento JComboBox()

```
32
33     combobox = new JComboBox(medidas);
34     combobox.setBounds(200, 50, 200, 30);
35     add(combobox);
36     //combobox.addItemListener(this);
37     combobox.addActionListener(this);
38
```


Metodo public actionPerformed

```
107
108
109     for (String a : lon) {
110         if (cbDe.getSelectedItem() == "Kilometros" && cbA.getSelectedItem() == a) {
111             rslt = unit * unitL.getKilo(a);
112         }
113         if (cbDe.getSelectedItem() == "Millas" && cbA.getSelectedItem() == a) {
114             rslt = unit * unitL.getMile(a);
115         }
116         if (cbDe.getSelectedItem() == "Yardas" && cbA.getSelectedItem() == a) {
117             rslt = unit * unitL.getYard(a);
118         }
119         if (cbDe.getSelectedItem() == "Centimetros" && cbA.getSelectedItem() == a) {
120             rslt = unit * unitL.getCent(a);
121         }
122         if (cbDe.getSelectedItem() == "Milimetros" && cbA.getSelectedItem() == a) {
123             rslt = unit * unitL.getMili(a);
124         }
125         if (cbDe.getSelectedItem() == "Pies" && cbA.getSelectedItem() == a) {
126             rslt = unit * unitL.getFeet(a);
127         }
128         if (cbDe.getSelectedItem() == "Pulgadas" && cbA.getSelectedItem() == a) {
129             rslt = unit * unitL.getInch(a);
130         }
131     }
132
133     for (String a : superficie) {
134         if (cbDe.getSelectedItem() == "K. cuadrados" && cbA.getSelectedItem() == a) {
135             rslt = unit * unitS.getKilo(a);
136         }
137         if (cbDe.getSelectedItem() == "M. cuadradas" && cbA.getSelectedItem() == a) {
138             rslt = unit * unitS.getMile(a);
139         }
140     }
141
142     if (cbDe.getSelectedItem() == "M. cuadradas" && cbA.getSelectedItem() == a) {
143         rslt = unit * unitS.getMile(a);
144     }
145     if (cbDe.getSelectedItem() == "Hectareas" && cbA.getSelectedItem() == a) {
146         rslt = unit * unitS.getHect(a);
147     }
148     if (cbDe.getSelectedItem() == "Acres" && cbA.getSelectedItem() == a) {
149         rslt = unit * unitS.getAcres(a);
150     }
151     if (cbDe.getSelectedItem() == "Mtr. cuadrados" && cbA.getSelectedItem() == a) {
152         rslt = unit * unitS.getMtrs(a);
153     }
154     if (cbDe.getSelectedItem() == "Pies cuadrados" && cbA.getSelectedItem() == a) {
155         rslt = unit * unitS.getFeet(a);
156     }
157 }
158
159 for (String a : velocidad) {
160     if (cbDe.getSelectedItem() == "Km/s" && cbA.getSelectedItem() == a) {
161         rslt = unit * unitV.getKms(a);
162     }
163     if (cbDe.getSelectedItem() == "Milla/s" && cbA.getSelectedItem() == a) {
164         rslt = unit * unitV.getMls(a);
165     }
166     if (cbDe.getSelectedItem() == "m/s" && cbA.getSelectedItem() == a) {
167         rslt = unit * unitV.getMs(a);
168     }
169     if (cbDe.getSelectedItem() == "Milla/hr" && cbA.getSelectedItem() == a) {
170         rslt = unit * unitV.getMlh(a);
171     }
172 }
```

```

168     }
169     if (cbDe.getSelectedItem() == "Pie/s" && cbA.getSelectedItem() == a) {
170         rslt = unit * unitV.getFeet(a);
171     }
172 }
173
174 for (String a : volumen) {
175     if (cbDe.getSelectedItem() == "Metro cubico" && cbA.getSelectedItem() == a) {
176         rslt = unit * unitVol.getMtr(a);
177     }
178     if (cbDe.getSelectedItem() == "Barril" && cbA.getSelectedItem() == a) {
179         rslt = unit * unitVol.getBrl(a);
180     }
181     if (cbDe.getSelectedItem() == "Litro" && cbA.getSelectedItem() == a) {
182         rslt = unit * unitVol.getLtr(a);
183     }
184     if (cbDe.getSelectedItem() == "Galon" && cbA.getSelectedItem() == a) {
185         rslt = unit * unitVol.getGal(a);
186     }
187     if (cbDe.getSelectedItem() == "C. cubico" && cbA.getSelectedItem() == a) {
188         rslt = unit * unitVol.getCnc(a);
189     }
190     if (cbDe.getSelectedItem() == "Pinta" && cbA.getSelectedItem() == a) {
191         rslt = unit * unitVol.getPnt(a);
192     }
193     if (cbDe.getSelectedItem() == "Mililitro" && cbA.getSelectedItem() == a) {
194         rslt = unit * unitVol.getMlr(a);
195     }
196     if (cbDe.getSelectedItem() == "Onza liquida" && cbA.getSelectedItem() == a) {
197         rslt = unit * unitVol.getOnz(a);
198     }
199 }

```

```

198     }
199 }
200
201 for (String a : peso) {
202     if (cbDe.getSelectedItem() == "Tonelada" && cbA.getSelectedItem() == a) {
203         rslt = unit * unitP.getTon(a);
204     }
205     if (cbDe.getSelectedItem() == "T. corta" && cbA.getSelectedItem() == a) {
206         rslt = unit * unitP.getTcr(a);
207     }
208     if (cbDe.getSelectedItem() == "Kilogramo" && cbA.getSelectedItem() == a) {
209         rslt = unit * unitP.getKgr(a);
210     }
211     if (cbDe.getSelectedItem() == "Libra" && cbA.getSelectedItem() == a) {
212         rslt = unit * unitP.getLbr(a);
213     }
214     if (cbDe.getSelectedItem() == "Gramo" && cbA.getSelectedItem() == a) {
215         rslt = unit * unitP.getGrm(a);
216     }
217     if (cbDe.getSelectedItem() == "Onza" && cbA.getSelectedItem() == a) {
218         rslt = unit * unitP.getOnz(a);
219     }
220 }
221
222 for (String a : temperatura) {
223     if (cbDe.getSelectedItem() == "Celsius" && cbA.getSelectedItem() == a) {
224         rslt = unit * unitT.getCel(a);
225     }
226     if (cbDe.getSelectedItem() == "Fahrenheit" && cbA.getSelectedItem() == a) {
227         rslt = unit * unitT.getFah(a);
228     }
229 }

```

```

228     }
229     if (cbDe.getSelectedItem() == "Kelvin" && cbA.getSelectedItem() == a) {
230         rslt = unit * unitT.getKel(a);
231     }
232 }
233
234 tfRslt.setText(Double.toString(rslt));
235 }
236
237 if(e.getSource() == combox){
238     if(combox.getSelectedItem() == "Longitud"){
239         cbDe.removeAllItems();
240         cbA.removeAllItems();
241         for(String item : lon){
242             cbDe.addItem(item);
243             cbA.addItem(item);
244         }
245     }if(combox.getSelectedItem() == "Superficie"){
246         cbDe.removeAllItems();
247         cbA.removeAllItems();
248         for(String item : superficie){
249             cbDe.addItem(item);
250             cbA.addItem(item);
251         }
252     }
253     if(combox.getSelectedItem() == "Volumen"){
254         cbDe.removeAllItems();
255         cbA.removeAllItems();
256         for(String item : volumen){
257             cbDe.addItem(item);

```

Modulo Volumen.java

Metodo volumen()

```

25 public Volumen() {
26     Mtr.put("Metro cubico", 1.00);
27     Mtr.put("Barril", 6.11);
28     Mtr.put("Litro", 1000.0);
29     Mtr.put("Galon", 227.02);
30     Mtr.put("C. cubico", 1000000.0);
31     Mtr.put("Pinta", 1816.17);
32     Mtr.put("Mililitro", 1000000.0);
33     Mtr.put("Onza liquida", 33814.02);
34
35     Brl.put("Metro cubico", 0.16);
36     Brl.put("Barril", 1.00);
37     Brl.put("Litro", 163.65);
38     Brl.put("Galon", 37.15);
39     Brl.put("C. cubico", 163.65);
40     Brl.put("Pinta", 297.22);
41     Brl.put("Mililitro", 163.65);
42     Brl.put("Onza liquida", 5759.66);
43
44     Ltr.put("Metro cubico", 0.001);
45     Ltr.put("Barril", 0.009);
46     Ltr.put("Litro", 1.00);
47     Ltr.put("Galon", 0.227);
48     Ltr.put("C. cubico", 1000.0);
49     Ltr.put("Pinta", 1.81);
50     Ltr.put("Mililitro", 1000.0);
51     Ltr.put("Onza liquida", 35.195);
52
53     Gal.put("Metro cubico", 0.005);
54     Gal.put("Barril", 0.039);
55     Gal.put("Litro", 4.54);

```

```

55     Gal.put("Litro", 4.54);
56     Gal.put("Galon", 1.00);
57     Gal.put("C. cubico", 4546.09);
58     Gal.put("Pinta", 8.25);
59     Gal.put("Mililitro", 4546.09);
60     Gal.put("Onza liquida", 160.00);
61
62     Cnc.put("Metro cubico", 0.000001);
63     Cnc.put("Barril", 0.000009);
64     Cnc.put("Litro", 0.001);
65     Cnc.put("Galon", 0.0002);
66     Cnc.put("C. cubico", 1.00);
67     Cnc.put("Pinta", 0.0018);
68     Cnc.put("Mililitro", 1.0);
69     Cnc.put("Onza liquida", 0.035);
70
71     Pnt.put("Metro cubico", 0.0005);
72     Pnt.put("Barril", 0.004);
73     Pnt.put("Litro", 0.56);
74     Pnt.put("Galon", 0.12);
75     Pnt.put("C. cubico", 568.26);
76     Pnt.put("Pinta", 1.00);
77     Pnt.put("Mililitro", 568.26);
78     Pnt.put("Onza liquida", 19.9);
79
80     Mlr.put("Metro cubico", 0.000001);
81     Mlr.put("Barril", 0.000009);
82     Mlr.put("Litro", 0.001);
83     Mlr.put("Galon", 0.0002);
84     Mlr.put("C. cubico", 1.0);
85     Mlr.put("Pinta", 0.001);

```

Modulo Velocidad.java

Método Velocidad()

```
19 public Velocidad() {
20     kms.put("Km/s", 1.00);
21     kms.put("Milla/s", 0.621);
22     kms.put("m/s", 1000.00);
23     kms.put("Milla/hr", 2236.936);
24     kms.put("Km/hr", 3600.0);
25     kms.put("Pie/s", 3280.84);
26
27     mls.put("Km/s", 1.609);
28     mls.put("Milla/s", 1.00);
29     mls.put("m/s", 1609.344);
30     mls.put("Milla/hr", 3600.0);
31     mls.put("Km/hr", 5793.63);
32     mls.put("Pie/s", 5280.00);
33
34     ms.put("Km/s", 0.001);
35     ms.put("Milla/s", 0.001);
36     ms.put("m/s", 1.00);
37     ms.put("Milla/hr", 2.237);
38     ms.put("Km/hr", 3.6);
39     ms.put("Pie/s", 3.281);
40
41     mlh.put("Km/s", 0.0004);
42     mlh.put("Milla/s", 0.0003);
43     mlh.put("m/s", 0.447);
44     mlh.put("Milla/hr", 1.00);
45     mlh.put("Km/hr", 1.60);
46     mlh.put("Pie/s", 1.46);
47
48     kh.put("Km/s", 0.0003);
49     kh.put("Milla/s", 0.0002);
```

Modulo Temperatura.java

Metodo Temperatura()

```
19
20 public Temperatura() {
21     cel.put("Celsius", 1.00);
22     cel.put("Fahrenheit", 33.8);
23     cel.put("Kelvin", 274.15);
24
25     fah.put("Celsius", -17.22);
26     fah.put("Fahrenheit", 1.00);
27     fah.put("Kelvin", 255.93);
28
29     kel.put("Celsius", -272.15);
30     kel.put("Fahrenheit", -457.72);
31     kel.put("Kelvin", 1.00);
32
33 }
34
35 public Double getCel(String key) {
36     return cel.get(key);
37 }
38 public Double getFah(String key) {
39     return fah.get(key);
40 }
41 public Double getKel(String key) {
42     return kel.get(key);
43 }
44 }
45
```

Modulo Superficie.java

Método Superficie

```
23 public Superficie() {  
24     kcud.put("K. cuadrados", 1.00);  
25     kcud.put("M. cuadradas", 0.386);  
26     kcud.put("Hectareas", 100.00);  
27     kcud.put("Acres", 247.106);  
28     kcud.put("Mtr. cuadrados", 1000000.0);  
29     kcud.put("Pies cuadrados", 10763.91);  
30  
31     mcud.put("K. cuadrados", 2.59);  
32     mcud.put("M. cuadradas", 1.00);  
33     mcud.put("Hectareas", 258.999);  
34     mcud.put("Acres", 640.0);  
35     mcud.put("Mtr. cuadrados", 2589987.83);  
36     mcud.put("Pies cuadrados", 27878395.92);  
37  
38     hect.put("K. cuadrados", 0.01);  
39     hect.put("M. cuadradas", 0.004);  
40     hect.put("Hectareas", 1.00);  
41     hect.put("Acres", 2.47);  
42     hect.put("Mtr. cuadrados", 10000.0);  
43     hect.put("Pies cuadrados", 107639.1);  
44  
45     acrs.put("K. cuadrados", 0.004);  
46     acrs.put("M. cuadradas", 0.002);  
47     acrs.put("Hectareas", 0.405);  
48     acrs.put("Acres", 1.00);  
49     acrs.put("Mtr. cuadrados", 4046.85);  
50     acrs.put("Pies cuadrados", 43559.97);  
51  
52     mrcd.put("K. cuadrados", 0.000001);
```



```
Source History
37
38     hect.put("K. cuadrados", 0.01);
39     hect.put("M. cuadradas", 0.004);
40     hect.put("Hectareas", 1.00);
41     hect.put("Acres", 2.47);
42     hect.put("Mtr. cuadrados", 10000.0);
43     hect.put("Pies cuadrados", 107639.1);
44
45     acrs.put("K. cuadrados", 0.004);
46     acrs.put("M. cuadradas", 0.002);
47     acrs.put("Hectareas", 0.405);
48     acrs.put("Acres", 1.00);
49     acrs.put("Mtr. cuadrados", 4046.85);
50     acrs.put("Pies cuadrados", 43559.97);
51
52     mrcd.put("K. cuadrados", 0.000001);
53     mrcd.put("M. cuadradas", 0.00000003);
54     mrcd.put("Hectareas", 0.0001);
55     mrcd.put("Acres", 0.000024);
56     mrcd.put("Mtr. cuadrados", 1.00);
57     mrcd.put("Pies cuadrados", 10.76);
58
59     fcud.put("K. cuadrados", 0.0000000092);
60     fcud.put("M. cuadradas", 0.0000000035);
61     fcud.put("Hectareas", 0.0000092);
62     fcud.put("Acres", 0.000022);
63     fcud.put("Mtr. cuadrados", 0.092);
64     fcud.put("Pies cuadrados", 1.00);
65
```

Modulo Peso.java

Método Peso()


```

23 public Peso() {
24     ton.put("Tonelada", 1.00);
25     ton.put("T. corta", 1.102);
26     ton.put("Kilogramo", 1000.0);
27     ton.put("Libra", 2204.62);
28     ton.put("Gramo", 1000000.0);
29     ton.put("Onza", 32150.74);
30
31     tcr.put("Tonelada", 0.907);
32     tcr.put("T. corta", 1.00);
33     tcr.put("Kilogramo", 907.18);
34     tcr.put("Libra", 2000.0);
35     tcr.put("Gramo", 907184.74);
36     tcr.put("Onza", 29166.66);
37
38     kgr.put("Tonelada", 0.001);
39     kgr.put("T. corta", 0.001);
40     kgr.put("Kilogramo", 1.00);
41     kgr.put("Libra", 2.20);
42     kgr.put("Gramo", 1000.0);
43     kgr.put("Onza", 32.15);
44
45     lbr.put("Tonelada", 0.0005);
46     lbr.put("T. corta", 0.0005);
47     lbr.put("Kilogramo", 0.45);
48     lbr.put("Libra", 1.00);
49     lbr.put("Gramo", 453.59);
50     lbr.put("Onza", 14.58);
51
52     grm.put("Tonelada", 0.000001);
53     grm.put("T. corta", 0.000001);
54
55     grm.put("Kilogramo", 0.001);
56     grm.put("Libra", 0.002);
57     grm.put("Gramo", 1.00);
58     grm.put("Onza", 0.03);
59
60     onz.put("Tonelada", 0.000002);
61     onz.put("T. corta", 0.000003);
62     onz.put("Kilogramo", 0.02);
63     onz.put("Libra", 0.06);
64     onz.put("Gramo", 28.34);
65     onz.put("Onza", 1.00);
66 }

```

Modulo Longitud.java

Método Longitud

```

23 public Longitud() {
24     kilo.put("Kilometros", 1.00);
25     kilo.put("Millas", 0.62);
26     kilo.put("Yardas", 1093.61);
27     kilo.put("Centimetros", 100000.0);
28     kilo.put("Milimetros", 10000000.0);
29     kilo.put("Pies", 3280.84);
30     kilo.put("Pulgadas", 39370.08);
31
32     mile.put("Kilometros", 1.61);
33     mile.put("Yardas", 1760.00);
34     mile.put("Centimetros", 160934.4);
35     mile.put("Milimetros", 1609344.0);
36     mile.put("Pies", 5280.00);
37     mile.put("Pulgadas", 63360.00);
38
39     yard.put("Millas", 0.000568);
40     yard.put("Kilometros", 0.000914);
41     yard.put("Centimetros", 91.44);
42     yard.put("Milimetros", 914.4);
43     yard.put("Pies", 3.00);
44     yard.put("Pulgadas", 36.00);
45
46     cent.put("Kilometros", 0.00001);
47     cent.put("Yardas", 0.01094);
48     cent.put("Millas", 0.00001);
49     cent.put("Milimetros", 10.00);
50     cent.put("Pies", 0.03281);
51     cent.put("Pulgadas", 0.3937);
52
53     mili.put("Millas", 0.000000062);
54     mili.put("Yardas", 0.00109361);

```

Modulo Convertidor.java

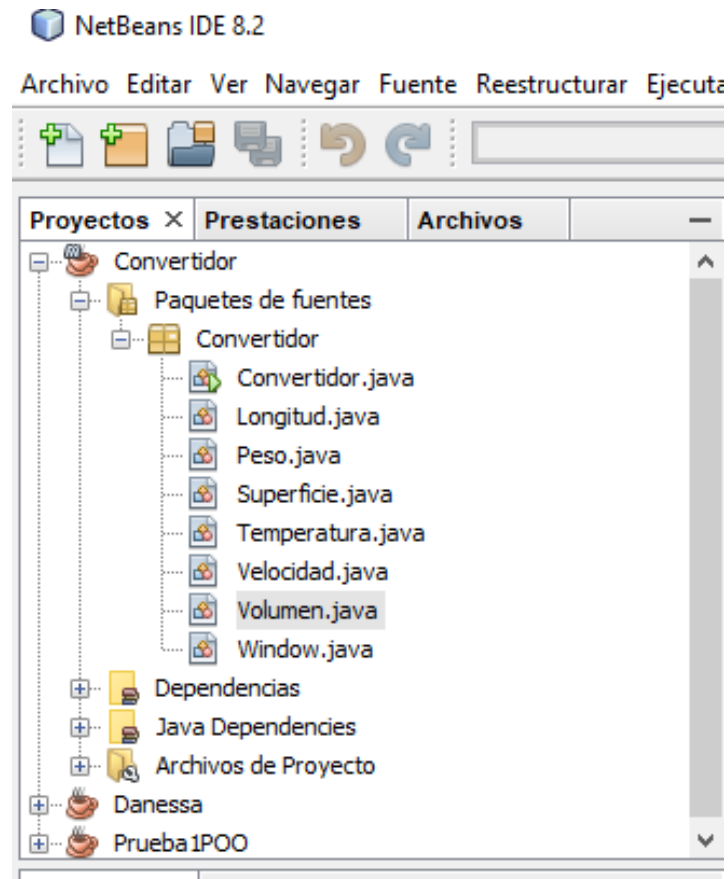
Métodos Convertidor()

```

6
7 public class Convertidor{
8     public static void main(String[] args) {
9         Window app = new Window();
10        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        app.setResizable(false);
12        app.setLocationRelativeTo(null);
13    }
14
15 }
16

```

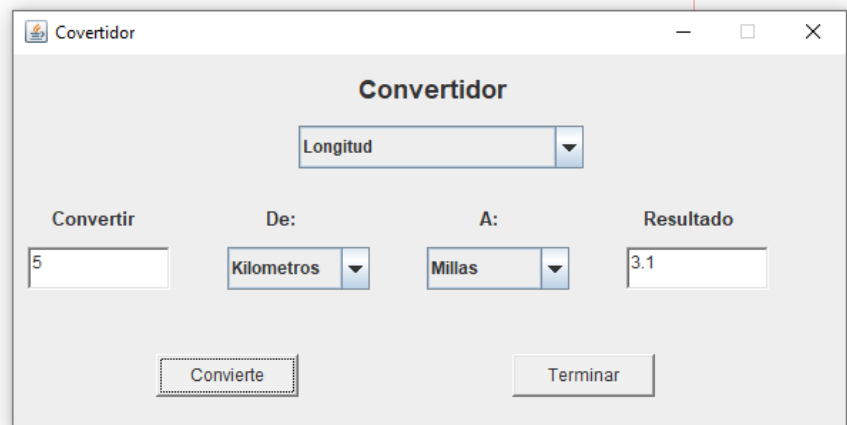
Imagen del navegador de proyectos



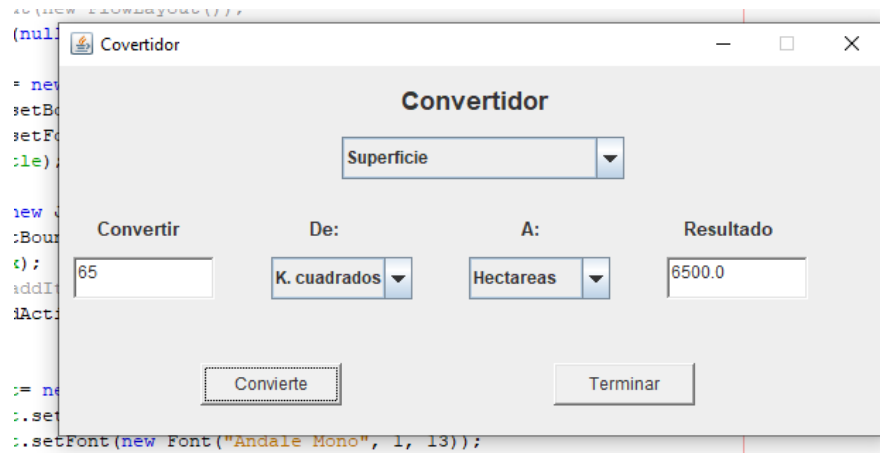
Copias de las ventanas

Despliegue de la ventana Convertidor

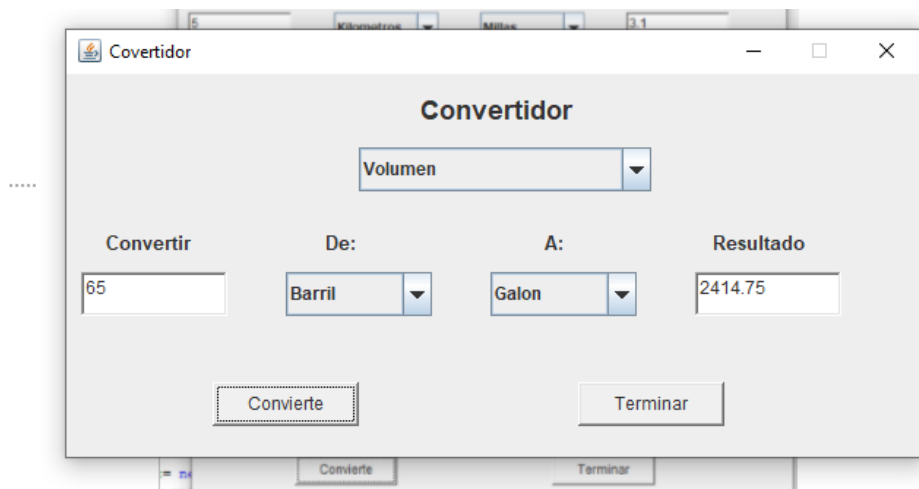
Prueba 1: Conversión de longitud



Prueba 2: Conversión de Superficie



Prueba 3: Conversión de Volumen



Prueba 4: Conversión de Peso

Convertidor

Peso

Convertir De: A: Resultado

65 Libra Tonelada 0.0325

Convierte Terminar

Prueba 5: Conversión de Velocidad

Convertidor

Velocidad

Convertir De: A: Resultado

65 Milla/hr Km/s 0.026000000000

Convierte Terminar

Prueba 6: Conversión de Temperatura

Convertidor

Temperatura

Convertir De: A: Resultado

65 Fahrenheit Kelvin 16635.45

Convierte Terminar

Comentarios finales

Santiago Landeros Edwin Uriel

Es importante considerar que la realización del proyecto centra su análisis únicamente en el informe final del equipo. Aunque se mantuvieron constantes las condiciones de cátedra impartidas por el profesor en cada clase, los temas abordados, las revisiones realizadas, las características solicitadas para el proyecto y el profesor, podrían existir factores particulares de los estudiantes, del profesor o del entorno que pudieran influir en las características del informe entregado ya que fue una nueva forma de abordar un proyecto.

Reveles Ramírez Jesús Emilio

El proyecto que realizamos ha contribuido de manera muy importante para identificar y resaltar los puntos que hay que cubrir y considerar para llevar a cabo una implementación exitosa del sistema. Nos deja muchas cosas importantes que reflexionar y muchas otras las ha reforzado como puntos angulares para llevar a cabo una buena implementación.

Villegas Nolasco Elide Yolotzin

A mi parecer el informe sobre el sistema ya había sido abordado con detalles por el profesor en las clases, lo que hizo que solo nos enfocáramos como equipo en una exitosa implementación, sin embargo, de la experiencia adquirida puedo decir que siempre es mucho mejor llevar acabo un análisis de las distintas posibilidades para reducir el riesgo de que las cosas no salgan de la manera que deseamos. Llevar a cabo un análisis detallado antes de comenzar en este proyecto, incrementa en gran proporción las probabilidades de tener éxito ya que de ante mano se conoce lo que se quiere lograr y cómo se va a hacer para lograrlo.

Herrera Cano Edwin Rubén

Una de las etapas que más nos costó trabajo a la hora de desarrollar el informe del proyecto, y la realización de la misma, fueron el inicio del desarrollo y la creación de las interfaces. El inicio del desarrollo fue complicado debido a la diferencia existente entre los integrantes sobre los conocimientos previos sobre cómo abordar la resolución de problemas ante el sistema que necesitábamos entender. El desarrollo del sistema usando Java también resultaría importante comparar nuestros conocimientos de cada uno de los integrantes del equipo para estar atentos en puntos de mejoras para todos.

Palmieri Mondragón Gerardo

Como trabajo futuro para el equipo se plantea repetir la forma de análisis y técnicas una retroalimentación a partir del primer informe entregado, así como con las revisiones y aclaraciones que el profesor nos aporta para aumentar la validación de los resultados y considerar el grado de generalización de los mismos, en el equipo de trabajo aplicamos métricas para determinar si hay diferencias significativas entre los avances que habíamos tenido en el transcurso de las clases y así poder tener con aportaciones diferentes.

