



Encapsulamiento de datos con Netbeans y concepto.

Como se puede observar de los diagramas, las variables del objeto se localizan en el centro o núcleo del objeto. Los métodos rodean y esconden el núcleo del objeto de otros objetos en el programa. Al empaquetamiento de las variables de un objeto con la protección de sus métodos se le llama *encapsulamiento*. Típicamente, el encapsulamiento es utilizado para esconder detalles de la puesta en práctica no importantes de otros objetos. Entonces, los detalles de la puesta en práctica pueden cambiar en cualquier tiempo sin afectar otras partes del programa.

El encapsulamiento de variables y métodos en un componente de software ordenado es, todavía, una simple idea poderosa que provee dos principales beneficios a los desarrolladores de software:

- **Modularidad**, esto es, el código fuente de un objeto puede ser escrito, así como darle mantenimiento, independientemente del código fuente de otros objetos. Así mismo, un objeto puede ser transferido alrededor del sistema sin alterar su estado y conducta.
- **Ocultamiento de la información**, es decir, un objeto tiene una "interfaz publica" que otros objetos pueden utilizar para comunicarse con él. Pero el objeto puede mantener información y métodos privados que pueden ser cambiados en cualquier tiempo sin afectar a los otros objetos que dependan de ello.

Los objetos proveen el beneficio de la modularidad y el ocultamiento de la información. Las clases proveen el beneficio de la reutilización. Los programadores de software utilizan la misma clase, y por lo tanto el mismo código, una y otra vez para crear muchos objetos.

En las implantaciones orientadas a objetos se percibe un objeto como un paquete de datos y procedimientos que se pueden llevar a cabo con estos datos. Esto encapsula los datos y los procedimientos. La realidad es diferente: los atributos se relacionan al objeto o instancia y los métodos a la clase. ¿Por qué se hace así? Los atributos son variables comunes en cada objeto de una clase y cada uno de ellos puede tener un valor asociado, para cada variable, diferente al que tienen para esa misma variable los demás objetos. Los métodos, por su parte, pertenecen a la clase y no se almacenan



en cada objeto, puesto que sería un desperdicio almacenar el mismo procedimiento varias veces y ello va contra el principio de reutilización de código.

Cómo lo hacemos en el Netbeans.

si tenemos varias variables de instancias y estas tenemos que generar los famosos **get y set** (ver material complementario , getters y setters de esta misma entrega) , el IDE Netbeans nos proporciona una forma muy sencilla de realizar esta tarea.

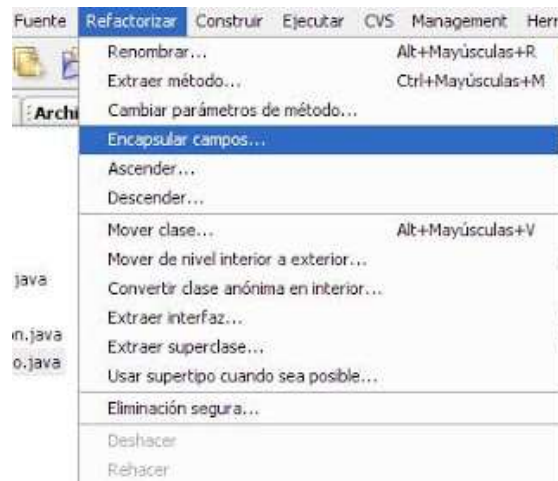
Pasos para encapsular las variables .

1.- crear una clase con sus atributos (del objeto)

```
12  /**
13   *
14   * @author jorge
15   */
16  public class encapsulamiento {
17
18
19      String nombre;
20      String apellido;
21      String rut;
22      int edad;
23
24
25  public encapsulamiento() {
26      }
27
28  }
```



2.- en la barra de netbeans buscar **refactorizar** y seleccionar **Encapsular campos**.



3.- luego saldrá este cuadro donde seleccionamos que atributos vamos a encapsular, (se puede elegir solamente **set** o **get**).



4.- luego Siguiente y nos muestra el siguiente cuadro en la parte inferior del ide.



5.- y por ultimo **refáctorizar** .como seleccionamos todos los atributos **private** y sus **get y set** respectivos nos queda la clase asi

```
public class encapsulamiento {
```

```
    private String nombre;
    private String apellido;
    private String rut;
    private int edad;
```

```
    public encapsulamiento() {
    }

```

```
    public String getNombre() {
        return nombre;
    }

```

```
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

```

```
    public String getApellido() {
```



```
return apellido;  
}
```

```
public void setApellido(String apellido) {  
    this.apellido = apellido;  
}
```

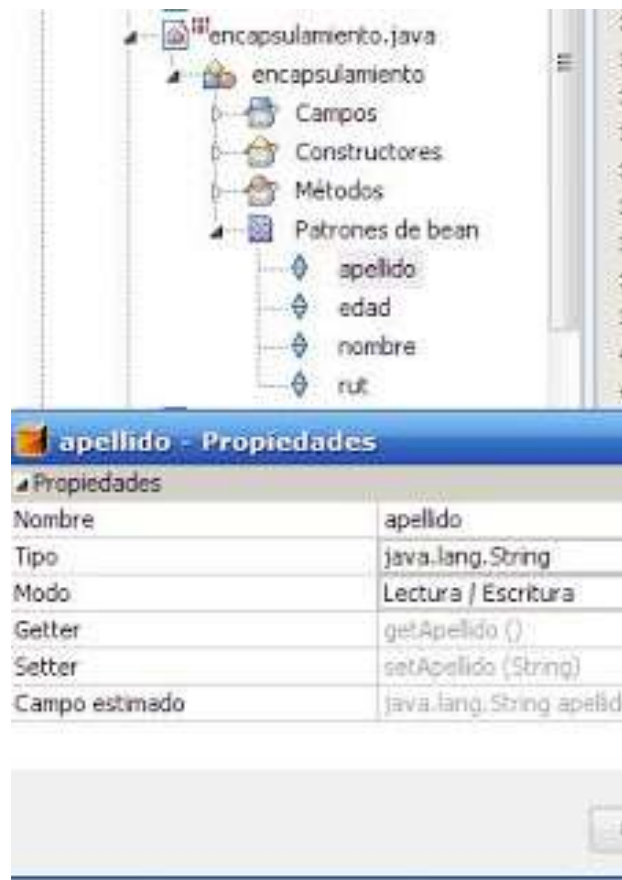
```
public String getRut() {  
    return rut;  
}
```

```
public void setRut(String rut) {  
    this.rut = rut;  
}
```

```
public int getEdad() {  
    return edad;  
}
```

```
public void setEdad(int edad) {  
    this.edad = edad;  
}  
}
```

otra forma de ver y modificar los **get** y **set** mediante el IDE es mediante los patrones de **beans** , bueno en la imagen quedara claro el cuadro inferior de la imagen es la propiedad de apellido y en el modo que podemos cambiar su estructura.



Video tutorial de muestra

<https://www.youtube.com/watch?v=8LyjCiuynKw>