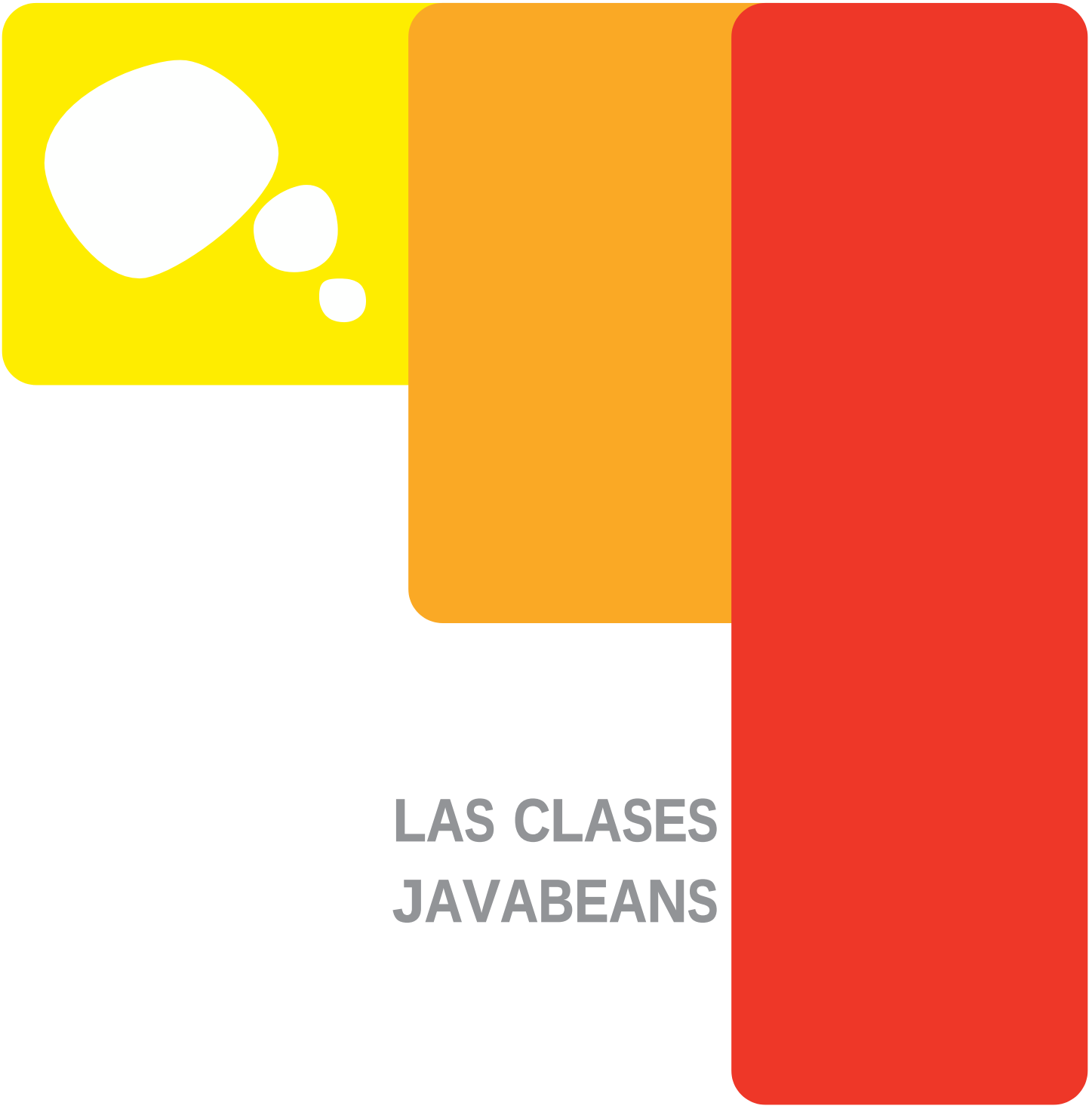
élogos



LAS CLASES JAVABEANS

élogos

ÍNDICE

LAS CLASES JAVABEANS

1. Clases de tipo JavaBean 3

1. Clases de tipo JavaBean

Diferencia entre un objeto y un componente

En Java un objeto y un componente aparentemente son lo mismo. Sin embargo, la diferencia radica en que un componente es reutilizable mientras que el periodo de vida de un objeto es desde que empieza el programa hasta que éste termina.

Ventaja de los JavaBeans

La especificación de JavaBeans de Sun Microsystems los define como "componentes de software reutilizables que se puedan manipular visualmente en una herramienta de construcción".

Entorno de desarrollo de aplicaciones (IDE)

Podemos crear una aplicación en un IDE seleccionando los componentes visibles e invisibles en una paleta de herramientas para situarlas sobre un panel o una ventana, arrastrándolos con el ratón.

Con el ratón unimos los sucesos (events) que generan un objeto (fuente), con los objetos (listeners) interesados en responder a las acciones sobre dicho objeto.

Características de los Componentes JavaBeans

A pesar de haber muchas semejanzas, los JavaBeans no deben confundirse con los Enterprise JavaBeans (EJB), una tecnología de componentes del lado servidor que es parte de Java EE a la que ya hicimos referencia en apartados anteriores.

Entre las principales características de los JavaBeans destacamos:

Introspección: permite analizar a la herramienta de programación o IDE como trabaja el Bean.

Customización: el programador puede alterar la apariencia y la conducta del Bean.

Events: informa al IDE de los sucesos que puede generar en respuesta a las acciones del usuario o del sistema, y también los sucesos que puede manejar.

Properties: permite cambiar los valores de las propiedades del Bean para personalizarlo (customization).

Persistencia: se puede guardar el estado de los Beans que han sido personalizados por el programador, cambiando los valores de sus propiedades.

Reglas para la creación de JavaBeans

No todos los JavaBeans tienen un aspecto visual. Por JavaBeans también se conoce a las clases que se utilizan en muchas aplicaciones Java para encapsular una serie de datos relativos a alguna entidad(libro, alumno, empleado, etc.), a fin de poder tener todos esos datos agrupados dentro de un mismo objeto.

Estas clases, deben de cumplir una serie de reglas:

- Un Bean tiene que tener un constructor por defecto (sin argumentos).
- Un Bean tiene que tener persistencia, es decir, implementar el interface Serializable.
- Un Bean tiene que tener introspección (instrospection). Los IDE reconocen ciertas pautas de diseño, nombres de las funciones miembros o métodos y definiciones de las clases, que permiten a la herramienta de programación observar dentro del Bean y conocer sus propiedades y su conducta.

Propiedades de un bean

Aunque en Java no existe el concepto de propiedad como tal, cuando se trabaja con JavaBean se habla de propiedades para hacer referencia a las características de los objetos.

Desde el punto de vista de la implementación, una propiedad se refiere a la pareja de métodos que se emplean para dar acceso al exterior a alguna característica del objeto. A dichos métodos se les conoce como **getter** y **setter**

El siguiente ejemplo corresponde a un JavaBean que encapsula los datos de un libro:

```
public class Libro {

   private String titulo;

   private String autor;

   private long isbn;

   //Constructores
```

```
public Libro(String titulo, String autor, long isbn) {
   this.titulo = titulo;
   this.autor = autor;
   this.isbn = isbn;
}
public Libro(){}
//metodos de propiedad
public String getAutor() {
   return autor;
}
public void setAutor(String autor) {
   this.autor = autor;
}
public long getIsbn() {
   return isbn;
}
public void setIsbn(long isbn) {
   this.isbn = isbn;
}
public String getTitulo() {
   return titulo;
}
```

```
    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }
}
```