

Alumno: Kevin Zamora Amela

Tarea: Tarea 6 (AD07)

Asignatura: Acceso a Datos

Programación Orientada a Componentes y JavaBeans

Actividad 1: Investigación

1. Concepto de Componente Software (¿Qué es un componente software?)

Un componente software es un elemento modular y autónomo que puede integrarse fácilmente en sistemas más grandes (javadeploy.com). Está diseñado como un bloque de construcción fundamental que facilita la creación de aplicaciones complejas mediante la combinación de elementos reutilizables.

2. Ventajas e Inconvenientes (¿Qué ventajas e inconvenientes tiene?)

Ventajas

Reutilización de código
Facilidad de mantenimiento
Integración flexible
Desarrollo paralelo
Pruebas más simples

Inconvenientes

Mayor complejidad inicial
Necesidad de documentación detallada
Posible sobrecarga en sistemas pequeños
Requiere planificación cuidadosa
Curva de aprendizaje inicial

3. Herramientas para Desarrollo de Componentes

(¿Qué herramientas existen para desarrollar componentes?)

- Frameworks de componentes (JavaBeans, Spring)
- IDEs especializados (NetBeans, Eclipse)
- Herramientas de empaquetado (Maven, Gradle)
- Entornos de desarrollo integrados

4. JavaBean: Concepto y Requisitos

(¿Qué es un JavaBean y qué requisitos debe cumplir?)

Un JavaBean es un componente software que sigue ciertas convenciones específicas (stackoverflow.com):

- Implementación de la interfaz 'Serializable'
- Constructor público sin argumentos
- Propiedades privadas con 'getters' y 'setters' públicos
- Encapsulación adecuada de datos

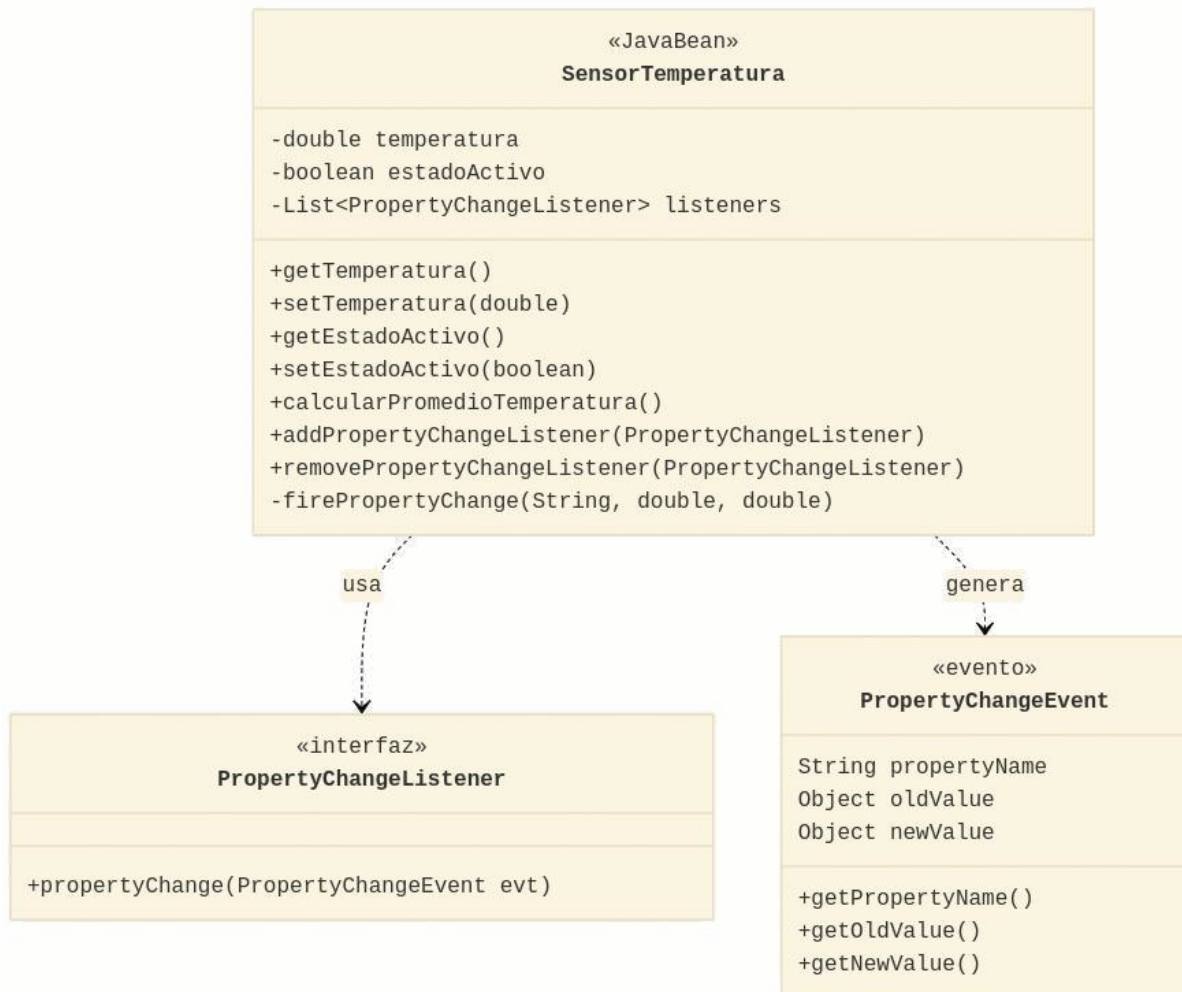
Actividad 2: Desarrollo del componente

JavaBean

Componente JavaBean: SensorTemperatura

Para cumplir con los requisitos del ejercicio, hemos desarrollado un componente JavaBean que simula un sensor de temperatura. Este componente será funcional y educativo, permitiendo entender todos los conceptos clave de los JavaBeans.

Primero, veamos la estructura del componente:

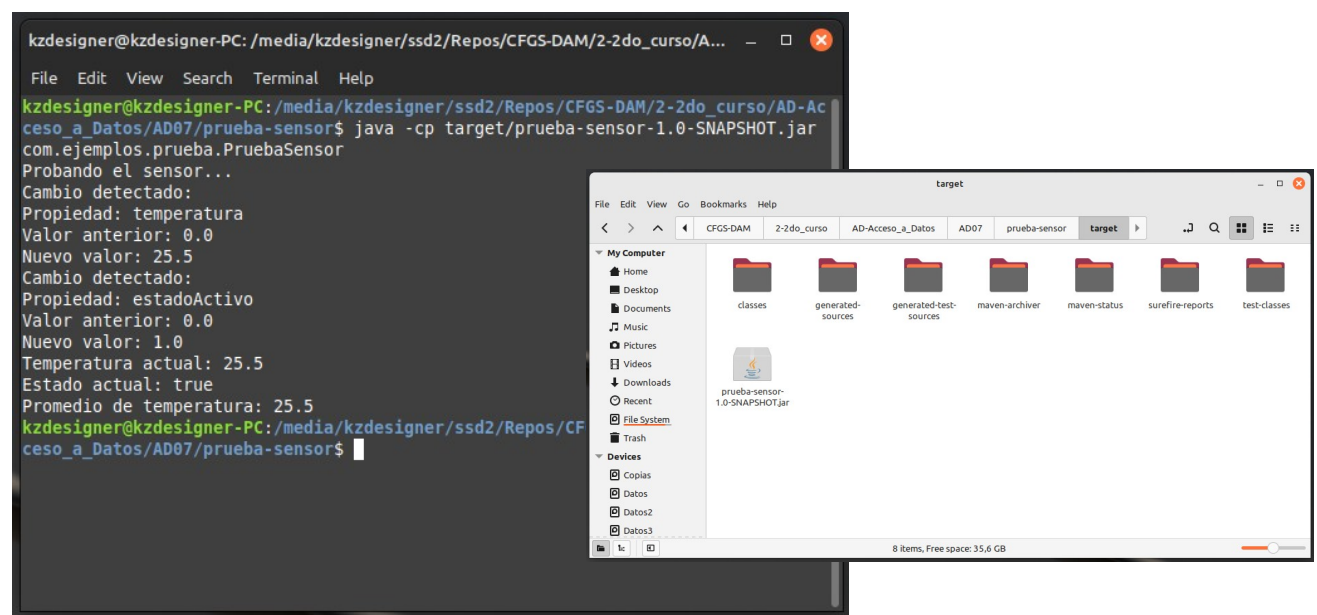
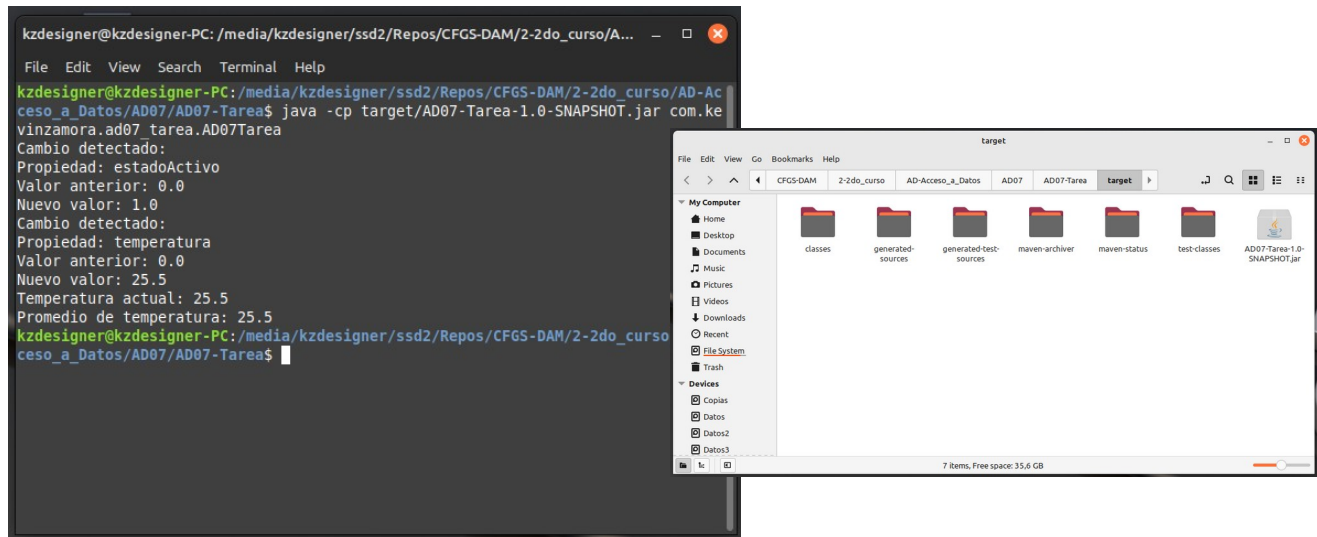


Explicación del Diagrama

El diagrama muestra la estructura completa de nuestro componente JavaBean **SensorTemperatura**:

- Las flechas punteadas (*..>*) indican dependencias: **SensorTemperatura** usa **PropertyChangeListener** y genera eventos **PropertyChangeEvent**.
- **PropertyChangeListener** es una interfaz que define el contrato para los oyentes de eventos.
- **PropertyChangeEvent** es una clase que encapsula la información del evento cuando cambian las propiedades.
- **SensorTemperatura** implementa todas las características requeridas de un JavaBean, incluyendo propiedades privadas y métodos públicos.

Anotación: Se adjuntan los dos archivos comprimidos .zip de los dos proyectos generados. En su interior y concretamente dentro de la carpeta 'target', se pueden encontrar los respectivos archivos ejecutables en formato .jar.



Actividad 3: Explicación del desarrollo

1. Introducción del componente y su finalidad

El componente SensorTemperatura es un JavaBean diseñado para simular un sensor de temperatura. Implementa las características estándar de un JavaBean, incluyendo serialización, propiedades con getters y setters, y un sistema de eventos para notificar cambios en las mediciones.

2. Su estructura:

El componente implementa dos propiedades principales:

- temperatura (double): Almacena el valor de temperatura actual
- estadoActivo (boolean): Indica si el sensor está operativo

Los métodos principales son:

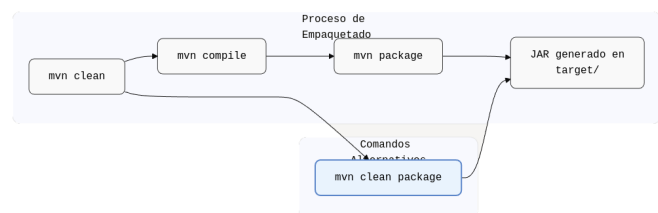
- getTemperatura() / setTemperatura(double)
- getEstadoActivo() / setEstadoActivo(boolean)
- calcularPromedioTemperatura(int)

El sistema de eventos notifica cambios en ambas propiedades mediante PropertyChangeEvents.

3. Proceso de empaquetado y prueba

El componente se empaqueta utilizando Maven, generando un archivo .jar en la carpeta target del proyecto. El proceso incluye:

1. Compilación del código fuente
2. Generación del MANIFEST.MF
3. Empaquetado en formato .jar
4. Instalación en el repositorio local de Maven

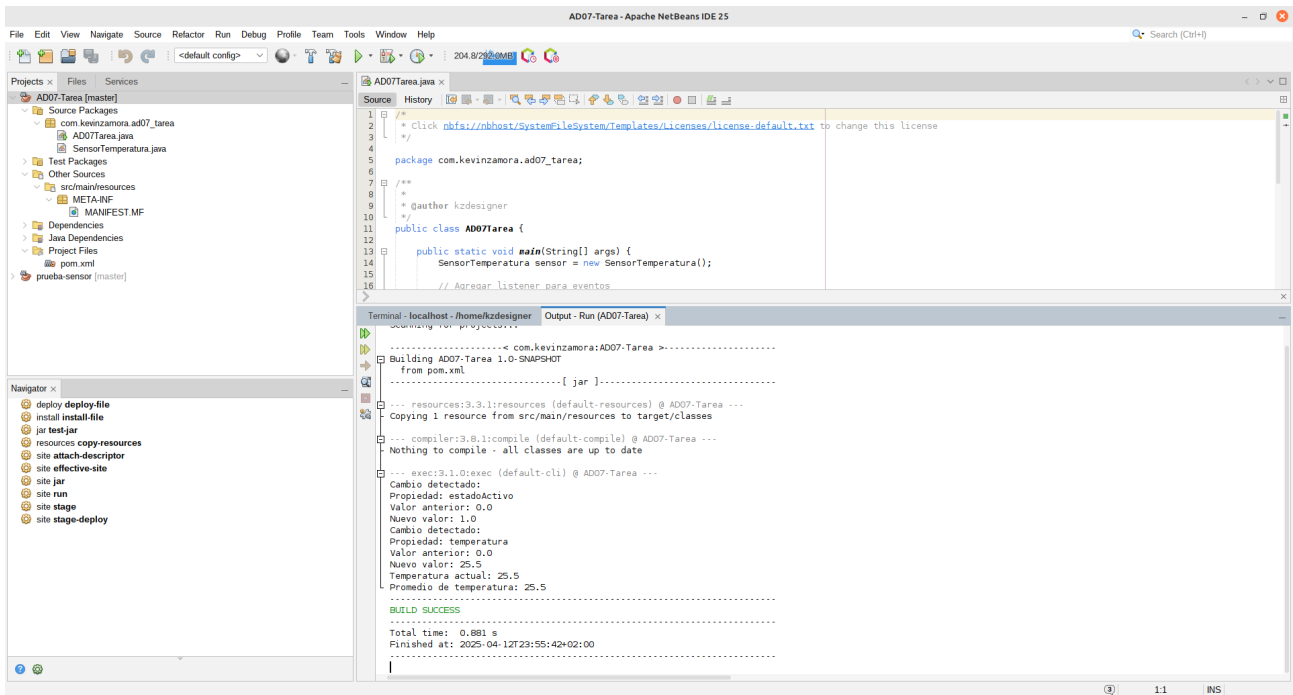


La prueba del componente verifica:

- La creación de la instancia
- La modificación de propiedades
- La recepción de eventos
- El funcionamiento del método de cálculo

4. Funcionamiento:

Componente:



Prueba del componente:

