

4.A. Refactorización

3. Refactorización en Eclipse.

3.9. Autoencapsular atributo.

Convierte una variable de clase en privada y genera los métodos *Get* y *Set* públicos para acceder a la misma. Opción también disponible desde el menú *código fuente* que se verá a continuación.

The diagram illustrates the Eclipse 'Encapsular campo' (Encapsulate Field) refactoring process. It shows the initial state of a Java class, the configuration of the dialog, and the final state of the code after refactoring.

Initial Code:

```
package clases;
// Clase utilizada para
public class persona {
    // Metodos de clase
    int i_Edad;
    String s_Nombre;
}
```

Encapsular campo Dialog:

- Nombre de método get: `getI_Edad` (nuevo getter creado)
- Nombre de método set: `setI_Edad` (nuevo setter creado)
- [Configurar convenciones de nombre...](#)
- Acceso a campos en tipo declarante: ☒ usar métodos setter y getter ☐ mantener referencia de campo
- Insertar métodos nuevos después de: Como primer método
- Modificador de acceso: ☐ public ☒ package
- ☐ Generar comentarios de método

Resulting Code:

```
package clases;
// Clase utilizada para ser herenc
public class persona {
    // Metodos de clase: Edad y nc
    private int i_Edad;
    String s_Nombre;

    int getI_Edad() {
        return i_Edad;
    }

    void setI_Edad(int i_Edad) {
        this.i_Edad = i_Edad;
    }
}
```