

Las anotaciones

Las anotaciones están muy presentes en las aplicaciones Java y merece la pena perder un rato en comprender que son exactamente.

En programación, una **Anotación Java** es una forma de añadir **metadatos** al código fuente Java que están disponibles para la aplicación en tiempo de ejecución. Muchas veces se usa como una alternativa a la tecnología XML.

<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-son-los-metadatos-y-cual-es-su-utilidad>

Las Anotaciones Java pueden añadirse a los elementos de programa tales como clases, métodos, propiedades y paquetes. Cuando se compila el código fuente de Java, el compilador Java almacena los metadatos de la Anotación en los ficheros/archivos de clases. Posteriormente, la JVM u otros programas pueden buscar los metadatos para determinar cómo interactuar con los elementos del programa o cambiar su comportamiento.

Las anotaciones permiten al programador declarar en su código fuente cómo debe comportarse el software.

Las anotaciones han aparecido a partir de Java 5 y cobran mucha importancia en los proyectos empresariales que utilizan JSF, EJB, JPA, etc.

La declaración de anotaciones tiene este formato:

```
@NombreAnotacion(atributo1="valor", atributo2="valor", .....)
```

Es posible crear nuestras propias anotaciones que serán declaradas como interfaces.

Anotaciones personalizadas

Para una mejor comprensión del funcionamiento de las anotaciones, vamos a crear un proyecto java estándar llamado PruebaAnotaciones donde crearemos una anotación personalizada llamada Editor.

```
package anotaciones;
```

```
import java.lang.annotation.*;
```

```
@Retention(RetentionPolicy.RUNTIME)
```

```
@Target(ElementType.METHOD)
```

```
public @interface Editor {
```

```
    String autor() default "PEPE SANCHEZ";
```

```

        int edicion() default 1;

        int anio() default 2005;
    }

```

Con el anterior ejemplo hemos creado la anotación personalizada Editor con los atributos autor, edición y año. Las anotaciones se crean como interfaces especiales usando el siguiente formato:

```

public @interface NombreAnotación {

}

```

Uso de la anotación

Ahora vamos a crear una clase llamada Coche con un método llamado acelerar que será anotado con la anotación Editor.

```

package prueba;

import anotaciones.*;

public class Coche {

    String marca;

    String modelo;

    int velocidad;

    public Coche(String marca, String modelo) {

        super();

        this.marca = marca;

        this.modelo = modelo;

        this.velocidad = 0;

    }

    @Editor(autor = "MIGUEL PICAFLOR", edicion = 3, anio = 2011)

    public void acelerar() {

        // Este es el método que contiene la anotación.

        this.velocidad +=10;
    }
}

```

```

    }

    @Override
    public String toString() {
        return "Coche [marca=" + marca + ", modelo=" + modelo + ", velocidad=" + velocidad + "];"
    }
}

```

Ahora vamos a crear una clase con método main donde declararemos un objeto de la clase Coche y aplicaremos conceptos de reflexión para obtener información de la clase, del método acelerar y de la anotación.

```

package prueba;

import java.lang.annotation.Annotation;
import java.lang.reflect.Method;
import anotaciones.Editor;

public class Principal {

    public static void main(String[] args) {

        Coche co = new Coche("Ford", "Fiesta");

        try {

            Class cl = co.getClass();

            Method m = cl.getMethod("acelerar");

            Annotation a = m.getAnnotation(Editor.class);

            Editor e = (Editor) a;

            System.out.println(a);

            System.out.println(e.autor());

            if (e.edicion() == 3) {

                co.acelerar();

                co.acelerar();

            }

            else {

```

```

        co.acelerar();
    }

    System.out.println(co.toString());
} catch (NoSuchMethodException | SecurityException e) {
    System.out.println(e.getMessage());
}
}
}
}

```

Ejemplos de aplicación de anotaciones

Un ejemplo de uso de anotaciones se encuentra en los servlets dentro de una aplicación web. La configuración del servlet se lleva a cabo por medio de anotaciones. En este caso se está estableciendo la url (ruta) de acceso al servlet y unos parámetros de inicialización que podrán recuperarse en el método init.

```

@WebServlet(
    description = "Servlet que nos saluda",
    urlPatterns = { "/HolaMundo" },
    initParams = {
        @WebInitParam(name = "Autor", value = "Amelia"),
        @WebInitParam(name = "Fecha", value = "Nov-2021")
    })

public class HolaMundo extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private String autor;
    private String fecha;

    public HolaMundo() {
        super();
    }

    @Override
    public void init(ServletConfig config) throws ServletException {
        // TODO Auto-generated method stub
        super.init(config);
        autor = config.getInitParameter("Autor");
        fecha = config.getInitParameter("Fecha");
    }
}

```

```

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        PrintWriter salida = response.getWriter();
        String nombre = request.getParameter("nombre");
        response.setContentType("text/html");
        salida.append("<!DOCTYPE html>");
        salida.append("<html>");
        salida.append("<head>");
        salida.append("<meta charset='UTF-8'>");
        salida.append("<title>Web dinámica</title>");
        salida.append("</head>");
        salida.append("<body>");

        salida.append("Hola "+nombre);
        salida.append("<h1>¿Qué pasa contigo?</h1>");
        salida.append("Autor: "+autor+"<br/>");
        salida.append("Fecha: "+fecha);

        salida.append("</body>");
        salida.append("</html>");
    }

    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}

```

Este servlet podría ser invocado desde una página web con un formulario similar a este:

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Formulario</title>
</head>
<body>
    <form action="HoLaMundo" method="get">
        <label>Dime tu nombre: </label>
        <input type="text" name="nombre">
        <input type="submit" value="Enviar">
    </form>
</body>
</html>

```