Paso 1. Crea un proyecto java en Eclipse.
Paso 2. Conviértelo en un proyecto Maven.
Paso 3. Incluye las dependencias a Spring Context y Aspectj:

```
<artifactId>spring-context</artifactId>
<artifactId>aspectjweaver</artifactId>
```

Paso 4. Crea una interfaz de negocio

```
package aspectos.negocio;
public interface ILogicaNegocio
{
public void foo(String p1);
public void ant(String p2);
}
```

Paso 5. Crea la implementación

```
package aspectos.negocio;
public class LogicaNegocio implements ILogicaNegocio
{
public void foo(String p1)
{
System.out.println("********************");
System.out.println("Dentro de LogicaNegocio.foo()");
System.out.println("********************");
}
public void ant(String p1)
{
System.out.println("********************");
System.out.println("Dentro de LogicaNegocio.ant()");
System.out.println("********************");
}
}
```

Paso 6. Añade las anotaciones necesarias para que la implementación sea un componente de Spring.
Paso 7. Crea el archivo de configuración de spring("beans.xml") con las etiquetas necesarias para reconocer las beans y aspectos anotadas .
Paso 8. Crea una clase Tester, que tenga un main, recupere un objeto de tipo ILogicaNegocio del applicationContext de spring y ejecute foo y ant.
Paso 9. Crear un Aspect que tenga un advice de tipo Before para el método ant (String param), el cual imprima la hora en la que se llama al método y el parámetro con el cual se le ha llamado.
Paso 10. Añade el siguiente código a la Clase de LogicaNegocio:

```
public String cadena(String p){
return p;
}
```

Paso 11. Crea un advice ensuciador, el cual añada la cadena ".." al return del método cadena. De esta forma si yo invoco a cadena pasándole "hola" este me devuelva "hola.." .

Ejercicio 2.

Paso 1. Crear un nuevo paquete.

Paso 2. Crea otro archivo de configuración de spring("beansEspectaculo.xml") con las etiquetas necesarias para reconocer los aspectos anotadas .

Paso 3. Crear el Aspect con sus determinados Advices y PointCut mediante código y anotaciones.

```
package espectaculo;
import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Pointcut;
@Aspect
public class Espectador {
public Espectador(){}
//Nuevo método para definir el pointcut
@Pointcut("execution(* *.interpretar(..))")
public void realizar(){}
//La anotación se apoya en un método creado donde se encuentra el pointcut
@Around("realizar()")
public Object miAdviceAspectJ(ProceedingJoinPoint joinpoint) throws
Throwable{
Object retVal = null;
this.tomarAsiento();
this.apagarTelefono();
try {
retVal= joinpoint.proceed();
this.aplaudir();
} catch (Throwable e) {
this.solicitarDevolucion();
throw e;
}
return retVal;
}

private void solicitarDevolucion() {
System.out.println("Esto es una excepción, devuelvame mi dinero");
}
private void aplaudir() {
System.out.println("aplausossss");
}
private void apagarTelefono() {System.out.println("apago telefono"); }
private void tomarAsiento() {System.out.println("tomo asiento"); }
}
```

Paso 4. Crear el resto de clases

```
package espectaculo;
public class Instrumento {
public void tocar(){
System.out.println("Tocando...");
}
}
package espectaculo;
```

```
public interface Interprete {
public String interpretar() throws Exception;
}
package espectaculo;
public class Musico implements Interprete {
private Instrumento instrumento;
private String cancion;
public Musico(){}
@Override
public String interpretar() throws Exception {
instrumento.tocar();
//Descomentar si se quiere probar el manejor de la exception
// throw new Exception("error de interpretación");
return cancion;
}
public Instrumento getInstrumento() {
return instrumento;
}
public void setInstrumento(Instrumento instrumento) {
this.instrumento = instrumento;
}
public String getCancion() {
return cancion;
}
public void setCancion(String cancion) {
this.cancion = cancion;
}
}

package espectaculo;
public class Espectaculo {
public static void main(String args[]){
ApplicationContext ac= new
ClassPathXmlApplicationContext("beansEspectaculo.xml");
Interprete m = (Interprete) ac.getBean("musico");
try {
m.interpretar();
} catch (Exception e) {
e.printStackTrace();
}
}
}
```

Paso 5. Añade al fichero de configuración los beans de músico e instrumento con la canción:"ODA Alegría" y ejecuta el main.