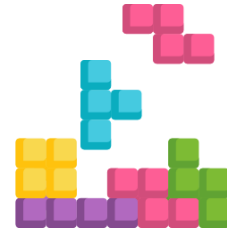


## EJEMPLO PRÁCTICO

Para aplicar el validador `RequiredStringValidator` a los atributos "usuario" y "clave", que se encargará de verificar que se ha incluido por lo menos un carácter en un atributo, hay que especificar la clase `Inicio`, por lo a continuación veremos un ejemplo de ello.



```
package ejemplo;

import com.opensymphony.xwork2.ActionSupport;
import com.opensymphony.xwork2.validator.annotations.*;

@Validation

public class Inicio extends ActionSupport {

    private String usuario;

    private String clave;

    public String execute() throws Exception {

        if(getClave().equals("administrador"))

            return SUCCESS; //se define una constante en ActionSupport

        else

            return ERROR; //se define una constante en ActionSupport

    }

    public void setUsuario(String usu){

        this.usuario = usu;

    }

    //anotación de cadena requerida

    @RequiredStringValidator(message="Debe incluir un usuario")

    public String getUsuario(){

        return this.usuario;

    }

    public void setClave(String cla){
```

```
this.clave = cla;

}

//anotación cadena requerida

@RequiredStringValidator(message="Debe incluir una clave")

public String getClave(){

return this.clave;

}

}
```

El empleo de anotaciones para especificar las reglas de validación hace que ya no sea necesario crear el fichero Inicio-validation.xml.

Si se diese el caso de que los mensajes se consiguen desde un fichero de recursos, en vez de emplear la propiedad *message* en la anotación, utilizaremos *key*, por medio de la cual definiremos la clave conectada al mensaje de error:

```
@RequiredStringValidator(key="clave.nulo")

public String getClave(){

return this.clave;

}
```

Si no se cumple con alguno de los criterios de validación especificados en los distintos métodos *get*, por medio de las anotaciones, el usuario será dirigido a la página de resultado señalada con el nombre "input".

Si esta página es la misma que posee los atributos de recolección de datos, no será requerido emplear el tag `<s:actionerror>` para ver los mensajes de error, dado que las anotaciones de validación incluyen como funcionalidad implícita la ejecución de los mensajes de error asociados a los atributos a cada método *get()*.