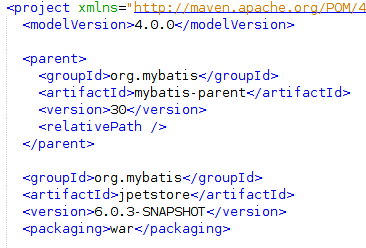
Intro

aplicaciones web

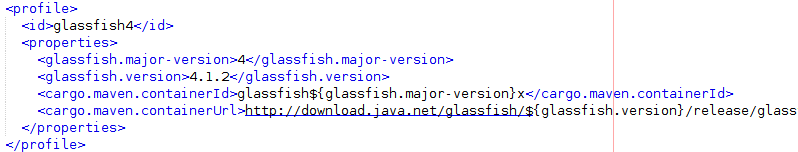
* empaquetado → .war *(el artefacto generado ha de ser desplegado en el servidor)*
* pom.xml → añadir <packaging>war</packaging> *(por defecto Maven empaqueta en .jar)*

**

* obtener .war
  + new action: generate war file
    - cmd: mvn clean package
* despliege → en contenedor web
  + servidor de aplicaciones Glassfish
  + plugin
    - cargo-maven2-plugin *(pone en marcha server + despliegue app en contenedor web)*
  + servidor de despliegue Maven - por defecto - → Tomcat
    - usamos: Glassfish4
      * pom.xml → definir perfil

<profiles>

…

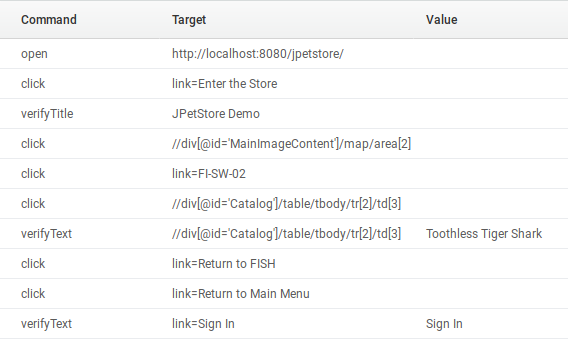


</profiles>

* + new action: start Glassfish and deploy war
    - cmd: mvn cargo:run -P glassfish4
* detener servidor
  + new action: stop Glassfish
    - cmd: mvn cargo:stop -P glassfish4

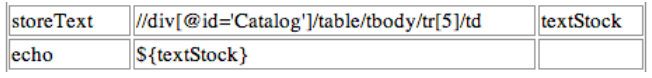
\*si falla unzip de glassfish4, pegar en /tmp/cargo/installs: http://download.java.net/glassfish/4.1.2/release/glassfish-4.1.2.zip\*

1. Test Suite - Ejecicio1
   1. grabar secuencia de acciones



* 1. guardar nuestra Suite → en formato .html
  2. ejecutar nuestro test
     1. observar traza de ejecución → pestaña "Log"

1. Test Suite - Ejecicio2
   1. init - test compra new user
   2. test compra old user
   3. variables
      1. almacenar texto en variable



* + 1. obtener todos los caracteres de una variable que cumplen un patrón (en este caso que sean números)



* + 1. operar con las variables almacenadas

