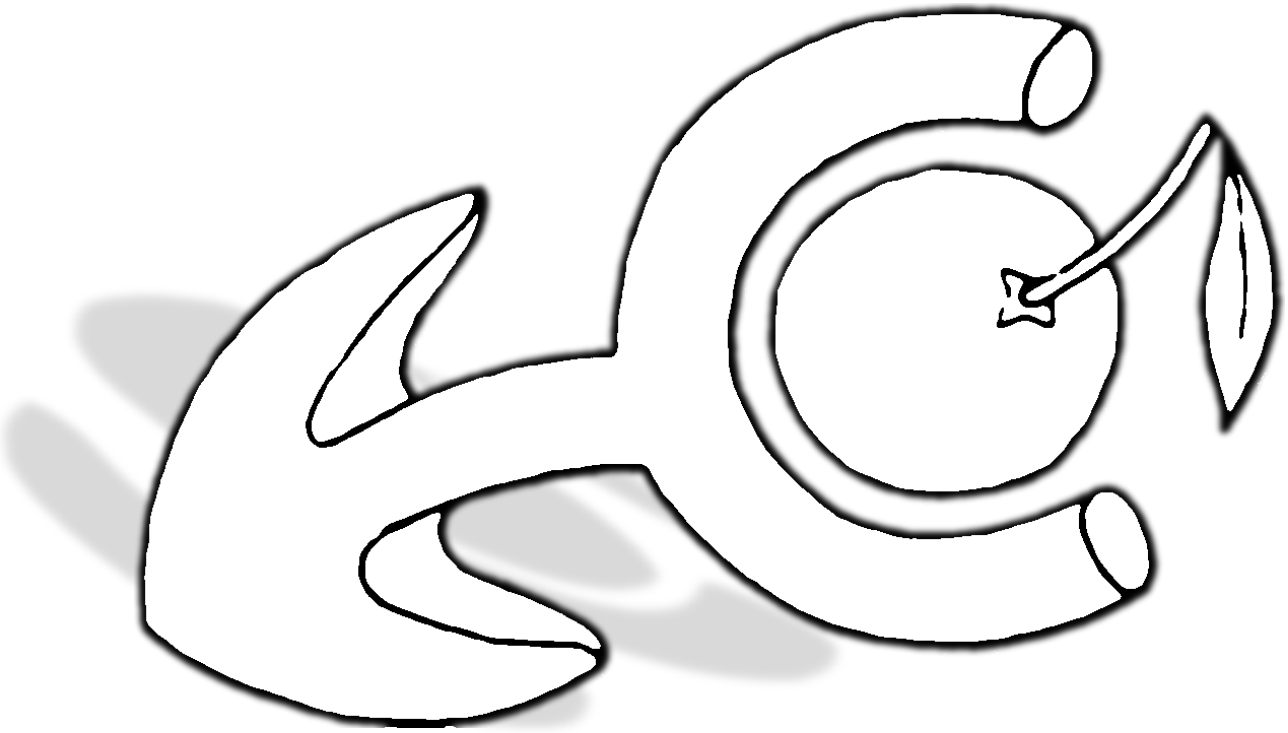


Diseño de máquinas de estado con eCo

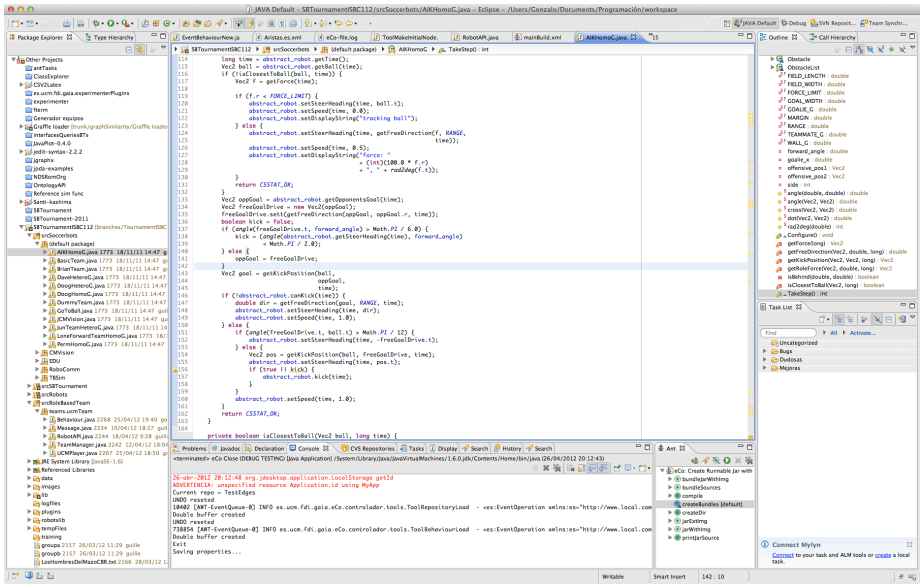
Gonzalo Flórez
gflorez@fdi.ucm.es

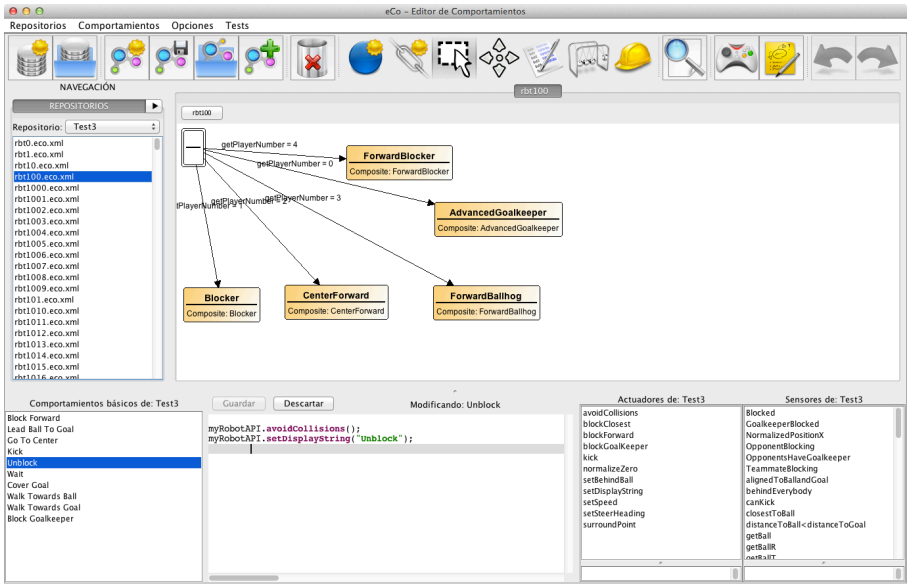
Departamento de Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid

¿Qué es eCo?

- Editor visual de máquinas de estado
- En lugar de programar los comportamientos

¿Por qué no los dibujamos?





INSTALACIÓN

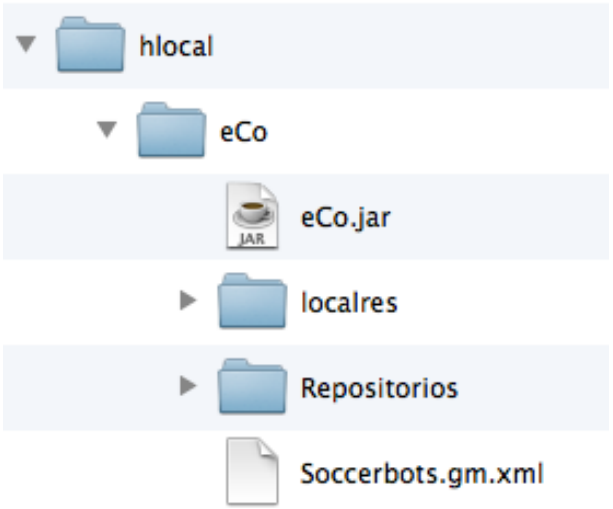
1. **Descargar**
2. Descomprimir
3. Ejecutar

- Descargar del campus virtual

INSTALACIÓN

1. Descargar
2. **Descomprimir**
3. Ejecutar

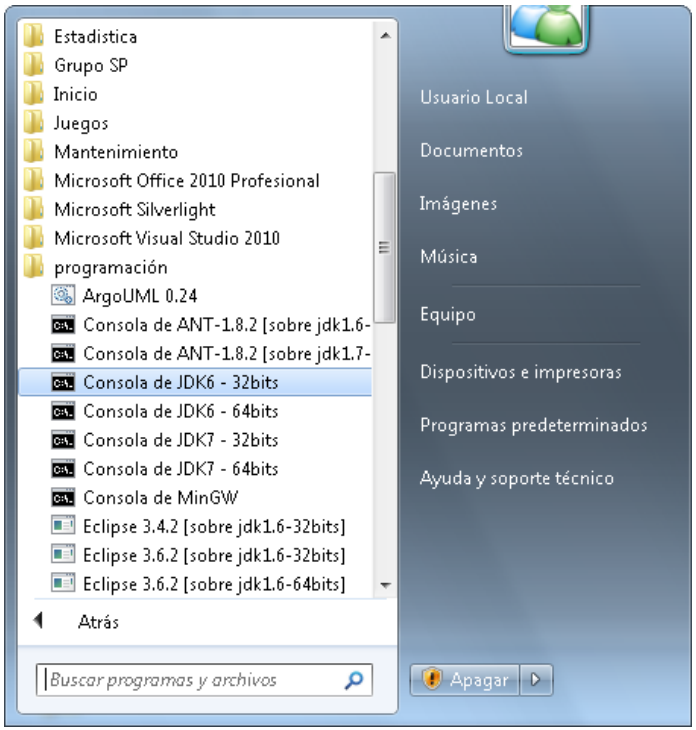
- Descomprimir el zip en c:\hlocal\eCo
- Crear la carpeta eCo\Repositorios



INSTALACIÓN

1. Descargar
2. Descomprimir
3. **Ejecutar**

- Abrir una consola de JDK 1.6 – 32bits



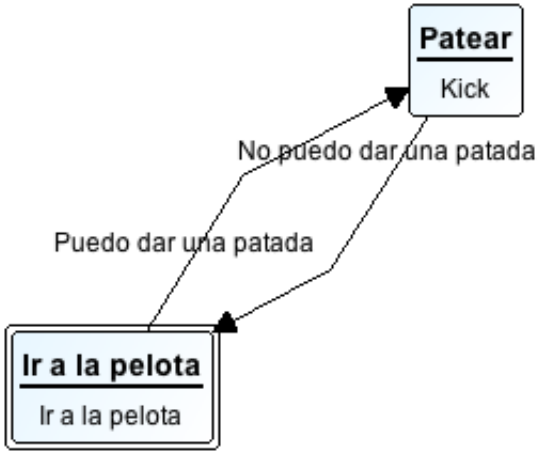
- Ir a la carpeta eCo y ejecutar el editor:

```
>c:
>cd c:\hlocal\eCo
>java –jar eCo.jar DEBUG
```

- Organización del editor
  - Repositorios
    - Colección de máquinas de estado para un juego

Organización del editor

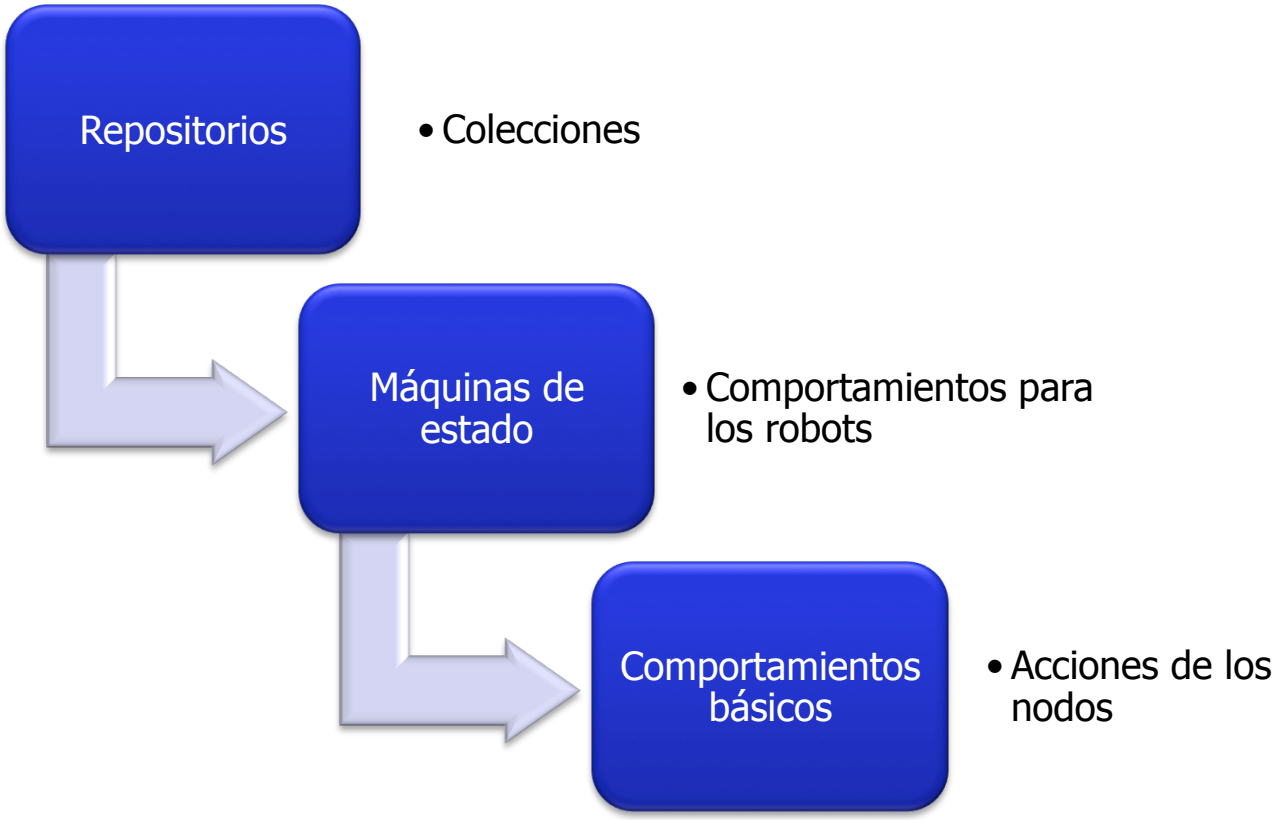
- Máquinas de estado
  - Cada una es un comportamiento
  - Formadas por NODOS y ARISTAS
  - Las aristas tienen condiciones
  - Los nodos tienen acciones ➔
    - Comportamientos básicos

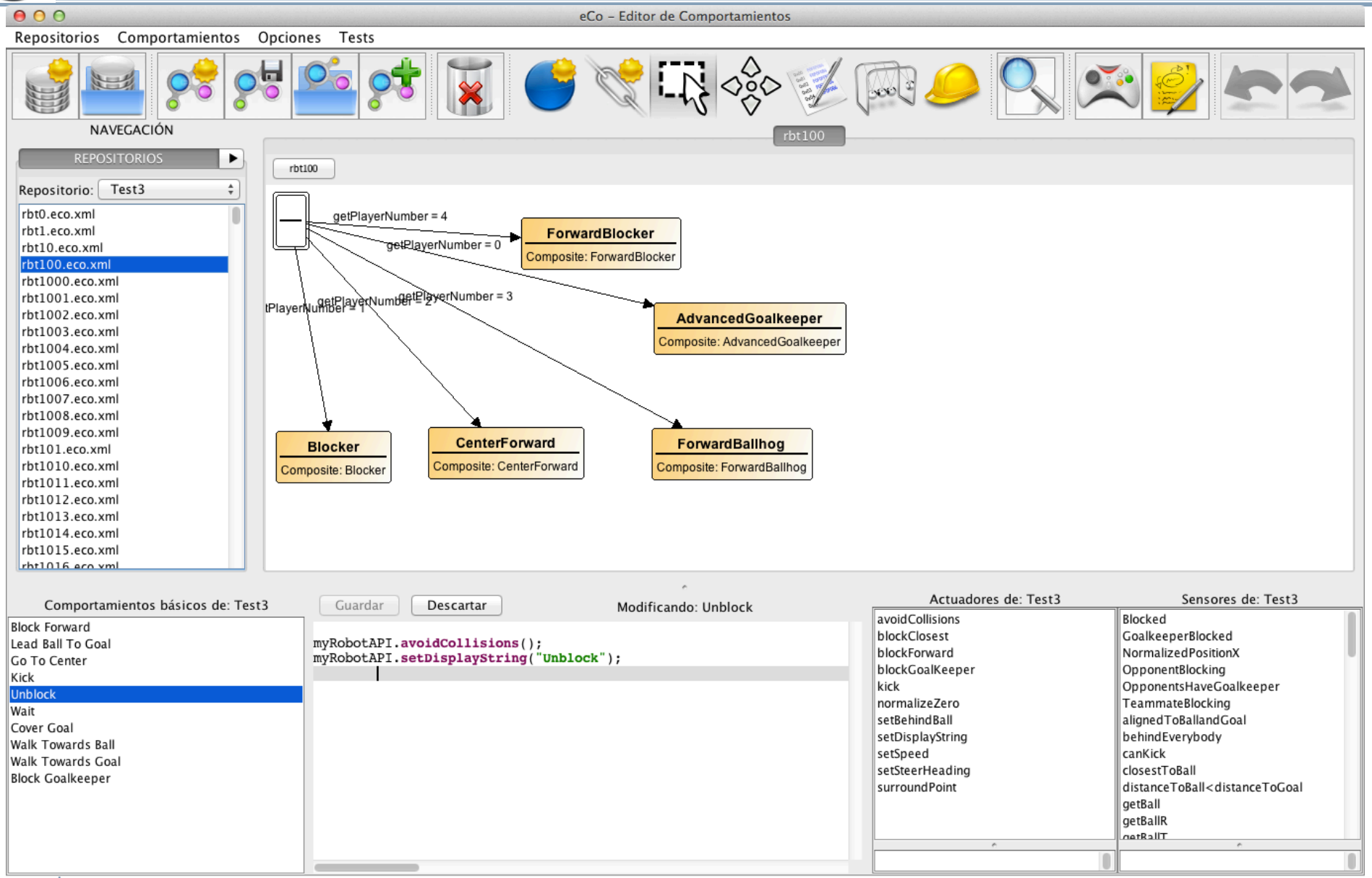


- Comportamientos básicos
  - Acciones que van en los nodos
  - Fragmentos de código Java
  - Utilizan la RobotAPI:

```
double angle = myRobotAPI.getBall().t;
myRobotAPI.setSteerHeading(angle);
myRobotAPI.setSpeed(1.0);
```

Organización del editor




eCo – Editor de Comportamientos
Repositorios
Comportamientos
Opciones
Tests
NAVEGACIÓN
rbt100
REPOSITORIOS
Repositorio: Test3
rbt0.eco.xml
rbt1.eco.xml
rbt10.eco.xml
rbt100.eco.xml
rbt1000.eco.xml
rbt1001.eco.xml
rbt1002.eco.xml
rbt1003.eco.xml
rbt1004.eco.xml
rbt1005.eco.xml
rbt1006.eco.xml
rbt1007.eco.xml
rbt1008.eco.xml
rbt1009.eco.xml
rbt101.eco.xml
rbt1010.eco.xml
rbt1011.eco.xml
rbt1012.eco.xml
rbt1013.eco.xml
rbt1014.eco.xml
rbt1015.eco.xml
rbt100
getPlayerNumber = 4
getPlayerNumber = 0
getPlayerNumber = 3
ForwardBlocker
Composite: ForwardBlocker
AdvancedGoalkeeper
Composite: AdvancedGoalkeeper
Blocker
Composite: Blocker
CenterForward
Composite: CenterForward
ForwardBallhog
Composite: ForwardBallhog
Comportamientos básicos de: Test3
Block Forward
Lead Ball To Goal
Go To Center
Kick
Unblock
Wait
Cover Goal
Walk Towards Ball
Walk Towards Goal
Block Goalkeeper
Guardar
Descartar
Modificando: Unblock
myRobotAPI.avoidCollisions();
myRobotAPI.setDisplayString("Unblock");
Actuadores de: Test3
avoidCollisions
blockClosest
blockForward
blockGoalKeeper
kick
normalizeZero
setBehindBall
setDisplayString
setSpeed
setSteerHeading
surroundPoint
Sensores de: Test3
Blocked
GoalkeeperBlocked
NormalizedPositionX
OpponentBlocking
OpponentsHaveGoalkeeper
TeammateBlocking
alignedToBallandGoal
behindEverybody
canKick
closestToBall
distanceToBall<distanceToGoal
getBall
getBallR

- Sesión 1 - día 27 de abril
  - Completar el **tutorial** del editor
- Semana del 27 al 4
  - **Diseñar** el comportamiento para un equipo de SoccerBots usando máquinas de estado
  - RESTRICCIONES
    - El equipo tendrá **al menos 3 roles** (uno diseñado por cada miembro del grupo)
    - Los comportamientos básicos tendrán un máximo de 10 líneas de código
    - Tenéis que entregar el comportamiento en papel al comienzo de la clase del día 4
    - No hace falta poner el código de los comportamientos básicos, pero tenéis que explicar bien qué debe hacer cada uno

- Sesión 2 - día 4 de mayo
  - **Implementar el comportamiento** del equipo que habéis diseñado usando el editor
  - 10 minutos antes de terminar la clase tendréis que **rellenar una encuesta** de uso del editor que colgaremos en el campus
  - Al final de la clase tenéis que **entregar por librero** los siguientes ficheros, que se encuentran dentro de la carpeta eCo:
    - El fichero **eCo-file.log.xml**
    - La carpeta **Repositorios** con los comportamientos
  - En los siguientes días celebraremos un **campeonato** con los equipos que habéis entregado