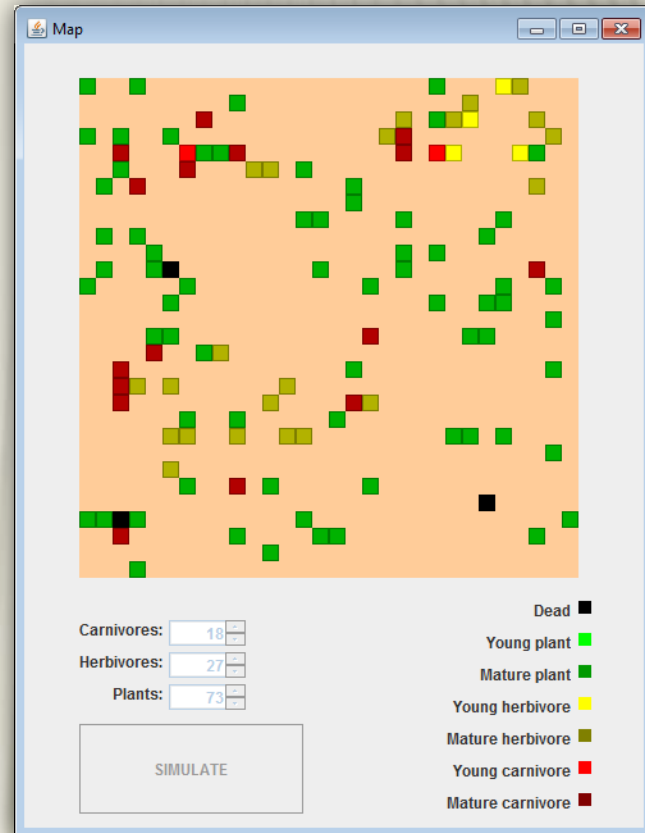


Tarea individual

# ECOSISTEMA



Iker Boyra Sarachaga

# DESCRIPCIÓN DEL ESCENARIO

---

- ✖ El escenario trata de un ecosistema en el cual interactúan plantas, herbívoros y carnívoros. Sus principales objetivos son sobrevivir y reproducirse.
- ✖ Con el tiempo las plantas, herbívoros y carnívoros maduran y tras otro periodo de tiempo mueren por vejez.
- ✖ Para vivir necesitan energía. Con el paso del tiempo van perdiendo energía. Al alimentarse obtienen energía. Al reproducirse consumen energía. Si se quedan sin energía mueren.
- ✖ Los carnívoros se alimentan de herbívoros. Los herbívoros se alimentan de plantas. Las plantas se alimentan de los seres muertos.
- ✖ Para reproducirse antes han de haber madurado. Los carnívoros y herbívoros necesitan una pareja dispuesta también a reproducirse. Las plantas no necesitan pareja pero necesitan que sea primavera.

# LOS AGENTES

---

- × **MapAgent:**  
(Genera el entorno gráfico y controla el llegar de la primavera)
- × **LivingAgent:**  
(Agente abstracto, que se extiende a otros tres tipos de agente)
  - + **PlantAgent:**  
(Controla el comportamiento de una planta)
  - + **HerbivoreAgent:**  
(Controla el comportamiento de un herbívoro)
  - + **CarnivoreAgent:**  
(Controla el comportamiento de un carnívoro)
- × **SenderAgent:**  
(Agente auxiliar que informa a MapAgent del inicio de la simulación)

# COMPORTAMIENTOS (MAPAGENT)

---

## × BlockBehaviour (OneShotBehaviour)

- + Bloquea al agente hasta recibir un mensaje INFORM “Start”

## × NotifySpringBehaviour (TickerBehaviour)

- + Envía un mensaje INFORM “Spring” a todos los PlantAgent

# COMPORTAMIENTOS (PLANTAGENT)

- × Comportamiento complejo creado mediante SequenceBehaviours y ParallelBehaviour

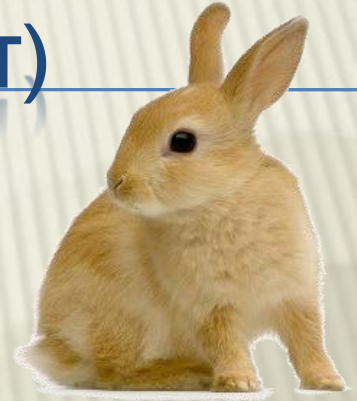


PlantGetMatureBehaviour	PlantGetOldBehaviour	DeadBehaviour
	PlantReproducingBehaviour	
EnergyLossBehaviour		
PlantGetFoodBehaviour		
BeEatenBehaviour		



# COMPORTAMIENTOS (HERBIVOREAGENT)

- × Comportamiento complejo creado mediante SequenceBehaviours y ParallelBehaviour



AnimalGetMatureBehaviour	AnimalGetOldBehaviour	DeadBehaviour
	AnimalReproducingBehaviour	
EnergyLossBehaviour		
MoveBehaviour		
BeEatenBehaviour		

# COMPORTAMIENTOS (CARNIVOREAGENT)

- × Comportamiento complejo creado mediante SequenceBehaviours y ParallelBehaviour



AnimalGetMatureBehaviour	AnimalGetOldBehaviour	DeadBehaviour
	AnimalReproducingBehaviour	
EnergyLossBehaviour		
MoveBehaviour		

# COMPORTAMIENTOS (WAKERBEHAVIOUR)

---

## ✕ PlantGetMatureBehaviour

- + Convierte al agente en adulto

## ✕ PlantGetOldBehaviour

- + Mata al agente por vejez

## ✕ AnimalGetMatureBehaviour

- + Convierte al agente en adulto

## ✕ AnimalGetOldBehaviour

- + Mata al agente por vejez



# COMPORTAMIENTOS (TICKERBEHAVIOUR)

## × PlantGetFoodBehaviour

- + Obtiene energía si hay agentes muertos cerca.

## × MoveBehaviour

- + Mueve al agente hacía una pareja o hacia la comida cercana.
- + Si está junto a la pareja y tiene energía suficiente para reproducirse le envía un mensaje REQUEST “Reproduction”
- + Si está junto a la comida, obtiene energía y le envía un mensaje INFORM “Eaten”.
- + Si no tiene pareja o comida cercana, se mueve en busca de alguna.

## × EnergyLoosBehaviour\*

- + Hace perder energía al agente. Si no le queda energía lo mata (\*y termina)

# COMPORTAMIENTOS (CYCLIC BEHAVIOUR)

## × PlantReproductionBehaviour

- + Si recibe un mensaje INFORM “Spring” y tiene suficiente energía, se reproduce.

## × AnimalReproductionBehaviour

- + Si recibe un mensaje REQUEST “Reproduction”:
  - × Si no tiene suficiente energía para reproducirse responde con un mensaje REJECT\_REQUEST “Reproduction”
  - × Si tiene suficiente energía responde con un mensaje ACCEPT\_REQUEST “Reproduction” y pierde energía por reproducirse.
- + Si recibe un mensaje ACCEPT\_REQUEST “Reproduction”:
  - × Si tiene suficiente energía para reproducirse genera un nuevo agente y pierde energía por reproducirse.
- + Si recibe un mensaje REJECT\_REQUEST “Reproduction”:
  - × Dice: “Tu te lo pierdes. Tal vez en otro momento”.

## × BeEatenBehaviour\*

- + Si recibe un mensaje INFORM “Eaten”, mata al agente (\*y termina)

# COMPORTAMIENTOS (3STEPBEHAVIOUR)

---

## × DeadBehaviour

### + Primer paso:

- × Dice: “Mi cuerpo empieza a pudrirse” y espera un tiempo.

### + Segundo paso:

- × Dice: “Mi cuerpo es pasto de las plantas” y espera un tiempo.

### + Tercer paso:

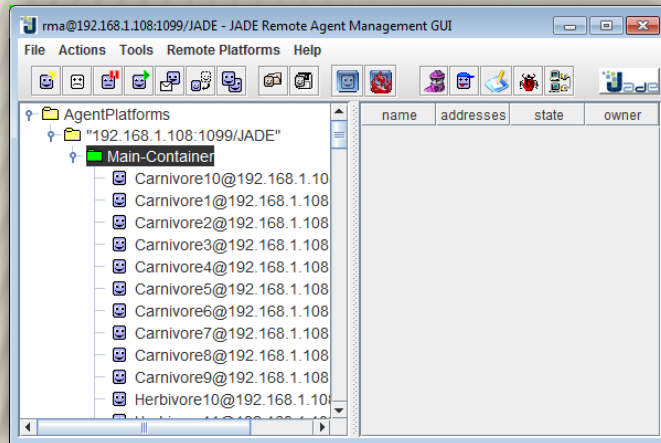
- × Dice: “Mi cuerpo ha dejado de existir” y elimina al agente.

# COMUNICACIONES Y MENSAJES

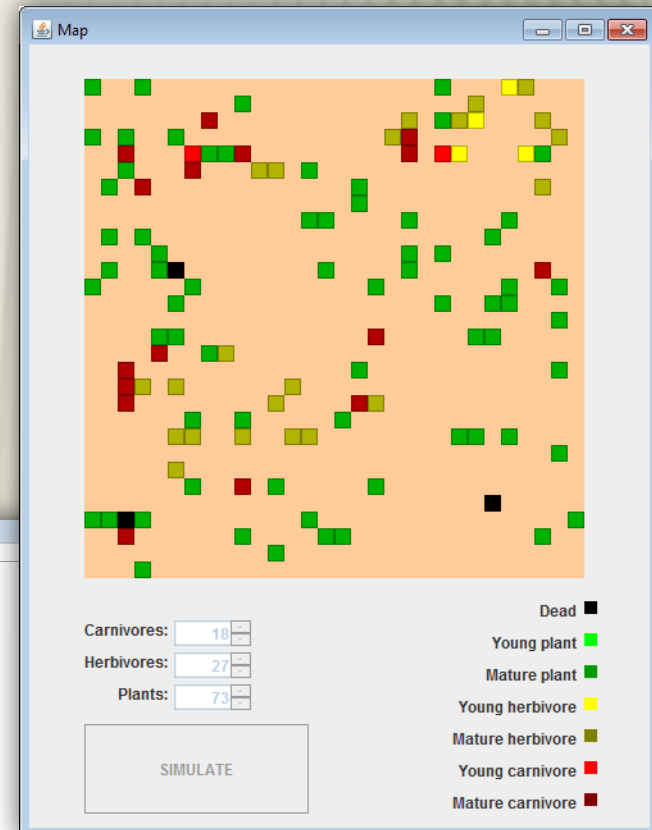
---

- × Informar del comienzo de simulación
  - + INFORM “Start”
- × Informar de la llegada de la primavera
  - + INFORM “Spring”
- × Informar a una presa que ha sido comida
  - + INFORM “Eaten”
- × Petición para reproducción
  - + REQUEST “Reproduction”
  - + ACCEPT\_REQUEST “Reproduction”
  - + REJECT\_REQUEST “Reproduction”

# EJECUCIÓN



```
Problems Javadoc Declaration Console
Ecosistema [Java Application] C:\Program Files\Java\jre1.8.0_51\bin\javaw.exe (8/11/2015 16:54:16)
Plant138: Me han comido!!
Plant138: Mi cuerpo empieza a pudrirse
Herbivore48: Ya soy adulto...
Herbivore10: Pareces apetitoso Plant133. Creo que voy a comerte
Herbivore41: Ya soy adulto...
Herbivore3: ¿Quieres pasar un buen rato Herbivore17?
Herbivore27: Estoy muy hambrieto/a para eso!
Herbivore2: ¿Quieres pasar un buen rato Herbivore54?
Plant20: Me han comido!!
Plant92: Me han comido!!
Plant20: Mi cuerpo empieza a pudrirse
Plant92: Mi cuerpo empieza a pudrirse
Herbivore45: Pareces apetitoso Plant102. Creo que voy a comerte
Herbivore45: Ya soy adulto...
Herbivore28: Pareces un buen partido, adelante.
Herbivore20: ¿Quieres pasar un buen rato Herbivore17?
Carnivore6: Pareces apetitoso Herbivore27. Creo que voy a comerte
Herbivore42: ¿Quieres pasar un buen rato Herbivore40?
Herbivore42: Pareces un buen partido, adelante.
Herbivore53: Ya soy adulto...
Herbivore40: ¿Quieres pasar un buen rato Herbivore42?
Herbivore40: Pareces un buen partido, adelante.
Plant135: Mi cuerpo es pasto de las plantas
Plant88: Mi cuerpo es pasto de las plantas
Herbivore24: Mi cuerpo ha dejado de existir
```





# CONCLUSIONES

---

- × **SequenceBehaviour** y **ParallelBehaviour** ofrecen una manera muy sencilla de crear comportamientos muy complejos a base de comportamientos simples e independientes.

# GRACIAS

---

¿¿¿Preguntas???