

## ACTIVITAT AVALUABLE

Programació  
CFGS DAW

Autors:

Joan V. Cassany – [jv.cassanycoscolla@edu.gva.es](mailto:jv.cassanycoscolla@edu.gva.es)

Guillermo Garrido – [g.garridoportes@edu.gva.es](mailto:g.garridoportes@edu.gva.es)

2022/2023

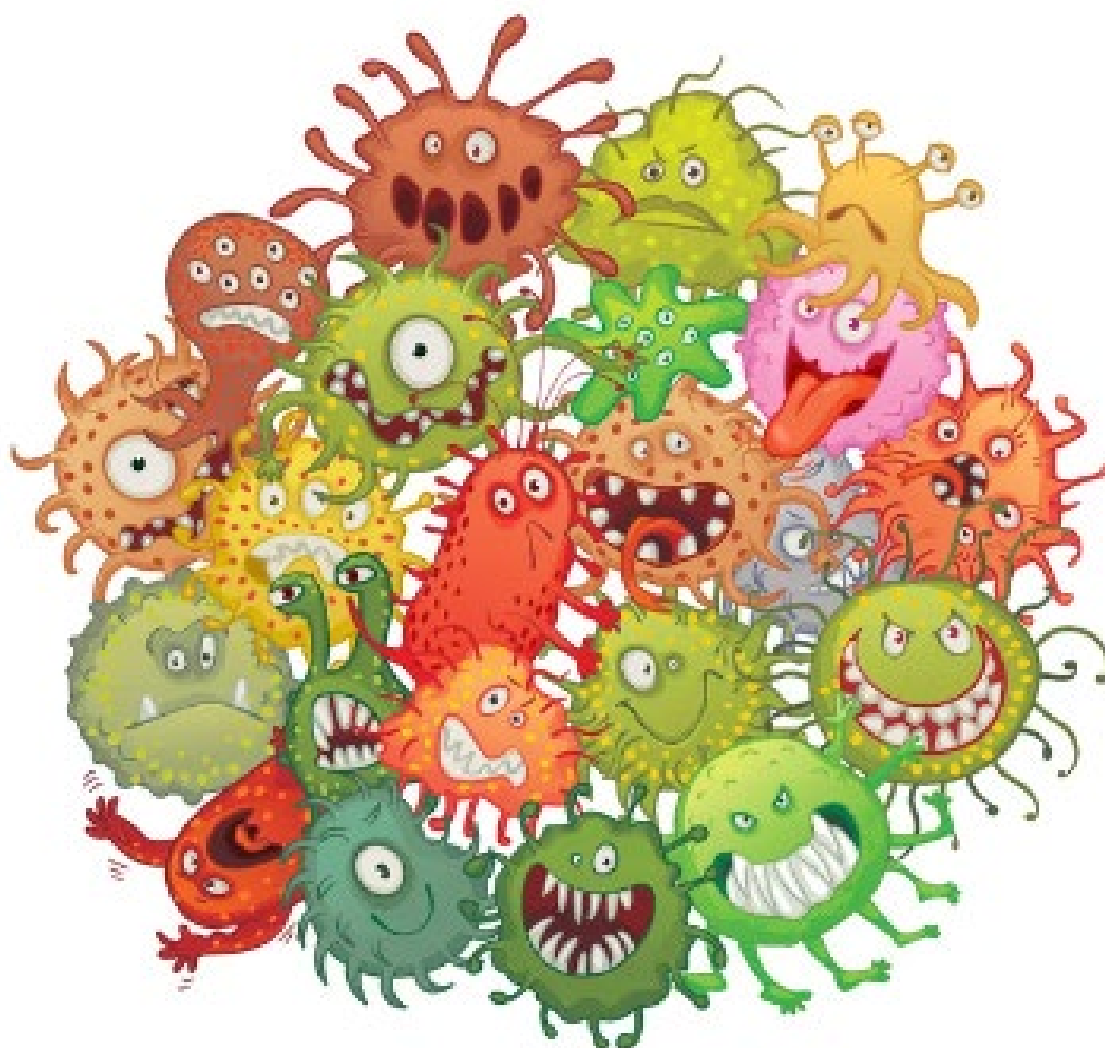
### Llicència



**[CC BY-NC-SA 3.0 ES](https://creativecommons.org/licenses/by-nc-sa/3.0/es/) Reconeixement – No Comercial – Compartir Igual (by-nc-sa)**

No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. Aquesta és una obra derivada de l'obra original de Carlos Cacho i Raquel Torres.

# ‘LA VIDA DELS MICROORGANISMES’



## 1. INTRODUCCIÓ

L'objectiu d'aquest exercici consisteix principalment en fer-vos utilitzar el conjunt de conceptes apresos, tant en la programació orientada a objectes com en la gestió d'excepcions. És per açò, que en alguns casos es pot pensar que alguns elements dels que es demanen no són estrictament necessaris o que es poden aplicar solucions més simples, però amb això no aconseguiríem l'objectiu i competències que amb aquest exercici es persegueixen.

Per tant, en aquest exercici es donen especificacions molt concretes (sense indicar la solució) per a que s'utilitzen les instruccions i estructures Java adequades. És molt important que les seguïu ja que la rubrica contemplarà si es fa ús d'aquestes o no.

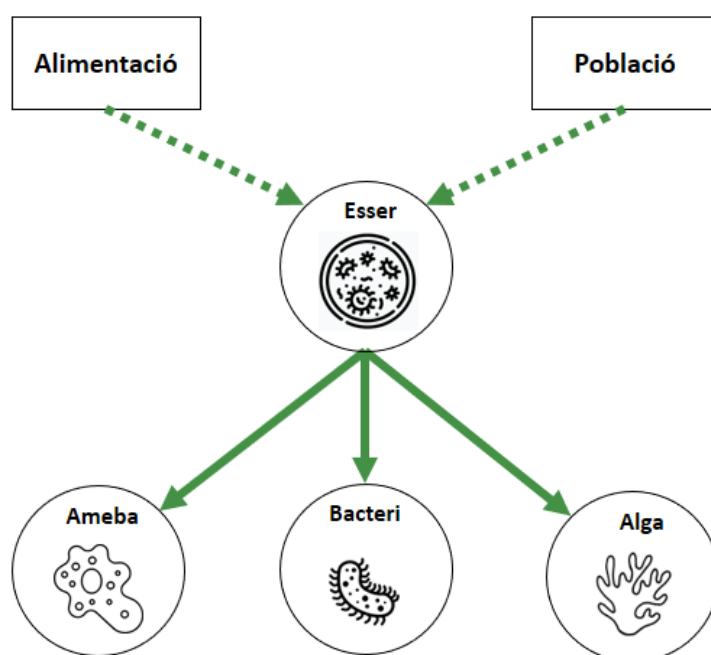
Igualment, donada la complexitat de les dependències, és important que vos ajusteu a la nomenclatura específica que es proporciona al llarg de l'exercici.

## 2. DESCRIPCIÓ DEL JOC: LA VIDA DELS MICROORGANISMES

En aquest projecte anem a construir un ecosistema d'**essers vius o microorganismes** que interactuaran entre ells per a alimentar-se i per a reproduir-se. Tots els microorganismes pertanyen a un grup comú d'essers que estableix un conjunt d'elements comuns, que poden vindre definits o deixar en mans dels diferents grups d'esses la sua definició.

Així doncs, el nostre ecosistema de microorganismes o Essers estarà format per Amebes, Bacteris i Algues. A més existiran uns nutrients dels que s'alimentarà la base de la cadena per anar fer engreixant al nostre ecosistema.

Tota la gestió d'aquest ecosistema es farà des de un programa principal amb el que l'usuari interactuarà per consultar o provocar canvis en aquest.



### 3. ESTRUCTURA BASE

Definirem dos esquemes des de on s'establiran determinats atributs i mètodes que la resta de components hauran d'implementar. D'aquesta manera podrem variar el comportament de l'ecosistema de forma fàcil simplement canviant alguns valors.

#### 3.1. ALIMENT

Limitarà a que el projecte sols puga utilitzar els següents aliments: Tot, Ameba, Bacteri, Alga, Nutrient.

#### 3.2. ALIMENTACIÓ

Es definiran les constants dels pesos que s'utilitzaran en el procés de creació dels diferents tipus d'essers:

- Pes de la ameba=20
- Pes de el bacteri=10
- Pes de l'alga=3
- Pes dels nutrients=5

També, indicarà els mètodes que tots els essers hauran d'implementar quan es programen:

- **menjar(...)**: on cada tipus d'esser especificarà com porta a terme l'alimentació (a qui es menja i com). Per tant, serà necessari tindre informació d'entrada al mètode sobre la llista completa de essers.

#### 3.3. POBLACIÓ

Es definirà una constant multiplicadora per al pes de reproducció i una altra per al màxim d'essers de cada tipus que existirà.

- Pes reproducció=3
- Número màxim=20

També, indicarà els mètodes que tots els essers hauran d'implementar quan es programen:

- **reduirPoblacio()**, que tindrà com a finalitat que cada tipus d'esser quan siga menjat reduísca el seu propi nombre total d'individus i el general.

- **reproduir(...)**, que s'utilitzarà per a que cada tipus d'organismes definisca el seu mètode de reproducció i puga incloure un nou esser del seu tipus en la llista de essers, per la qual cosa aquesta formarà part de les dades d'entrada al mètode.

## 4. LA SUPERCLASSE: ESSER

Serà la superclasse a partir de la qual es definirà la resta. Adoptarà els esquemes que s'hagen definit en el projecte, tant per a l'alimentació com per a la reproducció.

Aquesta superclasse es podrà heretar però no instanciar. En ella s'establirà la base de les definicions del comportament que tindran els essers. En alguns casos ho definirà completament, en altres obligarà a que les subclasses implementen eixos comportaments.

### 4.1. ATRIBUTS DE CLASSE

Es definiran aquests atributs.

- **totalEssers**, que s'incrementarà quan es cree algun nou microorganisme i es reduirà quan desaparega. S'ha de tindre en compte que la desaparició d'un esser haurà de reduir tant el seu comptador (que es veurà en el seu moment) com aquest total de essers.
- **consecutiu**, que s'anirà incrementant a cada nou microorganisme que es cree i que s'utilitzarà per a que forme part del text que s'assignarà al nom (atribut que es cita després). Exemple: *AMEBA1*, *ALGA2*, *ALGA3*, *BACTERIA4*, etc.

Aquests valors sols es poden modificar o accedir des de la pròpia classe.

### 4.2. ATRIBUTS D'INSTÀNCIA

Es definirà els següents atributs d'instància.

- **nom** (String): text descriptor que no es podrà modificar una vegada assignat.
- **pes** (enter): que podrà variar en qualsevol moment (al menjar o reproduir-se).

Aquests atributs sols podran ser acceditats o modificats des de aquesta classe Essers.

### 4.3. CONSTRUCTOR

- El **nom** es formarà per la concatenació d'un text que vindrà des de les subclasses (segons el tipus d'esser que corresponga) més el consecutiu d'essers (per exemple *AMEBA1*).
- El **pes** s'haurà d'incrementar en la quantitat que vinga indicada segons el tipus d'esser que s'haja creat.

No s'haurà d'oblidar en aquest punt actualitzar els atributs de classe que corresponga.

## 4.4. MÈTODES

### 4.4.1. MÈTODES QUE NO PODRAN SER SOBREESCRITS PER LES SUBCLASSES

- String **dirNom()**: ens donarà el nom.
- int **dirPes()**: ens donarà el pes.
- **canviaPes()**: incrementarà (o decrementarà) el pes de l'objecte segons el valor que li arribi.

### 4.4.2. MÈTODES QUE NO PODRAN SER SOBREESCRITS PER LES SUBCLASSES I QUE SOLS ES PODEN UTILITZAR A TRAVÉS DE LA CLASSE

- int **GeneraAleatori ( inicial, quantitat)**: generarà un número aleatori enter. Aquest mètode es podrà utilitzar des de qualsevol altra classe (ja siga subclasse o programa principal). S'utilitzarà per a generar valors aleatoris depenent de la funcionalitat que s'estiga implementant. El procés que crida a aquest mètode indicarà el valor inicial i la quantitat de valors que necessita per a obtenir un valor aleatori.

*Nota: Encara que aquesta classe no siga el millor lloc per a ubicar aquest mètode, ho farem així per qüestions didàctiques a l'hora de fer les cridades.*

### 4.4.3. MÈTODES QUE S'IMPLEMENTARAN EN TOTES LES CLASSES I QUE MOSTRARAN INFORMACIÓ DE LA CLASSE NO DE LA INSTÀNCIA

Per tant, aquests mètodes no han de permetre la sobreescritura al declarar-se.

- int **dirPoblacio()**: ens indicarà el total de individus d'aquesta classe.

### 4.4.4. MÈTODES QUE S'OBLIGARÀ A LES SUBCLASSES A IMPLEMENTAR

Per tant, no es requereix una definició detallada del seu comportament en aquest punt.

- String **mostrarEstat()**: mostrarà informació resumida de l'objecte.
- String **mostrarDetall()**: mostrarà informació detallada de l'objecte.

### 4.4.5. MÈTODES QUE TOTA CLASSE HA DE CONTINDRE

Ací s'inclouran tots els mètodes que ens trobem obligats a crear per definició.

Sols s'especificarà el comportament de aquells mètodes que tinga sentit a aquest nivell de classes.

## 5. LA CLASSE: AMEBA

Heretarà de la superclasse. Podrà crear instàncies però no permetrà la creació de subclasses.

### 5.1. ATRIBUTS DE CLASSE

Es definiran aquests atributs.

- **totalAmebes** que s'incrementarà quan es cree una nova i es reduirà quan desaparega. S'ha de tindre en compte que la desaparició d'una ameba haurà de reduir tant el seu acumulador com el general.

Aquests valors sols es poden modificar o accedir des de la pròpia classe.

### 5.2. ATRIBUTS D'INSTÀNCIA

A més dels atributs heretats, tindrà dos atributs més.

- **aliment** (Aliment): indicarà quin és el tipus d'alimentació que té.
- **reproduccio** (boolea): que ens indicarà si pot reproduir-se.

Aquests atributs sols podran ser accedit o modificats des de aquesta classe Ameba.

### 5.3. CONSTRUCTOR

- el **nom** de les amebes començarà amb la paraula «AMEBA».
- el **pes** de les amebes serà el que s'ha determinat com a constant.
- **aliment**, les amebes s'ho mengen tot.
- **reproduccio**, les amebes es reproduïxen.

No s'haurà d'oblidar en aquest punt actualitzar els atributs de classe que corresponga.

## 5.4.MÈTODES

### 5.4.1. MÈTODES QUE S'IMPLEMENTARAN EN TOTES LES CLASSES I QUE MOSTRARAN INFORMACIÓ DE LA CLASSE NO DE LA INSTÀNCIA

Per tant, aquests mètodes no han de permetre la sobreescritura al declarar-se.

- int **dirPoblacio()**: ens indicarà el total de individus d'aquesta classe.

### 5.4.2. MÈTODES HERETATS QUE S'HAN DE DEFINIR

- String **mostrarEstat()**: per a les amebes es mostrarà un missatge amb el següent text

"@ " + nom + " => PES: " + pes

- String **mostrarDetall()**: es mostrarà un missatge amb el següent text

"@ " + nom + " => PES " + pes + " - ALIMENTACIÓ: " + aliment + " - REPRODUCCIÓ: " + "SI" o "NO"

- **menjar(...)**

- Triarà aleatòriament un ésser qualsevol de la llista que no siga ella mateixa.
- Incrementarà el seu pes amb el pes de l'ésser que s'ha menjat.
- Informarà de qui és la víctima i a qui s'ha menjat amb el text  
"ALIMENTACIÓ \*\*\*\* " + nom + ": m'he menjat a " + nom víctima + ". Ara pese " + pes
- Reduirà els totals de població que corresponga.
- Eliminarà de la llista a la víctima.

- **reproduir(...)**

- Si el seu pes és major o igual al pes base de l'ameba multiplicat pel pes de reproducció es portarà a terme la reproducció.
- Crearà una nova ameba.
- Reduirà el seu pes amb el que s'ha quedat la nova ameba.
- Ens informarà amb el text  
"REPRODUCCIÓ \*\*\*\* " + nom + " m'he reproduït i he creat a " + nom nova + ". Ara pese " + pes
- Si no té el pes adequat, simplement ens dirà que no pot reproduir-se encara amb el text  
"REPRODUCCIÓ \*\*\*\* " + nom + " amb un pes de " + pes + " no em puc reproduir") (1)

**(1) NOTA:** Aquest últim cas es pot resoldre per missatge o per excepció. La puntuació serà diferent segons s'opte per una solució o l'altra. Reviseu la rúbrica.



## 6. LA CLASSE: BACTERI

Heretarà de la superclasse. Podrà crear instàncies però no permetrà la creació de subclasses.

### 6.1. ATRIBUTS DE CLASSE

Es definiran aquests atributs.

- **totalBacteris** que s'incrementarà quan es cree un nou i es reduirà quan desaparega. S'ha de tindre en compte que la desaparició d'un bacteri haurà de reduir tant el seu acumulador com el general.

Aquests valors sols es poden modificar o accedir des de la pròpia classe.

### 6.2. ATRIBUTS D'INSTÀNCIA

A més dels atributs heretats, tindrà dos atributs més.

- **aliment** (Aliment): indicarà quin és el tipus d'alimentació que té
- **reproduccio** (boolea): que ens indicarà si pot reproduir-se

Aquests atributs sols podran ser accedit o modificats des de aquesta classe Bacteri.

### 6.3. CONSTRUCTOR

- el **nom** de les amebes començarà amb la paraula «BACTERI».
- el **pes** dels bacteris serà el que s'ha determinat com a constant.
- **aliment**, els bacteris mengen algues.
- **reproduccio**, els bacteris es reproduïxen.

No s'haurà d'oblidar en aquest punt actualitzar els atributs de classe que corresponga.

## 6.4.MÈTODES

### 6.4.1. MÈTODES QUE S'IMPLEMENTARAN EN TOTES LES CLASSES I QUE MOSTRARAN INFORMACIÓ DE LA CLASSE NO DE LA INSTÀNCIA

Per tant, aquests mètodes no han de permetre la sobreescritura al declarar-se.

- int **dirPoblacio()**: ens indicarà el total de individus d'aquesta classe.

### 6.4.2. MÈTODES HERETATS QUE S'HAN DE DEFINIR

- String **mostrarEstat()**: per a els bacteris es mostrarà un missatge amb el següent text  
"/ " + nom + " => PES: " + pes

- String **mostrarDetall()**: es mostrarà un missatge amb el següent text  
"/ " + nom + " => PES " + pes + " - ALIMENTACIÓ: " + aliment + " - REPRODUCCIÓ: " + "SI" o "NO"

- **menjar(...)**

- Els bacteris sols mengen algues.
  - Si no hi en queden donarà un missatge d'avis. (2)
  - Si en queden triarà un alga aleatòriament de la llista.
- Incrementarà el seu pes amb el pes de l'esser que s'ha menjat.
- Informarà de qui és la víctima i a qui s'ha menjat amb el text  
"ALIMENTACIÓ \*\*\*\* " + nom + ": m'he menjat a " + nom víctima + ". Ara pese " + pes
- Reduirà els totals de població que corresponga.
- Eliminarà de la llista a la víctima.

**(2) NOTA:** Aquest últim cas es pot resoldre per missatge o per excepció. La puntuació serà diferent segons s'opte per una solució o l'altra. Reviseu la rúbrica.

- **reproduir(...)**

- Els bacteris es divideixen en dos fills d'igual mida.
- Si el seu pes és major o igual al pes de reproducció multiplicat pel doble del pes base de el bacteri es portarà a terme la reproducció.
- Crearà un nou bacteri amb la meitat del seu pes.
- Reduirà el seu pes amb el que s'ha quedat el nou bacteri.
- Ens informarà amb el text  
"REPRODUCCIÓ \*\*\*\* " + nom + " m'he reproduït i he creat a " + nom nou + ". Ara pese " + pes
- Si no te el pes adequat, simplement ens dirà que no pot reproduir-se encara amb el text  
"REPRODUCCIÓ \*\*\*\* " + nom + " amb un pes de " + pes + " no em puc reproduir") (3)

**(3) NOTA:** Aquest últim cas es pot resoldre per missatge o per excepció. La puntuació serà diferent segons s'opte per una solució o l'altra. Reviseu la rúbrica.

## 7. LA CLASSE: ALGA

Heretarà de la superclasse. Podrà crear instàncies però no permetrà la creació de subclasses.

### 7.1. ATRIBUTS DE CLASSE

Es definiran aquests atributs.

- **totalAlgues** que s'incrementarà quan es cree una nova i es reduirà quan desaparega. S'ha de tindre en compte que la desaparició d'una alga haurà de reduir tant el seu acumulador com el general.

Aquests valors sols es poden modificar o accedir des de la pròpia classe.

### 7.2. ATRIBUTS D'INSTÀNCIA

A més dels atributs heretats, tindrà dos atributs més.

- **aliment** (Aliment): indicarà quin és el tipus d'alimentació que té
- **reproduccio** (boolea): que ens indicarà si pot reproduir-se

Aquests atributs sols podran ser accedit o modificats des de aquesta classe Alga.

### 7.3. CONSTRUCTOR

- el **nom** de les amebes començarà amb la paraula «BACTERI».
- el **pes** de les algues serà el que s'ha determinat com a constant.
- **aliment**, les algues mengen nutrients.
- **reproduccio**, les algues es reproduïxen.

No s'haurà d'oblidar en aquest punt actualitzar els atributs de classe que corresponga.

## 7.4.MÈTODES

### 7.4.1. MÈTODES QUE S'IMPLEMENTARAN EN TOTES LES CLASSES I QUE MOSTRARAN INFORMACIÓ DE LA CLASSE NO DE LA INSTÀNCIA

Per tant, aquests mètodes no han de permetre la sobreescritura al declarar-se.

- int **dirPoblacio()**: ens indicarà el total de individus d'aquesta classe.

### 7.4.2. MÈTODES HERETATS QUE S'HAN DE DEFINIR

- String **mostrarEstat()**: per a les algues es mostrarà un missatge amb el següent text

"# " + nom + " => PES: " + pes

- String **mostrarDetall()**: es mostrarà un missatge amb el següent text

"# " + nom + " => PES " + pes + " - ALIMENTACIÓ: " + aliment + " - REPRODUCCIÓ: " + "SI" o "NO"

- **menjar(...)**

- Les algues sols mengen nutrients. Bàsicament el que es fa és incorporar el pes d'un nutrient al pes de l'alga.
- Incrementarà el seu pes amb el pes constant del nutrient que s'ha menjat.
- Informarà de qui és la víctima i a qui s'ha menjat amb el text "ALIMENTACIÓ \*\*\*\*\* " + nom + ": m'he menjat a " + nom víctima + ". Ara pese " + pes
- Si el pes actual de l'alga es major que el doble del pes base de l'alga per la constant pes de reproducció es reproduirà.

- **reproduir(...)**

Les algues es poden reproduir més d'una vegada en funció del pes que tinguen. Per tant, generarà algues (farà totes les accions del procés de reproducció) mentre el seu pes siga major o igual al pes base de l'alga per la constant del pes de reproducció.

- Crearà una nova alga.
- Reduirà el seu pes amb pes base de la nova alga creada.
- Ens informarà amb el text "REPRODUCCIÓ \*\*\*\*\* " + nom + " m'he reproduït i he creat a " + nom nou + ". Ara pese " + pes
- Si no te el pes adequat, simplement ens dirà que no pot reproduir-se encara amb el text "REPRODUCCIÓ \*\*\*\*\* " + nom + " amb un pes de " + pes + " no em puc reproduir") (4)

**(4) NOTA:** Aquest últim cas es pot resoldre per missatge o per excepció. La puntuació serà diferent segons s'opte per una solució o l'altra. Reviseu la rúbrica.

## 8. PROGRAMA PRINCIPAL: LAVIDADELSMICROORGANISMES

### 8.1. CREACIÓ DE L'ECOSISTEMA

Es crearà una llista d'essers. Tingueu en compte el tipus que ha de tindre la llista per a poder incorporar objectes de totes les classes.

Es demanarà a l'usuari el nombre d'essers que vol tindre de cada tipus:

- Introdueix el nombre d'amebes (entre 1 i (constant número màxim d'essers) );
- Introdueix el mínim de bacteris (entre 1 i (constant número màxim d'essers) );
- Introdueix el mínim d'algues (entre 1 i (constant número màxim d'essers) );

### 8.2. MÈTODE llegirNumero()

Per a llegir cadascun d'aquests valors es crearà la funció **llegirNumero(...)** que valide mitjançant un **context d'excepcions** que s'introduïska una dada numèrica i que estiga entre els límits indicats. La funció sols finalitzarà quan la dada siga correcta.

Si el número no és correcte donarà el missatge:

"ERROR: Ha d'introduir un número entre " + min + " i " + max + "."

Si no s'ha introduït un valor numèric el missatge serà:

"ERROR: No s'ha introduït un valor numèric."

### 8.3. MÈTODE crearEssers()

Una vegada tinguem les quantitats de cada esser a crear, invocarem procediment **crearEssers (...)** que, a partir de la llista buida d'essers i dels valors indicats per l'usuari crearà de forma aleatòria els essers de cada tipus que corresponga.

Es plantegen dues alternatives per a crear la llista.

Alternativa 1: Els esser s'introduiran en la llista de forma contigua (primer tots els d'un tipus, després altre, etc.).

Alternativa 2: Els esser s'introduiran en la llista de forma aleatòria fent us de la funció **generaAleatori()** utilitzant 3 valors per a triar en cada moment quin tipus d'esser s'insereix.

**Nota:** Les dues opcions seran vàlides, però **l'alternativa 2 valdrà 1 PUNT EXTRA** per la dificultat que comporta.

## 8.4.MÈTODE processaMenu()

Una vegada ja tenim creada la llista inicial d'essers donarem pas a la interacció amb l'usuari mitjançant un menú que gestionarem en el mètode processaMenu(...).

Es mostrarà a l'usuari un missatge amb les opcions que pot triar.

«OPCIONES==> 1.-Una Interacció, 2.-Deu Interaccions, 3.-Llistat, 4-Detall 0.-Eixir: «

Es validarà la informació introduïda utilitzant la funció ja vista **llegirNumero(...)**.

Opció 1: es cridarà al mètode **produeixInteraccio (..)** que veurem a continuació per a provocar una interacció entre els essers que tenim.

Opció 2: es cridarà al mètode **produeixInteraccio (..)** que veurem a continuació 10 vegades per a provocar una evolució ràpida dels microorganismes.

Opció 3: es mostrarà un llistat del estat en que es troba actualment l'ecosistema fent ús del mètode **mostraLlistaEssers (...)** que veurem a continuació.

Opció 4: es mostrarà el detall de l'esser que trie l'usuari fent ús del mètode **mostrarEsser(...)** que veurem a continuació.

Opció 0: es finalitzarà el bucle i anirem al final del programa per a mostrar la llista d'essers que queda fent ús del mètode **mostraLlistaEssers (...)** y donar un missatge d'acomiadament.

## 8.5.MÈTODE mostraLlistaEssers()

Recorrerà la llista d'essers mostrant-nos informació de l'estat de cadascun d'ells.

Finalment, donarà un missatge resum del estat actual del conjunt de microorganismes que tenim:

"POBLACIÓ: TOTAL ESSERS=>" + total d'esses + ", AMEBES=>" + total d'amebes + ", BACTERIES=>" + total de bacteris + ", ALGUES=>" + total d'algues

## 8.6.MÈTODE mostraEsser()

Es proposen dues alternatives entre les que podeu triar.

Alternativa 1: es demanarà un valor numèric enter per a accedir per index a una posició de la llista i mostrar el detall d'aquest microorganisme. Es validarà que el número estiga en els límits correctes per a accedir a la llista fent ús d'un mètode ja conegut.

Alternativa 2: L'usuari introduirà un nom per teclat i es buscarà en la llista per a proporcionar la informació detalla d'aquest microorganisme. Si no existeix eixe nom en la llista es donarà un error.

«ERROR: el microorganisme» + nom + « no apareix en la llista.»

**Nota:** Les dues opcions seran vàlides, però l'**alternativa 2 valdrà 0,5 PUNT EXTRA** per la dificultat que comporta.

## 8.7.MÈTODE produeixInteracció()

Es triarà un esser de la llista a l'atzar utilitzant la funció **generaAleatori(...)**.

Si l'element triat és una instància d'una ameba o d'un bacteri podrà menjar o reproduir-se, per tant generarem un altre número aleatori per a que s'opte per una o altra opció.

Si l'element triat és una instància d'un alga, sols pot menjar. Recordeu que les algues es reproduïxen, si cal, quan mengen.

## 9. EXEMPLE D'EXECUCIÓ

Introdueix el nombre d'amebes (entre 1 i 20):

4

Introdueix el nombre de bacteris (entre 1 i 20):

3

Introdueix el nombre d'algues (entre 1 i 20):

5

OPCIONES==> 1.-Una Interacció, 2.-Deu Interaccions, 3.-Llistat, 4-Detall 0.-Eixir: 3

/ BACTERI1 => PES: 10

# ALGA2 => PES: 2

@ AMEBA3 => PES: 20

/ BACTERI4 => PES: 10

@ AMEBA5 => PES: 20

@ AMEBA6 => PES: 20

@ AMEBA7 => PES: 20

/ BACTERI8 => PES: 10

# ALGA9 => PES: 2

# ALGA10 => PES: 2

# ALGA11 => PES: 2

# ALGA12 => PES: 2

POBLACIÓ: TOTAL ESSERS=>12, AMEBES=>4, BACTERIS=>3, ALGUES=>5

OPCIONES==> 1.-Una Interacció, 2.-Deu Interaccions, 3.-Llistat, 4-Detall 0.-Eixir: 2

REPRODUCCIÓ \*\*\*\* AMEBA5 amb un pes de 20 no em puc reproduir

ALIMENTACIÓ \*\*\*\* ALGA10: m'he menjat un nutrient que pesa 3. Ara pese 5

REPRODUCCIÓ \*\*\*\* BACTERI8 amb un pes de 10 no em puc reproduir

ALIMENTACIÓ \*\*\*\* AMEBA5: m'he menjat a ALGA12. Ara pese 22

ALIMENTACIÓ \*\*\*\* BACTERI4: m'he menjat a ALGA10. Ara pese 15

ALIMENTACIÓ \*\*\*\* AMEBA7: m'he menjat a AMEBA5. Ara pese 42

ALIMENTACIÓ \*\*\*\* BACTERI1: m'he menjat a ALGA11. Ara pese 12

ALIMENTACIÓ \*\*\*\* BACTERI1: m'he menjat a ALGA9. Ara pese 14

ALIMENTACIÓ \*\*\*\* AMEBA3: m'he menjat a AMEBA6. Ara pese 40

ALIMENTACIÓ \*\*\*\* AMEBA3: m'he menjat a ALGA2. Ara pese 42

OPCIONES==> 1.-Una Interacció, 2.-Deu Interaccions, 3.-Llistat, 4-Detall 0.-Eixir: 3

/ BACTERI1 => PES: 14

@ AMEBA3 => PES: 42

/ BACTERI4 => PES: 15

@ AMEBA7 => PES: 42

/ BACTERI8 => PES: 10

POBLACIÓ: TOTAL ESSERS=>5, AMEBES=>2, BACTERIS=>3, ALGUES=>0

OPCIONES==> 1.-Una Interacció, 2.-Deu Interaccions, 3.-Llistat, 4-Detall 0.-Eixir:



## 10. RUBRICA

Classe Esser puntuarà 1 PUNT.

Classe Ameba puntuarà 1,5 PUNTS.

Classe Bacteri puntuarà 1,5 PUNTS.

Classe Alga puntuarà 1,5 PUNTS.

Programa Principal puntuarà 1 PUNT.

En cada classe s'assignarà 0,5 punts dels 1,5 punts indicats a la correcta declaració de les classes i els mètodes (herència, polimorfisme, privacitat, etc.)

En les subclasses s'assignarà 0,5 punts dels 1,5 punts indicats al correcte funcionament dels mètodes menjar i reproduir.

Amb les consideracions anteriors la nota màxima computada que es pot treure serà de 6,5.

Si se inclou la gestió d'excepcions es comptarà amb 0,5 PUNTS EXTRA per cada excepció (hi ha 4 punts on s'ha proposat l'opció d'incloure una excepció).

Si s'opta per l'alternativa 2 en la creació de la llista es comptarà amb 1 PUNT EXTRA.

Si s'opta per l'alternativa 2 al mostrar el detall d'un esser es comptarà amb 0,5 PUNTS EXTRA.

Recomanem iniciar l'exercici amb les alternatives més senzilles i quan es tinga el projecte ja funcionant, si es disposa de temps, es facen les modificacions necessàries per a incorporar les alternatives complexes i millorar la nota.

S'avaluarà també la inclusió adequada de comentaris, ordenació del codi, seguiment de la nomenclatura proposada i la programació amb un codi eficient.