



# PROJECTE DE PROGRAMACIÓ CURS 2024-25 : InfekTòpia

## LLIURAMENT INICIAL

Mireia Ferrer Ferrer (u1990149)  
Romà Barrera Moreno (u1989957)  
Iria Auladell Gonzalez (u1990128)

### VALORACIÓ:

- bona feina !  
- els principals errors ja  
assenyalats

NOTA: 4.5 sobre 5

TOTAL: 9.5 sobre 5

## 1. Proposta inicial de divisió en mòduls, en forma d'interfícies o classes Java.

### Enunciat

Cal donar una descripció general de cada mòdul i la capçalera de les seves operacions públiques (només les capçaleres, deixeu la seva implementació per a més endavant) junt amb la seva especificació pre-post.

Heu de proposar tots els mòduls i operacions públiques que cregueu necessaris. S'entén que es tracta d'una proposta inicial que més endavant podreu canviar si fa falta, però heu de procurar afinar-la al màxim des de bon principi.

El que no heu de fer de cap manera en aquest primer lliurament és afegir res de codi ni d'estructures de dades (atributs) (això correspondria al nivell d'implementació).

### Implementació

1. Main (Modul Funcional)
2. Simulació (classe)
3. Interacció (Modul Funcional)
4. Regió (classe)
5. Virus (classe abstracta)
6. VirusARN (classe)
7. VirusADN (classe)
8. FamiliaVirus (classe)
9. Mutacio (Modul Funcional)
10. AfectacióVirusRegió (classe)
11. Classes de Llegir Fitxers (classes)

## 1)MAIN

```
public class Main {
    public static void main(String[] args) {
        // Pre: -
        // Post:
        // - Inicia el programa i posa en marxa la simulació.
        // - Gestiona la lectura dels fitxers.
        // - Configura l'entorn de la simulació perquè pugui avançar
        dia a dia sense errors.
        // - Finalitza mostrant els resultats de l'evolució del
        virus a cada regió.
    }
}
```

---

## 2)SIMULACIÓ

```
//@class Simulacio mostra per pantalla les dades de l'evolucio.
public class Simulacio{

    /** @brief Mostra una sortida de text amb les dades de la regio
    @pre rv no buit.
    @post ---
    @param territori Tipus de territori del que es vol obtenir dades
    (regio o territori total).
    @param tipusDada Dia actual o acumulada.
    */

    public void mostrarEstat(char territori, char tipusDada){
    }

    /** @brief Mostra un grafic de l'historial de dades de l'evolucio
    @pre rv no buit.
    @post --
    @param territori Tipus de territori del que es vol obtenir dades
    (regio o territori total).
    @param tipusDada Dia actual o acumulada.
    */

    public void mostrarGrafic(char territori, char tipusDada){
    }
}
```

### 3)INTERACCIÓ

Hi ha mètodes q semblen de simulació

```
/** @file Interaccio.java
@brief Classe Interaccio
@author Roman Carriera, Miquel Ferrer i Lluís Auladell
*/
//@class Interaccio enllac entre aplicatiu i usuari.
public class Interaccio{
    /** @brief Permet a introduir els noms dels fitxers.
    @pre Fitxers existents
    @post --
    */
    public static fitxersSimulacio(){
    }

    /** @brief Permet a l'usuari introduir el tipus de confinament
    abans de avançar la simulacio una unitat de temps (dia)
    @pre confinament no buit.
    @post Estat de simulacio actual modificat
    */
    public static interaccio(char confinament){
    }

    /** @brief Permet a l'usuari avançar la simulacio una unitat de
    temps (dia)
    @pre --
    @post Estat de simulacio incrementat en unitat de temps
    */
    public static incrementarDia(){
    }

    /** @brief Permet a l'usuari aturar la simulacio.
    @pre --
    @post Simulacio aturada i resetejada
    */
    public static aturarSimulacio(){
    }
}
```

Se suposa que les dades  
es guarden en variables  
static?  
No acabo de veure clar  
aquest mòdul...

✓

, falten  
els tipus  
de retorn

#### 4)REGIÓ

```
import java.util.List;

public class Regio {

    /* La classe Regio
       * representa una àrea geogràfica on es poden estudiar els efectes
d'un virus.
       * Aquesta classe serveix per emmagatzemar informació sobre la
població d'una regió
       * i els contactes que tenen les persones entre elles, un factor
important en la propagació d'un virus.
    */

    // Mètodes
    public String nom();
    // Pre: La regió ha de tenir un nom vàlid.
    // Post: Retorna un string que és el nom de la regió.

    public int poblacio();
    // Pre: La regió ha de tenir un nombre d'habitants vàlid.
    // Post: Retorna un enter que és el nombre d'habitants.

    public void mostrarConfinament() {
        //Pre:-----
        //Post:Mostra en pantalla les regions en confinament
    }

    // Operacions
    public void afegirCasos(Virus virus, int nousInfectats) {
        // Afegeix nous casos d'un virus a la regió
    }

    public void afegirConfinament(String confinament) {
        //Pre:
        //Post:Aplica un confinament a la regió
    }

    public void aixecarConfinament() {

        //Pre:
```

✓

✓

X

discut.  
,

cal q  
en  
parlem

```

        //Post:Elimina el confinament a una regió
    }
}

```

## 5) VIRUS

```

/**
 * @class Virus
 * @brief Implementem una classe abstracta per definir el comportament general
dels virus.
 * @details Aquesta classe serveix com a base per a diferents tipus de virus (ADN
i ARN).
 * En aquesta hi ha una sèrie de mètodes sense implementar-les directament.
 * Una classe abstracta no es pot instanciar directament i només proporciona
una estructura que altres classes han d'implementar.
 *
 * @author Roman Barrera, Mireia Ferrer i Iria Auladell
 * @version 2025.3.13
 */

// Com que tots els virus comparteixen atributs comuns, hem decidit fer una
classe abstracte de manera que
// d'ella heretin dos tipus de Virus, els que muten (ARN) i els que no muten
(ADN)
public abstract class Virus {

    /**
     * Primer és important fer el constructor de la classe Virus.
     * @param nom Nom del virus.
     * @param familia Família a la qual pertany el virus (de tipus FamiliaVirus).
     * @param pMal Probabilitat que una persona infectada desenvolupi la malaltia.
     * @param tInc Temps d'incubació
     * @param tLat Temps de latència
     * @param pMor Taxa de mortalitat
     * @param tCon Durada mitjana del període de contagi
     * @param pCon Taxa de contagi
     * @param tImm Durada del període d'immunitat
     */

    // Hem posat probabilitats com a double i les que són de temps, com a int
perquè no seran 2,5 dies
    public Virus(String nom, FamiliaVirus familia, double pMal, int tInc, int
tLat, double pMor, int tCon, double pCon,
                int tImm) {

```

.no haurien de ser abstractes!

no pregunteu pel tipus !

```

        // Pre: nom i familia no poden estar buits, i els valors numèrics han de
ser positius (no tindrem
        // períodes de temps o probabilitats negatives). A més, les
probabilitats han de ser valors entre 0 i 1
        // Post: Creem un nou virus amb les característiques que ens han passat.
    }

// Primer de tot hem fet els mètodes que ens serviràn per poder retornar algun
atribut
// de la classe Virus, i així podem utilitzar-los en altres classes

// Per exemple, si volem fer càlculs de la gent que morirà... doncs hem de
tenir
// consciència de quina probabilitat de Mortalitat te.

public abstract String nom(); // En comptes de getNom()
// Pre: El virus ha de tenir un nom vàlid.
// Post: Retorna un string que és el nom del virus.

public abstract FamiliaVirus familia();
// Pre: El virus ha de tenir una família vàlida.
// Post: Retorna a quina família pertany.

/**
 * Retorna si el virus és de tipus ADN o ARN.
 * @return "ADN" o "ARN" segons el tipus de virus.
 */
public abstract String tipusVirus();
// Pre: El virus ha de ser vàlid i existir
// Post: Retorna si el virus és de tipus ADN (no mutable) o ARN (mutable)

public abstract double probDesenvoluparMalaltia();
// Pre: El virus ha de ser vàlid i existir
// Post: Retorna un valor entre 0 i 1 que representa
// la probabilitat de desenvolupar la malaltia.

public abstract int tempsIncubacio();
// Pre: El virus ha de ser vàlid i existir, i ha de tenir un temps
d'incubació
// Post: Retorna el temps d'incubació. És a dir, període de temps mig
// transcorregut entre que una persona contreu el virus
// i comencen els símptomes de la malaltia

public abstract int tempsLatencia();
// Pre: El virus ha de ser vàlid i existir, i ha de tenir un temps de
lactància

```

?

.

?

.

x

?

.

?

.

?

```

    // Post: Retorna el temps de lactància. És a dir, període de temps mig
transcorregut
    // entre que una persona contreu el virus i pot començar a contagiar-lo.

    public abstract double taxaMortalitat();
    // Pre: El virus ha de ser vàlid i existir
    // Post: Retorna un valor entre 0 i 1 que representa
    // que una persona malalta per efecte del virus morí a causa de
    // la malaltia en qüestió.

    public abstract int tempsContagi();
    // Pre: El virus ha de ser vàlid i existir, i ha de tenir un temps de
contagi
    // Post: Retorna el temps de contagi. És a dir, període de temps mig durant
el qual la persona
    // infectada pot contagiar el virus a d'altres persones

    public abstract double taxaContagi();
    // Pre: El virus ha de ser vàlid i existir
    // Post: Retorna un valor entre 0 i 1 que representa la probabilitat que una
    // persona infectada, en període de contagi, li passi el virus
    // a una altra persona no infectada amb qui ha mantingut un contacte directe

    public abstract int tempsImmunitat();
    // Pre: El virus ha de ser vàlid i existir, i ha de tenir un temps
d'immunitat
    // Post: Retorna el temps d'immunitat. És a dir, temps mig que una persona
és immune
    // al virus (i a les seves possibles mutacions) un cop curada
}

```

---

## 6) VirusARN

```

/**
 * @class VirusARN
 * @brief Representa un virus de tipus ARN, que pot mutar.
 */
public class VirusARN extends Virus {

    // Tot i que ara no hem de posar atributs, si que expliquem aquí
    // per quan seguim amb el treball, que tindrem un atribut per a aquesta
    // classe de virus que muten, que és la
    // P_mutEC probabilitat de mutació per error de còpia
}

```



```

// Fem primer el constructor de la classe VirusARN, tenint en compte aquest
nou atribut
public VirusARN(String nom, FamiliaVirus familia, double pMal, int tInc, int
tLat, double pMor, int tCon, double pCon, int tImm, double pMutEC) {
    // Pre: Els valors han de ser positius (prob han d'anar de 0 a 1) i el
nom ha de ser vàlid.
    // Post: Es crea un virus de tipus ARN que pot mutar. A part de tenir
totes les
    // característiques del Virus, tindrà també una certa probabilitat de
mutació per error de còpia.

    super(nom, familia, pMal, tInc, tLat, pMor, tCon, pCon, tImm); // això
ho fem per
    // cridar al constructor de la classe abstracta Virus
}

public double probabilitatMutacioErrorCopia() {
    // Pre: El virus ha de ser vàlid i existir
    // Post: Retorna la probabilitat de mutació per error de còpia d'aquest
virus
}

@Override //aquí hem pensat que era bona idea fer un @Override, després
d'haver-lo vist
// a classe, basicament per poder sobreesciure el mètode tipusVirus, i
passar-li en aquest
// cas "ARN", ja que és el tipus d'aquest virus.
public String tipusVirus() {
    // Pre: L'objecte VirusARN ha d'estar inicialitzat correctament.
    // Post: Retorna un string per indicar de quin tipus de
    // virus es tracta. En aquest cas retorna un string indicant ARN.
}

// Per quan implementem la classe de Mutació, és important assegurar-nos
de
    que dos virus són
// de la mateixa família.
public boolean PertanyenMateixaFamilia (Virus A, Virus B){
// Pre: Virus A i Virus B han d'existir
// Post: Retorna cert si A i B formen part de la mateixa família,
altrament
    // retorna false
}
}

```

## 7) VirusADN

```
/**
 * @class VirusADN
 * @brief Representa un virus de tipus ADN, és a dir,
 * virus que muten amb molt poca freqüència (podem considerar que NO muten).
 */
public class VirusADN extends Virus {

    // Fem primer el constructor de la classe VirusADN.
    public VirusADN(String nom, FamiliaVirus familia, double pMal, int tInc, int
tLat, double pMor, int tCon, double pCon, int tImm) {
        // Pre: Els valors han de ser positius i el nom ha de ser vàlid.
        // Post: Es crea un virus de tipus ADN que NO pot mutar.

        super(nom, familia, pMal, tInc, tLat, pMor, tCon, pCon, tImm); // això
ho fem per
        // cridar al constructor de la classe abstracta Virus
    }

    public double probabilitatMutacioErrorCopia() {
        // Pre: El virus ha de ser vàlid i existir
        // Post: Retorna la probabilitat de mutació per error de còpia d'aquest
virus
    }

    @Override //aquí hem pensat que era bona idea fer un @Override, després
d'haver-lo vist
        // a classe, basicament per poder sobre escriure el mètode tipusVirus, i
passar-li en aquest
        // cas "ARN", ja que és el tipus d'aquest virus.
    public String tipusVirus() {
        // Pre: L'objecte VirusADN ha d'estar inicialitzat correctament.
        // Post: Retorna un string per indicar de quin tipus de
        // virus es tracta. En aquest cas retorna un string indicant ADN.
    }
}
```

## 8) FamiliaVirus

```
/**
 * @class FamiliaVirus
 * @brief Representa una família de virus, agrupant diferents virus que tenen
característiques comunes.
 * @details Cada virus pertany a una família. Diferents espècies de virus poden
pertànyer a una mateixa família,
 * però o tots els que hi ha son NO MUTABLES, o tots son MUTABLES, però no hi ha
barreja dels dos grups en les famílies.
 *
 *
 * @author Romà Barrera, Mireia Ferrer i Iria Auladell
 * @version 2025.2.18
 */
public class FamiliaVirus {

    /**
     * Constructor de la classe FamiliaVirus.
     * @param nom Nom de la família de virus.
     * @param Pmut2E és la probabilitat de mutació per coincidència (si és una
família de virus ADN,
     * valdrà 0)
     * @param tpcMax Tant per cent màxim de variació en mutacions per error de
còpia.
     */
    public FamiliaVirus(String nom, double Pmut2E, double tpcMax) {
        // Pre: El nom no pot ser buit i la taxa de mutació ha d'estar entre 0 i
1.
        // Post: Es crea una nova família de virus amb el nom, la taxa de
mutació i el tpcMax que ens han passat.
    }

    public String nom() {
        // Pre: L'objecte FamiliaVirus ha d'estar inicialitzat correctament.
        // Post: Retorna el nom de la família.
    }

    public double probMutacioCoincidencia() {
        // Pre: L'objecte FamiliaVirus ha d'estar inicialitzat correctament.
    }
```

```

        // Post: Retorna un valor entre 0 i 1 que és la taxa de mutació. si és
una família de virus ADN,
        // retornarà 0
    }

    public double tpcMax() {
        // Pre: L'objecte FamiliaVirus ha d'estar inicialitzat correctament.
        // Post: Retorna tant per cent màxim de variació (en mutacions per error
de còpia)
    }
}

```

## 9) Mutacio

```

/**
 * @class Mutacio
 * @brief Ens diu com es comporta una mutació i com aquesta afecta.
 * @details Quan un virus muta, apareix una nova variant amb característiques
diferents, que poden afectar
 * aspectes com la taxa de contagi, el temps d'incubació o la mortalitat. No
sempre implica que sigui més perillosa.
 * Els virus ARN són els que muten, i poden fer-ho de dues maneres:
 * - Per error de còpia: en reproduir-se, el virus pot generar mutacions
aleatòries.
 * - Per coincidència: quan dues soques de la mateixa família infecten el mateix
individu, poden combinar-se.
 * Quan una mutació passa, el nou virus es propaga dins del cos de l'infectat.
 *
 *
 * @author Romà Barrera, Mireia Ferrer i Iria Auladell
 * @version 2025.3.13
 */

// Com que tots els virus ARN muten seguint el mateix patró, llavors hem
decidit que seria una
// bona idea que Mutacio fos un Modul funcional. A més, així complim els
principis solids i evitem que la
// classe VirusARN s'ocupi tant de el virus en si, com de la mutació.
// D'aquesta manera ho tenim tot més organitzat i permetem que aquest modul
funcional
// només l'implementi la classe VirusARN, que és la única que muta.
public abstract class Mutacio {
    ...
}

```

Millor integrar mutació a VirusARN

```

    * Fem ús d'aquest mètode quan hi ha una mutació per error de còpia a un virus
    ARN.
    * @param virus El virus ARN que muta.
    * @return El nou virus ARN amb les seves característiques modificades.
    */
    public static VirusARN mutacioPerErrorCopia(VirusARN virus) {
        // Pre: El virus ha de ser ARN (els únics que poden mutar) i ha de tenir
        la seva probabilitat de mutació
        // per error de còpia.
        // Post: Es genera un nou virus ARN de la mateixa família que
        l'original, amb els paràmetres modificats.
        // El nom d'aques virus serà el nom del virus que muta però amb un
        número correlatiu que indiqui el número
        // de vegades que ha mutat.
    }

    /**
    * Fem servir aquest mètode quan hi ha una mutació per coincidència entre dos
    virus de la mateixa família.
    * @param A Persona ja està infectada d'aquest virus.
    * @param B Persona infectada de A, s'infecta també d'aquest virus.
    * @return C Un nou virus resultant de la combinació de A i de B, que agafarà
    característiques dels dos i formarà
    * part de la mateixa família
    */
    public static VirusARN mutacioPerCoincidencia(Virus A, Virus B) {
        // Pre: A i B han de pertànyer a la mateixa família de virus i els dos
        han d'estar presents en la mateixa
        // persona infectada al mateix moment.
        // Post: Retorna un Virus C, que pertany a la mateixa classe de A i de
        B. Aquest, tindrà característiques tant
        // de A com de B, i els seus paràmetres quedaran canviats. A més, la
        persona infectada passarà a tenir aquest
        // nou virus en comptes de A i de B. El nom d'aquest serà la
        concatenació dels noms de A i de B
    }

```

## 10) AfectacioVirusRegió:

```
/**
 * @class AfectacioVirusRegio
 * @brief Controla l'evolució d'un virus en una regió concreta dins la simulació.
 * @details Gestiona la infecció en una regió seguint l'evolució temporal del
virus. Es tenen en compte:
 *   - Infectats: persones que han contret el virus.
 *   - Contagiosos: infectats que poden transmetre'l.
 *   - Malalts: infectats que desenvolupen símptomes.
 *   - Nous infectats: persones contagiades en l'últim dia.
 *   - Immunitzats: persones recuperades.
 *   - Morts: persones que han mort per culpa de la malaltia.
 *
 * Cada dia s'actualitza l'estat de la regió:
 *   - Es calculen nous contagis segons la taxa de transmissió (P_con) i la
mobilitat.
 *   - Es distribueixen les morts proporcionalment al llarg dels dies de malaltia.
 *   - Es mantenen estadístiques acumulades des de l'inici de la simulació.
 *
 * La simulació treballa en mode text i permet consultar l'estat actual i
l'evolució històrica.
 *
 * @author Romà Barrera, Mireia Ferrer i Iria Auladell
 * @version 2025.3.13
 */

public class AfectacioVirusRegio {

    /**
     * Constructor de la classe AfectacioVirusRegio.
     * @param virus Virus que afecta la regió.
     * @param regio Regió que és infectada per el virus.
     */
    public AfectacioVirusRegio(Virus virus, Regio regio) {
        // Pre: virus ha de ser un objecte vàlid de la classe Virus, i regio ha
de ser una regió
        // que existeixi.
        // Creem una zona concreta que està afectada per un virus concret.
    }
```

```

/**
 * Que fa ? Si avancem la simulació un dia, llavors s'actualitza l'estat de la
 * infecció en aquesta la regió.
 */
public void avançarUnDia(){
    // Pre: La simulació s'ha d'estar executant
    // Post: S'actualitza l'estat, calculant de nou els nous contagis,
    // morts, persones immunes, segons
    // les característiques de la regió i del virus que està afectant.
}
}

```

✓ FALTEN MOLTS

## 11) Classes dels FITXERS

```

public abstract class FitxerVirus {
    // hegiq...

    public static List<Virus> llegir_virus(File fitxer) {
        //Pre: El fitxer ha d'existir
        //Post: Es retorna una llista de virus.
    }
}

public abstract class FitxerRegio{

    public static List<Regio> llegir_regions(File fitxer) {
        //Pre: Fitxer ha d'existir
        //Post: Retorna una llista de les regions existents
    }
}

public abstract class FitxerIncidencia{

    public static List<Incidencia> llegir_incidenca(File fitxer) {
        //Pre: Fitxer ha d'existir
        //Post: Retorna una llista de les incidencies existents
    }
}

```

Cal parlar d'aquestes dependències amb Regió, Virus...

