

# **Programació Avançada i Estructura de Dades**

**Projecte del Segon Semestre – Estructures No Lineals**

***Els Cavallers de la Taula de Hash***

**Departament d'Enginyeria  
La Salle - Universitat Ramon Llull**

9 maig de 2023

*But first, some words of wisdom...*

*“Generally, the craft of programming is the factoring of  
a set of requirements into a set of functions and data structures.”*

---

DOUGLAS CROCKFORD, JavaScript: The Good Parts, 2008

*“The difference between a bad programmer and a good one is  
Whether he considers his code or his data structures more important.”*

---

LINUS TORVALDS, Re: Licensing and the library version of git, 2006

*“Data dominates. If you’ve chosen the right data structures  
And organized things well, the algorithms will almost always be self-evident.”*

---

ROB PIKE, Notes on programming in C, 1989

*“It is better to have 100 functions operate  
on one data structure than 10 functions on 10 data structures.”*

---

ALAN PERLIS, Epigrams on Programming, 1982

*And now for something completely different...*

## Índex

<b>1</b>	<b>Introducció.....</b>	<b>1</b>
<b>2</b>	<b>Requeriments .....</b>	<b>2</b>
2.1	Sobre orenetes i cocos (Grafs).....	2
2.1.1	Dades.....	2
2.1.2	Funcionalitats .....	3
2.2	Caça de bruixes (Arbres binaris de cerca) .....	4
2.2.1	Dades.....	4
2.2.2	Funcionalitats .....	5
2.3	Tanques de bardissa (Arbres R).....	6
2.3.1	Dades.....	6
2.3.2	Funcionalitats .....	7
2.4	D'heretges i blasfems (Taules).....	8
2.4.1	Dades.....	8
2.4.2	Funcionalitats .....	8
<b>3</b>	<b>Execució.....</b>	<b>10</b>
3.1	Menú inicial .....	10
3.2	Menú “Sobre orenetes i cocos” .....	10
3.3	Menú “Caça de bruixes” .....	12
3.4	Menú “Tanques de bardissa” .....	14
3.5	Menú “D'heretges i blasfems” .....	15
<b>4</b>	<b>Consideracions generals .....</b>	<b>18</b>
4.1	Grups .....	18
4.2	Llenguatge .....	18
4.3	Implementacions de tercers.....	18
4.4	Desenvolupament .....	18
4.5	Detecció de frau .....	18
<b>5</b>	<b>Avaluació.....</b>	<b>19</b>
5.1	Data d'entrega .....	19
5.2	Format d'entrega.....	19
5.3	Contingut de la memòria.....	19
5.4	Examen de validació .....	20
5.5	Recuperació .....	20

## 1 Introducció

En el món de l'enginyeria, com a tot arreu, trobar un bon equip de treball és un factor que pot tenir un impacte crític sobre els resultats que s'aconsegueixen.

Dit això, avui en dia disposem d'eines que faciliten en gran mesura la formació d'equips, o fins i tot el treball a distància. Aquesta realitat ens ajuda a superar els reptes als que molta gent, incloent figures històriques, ja s'han enfrontat.

Un d'aquests casos va ser el del Rei Artús, fill d'Uterpendragó i monarca sobirà de la Bretanya, qui va decidir formar la Taula Rodona<sup>1</sup> de Camelot. Aquesta taula, en la que ningú podia seure al cap, estava formada pels cavallers més poderosos del regne, així com de regnes veïns.

Tant el procés de formació de la Taula com el seu funcionament dia a dia van resultar complexos degut a les limitacions del coneixement de l'època. És per això que volem desenvolupar **The Hashy Grail**, un sistema informàtic que, mitjançant la tecnologia moderna, ens permeti esbrinar com es podrien haver optimitzat els processos de treball de l'edat mitjana.

En aquest document, que s'anirà actualitzant al llarg del semestre, s'exposen els requeriments del projecte, se'n donen exemples d'execució, se n'indiquen consideracions rellevants i se'n descriu el sistema d'avaluació.

---

<sup>1</sup> Algunes fonts secundàries discrepen respecte la forma de la taula.

## 2 Requeriments

La vida durant l'edat mitjana era força difícil. Com a tota etapa de la història de la humanitat, la intersecció de diferents constructes socials acabava generant dinàmiques més complexes que la suma de les seves parts. Tot i això, hem decidit dividir els requeriments en diferents blocs, per tal de reduir-ne la dimensió.

A les següents seccions s'expliquen les funcionalitats a implementar en aquest projecte, juntament amb descripcions de les dades amb les que tractarem.

### 2.1 Sobre orenetes i cocos (Grafs)

Un dels sistemes de comunicació més utilitzats durant l'edat medieval eren els ocells missatgers, sent les orenetes les aus que més sovint es feien servir per transportar secrets. Tanmateix, hi ha un aspecte del protocol aviari que fins i tot en aquella època passava desapercebut: La diferència entre les orenetes europees i les africanes.

Aquestes dues espècies tenen capacitats de vol radicalment diferents, especialment si considerem situacions en les que han de portar missatges llargs o objectes relativament pesats, com per exemple cocos per nodrir-se. Per això, ens hem proposat trobar formes d'optimitzar els seus trajectes entre llocs d'interès.

#### 2.1.1 Dades

Pels motius prèviament descrits disposarem d'un fitxer `.paed`, que contindrà el següent conjunt de dades en text pla:

- **Nombre de llocs d'interès:** La primera línia del fitxer contindrà un enter  $N$ , indicant el nombre de llocs d'interès que trobarem a continuació.
- **Llista de llocs d'interès:**  $N$  línies de text on cadascuna conté els següents camps, separats per punts i coma:
  - **Id:** Enter identificatiu del lloc d'interès. Serà únic però no necessàriament consecutiu. Tampoc no estarà ordenat.
  - **Nom:** Nom del lloc d'interès. Pot contenir més d'una paraula. No es garanteix que sigui únic.
  - **Regne:** Nom del regne al que pertany. Pot contenir més d'una paraula. No es garanteix que sigui únic.
  - **Clima:** Tipus de clima del lloc d'interès. El valor pot ser `POLAR`, `CONTINENTAL` o `TROPICAL`.
- **Nombre de trajectes coneguts:** La següent línia del fitxer contindrà un enter  $T$ , indicant el nombre de mesures de temps i distància que tindrem, cadascuna corresponent a un trajecte entre dos llocs d'interès.
- **Llista de trajectes coneguts:**  $T$  línies de text on cadascuna conté els següents camps, separats per punts i coma:

- **Lloc A:** Identificador del primer lloc d'interès.
- **Lloc B:** Identificador del segon lloc d'interès.
- **Temps E:** Temps en minuts que triga una oreneta europea a volar del lloc A al lloc B (o viceversa).
- **Temps A:** Temps en minuts que triga una oreneta africana a volar del lloc A al lloc B (o viceversa).
- **Distància:** Quilòmetres de distància que hi ha entre els llocs A i B, mesurada en línia recta.

### 2.1.2 Funcionalitats

Caldrà que la implementació d'aquest apartat ofereixi les següents funcionalitats:

#### 2.1.2.1 Representació com a graf

El codi haurà de definir i implementar la representació d'un graf per a desar les dades proporcionades de forma **eficient**.

La realització d'aquesta tasca és imprescindible per desenvolupar la resta d'aquest apartat.

#### 2.1.2.2 Exploració del regne

Per tenir una idea de la situació geopolítica dels llocs que coneixem, en aquesta funcionalitat caldrà demanar a l'usuari l'identificador d'un lloc d'interès d'origen. A continuació, es mostraran tots els llocs d'interès del mateix regne als que es pugui arribar de qualsevol forma des d'aquest.

En termes d'ordre, cal prioritzar aquells llocs d'interès als que es pugui arribar sense sortir del regne.

#### 2.1.2.3 Detecció de trajectes habituals

A continuació, volem intentar detectar els trajectes més habituals en el sector de la missatgeria aèria, independentment del tipus d'ocell.

Assumint que les aus tendeixen a anar en línia recta, trobeu el conjunt de trajectes que connecten tots els llocs d'interès i minimitzen la distància a recórrer.

#### 2.1.2.4 Missatgeria *premium*

Finalment, també volem oferir una funcionalitat per calcular quina és la forma més ràpida (que requereixi menys temps) de fer arribar un missatge des d'un lloc d'interès fins a un altre, passant pels punts intermedis que calgui.

Aquesta funcionalitat haurà de demanar a l'usuari:

- Identificador del lloc d'interès d'origen.
- Identificador del lloc d'interès destí.
- Si l'oreneteta farà el vol carregant amb un coco o no.

La sortida esperada és una llista de llocs d'interès a visitar per arribar de l'origen al destí, en l'ordre que calgui seguir. També caldrà informar a l'usuari de si seria més eficient enviar una oreneta europea o africana, així com el temps i distància totals que suposaria fer-ho.

A continuació es descriuen un seguit de consideracions a tenir en compte en aquesta funcionalitat:

- Les orenetes europees només poden visitar climes polars o continentals.
- Les orenetes africanes només poden visitar climes continentals o tropicals.
- Les orenetes que portin un coco no poden fer trajectes de més de 50 quilòmetres sense aturar-se a un lloc d'interès. Podem negligir el temps que descansin.
- Si no coneixem el trajecte entre dos llocs, assumirem que les orenetes no poden realitzar-lo directament.
- No es garanteix que existeixi una solució.

## 2.2 Caça de bruixes (Arbres binaris de cerca)

Una de les qüestions que preocupava a les masses populars durant l'època medieval era la possibilitat de que hi hagués bruixes infiltrades entre la població. Ara bé, deixar-se portar per aquesta por podia acabar causant la presa de decisions irracionals. Per això, cada cop que hi havia rumors sobre l'existència de bruixes es duïen a terme experiments científics per validar-los o desmentir-los.

En aquest apartat ens dedicarem a simular aquests experiments.

### 2.2.1 Dades

Pels motius prèviament descrits disposarem d'un fitxer `.paed`, que contindrà el següent conjunt de dades en text pla:

- **Nombre d'habitants:** La primera línia del fitxer contindrà un enter  $\mathbb{N}$ , indicant el nombre d'habitants que trobarem a continuació.
- **Llista d'habitants:**  $\mathbb{N}$  línies de text on cadascuna conté els següents camps, separats per punts i coma:
  - **Id:** Enter identificatiu de l'habitant. Serà únic, però no necessàriament consecutiu ni ordenat.
  - **Nom:** Nom de l'habitant. Pot contenir més d'una paraula i no es garanteix que sigui únic.
  - **Pes:** Real que representa el pes en quilograms de l'habitant.
  - **Regne:** Nom del regne al qual viu l'habitant. Pot contenir més d'una paraula i no es garanteix que sigui únic.

### 2.2.2 Funcionalitats

Caldrà que la implementació d'aquest apartat ofereixi les següents funcionalitats:

#### 2.2.2.1 Representació com a arbre binari de cerca

El codi haurà de definir i implementar la representació d'un arbre binari de cerca per a desar les dades proporcionades de forma **eficient**.

La realització d'aquesta tasca és imprescindible per desenvolupar la resta d'aquest apartat.

#### 2.2.2.2 Funcionalitats bàsiques

Un cop representades les dades del dataset com un arbre binari de cerca, caldrà poder modificar-lo en temps d'execució. Persistir els canvis al fitxer NO és obligatori.

Concretament, el programa ha de permetre a l'usuari afegir nous habitants (introduint-ne tota la informació), així com eliminar-los (a partir del seu identificador).

#### 2.2.2.3 Representació visual

El programa haurà de permetre visualitzar l'arbre binari de cerca. Aquesta visualització pot basar-se en text, seguint el format que es presenta als exemples d'execució. Opcionalment es pot implementar una representació gràfica 2D.

#### 2.2.2.4 Identificació de bruixes

Gràcies a l'avançada tecnologia i als amplis coneixements científics de l'època sabem que les bruixes es poden trobar comparant el seu pes amb el d'algun objecte (sí, això està demostrat científicament).

En aquest apartat caldrà demanar-li a l'usuari que introdueixi la informació de l'objecte amb el que vol comparar els habitants registrats al sistema informació. En concret, cal que introdueixi:

- Nom de l'objecte.
- Pes de l'objecte (valor real, en quilograms).
- Tipus d'objecte. Ha de ser un dels següents valors: DUCK, WOOD o STONE.

El funcionament d'aquesta funcionalitat dependrà del tipus de l'objecte:

- Si el tipus és DUCK, aquesta funcionalitat mostrarà aquells habitants que tinguin un pes exactament igual al de l'objecte introduït.
- Si el tipus és WOOD, aquesta funcionalitat mostrarà el primer habitant que pesi menys que l'objecte.
- Si el tipus és STONE caldrà mostrar el primer habitant que pesi més que l'objecte.



#### 2.2.2.5 Batuda

Un cop detectades les bruixes entre la població, els cavallers més versats en ciència procedien a realitzar batudes per capturar-les.

En aquesta funcionalitat caldrà demanar a l'usuari que introdueixi dos valors reals: Un pes mínim i un pes màxim. A continuació, el programa mostrarà la informació de tots aquells habitants que tinguin un pes comprès entre aquests valors (ambdós inclosos).

### 2.3 Tanques de bardissa (Arbres R)

Avui en dia és típic veure tanques vegetals, també conegudes com tanques de bardissa, als jardins laberíntics de grans castells. I és que aquest tipus de tanques té el seu origen a l'edat medieval, època en que se'ls donava un ús diferent. Concretament, els cavallers apreciaven tant la feina dels artesans bardissaires que sovint es feien servir com recompenses per les missions més difícils.

Era tanta l'afició dels cavallers per les tanques vegetals, que les bardisses en estat natural costaven de trobar. És per això que ens hem plantejat crear una eina que permeti simular com haguessin estat aquests processos de forma eficient.

#### 2.3.1 Dades

Pels motius prèviament descrits disposarem d'un fitxer `.paed`, que contindrà el següent conjunt de dades en text pla:

- **Nombre de bardisses:** La primera línia del fitxer contindrà un enter `N`, indicant el nombre de bardisses que trobarem a continuació.
- **Llista de bardisses:** `N` línies de text on cadascuna conté els següents camps, separats per punts i coma:
  - **Tipus:** Text indicant la forma geomètrica de la bardissa. El valor pot ser `CIRCLE` o `SQUARE`.
  - **Mida:** Real indicant la magnitud de la dimensió principal de la bardissa, en metres. En el cas del tipus `CIRCLE` correspon al radi, mentre que en les `SQUARE` correspon al costat.
  - **Latitud:** Real representant la primera coordenada del centre geomètric de la bardissa, en graus.
  - **Longitud:** Real representant la segona coordenada del centre geomètric de la bardissa, en graus.
  - **Color:** Codi RGB hexadecimal representant el color de les fulles de la bardissa. Sempre serà una tonalitat de verd.

### 2.3.2 Funcionalitats

Caldrà que la implementació d'aquest apartat ofereixi les següents funcionalitats:

#### 2.3.2.1 Representació com a arbre R

El codi haurà de definir i implementar la representació d'un arbre R per a desar les dades proporcionades de forma **eficient**.

La realització d'aquesta tasca és imprescindible per desenvolupar la resta d'aquest apartat.

**Nota:** No és obligatori tenir en compte la forma i la mida de les bardisses en la indexació de l'estructura de dades, es pot treballar únicament amb la posició. Tanmateix, tenir-ho en compte es valorarà positivament.

#### 2.3.2.2 Funcionalitats bàsiques

Un cop representades les dades del dataset com un arbre R, caldrà poder modificar-lo en temps d'execució. Persistir els canvis al fitxer NO és obligatori.

Concretament, el programa ha de permetre a l'usuari afegir noves bardisses (introduint-ne tota la informació), així com eliminar-les (a partir de la seva posició).

#### 2.3.2.3 Visualització

El programa haurà de permetre visualitzar la indexació de les bardisses.

Concretament, volem poder llistar totes les divisions internes de l'arbre així com el contingut a dins seu. És altament recomanable, si bé no obligatori, mostrar-ho de forma gràfica.

#### 2.3.2.4 Cerca per àrea

Per simular contextos locals, hem de poder centrar-nos en una única regió del planeta.

Així doncs, l'usuari haurà de poder consultar les bardisses que es troben en una zona concreta, determinada per dos punts (és a dir, dos parells de coordenades).

#### 2.3.2.5 Optimització estètica

No totes les bardisses es poden unir en una tanca vegetal de qualsevol forma. És per això que el programa ha de permetre consultar quin és el disseny més viable en un punt concret del planeta.

En primer lloc, es demanarà a l'usuari la localització que vol consultar (com a parell de coordenades) i un valor enter  $K$ . A continuació, caldrà trobar les  $K$  bardisses més properes i calcular:

- El tipus majoritari entre elles, CIRCLE o SQUARE.
- El valor mitjà dels seus colors.

## 2.4 D'heretges i blasfems (Taules)

Com hem pogut apreciar, eren moltes les responsabilitats dels Cavallers de la Taula Rodona. En ocasions, això impedia que poguessin obtenir bons resultats en totes les seves tasques. Per això, el Reial Decret Episòdic número XV va redactar-se amb l'objectiu de reduir la seva càrrega laboral.

El que ningú no esperava era la despietada eficiència d'alguns dels seus successors. I és que ningú no s'espera la Inquisició Espanyola, l'organisme que va recollir la cruel responsabilitat d'identificar i combatre els heretges del regne.

Per entendre el seu funcionament, volem crear una eina que simuli el procés d'identificació d'heretges de forma eficient.

### 2.4.1 Dades

Pels motius prèviament descrits disposarem d'un fitxer `.paed`, que contindrà el següent conjunt de dades en text pla:

- **Nombre d'acusats:** La primera línia del fitxer contindrà un enter  $N$ , indicant el nombre d'acusats que trobarem a continuació.
- **Llista d'acusats:**  $N$  línies de text on cadascuna conté els següents camps, separats per punts i coma:
  - **Nom:** Text que conté el nom de l'acusat. Pot està format per més d'una paraula però és únic per cada acusat.
  - **Nombre de conills:** Enter positiu que indica la quantitat de conills que l'acusat ha vist al llarg de la seva vida.
  - **Professió:** Text que conté la professió de l'acusat. El valor pot ser MINSTREL, KNIGHT, KING, QUEEN, PEASANT, SHRUBBER, CLERGYMAN o ENCHANTER.

### 2.4.2 Funcionalitats

Caldrà que la implementació d'aquest apartat ofereixi les següents funcionalitats:

#### 2.4.2.1 Representació com a taules

El codi haurà de definir i implementar la representació de les taules/diccionaris/mapes que calguin per a desar les dades proporcionades de forma **eficient**.

La realització d'aquesta tasca és imprescindible per desenvolupar la resta d'aquest apartat.

#### 2.4.2.2 Funcionalitats bàsiques

Un cop representades les dades del dataset mitjançant taules, caldrà poder modificar-les en temps d'execució. Persistir els canvis al fitxer NO és obligatori.

Concretament, el programa ha de permetre a l'usuari afegir nous acusats (introduint-ne tota la informació), així com eliminar-los (a partir del seu nom).

#### 2.4.2.3 Edicte de gràcia

Durant el procés d'investigació dels acusats, els inquisidors oferien la possibilitat a poblacions senceres de confessar a canvi de rebre penes reduïdes.

De forma similar, el programa ha de permetre marcar o desmarcar un acusat existent com a heretge. Ara bé, els acusats amb professió KING, QUEEN i CLERGYMAN mai poden ser heretges, pel que caldrà limitar la funcionalitat en aquest aspecte.

#### 2.4.2.4 Judici final

Per acabar de prendre decisions respecte els acusats, els inquisidors consideraven molts factors i estadístiques. El nostre programa haurà de poder efectuar algunes consultes relacionades.

En primer lloc, caldrà poder mostrar la informació d'un acusat a partir del seu nom.

En segon lloc, caldrà implementar una funcionalitat que, donat un rang de nombre de conills, llisti els acusats que hagin vist un nombre de conills dins del rang. Cal mostrar la informació de cada acusat en el rang.

Finalment, caldrà poder visualitzar un histograma on es vegi la distribució de la quantitat d'heretges en funció de les diferents professions.

El format de l'histograma és lliure, però com a mínim cal mostrar el nombre d'heretges per professió com a text. És altament recomanable, si bé no obligatori, mostrar-lo de forma gràfica.

#### 2.4.2.5 Consideracions

Per defecte, un acusat és un heretge si ha vist més de 1975 conills, excepte si la seva professió és KING, QUEEN o CLERGYMAN.

### 3 Execució

Per estandarditzar els aspectes més formals del projecte, us proporcionem els següents exemples d'execució que representen interaccions per terminal d'un usuari amb el vostre codi.

**Nota:** Com a recurs de prova i per ajudar en l'anàlisi de resultats es proveiran datasets. Aquests exemples no són representatius dels resultats que hauríeu d'obtenir, sinó que indiquen el flux d'execució del programa. Per descomptat, no s'acceptaran projectes amb dades *hardcoded*.

#### 3.1 Menú inicial

La Figura 1 mostra el menú que veurà l'usuari en primer moment quan faci servir el programa. Aquest permetrà escollir un dels apartats del projecte o sortir de l'execució.

```
'^^\ The Hashy Grail /^`'  
  
1. Sobre orenetes i cocos (Grafs)  
2. Caça de bruixes (Arbres binaris de cerca)  
3. Tanques de bardissa (Arbres R)  
4. D'heretges i blasfems (Tauls)  
  
5. Exit  
  
Esculli una opció:
```

Figura 1. Menú inicial del programa.

Quan l'usuari decideixi sortir del menú caldrà mostrar un missatge de comiat com el que apareix a la Figura 2.

```
Esculli una opció: 5  
  
Aturant The Hashy Grail...
```

Figura 2. Missatge de sortida del programa.

Per descomptat, cadascun dels apartats disposarà del seu propi submenú, que s'exposarà a les següents seccions.

#### 3.2 Menú "Sobre orenetes i cocos"

Un cop l'usuari esculli la primera opció del menú inicial (Sobre orenetes i cocos), mostrarem el menú de la Figura 3.

Esculli una opció: 1

- A. Exploració del regne
- B. Detecció de trajectes habituals
- C. Missatgeria premium
- D. Tornar enrere

Quina funcionalitat vol executar?

Figura 3. Submenú "Sobre orenetes i cocos".

Com podeu imaginar, quan l'usuari seleccioni l'opció D caldrà tornar a mostrar el menú inicial, vist a la Figura 1.

El que hauria de passar a la resta de funcionalitats s'exemplifica a la Figura 4, la Figura 5 i la Figura 6.

Quina funcionalitat vol executar? A

Quin lloc vol explorar? 42

42 – Castell de Camelot, Regne de Mercia (Clima Continental)

Els llocs del Regne de Mercia als que es pot arribar són:

7 – Castell d'Aaargh, Regne de Mercia (Clima Continental)

13 – Pont de la Mort, Regne de Mercia (Clima Polar)

12 – Congost del Perill Etern, Regne de Mercia (Clima Polar)

Figura 4. Interacció amb l'usuari executant l'opció "Exploració del regne".

Quina funcionalitat vol executar? B

Els trajectes més habituals són els següents:

...

Figura 5. Interacció amb l'usuari executant l'opció "Detecció de trajectes habituals". El format de la resta és lliure.

```
Quina funcionalitat vol executar? C
Quin és el lloc d'origen? 13
Quin és el lloc de destí? 7
L'oreneta carrega un coco? SI

L'opció més eficient és enviar una oreneta africana.
  Temps: 4213 minuts
  Distància: 3742 quilòmetres
  Camí:
```

Figura 6. Interacció amb l'usuari executant l'opció "Missatgeria premium". El format de la resta és lliure.

### 3.3 Menú "Caça de bruixes"

Un cop l'usuari esculli la segona opció del menú inicial (Caça de bruixes), mostrarem el menú de la Figura 7.

```
Esculli una opció: 2

A. Afegir habitant
B. Eliminar habitant
C. Representació visual
D. Identificació de bruixes
E. Batuda

F. Tornar enrere

Quina funcionalitat vol executar?
```

Figura 7. Submenú "Caça de bruixes".

Com podeu imaginar, quan l'usuari seleccioni l'opció F caldrà tornar a mostrar el menú inicial, vist a la Figura 1.

El que hauria de passar a la resta de funcionalitats s'exemplifica a la Figura 8, la Figura 9, la Figura 10, la Figura 11 i la Figura 12.

```
Quina funcionalitat vol executar? A

Identificador de l'habitant: 42
Nom de l'habitant: Roger the Shrubber
Pes de l'habitant: 75.19
Regne de l'habitant: Nni

Roger the Shrubber ens acompanyarà a partir d'ara.
```

Figura 8. Interacció amb l'usuari executant l'opció "Afegir habitant".

```
Quina funcionalitat vol executar? B
```

```
Identificador de l'habitant: 3
```

```
Sir Robin the Not-Quite-So-Brave ha estat transformat en un grill.
```

Figura 9. Interacció amb l'usuari executant l'opció "Eliminar habitant".

```
Quina funcionalitat vol executar? C
```

```
    |--- Rabbit of Caerbannog (666, Regne d'Aaargh): 1.5kg
    |
|--- Dingo (736, Regne d'Anthrax): 43.85kg
|   |
|   |--- Brother Maynard (7, Regne d'Antioch): 49.9kg
|   |
* Tim the Enchanter (1, Regne d'Aaargh): 69.42kg
|
|--- The Old Man from Scene 24 (24, Regne de Mercia): 72.4kg
    |
    |   |--- Roger the Shrubber (42, Regne de Nni): 75.19kg
    |   |
    |   |--- A Famous Historian (1, Regne de UK): 77.7kg
    |   |
    |--- King Arthur (5, Regne de Mercia): 78.1kg
```

Figura 10. Interacció amb l'usuari executant l'opció "Representació visual".

```
Quina funcionalitat vol executar? D
```

```
Nom de l'objecte: Shrubbery
```

```
Pes de l'objecte: 44.4
```

```
Tipus de l'objecte: WOOD
```

```
S'ha descobert 1 bruixa!
```

```
* Dingo (736, Regne d'Anthrax): 43.85kg
```

Figura 11. Interacció amb l'usuari executant l'opció "Identificació de bruixes".



Quina funcionalitat vol executar? E

Pes mínim: 44.4

Pes màxim: 73.3

S'han capturat 3 bruixes!

\* Brother Maynard (7, Regne d'Antioch): 49.9kg

\* Tim the Enchanter (1, Regne d'Aaargh): 69.42kg

\* The Old Man from Scene 24 (24, Regne de Mercia): 72.4kg

Figura 12. Interacció amb l'usuari executant l'opció "Batuda".

### 3.4 Menú "Tanques de bardissa"

Un cop l'usuari esculli la tercera opció del menú inicial (Tanques de bardissa), mostrarem el menú de la Figura 13.

Esculli una opció: 3

A. Afegir bardissa

B. Eliminar bardissa

C. Visualització

D. Cerca per àrea

E. Optimització estètica

F. Tornar enrere

Quina funcionalitat vol executar?

Figura 13. Submenú "Tanques de bardissa".

Com podeu imaginar, quan l'usuari seleccioni l'opció F caldrà tornar a mostrar el menú inicial, vist a la Figura 1.

El que hauria de passar a la resta de funcionalitats s'exemplifica a la Figura 14, la Figura 15, la Figura 16, la Figura 17 i la Figura 18.

Quina funcionalitat vol executar? A

Tipus de la bardissa: CIRCLE

Mida de la bardissa: 1.742

Latitud de la bardissa: 52.898379

Longitud de la bardissa: -3.888177

Color de la bardissa: #388E3C

Una nova bardissa aparegué a la Bretanya.

Figura 14. Interacció amb l'usuari executant l'opció "Afegir bardissa".

Quina funcionalitat vol executar? **B**

Latitud de la bardissa: **52.898379**

Longitud de la bardissa: **-3.888177**

La bardissa s'ha eliminat, per ser integrada a una tanca.

*Figura 15. Interacció amb l'usuari executant l'opció "Eliminar bardissa".*

Quina funcionalitat vol executar? **C**

Generant la visualització...

*Figura 16. Interacció amb l'usuari executant l'opció "Visualització". El format de la resta és lliure.*

Quina funcionalitat vol executar? **D**

Entra el primer punt de l'àrea (lat,long): **52.905,-3.905**

Entra el segon punt de l'àrea (lat,long): **52.725,-3.875**

S'han trobat 3 bardisses en aquesta àrea:

- \* 52.898981, -3.889005: CIRCLE (r=1.171m) #43A047
- \* 52.894735, -3.895871: SQUARE (s=3.514m) #1B5E20
- \* 52.886559, -3.883433: CIRCLE (r=2.222m) #4CAF50

*Figura 17. Interacció amb l'usuari executant l'opció "Cerca per àrea".*

Quina funcionalitat vol executar? **E**

Entra el punt a consultar (lat,long): **52.9,-3.9**

Entra la quantitat de bardisses a considerar (K): **7**

Tipus majoritari: CIRCLE

Color mitjà: #689F38

*Figura 18. Interacció amb l'usuari executant l'opció "Optimització estètica".*

### 3.5 Menú "D'heretges i blasfems"

Un cop l'usuari esculli la quarta opció del menú inicial (D'heretges i blasfems), mostrarem el menú de la Figura 19.

Esculli una opció: 4

- A. Afegir acusat
- B. Eliminar acusat
- C. Edicte de gràcia
- D. Judici final (un acusat)
- E. Judici final (rang)
- F. Histograma per professions
  
- G. Tornar enrere

Quina funcionalitat vol executar?

Figura 19. Submenú "D'heretges i blasfems".

Com podeu imaginar, quan l'usuari seleccioni l'opció G caldrà tornar a mostrar el menú inicial, vist a la Figura 1.

El que hauria de passar a la resta de funcionalitats s'exemplifica a la Figura 20, la Figura 21, la Figura 22, la Figura 23, la Figura 24 i la Figura 25.

Quina funcionalitat vol executar? **A**

Nom de l'acusat: **Tim**

Nombre de conills vistos: **1478**

Professió: **ENCHANTER**

S'ha enregistrat un nou possible heretge.

Figura 20. Interacció amb l'usuari executant l'opció "Afegir acusat".

Quina funcionalitat vol executar? **B**

Nom de l'acusat: **Tim**

L'execució pública de Tim ha estat un èxit.

Figura 21. Interacció amb l'usuari executant l'opció "Eliminar acusat".

Quina funcionalitat vol executar? **C**

Nom de l'acusat: **Tim**

Marcar com a heretge (Y/N)? **Y**

La Inquisició Espanyola ha conclòs que Tim és un heretge.

Figura 22. Interacció amb l'usuari executant l'opció "Edicte de gràcia".

Quina funcionalitat vol executar? **D**

Nom de l'acusat: **Uther**

Registre per "Uther":

- \* Nombre de conills vistos: 12533
- \* Professió: KING
- \* Heretge? No

Figura 23. Interacció amb l'usuari executant l'opció "Judici final (un acusat)".

Quina funcionalitat vol executar? **E**

Nombre mínim de conills: **27**

Nombre màxim de conills: **154**

S'han trobat els següents acusats:

Sir Bedevere:

- \* Nombre de conills vistos: 73
- \* Professió: KNIGHT
- \* Heretge? No

Zeut:

- \* Nombre de conills vistos: 27
- \* Professió: MINSTREL
- \* Heretge? Sí

Brother Maynard:

- \* Nombre de conills vistos: 100
- \* Professió: CLERGYMAN
- \* Heretge? No

Figura 24. Interacció amb l'usuari executant l'opció "Judici final (rang)".

Quina funcionalitat vol executar? **F**

Generant histograma...

Figura 25. Interacció amb l'usuari executant l'opció "Histograma per professions". El format de la resta és lliure.

## 4 Consideracions generals

### 4.1 Grups

El projecte es realitzarà en grups de 3 o 4. Cal formar-los mitjançant l'eina corresponent de l'eStudy en els períodes establerts pel professorat de l'assignatura.

Recordeu que el treball en equip no és una estratègia *Divide and Conquer*, sinó que l'aprenentatge sorgeix de compartir perspectives i prendre les decisions importants en grup.

### 4.2 Llenguatge

El projecte es pot implementar en qualsevol llenguatge de programació, a escollir pels estudiants. Si es programa en C, aquesta haurà de compilar, executar i funcionar correctament al servidor de la universitat (matagalls).

### 4.3 Implementacions de tercers

Les estructures de dades principals del projecte (grafs, arbres, taules...) han de ser implementades pels alumnes. No es permet l'ús de llibreries o estructures ja implementades que solucionin funcionalitats rellevants.

Pel que fa a estructures de dades auxiliars (llistes, piles, cues...) es recomana altament que les implementin els alumnes. En cas de fer-ne servir de ja implementades, caldrà justificar la seva elecció a la memòria i **realitzar-ne un estudi dels seus costos asimptòtics**.

De forma similar, és responsabilitat dels alumnes **explicar l'ús que s'ha fet de qualsevol eina basada en la IA** durant el desenvolupament del projecte. Tingueu present els riscos potencials de fer-la servir com a font d'informació, i inclogueu l'annex requerit a la memòria si ho feu. L'ús d'eines basades en IA fora d'aquest marc es considerarà frau (veure l'apartat 4.5).

### 4.4 Desenvolupament

Un cop establert un grup, s'assignarà un projecte a les eines Atlassian de la universitat. Independentment del llenguatge escollit cal fer-les servir pel desenvolupament, especialment Bitbucket pel versionament amb git.

**Nota:** És imprescindible fer un ús responsable de les eines de les que disposem. Per això us demanem que NO inclogueu els datasets al control de versions. Recordeu que podeu fer servir un fitxer `.gitignore` per configurar excepcions.

### 4.5 Detecció de frau

El frau s'interposa en l'aprenentatge dels alumnes, alhora que és una falta de respecte als companys que han dedicat temps i esforç a la realització del treball.

El projecte es classifica com a activitat **altament significativa**. La còpia, parcial o total, d'un company o d'internet es considerarà **acció premeditada**. Per aquests motius, aplicant la [normativa de còpies de la universitat](#), es considerarà una **falta molt greu**.

Com s'especifica a la normativa de l'assignatura, en aquest projecte es considerarà frau la còpia total o parcial d'entregues d'anys anteriors, incloent en el cas d'alumnes repetidors.

## 5 Avaluació

El projecte està dividit en quatre fases, cadascuna corresponent a un bloc de continguts. Cada fase començarà amb la publicació de l'enunciat corresponent i acabarà amb dues activitats d'avaluació: Un *checkpoint* en forma de reunió amb l'equip de l'assignatura per validar l'estat del projecte, així com un qüestionari d'autoavaluació P2P. Amb aquests dos elements es calcularà una nota orientativa per cada fase.

L'avaluació definitiva es realitzarà al final del semestre, i consistirà en dues entregues diferents: **Codi i memòria**. La nota del projecte ponderarà aquests dos elements, però cada integrant del grup obtindrà una nota individual resultant de multiplicar-la per dos factors entre 0.5 i 1.5: **Mentoria i P2P**.

$$\text{Nota\_Projecte\_S2} = 0.6 \cdot \text{Nota\_Codi} + 0.4 \cdot \text{Nota\_Memòria}$$

$$\text{Nota\_S2} = \text{Nota\_Projecte\_S2} \cdot \text{Factor\_Mentoria} \cdot \text{Factor\_P2P}$$

El factor de mentoria el determinarà l'equip de l'assignatura segons el seguiment en les sessions pràctiques, així com els resultats dels *checkpoints*.

El factor de P2P es calcularà a partir dels quatre qüestionaris d'autoavaluació. En cas de no respondre un dels qüestionaris, el factor es calcularà a partir dels altres tres. En cas de no respondre'n dos o més, la nota del projecte serà de **NP** i caldrà recuperar el semestre.

Per aprovar el codi cal que totes les fases compleixin els requeriments mínims de forma independent. En cas contrari, la nota corresponent serà com a molt de **4**.

Cal aprovar el codi i la memòria per separat amb una nota de **5** o més. En cas contrari, la nota del projecte serà la més baixa de les dues i no s'aplicaran factors individuals a cap membre del grup.

### 5.1 Data d'entrega

Les dates límit per entregar cadascuna de les parts són el **21 de maig de 2023 a les 23:55h** i el **28 de maig de 2023 a les 23:55h**, respectivament.

### 5.2 Format d'entrega

El codi ha d'entregar-se en forma de fitxer **ZIP** que contingui la carpeta del projecte desenvolupat juntament amb un fitxer README en format **TXT** o **MD**, explicant de forma detallada tot allò que sigui necessari per executar-lo (llenguatge de programació, IDE, versions, instruccions...).

L'estat del repositori Bitbucket ha de ser equivalent al projecte que s'entregui a l'eStudy. La memòria ha d'entregar-se en format **PDF** i ha d'estar formada pels continguts exposats a continuació.

### 5.3 Contingut de la memòria

La memòria hauria de contenir els següents apartats:

- Portada (amb el número de grup i els noms complets i *logins* dels membres).
- Índex numerat.
- Explicació del llenguatge de programació escollit, amb avantatges i inconvenients.
- Per cadascuna de les estructures de dades del projecte:
  - Disseny de l'estructura i justificació de decisions tècniques.
  - Explicació dels algorismes implementats.
  - Anàlisi de rendiment i resultats per algorisme.
  - Explicació del mètode de proves utilitzat.
  - Problemes observats.
- Conclusions (a nivell personal però sobretot a nivell tecnològic).
- Bibliografia (segons la norma ISO 690 o APA 7).
- Si es fa ús d'eines basades en IA, un annex explicant-ne l'ús que s'ha fet. Ha d'incloure una descripció de l'enfoc així com una **llista completa d'eines i peticions**.

Si bé l'entrega de la memòria té una data límit posterior a la del codi, això no vol dir que sigui recomanable començar-la al final. Anoteu les vostres decisions, problemes, resultats...

De forma similar, és altament recomanable analitzar el rendiment de la vostra implementació durant el desenvolupament del projecte per poder corregir possibles errades a temps.

Es proposa l'ús de LaTeX per escriure la memòria, que ha d'estar escrita en **llenguatge formal**. Es tindrà en compte que el contingut **prioritzi la qualitat abans que la quantitat**.

#### 5.4 Examen de validació

Per validar que tot i treballar en grup s'han adquirit els coneixements del semestre, tot alumne que entregui el projecte s'haurà de presentar a un examen final.

Si s'aprova amb un **5** o més, aquest examen no tindrà efecte sobre la nota final. En cas contrari la substituirà, havent de recuperar el semestre.

#### 5.5 Recuperació

En cas de suspendre el projecte (incloent no presentar-lo o suspendre l'examen de validació), es realitzarà un examen de recuperació en convocatòria extraordinària, on cal treure un **5** o més.

De forma excepcional, en funció del seguiment i l'estat del projecte es pot permetre una entrega de recuperació (sobre **8**) a aquells grups que estiguin a prop d'aprovar. L'equip de l'assignatura té la potestat de permetre re-entregar el projecte només a una part del grup, havent-se de presentar la resta de membres a l'examen de recuperació.

La re-entrega i l'examen de recuperació són mecanismes mútuament excloents. En cas que un alumne pugui re-entregar i decideixi presentar-se a l'examen aquesta serà la nota que prevaldrà.