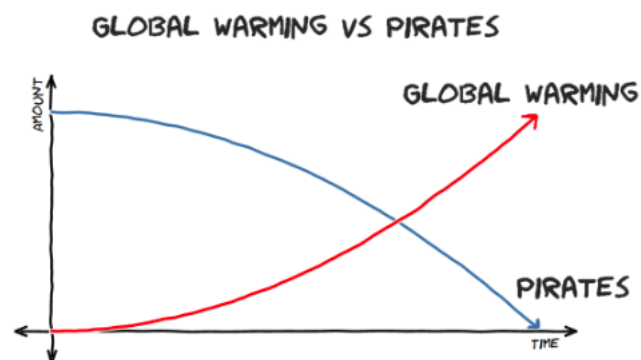

Pirates del Mediterrani

Projecte S2 - Estructures de Dades



"Smart data structures and dumb code works a lot better than the other way around."

ERIC S. RAYMOND, The Cathedral and The Bazaar

ÍNDEX

1	Introducció	4
2	Requeriments	5
2.1	Rutes marítimes (Grafs)	5
2.1.1	Dades	5
2.1.2	Funcionalitats	6
2.1.2.1	Representació com a graf	6
2.1.2.2	Cerca DFS i BFS	6
2.1.2.3	Càlcul del MST	6
2.1.2.4	Cerca del camí més curt (Dijkstra)	6
2.2	Gestió del botí (Arbres binaris)	6
2.2.1	Dades	7
2.2.2	Funcionalitats	7
2.2.2.1	Representació com a arbre de cerca binari	7
2.2.2.2	Funcionalitats bàsiques	7
2.2.2.3	Recorreguts	7
2.2.2.4	Cerques	7
2.3	Coberta (Arbres R)	8
2.3.1	Dades	8
2.3.2	Funcionalitats	8
2.3.2.1	Representació com a arbre R	8
2.3.2.2	Funcionalitats bàsiques	8
2.3.2.3	Visualització	8
2.3.2.4	Cerques	9
2.4	Tripulació (Taules)	9
2.4.1	Dades	9
2.4.2	Funcionalitats	9
2.4.2.1	Representació com a taules	9
2.4.2.2	Funcionalitats bàsiques	10
2.4.2.3	Consultes	10
3	Execució	11
3.1	Menú inicial	11
3.2	Menú de les rutes (Grafs)	12
3.3	Menú del botí (Arbres binaris)	13
3.4	Menú de la coberta (Arbres R)	16
3.5	Menú de la tripulació (Taules)	18

4	Consideracions	20
4.1	Grups	20
4.2	Llenguatge	20
4.3	Estructures de dades	20
4.4	Desenvolupament	20
4.5	Detecció de còpia	20
5	Avaluació	21
5.1	Checkpoints	21
5.2	Data d'entrega	21
5.3	Recuperació	21
5.4	Format d'entrega	22
5.5	Continguts de la memòria	22

1 INTRODUCCIÓ

Al llarg de la història in comptables persones han sentit la crida a l'aventura que representava poder navegar els set mars tot perseguint fama i fortuna.

Si bé l'edat daurada de la pirateria va desenvolupar-se principalment a les Antilles i diverses zones dels oceans Pacífic, Atlàntic i Índic, aquest ofici de dubtosa legalitat va nèixer en aigües mediterrànies.

De forma similar, el Mediterrani va ser testimoni de les activitats d'alguns dels corsaris ¹ més temuts de la història, com és el cas de Khair ed-Din Barba-rossa.

Per sort o per desgràcia, en els darrers segles el planeta ha observat una davallada important del nombre de pirates en actiu, "curiosament" correlacionada amb l'escalfament global:

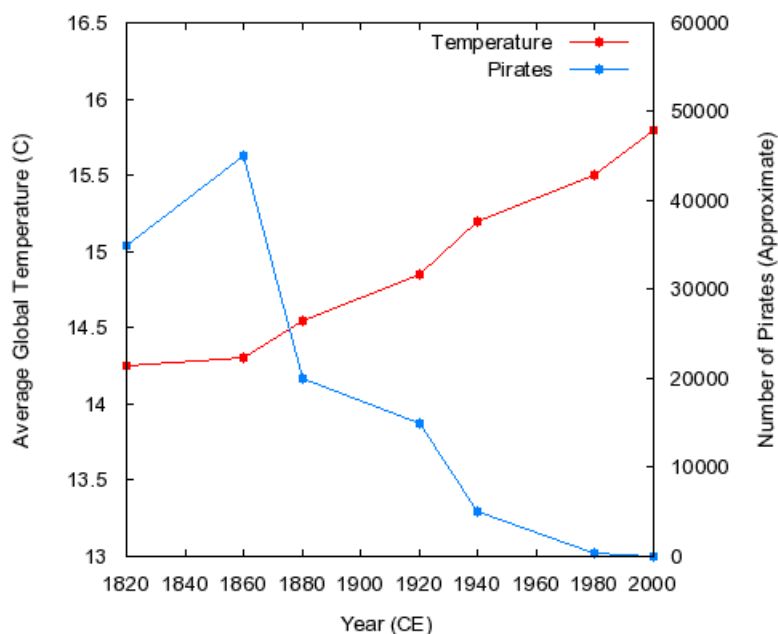


Figura 1: Correlació entre l'augment de la temperatura mitjana i la desaparició dels pirates.

En aquest projecte intentarem simular parts de l'experiència pirata, tot aprenent a implementar i utilitzar diferents estructures de dades modernes.

¹ Si bé és cert que es pot fer una distinció entre pirates i corsaris, aquesta no és rellevant pel projecte.

2 REQUERIMENTS

La vida del pirata era més complexa del que pugui semblar, i entre saquejos i espolis havien de preocupar-se d'organitzar-se bé per evitar mals de caps innecessaris.

A les següents seccions s'expliquen els aspectes logístics de la vida a alta mar que simularem en aquest projecte, juntament amb descripcions de les dades amb les que tractarem.

2.1 RUTES MARÍTIMES (GRAFS)

Un dels punts més importants en la planificació piràtica consistia en trobar rutes òptimes entre punts d'interès (coves del tresor, mercats negres...) que esquivessin certs obstacles (ports altament vigilats, llars de monstres marítims...).

Tot i així, els mapes de l'època no eren massa precisos i calia confiar més en les distàncies conegudes entre punts que en les posicions d'aquests.

2.1.1 DADES

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

- **Nombre de llocs:** La primera línia del fitxer contindrà un enter N indicant el nombre de llocs que trobarem a continuació.
- **Llista de llocs:** N línies de text on cadascuna conté els següents camps, separats per comes:
 - **Id:** Enter identificatiu del lloc, serà únic però pot estar desordenat.
 - **Nom:** Nom del lloc, pot incloure més d'una paraula.
 - **Tipus:** Una cadena de caràcters indicant si el lloc és un punt d'interès o és perillós. El valor pot ser INTEREST o DANGER respectivament.
- **Nombre de distàncies conegudes:** La següent línia del fitxer contindrà un enter D indicant el nombre de distàncies que trobarem a continuació.
- **Llista de distàncies conegudes:** D línies de text on cadascuna conté els següents camps, separats per comes:
 - **Origen:** Identificador del lloc d'origen.
 - **Destí:** Identificador del lloc destí.
 - **Distància:** Nombre real, resultat de mesurar en dies el temps que es triga a viatjar del lloc origen al destí.

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 Cerca DFS i BFS

Per ajudar els pirates a conèixer la zona en la que operen, caldrà poder llistar els llocs segons el seu tipus.

En concret, caldrà oferir una opció per llistar llocs d'interès implementada mitjançant l'algorisme Depth-First Search. De forma similar, caldrà poder llistar llocs perillosos fent ús de l'algorisme Breadth-First Search.

2.1.2.3 Càlcul del MST

Una altra funcionalitat important serà la generació de la "carta nàutica universal".

Per a fer-ho, caldrà calcular el Minimum Spanning Tree del graf mitjançant un dels dos algorismes vistos a l'assignatura. El format per visualitzar-lo és de lliure elecció.

2.1.2.4 Cerca del camí més curt (Dijkstra)

Finalment, caldrà poder trobar la ruta òptima entre dos llocs d'interès. En concret, voldrem trobar el camí més curt que eviti llocs perillosos sempre que sigui possible.

Cal fer servir l'algorisme de Dijkstra vist a classe, modificant-lo segons sigui necessari per fer servir el criteri descrit. El format per visualitzar el resultat és de lliure elecció.

2.2 GESTIÓ DEL BOTÍ (ARBRES BINARIS)

Després de recórrer les seves rutes preferides, ara ben optimitzades gràcies a la feina feta a l'apartat anterior, els pirates solien obtenir muntanyes de tresors com a botí.

En aquest apartat ens dedicarem a posar-hi ordre per poder gestionar millor el valor econòmic (en doblons) de tot el que trobem.

2.2.1 DADES

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

- **Nombre de tresors:** La primera línia del fitxer contindrà un enter N indicant el nombre de tresors que trobarem a continuació.
- **Llista de tresors:** N línies de text on cadascuna conté els següents camps, separats per comes:
 - **Nom:** Nom del tresor, pot incloure més d'una paraula.
 - **Valor:** Valor en doblons del tresor. A diferència de les unitats modernes de valor econòmic, els doblons només poden ser nombres enters.

2.2.2 FUNCIONALITATS

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

2.2.2.1 Representació com a arbre de cerca binari

El codi haurà de definir i implementar la representació d'un arbre de cerca binari 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, caldrà poder modificar-lo en execució. Persistir els canvis al fitxer NO és obligatori.

En concret, el programa ha d'oferir les opcions afegir nous tresors, així com eliminar-los.

2.2.2.3 Recorreguts

Per a que els pirates puguin fer una mica d'inventari i controlar com porten el botí, caldrà que oferim formes de llistar tots els seus tresors.

Concretament, volem poder conèixer els recorreguts en preordre, postordre, inordre i per nivells.

2.2.2.4 Cerques

Finalment, oferirem dues formes de trobar tresors concrets dins tots els que hi hagi al botí, que sovint pot ser caòtic.

Específicament, oferirem una cerca per valor exacte i una altra per rang de valors, que retornarà tots els tresors entre un valor mínim i un de màxim.

2.3 COBERTA (ARBRES R)

A banda de gestionar el valor econòmic dels seus tresors, els pirates també necessitaven poder repartir-los per la coberta dels seus vaixells.

La nostra tasca serà desar de forma eficient la posició de cada tresor, amb l'objectiu de poder realitzar consultes segons proximitat.

2.3.1 DADES

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

- **Nombre de tresors:** La primera línia del fitxer contindrà un enter N indicant el nombre de tresors que trobarem a continuació.
- **Llista de tresors:** N línies de text on cadascuna conté els següents camps, separats per comes:
 - **Nom:** Nom del tresor, pot incloure més d'una paraula.
 - **X:** Coordenada X del punt de l'espai on es troba el tresor.
 - **Y:** Coordenada Y del punt de l'espai on es troba el tresor.

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.

2.3.2.2 Funcionalitats bàsiques

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

En concret, el programa ha d'oferir les opcions afegir nous tresors, així com eliminar-los.

2.3.2.3 Visualització

Els pirates han de poder conèixer l'estat de la coberta en tot moment.

Concretament, volem poder llistar tots els rectangles de l'arbre així com el contingut a dins seu. És altament recomanable, si bé no obligatori, mostrar-los de forma gràfica.

2.3.2.4 Cerques

Finalment, oferirem dues formes de trobar tresors concrets dins tots els que hi hagi a la coberta, que sovint pot ser plena.

Específicament, oferirem una cerca per àrea (trobant tot el que hi hagi en un rectangle introduït per l'usuari) i una cerca per proximitat (trobant els tresors més propers a un punt introduït per l'usuari).

2.4 TRIPULACIÓ (TAULES)

Hem estat tan concentrats en els tresors que ens hem oblidat del més important: Els propis pirates! Gestionar la tripulació d'un vaixell també requeria molta organització, i era vital per evitar motins.

En aquest apartat ens dedicarem a organitzar la informació de tots els tripulants per tenir-hi fàcil accés, així com a fer-ne un recompte per edat.

2.4.1 DADES

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

- **Nombre de pirates:** La primera línia del fitxer contindrà un enter N indicant el nombre de pirates que trobarem a continuació.
- **Llista de pirates:** N línies de text on cadascuna conté els següents camps, separats per comes:
 - **Nom:** Nom del pirata, pot estar format per més d'una paraula.
 - **Edat:** Edat del pirata, un enter entre 14 i 50. Font.
 - **Rol:** Rol del pirata, describint la seva posició en la jerarquia de la tripulació. Pot estar format per més d'una paraula.

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 de la tripulació 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 execució. Persistir els canvis al fitxer NO és obligatori.

En concret, el programa ha d'oferir les opcions afegir nous pirates, així com eliminar-los.

2.4.2.3 Consultes

El capità dels pirates ha de poder conèixer l'estat de la seva tripulació en tot moment.

Oferirem dues consultes: Conèixer la informació d'un pirata a partir del seu nom i veure un histograma de la quantitat de pirates per cada valor possible d'edat.

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

3 EXECUCIÓ

Per estandaritzar 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 analitzar 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

L'[snippet 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.

```
-- Pirates del Mediterrani ==  
  
1. Rutes (Grafes)  
2. Botí (Arbres binaris)  
3. Coberta (Arbres R)  
4. Tripulació (Taules)  
  
5. Sortir  
  
Escull una opció:
```

Snippet 1: Menú inicial del programa.

Quan l'usuari decideixi sortir del menú caldrà mostrar un missatge de comiat com el que apareix a l'[snippet 2](#).

```
Escull una opció: 5  
  
A reveure camarada!
```

Snippet 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 DE LES RUTES (GRAFS)

Un cop l'usuari esculli l'opció de gestionar rutes marítimes, mostrarem el menú de l'[snippet 3](#).

```
Escull una opció: 1

A. Cercar llocs d'interès (DFS)
B. Cercar llocs perillosos (BFS)
C. Mostrar la Carta Nàutica Universal (MST)
D. Trobar la ruta òptima (Dijkstra)

E. Tornar enrere

Quina funcionalitat vols executar?
```

Snippet 3: Menú per gestionar rutes.

Com podeu imaginar, quan l'usuari sel·leccioni l'opció E caldrà tornar a mostrar el menú inicial, vist a l'[snippet 1](#).

Els [snippets 4](#), [5](#), [6](#) i [7](#) exemplifiquen el que hauria de passar a la resta de les funcionalitats.

```
Quina funcionalitat vols executar? A

Entra l'identificador del node origen: 42

Amb DFS s'han trobat els següents llocs d'interès:

Base pirata
Cova del tresor
```

Snippet 4: Interacció amb l'usuari executant la primera opció de rutes.

Quina funcionalitat vols executar? **B**

Entra l'identificador del node origen: **42**

Amb BFS s'han trobat els següents llocs perillosos:

Niu del kraken

Aigües profundes

Snippet 5: Interacció amb l'usuari executant la segona opció de rutes.

Quina funcionalitat vols executar? **C**

Obtenint el MST...

Snippet 6: Interacció amb l'usuari executant la tercera opció de rutes. El format del resultat és lliure.

Quina funcionalitat vols executar? **D**

Entra l'identificador del node origen: **42**

Entra l'identificador del node destí: **24**

Obtenint la ruta òptima...

Snippet 7: Interacció amb l'usuari executant la quarta opció de rutes. El format del resultat és lliure.

3.3 MENÚ DEL BOTÍ (ARBRES BINARIS)

Un cop l'usuari esculli l'opció de gestionar el botí, mostrarem el menú de l'[snippet 8](#).

Escull una opció: 2

- A. Afegir tresor
- B. Eliminar tresor
- C. Llistar botí
- D. Cerca per valor (exacte)
- E. Cerca per valor (rang)

- F. Tornar enrere

Quina funcionalitat vols executar?

Snippet 8: Menú per gestionar el botí.

Com podeu imaginar, quan l'usuari sel·leccioni l'opció F caldrà tornar a mostrar el menú inicial, vist a l'[snippet 1](#).

Els [snippets 9, 10, 11, 12 i 13](#) exemplifiquen el que hauria de passar a la resta de les funcionalitats.

Quina funcionalitat vols executar? A

Entra el nom del tresor a afegir: **El tresor de Rackham el Roig**

Entra el valor del tresor a afegir: **194302190**

El tresor s'ha afegit correctament al botí.

Snippet 9: Interacció amb l'usuari executant la primera opció del botí.

Quina funcionalitat vols executar? B

Entra el nom del tresor a eliminar: **El tresor de Rackham el Roig**

El tresor s'ha eliminat correctament del botí.

Snippet 10: Interacció amb l'usuari executant la segona opció del botí.

Per l'opció C cal tenir present que l'usuari pot escollir diferents recorreguts, com s'aprecia a l'[snippet 11](#).

Quina funcionalitat vols executar? **C**

- I. Preordre
- II. Postordre
- III. Inordre
- IV. Per nivells

Quin recorregut vols fer servir? **IV**

El tresor perdut de Lima - 180.000.000 doblons
Un cofre sospitós - 310.000 doblons
El tresor de Rackham el Roig - 194.302.190 doblons
...

Snippet 11: Interacció amb l'usuari executant la tercera opció del botí.

Quina funcionalitat vols executar? **D**

Entra el valor a cercar: **194302190**

S'ha trobat un tresor amb aquest valor: El tresor de Rackham el Roig

Snippet 12: Interacció amb l'usuari executant la quarta opció del botí.

Quina funcionalitat vols executar? **E**

Entra el valor mínim a cercar: **500.000**

Entra el valor màxim a cercar: **190.000.000**

S'han trobat 2 tresors en aquest rang:

El tresor d'en Cortés - 7.500.000 doblons
El tresor perdut de Lima - 180.000.000 doblons

Snippet 13: Interacció amb l'usuari executant la cinquena opció del botí.

3.4 MENÚ DE LA COBERTA (ARBRES R)

Un cop l'usuari esculli l'opció de la coberta, mostrarem el menú de l'[snippet 14](#).

```
Escull una opció: 3

A. Afegir tresor
B. Eliminar tresor
C. Visualitzar
D. Cerca per àrea
E. Cerca per proximitat

F. Tornar enrere

Quina funcionalitat vols executar?
```

Snippet 14: Menú per gestionar la coberta.

Com podeu imaginar, quan l'usuari sel·leccioni l'opció F caldrà tornar a mostrar el menú inicial, vist a l'[snippet 1](#).

Els [snippets 15](#), [16](#), [17](#), [18](#) i [19](#) exemplifiquen el que hauria de passar a la resta de les funcionalitats.

```
Quina funcionalitat vols executar? A

Entra el nom del tresor a afegir: La Chouette d'Or
Entra la coordenada X de la posició del tresor a afegir: 4.2
Entra la coordenada Y de la posició del tresor a afegir: 9.9

El tresor s'ha afegit correctament a la coberta.
```

Snippet 15: Interacció amb l'usuari executant la primera opció de la coberta.


```
Quina funcionalitat vols executar? B

Entra el nom del tresor a eliminar: La Chouette d'Or

El tresor s'ha eliminat correctament de la coberta.
```

Snippet 16: Interacció amb l'usuari executant la segona opció de la coberta.

```
Quina funcionalitat vols executar? C

Generant la representació de la coberta...
```

Snippet 17: Interacció amb l'usuari executant la tercera opció de la coberta. El format és lliure.

```
Quina funcionalitat vols executar? D

Entra del primer punt del rectangle (X,Y): 1.0,1.0
Entra del segon punt del rectangle (X,Y): 3.0,3.0

S'han trobat 2 tresors en aquesta àrea:

    Golden Hare (1.3, 2.6)
    The Masquerade (2.6, 1.3)
```

Snippet 18: Interacció amb l'usuari executant la quarta opció de la coberta.

```
Quina funcionalitat vols executar? E

Entra el nombre de tresors a trobar: 3
Entra el punt a on cercar (X,Y): 5.0,5.0

Els 3 tresors més propers a aquest punt són:

Thucydides (4.9, 5.6)
Xenophon (5.9, 4.8)
The Secret (4.3, 3.8)
```

Snippet 19: Interacció amb l'usuari executant la cinquena opció de la coberta.

3.5 MENÚ DE LA TRIPULACIÓ (TAULES)

Un cop l'usuari esculli l'opció de la tripulació, mostrarem el menú de l'[snippet 20](#).

```
Escull una opció: 4

A. Afegir pirata
B. Eliminar pirata
C. Consultar pirata
D. Histograma per edats

E. Tornar enrere

Quina funcionalitat vols executar?
```

Snippet 20: Menú per gestionar la tripulació.

Com podeu imaginar, quan l'usuari sel·leccioni l'opció E caldrà tornar a mostrar el menú inicial, vist a l'[snippet 1](#).

Els [snippets 21](#), [22](#), [23](#) i [24](#) exemplifiquen el que hauria de passar a la resta de les funcionalitats.

Quina funcionalitat vols executar? **A**

Entra el nom del pirata a afegir: **Jericho Swain**

Entra l'edat del pirata a afegir: **50**

Entra el rol del pirata a afegir: **Grand General**

El pirata s'ha afegit correctament a la tripulació.

Snippet 21: Interacció amb l'usuari executant la primera opció de la tripulació.

Quina funcionalitat vols executar? **B**

Entra el nom del pirata a eliminar: **Jericho Swain**

El pirata s'ha eliminat correctament de la tripulació. F.

Snippet 22: Interacció amb l'usuari executant la segona opció de la tripulació.

Quina funcionalitat vols executar? **C**

Entra el nom del pirata a consultar: **Jericho Swain**

Nom: Jericho Swain

Edat: 50

Rol: Grand General

Snippet 23: Interacció amb l'usuari executant la tercera opció de la tripulació.

Quina funcionalitat vols executar? **D**

Generant histograma...

Snippet 24: Interacció amb l'usuari executant la quarta opció de la tripulació. El format és lliure.

4 CONSIDERACIONS

4.1 GRUPS

El projecte es realitzarà en grups de 4. Un cop s'estableixi el grup, cal comunicar-ho per correu al professor de l'assignatura (Pol Muñoz - pol.munoz@salle.url.edu). Per desfer grups, referiu-vos a la normativa de pràctiques.

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 pel grup. Si es programa en C, haurà de compilar, executar i funcionar correctament als servidors de la universitat.

4.3 ESTRUCTURES DE DADES

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 (l·listes, 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 un estudi dels seus costos asimptòtics quan s'analitzin els resultats.

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 CÒPIA

Una còpia 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 altre grup o d'internet es considerarà **acció premeditada**. Per aquests motius, aplicant la [normativa de còpies de la universitat](#), es considerarà una **falta molt greu**.

5 AVALUACIÓ

El projecte es realitzarà durant el segon semestre, creixent a mesura que s'adquireixi el coneixement necessari per a dur a terme cadascun dels apartats.

L'avaluació del projecte tindrà els següents components:

- **Seguiment:** L'equip de l'assignatura mentoritzarà els grups, realitzant un seguiment setmana a setmana per promoure el treball continuat.
- **Codi:** Principalment es tindrà en compte el funcionament de la solució implementada, tot i que s'anima a l'alumnat a tenir cura de la qualitat d'aquesta.
- **Memòria:** El treball de tot el grup durant el semestre haurà de ser documentat en forma de memòria. Els seus continguts es descriuen a l'apartat [5.5](#).
- **Entrevista:** Es realitzarà una entrevista de coneixements a tots els grups que entreguin el projecte.

Cal aprovar les quatre parts per separat, altrament la nota del projecte serà de 4 i caldrà recuperar-lo per aprovar l'assignatura.

5.1 CHECKPOINTS

Com a part del mecanisme de seguiment es realitzaran *checkpoints* de forma periòdica. Uns dies després d'acabar la teoria de cada apartat (i amb previ avís) es realitzarà un exercici d'avaluació *peer to peer*, i el professor posarà una nota al grup segons l'estat del projecte.

5.2 DATA D'ENTREGA

El projecte constarà de dues entregues, una pel codi i una altra per la memòria. Les dates límit per entregar cadascuna de les parts són el **23 de maig de 2021 a les 23:55h** i el **30 de maig de 2021 a les 23:55h**, respectivament.

5.3 RECUPERACIÓ

Si l'entrega no es realitza a temps (NP) o es suspèn, el projecte haurà de ser entregat de nou durant la convocatòria extraordinària. La data límit serà el **30 de juny del 2021 a les 23:55h**, i s'optarà a obtenir una nota màxima de 7.

En aquest escenari la memòria s'entregarà el mateix dia que el codi, i no hi haurà nota de seguiment. També es realitzarà una entrevista de coneixements a tots els grups que entreguin.

5.4 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.

5.5 CONTINGUTS 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 els avantatges que proporciona.
- 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 690:2010 o APA 6th).

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 decisions que preneu, els problemes que teniu, els resultats que observeu...

La memòria ha d'estar escrita en **llenguatge formal**. Aquesta és igual o més important que la implementació del treball, ja que reflexa el coneixement adquirit al projecte i l'assignatura.

Es tindrà en compte que el contingut de la memòria **prioritzi la qualitat abans que la quantitat**. És altament recomanable fer servir \LaTeX per escriure la memòria, tot i que no afectarà a la nota.