

DOCUMENT DE PROVES

PROJECTE HIBERNATE

Héctor Lara,
Marc Sánchez,
Eric Morales
2n DAMvi A
2024-2025

ÍNDEX

1. Menú.....	1
2. Relacions i càrrega de dades a la BD.....	3
3. El joc.....	10

1. Menú

El nostre projecte té un menú per poder jugar i on tens diferents opcions a escollir. Aquest menú s'inicia quan s'executa el joc:

```
Benvingut a Bang! Què vols fer?  
1-> Jugar partida :)  
2-> Veure els jugadors que tenen un personatge associat  
3-> Finalitzar la partida  
4-> El jugador agafa una carta  
5-> Sortir del menu :(
```

Si es posa un "1" a la consola, s'executarà el joc (primer et demanarà la quantitat de jugadors):

```
1  
Quants jugadors vols a la simulació? (Entre 4 i 7 jugadors)  
4
```

```
Ha guanyat: Jugadors4 amb el rol: MALFACTOR!!  
Ha guanyat: Jugadors1 amb el rol: RENEGAT!!  
Ha guanyat: Jugadors2 amb el rol: MALFACTOR!!
```

Com podem veure, s'ha executat el joc i ha guanyat els malfactors i els renegats :(

Si es posa un "2", sortiran aquells jugadors amb personatges:

```
2  
Mostrant els jugadors que tenen un personatge associat!  
7  
Jugador: Jugadors{idJugador=1, nom='Jugadors1', guanyats=0, posicio=0}  
Jugador: Jugadors{idJugador=2, nom='Jugadors2', guanyats=0, posicio=0}  
Jugador: Jugadors{idJugador=3, nom='Jugadors3', guanyats=0, posicio=0}  
Jugador: Jugadors{idJugador=4, nom='Jugadors4', guanyats=0, posicio=0}  
Jugador: Jugadors{idJugador=5, nom='Jugadors5', guanyats=0, posicio=0}  
Jugador: Jugadors{idJugador=6, nom='Jugadors6', guanyats=0, posicio=0}  
Jugador: Jugadors{idJugador=7, nom='Jugadors7', guanyats=0, posicio=0}
```

Si es posa un "3", es finalitzarà la partida:

```
Benvingut a Bang! Què vols fer?  
1-> Jugar partida :)  
2-> Veure els jugadors que tenen un personatge associat  
3-> Finalitzar la partida  
4-> El jugador agafa una carta  
5-> Sortir del menu :(  
3  
Ha guanyat: Jugadors3 amb el rol: XERIF!!  
ACABA EL JOC
```

Si es posa un "4", un jugador agafarà una carta:

```
Benvingut a Bang! Què vols fer?
1-> Jugar partida :)
2-> Veure els jugadors que tenen un personatge associat
3-> Finalitzar la partida
4-> El jugador agafa una carta
5-> Sortir del menu :(
4
El jugador Provisional ha agafat una carta: Cartes{idCarta=60, palCarta=DIAMANTS, numeroCarta=15}
```

Si es posa un “5”, s’acabarà l’execució del programa:

```
Benvingut a Bang! Què vols fer?
1-> Jugar partida :)
2-> Veure els jugadors que tenen un personatge associat
3-> Finalitzar la partida
4-> El jugador agafa una carta
5-> Sortir del menu :(
5
dic 04, 2024 12:07:28 A. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/bang?createDatabaseIfNotExist=true]
FINAL PARTIDA
```

2. Relacions i càrrega de dades a la BD

Nosaltres tenim un mètode carregar() que el que fa és omplir la BD amb dades a l'inici de l'execució per poder utilitzar les possibles opcions del menú.

```
public static void Carregar() { 2 usages 1 Keita58 +2
    ICartaDAO crtDAO = (ICartaDAO) daoFactory.create("carta");
    ITipusCartaDAO tcrDAO = (ITipusCartaDAO) daoFactory.create("tipusCarta");
    IJugadorDAO jDAO = (IJugadorDAO) daoFactory.create("jugador");
    IPersonatgeDAO pDAO = (IPersonatgeDAO) daoFactory.create("personatge");
    IArmaDAO armDAO = (IArmaDAO) daoFactory.create("arma");
    IPartidaDAO partDAO = (IPartidaDAO) daoFactory.create("partida");
    IRolDAO rDAO = (IRolDAO) daoFactory.create("rol");

    // Creem tots els personatges i els afegim.
    Personatges bart = new Personatges(nom: "Bart Cassidy", "g5".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges black = new Personatges(nom: "Black Jack", "descripció de prova".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges calamity = new Personatges(nom: "Calamity Janet", "Soc en Calamity Janet".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges gringo = new Personatges(nom: "El Gringo", "Soc en Gringo".getBytes(StandardCharsets.UTF_8), bales: 3);
    Personatges jesse = new Personatges(nom: "Jesse Jones", "Soc en Jesse Jones".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges jourdonnais = new Personatges(nom: "Jourdonnais", "Soc en Jourdonnais".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges kit = new Personatges(nom: "Kit Carlson", "Soc en Kit Carlson".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges lucky = new Personatges(nom: "Lucky Duke", "Soc en Lucky Duke".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges paul = new Personatges(nom: "Paul Regret", "Soc en Paul Regret".getBytes(StandardCharsets.UTF_8), bales: 3);
    Personatges pedro = new Personatges(nom: "Pedro Ramirez", "Soc en Pedro Ramirez".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges rose = new Personatges(nom: "Rose Dolan", "Soc na Rose Dolan".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges sid = new Personatges(nom: "Sid Ketchum", "Soc en Sid Ketchum".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges slab = new Personatges(nom: "Slab 'el Asesino'", "Soc en Slab".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges lucy = new Personatges(nom: "Lucy Lafayette", "Soc na Lucy Lafayette".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges buitre = new Personatges(nom: "Buitre' Sam", "Soc en Sam".getBytes(StandardCharsets.UTF_8), bales: 4);
    Personatges willy = new Personatges(nom: "Willy 'El Niño'", "Soc en Willy".getBytes(StandardCharsets.UTF_8), bales: 4);
    pDAO.create(bart);
    pDAO.create(black);
    pDAO.create(calamity);
    pDAO.create(gringo);
    pDAO.create(jesse);
    pDAO.create(jourdonnais);
    pDAO.create(kit);
    pDAO.create(lucky);
    pDAO.create(paul);
    pDAO.create(pedro);
    pDAO.create(rose);
    pDAO.create(sid);
    pDAO.create(slab);
    pDAO.create(lucy);
    pDAO.create(buitre);
```

```
Armes Colt45 = new Armes(nomArma: "Colt 45", distanciaArma: 1);
Armes AllenThurber = new Armes(nomArma: "ALLEN & THURBER PEPPERBOX 6 SHOTS", distanciaArma: 1);
Armes Remington = new Armes(nomArma: "Remington", distanciaArma: 2);
Armes Carabina = new Armes(nomArma: "Rev. Carabina", distanciaArma: 2);
Armes Winchester = new Armes(nomArma: "Winchester", distanciaArma: 3);
Armes Sharps = new Armes(nomArma: "Fusil Militar Sharps", distanciaArma: 2);
Armes MaresLeg = new Armes(nomArma: "Rifle 'Mares Leg'", distanciaArma: 1);
armDAO.create(Colt45);
armDAO.create(AllenThurber);
armDAO.create(Carabina);
armDAO.create(Remington);
armDAO.create(Sharps);
armDAO.create(MaresLeg);
armDAO.create(Winchester);

// Creem els rols i els afegim.
Rols rolXerif = new Rols(Rol.XERIF);
Rols rolAjudant = new Rols(Rol.AJUDANT);
Rols rolMalfactor = new Rols(Rol.MALFACTOR);
Rols rolRenegat = new Rols(Rol.RENEGAT);
rDAO.create(rolXerif);
rDAO.create(rolAjudant);
rDAO.create(rolRenegat);
rDAO.create(rolMalfactor);

// Creem una partida per una de les proves del menú.
Partides joc = new Partides(finalitzada: true, LocalDateTime.now(), LocalDateTime.now());
partDAO.create(joc);
```

```

List<Cartes> cartes = new ArrayList<>();
for (int i = 1; i < 28; i++){
    Cartes cartaP = new Cartes(Pal.PIQUES, i);
    cartes.add(cartaP);
    crtDAO.create(cartaP);

    Cartes cartaT = new Cartes(Pal.TREBOLS, i);
    cartes.add(cartaT);
    crtDAO.create(cartaT);

    Cartes cartaC = new Cartes(Pal.CORS, i);
    cartes.add(cartaC);
    crtDAO.create(cartaC);

    Cartes cartaD = new Cartes(Pal.DIAMANTS, i);
    cartes.add(cartaD);
    crtDAO.create(cartaD);
}
Collections.shuffle(cartes);

// Creem tots els tipus de cartes del joc i els afegim.
TipusCartes bang = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "Bang!", Color.MARRO);
tcrDAO.create(bang);
TipusCartes miraTelescopica = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "MiraTelescòpica", Color.BLAU);
tcrDAO.create(miraTelescopica);
TipusCartes fallat = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "Has Fallat!", Color.MARRO);
tcrDAO.create(fallat);
TipusCartes panic = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "Pànic", Color.MARRO);
tcrDAO.create(panic);
TipusCartes ingenua = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "Ingenua", Color.MARRO);
tcrDAO.create(ingenua);
TipusCartes esquivar = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "Esquivar", Color.BLAU);
tcrDAO.create(esquivar);
TipusCartes indis = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "Indis", Color.BLAU);
tcrDAO.create(indis);
TipusCartes cervesa = new TipusCartes( deixarCarta: false, agafarCarta: false, nom: "Cervesa", Color.BLAU);
tcrDAO.create(cervesa);

```

Com podem veure primer creem els personatges i els inserim a la base de dades. Seguidament, creem les armes, els rols i una partida (aquesta partida serveix per a una de les proves del menú). Tot això ho inserim a la base de dades. El mateix amb les cartes i els tipus de carta.

A continuació fem la primera relació de la partida:

```

int ca = 0;
for(Cartes carta : cartes){
    if(ca < 30) {
        bang.getCartes().add(carta);
        carta.setCartaTipusCarta(bang);
        tcrDAO.update(bang);
    }
    else {
        switch (ca%7) {
            case 0:
                miraTelescopica.getCartes().add(carta);
                carta.setCartaTipusCarta(miraTelescopica);
                tcrDAO.update(miraTelescopica);
                break;
            case 1:
                fallat.getCartes().add(carta);
                carta.setCartaTipusCarta(fallat);
                tcrDAO.update(fallat);
                break;
            case 2:
                panic.getCartes().add(carta);
                carta.setCartaTipusCarta(panic);
                tcrDAO.update(panic);
                break;
            case 3:
                ingenua.getCartes().add(carta);
                carta.setCartaTipusCarta(ingenua);
                tcrDAO.update(ingenua);
                break;
            case 4:
                esquivar.getCartes().add(carta);
                carta.setCartaTipusCarta(esquivar);
                tcrDAO.update(esquivar);
                break;
            case 5:
                indis.getCartes().add(carta);
                carta.setCartaTipusCarta(indis);
                tcrDAO.update(indis);
                break;
            case 6:
                cervesa.getCartes().add(carta);
                carta.setCartaTipusCarta(cervesa);
                tcrDAO.update(cervesa);
                break;
        }
    }
    crtDAO.update(carta);
    ca++;
}

```

```
// Creem tots els jugadors a la base de dades, els quals anirem escollint
// aleatòriament per jugar.
HashSet<Jugadors> ju = new HashSet<>();
for(int i = 1; i < 8; i++) {
    Jugadors j = new Jugadors( nom: "Jugadors" + i);
    jDAO.create(j);
    j.setPartidesPropies(Set.of(joc));
    ju.add(j);
    jDAO.update(j);
}
joc.setPartidaJugador(ju);
partDAO.update(joc);

RepartirPersonatges( numJugadors: 7);
RepartirRol( numJugadors: 7);
System.out.println("FINAL CARREGAR");
```

En aquesta part li assignem a cada carta un tipus i a cada tipus una carta, ja que al nostre projecte totes les relacions són bidireccionals. Finalment, fem un update de la carta. A continuació creem els jugadors i els hi posem la partida creada anteriorment i viceversa. Seguidament cridem a la funció RepartirPersonatges on fem el següent:

```
for(Jugadors j : jugadorsList){
    int randomNumber = r.nextInt( origin: 0, personatgesList.size());
    j.setPersonatgeDelJugador(personatgesList.get(randomNumber));
    personatgesList.get(randomNumber).setPersonatgeAmbJugador(j);
    pDAO.update(personatgesList.get(randomNumber));
    jDAO.update(j);
    personatgesList.remove(randomNumber);
}
```

A cada jugador que estigui jugant li repartim un personatge. A repartir rols fem un shuffle dels jugadors i segons el nombre de jugadors repartim una quantitat de rols:


```

switch (numJugadors) {
    case 4:
        for(Rols rol: rolsList){
            if(rol.getNomRol().equals(Rol.XERIF)) {
                jugadors.get(0).setRolJugador(rol);
            }
            else if(rol.getNomRol().equals(Rol.RENEGAT)) {
                jugadors.get(1).setRolJugador(rol);
            }
            else if(rol.getNomRol().equals(Rol.MALFACTOR)) {
                jugadors.get(2).setRolJugador(rol);
                jugadors.get(3).setRolJugador(rol);
            }
        }
        break;
    case 5:
        for(Rols rol: rolsList){
            if(rol.getNomRol().equals(Rol.XERIF)) {
                jugadors.get(0).setRolJugador(rol);
            }
            else if(rol.getNomRol().equals(Rol.RENEGAT)) {
                jugadors.get(1).setRolJugador(rol);
            }
            else if(rol.getNomRol().equals(Rol.MALFACTOR)) {
                jugadors.get(2).setRolJugador(rol);
                jugadors.get(3).setRolJugador(rol);
            }
            else if(rol.getNomRol().equals(Rol.AJUDANT)) {
                jugadors.get(4).setRolJugador(rol);
            }
        }
        break;
}

```

Una vegada vist tot això, quan s'executa aquest mètode la base de dades queda tal que així:

idCarta	idJugador	idTipusCartes	numeroCarta	palCarta
7	NULL	3	2	CORS
8	NULL	1	2	DIAMANTS
9	NULL	4	3	PIQUES
10	NULL	2	3	TREBOLS
11	NULL	6	3	CORS
12	NULL	2	3	DIAMANTS
13	NULL	7	4	PIQUES
14	NULL	7	4	TREBOLS

Com es pot veure, carta s'ha iniciat correctament i amb la relació amb tipusCarta.

AgafarCarta	DeixarCarta	Distancia	PuntsVida	idTipusCarta	Nom	Color
0	0	0	0	1	Bang!	MARRO
0	0	0	0	2	Miratelescòpica	BLAU
0	0	0	0	3	Has Fallat!	MARRO
0	0	0	0	4	Pànic	MARRO
0	0	0	0	5	Ingenua	MARRO
0	0	0	0	6	Esquivar	BLAU
0	0	0	0	7	Indis	BLAU
0	0	0	0	8	Cervesa	BLAU
NULL	NULL	NULL	NULL	NULL	NULL	NULL

I tipusCarta també s'ha inicialitzat correctament amb la relació.

Distancia	idArma	nomArma	descripcioArma
1	1	Colt 45	NULL
1	2	ALLEN & THURBER PEPPERBOX 6 SHOTS	NULL
2	3	Rev. Carabina	NULL
2	4	Remington	NULL
2	5	Fusil Militar Sharps	NULL
1	6	Rifle 'Mares Leg	NULL
3	7	Winchester	NULL
NULL	NULL	NULL	NULL

Guanyats	Posicio	idArma	idJugador	idRol	Nom
0	0	NULL	1	4	Jugadors1
0	0	NULL	2	1	Jugadors2
0	0	NULL	3	2	Jugadors3
0	0	NULL	4	4	Jugadors4
0	0	NULL	5	4	Jugadors5
0	0	NULL	6	2	Jugadors6
0	0	NULL	7	3	Jugadors7

Jugador té correctament el rol (la posició no la té perquè en aquest punt no s'està jugant)

Finalitzada	idPartida	DataFinal	DataInici
1	1	2024-12-03 23:35:00.450654	2024-12-03 23:35:00.450654

idJugador	idPartida
1	1
2	1
3	1
4	1
5	1
6	1
7	1

Com es pot veure, la taula intermitja entre partida y jugador també s'ha creat correctament.

Bales	idJugador	idPersonatge	Nom	Descripcio
4	NULL	1	Bart Cassidy	BLOB
4	NULL	2	Black Jack	BLOB
4	1	3	Calamity Janet	BLOB
3	2	4	El Gringo	BLOB
4	7	5	Jesse Jones	BLOB
4	NULL	6	Jourdonnais	BLOB
4	3	7	Kit Carlson	BLOB
4	NULL	8	Lucky Duke	BLOB
3	5	9	Paul Regret	BLOB
4	NULL	10	Pedro Ramírez	BLOB
4	4	11	Rose Dolan	BLOB
4	NULL	12	Sid Ketchum	BLOB
4	6	13	Slab 'el Asesino'	BLOB
4	NULL	14	Lucy Lafayette	BLOB
4	NULL	15	'Buitre' Sam	BLOB
4	NULL	16	Willy 'El Niño'	BLOB

Els personatges tenen els jugadors.

idRol	Descripcio	NomRol
1	NULL	XERIF
2	NULL	AJUDANT
3	NULL	RENEGAT
4	NULL	MALFACTOR

Com es pot veure, totes les taules s'han creat correctament.

3. El joc

El primer que fem en iniciar una partida és preguntar la quantitat de jugadors que hi haurà a la partida.

```
do {
    System.out.println("Quants jugadors vols a la simulació? (Entre 4 i 7 jugadors)");
    j = Integer.parseInt(lectura.nextLine());
} while(j > 8 || j < 3);
```

```
Quants jugadors vols a la simulació? (Entre 4 i 7 jugadors)
4
```

Seguidament, executem el mètode TornarAJugar(int), on ens descuidem del nombre de jugadors que ha escrit el jugador per al terminal i afegim a aquests jugadors una nova partida, després realitzem la resta de mètodes on repartim armes, personatges, rols, cartes, partides i posicions... Bàsicament on fem totes les relacions i acabem:

```
TornarAJugar(j);
RepartirArma(j);
RepartirRol(j);
RepartirJugadors(j);
RepartirPersonatges(j);
RepartirCartes(j);
BucleJoc(j);
```

Per tant, quan iniciem una partida podem veure que s'executen aquests mètodes abans de començar el joc:

```
INICI REPARTIR ARMES
FINAL REPARTIR ARMES
INICI REPARTIR ROLS
FINAL REPARTIR ROLS
INICI REPARTIR POSICIONS
```

```
FINAL REPARTIR POSICIONS
INICI REPARTIR PERSONATGE
FINAL REPARTIR PERSONATGE
INICI REPARTIR CARTES
FINAL REPARTIR CARTES
COMENÇA EL JOC
```

Una vegada comença el joc (els jugadors ja tenen les posicions, els rivals i els rols...), el jugador al qual li toca el torn agafa dues cartes aleatòries de les quals no tenen cap relació amb cap jugador. Seguidament, el jugador juga 4 cartes: la primera és un Bang! contra el jugador 3, però com que tenia un Has fallat! no li fa cap mal. La segona és una Mira telescòpica contra el jugador 2, la qual disminueix la distància del rival en 1. Finalment, la tercera i última carta són cerveses, les quals curen al personatge +1 punt de vida (sempre que tinguin menys vida que la màxima).

```
El torn és del jugador Jugadors1. Té actualment 4 vides.  
El jugador Jugadors1 ha agafat una carta: Cartes{idCarta=79, palCarta=CORS, numeroCarta=20}  
El jugador Jugadors1 ha agafat una carta: Cartes{idCarta=73, palCarta=PIQUES, numeroCarta=19}  
El jugador Jugadors1 jugarà 4 cartes de les 10 cartes que té:  
Ha jugat un BANG! contra el jugador Jugadors3! Quina mala baba.  
L'enemic ara té 3 vides.  
L'enemic tenia un Has Fallat! No ha servit de res el BANG :(  
Ha jugat una Miratelescòpica contra el jugador Jugadors2! Està maquinant alguna cosa...  
Ha tirat una Cervesa. Mai va malament emborratxar-se una mica.  
Ha tirat una Cervesa. Mai va malament emborratxar-se una mica.
```

Aquest és el torn d'un jugador. A continuació es mostra què passa quan s'ha jugat una ronda, és a dir, tots els jugadors, que són vius, han jugat un torn:

```
El jugador Jugadors1 ha deixat una carta.  
El jugador Jugadors1 ha deixat una carta.  
El jugador Jugadors1 ha deixat una carta.  
El jugador Jugadors1 ha deixat una carta.  
El jugador Jugadors4 ha deixat una carta.  
El jugador Jugadors4 ha deixat una carta.  
El jugador Jugadors4 ha deixat una carta.  
El jugador Jugadors4 ha deixat una carta.  
El jugador Jugadors4 ha deixat una carta.  
El jugador Jugadors4 ha deixat una carta.
```

Com es pot veure, els jugadors 4 i 1 han deixat 7 i 4 cartes respectivament. Això és perquè els jugadors, si tenen més cartes que la seva vida màxima, han de descartar cartes fins a tenir tantes cartes com vides té el seu personatge. És a dir, si el jugador té 4 vides i té 8 cartes, haurà de descartar 4 cartes. Això es fa amb aquesta funció amb la qual comprovem primer que la vida del jugador sigui més gran que 0 i que el jugador tingui més cartes que vides. Després fem una resta entre la vida del jugador i la quantitat de cartes que té, tot això dins d'un for. Seguidament per cada iteració cridem a DeixarCartes:

```
for(Jugadors j : jList) {  
    if(j.getPersonatgeDelJugador().getBales() > 0 && j.getCartes().size() > j.getPersonatgeDelJugador().getBales()) {  
        int cartes = j.getCartes().size();  
        for(int i = 0 ; i < (Math.abs(j.getPersonatgeDelJugador().getBales() - cartes)); i++) {  
            DeixarCartes(j);  
        }  
    }  
}
```

```

public static void DeixarCartes(Jugadors j) { 2 usages 1 Keita58 +1
    DAOFactory daoFactory = DAOFactoryImpl.getFactory( choice: "MySQL");

    ICartaDAO cDAO = (ICartaDAO) daoFactory.create("carta");
    IJugadorDAO jDAO = (IJugadorDAO) daoFactory.create("jugador");

    List<Cartes> cartes = cDAO.getCartesJugador(j);
    Collections.shuffle(cartes);

    if(!j.getCartes().isEmpty()) {
        cartes.get(0).setCartesJugador(null);
        j.getCartes().remove(cartes.get(0));
        cDAO.update(cartes.get(0));
        jDAO.update(j);
        System.out.println("El jugador " + j.getNom() + " ha deixat una carta.");
    }
    else {
        System.out.println("El jugador no té cartes per descartar :(");
    }
}
}

```

En aquesta funció bàsicament agafem les cartes del jugador, les barregem i deixem una aleatòria.

Si un jugador perd totes les vides, recalculem les distàncies entre els jugadors amb la funció CalcularDistància següent:

```

for(Jugadors j : jugadorsList) {
    for(Jugadors k : jugadorsList) {
        // Creem dos bucles, per parsejar tots els jugadors entre ells.
        if(!j.equals(k)) {
            // Si els jugadors són diferents (no és un jugador contras si mateix) creem el valor aux de JugadorRivalsId.
            JugadorRivalsId aux = new JugadorRivalsId(k, j);
            if(a > 0) {
                // Mirem dins del set creat abans per si ja existeix la dupla d'enemics que estem intentant afegir.
                for (JugadorsRivals jr : enemies) {
                    // Aquí comprovem si el JugadorRivals existent no és el mateix que volem afegir però en l'ordre invers. Si ho és passem al següent.
                    if (!(jr.getIdRival().getIdJugador().getIdJugador() == k.getIdJugador() && jr.getIdRival().getIdRival().getIdJugador() == j.getIdJugador())) {
                        if (Math.abs(j.getPosicio() - k.getPosicio()) < jugadorsList.size()/2) {
                            enemies.add(new JugadorsRivals(aux, Math.abs(j.getPosicio() - k.getPosicio())));
                        } else {
                            enemies.add(new JugadorsRivals(aux, DistanciaRival: jugadorsList.size() - Math.abs(j.getPosicio() - k.getPosicio())));
                        }
                        break;
                    }
                }
            }
            else {
                // Aquí només entrarem la primera vegada amb el primer jugador, perquè no hi haurà cap valor al set en haver-lo acabat de crear.
                if (Math.abs(j.getPosicio() - k.getPosicio()) < jugadorsList.size()/2) {
                    enemies.add(new JugadorsRivals(aux, Math.abs(j.getPosicio() - k.getPosicio())));
                } else {
                    enemies.add(new JugadorsRivals(aux, DistanciaRival: jugadorsList.size() - Math.abs(j.getPosicio() - k.getPosicio())));
                }
            }
        }
    }
    j.setJugadorsRivals(enemies);
    a++;
}
}

```

Per cada torn de jugador, es comprova que el jugador actual no estigui mort, i si ho està (perquè l'han matat en un torn anterior al seu) es comprova quin rol té; si és el xèrif s'acaba la partida immediatament, si és un renegat o un malfactor s'acumula en un comptador que utilitzem per poder saber si s'han eliminat tots els dolents de la partida.

També comprovem si queden dos jugadors vius, en aquest cas mirem si aquests dos no són un xèrif i un ajudant, si es donés el cas que ho fossin s'acaba la partida.

```
if(jAmbVida.size() == 2) {
    // Si hi ha dos jugadors vius i són el xèrif i l'ajudant s'acaba el joc
    if((jAmbVida.get(0).getRolJugador().getNomRol() == Rol.XERIF || jAmbVida.get(0).getRolJugador().getNomRol() == Rol.AJUDANT) &&
        (jAmbVida.get(1).getRolJugador().getNomRol() == Rol.XERIF || jAmbVida.get(1).getRolJugador().getNomRol() == Rol.AJUDANT)) {
        acabarPartida = true;
        break;
    }
}
//Si el jugador mort és el xèrif s'acaba el joc
if (j.getPersonatgeDelJugador().getBales() == 0 && j.getRolJugador().getNomRol() == Rol.XERIF) {
    acabarPartida = true;
    break;
}
else if(j.getPersonatgeDelJugador().getBales() == 0) {
    boolean trobat = false;
    for(Jugadors jM : jMorts){
        // Si el rol del jugador mort és cap dels dolents i aquest no està dins la llista dels morts l'afegim i el contem.
        if(jM.equals(j)) {
            trobat = true;
            break;
        }
    }
    if(!trobat) {
        jMorts.add(j);
    }
    else {
        // Si hi ha un jugador mort hem de tornar a contar les distàncies dels nous jugadors
        CalcularDistancia(numJugadors);
        trobat = false;
    }
}
if(j.getRolJugador().getNomRol() == Rol.RENEGAT || j.getRolJugador().getNomRol() == Rol.MALFACTOR) {
    for(Jugadors jM : jMorts){
        // Si el rol del jugador mort és cap dels dolents i aquest no està dins la llista dels morts l'afegim i el contem.
        if(jM.equals(j)) {
            trobat = true;
            break;
        }
    }
}
else
    continue;
// Si no ha trobat l'enemic dins la llista de morts l'afegeix i augmenta en 1 la variable dolentsMorts.
if(!trobat) {
    dolentsMorts++;
    jMorts.add(j);
}
}
```

Finalment, en acabar la partida surt el guanyador, el qual només pot ser el xerif i / o l'ajudant o els malefactors i / o renegats i / o ajudant:

```
Ha guanyat: Jugadors1 amb el rol: MALFACTOR!!
ACABA EL JOC
```

Una vegada acabat el joc, surt una altra vegada el menú per si el jugador vol tornar a fer una partida d'aquest trepidant lloc d'acció: el BANG!

```
Ha guanyat: Jugadors1 amb el rol: MALFACTOR!!  
ACABA EL JOC  
  
Benvingut a Bang! Què vols fer?  
1-> Jugar partida :)  
2-> Veure els jugadors que tenen un personatge associat  
3-> Finalitzar la partida  
4-> El jugador agafa una carta  
5-> Sortir del menu :(
```