

# PRÀCTICA Arquitectura del Software

## 1er Lliurament

### QT14-15

- La solució de la pràctica s'ha d'entregar en un únic fitxer .pdf al Racó. Poseu el número de grup que us he assignat al nom del fitxer. Només l'ha d'entregar un dels components del grup.
- Tots els diagrames s'han de poder llegir correctament.
- La data de lliurament de la pràctica és el 27/10/2014

### Enunciat: Joc “Buscamines”

Una empresa que es dedica al desenvolupament de jocs ens ha demanat que dissenyem una part d'una versió del joc *Buscamines*.

El joc enregistra dos tipus d'usuaris: l'administrador i el jugador. L'administrador és l'encarregat de definir els diferents nivells de dificultat i les diferents estratègies de puntuació del joc. El jugador és el que juga partides del joc. El joc consisteix a buidar totes les caselles d'un tauler que no amaguin una mina. El jugador selecciona un nivell de dificultat del joc. Cada nivell de dificultat té un nom (principiant, expert, etc...), el nombre de caselles per fila i columna que tindrà el tauler i el nombre de mines. Algunes caselles tenen un nombre que indica les mines que sumen totes les caselles que hi ha al voltant. Per exemple, si una casella té el nombre 3 vol dir que de les vuit caselles que hi ha al voltant (si no és una cantonada o vora) hi ha 3 amb mines i 5 sense mines. Si es descobreix una casella sense nombre indica que cap de les seves caselles veïnes té mina i aquestes es descobreixen automàticament.

Si es descobreix una casella amb una mina es perd la partida. Es pot posar una marca en les caselles que el jugador pensa que hi ha mines per ajudar a descobrir les que estan a prop.

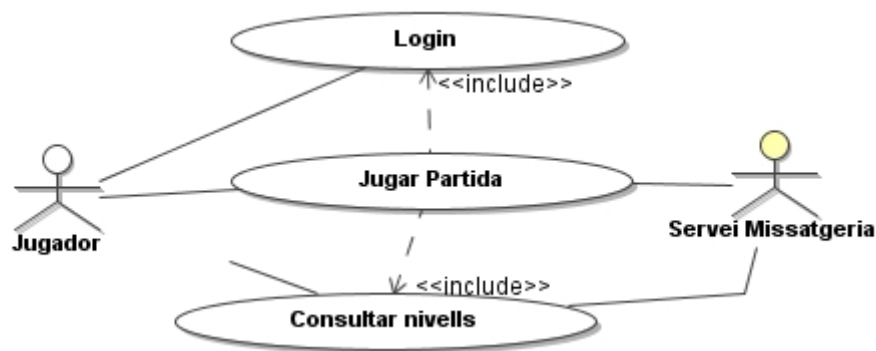
Inicialment el joc té dues estratègies per calcular la puntuació d'una partida guanyada: l'estratègia de puntuació per temps i l'estratègia de puntuació per nombre de tirades. La primera dóna una puntuació inversa a un temps màxim definit per l'administrador. Per exemple, si l'administrador defineix un temps màxim de 1000 segons i un jugador ha guanyat una partida en 450 segons, aquest obtindrà una puntuació de 550 punts. La segona estratègia dóna una puntuació inversa a un nombre de tirades màxim definit per l'administrador. Per exemple, si l'administrador defineix un nombre de tirades màxim de 300 tirades i un jugador ha guanyat una partida en 100 tirades, aquest obtindrà una puntuació de 200 punts. L'estratègia de puntuació s'assigna en el moment de fer la creació d'una partida (mireu la descripció del cas d'ús i els contractes per veure com es decideix l'estratègia de puntuació d'una partida). És important que fem un disseny i una implementació del joc que permeti en el futur afegir noves estratègies de puntuació.

De les diverses funcionalitats que oferirà el joc, ens centrem en tres casos d'ús. A continuació disposeu d'una descripció textual de les funcionalitats, del diagrama de casos d'ús, de la descripció textual dels casos d'ús, del diagrama de seqüència d'esdeveniments del sistema i de l'esquema conceptual de l'especificació:

#### Descripció de les funcionalitats:

- Login: permet a un jugador autenticar-se al sistema per poder jugar.
- Consultar nivells: permet obtenir els nivells disponibles del joc.
- Jugar partida: permet a un jugador jugar una partida del joc.

#### Diagrama de casos d'ús:



#### Descripció textual dels casos d'ús:

**Nom:** Login

**Resum de la funcionalitat:** permet a un jugador autenticar-se al sistema per poder jugar.

**Escenari principal:**

1. El jugador vol utilitzar el sistema.
2. Introdueix el seu usuari i password.
3. El sistema comprova que les dades introduïdes siguin correctes.

**Extensions:**

**3a. El jugador no introdueix un usuari i password vàlid**

- 3a1. El sistema informa a l'usuari que les dades introduïdes no són vàlides.
- 3a2. L'usuari torna a introduir les dades o indica que vol acabar el cas d'ús.

**Nom:** Consultar nivells

**Resum de la funcionalitat:** permet obtenir els nivells disponibles al joc.

**Escenari principal:**

1. El jugador indica al sistema que vol consultar els diferents nivells disponibles al joc.
2. El sistema valida si hi ha nivells.
3. El sistema retorna una el nom, el nombre de caselles per fila, el nombre de caselles per columna i el nombre de mines de tots els nivells disponibles al joc.

**Extensions:**

**2a. No existeixen nivells**

- 2a1. El sistema informa que no hi ha nivells.
- 2a2. S'acaba el cas d'ús

**Nom:** Jugar partida

**Resum de la funcionalitat:** permet a un jugador jugar una partida del joc.

**Escenari principal:**

1. El jugador vol jugar una partida del joc.
2. S'executa el cas d'ús *Login*.
3. El jugador indica que vol jugar una partida.
4. S'executa el cas d'ús *Consultar nivells*.
5. El jugador indica un nivell.
6. El sistema crea la partida amb tota la seva informació, li assigna l'estratègia de puntuació que serà aleatòria i el nivell indicat pel jugador. També es creen les caselles de la partida amb tota la informació necessària per començar el joc. Les caselles on es defineixen les mines es calculen de forma aleatòria.
7. El jugador indica una acció a realitzar (descobrir casella, marcar casella o desmarcar casella) i la casella sobre la que s'ha de realitzar l'acció.
8. El sistema comprova si l'acció és correcta.
9. El sistema enregistra l'acció i informa al jugador.
10. El sistema comprova si s'ha guanyat la partida (s'han descobert totes les caselles sense descobrir les mines).
11. El sistema notifica al jugador que ha guanyat la partida i la puntuació obtinguda segons l'estratègia de puntuació assignada. La partida passarà a ser una partida jugada. A més el sistema envia un correu electrònic al jugador amb la puntuació obtinguda i s'acabarà el cas d'ús. Per enviar el correu, el sistema invoca a un servei extern de missatgeria que fa l'enviament del correu al jugador.

**Extensions:**

**8a. El jugador marca una casella que ja està marcada o una casella descoberta**

- 8a1. El sistema informa al jugador de que l'acció no és vàlida. Es torna a introduir una nova acció (pas 7).

**8b. El jugador desmarca una casella que no està marcada o una casella descoberta**

- 8b1. El sistema informa al jugador de que l'acció no és vàlida. Es torna a introduir una nova acció (pas 7).

**8c. El jugador descobreix una casella que està descoberta o una casella marcada**

- 8c1. El sistema informa al jugador de que l'acció no és vàlida. Es torna a introduir una nova acció (pas 7).

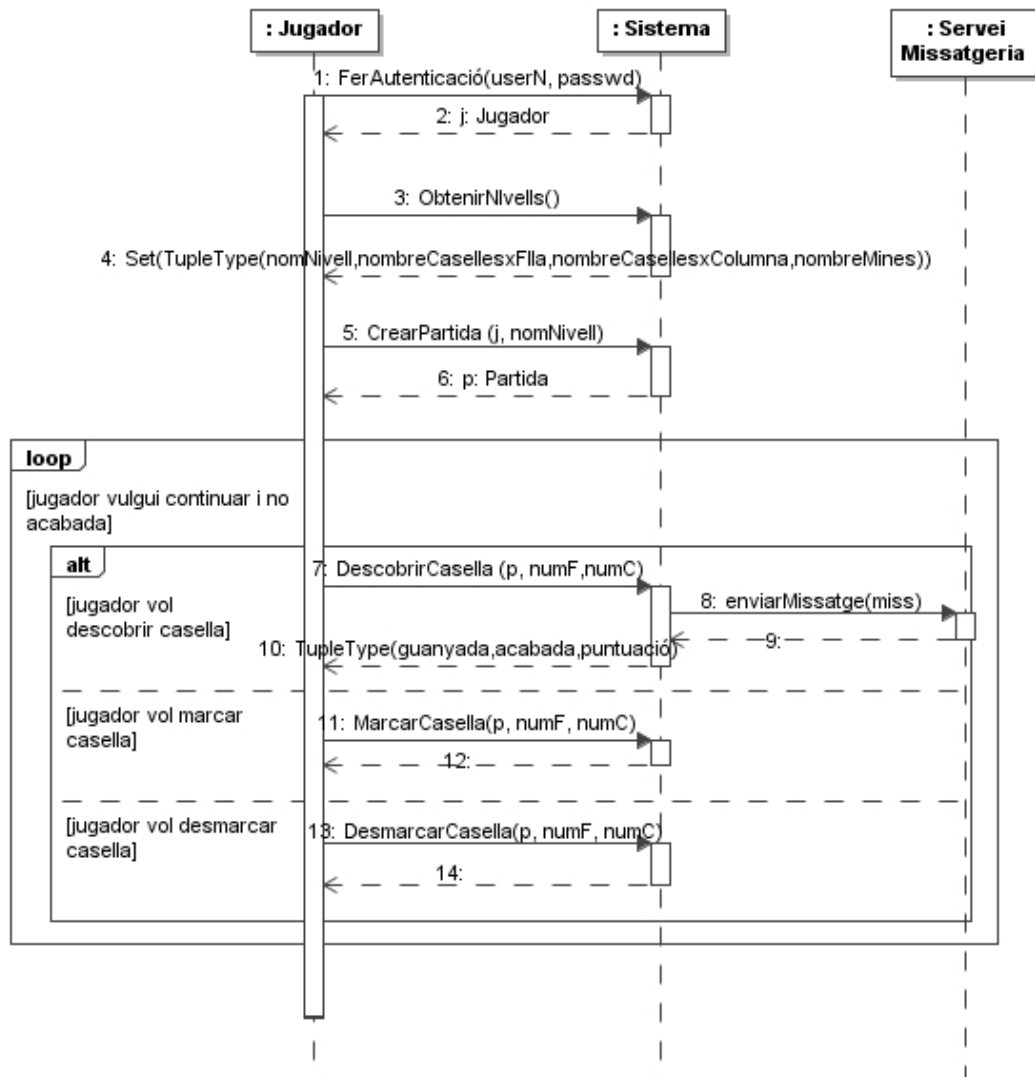
**10a. El jugador ha descobert una casella amb mina**

- 10a1. El sistema informa al jugador de que ha perdut la partida i s'acaba el cas d'ús.

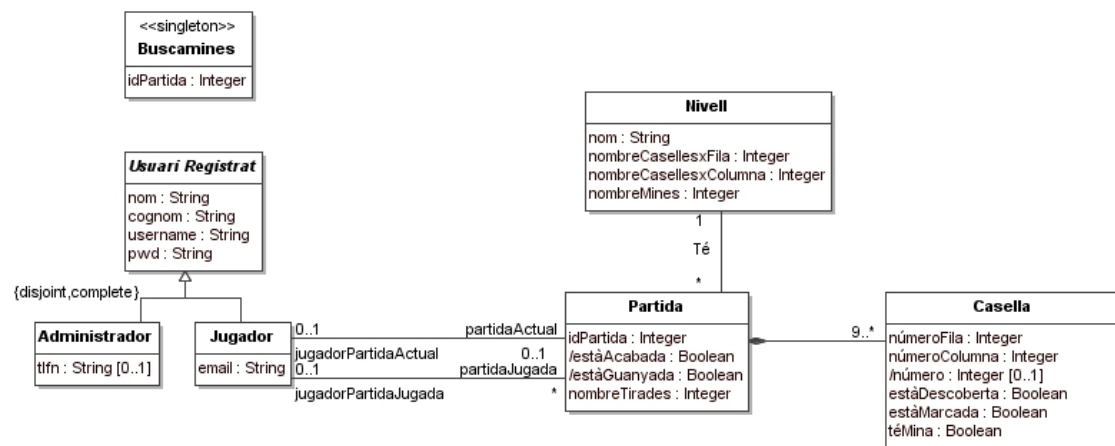
**10b. El jugador no ha descobert una casella amb mina i queden caselles per descobrir**

- 10b1. El jugador indica si vol acabar el cas d'ús.
- 10b2. Si el jugador no vol acabar el cas d'ús es torna al pas 7.

Diagrama de seqüència d'esdeveniments del sistema del cas d'ús Jugar partida:



Esquema conceptual de l'especificació:



Restriccions d'integritat textuais:

1. *Usuari Registrat* s'identifica per *username*.
2. *Jugador* s'identifica també pel seu *email*.
3. *Nivell* s'identifica per *nom*.
4. *Partida* s'identifica per *idPartida*.
5. *Casella* s'identifica per *idPartida*, *númeroFila*, *númeroColumna*.
6. El *nombreCasellesxFila* i el *nombreCasellesxColumna* ha de ser més gran que 3.
7. El nombre de caselles d'una partida ha de ser igual a (*nombreCasellesxFila* \* *nombreCasellesxColumna*).
8. Una partida d'un jugador no pot ser alhora actual i jugada.
9. El *nombreMines* ha de ser més gran que 0 i més petit que 0,4 \* (*nombreCasellesxFila* \* *nombreCasellesxColumna*).
10. El *nombreTirades* d'una partida ha de ser més gran o igual a 0 i l'*idPartida* d'una partida ha de ser més gran o igual a 1.
11. Una casella que té mina no pot tenir número.
12. El *númeroFila* d'una casella d'una partida ha de tenir un valor entre 1 i *númeroCasellesxFila* del nivell de la partida.
13. El *númeroColumna* d'una casella d'una partida ha de tenir un valor entre 1 i *númeroCasellesxColumna* del nivell de la partida.

Altres restriccions no rellevants pel nostre disseny....

Regles de derivació:

1. L'atribut *número* d'una casella té com a valor el número de caselles que hi ha al voltant que tenen mines o no té valor si la casella té una mina.
2. L'atribut *estàAcabada* de la classe *Partida* té com a valor cert si l'atribut *estàGuanyada* té com a valor cert o hi ha alguna casella descoberta amb una mina. En cas contrari, té valor fals.
3. L'atribut *estàGuanyada* de la classe *Partida* té com a valor cert si totes les caselles que no tenen mina estan descobertes i totes les que tenen mina no estan descobertes. En cas contrari, té el valor fals.

Després de fer l'assignació de responsabilitats a capes dels diferents casos d'ús i de materialitzar tota la informació que era derivada, hem obtingut els contractes següents de la capa de domini:

Contracte Capa Domini *Login*:

**context** CapaDomini :: Login(userN: String, passwd: String)

**exc** *usernameNoExisteix*: el username *userN* no existeix.

**exc** *pwdIncorrecte*: el password *passwd* és incorrecte.

**post** *cert*

(La post d'aquest contracte és cert o buida ja que per simplificar no enregistrem al sistema que el jugador ha fet el login)

Contracte Capa Domini *Consultar nivells*:

**context** CapaDomini :: ConsultarNivells(): Set(TupleType(nom: String, nombreCasellesxFila: Integer, nombreCasellesxColumna: Integer, nombreMines: Integer))

**exc** *noHiHaNivells*: no hi ha nivells disponibles al sistema.

**post** *result*= *nom*, *nombreCasellesxFila*, *nombreCasellesxColumna*, *nombreMines* de tots els nivells disponibles.

### Contractes Capa Domini Jugar partida:

**context** CapaDomini :: FerAutenticació (userN: String, passwd: String)

**exc** *usernameNoExisteix*: el username *userN* no existeix.

**exc** *pwdIncorrecte*: el password *passwd* és incorrecte.

**exc** *usuariNoJugador*: el username *userN* no correspon a un jugador.

**post** *emmagatzemaDades*: s'emmagatzema a la capa de domini el jugador amb el username *userN*.

**context** CapaDomini :: ObtenirNivells ():Set(TupleType(nom: String, nombreCasellesxFila: Integer, nombreCasellesxColumna: Integer, nombreMines: Integer))

**exc** *noHiHaNivells*: no hi ha nivells disponibles al sistema.

**post** *result*= *nom*, *nombreCasellesxFila*, *nombreCasellesxColumna*, *nombreMines* de tots els nivells disponibles.

**context** CapaDomini :: CrearPartida (nomNivell: String)

**pre** *nivellExisteix*: el nivell *nomNivell* existeix al sistema.

**post** *creaPartida*: es crea un objecte de la classe *Partida*. L'identificador de partida de l'objecte creat pren per valor l'identificador de partida definit a la classe *Buscamines* més 1. Els atributs *estàAcabada* i *estàGuanyada* prenen per valor fals i l'atribut *nombreTirades* s'inicialitza a 0. S'assigna a la partida l'estratègia de puntuació de forma aleatòria.

**post** *assignaNivell*: s'assigna a la partida el nivell identificat per *nomNivell*.

**post** *creaCaselles*: es creen (*nombreCasellesxFila* \* *nombreCasellesxColumna*) objectes casella on *nombreCasellesxFila* i *nombreCasellesxColumna* corresponen al nivell *nomNivell*. Per a cada casella s'inicialitzarà l'atribut *númeroFila* amb un valor entre 1 i el *nombreCasellesxFila* i s'inicialitzarà l'atribut *númeroColumna* amb un valor entre 1 i el *nombreCasellesxColumna*. Els atributs *estàDescoberta* i *estàMarcada* de totes les caselles s'inicialitzaran a fals. Es calcularà aleatòriament les caselles que tindran mines (tantes com l'atribut *nombreMines* del nivell *nomNivell* indiqui). Aquestes caselles tindran l'atribut *téMina* inicialitzat a cert. La resta de caselles el tindran inicialitzat a fals. L'atribut *número* de cada casella té com a valor el número de caselles que hi ha al voltant que tenen mines o no té valor si la casella té una mina. Els objectes casella creats s'associaran amb la partida creada.

**post** *assignaJugador*: es forma l'associació entre la partida creada (que serà la partida actual) i el jugador emmagatzemat a la capa de domini.

**post** *actualitzaIdPartida*: s'incrementa en 1 l'identificador de partida definit a la classe *Buscamines*.

**post** *emmagatzemaDades*: s'emmagatzema a la capa de domini la partida creada.

**context** CapaDomini :: DescobrirCasella (numF: Integer, numC: Integer): TupleType(guanyada: Boolean, acabada: Boolean, puntuació: Integer [0..1])

**pre** *posicióCorrecta*: el *numF* i el *numC* identifiquen una casella dins de la partida.

**exc** *casellaJaDescoberta*: la casella a descobrir ja està descoberta.

**exc** *casellaMarcada*: la casella a descobrir està marcada.

**post** *casellaDescoberta*: l'atribut *estàDescoberta* de la casella es posa a cert. El *nombreTirades* s'incrementa en 1 unitat. Si el número de la casella té valor i aquest és 0, s'invoca a l'operació *DescobrirCasella* per a totes les caselles que hi ha al voltant.

**post** *comprovaPartidaPerduda*: si la casella a descobrir té una mina llavors l'atribut *estàAcabada* pren per valor cert.

**post** *comprovaPartidaGuanyada*: si la casella a descobrir no té una mina i totes les caselles estan descobertes llavors els atributs *estàAcabada* i *estàGuanyada* prenen per valor cert. El

sistema invoca a l'operació *enviarMissatge(missatge:String)* del Servei de Missatgeria amb un text que conté l'*idPartida* i la puntuació obtinguda a la partida guanyada.

**post** *partidaJugada*: si la partida ha acabat (l'atribut *estàAcabada* és cert) llavors s'elimina la instància de l'associació entre la partida actual i el jugador i es crea la instància de l'associació entre el jugador i la partida amb el rol *partidaJugada*.

**post** *result*= valor dels atributs *estàGuanyada* i *estàAcabada* de la partida i la puntuació de la partida només en el cas d'estar guanyada. Aquesta puntuació es calcula segons l'estratègia de puntuació assignada a la partida.

**context** *CapaDomini :: MarcarCasella (numF: Integer, numC: Integer)*

**pre** *posicióCorrecta*: el *numF* i el *numC* identifiquen una casella dins de la partida.

**exc** *casellaJaMarcada*: la casella a marcar ja està marcada.

**exc** *casellaJaDescoberta*: la casella a marcar ja està descoberta.

**post** *marcarCasella*: l'atribut *estàMarcada* de la casella es posa a cert.

**context** *CapaDomini :: DesmarcarCasella (numF: Integer, numC: Integer)*

**pre** *posicióCorrecta*: el *numF* i el *numC* identifiquen una casella dins de la partida.

**exc** *casellaNoMarcada*: la casella a desmarcar no està marcada.

**exc** *casellaJaDescoberta*: la casella a desmarcar ja està descoberta.

**post** *desmarcarCasella*: l'atribut *estàMarcada* de la casella es posa a fals.

### Es demana:

1.- Hibernate és una eina de lliure distribució que soluciona el problema de la coexistència entre els dos models de dades existents en un sistema software (l'utilitzat pel programa (orientació a objectes)) i l'utilitzat per la base de dades (model relacional). Es demana que utilitzeu Hibernate (amb anotacions) per definir una part de l'esquema conceptual (les classes *Partida* i *Nivell* amb els seus atributs, l'associació entre elles i els getters i els setters de cada classe) i la part corresponent de l'esquema de la BD. En concret, es demana el codi en Java de les classes, l'esquema de la BD i un programa de prova que utilitzi les operacions definides a les classes (poseu el codi en el fitxer pdf). Podeu trobar molta informació sobre Hibernate a internet. Alguns links que us poden ser d'utilitat són: <http://www.davidmarco.es/tutoriales/hibernate-reference/> i <http://www.youtube.com/watch?v=GINvxAaXDbY&feature=related> (són 18 videos on es descriu com instal·lar i configurar Hibernate i com connectar-lo amb Eclipse i la BD. A més podeu veure com utilitzar Hibernate amb anotacions). La instal·lació i configuració de Hibernate i com connectar-lo amb Eclipse/Netbeans i PostgreSQL també la podeu trobar en el manual que us adjunto.

2.- Diagrames de seqüència de les operacions de la capa de domini corresponents al cas d'ús *Jugar partida*. Heu de considerar que aquest cas d'ús utilitza els casos d'ús *Consultar nivells* i *Login*. És valorarà especialment la reutilització dels mateixos. És molt important que per fer l'apartat 2 i 3 entengueu perfectament l'especificació i l'assignació de responsabilitats a capes que s'ha fet.

3.- Diagrama de classes de la capa de domini. Heu de fer servir un controlador de cas d'ús per guardar l'estat del cas d'ús.

4.- (Competència transversal) Justifiqueu de forma raonada els patrons de disseny que heu aplicat en aquesta pràctica.