Exercicis Resolts de Diagrames de Casos d'ús

Disseny de software

Contingut

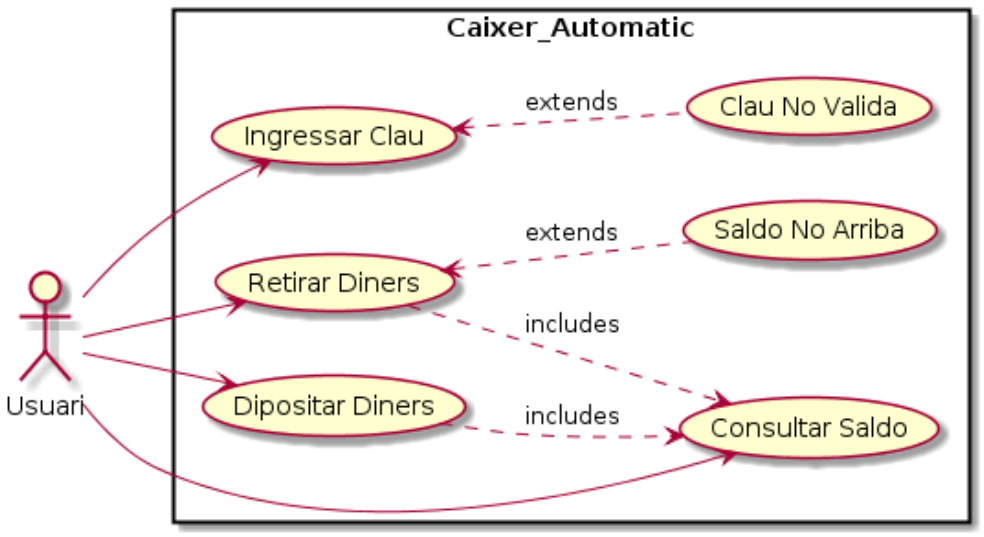
1. Caixer automàtic

El banc **MaloBank** necessita ajuda per modelar el sistema que farà funcionar els seus nous caixers automàtics portàtils. Aquests, del port d'un telèfon públic, li permetran a l'usuari realitzar només les operacions més simples: retirar, dipositar i consultar saldo (ni somiar amb moviments entre comptes o compres de targetes de prepagament telefònic). Per a això tingues en consideració que:

- Es demana ingressar la clau de l'usuari posteriorment al pas de la targeta per la ranura.
- No es pot retirar més fons dels quals realment hi ha, notificant d'aquesta situació a l'usuari.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect

actor Usuari
rectangle Caixer_Automatic {

Usuari -> (Ingressar Clau)
Usuari -> (Retirar Diners)
Usuari ->  (Consultar Saldo)
Usuari -> (Dipositar Diners)

(Dipositar Diners) ..> (Consultar Saldo) : includes
(Retirar Diners) ..> (Consultar Saldo): includes
(Retirar Diners) <.. (Saldo No Arriba): extends
(Ingressar Clau) <.. (Clau No Valida): extends
}


@enduml
```

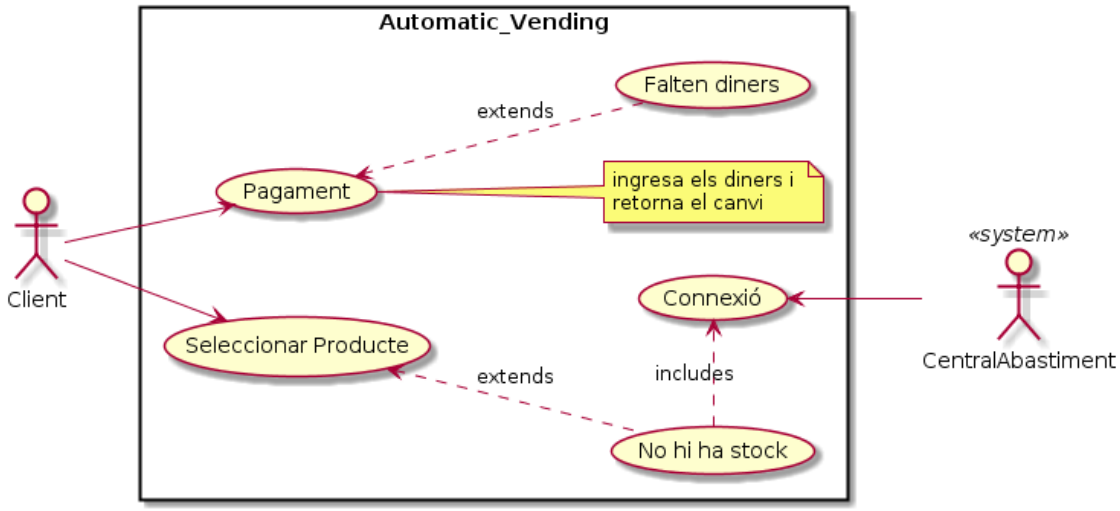


2. Venda Automàtica

L'empresa **AutomaticVending** té plans per instal·lar una nova màquina de venda automàtica intel·ligent a la facultat. Intel·ligent perquè quan detecta que un client intenta comprar un producte esgotat, es connectarà automàticament a la central de proveïment i donarà avís per realitzar la reposició. A més, com a bona màquina de venda automàtica, ha de retornar canvi i no deixar que paguin els clients menys del preu del que està venent.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect
actor Client as client
actor CentralAbastiment <<system>> as central
rectangle Automatic_Vending{

client -> (Pagament)
note right of (Pagament)
 ingresa els diners i
 retorna el canvi
endnote
client -> (Seleccionar Producte)

(Pagament) <.. (Falten diners) : extends
(Seleccionar Producte) <.. (No hi ha stock): extends
(No hi ha stock) .> (Connexió): includes
(Connexió) <- central
}
@enduml
```



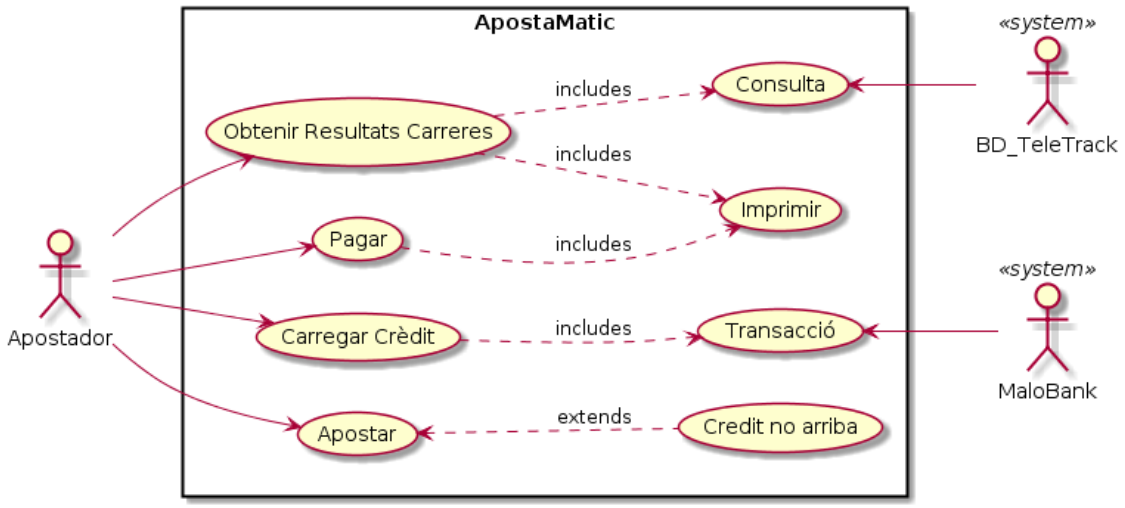
3. Sistema d'apostes automàtiques

Aquesta és l'última aplicació anomenada ApostaMatic pels apostadors empedernits: una màquina que els permet obtenir informació de cavalls / carreres / premis, carregar crèdit de diners des del seu compte corrent (accessible via MaloBanc), realitzar apostes i fins a imprimir un bitllet que és canviable per efectiu en la caixa del local d'apostes (ja que tornar a dipositar-la és incentiu perquè no la gasti).

- No s'accepten apostes que involucrin més diners que el del crèdit actual
- El crèdit que l'apostador desitgi carregar ha de sol·licitar-se al servidor de MaloBank mitjançant una connexió
- Tant l'obtenció d'informació com el pagament d'apostes utilitzen la impressora inclosa en l'ApostaMatic
- La informació de carreres/cavalls/apostes es manté en un computador amb la base de dades de TeleTrak

Solució

```
@startuml
left to right direction
skinparam packageStyle rect
actor Apostador as client
actor MaloBank <<system>> as bank
actor BD_TeleTrack <<system>> as bd
rectangle ApostaMatic{

client -> (Obtenir Resultats Carreres)
client -> (Pagar)
client -> (Carregar Crèdit)
client -> (Apostar)
(Obtenir Resultats Carreres) ..> (Consulta) : includes
(Obtenir Resultats Carreres) ..> (Imprimir) : includes
(Pagar) ..> (Imprimir) : includes
(Carregar Crèdit) ..> (Transacció) : includes
(Apostar) <.. (Credit no arriba) : extends
(Consulta) <- bd
(Transacció) <- bank
}
@enduml
```

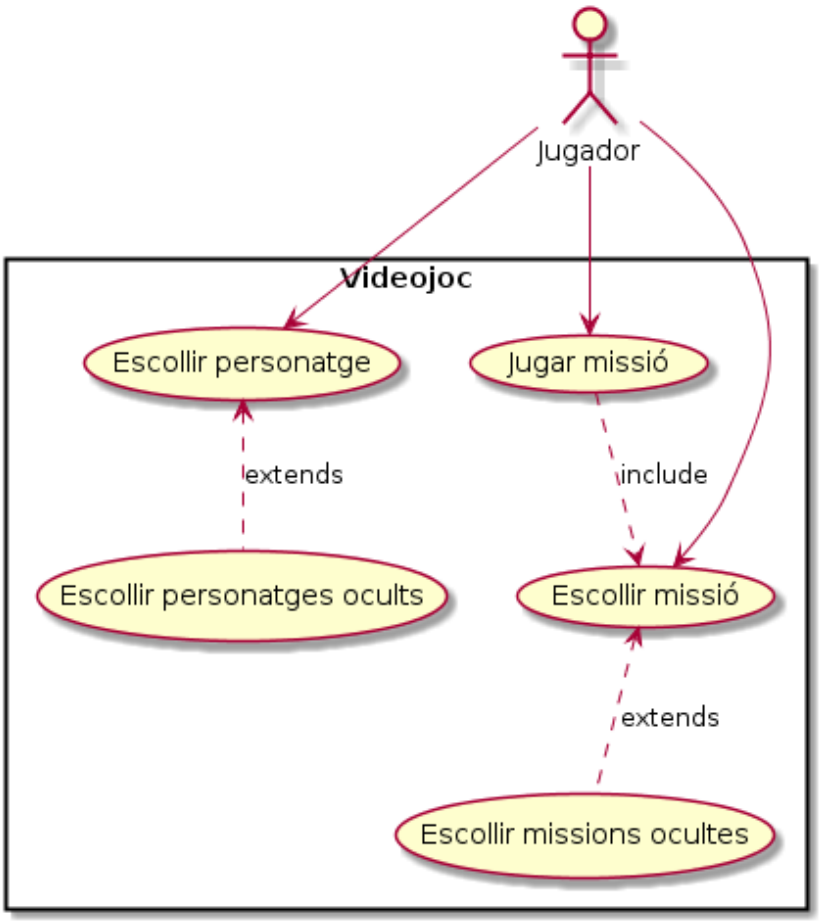


4. Videojoc

Això és una invenció dels anys 70, que per ser reviscuts dins d'un computador a casa han d'utilitzar-se els anomenats "emuladors". Per construir un es demana començar per dissenyar els casos d'ús del sistema (suposant que és una màquina arcade original) on el jugador pot escollir un personatge, una missió, jugar la missió i, si aconsegueix un bon acompliment de la mateixa, ingressar el seu "top-score". També es demana incloure els casos en què el jugador coneix del tema i activa les claus per accedir als personatges i missions ocultes del joc.

Solució

```
@startuml
'left to right direction
'top to bottom direction
skinparam packageStyle rect
actor Jugador as jug


rectangle Videojoc{

"Escollir personatge" as (Escollir)
jug -> (Escollir)

jug -> (Escollir missió)
jug -> (Jugar missió)
(Jugar missió) ..> (Escollir missió) : include

(Escollir missió) <.. (Escollir missions ocultes) : extends
(Escollir) <.. (Escollir personatges ocults) : extends
}
@enduml
```



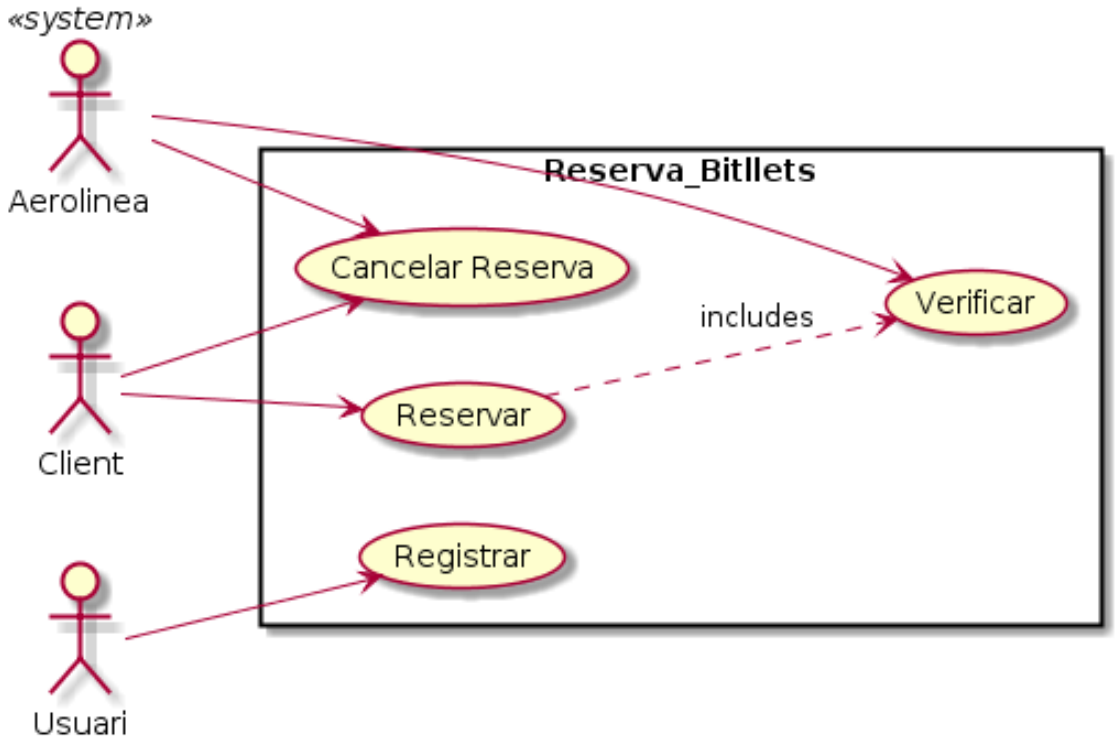
5. Reserva Bitllets Avions

Identifiqui els actors i dibuixi el Diagrama de C.U. que representi un programari que permeti realitzar la reserva de bitllets d'avió en una agència turística, considerant els següents processos del negoci (especificació de C.U.):

- Tot client ha de registrar-se al programari abans de reservar. (usuari)
- El client pot fer una reserva amb un dia i hora, perquè el sistema es comuniqui amb el programari de l'aerolínia desitjada a verificar l'estat del vol. Si no hi ha disponibilitat, el client pot seleccionar un altre vol.
- El client pot cancel·lar una reserva amb 48 hores d'anticipació mínim al sistema. Si és així, la reserva es cancel·la en l'aerolínia que es va fer deixant disponibilitat per a un altre client.
- Un agent de viatges pot realitzar la funció del client en cas que sigui des d'una oficina física, registrant al mateix client i li lliura una clau perquè es comuniqui ell amb el sistema.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect

actor Usuari
actor Client
actor Aerolinia <<system>>
rectangle Reserva_Bitllets {

Usuari -> (Registrar)
Client -> (Reservar)
Client ->  (Cancel·lar Reserva)
Aerolinia -> (Cancel·lar Reserva)
Aerolinia -> (Verificar)

(Reservar) ..> (Verificar): includes
}
@enduml
```



6. Resultats Lliga Futbol Infantil

L'ANFP vol comprar un programari per mantenir en línia els resultats dels partits de futbol en un servidor web existent. Aquest programari ha de ser operat per uns especialistes que es troben en la caseta de transmissió de l'estadi, i seria alimentat amb les següents dades:

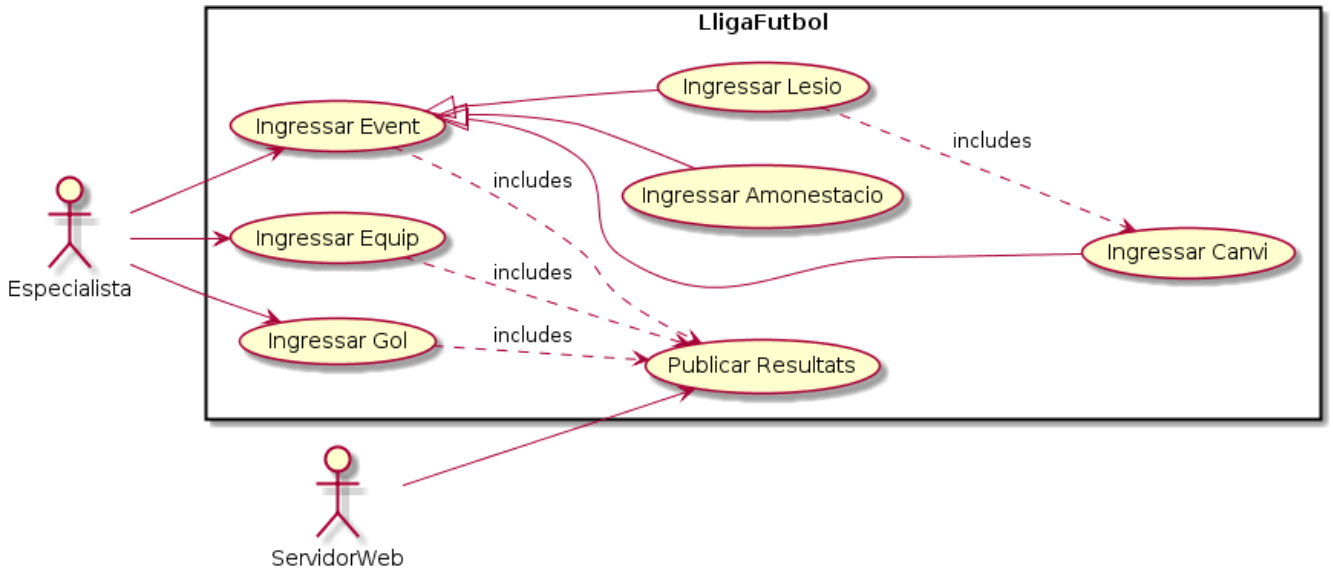
- A l'inici del sofware, ingressa els noms dels equips i la nòmina de jugadors.
- Durant el partit es van emmagatzemant els gols indicant el minut, el jugador i equip que va convertir l'equip.
- També es poden ingressar events com a targetes grogues, targetes vermelles, lesions i canvis en la formació de l'equip.

Consideri que el servidor web està fora del sistema a modelar.

Es demana que identifiqui els casos d'ús i els actors que permetin dibuixar un diagrama de casos d'ús del sistema.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect

actor Especialista
actor ServidorWeb <<system>>
rectangle LligaFutbol {

Especialista -> (Ingressar Event)
Especialista -> (Ingressar Equip)
Especialista -> (Ingressar Gol)
ServidorWeb -> (Publicar Resultats)

(Ingressar Event) <|-- (Ingressar Amonestacio)
(Ingressar Event) <|-- (Ingressar Lesio)
(Ingressar Event)<|-- (Ingressar Canvi)

(Ingressar Event) ..> (Publicar Resultats): includes
(Ingressar Equip) ..> (Publicar Resultats): includes
(Ingressar Gol) ..> (Publicar Resultats): includes
(Ingressar Lesio) ..> (Ingressar Canvi): includes
}
@enduml
```



7. Control Sortides Busos

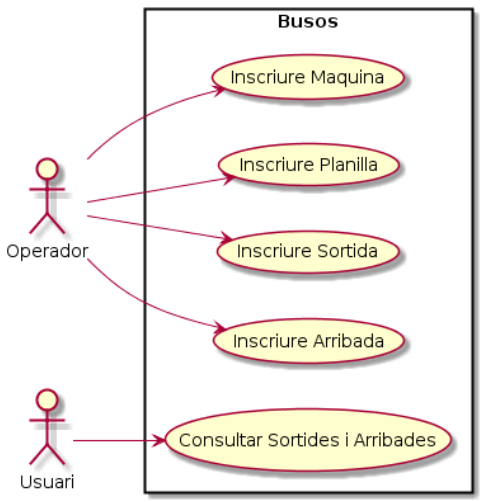
Donat el següent sistema de control de sortides de busos, en la seva especificació de casos d'ús:

- **Procés d'Inscripció de Màquina:** En aquest procés, l'operador ingressa un bus identificat per la seva patent, chofer, sobrecàrrec, capacitat de passatgers i distribució de seients i queda guardat en la base de dades del sistema.
- **Procés d'Ingrés de Planilla:** En aquest procés, l'operador indica les patents dels busos que han de sortir, andana i l'horari de sortida d'aquest. Això es fa 1 vegada al dia i es planifiquen totes les sortides del dia.
- **Procés d'Ingrés de Sortida:** En aquest procés, l'operador ingressa la patent del bus que va sortint i el sistema guarda l'hora d'arribada. A més, el sistema actualitza que l'andana en la que estava ara està buit.
- **Procés d'Ingrés d'Arribada:** En aquest procés, l'operador ingressa la patent del bus que ve arribant i el sistema guarda l'hora d'arribada. A més, el sistema retorna l'andana en la que ha d'estacionar-se el bus (andana buida).
- **Procés de Consulta de Sortida i Arribades:** En aquest procés, l'usuari veu una planilla obtinguda des de la base de dades amb totes les properes sortides (propera hora) i les arribades que han ocorregut en aquesta última mitja hora.

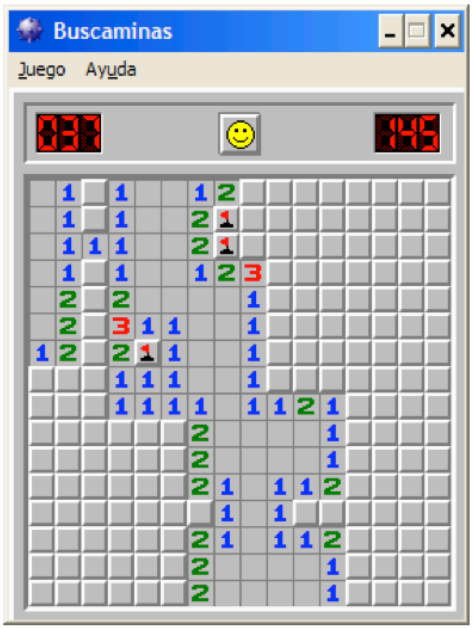
Realitzi el diagrama de casos d'ús.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect

actor Operador
actor Usuari
rectangle Busos {

Operador -> (Inscriure Maquina)
Operador -> (Inscriure Planilla)
Operador ->  (Inscriure Sortida)
Operador ->  (Inscriure Arribada)
Usuari -> (Consultar Sortides i Arribades)
}

@enduml
```



8. Buscamines

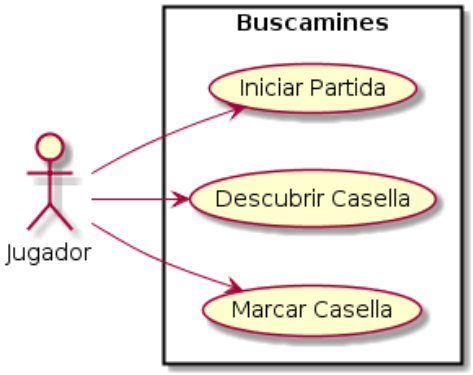


Buscamines és un joc que consisteix a buidar un camp de joc ple de mines sense detonar-ne cap. El joc està composat per una pantalla omplerta de caselles que un únic jugador haurà de buidar de mines. El jugador després d'iniciar la partida , haurà d'anar clicant sobre les caselles de la pantalla per marcar-les en cas de dubte de si hi ha una mina, o bé per descobrir el número que amaguen. Els números que es desvetllen indiquen la quantitat de mines amagades a les caselles veïnes.

Realitzi el diagrama de casos d'ús del problema plantejat.

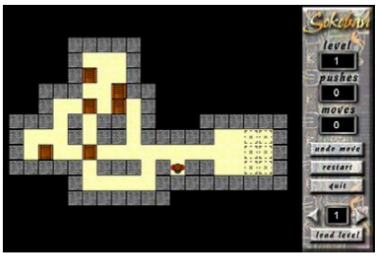
Solució

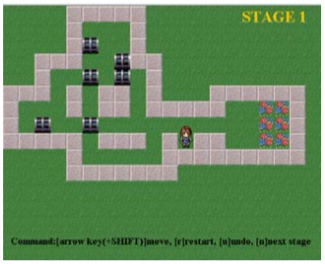
```
@startuml
left to right direction
skinparam packageStyle rect

actor Jugador
rectangle Buscamines {

Jugador -> (Iniciar Partida)
Jugador -> (Descubrir Casella)
Jugador ->  (Marcar Casella)
}

@enduml
```

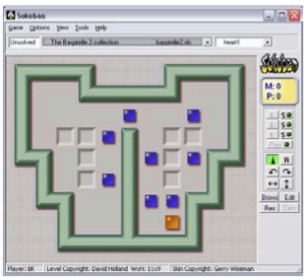


9. Sokoban

Sokoban és un joc de diversos nivells. Cada nivell està compost per un jugador, caixes, cartel·les i murs. L'objectiu del jugador és empènyer totes les caixes sobre les cartel·les. Quan això succeeix el jugador passa al següent nivell. Per moure una caixa, el jugador ha de col·locar-se al costat i empènyer-la. Si la casella cap a la qual està empenyent la caixa està lliure, la caixa es mourà. Si el jugador es queda bloquejat, és a dir, no pot acabar el nivell, pot reiniciar el nivell perdent una vida. Quan el jugador perd totes les seves vides la partida acaba.

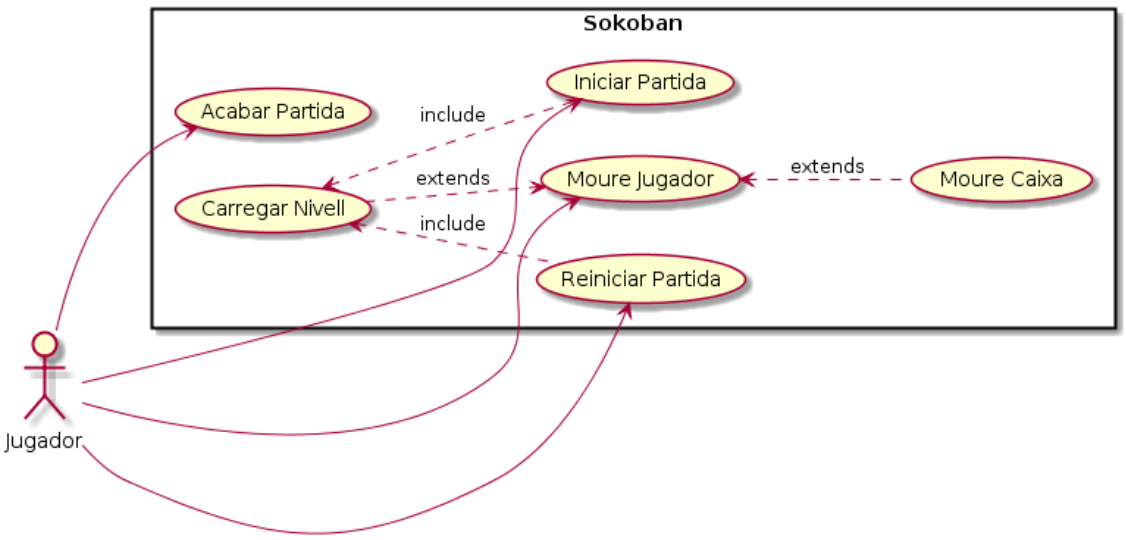






Solució

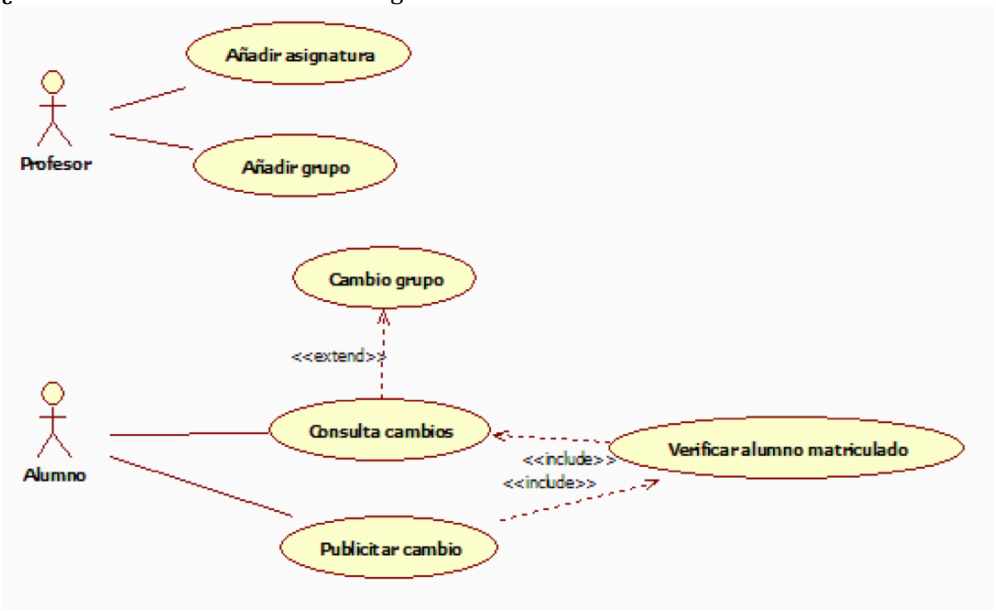
```
@startuml
left to right direction
skinparam packageStyle rect

Actor Jugador
rectangle Sokoban {

Jugador -> (Iniciar Partida)
Jugador -> (Moure Jugador)
Jugador ->  (Reiniciar Partida)
Jugador ->  (Acabar Partida)
(Carregar Nivell) <.. (Iniciar Partida): include
(Carregar Nivell) <.. (Reiniciar Partida): include
(Moure Jugador) <.. (Moure Caixa): extends
(Carregar Nivell) ..> (Moure Jugador): extends

}
@enduml
```



10. Canvis Grups de pràctiques

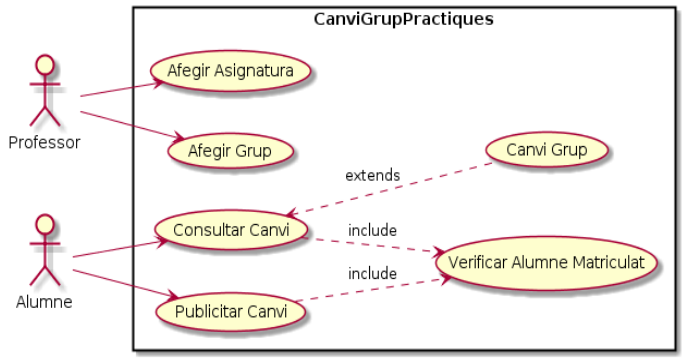
Un sistema automàtic de canvi de grups per a assignatures funciona de la següent manera:
El professor dóna d'alta una assignatura i proporciona al sistema un llistat amb els alumnes matriculats en aquesta assignatura. Un alumne que vulgui canviar de grup en una assignatura pot consultar les peticions de canvi. Si troba alguna que li interessi, l'alumne sol·licita el canvi i el sistema ho emmagatzema. Si no, l'alumne pot deixar el canvi que desitja per si a un altre alumne li interessés. Els alumnes només poden consultar i publicitar canvis de les assignatures en les quals estan matriculats.

¿On es troben les errades en el diagrama?



Solució

```
startuml
left to right direction
skinparam packageStyle rect

Actor Professor
Actor Alumne
rectangle CanviGrupPractiques {

Professor -> (Afegir Asignatura)
Professor -> (Afegir Grup)
Alumne ->  (Consultar Canvi)
Alumne ->  (Publicitar Canvi)
(Consultar Canvi) <.. (Canvi Grup): extends
(Publicitar Canvi) ..> (Verificar Alumne Matriculat): include
(Consultar Canvi) ..> (Verificar Alumne Matriculat): include
}
@enduml
```

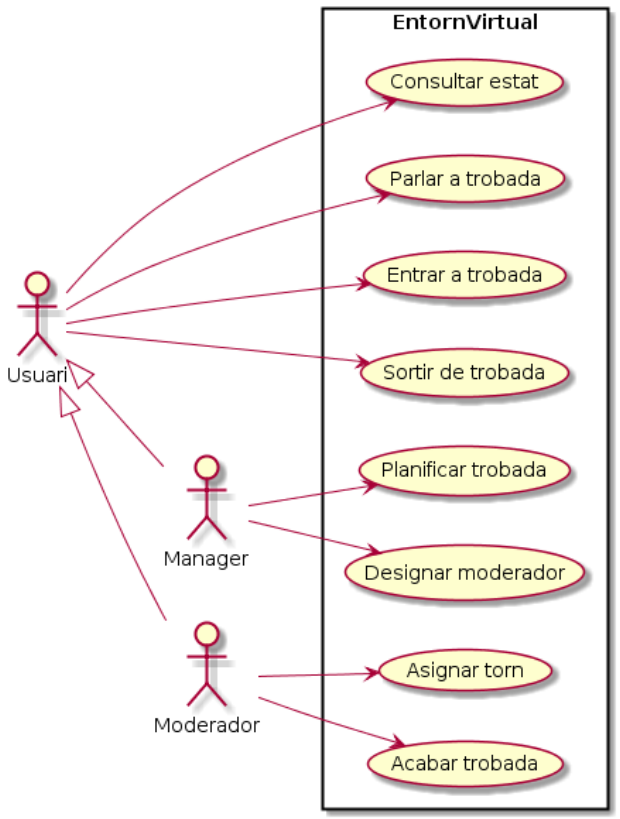


11. Entorn Virtual

Es desitja desenvolupar un sistema de trobades virtuals (semblant a un xat).
Quan es connecta al servidor, un usuari pot entrar o sortir d'una trobada. Cada trobada té un manager. El manager és l'usuari que ha planificat la trobada (el nom de la trobada, l'agenda de la trobada i el moderador de la trobada). Cada trobada pot tenir també un moderador designat pel manager. La misió del moderador és assignar els torns de paraula perquè els usuaris parlin. El moderador també podrà donar per conclòs la trobada a qualsevol moment. A qualsevol moment un usuari pot consultar l'estat del sistema, per exemple les trobades planejades i la seva informació.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect

Actor Usuari
Actor Manager
Actor Moderador
Usuari <|-- Manager
Usuari <|-- Moderador
rectangle EntornVirtual {

Usuari -> (Consultar estat)
Usuari -> (Parlar a trobada)
Usuari -> (Entrar a trobada)
Usuari -> (Sortir de trobada)
Manager -> (Planificar trobada)
Manager -> (Designar moderador)
Moderador -> (Asignar torn)
Moderador -> (Acabar trobada)

}
@enduml
```

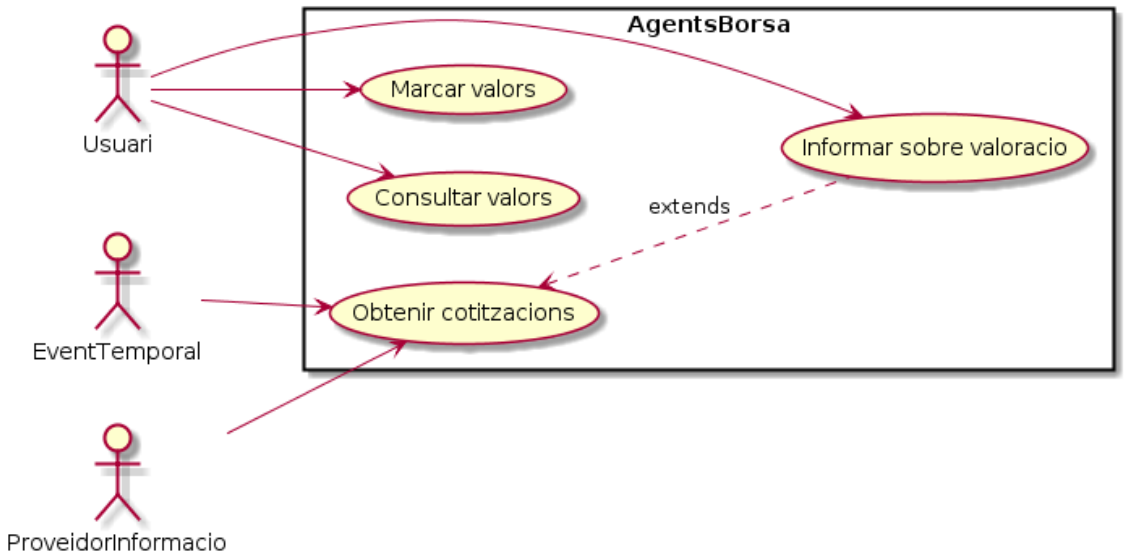


12. Agents de Borsa

Un sistema personal de borsa es connecta periòdicament a servidors que ofereixen informació de les cotitzacions. El sistema personal permet marcar una sèrie de valors per realitzar un seguiment i consultar les dades d'aquests valors. Si a l'hora d'actualitzar les cotitzacions un dels valors marcats presenta una gran pujada o baixada, informarà a l'usuari d'això.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect

actor Usuari
actor EventTemporal
actor ProveidorInformacio
rectangle AgentsBorsa {

Usuari -> (Consultar valors)
Usuari -> (Marcar valors)
Usuari -> (Informar sobre valoracio)
EventTemporal -> (Obtenir cotitzacions)
ProveidorInformacio -> (Obtenir cotitzacions)

(Obtenir cotitzacions) <.. (Informar sobre valoracio): extends

}
@enduml
```



13. Joc de mòbil

Un joc de telèfon mòbil on participen dos jugadors cadascun amb la seva pròpia terminal. Quan dos jugadors desitgen jugar, un d'ells crea una nova partida i l'altre es connecta. L'objectiu del joc és manejar una nau i disparar al contrari. Si un dels dos jugadors encerta, la partida acaba. Si un dels dos jugadors deixa la partida (o es perd la conexió) la partida acaba.

Solució

```
@startuml
left to right direction
skinparam packageStyle rect

Actor JugadorA
Actor JugadorB
rectangle AgentsBorsa {

JugadorA -> (Iniciar Partida)
JugadorA -> (Moure Nau)
JugadorA -> (Disparar)
JugadorA -> (Finalitzar Partida)
JugadorB -> (Conectar Partida)
JugadorB -> (Moure Nau)
JugadorB -> (Disparar)
JugadorB -> (Finalitzar Partida)
(Disparar) <.. (Finalitzar Partida): extends

}
@enduml
```