Introducció a l’Enginyeria del Programari, Curs 2023/24 Q1

**Sistema de gestió de videojocs**

**Disseny**

**Pau Alcázar Perdomo i Ausiàs Plaza Juan**

**Instruccions per fer el disseny**

Per poder fer el disseny, es farà servir l’especificació que s’inclou en aquest document, a

l'excepció de l’associació ternària que s’inclou al model de comportament. Això vol dir que el disseny no ha d’incloure la informació de les etiquetes ni cap funcionalitat relacionada amb la seva gestió. Igualment, ens restringirem a cinc casos d’ús:

● Comprar videojoc

● Comprar paquet

● Consultar compres

● Modificar usuari

● Esborrar usuari

Per a aquests cinc casos d’ús, el document de disseny ha d'incloure:

● Disseny de la interfície (mapa navegacional)

● Especificació dels casos d’ús individuals concrets (incloent informació de la interfície)

● Diagrames de seqüència complets (és a dir, incloent les tres capes de totes les

operacions que sorgeixen en aquests casos d’ús

● Contracte de les operacions de la capa de domini (exceptuant getters i setters).

A més, cal incloure el diagrama de classes de disseny amb tots els atributs i operacions:

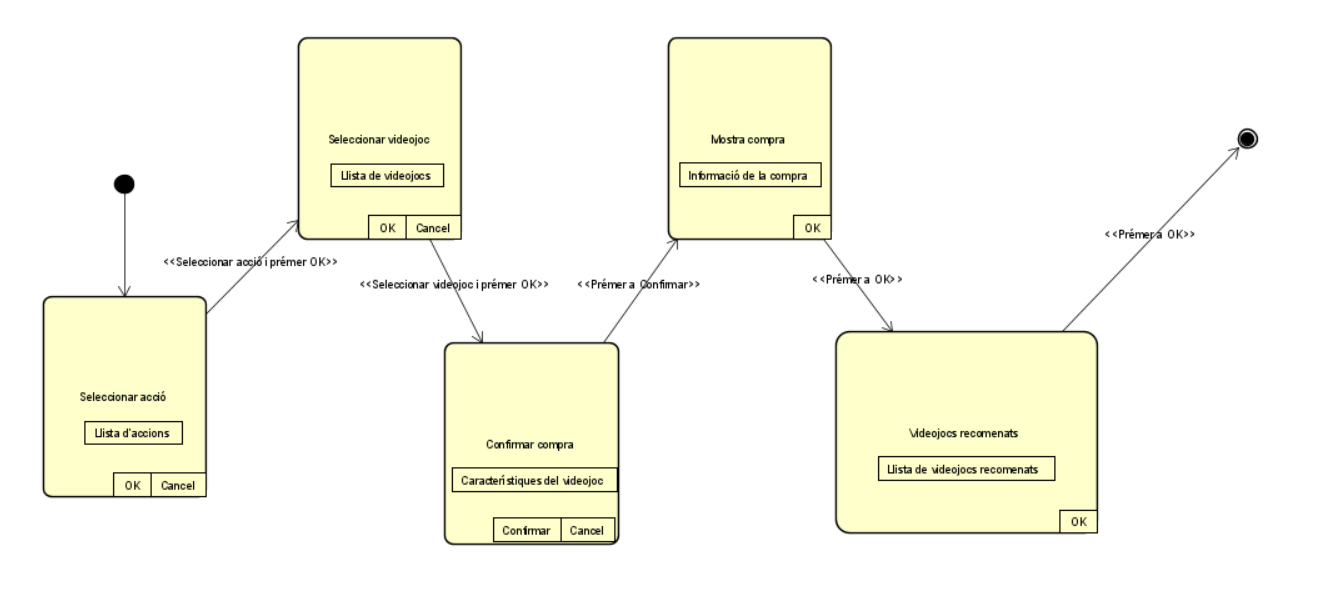
● no incloure visibilitat dels atributs i operacions (es considera la visibilitat per defecte)

● no incloure getters ni setters

● no incloure les operacions de la capa de dades

Recordeu que els requisits no funcionals que estem prioritzant en aquesta assignatura són la canviabilitat, eficiència, portabilitat i reusabilitat.

1. **Comprar videojoc**
   1. Disseny de la interfície (mapa navegacional)



* 1. Especificació dels casos d’ús individuals concrets (incloent informació de la interfície)

**cas d’ús** Comprar videojoc

**activació** Un usuari loggejat vol comprar un videojoc.

**escenari principal**

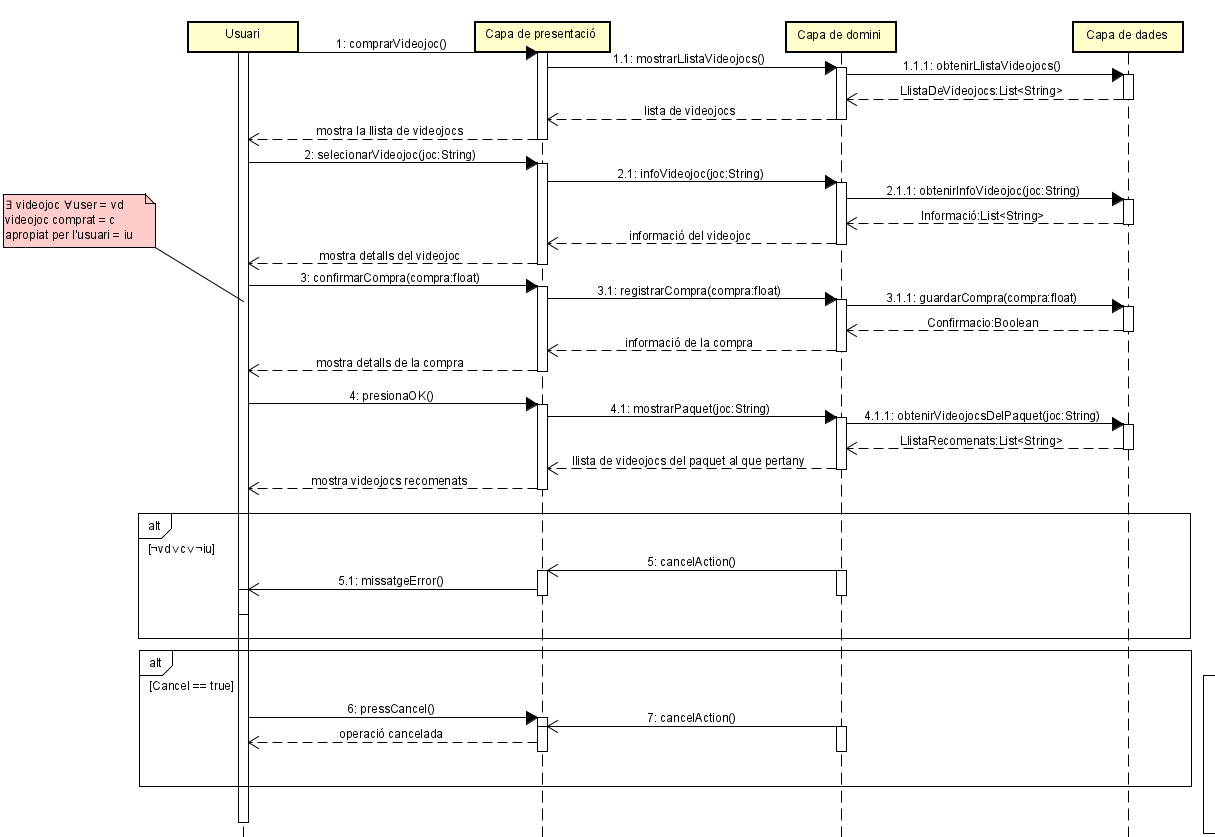
1. L’usuari ha de seleccionar l’acció que vol realitzar (Comprar videojocs) i posteriorment ha de prémer el botó OK.
2. L’usuari selecciona el videojoc dins de la llista de jocs del sistema, i després de seleccionar-lo prem el botó OK, confirmant la selecció.
3. El Sistema li mostra tota la informació del videojoc (nom, descripció, qualificació per edats, gènere i data de llançament) juntament amb el preu. El sistema pregunta a l’usuari si vol continuar amb la compra. I l’usuari per confirmar aquesta compra ha de prémer la tecla Confirmar.
4. Quan l’usuari confirma que vol continuar amb la compra, el Sistema enregistra la compra amb data corresponent a la data i hora en el que es fa la compra.
5. Una vegada enregistrada la compra, el sistema mostra els detalls de la compra que acaba de enregistrar.
6. I quan l’usuari prem OK, el Sistema mostra el nom, descripció i preu dels jocs que estan a paquets on es troba el videojoc comprat que estiguin disponibles per la compra (que la seva data de llançament sigui menor o igual a la data d’avui). Al prémer OK, haurà acabat l’acció.

**escenaris alternatius**

* El videojoc no existeix. El sistema mostra un missatge d’error i finalitza el cas d’ús.
* L’usuari ja ha comprat el videojoc amb anterioritat. El Sistema mostra un missatge d’error i finalitza el cas d'ús.
* El videojoc no és apropiat per l’edat de l’usuari. El sistema mostra un missatge d’error i finalitza el cas d’ús.
* L’usuari no confirma la compra. El cas d’ús finalitza.
* L’usuari prem el botó Cancel en qualsevol pas del procés. El cas d’ús finalitza.

**requisits no funcional**

* Els missatges d’error seran entenedors (donant una explicació entenedora de perquè ha passat l’error).
  1. Diagrames de seqüència complets (és a dir, incloent les tres capes de totes les operacions que sorgeixen en aquests casos d’ús.



* 1. Contracte de les operacions de la capa de domini (exceptuant getters i setters).

**Operació: mostrarLlistaVideojocs()**

* Descripció: Obté la llista de videojocs disponibles en el catàleg.
* Precondició: Ha d’existir al menys un videojoc dins del catàleg de videojocs del sistema.
* Postcondició: Es crea una instància de tipus LlistaDeVideojocs amb tots els videojocs que formen part del catàleg.
* Retorna: L’objecte LlistaDeVideojocs amb tots els jocs disponibles.

**Operació: infoVideojoc(joc: String)**

* Descripció: Obté la informació detallada del videojoc especificat.
* Precondició: El videojoc identificat per 'joc' ha d'existir dins del catàleg de videojocs del sistema.
* Postcondició: Es crea una instància de tipus DetallsVideojoc amb la informació completa del videojoc, incloent-hi el nom, descripció, classificació per edat, gènere, data de llançament i preu.
* Retorna: L'objecte DetallsVideojoc amb la informació requerida.

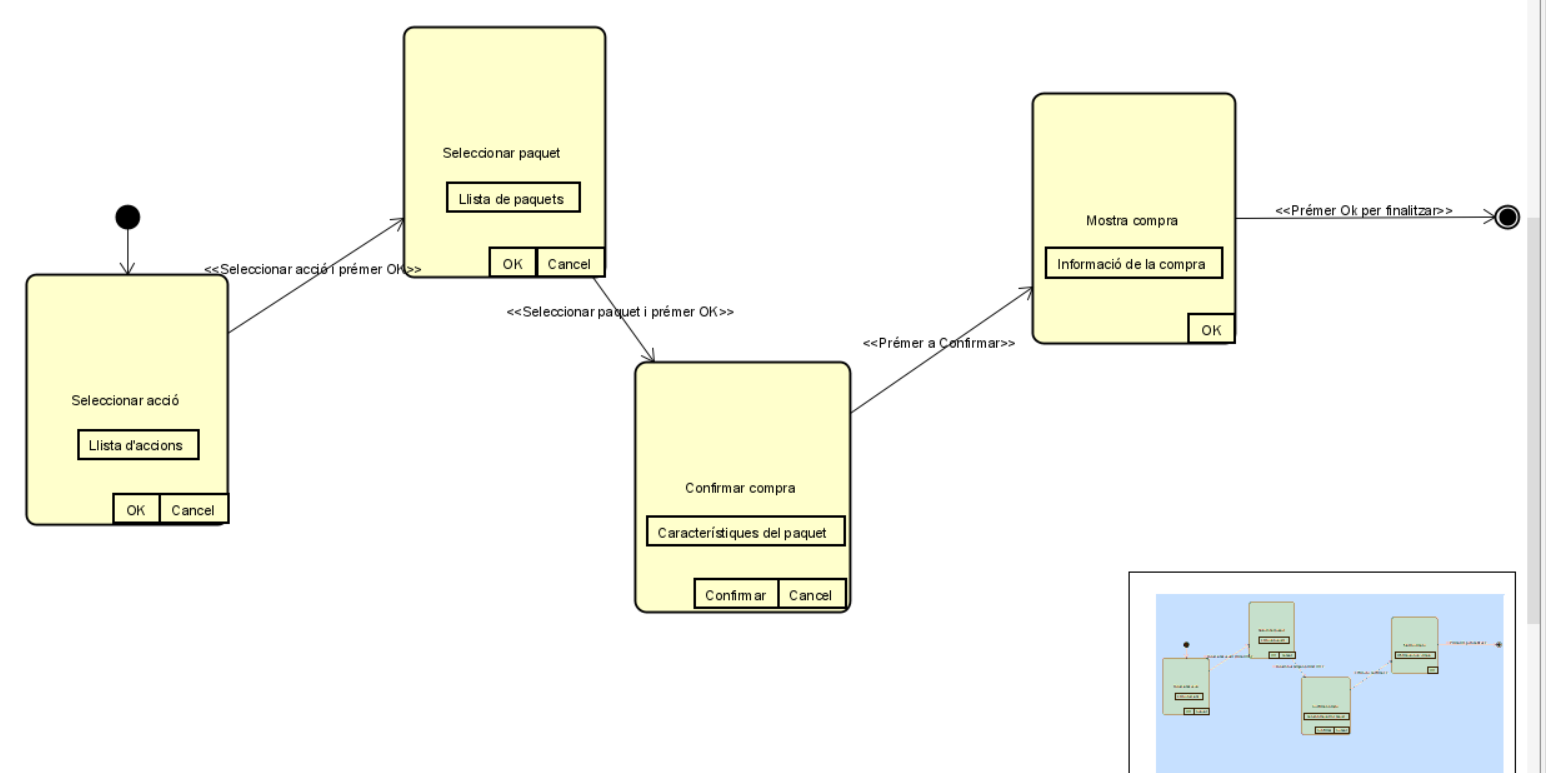
**Operació: registrarCompra(compra: float)**

* Descripció: Registra la compra del videojoc per part de l'usuari autenticat.
* Precondició: L'usuari ha de tenir una sessió activa i ha d'haver seleccionat un videojoc vàlid per a la compra.
* Postcondició: Es crea un registre de compra que inclou la data i hora actuals, així com el preu pagat. Aquesta informació s'associa amb l'usuari que fa la compra i el videojoc comprat.
* Retorna: Un booleà que indica si la compra s'ha registrat amb èxit.

**Operació: mostrarPaquetjoc(joc: String)**

* Descripció: Mostra els paquets de videojocs que inclouen el videojoc especificat.
* Precondició: La compra del videojoc identificat per 'joc' ha d'haver-se completat amb èxit.
* Postcondició: Es genera una llista dels paquets disponibles que contenen el videojoc comprat. Cada paquet inclou la llista de noms dels videojocs que conté i el seu preu.
* Retorna: Una llista de paquets que són recomanats basant-se en el videojoc comprat.

1. **Comprar paquet**
   1. Disseny de la interfície (mapa navegacional)



* 1. Especificació dels casos d’ús individuals concrets (incloent informació de la interfície)

**cas d’ús** Comprar paquet videojocs

**activació** Un usuari loggejat vol comprar un paquet de videojocs.

**escenari principal**

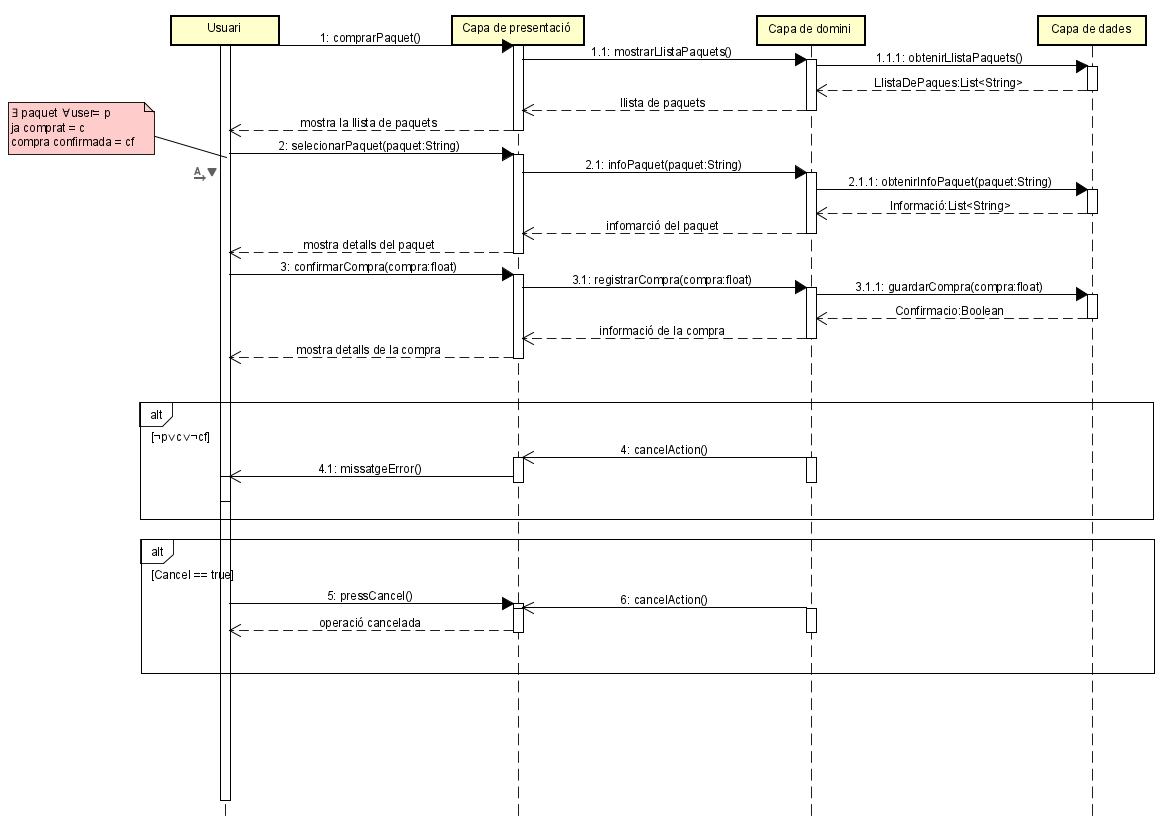
1. L’usuari ha de seleccionar l’acció que vol realitzar (Comprar paquet de videojocs) i posteriorment ha de prémer el botó OK.
2. L’usuari selecciona el nom del paquet que vol comprar, i prem el botó OK.
3. El Sistema, del paquet li mostra la descripció, el preu i l’estalvi amb relació a comprar tots el jocs per separat i la llista dels noms i preus de tots el videojocs que conté el paquet. El sistema pregunta a l’usuari si vol continuar amb la compra, i l’usuari ha de confirmar la compra prement a al botó Confirmar.
4. Quan l’usuari confirma la compra, el Sistema registra la compra amb data corresponent a la data i hora en el que es fa la compra, i mostra per pantalla la informació de la compra, i quan l’usuari prem OK, finalitza l’acció.

**escenaris alternatius**

* El paquet no existeix o l’usuari ja l’ha comprat. El sistema mostra un missatge d’error i finalitza el cas d’ús.
* L’usuari no confirma la compra. El cas d’ús finalitza.
* L’usuari prem el botó Cancel en qualsevol pas del procés. El cas d’ús finalitza.

**requisits no funcional**

* Els missatges d’error seran entenedors (donant una explicació entenedora de perquè ha passat l’error).
  1. Diagrames de seqüència complets (és a dir, incloent les tres capes de totes les operacions que sorgeixen en aquests casos d’ús.



* 1. Contracte de les operacions de la capa de domini (exceptuant getters i setters).

**Operació: mostrarLlistaPaquets()**

* Descripció: Mostra una llista dels paquets de videojocs disponibles per a la compra.
* Precondició: L'usuari ha d'estar autenticat i ha sol·licitat veure els paquets disponibles.
* Postcondició: Es presenta a l'usuari una llista de paquets amb la seva descripció, preu i l'estalvi respecte a comprar els videojocs per separat.

Retorna: No retorna res ja que és una acció de presentació.

**Operació: seleccionarPaquet(paquet: String)**

* Descripció: Selecciona un paquet específic de la llista per a més detalls o per a la compra.
* Precondició: El paquet ha d'existir dins la llista mostrada a l'usuari.
* Postcondició: El sistema estableix el paquet seleccionat com a candidat per a la compra.
* Retorna: No retorna res ja que és una acció de presentació.

**Operació: confirmarCompra(compra: float)**

* Descripció: Registra la compra del paquet seleccionat per l'usuari.
* Precondició: L'usuari ha d'haver seleccionat un paquet vàlid i confirmat que vol continuar amb la compra.
* Postcondició: La compra del paquet es registra amb la data i hora actuals, i el sistema actualitza l'estat del paquet a "comprat" per aquest usuari.
* Retorna: Un booleà que indica si la compra ha estat registrada correctament.

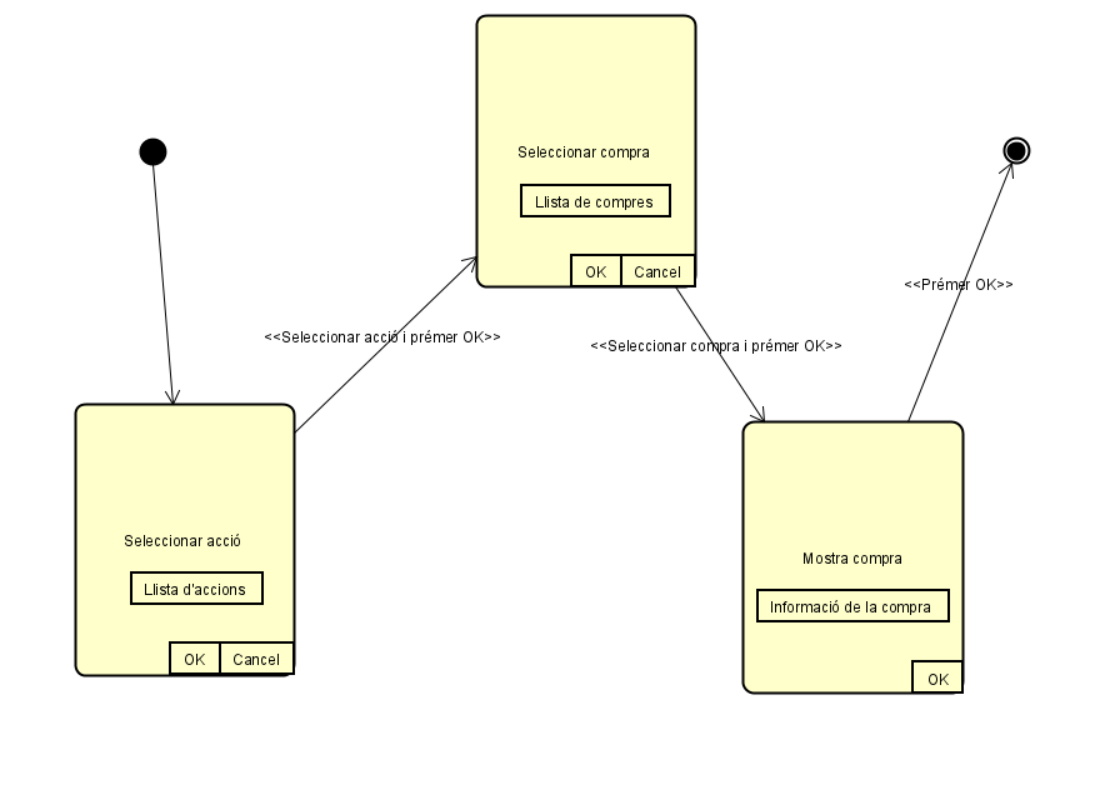
**Operació: mostrarError()**

* Descripció: Mostra un missatge d'error si el paquet no existeix, l'usuari ja l'ha comprat, o si l'usuari cancel·la la compra.
* Precondició: Es dóna qualsevol de les condicions especificades per a un escenari alternatiu.
* Postcondició: S'informa a l'usuari de l'error i el procés de compra s'interromp.
* Retorna: No retorna res ja que és una acció de presentació.

**Operació: cancelAction()**

* Descripció: Cancel·la el procés de compra del paquet a petició de l'usuari.
* Precondició: L'usuari prem el botó de cancel·lació durant qualsevol punt del procés de compra.
* Postcondició: El procés de compra s'interromp i cap canvi no es realitza en l'estat de la compra.
* Retorna: No retorna res ja que és una acció de control de flux.

1. **Consultar compres**
   1. Disseny de la interfície (mapa navegacional)



* 1. Especificació dels casos d’ús individuals concrets (incloent informació de la interfície)

**cas d’ús** Consultar compres

**activació** Un usuari loggejat vol consultar la informació de les compres que ha realitzat. **escenari principal**

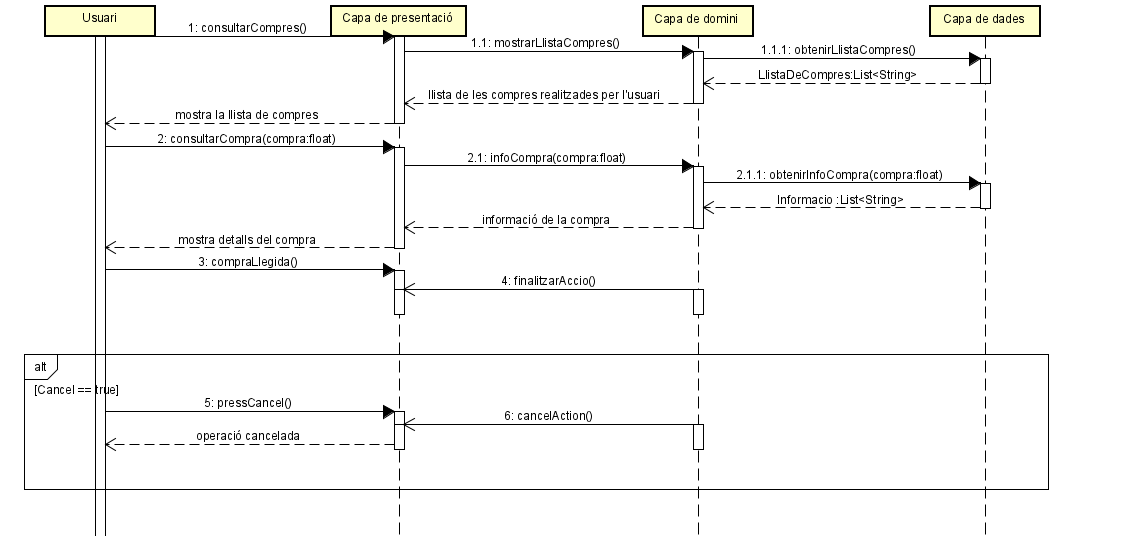
1. L’usuari ha de seleccionar l’acció que vol realitzar (Consultar compres) i posteriorment ha de prémer el botó OK.
2. El sistema mostra la llista amb la data de compra, nom, descripció i preu de l'element per totes les compres realitzades per l’usuari. Quan la compra és d’un paquet, també es mostrarà la llista de noms dels videojocs continguts al paquet. Aquesta llista estarà ordenada descendentment per data de compra. L’usuari ha de seleccionar la compra que vol que es mostri detalladament, després ha de prémer el botó OK.
3. El sistema mostra totes les característiques de la compra de manera detallada, i quan l’usuari ha acabat de llegir-les, prem el botó de Confirmar.

**escenaris alternatius**

* L’usuari prem el botó Cancel en qualsevol pas del procés. El cas d’ús finalitza.

**requisits no funcional**

* 1. Diagrames de seqüència complets (és a dir, incloent les tres capes de totes les operacions que sorgeixen en aquests casos d’ús



* 1. Contracte de les operacions de la capa de domini (exceptuant getters i setters).

**Operació: mostrarLlistaCompres()**

* Descripció: Mostra la llista de totes les compres realitzades per l'usuari.
* Precondició: L'usuari ha d'estar autenticat i haver sol·licitat veure les seves compres.
* Postcondició: Es presenta a l'usuari una llista de compres ordenada descendentment per data de compra. Aquesta llista inclou la data de compra, nom, descripció i preu de cada element comprat, així com la llista de noms dels videojocs continguts en qualsevol paquet comprat.
* Retorna: No aplica, ja que la informació es mostra directament a l'usuari.

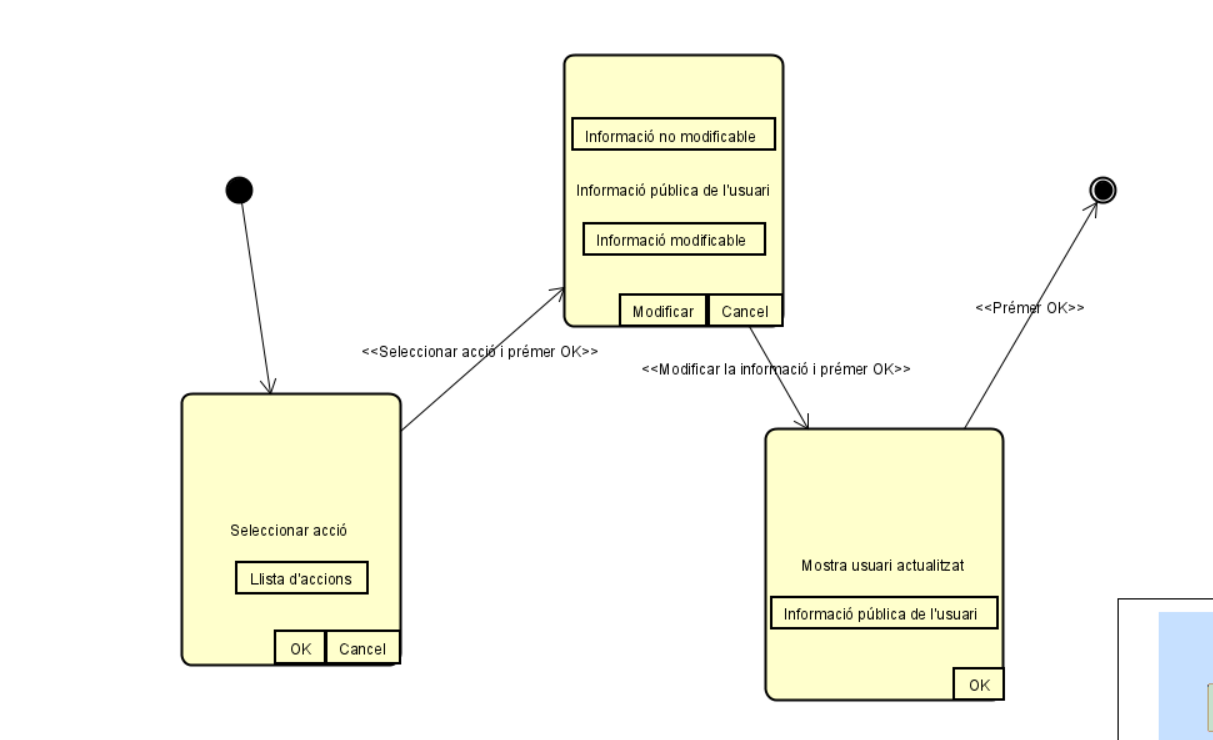
**Operació: infoCompra(compra: float)**

* Descripció: Proporciona detalls d'una compra específica seleccionada per l'usuari.
* Precondició: L'usuari ha de haver seleccionat una compra de la llista mostrada.
* Postcondició: Es mostra a l'usuari tota la informació detallada de la compra seleccionada.
* Retorna: No aplica, ja que la informació es mostra directament a l'usuari.

**Operació: finalitzarAccio()**

* Descripció: Finalitza la consulta de compres i retorna a l'usuari al menú principal o a la pantalla anterior.
* Precondició: L'operació infoCompra s'ha executat i l'usuari ha revisat els detalls de la compra.
* Postcondició: L'acció de consulta de compres es dóna per acabada i l'usuari pot continuar amb altres accions dins del sistema.
* Retorna: No aplica, ja que és una acció de finalització de procés.

1. **Modificar usuari**
   1. Disseny de la interfície (mapa navegacional)



* 1. Especificació dels casos d’ús individuals concrets (incloent informació de la interfície)

**cas d’ús** Modificar usuari

**activació** Un usuari loggejat vol modificar la seva informació.

**escenari principal**

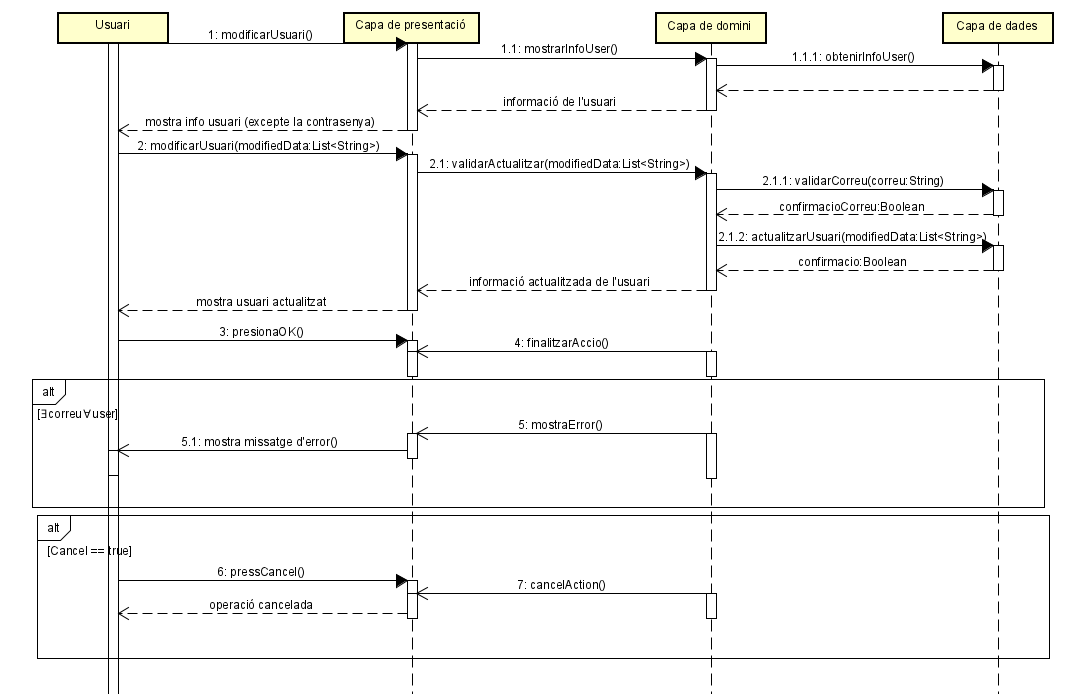
1. L’usuari ha de seleccionar l’acció que vol realitzar (Modificar usuari) i posteriorment ha de prémer el botó OK.
2. El Sistema mostra tota la informació de l’usuari loggejat (menys la contrasenya) i l’usuari les dades que vol canviar (nom, adreça de correu electrònic, data de naixement); el sobrenom i la contrasenya no es poden modificar.
3. L’usuari modifica la informació que cregui necessària i li dona al botó Modificar.
4. El Sistema modifica la informació de l’usuari, i mostra per pantalla la informació modificada, quan l’usuari prem OK, finalitza l’acció.

**escenaris alternatius**

* El correu electrònic ja existeix. El sistema mostra un missatge d’error i finalitza el cas d’ús.
* L’usuari prem el botó Cancel en qualsevol pas del procés. El cas d’ús finalitza.

**requisits no funcional**

* El Sistema no mostrarà per pantalla la contrasenya (per privacitat).
* El Sistema guarda la informació de la contrasenya de manera encriptada (per seguretat i privacitat).
* Els missatges d’error seran entenedors (donant una explicació entenedora de perquè ha passat l’error).
  1. Diagrames de seqüència complets (és a dir, incloent les tres capes de totes les operacions que sorgeixen en aquests casos d’ús



* 1. Contracte de les operacions de la capa de domini (exceptuant getters i setters).

**Operació: mostrarInfoUser()**

* Descripció: Obté la informació actual de l'usuari.
* Precondició: L'usuari ha d'estar autenticat i haver sol·licitat modificar la seva informació.
* Postcondició: Es mostra tota la informació de l'usuari, llevat de la contrasenya.
* Retorna: No aplica, ja que la informació es mostra directament a l'usuari.

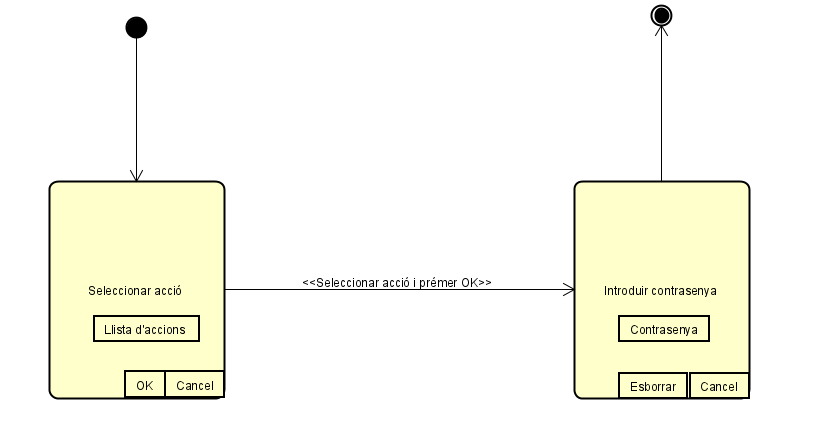
**Operació: validarActualitzar(modifiedData: List<String>)**

* Descripció: Valida i actualitza la informació de l'usuari amb les dades proporcionades.
* Precondició: L'usuari ha introduït les dades que vol canviar.
* Postcondició: Si la validació és correcta, la informació de l'usuari s'actualitza. Si no és correcta o el correu electrònic ja existeix, es mostra un missatge d'error.
* Retorna: Un booleà que indica si l'actualització ha estat exitosa.

**Operació: finalitzarAccio()**

* Descripció: Finalitza el procés de modificació de la informació de l'usuari.
* Precondició: Les dades de l'usuari s'han actualitzat correctament o s'ha mostrat un missatge d'error.
* Postcondició: L'acció de modificació de l'usuari s'ha completat i l'usuari pot continuar amb altres accions dins del sistema.
* Retorna: No aplica.

1. **Esborrar usuari**
   1. Disseny de la interfície (mapa navegacional)



* 1. Especificació dels casos d’ús individuals concrets (incloent informació de la interfície)

**cas d’ús** Esborrar usuari

**activació** Un usuari loggejat vol donar-se de baixa el sistema.

**escenari principal**

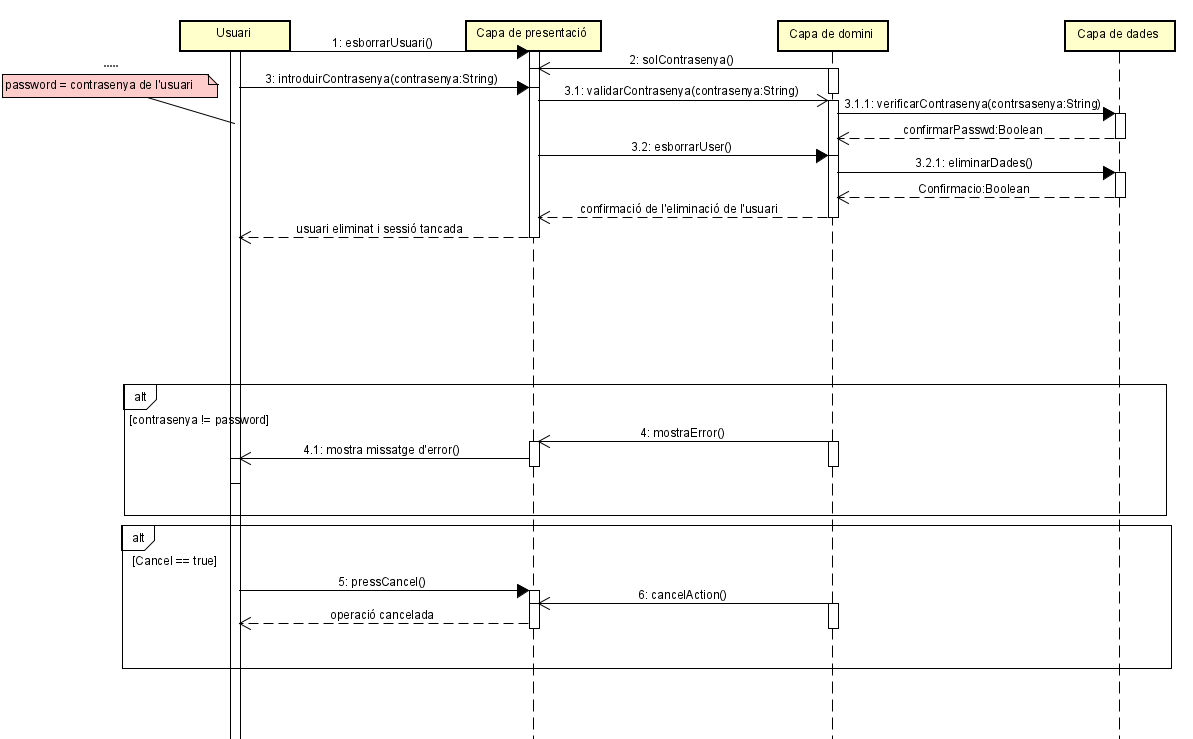
1. L’usuari ha de seleccionar l’acció que vol realitzar (Esborrar usuari) i posteriorment ha de prémer el botó OK.
2. L’usuari entra la contrasenya i prem al botó Confirmar, i el Sistema esborra tota la informació de l’usuari (usuari, compres i etiquetes dels jocs).
3. Una vegada esborrat l’usuari es tanca la sessió per aquest usuari.

**escenaris alternatius**

* La contrasenya no és correcta.
* El sistema mostra un missatge d’error i finalitza el cas d’ús.
* L’usuari prem el botó Cancel en qualsevol pas del procés. El cas d’ús finalitza.

**requisits no funcional**

* El Sistema no mostrarà per pantalla la contrasenya (per privacitat).
* Els missatges d’error seran entenedors (donant una explicació entenedora de perquè ha passat l’error).
  1. Diagrames de seqüència complets (és a dir, incloent les tres capes de totes les operacions que sorgeixen en aquests casos d’ús



* 1. Contracte de les operacions de la capa de domini (exceptuant getters i setters).

**Operació: solContrasenya()**

* Descripció: Demana a l'usuari que introdueixi la seva contrasenya per verificar la seva identitat abans d'esborrar el seu compte.
* Precondició: L'usuari ha seleccionat l'opció d'esborrar el seu usuari.
* Postcondició: El sistema ha rebut la contrasenya de l'usuari per a la seva verificació.
* Retorna: No aplica, ja que és un pas intermedi.

**Operació: validarContrasenya(contrasenya: String)**

* Descripció: Valida la contrasenya introduïda per l'usuari.
* Precondició: L'usuari ha introduït la seva contrasenya.
* Postcondició: Si la contrasenya és correcta, el sistema procedirà amb l'esborrament de l'usuari. Si no, es retornarà un missatge d'error.
* Retorna: Un booleà que indica si la contrasenya és correcta.

**Operació: esborrarUser()**

* Descripció: Esborra totes les dades associades amb l'usuari del sistema.
* Precondició: La contrasenya de l'usuari ha estat validada correctament.
* Postcondició: Totes les dades de l'usuari han estat eliminades del sistema, i la sessió de l'usuari ha estat tancada.
* Retorna: Una confirmació que l'usuari ha estat esborrat.

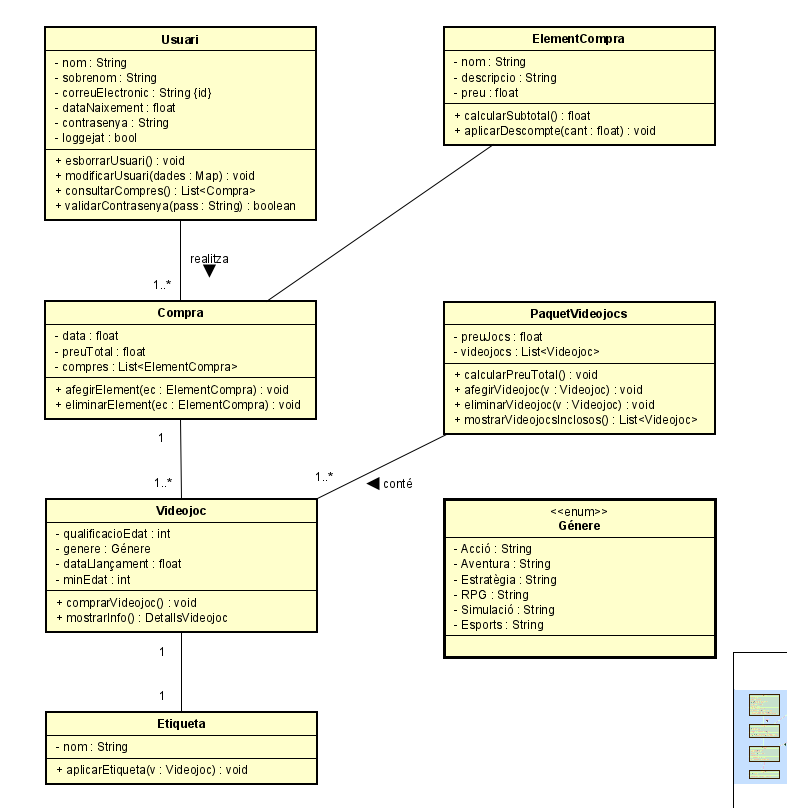
**Operació: mostrarError()**

* Descripció: Mostra un missatge d'error si la contrasenya no és correcta o si es produeix un altre error durant el procés d'esborrament.
* Precondició: Hi ha hagut un error en el procés de validació de la contrasenya o en l'esborrament de les dades de l'usuari.
* Postcondició: S'ha mostrat a l'usuari un missatge d'error clar i entenedor.
* Retorna: No aplica.

**Operació: cancelAction()**

* Descripció: Cancel·la el procés d'esborrament de l'usuari si l'usuari prem el botó de cancel·lació.
* Precondició: L'usuari ha seleccionat l'opció de cancel·lar l'acció en qualsevol punt del procés.
* Postcondició: L'operació d'esborrament ha estat cancel·lada i l'usuari pot continuar utilitzant el sistema normalment.
* Retorna: No aplica.

1. **Diagrama de classes**

****

6.1. **Restriccions d’Integritat Textual**

**RIT1: noHiHaDuplicatsCorreuElectronic**

* Descripció: S'ha de garantir que cada Usuari tingui un correuElectronic únic dins del sistema. Això significa que no es pot registrar un nou usuari amb un correu electrònic que ja estigui assignat a un altre usuari.
* Expressió: usuari->forAll(u1, u2 | u1 <> u2 implies u1.correuElectronic <> u2.correuElectronic)

**RIT2: preuElementPositiu**

* Descripció: El preu de cada ElementCompra ha de ser un valor positiu, el que reflecteix la naturalesa d'una transacció comercial on el preu no pot ser negatiu.
* Expressió: ElementCompra->forAll(e | e.preu > 0)

**RIT3: preuPaquetInferiorAlSumatoriPreusIndividuals**

* Descripció: El preu total d'un PaquetVideojocs ha de ser menor o igual a la suma dels preus individuals dels Videojocs que inclou.
* Expressió: PaquetVideojocs->forAll(p | p.preuJocs <= p.videojocs.preu->sum())

**RIT4: qualificacioEdatDinsElsValorsPermesos**

* Descripció: La qualificacioEdat d'un Videojoc ha d'estar dins dels valors estandarditzats acceptats i reconeguts per les normatives de classificació d'edat.
* Expressió: Videojoc->forAll(v | v.qualificacioEdat = 3 or v.qualificacioEdat = 7 or v.qualificacioEdat = 12 or v.qualificacioEdat = 16 or v.qualificacioEdat = 18)

**RIT5: dataLlancamentEsPassadaOActual**

* Descripció: La dataLlancament d'un Videojoc ha de ser igual o anterior a la data actual, assegurant-se que no s'està registrant un videojoc amb una data de llançament futura.
* Expressió: Videojoc->forAll(v | v.dataLlancament <= today())

**RIT6: compraEsAmbElementsExistents**

* Descripció: Cada Compra només pot contenir ElementCompras que representin Videojocs o PaquetVideojocs que realment existeixin i estiguin disponibles a la data de la compra.
* Expressió: Compra->forAll(c | c.elements->forAll(e | Videojoc->includes(e) or PaquetVideojocs->includes(e)))

6.2. **Informació derivada**

**ID1: preuPaquetEsIgualALaSumaDePreusVideojocs**

* Descripció: El preuJocs d'un PaquetVideojocs es calcula com la suma de tots els preus dels Videojocs que conté.
* Expressió: PaquetVideojocs->forAll(p | p.preuJocs = p.videojocs.preu->sum())