

# Enginyeria del Software

Grau en Enginyeria Informàtica

Escola d'Enginyeria

**Curs 2020 - 2021**

Universitat Autònoma de Barcelona (UAB)

---

Document: **Especificacions**

(Versió 4.0)

**Grup 432-04**

## Document d'especificacions: Sprint 1

### Llista dels integrants de l'equip 432-4

- Equip de desenvolupament (alumnes):
  - Vaneet Kumar ([Perfil GitHub](#))
  - Óscar García ([Perfil GitHub](#))
  - Braian Maldonado ([Perfil GitHub](#))
  - Pol Pagès ([Perfil GitHub](#))
  - Austin León ([Perfil GitHub](#))
- Supervisor (professor):
  - Daniel Soto

### Enllaços

- Projecte Azure DevOps:  
<https://dev.azure.com/UAB-EnginySw-432-04/UAB-ES-432-04>
- Projecte GitHub:  
<https://github.com/polpages1999/es2021uab.git>

### Llista formal de requisits organitzada per categories

#### Requisits Funcionals (RF)

1. RF-1-01 (v1.0): **Borsa de treball per a transportistes**: Implementar una borsa de treball per als transportistes de la plataforma.
2. RF-1-02 (v1.0): **Ingressos**: A través de comissions i publicitat (requisits RF-3-01 i RF-6-01), apart de les vendes, el servei obtindrà ingressos a base d'aquests.
3. RF-2-01 (v1.0): **Sistema de sol·licitud de comanda**. Una sol·licitud de comanda és demanar al sistema a través d'una aplicació amigable una llista de queviures que necessita la persona. Les comandes podran contenir un o més productes, amb quantitats diferents per a cada producte. El client haurà d'especificar la quantitat, però no té perquè cercar què vol.

4. RF-2-02 (v1.0): **Presentació d'ofertes.** La presentació d'ofertes és notificar a la persona de quines tendes properes li poden servir el que demana. Aquesta oferta llistarà els productes que se li serviran, indicant quantitat, preu, etc.; així com el dia d'entrega i hora aproximada. I també el preu total, incloent els càrrecs addicionals com pot ser el transport.
5. RF-2-03 (v1.0): **Sistema de selecció de comanda i enviament.** La selecció de la comanda i enviament succeeix quan l'usuari vol acceptar una de les ofertes rebudes. Llavors es genera una comanda en ferm i el botiguer rep totes les dades necessàries per a servir-la. De la mateixa manera, l'usuari ha de ser proveït amb un mapa en el qual es pugui veure on es troba el venedor, amb data i hora de lliurament aproximades.
6. RF-3-01 (v1.0): **Pagaments i comissions:** Apart de l'especificat al requisit RF-2-02, a l'hora de vendre també s'haurà de tenir en compte el càlcul d'una comissió amb una variable pactada amb la botiga, pagant només pel volum de vendes en comptes de ser un valor específic. La comissió es farà en el moment de pagament.
7. RF-3-02 (v1.0): **Implementar retorn de pagaments.** Hi haurà un procés d'avaluació i supervisió que caldrà seguir per a que l'abonament s'accepti.
8. RF-4-01 (v1.0): **L'hora de transportar:** La botiga ha de ser capaç d'escollir si vol a un transportista lliurant la comanda del client o si ho vol fer de manera local a través d'un SW. Si la botiga no té cap transportista disponible, ho haurà de fer de manera local. Amb les distàncies recorregudes pels transportistes externs es calcularà quant ha de pagar la botiga al transportista, i aquests diners aniran de manera directa al compte bancari del transportista. Per tal de reduir aquests costos a la botiga, també es calcularà la ruta planificada amb menys despeses, per tal que, per exemple, dues comandes puguin ser lliurades en un mateix transport.
9. RF-4-02 (v1.0): **GPS:** Els transportistes (o botiguers locals) han de disposar d'un GPS que envia dades a la plataforma quan les comandes estiguin en fase de distribució. Així doncs, sabrà de manera exacta on es troba el transportista. Amb això, podrem accedir al requisit RF-4-01.
10. RF-5-01 (v1.0): **Canvis a comandes:** La plataforma ha de ser capaç de recalculer els costos de transport quan es canviï alguna cosa del lliurament. **NOMÉS ES PODRÀ SI LA COMANDA ESTÀ REBUDA**, la qual cosa comportarà un nou càrrec al preu.

11. RF-5-02 (v1.0): **Votacions:** Els clients podran votar a través de la plataforma a la botiga que els ha atès i el sistema s'encarregarà en calcular, entre totes les votacions, la valoració general (un simple càlcul de mitjana).
12. RF-6-01 (v1.0): **Publicitat:** Com citat al requisit RF-3-01, certes comissions seran aplicades per les vendes pel servei, però també s'hauran de calcular aquests ingressos amb una suma de la publicitat que hi entri, amb els banners que siguin. Aquesta publicitat serà mostrada a través de la plataforma als clients.
13. RF-7-01 (v1.0): **Registraments:** L'usuari es podrà registrar de qualsevol manera (s'ha de preparar la programació per això), tant per mail, com mòbil, com altres tipus d'autenticació, com Facebook o Google, entre d'altres. En canvi, si un vol ser transportista, només es podrà registrar amb dades fiscals per poder pagar-li el que li correspon. Els usuaris es podran donar de baixa o modificar o eliminar les seves dades quan vulguin.

### **Requisits No Funcionals (RNF)**

1. RNF-1-01 (v1.0): **Venedors:** Ni els venedors ni els transportistes no poden formar part d'un grup empresarial. Els venedors han de ser de particulars.
2. RNF-2-01 (v1.0): **Usuaris i aplicació:** L'aplicació o plataforma ha de ser amigable i comprensible per tot usuari que hi entri. La interfície de seguiment de la comanda ha de ser també amigable per l'usuari.
3. RNF-2-02 (v1.0): **Mapa (SW):** La interfície utilitzada pel mapa serà, preferiblement, Google Maps.
4. RNF-2-03 (v1.0): **Màxim comandes:** El màxim de comandes que un usuari podrà tenir en estat d'enviament és 100.
5. RNF-3-01 (v1.0): **Pagaments:** Els pagaments podran ser en efectiu, Paypal, Bitcoins o targeta, encara que les botigues podran escollir quines accepten.
6. RNF-3-02 (v1.0): **Protecció:** La plataforma ha de ser capaç de protegir als venedors per evitar frau. Per això, de primera mà, s'acceptaran reclamacions de les botigues en cas que el pagament de la comanda s'hagi anul·lat. No seran automàtiques, però si controlades per una companyia externa. D'aquesta manera, també estaran protegits els usuaris.

7. RNF-4-01 (v1.0): **Transportistes:** La botiga s'encarregarà de pagar als transportistes, no la plataforma, ja que aquesta no contracta a cap.
8. RNF-5-01 (v1.0): **Compte amb valoracions:** Per protegir la transparència de les botigues, el sistema proveirà igualment d'un sistema de control i valoració del servei. De tal forma que dins les ofertes que es presentaran als clients, juntament amb el preu final i el nom de la botiga, també hi haurà sempre una valoració de la botiga. Així doncs, després de realitzar-se el lliurament de la comanda, se li preguntarà al client sempre per la seva valoració.
9. RNF-6-01 (v1.0): **Sobre la publicitat:** El volum dels banners que es mostraran serà variable, entre un mínim de 1 per oferta i una màxim de 10, encara que es tractarà de que siguin els menors possibles, per no semblar invasiu.
10. RNF-7-01 (v1.0): **Protecció sobre usuaris i transportistes en l'enregistrament:**  
Estaran protegits perquè la plataforma no els hi cobrarà res i l'ús del servei serà obert. Però així i tot caldrà registrar-se en el servei per utilitzar-lo i tenir-ne les dades necessàries per a poder fer els lliuraments de les comandes, així com també per mostrar la publicitat.
11. RNF-8-01 (v1.0): **Disponibilitat:** La disponibilitat de la plataforma haurà de ser 24/7, amb les finestres de suport habituals per a realitzar les operacions de manteniment necessàries. En aquest sentit, com a preconditioni tècnica, es defineix que s'utilitzarà Amazon Web Services com proveïdor IaaS (Infrastructure as a Service) per garantir aquest nivell de disponibilitat amb un cost raonable.
12. RNF-8-02 (v1.0): **Dades clients:** Totes les dades dels clients hauran d'estar enregistrades en SQL.
13. RNF-8-03 (v1.0): **UI:** Totes les UI contindran el logo de la plataforma, el color predominant haurà de ser el verd.
14. RNF-8-04 (v1.0): **Tipografia:** Només es podran fer servir tipografies Open Source.
15. RNF-8-05 (v1.0): **BBDD:** La base de dades amb tota la informació dels clients haurà d'estar en una base de dades relacional amb interfície SQL. Com a preconditioni tècnica es decideix que el motor haurà de ser MySQL.

**Llista de les històries d'usuari**

- Hi haurà l'equip de manteniment o solució d'errors, en el cas que es trobes alguna falla o bug.
- Com a dissenyadors, dissenyaran un logo que representi la plataforma.
- Com a dissenyadors, treballaran per tal de marcar uns patrons per al disseny de la plataforma. Colors, estructures, interfície amigable.
- Com a usuari, hauré de registrar-me, és a dir aportar les meves dades per poder realitzar i rebre comandes.
- Com a client:
  - Podré realitzar una valoració, ja sigui del producte, botiga i/o procés d'entrega o transportista.
  - Si no és el producte que esperava o no estic satisfet, tinc el dret a fer una devolució del producte en qüestió.
  - Puc consultar la pàgina web en qualsevol moment del dia.
  - Puc consultar el estat de la meva comanda (tracking).
  - Puc fer el pagament d'un producte.
  - Puc rebre el pagament d'un producte.
  - Puc modificar el meu perfil, es a dir la meva informació i dades.
  - Podré contactar amb atenció al client per resoldre dubtes i problemes.
  - Puc fer una búsqueda dels productes o botigues que siguin del nostre interès. A més a més poder veure això dins d'un mapa.
  - Puc demanar un transportista per entregar una comanda. En cas de no haver cap disponible, es podra demanar un transportista local, es a dir, un veí de la zona que estigui disposat.
- Com a botiguer, he de poder calcular el costos per km que ha de fer un transportista a l'hora d'entregar una comanda i fer aquest pagament.
- El GPS pot calcular una o més rutes, per tal de optimitzar les entregues que ha de fer un repartidor.
- Com a publicitant, puc publicar anuncis i ofertes dels meus productes.
- Com a transportista he de poder:
  - Oferir el meu servei registrant-me en una borsa de treball.

- Aceptar les comandes per dur a terme el transport.
  - Fer el recorregut que m'han assignat.
  - Quan estic fent una entrega, he d'activar el GPS per poder facilitar el tracking.
- El servei de Maps serà de gran ajuda al transportista per poder fer una entrega.
- Es podrà associar el servei de Gmail o Facebook a la nostra plataforma per tal de poder fer un registre o iniciar sessió.
- El servei de Google/Facebook Ads, serà imprescindible per poder fer la publicitat necessària dins la nostra plataforma.
- El GPS serà activat pel transportista, quan està fent una o més entregues. Aquest enviarà la ubicació en tot moment per saber on es troba la comanda.
- Aquest GPS, estarà enllaçat amb el servei de Google Maps, i ha de tenir la seva pròpia interfície, a més a més, ens serà útil per saber quants km s'han fet durant el recorregut.
- Com a revisors:
  - Han de supervisar les sol·licituds de retorn. Això es farà mitjançant un procediment a més d'una avaluació d'aquest.
  - Per altra banda, també han d'encarregar-se de les reclamacions que pugui haver, ja sigui per als botigues com els clients.
- Per tal de tenir un suport de la web es treballarà amb Amazon Web Services.
- L'equip de desenvolupadors:
  - Haurà de fer el codi que donarà lloc a la nostra plataforma.
  - Una vegada, la tenim en funcionament, la seva feina es transforma en dur a terme les noves versions i actualitzacions.
  - Aquest codi tindrà funcionalitats essencials com el sistema de registre i login. També hi ha sistema de base de dades, poder fer un pagament(API), implementar la interfície de Maps i calcular la ruta òptima, etc.
  - També, hauran de treballar amb els encarregats de disseny per implementar les seves idees i propostes.
- Per una banda tenim els publicitans externs, que mitjançant els serveis de Ads podran anunciar-se dins la nostra plataforma.

- També hi ha els propis venedors, que formen part de xarxa local. Aquests mostraran les ofertes que tenen al seu negoci i que poden ser interessants pel client. Mitjançant aquest servei, podrem dur a terme les transaccions entre client i botiga.
- Un dels mètodes de pagaments més bàsic serà mitjançant la tarjeta bancària. S'utilitzarà la passarel·la bancària associada a la tarjeta en qüestió. També hi haurà la possibilitat d'associar el paypal. Una altra alternativa, és el pagament en Bitcoin, aquest no està associat a cap banc. Finalment hi ha la forma tradicional de fer-ho en efectiu.

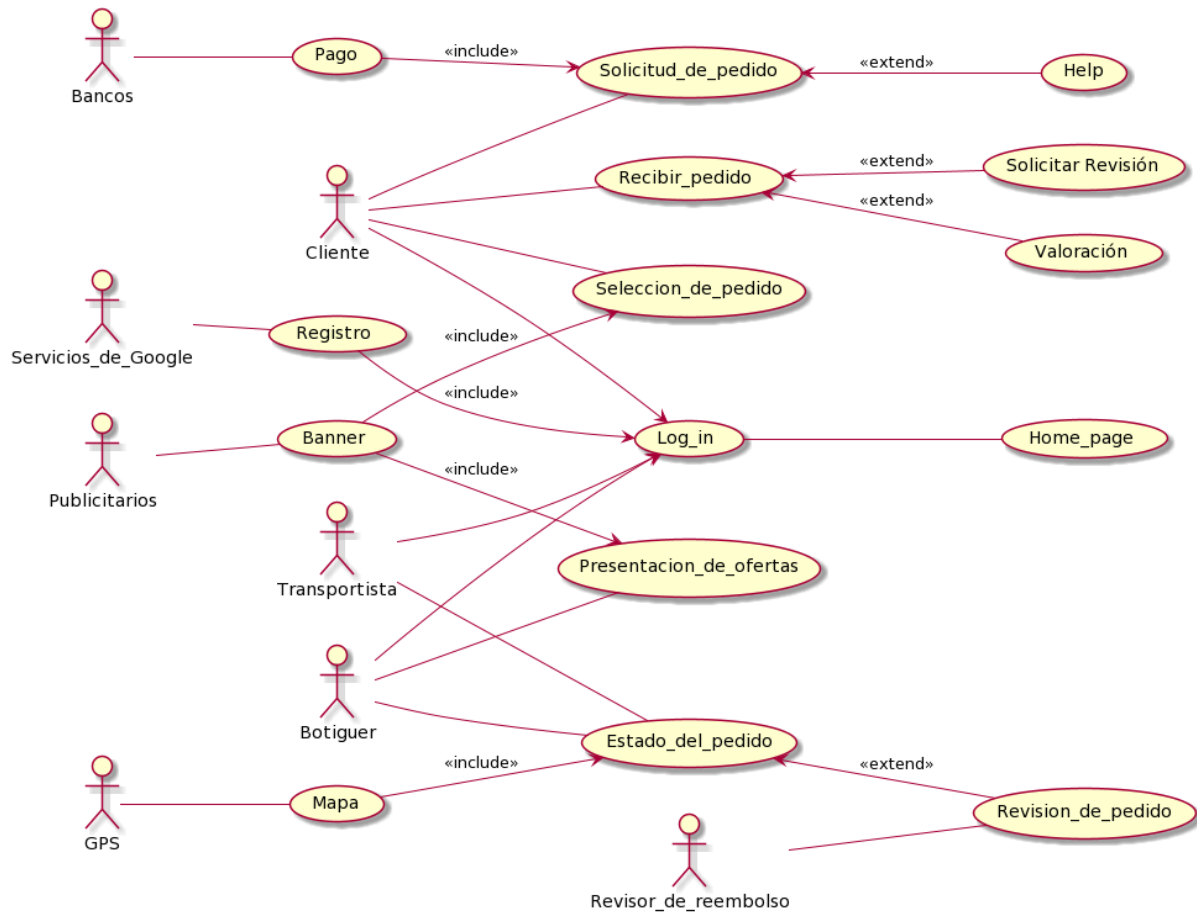


**Llista dels perfils d'usuari identificats**

- Botiguer
- Client
- Transportista
- Empresa de tercers que gestioni les reclamacions
- Publicitants
- Servei de Banc (pagaments)
- Serveis de Google
- GPS

## Sprint #2

## Diagrama de Casos d'Ús



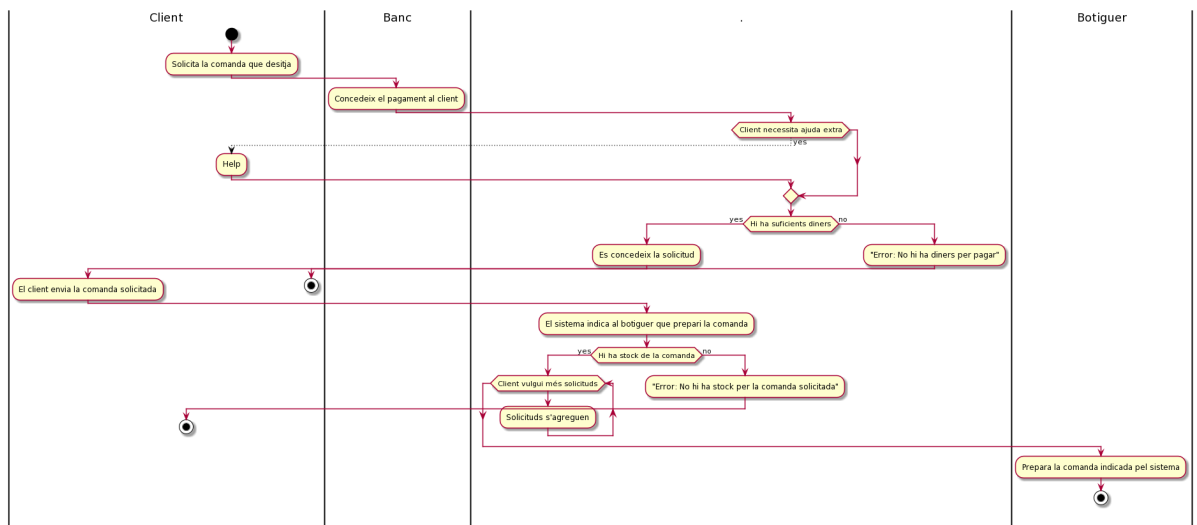
Hem escollit els actors “Publicitarios, Cliente, Bancos, GPS, Botiguer, Revisor\_de\_reembolso, Servicios de Google i Transportista”, perquè representen el comportament del sistema dut a terme per la guia de comandament de la plataforma. En aquest diagrama de casos d’ús es troben els requeriments funcionals “Solicitud\_de\_pedido, Estado\_del\_pedido, Seleccion\_del\_pedido, Presentacion\_de\_ofertas i Recibir\_pedido”, necessaris per poder demanar i rebre la comanda de manera correcte. L’actor “Cliente” interactua amb el requisit funcional “Seleccion\_de\_pedido” ja que és ell qui s’encarrega d’escollir què vol comprar. També interactua amb “Solicitud\_del\_pedido” perquè és ell qui s’encarrega de demanar la comanda. Està connectat amb “Estado\_del\_pedido” perquè ell ha de saber on és el seu producte.

També està connectat amb “Recibir\_pedido” perquè un cop rebi la comanda, pugui demanar una revisió o fer una valoració. I, per últim, està connectat amb “Log\_in”, com “Botiguer i Transportista”, per poder realitzar comandes a la pròpia plataforma.

L'actor “Botiguer” està connectat a “Presentacion\_de\_ofertas” ja que és ell qui s'encarrega d'oferir al client certes comandes que podria realitzar. També està connectat a “Estado\_del\_pedido”, perquè ell ha de saber també on és la comanda que “Cliente” ha demanat (igual que “Transportista”). Està també connectat a “Log\_in” per tal de poder realitzar el seu treball. L'actor “Transportista” està connectat a “Estado\_del\_pedido” perquè és ell qui porta la comanda a “Client”. Fa “Log\_in” per tal de poder cobrar i realitzar el seu treball. Els “Publicitarios” realitzen un Banner per mostrar anuncis mentre “Cliente” demana una comanda o quan se li està presentant una oferta. “Bancos” està connectada a “Pago”, ja que són ells qui tenen els diners per tal de poder pagar segons el tipus de pagament que volen realitzar els “Clients”. L'actor “GPS” està connectat a “Mapa” per tal de proporcionar aquest a l'estat de la comanda. “Servicios de Google” està connectat a “Registro” perquè és ell qui dóna suport digital per tal que qualsevol persona pugui registrar-se a la plataforma.

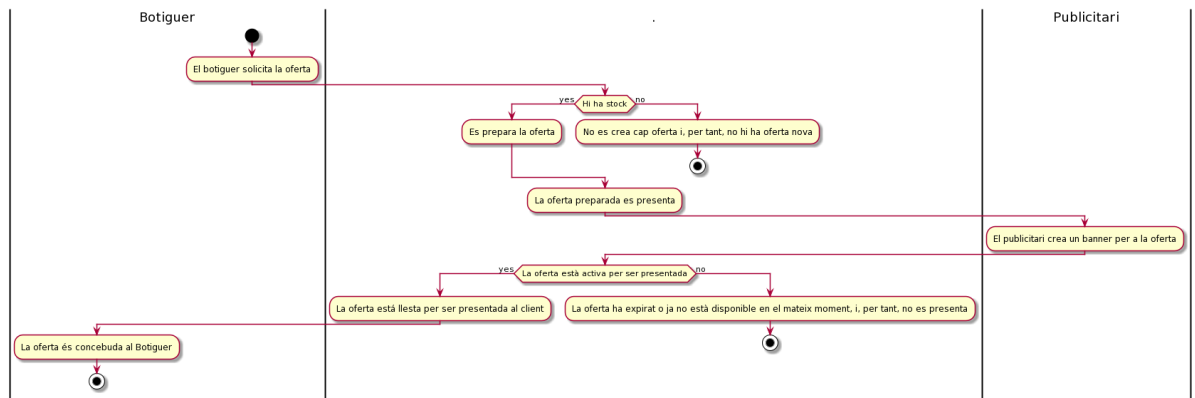
## Diagrames d'activitats

- sprint2\_diag\_activitats\_sollicitud\_comanda:



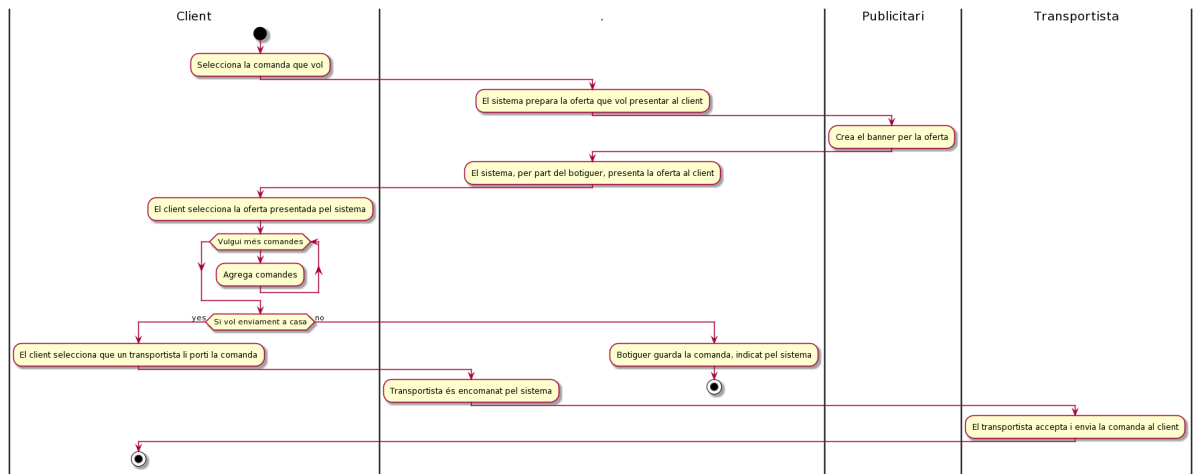
Els actors d'aquest diagrama són: "Client, Banc, Sistema (.) i Botiguer". L'estat inicial es troba al client, que sollicita la comanda que desitja com a activitat inicial. Després, el banc concedeix el pagament al client i el sistema passa a ser l'actor, que comprova si el client necessita ajuda extra; si diu que sí, el client selecciona l'activitat "Help" i l'estat retorna al sistema; si diu que no, es passa a comprovar directament si hi han suficients diners. Si no hi han suficients diners, s'obté "Error: No hi ha diners per pagar" i es finalitza. Si hi han diners, el sistema concedeix la sollicitud i el client envia la comanda sollicitada. A continuació, el sistema indica al botiguer que prepari la comanda i, si hi ha stock de la comanda, es comprova si el client vol més sol·licituds, i si és el cas, aquestes s'agreguen. Si no hi ha stock de la comanda, es finalitza. Finalment, quan la sollicitud està preparada, el botiguer prepara la comanda indicada pel sistema i es finalitza.

- sprint2\_diag\_activitats\_presentacio\_ofertes:



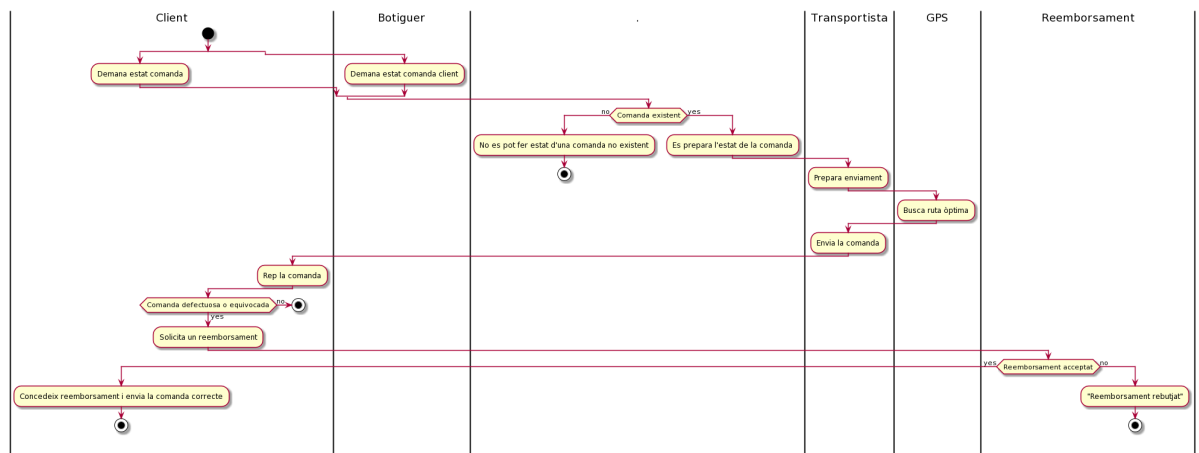
Els actors d'aquest diagrama són: "Botiguer, Sistema (.) i Publicitari". A l'estat inicial, el botiguer sol·licita l'oferta. A continuació es comprova si hi ha stock; si no hi ha stock no es crea cap oferta i, per tant, no hi ha oferta nova, i es finalitza; si hi ha stock, es prepara l'oferta i seguidament l'oferta preparada es presenta. Després, el publicitari crea un banner per a la oferta i el sistema comprova si l'oferta està activa per ser presentada. Si no està activa, significa que l'oferta ha expirat o ja no està disponible en el mateix moment i, per tant, no es presenta, i es finalitza. Si l'oferta està activa, s'enllesteix per ser presentada al client i el botiguer concep l'oferta, i a continuació es finalitza.

- sprint2\_diag\_activitats\_seleccio\_comanda:



Els actors d'aquest diagrama són: "Client, Sistema (.), Publicitari i Transportista". A l'estat inicial, el client selecciona la comanda que vol i seguidament el sistema prepara l'oferta que vol presentar al client. Després, el publicitari crea el banner per l'oferta i el sistema, per part del botiguer, presenta l'oferta al client. A continuació, el client selecciona l'oferta presentada pel sistema i si vol més comandes s'agreguen, si no, es comprova si vol enviament a casa. Si diu que no, el sistema indica que el botiguer guarda la comanda i es finalitza; si diu que sí, el client selecciona que un transportista li porti la comanda, el sistema encomana al transportista i el transportista accepta i envia la comanda al client, finalitzant a continuació.

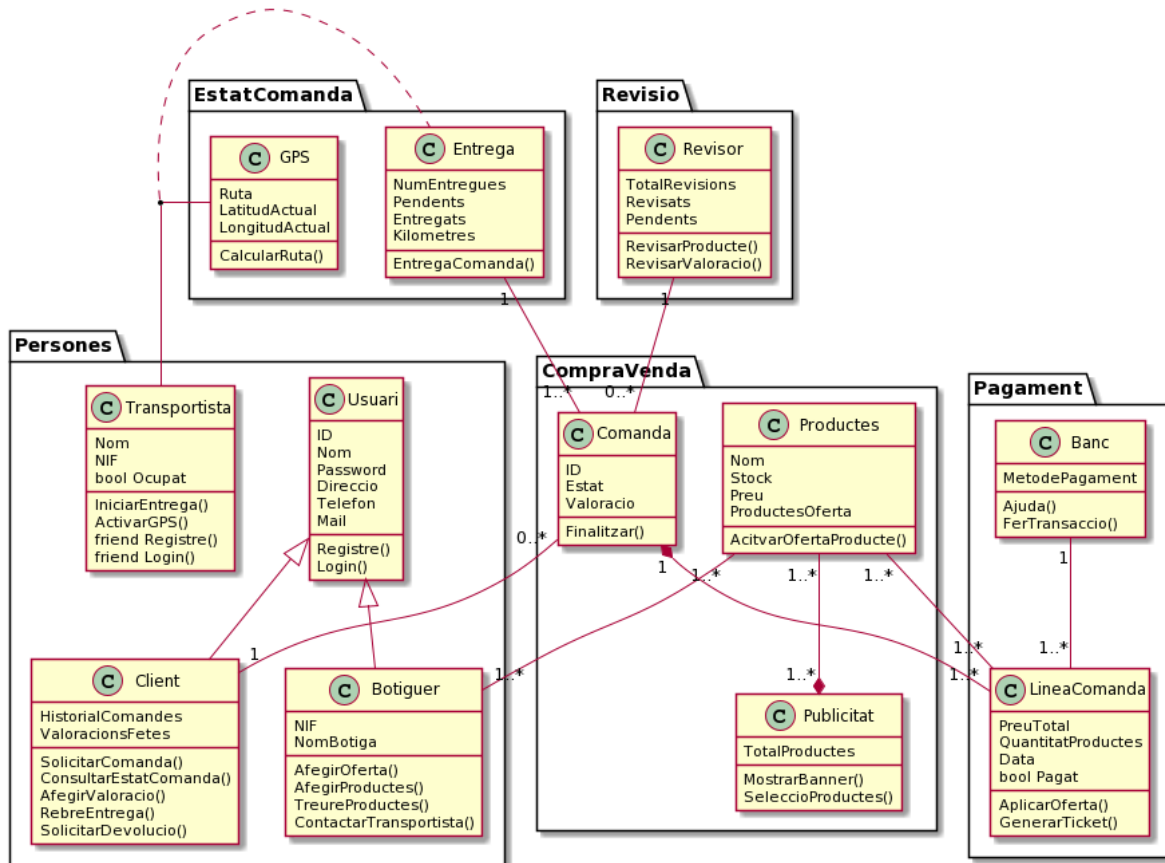
- sprint2\_diag\_activitats\_estat\_comanda:



Els actors d'aquest diagrama són: "Client, Botiguer, Sistema (.), Transportista, GPS i Reemborsament". A l'estat inicial, tant el client com el botiguer poden demanar l'estat de la comanda, i el sistema comprova si existeix aquesta. Si no existeix, el sistema indica que no es pot fer estat d'una comanda no existent i finalitza; si existeix, es prepara l'estat de la comanda, el transportista prepara l'enviament, el GPS busca la ruta òptima, el transportista envia la comanda, el client rep la comanda i comprova si està defectuosa o equivocada. Si no ho està, es finalitza. En canvi, si està defectuosa, el client sol·licita un reemborsament. El sistema de reemborsament pot no acceptar el reemborsament, i per tant es rebutja i finalitzem; o pot acceptar el reemborsament, pel que el client concedeix el reemborsament i envia la comanda correcte, i es finalitza a continuació.

## Sprint #3

## Diagrama de classes



En aquest diagrama trobem aquestes classes:

- GPS ([atributs = Ruta, LatitudActual, LongitudActual] [operacions = CalcularRuta()]).

Aquesta classe està relacionada amb Transportista, ja que aquest necessita el GPS per realitzar les seves operacions; i també està relacionada amb Entrega, ja que aquesta necessita els kilòmetres de la ruta calculada pel GPS.



- Usuari ([atributs = ID, Nom, Password, Direccio, Telefon, Mail] [operacions = Registre(), Login()]).

Aquesta classe és una generalització, i d'ella descendeixen les classes Client i Botiguer, les quals hereten les seves propietats.

- Productes ([atributs = Nom, Stock, Preu, ProductesOferta] [operacions = ActivarOfertaProducte()]).

Un producte pot pertànyer a més d'un Botiguer; un o més productes componen una o més publicitats; i un o més productes poden pertànyer a una o més d'una línia de comandes.

- Entrega ([atributs = NumEntregues, Pendants, Entregats, Kilometres] [operacions = EntregaComanda()]).

Aquesta classe obté un dels seus atributs gràcies a la relació amb GPS i una Entrega pot contenir una o més Comandes.

- Transportista ([atributs = Nom, NIF, bool Ocupat] [operacions = IniciarEntrega(), ActivarGPS(), friend Registre(), friend Login()]).

Aquesta classe està relacionada amb GPS i Entrega perquè obté informació d'ambdues per assignar els valors dels seus atributs.

- Client ([atributs = HistorialComandes, ValoracionsFetes] [operacions = SolicitarComanda(), ConsultarEstatComanda(), AfegirValoracio(), RebreEntrega(), SolicitarDevolucio()]).

Aquesta classe descendeix de la classe Usuari, i un Client pot tenir 0 o varies Comandes actives.

- Botiguer ([atributs = NIF, NomBotiga] [operacions = AfegirOferta(), AfegirProductes(), TreureProductes(), ContactarTransportista()]).

Aquesta classe descendeix de la classe Usuari, i un o més Botiguers poden contenir el mateix producte o més productes diferents.

- Publicitat ([atributs= TotalProductes, operacions = Banner(), SeleccioProductes()]).

Una o més publicitats poden estar composades d'un o més productes.

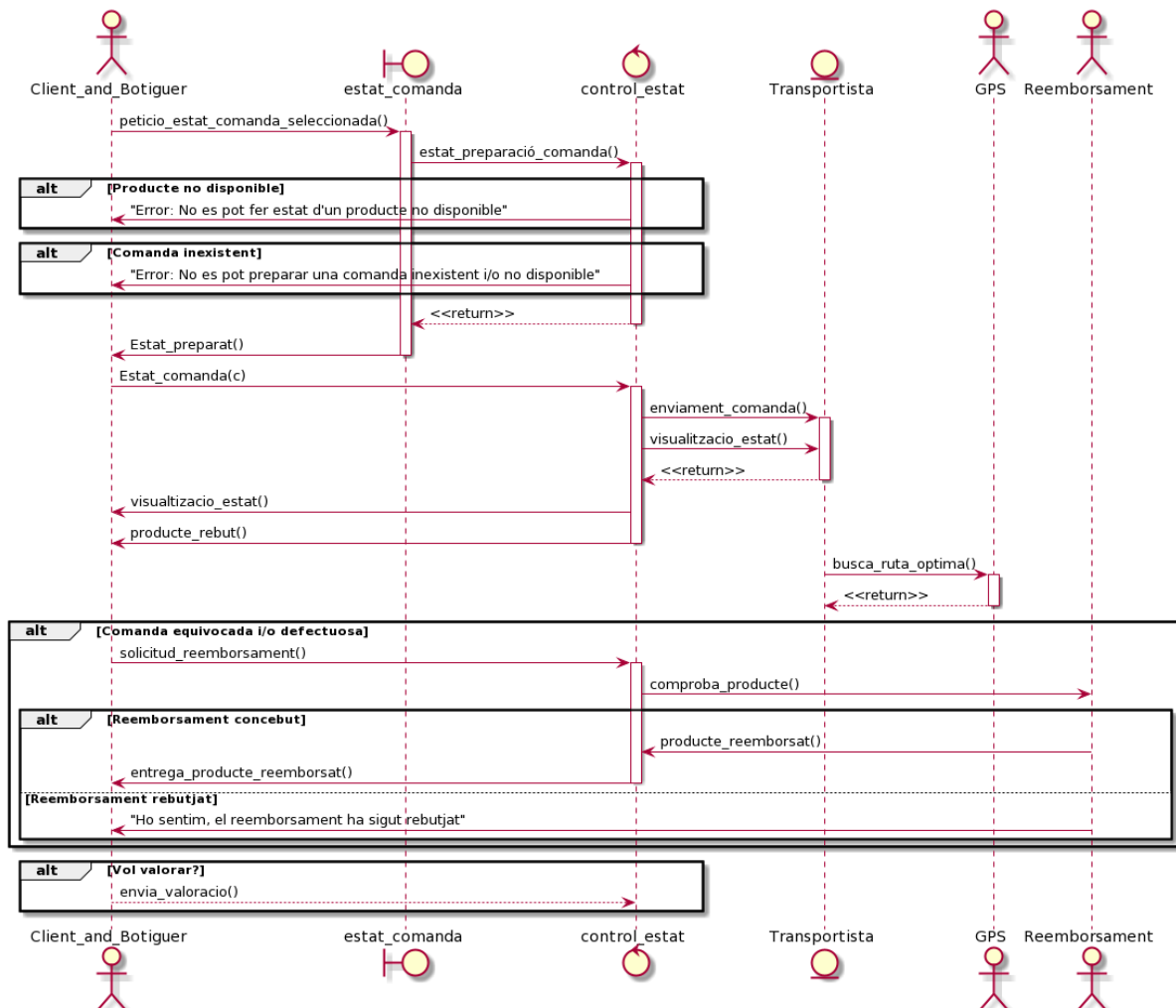
- Revisor ([atributs = TotalRevisions, Revisats, Pendants] [operacions = RevisarProducte(), RevisarValoracio()]).

Un revisor únicament pot no tenir cap comanda a revisar o en pot tenir varies, però aquesta revisió només estarà feta per un únic encarregat al mateix temps.

- Banc ([atributs = MetodePagament] [operacions = Ajuda(), FerTransaccio())).  
Un banc pot atendre una o més Línies de Comanda.
- Comanda [atributs = ID, Estat, Valoracio] [operacions=Finalitzar()].  
Una Comanda nomes pot pertànyer a un sol Client; una comanda en concreta només podrà estar en una unica entrega; i una Comanda està composta per una o més Línies de Comanda.
- LineaComanda ([atributs = PreuTotal, QuantitatProductes, Data, bool Pagat] [operacions = AplicarOferta(), GenerarTicket())).  
Una o més Línies de Comanda poden ser ateses per el mateix Banc; Una o més Línies de Comanda poden compondre la mateixa Comanda; i una o més Línies de Comanda poden assignar-se a un o més productes.
- Paquet de Persones: S'ha optat per agrupar les classes Transportista, Usuari i per tant les seves herències Client i Botiguer, ja que son els individus que formen part de la plataforma i faran us d'aquesta.
- Paquet de CompraVenda: En aquest s'agrupen les classes Comanda, Productes i Publicitat per que son els principals motors que formalitzaran les compres i vendes que es faran dins la nostra aplicació.
- Paquet de Pagament: En aquest cas, hi ha les classes Banc i LineaComanda ja que com es pot intuir, una vegada em fet una comanda amb tot allò que desitjem, aquesta s'haurà de tramitar mitjançant un pagament.
- Paquet EstatComanda: Una vegada d'ha realitzat la nostra transacció, voldrem saber en quina situació es troba la nostra comanda, es aquí on actuen les classes GPS i Entrega, que treballen conjuntament per donar informació sobre on es troba, quan serà entregada, etc.
- Paquet Revisio: Finalment, ja sigui que estem conformes amb la nostra compra o no, hi ha la possibilitat de fer una valoració com sol·licitar una devolució. I es aquí on actua la classe Revisor per tal de fer possible aquestes peticions del client.

## Diagrames de seqüència

- sprint3\_diag\_sequencia\_estat\_comanda:



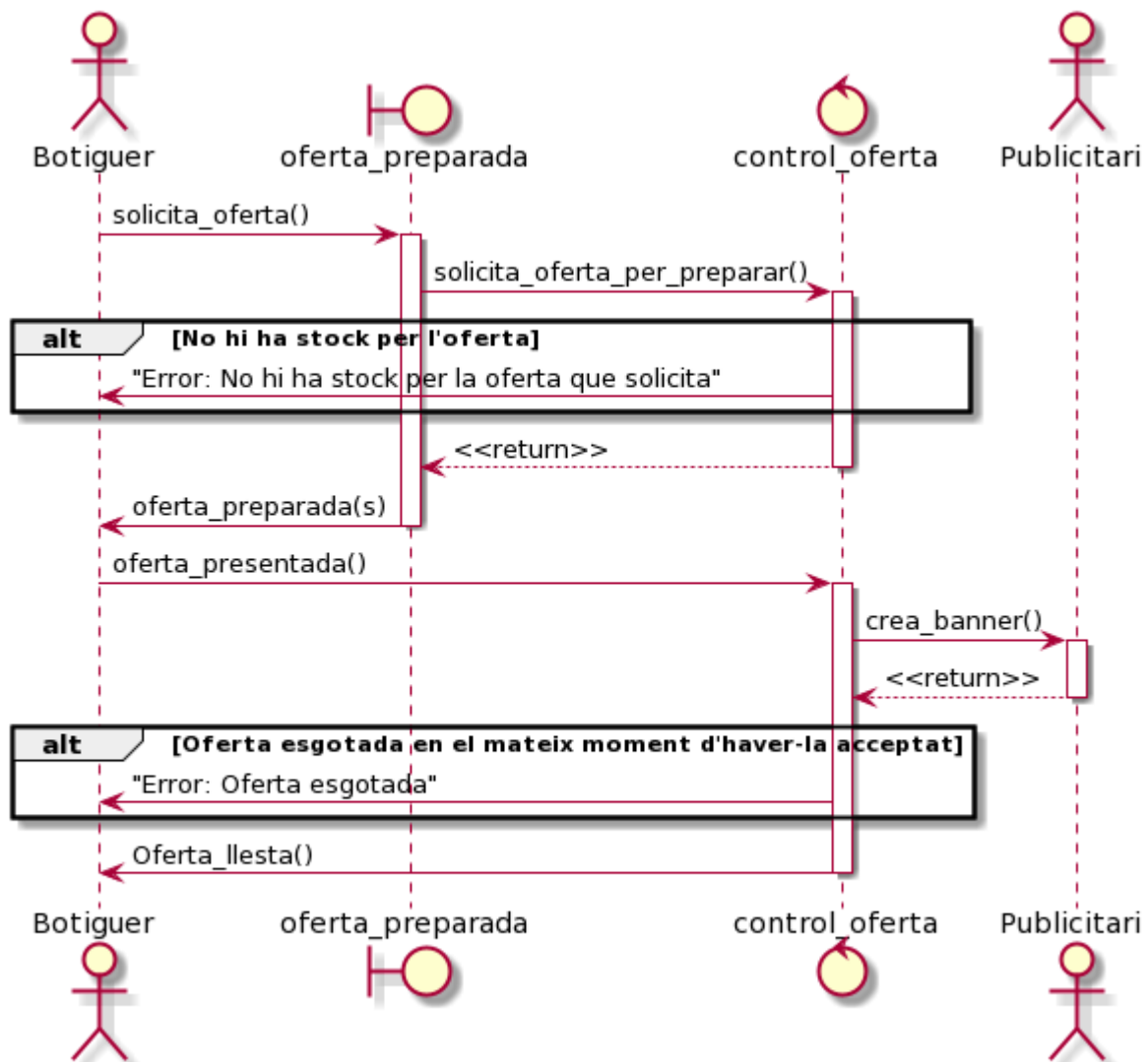
L'actor Client inicia l'acció `petició_estat_comanda_seleccionada()` i el Botiguer inicia `estat_preparació_comanda()`. Idealment, això ens retorna `estat_preparat(c)` per al Client i `estat_preparat(b)` per al Botiguer, confirmant que l'estat de la comanda està preparat per ser mostrat. En cas de que el producte no estigui disponible, ens retorna un error tant al Client com al Botiguer. A continuació, tant el Client com el Botiguer inicien l'acció `estat_comanda(c)` i `estat_comanda(b)`, respectivament. En aquest punt, el control d'estat inicia les accions `enviament_comanda()` i `visualitzacio_estat()` i el Transportista retorna el resultat d'aquestes operacions. Acte seguit, es presenta la `visualitzacio_estat()` tant al Botiguer com al Client, i a aquest últim se li entrega també `producte_rebut()`. A continuació, el Transportista sollicita `ruta()` al mapa, i aquest li retorna `troba_ruta()`. El Transportista també demana `troba_ruta_optima()` i el control del mapa busca `ruta_optima()`

mitjançant el GPS. Després se li retorna la ruta\_optima() al Transportista i es passa a les condicions especials.

Si la [comanda equivocada i/o defectuosa], el Client envia una sol·licitud de reemborsament, i el control d'estat realitza comprova\_producte() amb el Reemborsament. Si el Reemborsament retorna producte\_reemborsat(), ens trobem a [Reemborsament concebut] i el Client rep entrega\_producte\_reemborsat(). Si no s'accepta el reemborsament, el Client rebrà [Reemborsament rebutjat].

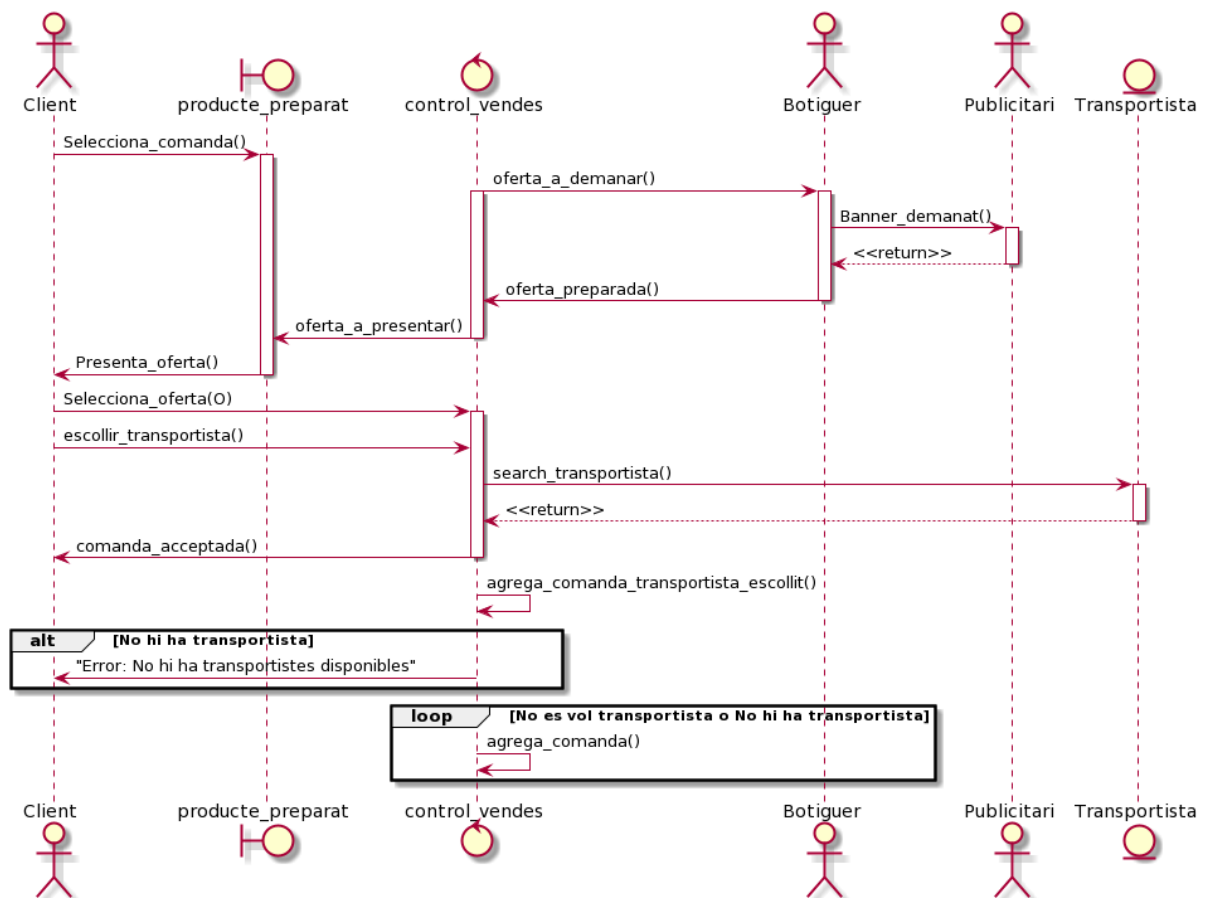
Finalment, el Client té l'opció de envia\_valoracio() si [Vol valorar?], i sinó, es finalitza la seqüència.

- sprint3\_diag\_sequencia\_presentacio\_ofertes:



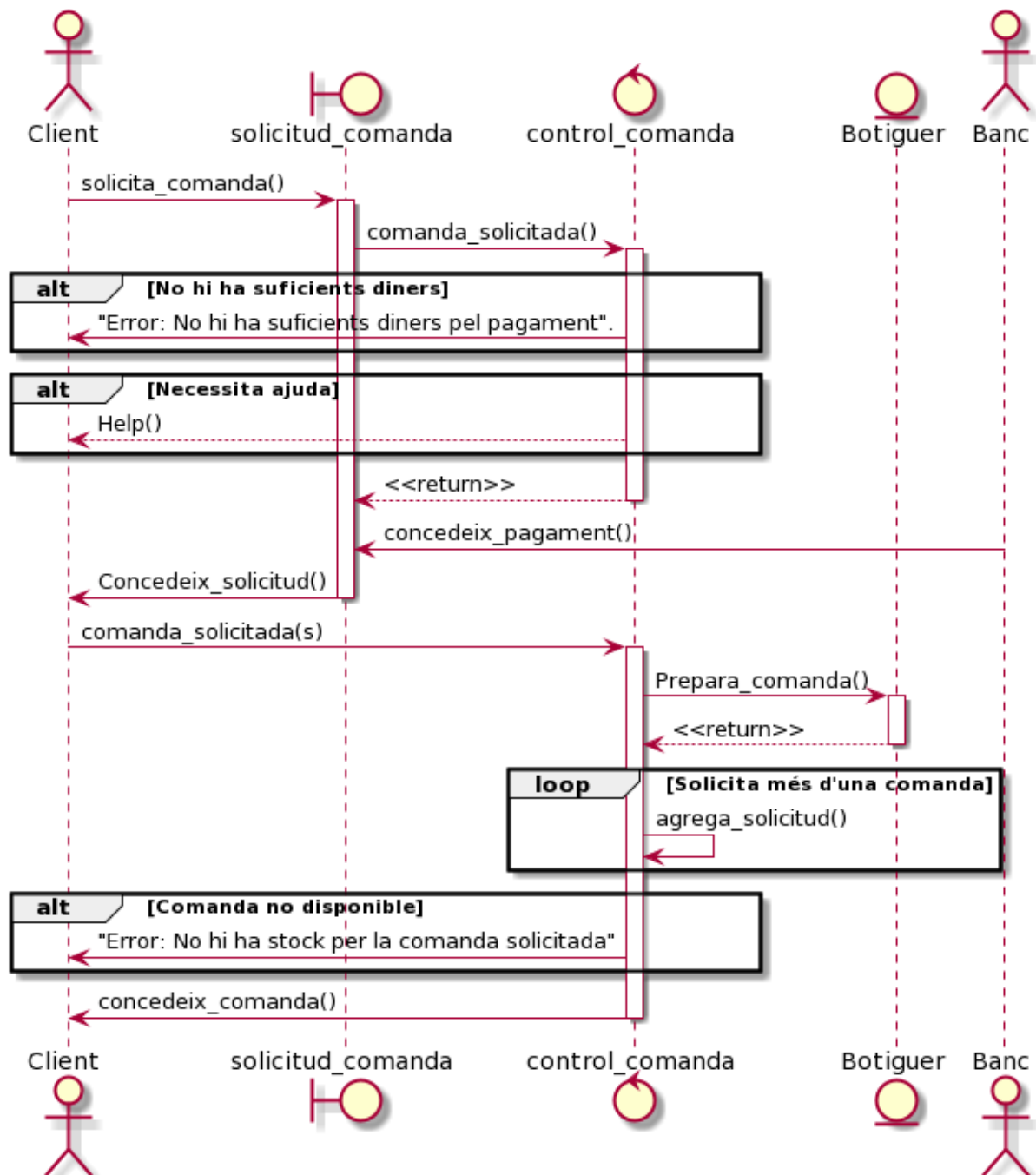
En aquest diagrama, identifiquem l'actor Botiguer com l'iniciador de l'acció `solicita_oferta()`, i idealment passa a l'estat d'`oferta_preparada`, en cas de no haver-hi stock suficient s'entraria en la condició especial `[No hi ha stock per l'oferta]` i retornaria un error. Quan la oferta es preparada, el Botiguer s'encarrega de presentar l'oferta a través de `oferta_presentada()` i es passa a l'estat de `control_oferta`. Aquí entraria un nou actor, el Publicitari, que rebia la petició de `crea_banner()` i retorna el banner publicitari. Seguidament ens podem trobar en una condició especial, que s'esgotés l'oferta `[Oferta esgotada en el mateix moment d'haver-la acceptat]`, si no es dona, s'entrega l'oferta completada a través de `Oferta_llesta()` i termina la seqüència.

- sprint3\_diag\_sequencia\_seleccio\_comanda:



Al nostre diagrama de seqüència de SELECCIÓ\_COMANDA, perquè es doni aquesta seqüència hi ha una sèrie de requisits previs que s'han de complir els quals es basen en que el producte estigui preparat. Per que es doni aquesta condició, el Client ha d'haver escollit la comanda amb Selecciona\_comanda(), el Botiguer ha de preparar l'oferta amb Oferta\_preparada(), el Publicitari ha d'haver dissenyat el banner de l'oferta a través de Publicitat() i l'oferta ha sigut presentada al Client. Identifiquem l'actor Client com l'iniciador de l'acció a través de selecciona\_oferta() i escollir\_transportista(), que passen l'estat a control\_vendes, es quan es passa a buscar el transportista amb search\_transportista() idealment la comanda es acceptada amb comanda\_acceptada() i s'afegeix el transportista escollit a la comanda a través de agrega\_comanda\_transportista\_escollit() es llavors quan es pot produir una excepció i es doni la condició especial de que no hi hagi un transportista disponible [No hi ha transportista] i entraria en un loop [No es vol transportista o no hi ha transportista] fins que s'aconsegueixi agregar la comanda amb agrega\_comanda(). Al completar-se termina la seqüència.

- sprint3\_diag\_sequencia\_solicitud\_comanda:



En aquest diagrama, identifiquem l'actor Client com l'iniciador de l'acció a través de `solicita_comanda()`, que passa l'estat a `solicitud_comanda`, seguidament el banc li proporciona el formulari a emplenar per realitzar el pagament a través de `concedeix_pagament()`. Aquí es pot donar la condició especial de que el Client no disposi dels fons necessaris per a realitzar el pagament [No hi ha suficients diners], el qual cancel·laria el procés de compra. també es possible que el Client requereixi d'assistència a l'hora de realitzar la compra [Necessita ajuda] i se li hagi de proporcionar l'assistència a

través de Help(). Seguidament si tot continua amb normalitat se li concedirà la sol·licitud al Client amb Concedeix\_sol·licitud() i ara li arribarà la comanda al Botiguer que s'encarregarà de preparar-la amb Prepara\_comanda() i entrarà en un loop dins que hagi preparat els diferents productes en agrega\_sol·licitud() llavors es pot donar la excepció de que s'entri en la condició especial de falta de stock [Comanda no disponible] la qual cosa detindrà el procés de preparació fins a tenir stock. Finalment se li concedeix la comanda al Client a través de Concedeix\_comanda() i termina la seqüència.



## Sprint #4

### Demostració de funcionament del TDD.

Per demostrar que el nostre test funciona, adjuntem una captura del seu funcionament:

```
✓ Tests passed: 7 of 7 tests – 16 ms

===== warnings summary =====
...\\Anaconda3\\lib\\site-packages\\pyreadline\\py3k_compat.py:8
C:\\Users\\ausle\\Anaconda3\\lib\\site-packages\\pyreadline\\py3k_compat.py:8: DeprecationWarning:
    return isinstance(x, collections.Callable)

-- Docs: https://docs.pytest.org/en/stable/warnings.html
===== 7 passed, 1 warning in 0.15s =====

Process finished with exit code 0
```