

Enginyeria del Software

Grau en Enginyeria Informàtica

Escola d'Enginyeria

Curs 2020 - 2021

Universitat Autònoma de Barcelona (UAB)

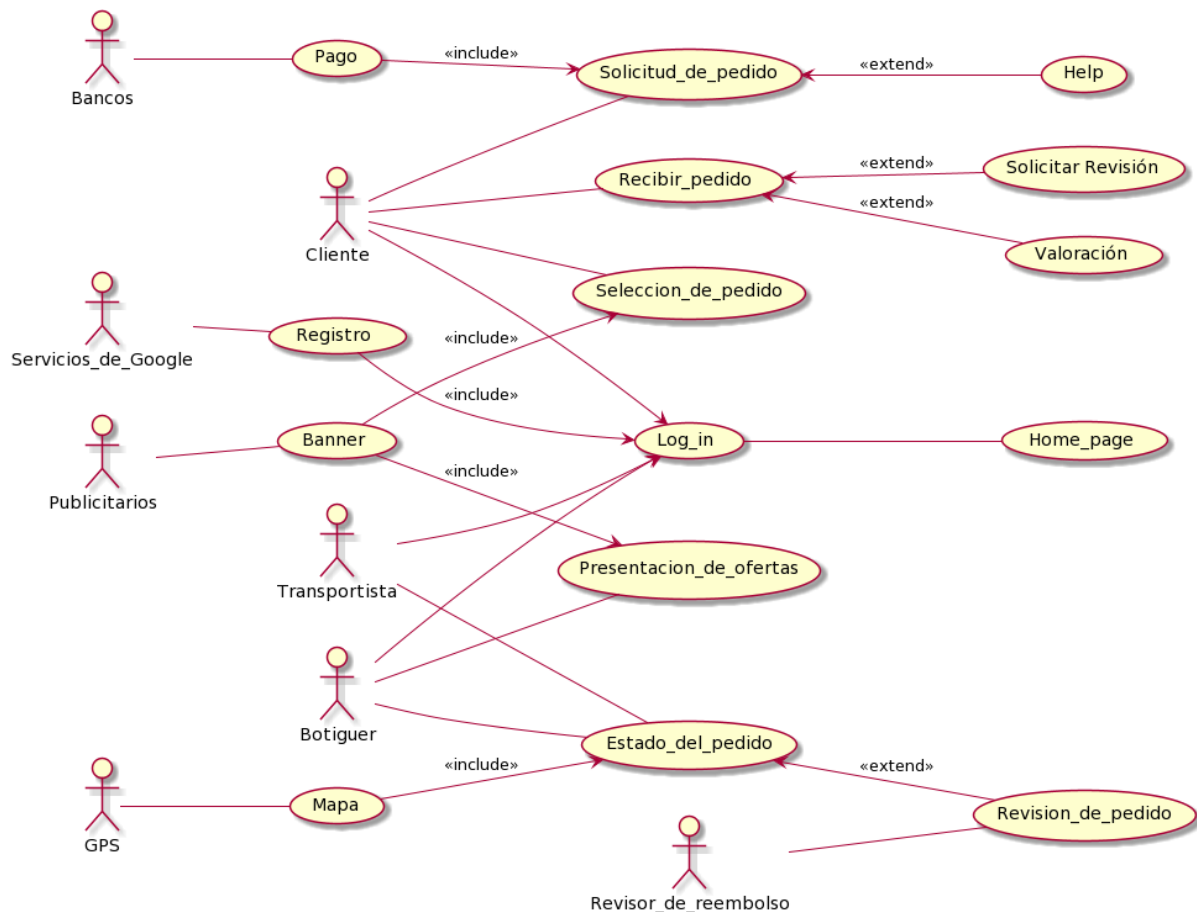
Document: **Plantilla Resultats**

(Versió 2.0)

Grup 432-04

Sprint #2

Diagrama de Casos d'Ús



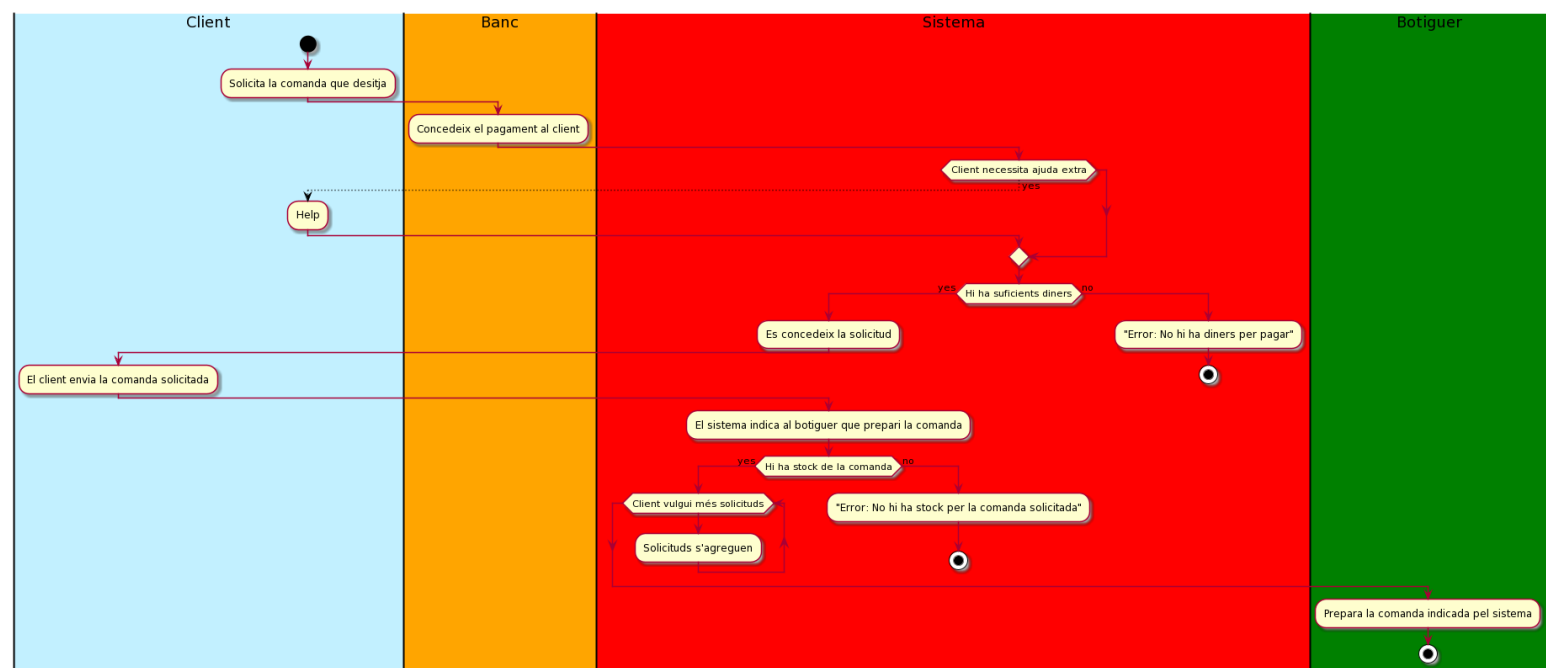
Hem escollit els actors “Publicitarios, Cliente, Bancos, GPS, Botiguer, Revisor_de_reembolso, Servicios de Google i Transportista”, perquè representen el comportament del sistema dut a terme per la guia de comandament de la plataforma. En aquest diagrama de casos d’ús es troben els requeriments funcionals “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.

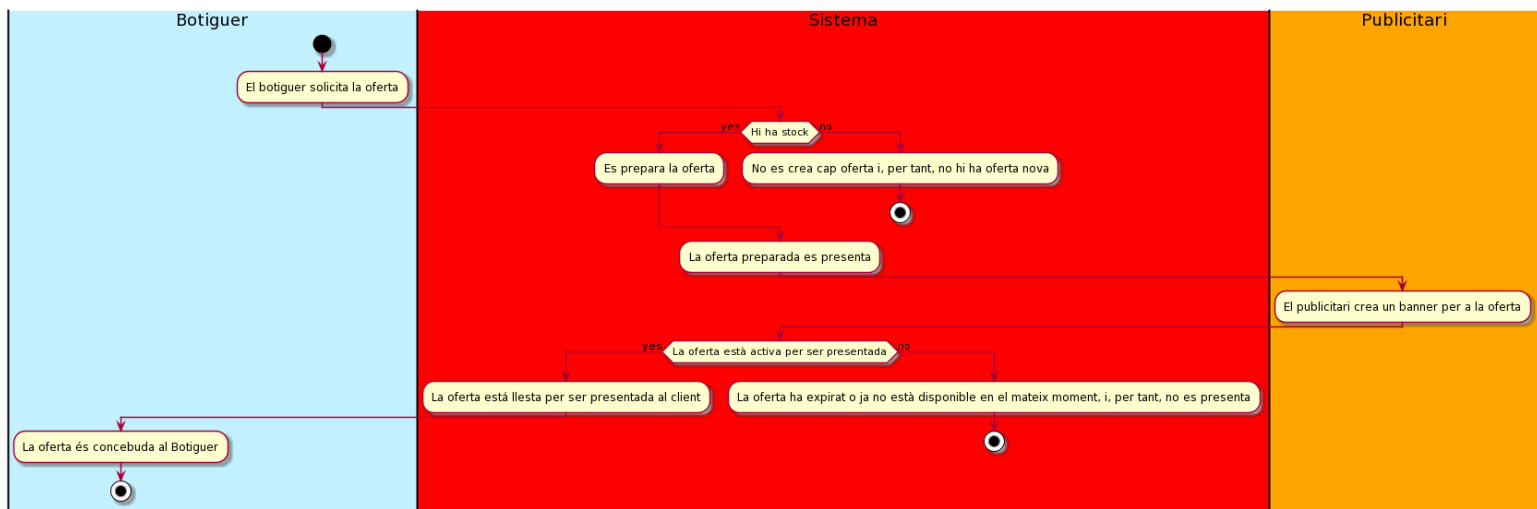
Diagrames d'activitats

- sprint2_diag_activitats_solicitud_comanda:



Els actors d'aquest diagrama són: "Client, Banc, Sistema i Botiguer". L'estat inicial es troba al client, que sol·licita 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 sol·licitud i el client envia la comanda solicitada. 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 sol·licitud 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 presentada, 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.

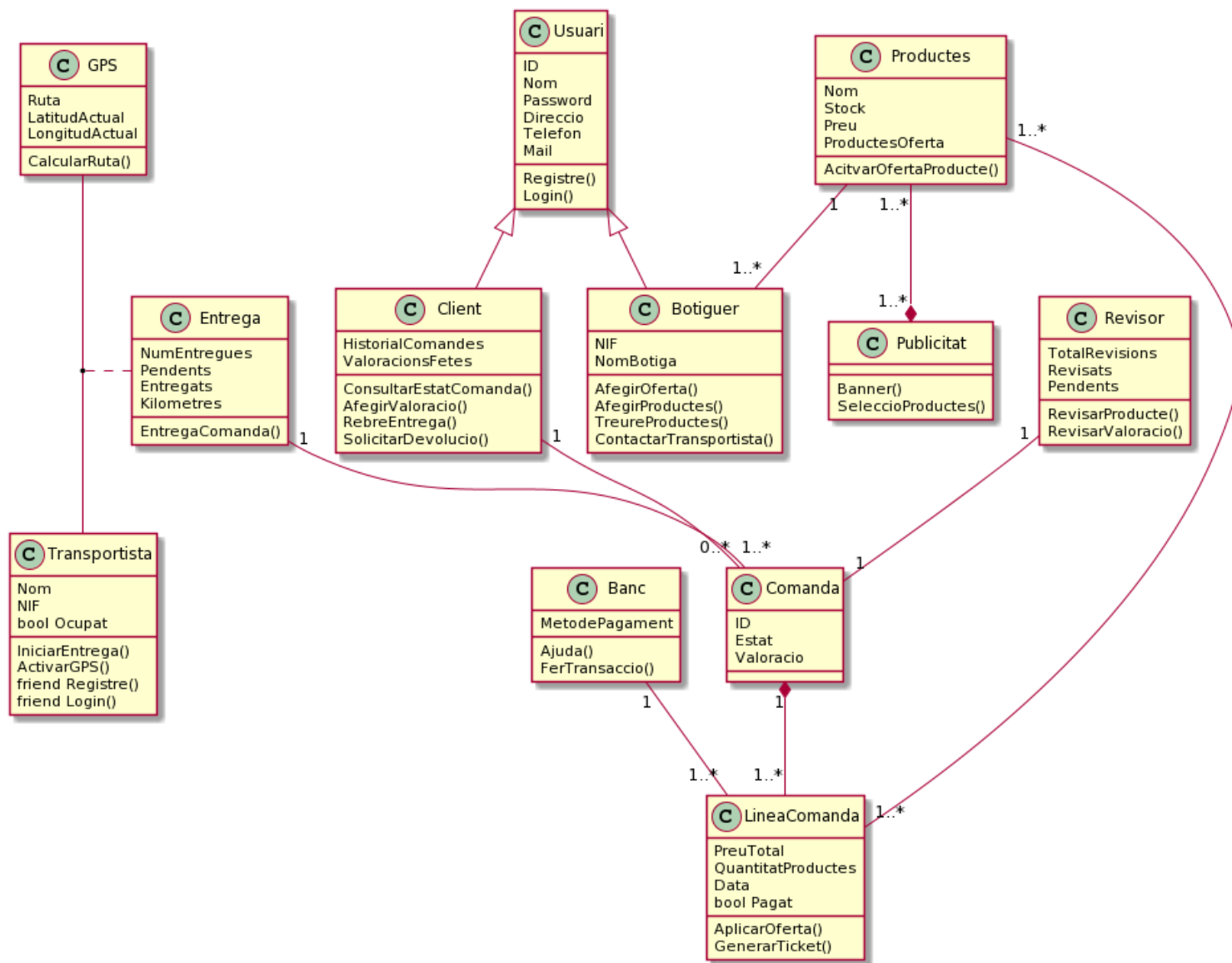
- sprint2_diag_activitats_estat_comanda:



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 = 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.

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

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

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

Un revisor únicament pot revisar una comanda alhora.

- Banc ([atributs = MetodePagament] [operacions = Ajuda(), FerTransaccio()]).

Un banc pot atendre una o més Línies de Comanda.

- Comanda [atributs = ID, Estat, Valoracio].

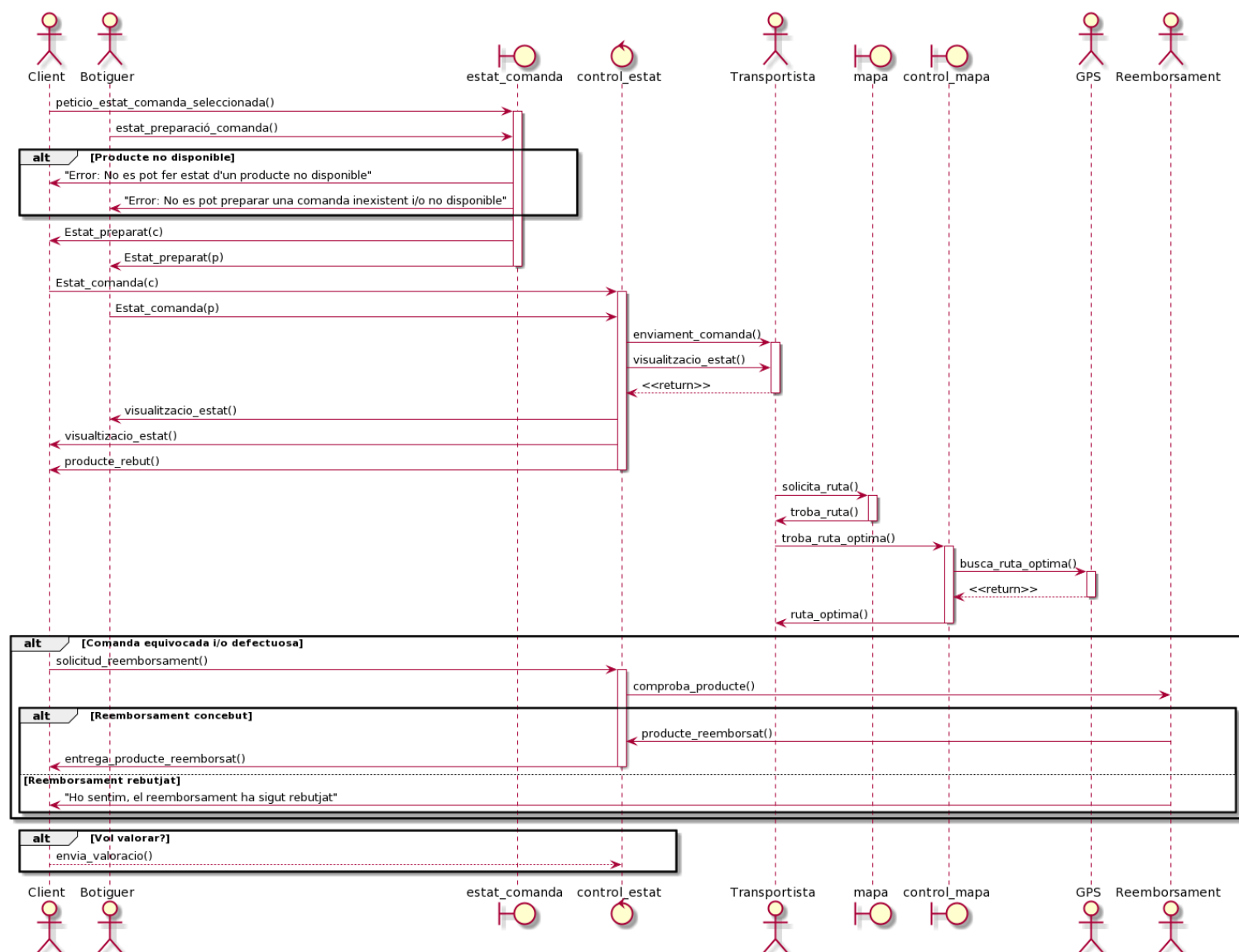
Una o més Comandes poden pertànyer a un sol Client; cap o vàries Comandes poden pertànyer a una mateixa entrega; i una Comanda està composta per una o més Línies de Comanda.

- 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.

Diagrames de seqüència

- sprint3_diag_sequencia_estat_comanda:



L'actor Client inicia l'acció `petició_estat_comanda_seleccionada()` i el Botiguer inicia `estat_preparacio_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

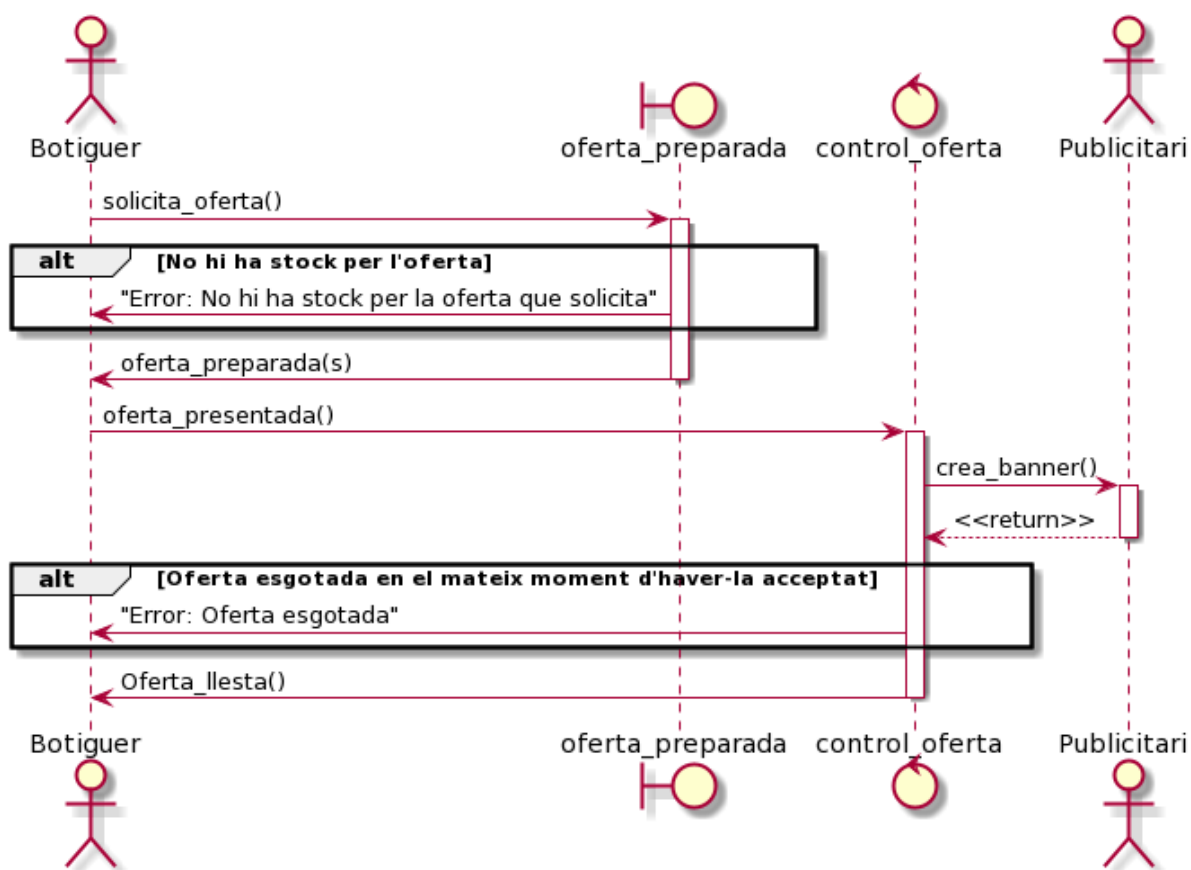
resultat d'aquestes operacions. Acte seguit, es presenta la `visualitzacio_estat()` tant al Botiguer com al Client, i a aquest últim se li entrega també `producte_rebut()`.

A continuació, el Transportista sol·licita `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_remborsat()`, ens trobem a [Reemborsament concebut] i el Client rep `entrega_producte_remborsat()`. Si no s'accepta el reemborsament, el Client rebrà [Reemborsament rebutjat].

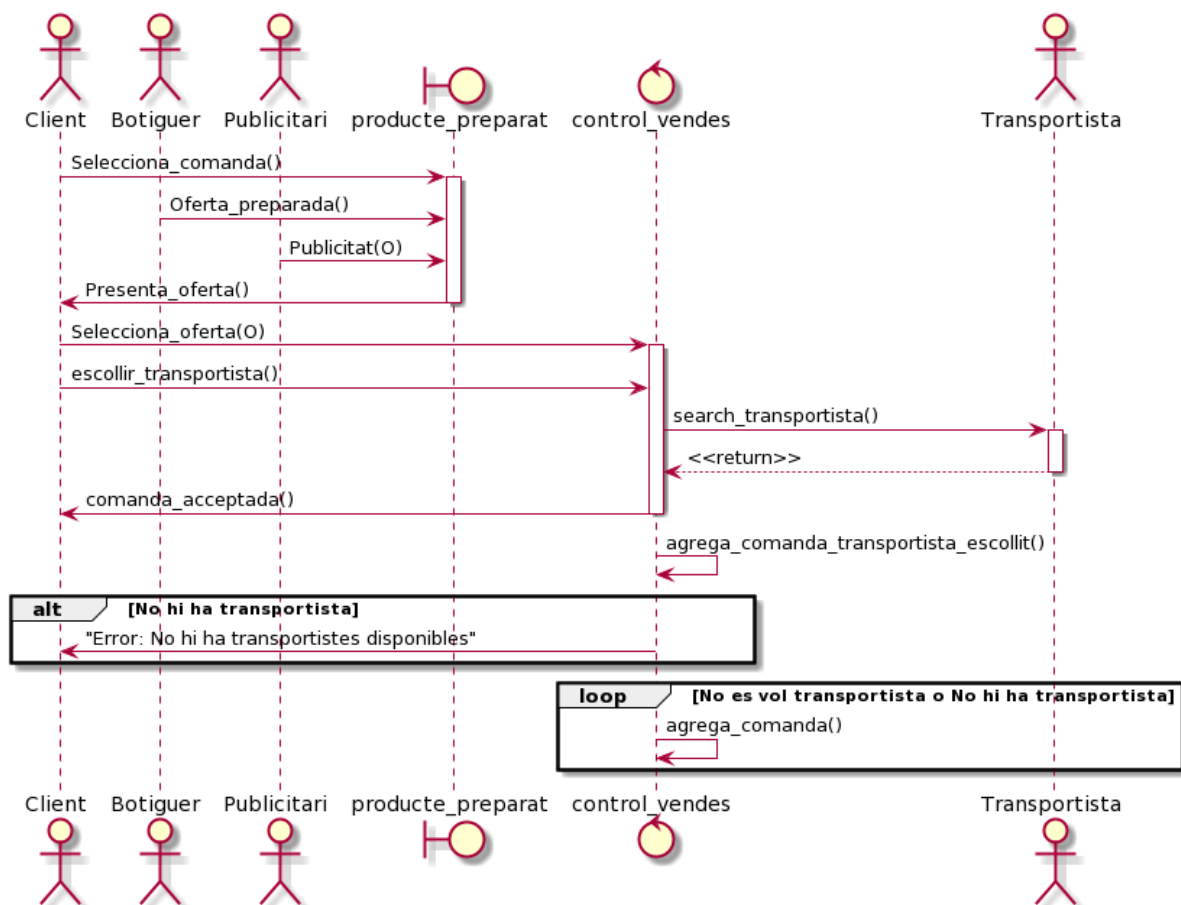
Finalment, el Client té l'opció de `envia_valoracio()` si [Vol valorar?], i sinó, es finalitza la seqüència.

- sprint3_diag_sequencia_preparacio_oferta:



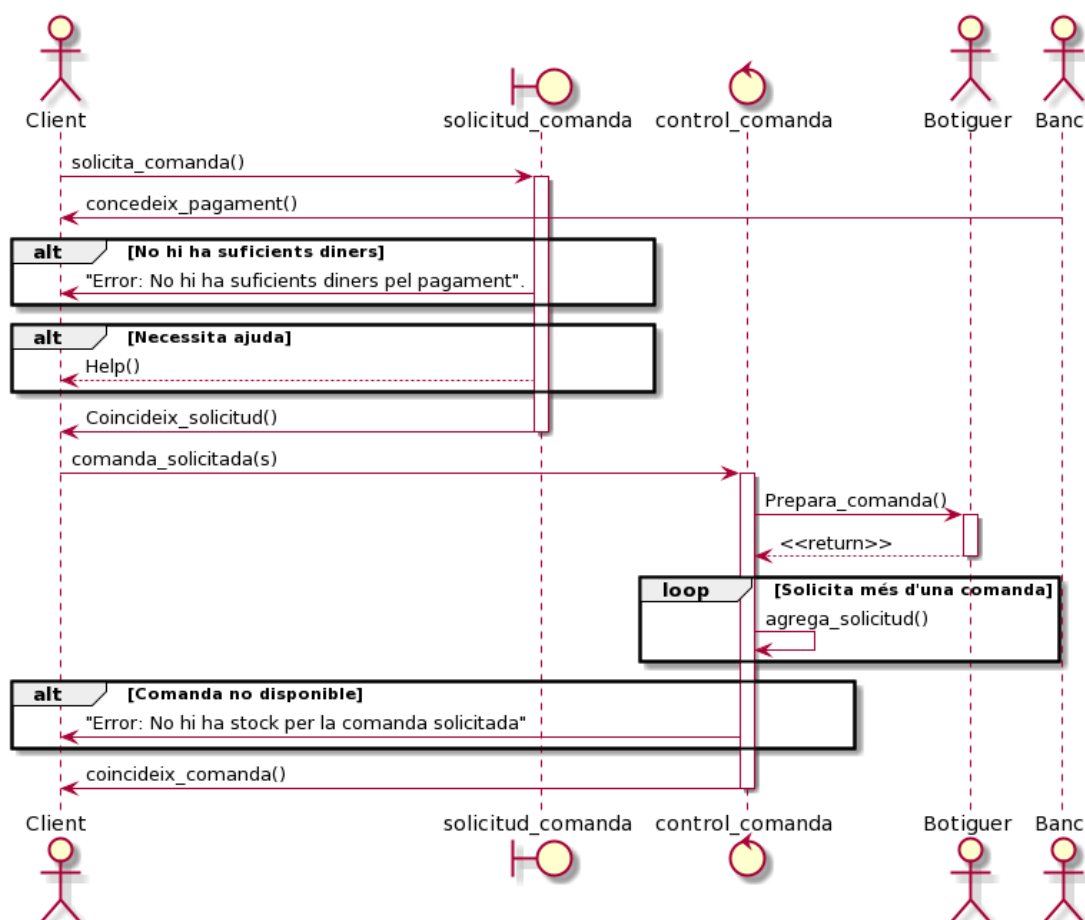
En aquest diagrama, identifiquem l'actor Botiguer com l'iniciador de l'acció `solicita_oferta()`, i idealment passa a l'estat d'`oferta_preparada`, en cas de no haver-hi stock suficient s'entraria en la condició especial `[No hi ha stock per l'oferta]` i retornaria un error. Quan la oferta es preparada, 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'aconsegueix 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_solicitud()` i ara li arribarà la comanda al Botiguer que s'encarregarà de prepararla amb `Prepara_comanda()` i entrarà en un loop dins que hagi preparat els diferents productes en `agrega_solicitud()` 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.