

## 1.-Realitza el diagrama de classes per aquest supòsit.

@startuml

```
class Client {  
    - nom: str // El nom del client  
    - adreca: str // L'adreça del client  
    - telefon: str // El número de telèfon del client  
    - email: str // El correu electrònic del client  
    + __init__(nom: str, adreca: str, telefon: str, email: str) // Constructor de la classe Client  
}
```

```
class UsuariWeb {  
    - login: str // El nom d'usuari per iniciar sessió  
    - password: str // La contrasenya per iniciar sessió  
    - estat: str // L'estat de l'usuari (per exemple: actiu, inactiu)  
    + __init__(login: str, password: str, estat: str) // Constructor de la classe UsuariWeb  
}
```

```
class CarroCompra {  
    - client: Client // El client que realitza la compra  
    - factura: Factura // La factura associada al carro de compra  
    - línies_comanda: list[LineaComanda] // Llista de línies de comanda d'aquest carro de compra  
    + __init__(client: Client, factura: Factura) // Constructor de la classe CarroCompra  
}
```

```
class Producte {  
    - codi_identificatiu: str // El codi identificatiu del producte  
    - nom: str // El nom del producte  
    - proveïdor: str // El proveïdor del producte  
    + __init__(codi_identificatiu: str, nom: str, proveïdor: str) // Constructor de la classe Producte  
}
```

```
class Factura {  
    - codi_factura: str // El codi de la factura  
    - adreca_facturacio: str // L'adreça de facturació  
    - data_emissio: str // La data d'emissió de la factura  
    - data_tancament: str // La data de tancament de la factura  
    - pagament: str // El mètode de pagament de la factura  
    + __init__(codi_factura: str, adreca_facturacio: str, data_emissio: str, data_tancament: str, pagament: str) // Constructor de la classe Factura  
}
```

```
class LiniaComanda {  
    - producte: Producte // El producte associat a la línia de comanda  
    - quantitat: int // La quantitat del producte  
}
```

```

- preu: float // El preu unitari del producte
+ __init__(producte: Producte, quantitat: int, preu: float) // Constructor de la classe
LiniaComanda
}

class OrdreEnviament {
- codi_identificatiu: str // El codi identificatiu de l'ordre d'enviament
- data_comanda: str // La data de la comanda
- data_enviament: str // La data d'enviament
- destinacio: str // La destinació de l'enviament
- import: float // L'import de l'enviament
- estat: str // L'estat de l'ordre d'enviament (per exemple: pendent, enviat)
+ __init__(codi_identificatiu: str, data_comanda: str, data_enviament: str, destinacio: str,
import: float, estat: str) // Constructor de la classe OrdreEnviament
}

Client "1" *-- "0..*" UsuariWeb // Relació 1 a molts entre Client i UsuariWeb
Client "1" *-- "*" CarroCompra // Relació 1 a molts entre Client i CarroCompra
CarroCompra "*" *-- "1" Factura // Relació molts a 1 entre CarroCompra i Factura
CarroCompra "1" *-- "*" LiniaComanda // Relació 1 a molts entre CarroCompra i
LiniaComanda
Factura "*" *-- "*" OrdreEnviament // Relació molts a molts entre Factura i OrdreEnviament

```

@enduml

## Resum

Un client pot tenir diversos usuaris web associats a ell. Això es representa amb la línia que va des del costat dret de la classe "Client" fins al costat esquerre de la classe "UsuariWeb" amb un multiplicador "0..\*".

Un client pot tenir diversos carrets de compra. Això es representa amb la línia que va des del costat dret de la classe "Client" fins al costat esquerre de la classe "CarroCompra" amb un multiplicador "\*".

Un carret de compra està associat a una factura. Això es representa amb la línia que va des del costat dret de la classe "CarroCompra" fins al costat esquerre de la classe "Factura" amb un multiplicador "1".

Un carret de compra pot tenir diverses línies de comanda. Això es representa amb la línia que va des del costat dret de la classe "CarroCompra" fins al costat esquerre de la classe "LiniaComanda" amb un multiplicador "\*".

Una factura pot estar associada a diverses ordres d'enviament. Això es representa amb la línia que va des del costat dret de la classe "Factura" fins al costat esquerre de la classe "OrdreEnviament" amb un multiplicador "\*".

## Resultat



## **2.-Converteix el diagrama a codi Python automàticament.**

**\*\*També podem fer enginyeria inversa passant codi python a plantUML, amb el chatgpt, o usant llibreries de python com pyreverse, que actualment forma part de la llibreria pylint <https://www.bhavaniravi.com/python/generate-uml-diagrams-from-python-code> (NO OBLIDEU CREAR UN ENTORN SI LA INSTAL·LEU)**

*\* De l'exercici anterior*

[Amb copilot \(ia de microsoft\)](#)

*\*(arxiu python)*