UML (Unified Modeling Language): Són un conjunt de notacions gràfiques que serveixen per especificar, dissenyar, elaborar i documentar models de sistemes i, en particular, d'aplicacions informàtiques.

Avantatges:

- Està recolzat per la OMG(Object Management Group) com a notació estàndard.
- Es basa en una notació gràfica, completada amb explicacions escrites.
- L'analista i/o al dissenyador els permet fer ús dels diagrames i amb el grau de detall que considerin oportuns.
- Permet tenir una visió global del sistema a implementar.
- Promou la reutilització

Inconvenients:

- UML és una notació.
- UML no es un llenguatge de programació.
- Pot resultar complex obtenir un coneixement complet del llenguatge.

Els diagrames que es poden fer s'engloben en:

- **Diagrames de visió estàtica:** Descriuen aspectes del sistema que són estructurals i, per tant, permanents. Podem trobar:
  - **Diagrama de classes**
  - **Diagrama de paquets**
  - **Diagrama d'objectes**
  - **Diagrama d'estructures compostes**
  - **Diagrama de components**
  - **Diagrama de desplegament**
  - **Diagrama de perfil**

- **Diagrames de visió dinàmica:** Representen allò que pot fer el sistema modelitzat. Podem trobar:
  - **Diagrama de casos d'ús**
  - **Diagrama d'estats**
  - **Diagrama d'activitats**
  - **Diagrames d'interacció:**
    - **Diagrama de seqüències**
    - **Diagrama de comunicacions**
    - **Diagrama de visió general de la interacció**
    - **Diagrama temporal**

Utilitzem l'UML per representar aquests diagrames, alguns objectes que trobem són:

- **Model:** L'UML serveix per a fer documentació descriptiva formalitzada. Aquests models es traduiran als programes.

- **Element:** Són els diferents tipus d'icones que representen idees dins del diagrama.

- **Classificador:** Es un objecte que te valors en comú. Es l'objecte base amb el qual instanciarem els demés objectes (Un objecte neix d'una classe, Carlos neix de Persona).

    - **Comportament:** Senyala tot allò que un classificador pot fer.

    - **Propietats:** El tipus de valors que pot tenir un classificador (nom=”Carlos”).

    - **Generalització, especialització i herència dels classificadors:** Les propietats poden passar d'un objecte.

        - **Generalització:** l'objecte te menys especificacions que abans.

        - **Especialització:** l'objecte adquireix diferents especificacions.

        - **Herència:** Un objecte obté les diferents especificacions d'un altre.

- **Estereotip:** Es una especialització d'un element a partir d'un altre que anomenarem element base.

Diagrama de classes: representa les classes que seran utilitzades dins el sistema i les relacions que existeixen entre elles. Un diagrama de classes te vinculats els següents conceptes:

- **Classe, atribut i mètode(operacions):**

    - **Classe:** Una classe descriu un conjunt d'objectes que comparteixen els mateixos atributs, que representen característiques estables de les classes, i les operacions, que representen les accions de les classes.

    - **Atribut:** Són les dades detallades que contenen els objectes.

    - **Mètodes:** Implementen les accions que es podran dur a terme sobre els atributs.

- **Visibilitat:** Defineix l'àmbit des del qual poden ser utilitzats aquests elements.

    - **Public (+):** L'element és accessible per tots els altres elements del sistema.

- **Private (-):** L'element només es accessible pels elements continguts dins el mateix objecte.

- **Protected (#):** L'element només és visible per als elements del seu mateix objecte i per als elements que pertanyen a objectes que són especialitzacions.

- **Package (~):** Només es pot aplicar quan l'objecte no és un paquet, per tant l'element només es visible per als elements continguts directa o indirectament dins el paquet que conté l'objecte.

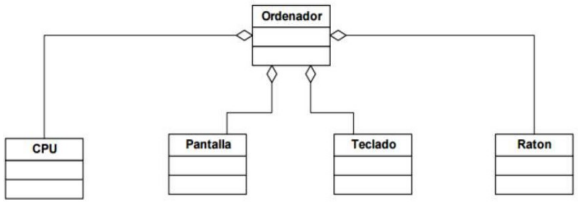
- **Objecte:** Es una unitat de memòria relacionada que, en temps d'execució, du a terme accions dintre un programari.

- **Relacions:** S'entén per relació que un objecte 1 demani a un objecte 2, mitjançant un missatge, que executi una operació de les definides en la classe de l'objecte 2.

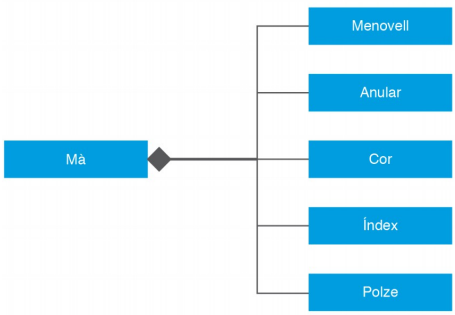
  - **Multiplicitat:** Indica el nombre màxim d'enllaços donats en una relació. S'utilitza un sol número (1) per indicar la quantitat de relacions, (*) per indicar que la relació es a molts, (12..18) per indicar que la relació es de 12 a 18 y (12,24) per indicar que es de 12 o 24.

  - **Relació d'associació:** Es la relació base entre dos objectes, es representa amb una línia contínua sense fletxes ni cap altre símbol als extrems.

  - **Relació d'agregació:** Es la relació entre dos o més objectes que l'agregació d'aquest dona lloc a l'objecte però si desapareix un d'ells els altres objectes poden continuar existint, es representa amb un rombe buit (ordinador es crea a partir d'agregar diversos components).

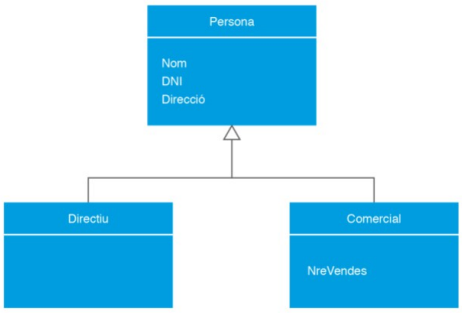


  - **Relació de composició:** es el mateix que la d'agregació amb la diferencia que aquesta relació depèn que tots els elements existeixin. Es representa amb un rombe negre.

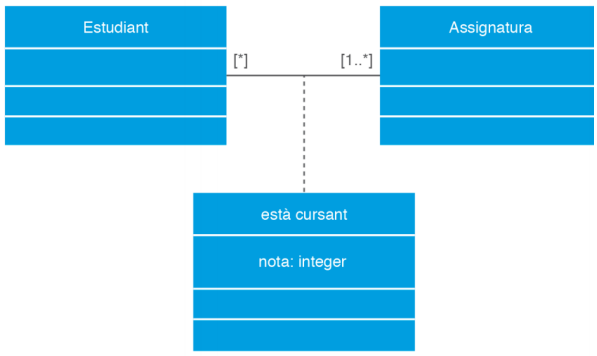


- **Relació de dependència:** Un objecte depèn de l'altre (si un canvia l'altre també), es representa amb una fletxa discontínua (Pel·lícula depèn d'actor).

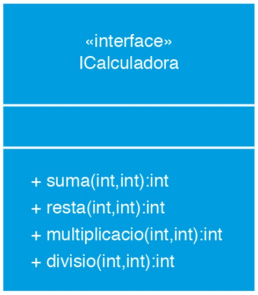
- **Relació d'herència:** Una classe hereta els mètodes d'una altre més general, es representa amb un triangle que surt de la classe pare.



- **Classe associativa:** Es troben quan una associació té propietats o mètodes propis i es representa com una classe unida a la línia de l'associació per mitjà d'una línia discontínua (Tant la línia com la classe associativa representen la mateixa associació).



- **Interfícies:** Es la declaració de les operacions sense la seva implementació que hauran de ser implementades per una classe o component.



Exemple de classe:

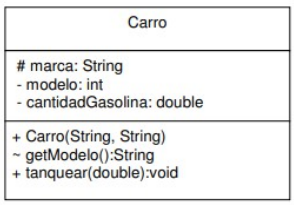
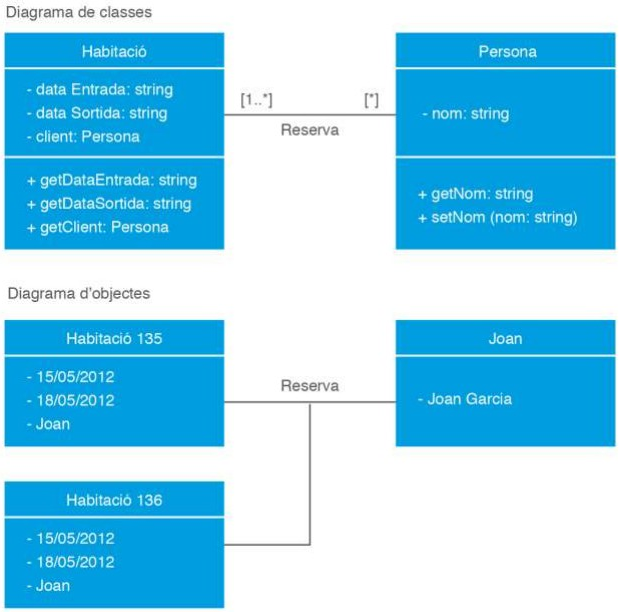


Diagrama d'objectes: Només pot contenir instancies i relacions entre objectes.



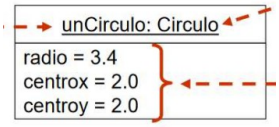
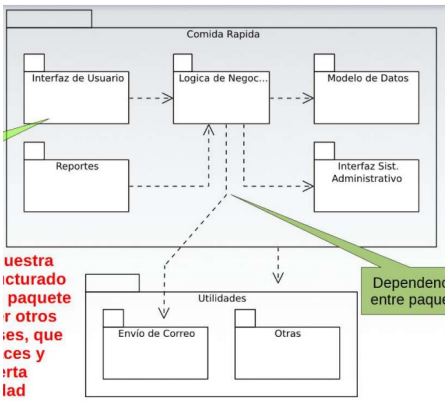


Diagrama de paquets: Serveix per descriure l'estructura d'un model en termes de paquets interrelacionats.



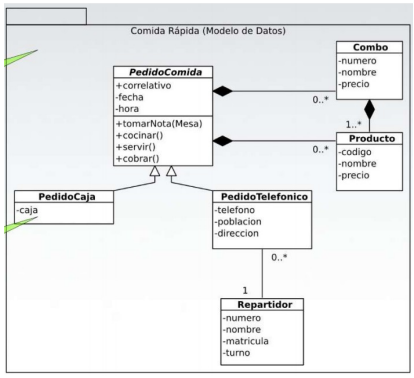


Diagrama d'estructures compostes: És un conjunt d'elements interconnectats que col·laboren en temps d'execució per aconseguir algun propòsit.

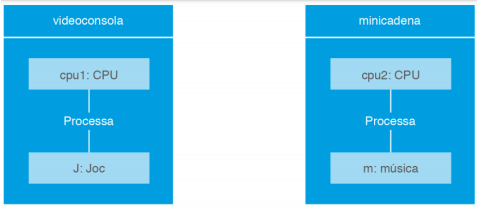


Diagrama de components: mostra els components que conformen el sistema i com es relacionen entre si.

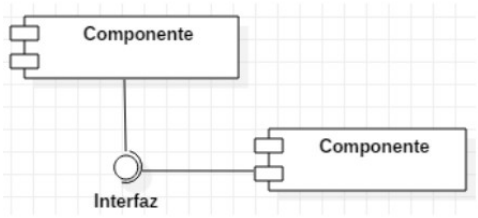


Diagrama de desplegament: Descriu la distribució de les parts d'una aplicació i les seves interrelacions, tot en temps d'execució.