Lucas Pérez González

Activitat 1 - Diagrama de Classes

Volem crear un projecte anomenat **Associacio** en què es dissenyi un diagrama de classes que modeli el procés de donar d'alta a cada una de les persones que s'apunten a una associació.

De cada **persona** ens interessa saber les seves dades bàsiques: **NIF**, **nom complert**, i la **data de naixement**.

Quan cada **soci** es dóna d'alta, se li assigna un **codi d'associat** alfanumèric i s'anota la **data d'alta**.

La classe **Data** es modela amb tres camps (**dia, mes** i **any**) de tipus enter i hauria de comprovar que es tracta d'una data vàlida.

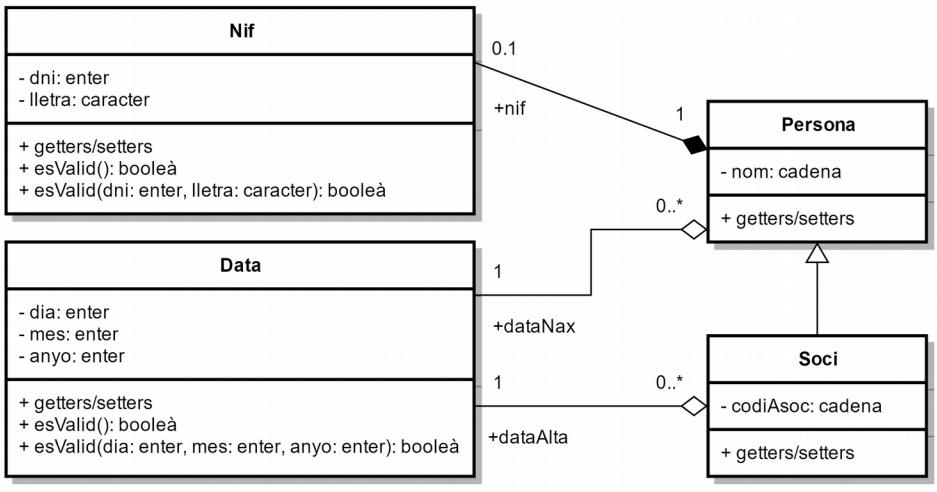
La classe **NIF** es modela amb un camp de tipus enter, anomenat **dni**, i un camp de tipus caràcter anomenat **lletra**. La classe hauria de permetre comprovar que el NIF és vàlid.

Classes:

En el **Nif** hi ha un dni i una lletra, i validarem si està bé.
La classe **Data** tindrà un dia, un mes i un any, i els mètodes de validació per comprovar que es una data correcta.
Persona només li indiquem un nom, per que el nif i la data de naixement la agafara després amb la relació a Nif i Data.
Soci només te un codi de associat, la data en que s'ha donat d'alta, la agafara després amb la relació a Data.

Relacions:

Nif te una relació forta amb persona, per que una persona te un nif si o si, sinó te, no el considerem que sigui una persona en el nostre programa.
Data te una relació a persona i soci, per indicar la data de naixement de la persona i la data d'alta del soci, aquestes relacions son febles per que aquella mateixa data pot estar en varis socis i persones.
Soci hereta de Persona, per que un soci es una persona. Soci hereta el nom, nif i data de naixement



Lucas Pérez González

Activitat 2

Realitza un diagrama de classe per representar les relacions entre empleats i departaments:

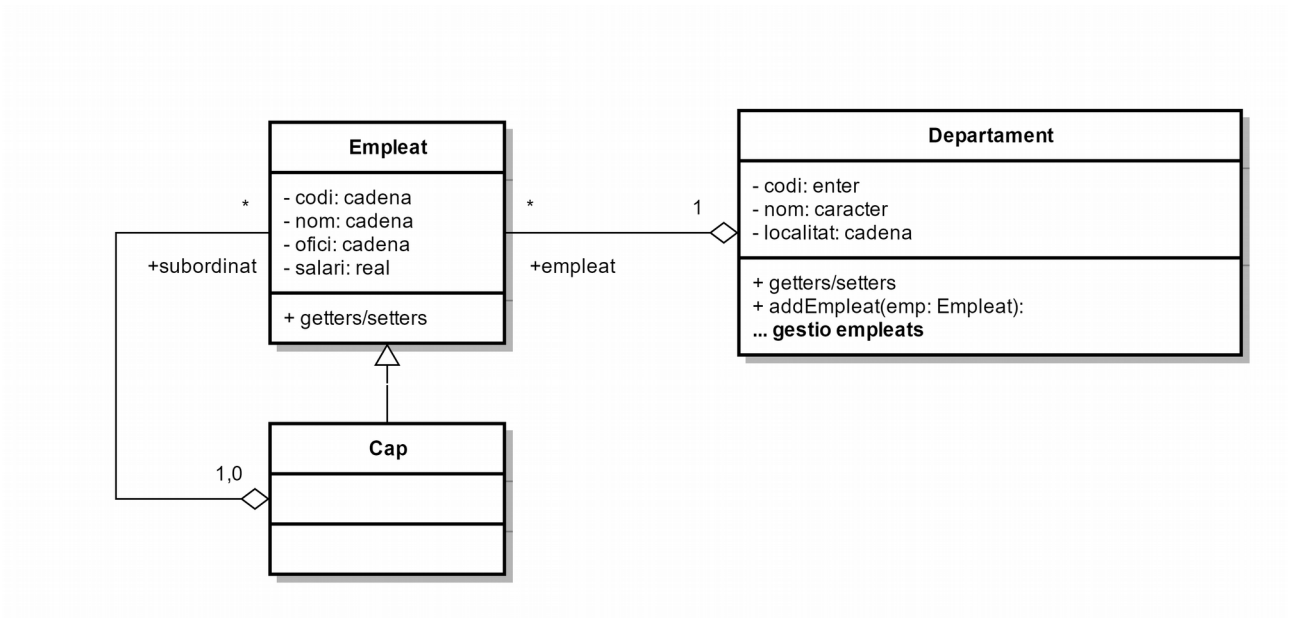
- Considerem que un empleat treballa en un departament i en el departament treballen molts empleats.
- Dades dels empleats són codi, nom, ofici i salari
- Dades del departament són codi, nom i localitat
- A més un empleat pot ser cap de diversos empleats
- Es necessita crear els mètodes per assignar dades als empleats i departaments i tornar-los (getters i setters)

Classes:

Departament te un codi, un nom, una localitat, i una llista de empleats que agafara després amb la relació. I te uns mètodes de gestió dels empelats.
Empleat te un codi, un nom, un ofici i un salari.
Cap te una llista de subordinats i tot el que hereti de **Empleat.**

Relacions:

Empleat és relaciona amb **Departament**, per que un departament te varis empleats, però un empleat només un departament. És feble per que sense els empleats segueix estan el departament, i sense departament, els empleats segueixen sent empleats.
Empleat te una relació també amb cap, per que un cap de te varis empleats, i els empleats tenen un o ningun cap. I és feble per que sense cap segueixen sent empleats, i sense subordinats, segueix sent un cap.
Cap hereta de **Empleat** per que un cap es un empleat.



Lucas Pérez González

Activitat 3 – Persones i Empresa

Representa mitjançant un diagrama de classes la següent especificació:

- Una aplicació necessita emmagatzemar informació sobre empreses, els seus empleats i els seus clients.
- Tots dos es caracteritzen pel seu nom i edat.
- Els empleats tenen un sou brut, i caldrà calcular el net.
- Els empleats que són directius tenen una categoria, així com un conjunt de empleats subordinats.
- Dels clients més es necessita conèixer el seu telèfon de contacte.
- L'aplicació necessita mostrar les dades de treballadors i clients.

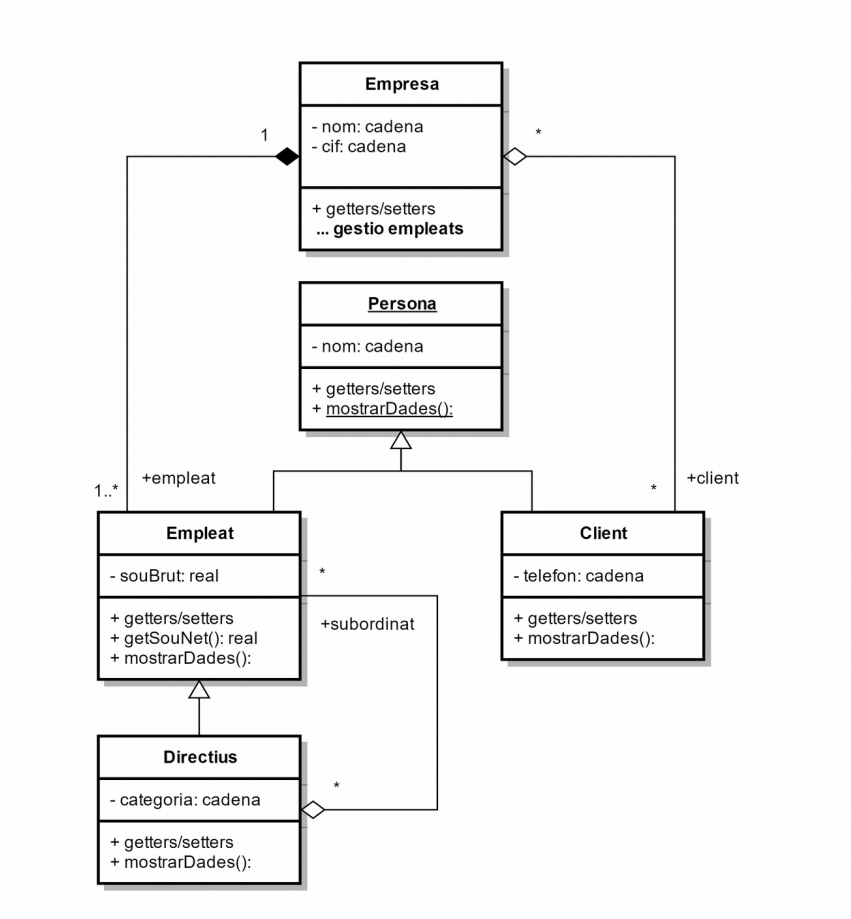
Classes:

Empleat te un codi, un nom i un cif.
Persona és una classe abstracta, de la qual hereten persona i client, te un atribut que es nom, i un metode abstracta que es per mostrar totes les dades.
Client te un telefon i un nom que hereta de **Persona.**
Empleat te un sou brut i un nom que hereta de **Persona.** També te un mètode que és per aconseguir el sou net.
Directius te una categoria i subordinats, més tot el que hereti de **Empleat.**

Relacions:

Directius hereta de **Empleat**, per que un directiu és un empleat.
Empleat te una relació amb **Directius**, per que els empleats te un o més directius, i un directiu te cap o varis empleats.
Empleat te una relació forta amb **Empresa**, per que els empleats només considerem que poden estar en una empresa, i si no estan en una empresa, ja no son empleats, si en cas, son aturats. I la empresa sense empleats no pot existir, per que al menys te de tindre un empleat que seria el que la creat.
Empleat i **Client** hereten de **Persona**, i agafen el mètode abstracta de mostrarDades() i la modifiquen al seu gust.
Client te una relació feble amb **Empresa**. Una empresa te uns client, i son client per que compren en aquella empresa. Els clients poden ser clients de varies empreses, i la empresa pot tindre cap o varis clients.

Lucas Pérez González



Lucas Pérez González

Activitat 4 – Torneig de TT

Crea un projecte UML anomenat **Torneig** en què es dissenyi un diagrama de classes que modeli l'estructura necessària per manegar les dades dels **matxs** d'un **torneig de tennis de taula** en la **modalitat de sorteig i eliminatòria**.

•Del torneig interessa conèixer la **data del torneig**, els **matxs celebrats** i el **guanyador**. De cada **jugador**, que **ha de conèixer perfectament les regles**, interessa saber el **número de federat de la federació de què és membre**.

•De cada **persona** interessa saber les **seves dades bàsiques**: **NIF, nom complert** i **data de naixement**. La classe **Data** es modela amb tres camps (**dia, mes** i **any)** de tipus enter. La classe **NIF** es modela amb un camp de tipus enter anomenat **dni** i un camp de tipus caràcter anomenat **lletra**.

•De cada **matx** interessa conèixer els **oponents**, el **guanyador** i el **resultat final** del marcador de cadascuna de les **tres partides** que es juguen a **21 punts**.

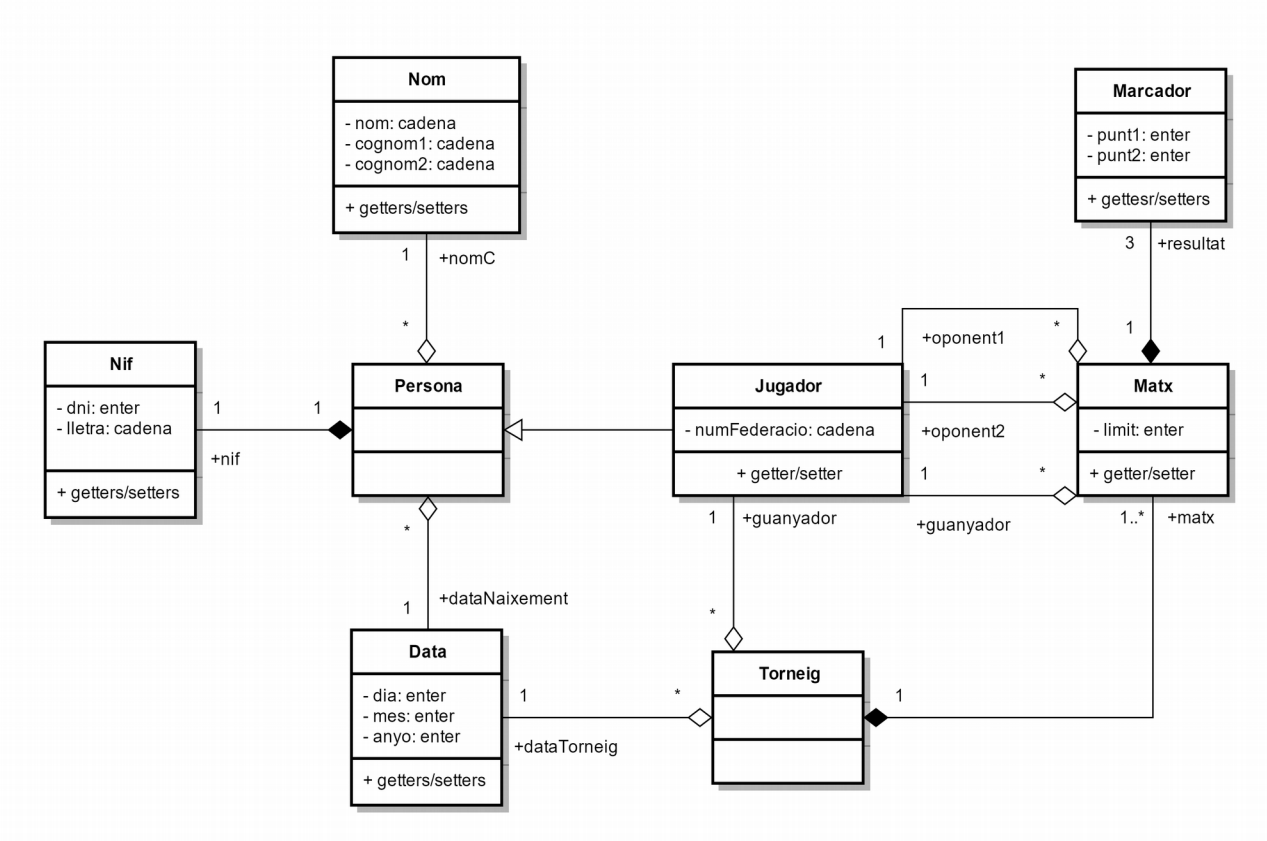
Classes:

Nif te un dni i una lletra.
Nom te un nom, dos cognoms,
Data te un dia, un mes, i un any.
Persona no te res en un principi, però amb les relacions acaba tenint un nif, un nom, i una data de naixement.
Jugador te un numero de federacio i tot el que tingui la classe **Persona.**
Marcador te els punts de cada un dels oponents.
Matx te un limit, que en aquest cas es en 21. també amb les relacions acaba tenint dos oponents, un guanyador i un resultat.
Torneig no te res en un principi, però amb les relacions acaba tenint una data de torneig, un guanyador, i un matx.

Relacions:

Persona te varies relacions, una es a **Nom** on agafa el nom, un altra es a **Data** on agafa la data de naixement, i una altra es a **Nif** on agafa el nif. Les relacions de **Nom** i **Data** son febles perquè aquella data o aquell nom pot estar en més de una persona. Però la relació amb el **Nif** si és forta, per que una persona te un nif i un nif només pot estar en una person.
Torneig te varies relacion cap a ella, una es amb el **Matx** on agafa el matx, una altra es amb el **Jugador** on agafa el guanyador del torneig, i una altra amb la **Data** on agafa la data la qual es va celebrar el torneig. Les relacions de **Jugador** i **Data** son febles per que poden estar en mes d'un torneig diferent, i segueixen estan aun que no s'aguï celebrat el torneig. I la relació amb el **Matx** es forta, per que aquell matx només està en un torneig, no pot estar en més d'un.
Marcador te una relació forta cap a **Matx.** És forta per que aquell marcador només pot estar en aquell matx, no reutilitzaríem un altre marcador per varis matx, i el matx necessita 3 marcadors.
Jugador te tres relacions fins a **Matx** per que el matx necessita els dos oponent i el guanyador. I és feble per que aquell jugador pot haver guanyat o jugat en varis Matx.

Lucas Pérez González



Lucas Pérez González

Exemple 1 – Llibre

- Dibuixa un diagrama de classe que represente un llibre definit per la següent frase: "Un llibre està compost per un nombre de parts, que a la seva vegada està compost per un nombre de capítols. Els capítols estan compostos per seccions" Centra't només en la relació entre classes.
- Amplia el diagrama anterior de manera que inclogui els següents atributs:
  - Un **llibre** té un editor, una data de publicació i un ISBN.
  - Una **part** inclou un títol i un nombre.
  - Un **capítol** inclou un títol, un nombre i un resum.
  - Una **secció** inclou un títol i un nombre.
- Donat el diagrama anterior, podem veure que Part, Capítol i Secció inclouen un títol, un nombre com a atributs. Afegeix una classe abstracta i una relació de generalització per tal de factoritzar aquests atributs en la classe abstracta.

Classes:
Distribució te un títol i un nombre,
Secció te el que hereti de **Distribució**.
Capítol te un resum, les seccions que te, i el que hereti de **Distribució**.
Part te uns capítols, i el que hereti de **Distribució**.
Llibre te un editor, una data de publicació, un isbn i les parts que la componen.

Relacions:
Part, Capítol i Secció hereten de Distribució.
Secció te una relació amb **Capítol**, i és feble per que les seccions segueixen estan si no està el capítol.
Capítol te una relació amb **Part**, i és feble per que els capitols segueixen estan si no està la part.
Part te una relació amb **Llibre**, i és feble per que les parts segueixen estan si no està el llibre.