

Faculdade de Engenharia da Universidade do Porto

Base de Dados

2ºANO - MIEIC

Estudantes & Autores

Diogo Silva, up201706892

up201706892@fe.up.pt

Ana Mafalda Santos, up201706791

up201706791@fe.up.pt

João Luz, up201703782

up201703782@fe.up.pt

14 de abril de 2019

2MIEIC07_06

ÍNDICE

A. DEFINIÇÃO DO MODELO CONCEPTUAL	3
CONTEXTO	3
DIAGRAMA UML	4
RESTRIÇÕES.....	5
B. DEFINIÇÃO DO ESQUEMA RELACIONAL	6
C. ANÁLISE DE DEPENDÊNCIAS FUNCIONAIS E FORMAS NORMAIS.....	7
E. ADIÇÃO DE RESTRIÇÕES À BASE DE DADOS	11

A. Definição do Modelo Conceptual

Contexto

Ao longo do desenvolvimento deste projeto pretende-se construir uma base de dados que se revele útil na gestão de associações responsáveis pela adoção de animais. Assim, deve ser tomada em conta, não só a quantidade e diversidade de dados a ser armazenados, como também a forma como estes se relacionam entre si, representando melhor os intervenientes no bom funcionamento da associação e a sua importância para a mesma.

Dentro dos intervenientes, podem destacar-se a associação, as pessoas e os animais.

Uma associação requer um **nome**, **morada**, **telefone**, **localidade**, **NIF**, **data de fundação** e pode ser responsável por um ou mais **abrigos**.

Todas as **pessoas** envolvidas, caracterizadas por **CC**, **nome**, **morada** e **contacto**, podem ser voluntários, sócios ou adotantes.

Cada **voluntário** ajuda um ou mais abrigos num determinado **número de horas semanais** responsabilizando-se por uma ou mais **áreas**.

Para cada **sócio**, interessa registar: **ID**, **profissão**, **NIF**, **veterinário habitual**, **data de adesão** e **data do último pagamento da quota anual**.

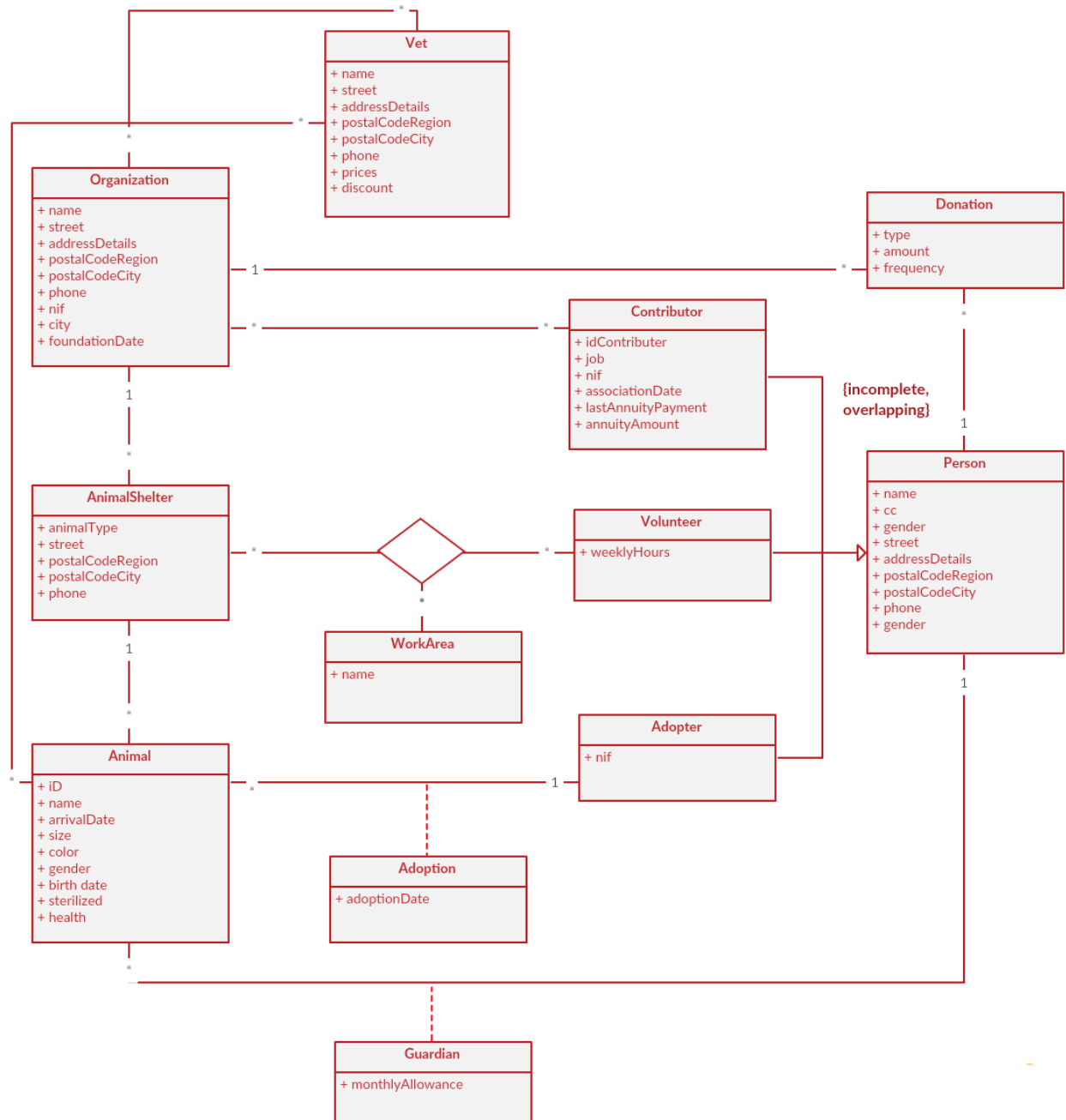
Para os **adotantes** é necessário registar a **data de adoção** e o **animal adotado**.

Todos podem **apadrinhar** animais (um ou mais), ficando responsáveis pelo pagamento de um **abono mensal** e/ou fazer **doações** de um **tipo** com determinada **frequência** e **quantidade**.

Cada **animal** – cão ou gato – tem um **ID**, **nome**, **porte**, **idade**, **data de chegada**, **sexo**, **cor**, se está **esterilizado** e respetivos **detalhes de saúde** (doença ou condição especial).

A associação tem ainda acordos com alguns **veterinários**, interessando, portanto, conhecer o **nome**, **morada**, **contacto**, **preço por consulta** e o respetivo **desconto para sócios e para a associação**.

Diagrama UML



Restrições

- **NIF, Phone:** 9 digits
- **CC:** 12 digits (8 digits + 4 alphanumerics)
- **Prices, Amount, Frequency, Weekly hours, Monthly allowance, Annuity Amount** > 0
- **Discount** >= 0
- **Birthdate** > **Arrival date** > **Adoption date**
- **Last annuity payment** > **Foundation date**
- **Size:** {small, medium, large}
- **Sterilized:** {1, 0}
- **Gender:** {female, male}
- **Animal type:** {dog, cat}
- **Donation type:** {money, food, toys, accessories, other}
- **Discount:** exclusive to the organization and its contributors.

B. Definição do Esquema Relacional

Na definição das generalizações do nosso esquema relacional optamos pela estratégia de conversão **E/R style**, por ser eficaz em generalizações com grande número de subclasses.

Organization (idOrganization, name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, nif, foundationDate)

Vet (idVet, name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, prices, discounts)

OrganizationVet (idOrganization → Organization, idVet → Vet)

Person (idPerson, name, cc, gender, street, addressDetails, postalCodeRegion, postalCodeCity, phone, gender, birthday)

Contributor (idPerson → Person, idContributor, job, nif, associationDate, lastAnnuityPayment)

Volunteer (idPerson → Person, weeklyHours)

WorkArea (idWorkArea, area)

Adopter (idPerson → Person, nif)

Adoption (idPerson → Adopter, idAnimal → Animal, adoptionDate)

OrganizationContributor (idOrganization → Organization, idPerson → Contributor)

Donation (idDonation, type, amount, frequency, idOrganization → Organization, idPerson → Person)

AnimalShelter (idAnimalShelter, animalType, street, addressDetails, postalCodeRegion, postalCodeCity, phone, idOrganization → Organization)

AnimalShelterVolunteerWorkArea (idAnimalShelter → AnimalShelter, idPerson → Volunteer, idWorkArea → WorkArea)

Animal (idAnimal, name, arrivalDate, size, color, gender, birthDate, sterilized, health, idAnimalShelter → AnimalShelter)

AnimalVet (idAnimal → Animal, idVet → Vet)

Guardian (idAnimal → Animal, idPerson → Person, monthlyAllowance)

C. Análise de Dependências Funcionais e Formas Normais

Organization (idOrganization, name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, nif, foundationDate)

idOrganization → name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, nif, foundationDate

nif → idOrganization, name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, foundationDate

phone → idOrganization, name, street, addressDetails, postalCodeRegion, postalCodeCity, foundationDate

name, street, addressDetails, postalCodeRegion, postalCodeCity → idOrganization, phone, nif, foundationDate

postalCodeRegion, postalCodeCity → street

- > Existe violação da **Forma Normal de Boyce-Codd** (BCNF) uma vez que na última dependência funcional (DF), do lado esquerdo não se encontra uma **superkey**. No entanto, não viola a **3ª Forma Normal** já que, também na última DF, à direita está apenas um **prime attribute**.

Vet (idVet, name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, prices, discounts)

idVet → name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, prices, discounts

name, street, addressDetails, postalCodeRegion, postalCodeCity → idVet, phone, prices, discounts

phone → idVet, name, address, street, addressDetails, postalCodeRegion, postalCodeCity, discounts

postalCodeRegion, postalCodeCity → street

- > Tal como na anterior, existe violação da **Forma Normal de Boyce-Codd** (BCNF) uma vez que na última dependência funcional (DF), do lado esquerdo não se encontra uma **superkey**. No entanto, não viola a **3ª Forma Normal** já que, também na última DF, à direita está apenas um **prime attribute**.

Person (idPerson, name, cc, gender, street, addressDetails, postalCodeRegion, postalCodeCity, phone, birthday)

idPerson → name, cc, street, addressDetails, postalCodeRegion, postalCodeCity, phone, gender, birthday

cc → idPerson, name, street, addressDetails, postalCodeRegion, postalCodeCity, phone, gender, birthday

phone → idPerson, name, cc, street, addressDetails, postalCodeRegion, postalCodeCity, gender, birthday

postalCodeRegion, postalCodeCity → street

- > Mais uma vez, há violação da **Forma Normal de Boyce-Codd** (BCNF) uma vez que na última dependência funcional (DF), do lado esquerdo não se encontra uma **superkey**. No entanto, não viola a **3ª Forma Normal** já que, também na última DF, à direita está apenas um **prime attribute**.

Contributor (idPerson → Person, idContributor, job, nif, associationDate, lastAnnuityPayment, annuityAmount)

idPerson → idContributor, job, nif, associationDate, lastAnnuityPayment, annuityAmount

idContributor → idPerson, job, nif, associationDate, lastAnnuityPayment, annuityAmount

nif → idPerson, idContributor, job, associationDate, lastAnnuityPayment, annuityAmount

Volunteer (idPerson → Person, weeklyHours)

idPerson → weeklyHours

WorkArea (idWorkArea → WorkArea, name)

idWorkArea → name

name → idWorkArea

Adopter (idPerson → Person, nif)

idPerson → nif

nif → idPerson

Adoption (idPerson → Adopter, idAnimal → Animal, adoptionDate)

idAnimal → idPerson, adoptionDate

Donation (idDonation, type, amount, frequency, idOrganization → Organization, idPerson → Person)

idDonation → type, amount, frequency, idOrganization, idPerson

AnimalShelter (idAnimalShelter, animalType, street, addressDetails, postalCodeRegion, postalCodeCity, phone, idOrganization → Organization)

idAnimalShelter → animalType, street, addressDetails, postalCodeRegion, postalCodeCity, phone, idOrganization

phone → idAnimalShelter, animalType, street, addressDetails, postalCodeRegion, postalCodeCity, idOrganization

idOrganization, street, addressDetails, postalCodeRegion, postalCodeCity → idAnimalShelter, animalType, phone

postalCodeRegion, postalCodeCity → street

- > Existe violação da **Forma Normal de Boyce-Codd** (BCNF) uma vez que na última dependência funcional (DF), do lado esquerdo não se encontra uma **superkey**. No entanto, não viola a **3ª Forma Normal** já que, também na última DF, à direita está apenas um **prime attribute**.

Animal (idAnimal, name, arrivalDate, size, color, gender, bithDate, sterilized, health, idAnimalShelter → AnimalShelter)

idAnimal → name, arrivalDate, size, color, gender, bithDate, sterilized, health, idAnimalShelter

Guardian (idAnimal → Animal, idPerson → Person, monthlyAllowance)

idAnimal → idPerson, monthlyAllowance

Em todas as outras relações em que não foram indicadas violações das formas normais, isto não acontece porque em todas as suas DFs, do lado esquerdo, encontra-se sempre uma **superkey**, o que é suficiente para satisfazer tanto a forma BCNF como a 3ª.

E. Adição de restrições à base de dados

- **NIF, Phone:** 9 digits
 - > restrição CHECK
- **CC:** 12 digits (8 digits + 4 alphanumerics)
 - > restrição CHECK
- **Postal Code Region:** 4 digits
 - > restrição CHECK
- **Postal Code City:** 3 digits
 - > restrição CHECK
- **Prices, Amount, Frequency, Weekly hours, Monthly allowance, Annuity**
 - Amount** > 0
 - > restrição CHECK
- **Discount** >= 0
 - > restrição CHECK
- **Birthdate** > **Arrival date** > **Adoption date**
 - > restrição CHECK
- **Last annuity payment** > **Foundation date**
 - > restrição CHECK
- **Size:** {small, medium, large}
 - > restrição CHECK
- **Sterilized:** {0, 1} (valores booleanos que representam verdadeiro e falso, respetivamente)
 - > restrição CHECK
- **Gender:** {female, male}
 - > restrição CHECK
- **Animal type:** {dog, cat}
 - > restrição CHECK
- **Donation type:** {money, food, toys, accessories, other}
 - > restrição CHECK
- **Phone, NIF, CC:** não podem ser repetidos ou nulos
 - > restrição NOT NULL, chave (UNIQUE)

- **Birthday, Job, Frequency, BirthDate, Color, Health:** são os únicos atributos que podem ser NULL, todos os outros têm restrição NOT NULL.

Todas as ***Primary Keys*** têm restrição chave (PRIMARY KEY), tal como as chaves estrangeiras têm restrição de integridade referencial.