



Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de trabalho por minha honra que este diagrama foi realizada apenas pelos elementos que constituem o grupo de trabalho de projeto. Francisca Mendonça 109264

ANA SANTOS 109260

Responsibility Not Assigned Exception

Animal already exists Exception

Employee already exists Exception

Habitat already exists Exception

Tree already exists Exception

Vaccine already exists Exception

Animal Not Found Exception

Employee Not Found Exception

Habitat Not Found Exception

Species Not Found Exception

Exception

Tree Not Found Exception

Vaccine Not Found Exception

Veterinarian Not Found Exception

Unauthorized Veterinarian Exception

File Cannot Be Opened Exception

Import File Exception

Missing File Association Exception

Unreachable File Exception

Unrecognized Entry Exception

## HotelManager

```
- - hotel : Hotel  
+ save() : void  
+ saveAs(filename : String) : void  
+ load(filename : String) : void  
+ importFile(filename : String) : void
```

## Hotel

```
- - habitats : Habitats  
- - employee : List < Employee >  
- - vaccines : Vaccines  
  
+ showAllAnimals() : void  
+ showAllTrees() : void  
+ calculateGlobalSatisfaction() : double  
+ showSatisfactionOfAnimal(id : String) : double  
+ registerEmployee(idEmployee : String, name : String, type : String) : void  
+ addResponsibility(idEmployee : String, idNewResponsibility : String) : void  
+ removeResponsibility(idEmployee : String, idResponsibilityToRemove : String) : void  
+ showAllEmployees() : void  
+ showSatisfactionOfEmployee(idEmployee : String) : void  
+ showAllVaccines() : void
```

## Habitats

- - habitat: List <Habitat>  
- - species: List <Species>  
- - animal: List <Animals>

+ registerHabitat (idHabitat: String, name: String, area: int): void  
+ changeHabitatArea (idHabitat: String, area: int): void  
+ changeHabitatInfluence (idHabitat: String, idSpecies: String, influence: String): void  
+ addTreeToHabitat (idHabitat: String, idNewTree: String, name: String, age: String, cleaningDifficulty: int, type: String): void  
+ registerAnimal (id: String, name: String, idSpecies: String, idHabitat: String): void  
+ transferToHabitat (id: String, idDestHabitat: String): void

## Habitat

- - idHabitat: String  
- - name: String  
- - area: int  
- - numberoftrees: int  
- - animal: List <Animal>  
- - trees: List <Trees>

## Animal

- - idAnimal: String  
- - idSpecies: String  
- - animalName: String  
- - healthHistorical: String  
- - idHabitat: String  
- - satisfaction: SatisfactionAnimal  
- - vaccinations: List <MedicalActs>

+ calculateSatisfactionAnimal(): double  
+ updateHealthHistorical (damage: int): void

## Species

- \_idSpecies: String
- \_animals: List<Animals>
- \_speciesName: String
- .

«abstract»

## Employee

- tipo: String
- id: String
- name: String

## Tet

- type: String
- idResponsibilities: String
- habitats: List<Habitats>
- \_satisfaction: Satisfaction Tet

+ satisfaction(): int

## Vet

- type: String
- idResponsibilities: String
- \_clients: List<Species>
- medicalActs: List<Medical acts>
- \_satisfaction: Satisfaction Vet

+ satisfaction(): int

## Vaccines

- \_vaccine: List< Vaccine >
- \_wrongVaccine: List< Vaccine >
- allVaccination: List< Medicalacts >

- + registerVaccine (idVaccine: String, nameVaccine: String, idSpecies: String): void
- + vaccinateAnimal (idVaccine: String, idVET: String, idAnimalToVaccinate: String): void
- + showVaccinations (): void

## Vaccine

- \_idVaccine: String
- \_name: String
- \_number of applications: int
- \_species: List< Species >

- + sizeNames (e1: Species, e2: Species): int
- + damage (v: Vaccine, a: Animal): int

## Medicalacts

- \_idVet: String
- \_idSpecies: String

« interface »

CalculateSatisfaction

+ satisfaction(): double

Satisfaction Animal

- - m Some Animals: int
- - m Different animals: int
- - area: int
- - population: int
- - regeneration: int

+ satisfaction(): double

Satisfaction Zookeeper

- - trabalho: double

+ satisfaction(): double

Satisfaction Vet

- - trabalho: double

+ satisfaction(): double

<<abstract>>

Tree

- idArbol: String  
- nombre: String  
- edad: int  
- cleaningDifficulty: int  
- tipo: String  
- cicloBiologico: String  
- season: int  
  
+ cleaningEffort(): int  
+ seasonalEffort(a: Tree): int  
+ incrementEdad(season: Season) void

Per

+ seasonalEffort(a: Tree): int

Cad

+ seasonalEffort(a: Tree): int

Season

- currentSeason: int  
- seasons: String[]  
  
+ advanceSeason(): void