

Trees
- id: String
- name: String
- age: int
- seasonOrigin: String
- baseCleaningDifficulty: int
+ cleaningWork(): int
+ cleaningDifficulty(): int
+ seasonalEffect(): int
+ incrementAge(): void
+ getSeasonalEffect(): int

Species
- id: String
- name: String

Animal
- id: String
- name: String

Habitat
- id: String
- name: String

ModelManager
- hotel: Hotel
+ save(): void
+ saveAs(fileName: String): void
+ load(fileName: String): void
+ export(fileName: String): void

<<interface>>
SeasonState
+ nextSeason(context: SeasonContext): void

Winter
+ nextSeason(context: SeasonContext): void

SeasonContext
- currentState: SeasonState
+ setState(state: SeasonState): void
+ next(): void

<<abstract>>
WorkLog
- id: String
- name: String

Spring
+ nextSeason(context: SeasonContext): void

Summer
+ nextSeason(context: SeasonContext): void

Autumn
+ nextSeason(context: SeasonContext): void

AnimalSatisfaction
+ AnimalSatisfaction(a: Animal): int
+ equals(a: Animal, h: Habitat): int
+ difference(a: Animal, h: Habitat): int
+ area(h: Habitat): int
+ Population(h: Habitat): int
+ adequacy(a: Animal, h: Habitat): int

VetSatisfaction
+ VetSatisfaction(v: Vet): int
+ work(k: Keeper): int
+ Population(s: Species): int
+ n_vets(s: Species): int

KeeperSatisfaction
+ KeeperSatisfaction(k: Keeper): int
+ work(k: Keeper): int
+ habitat_work(h: Habitat): int
+ n_habitat_keepers(h: Habitat): int
+ area(h: Habitat): int
+ Population(h: Habitat): int
+ cleaningWork(t: Tree): int

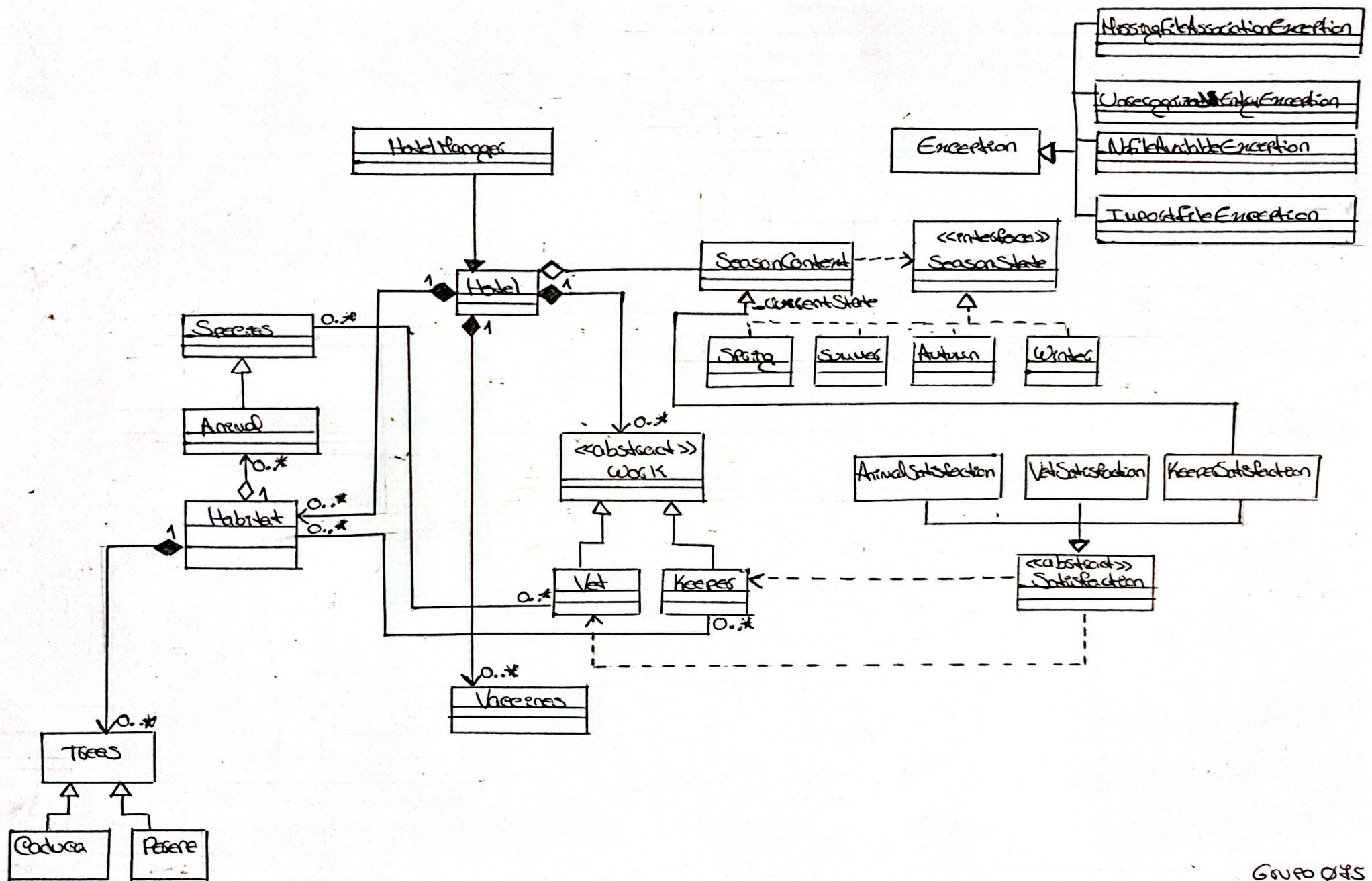
Vet
+ addSpeciesToVet(specie: String): void
+ removeSpeciesToVet(specie: String): void
+ countAnimalsInSpecies(): int
+ countVetsForSpecies(): int

Keeper
+ addHabitatToKeepers(habitat: String): void
+ removeHabitatToKeepers(habitat: String): void
+ countKeepersForHabitats(): int

Vaccines
- id: String
- name: String
- vaccinesSpeciesList[]: String
- n_vses: int
- collection: Collection<vaccineLog>
+ addVaccineToSpecies(id: Species: String): void
+ printVaccineForSpecies(id: Species: String): void
+ addVaccineLog(animal: String, vetId: String, speciesId: String): void

Hotel

```
+ advanceSeason(): void
+ showGlobalSatisfaction(): void
+ openAnimalsMenu(): void
+ openEmployeesMenu(): void
+ openHabitatMenu(): void
+ openVaccinesMenu(): void
+ openLookupsMenu(): void
+ showAllAnimals(): void
+ registerAnimal(name: String, id: String): void
+ transferToHabitat(animalid: String, habitatid: String): void
+ showSatisfactionOfAnimals(id: String): void
+ showAllEmployees(): void
+ registerEmployee(id: String, name: String, work: String): void
+ addResponsibility(employeeid: String, responsibilityid: String): void
+ removeResponsibility(employeeid: String, responsibilityid: String): void
+ showSatisfactionOfEmployee(id: String): void
+ showAllHabitats(): void
+ registerHabitat(id: String, name: String, area: int): void
+ changeHabitatArea(area: int): void
+ changeHabitatInfluence(id: String, influence: String): void
+ addTreeToHabitat(habitatid: String, treeid: String, name: String, age: int, cleardifficulty: int, type: String): void
+ showAllTreesInHabitat(): void
+ showVaccine(): void
+ registerVaccine(id: String, name: String, ids: String[]): void
+ vaccinateAnimal(vaccineid: String, vetid: String, animalid: String): void
+ showVaccinations(id: String): void
+ showAnimalsInHabitat(id: String): void
+ showMedicalActsOnAnimals(id: String): void
+ showMedicalActsByVeterinarian(id: String): void
+ showWrongVaccinations(id: String): void
```



Grupo 075

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Tiago Mateus
Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Mariana Carvalho