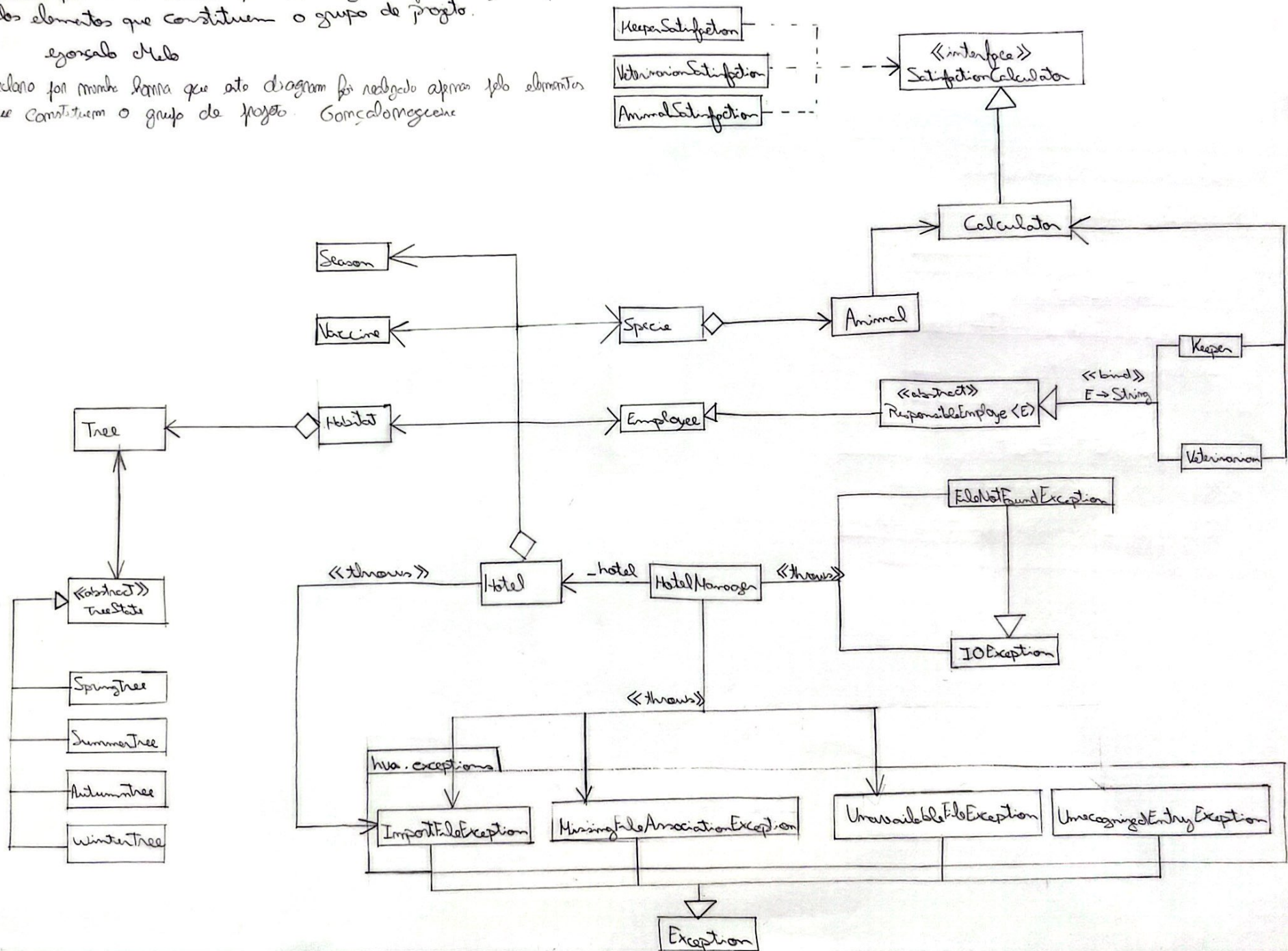


Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto.

egorsalo Melo

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Gonçalo Morges



Package: hwa.exceptions

ImportFileException

- «final» serialVersionUID: long
- «final» ERROR_MESSAGE: String

MissingFileAssociationException

- «final» serialVersionUID: long

UnavailableFileException

- «final» serialVersionUID: long
- ~ - filename: String

UnrecognizedEntryException

- «final» serialVersionUID: long
- «final» - entrySpecification: String

| Hotel |
|--|
| <pre># - species : ArrayList <Specie> # - habitats : ArrayList <Habitat> ## - employees : ArrayList <Employee> # - vaccines : ArrayList <Vaccine> # - season : Season</pre> |
| <pre>+ newSpecies (key : String, name : String) : void + newHabitat (key : String, name : String, area : int) : void + newEmployee (key : String, name : String) + newVaccine (key : String, name : String, species : String) : void + removeSpecie (specie : Specie) : void + removeHabitat (habitat : Habitat) : void + removeEmployee (employee : Employee) : void + removeVaccine (vaccine : Vaccine) : void + findSpecie (key : String) : Specie + findHabitat (key : String) : Habitat + findEmployee (key : String) : Employee + findVaccine (key : String) : Vaccine ~ importFile (filename : String) : void</pre> |

| HotelManager |
|--|
| <pre>- _hotel : Hotel + save() : void + saveAs(filename : String) : void + load(filename : String) : void + importFile(filename : String) : void</pre> |


```

Specie
- _key : String
- _name : String
- _log : HashMap<int, Animal>
+ hash (animalKey : String) : int
+ insert (animal : Animal) : void
+ remove (animal : Animal) : void
+ findAnimal (animalKey : String) : Animal

```

```

<<abstract>>
Employee
- _key : String
- _name : String
+ <<abstract>> work() : int

```

```

<<abstract>>
ResponsibleEmployee <E>
- _responsibility : ArrayList<E>
+ insert (value : E) : void
+ remove (value : E) : void

```

```

Veterinarian
- _calculator : Calculator
+ newResponsibility (specie : Specie) : void
+ removeResponsibility (specie : Specie) : void
+ work() : int

```

```

Keeper
- _calculator : Calculator
+ newResponsibility (habitat : Habitat) : void
+ removeResponsibility (habitat : Habitat) : void
+ work() : int
+ workInHabitat() : int

```

```

VeterinarianSatisfaction
+ satisfaction (o : Object) : int
KeeperSatisfaction
+ satisfaction (o : Object) : int

```

```

Animal
- _key : String
- _name : String
- _specieKey : String
- _healthState : String
- _habitat : Habitat
- _calculator : Calculator
+ equals() : int
+ differs() : int
+ adequation() : int
+ updateHealthState (state : String) : void

```

```

<<interface>>
SatisfactionCalculator
+ satisfaction (o : Object) : int

```

```

AnimalSatisfaction
+ satisfaction (o : Object) : int

```

```

Calculator
- _satisfactionCalculator : SatisfactionCalculator
+ calculateSatisfaction (o : Object, rc : SatisfactionCalculator) : int

```

```

Habitat
- _key : String
- _name : String
- _area : int
- _treeCount : int
- _animalCount : int
- _specieCount : HashMap<int, int>
- _adequation : HashMap<int, String>
- _treeLog : HashMap<int, Tree>
+ hash (key : String) : int
+ insertAdequation (specie : Specie) : void
+ insertSpecieCount (animal : Animal) : void
+ insertTree (tree : Tree) : void
+ findAdequation (animal : Animal) : String
+ findSpecieCount (animal : Animal) : int
+ findTree (treeKey : String) : Tree
+ removeTree (tree : Tree) : void
+ changeAdequation (specie : Specie) : void
+ updateSpecieCount (animal : Animal) : void
+ updateTreeCount() : void
+ updateAnimalCount() : void

```

```

Tree
- _key : String
- _name : String
- _age : int
- _type : String
- _cleaningDifficulty : int
- _creationSeason : int
- _state : TreeState
+ cleaningEffort() : int
+ changeState() : void
+ setState (state : TreeState) : void

```

```

<<abstract>>
TreeState
# - tree : Tree
- <<abstract>> _deciduous : int
- <<abstract>> _evergreen : int
+ <<abstract>> setState() : void
+ <<abstract>> reasonEffort (treeType : String) : int

```

```

Season
- _currentSeason : int
+ advanceSeason() : void

```

```

Vaccine
- _key : String
- _name : String
- _specie : String[]
- _log : ArrayList<String>
+ damage (animal : Animal) : int
+ nameSize (name1 : String, name2 : String) : int
+ commonCharacters (name1 : String, name2 : String) : int
+ findSpecie (animal : Animal) : boolean

```

```

SpringTree
+ setState() : void
+ reasonEffort (treeType : String) : int

```

```

AutumnTree
+ setState() : void
+ reasonEffort (treeType : String) : int

```

```

SummerTree
+ setState() : void
+ reasonEffort (treeType : String) : int

```

```

WinterTree
+ setState() : void
+ reasonEffort (treeType : String) : int

```