ManaPotion +HealthPotion(String, double,int) +use() HWeapon(String, double, int, do HWeapon(Weapons) HyeathtackEs(): int HgethersePts(): int Hgethange(): int Huse(): Hotel HealthPotion Weapon ittackpts: int ange: double restorepts: int Potion(String, double, in +Potion(Potions) +getRestorePts(): int +getDefensePts(): int +getAttackPts(): int +use() name: String
-price: double
-trem(String Jouble)
-trem(String Jouble)
-trem(Trems)
-tgetName(): String
-getPrice(): double
-tuse()
-tuse(): int
-getUtensePts(): int
-getUttacGPts(): int <<abstract>>
Potion <<abstract>> Item attackpts: int weight: double Threstory()

egeffoatBook(); double
egetfoatBook(); int
egetMitmas(); item
egetSpaces(int); item
egetSpaces(int); item
egetMitmas(); booken
enemvetEtem(itt)
equipItmas(itt)
equipItmas(itt) -spaces: int -gold: double -pair: ArrayList <Pair> Inventory +Pair() +Pair(T,B) +getItem(): 1 +getBool(): E +setItem(T) +setBool(B) Pair #power: int
+Knight(String.int)
+Knight(String.int)
#getDefensePoints(): int
#attack(Character)
+attack(Character)
+addPower(int) <enumeration
Element</pre> air earth fire water physical <<enumeration
Color</pre> blue red green yellow white black #Power: int
#Poladin(String.int)
#Poladin(String.int)
#Poladin(String.int)
#GethtRackonists(): int
#GetheRosePoints(): int
#attack(Creature)
"4Power(int) Paladin name: String .color: Color .win: int .lose: int .draw: int .characters: Arraylist <Character> +Toan(String,Color)

+Optimane():String

+Optimane():String

+Optimane():String

+Optimane():String

+Optimane():String

+Optimane():String

+Optimane()

+Optima +Druid(String,int)
#getktackPoints(): int
#getktackPoints(): int
+attack(Character)
+attack(Creature)
+heal(Individual)
+addMisdom(int) Druid +Sorcerer(String,int)
#getAttackPoints(): int
#getDefensePoints(): int
#strack(Inaracter)
+attack(Creature)
+heal(Individual)
+heal(Midvidual) Sorcerer +Dragon(String)
+attack(Individual)
+getDefensePoints(): int
+getAttackPoints(): int +Individual(String)
+addHP(Lit)
+addHP(Lit)
+getHP(): int
+getHP(): int
+getHame(): String
#getDefensePoints(): int
#getAttack(Characke)
+attack(Characke): int
+attack(Characke) +EarthStrike(): int
+WaterStrike(): int
+FireStrike(): int
+FireStrike(): int
+Healing(Indixidual): int
+IntenseHealing(Indixidual): in <<abstract>>
Individual Dragons <<ir>
<ir>
<ir>
<ir>
<ir>
<ir>
<ir>

Spells e: String int int -element: Element
+Creature(String)
+getElement(Creature): Element
+getName()(): String +Witch(String)
+addWisdom(int)
+attack(Individual)
+getDefensePoints(): int
+getAttackPoints(): int <<abstract>>
Creature Ą Witchs +addPower(int)
+attack(Individual)
+getDefensePoints(): int
+getAttackPoints(): int #WASOOM: INT +Demon(String) +addWisdom(int) +attack(Individual) +getDefensePoints(): int +getAttackPoints(): int Demons

Diagrama UML - Tabalho 3 Pogramação Orientada a Objetos Lucas Tgnoli Munhoz e Rafael Martins de Teitas