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 Armor (String, double, int, double)
Armor (Armor K)

+getbefensePts(): int

+getAtackPts(): int

+getAtackPts(): int

+getAtackPts(): int

+getAtackPts(): int

+getAtackPts(): int restorepts: int Potion(String, double, in +Potion(Potions) +getRestorePts(): int +getDefensePts(): int +getAttackPts(): int +use() <<abstract>>
Potion -name: String
-price: double
+Item(String,double)
+Gtem(Item%)
+getName(): String
+getPrice(): double
+use() getDefensePts(): int getAttackPts(): int <abstract>> attackpts: int weight: double Threatory()

**Burnentory()

* -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)
#gctAttackDoints(): int
#gctDefensePoints(): int
+attack(Character)
+attack(Creature)
+addPower(int) <enumeration
Element</pre> air earth fire water physical <<enumeration
Color</pre> blue red green yellow white black +Paladin(String.int)
#getAttackPoints(): int
#getDefensePoints(): int
+attack(Character)
+attack(Creature)
+addPower(int) Paladin name: String .color: Color .win: int .lose: int .draw: int .characters: Arraylist <Character> +Toan(String Color)
+Toan(String Color)
+Operation (): String
+Ope +Druid(String,int)
#getktackPoints(): int
#getbeforsePoints(): int
+attack(Character)
+attack(Creature)
+adoMisdom(int) Druid +Sorcerer(String,int)
#getAttackPoints(): int
#getDefensePoints(): int
#strack(Inaracter)
+attack(Creature)
+heal(Individual) Sorcerer +Dragon(String)
+attack(Individual)
+getDefensePoints(): int
+getAttackPoints(): int +Individual(String)
+addHP(Lint)
+addHP(Lint)
+getHP(): int
+getHP(): int
+getHame(): String
+getDefensePoints(): int
+getAttack(Character)
+attack(Character) +EarthStrike(): int
+WaterStrike(): int
+FireStrike(): int
+FaireStrike(): int
+Healing(Indixidual): int
+IntenseHealing(Indixidual): in <<abstract>>
Individual Dragons 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 +Demon(String)
+addwisdom(int)
+attack(Individual)
+getDefensePoints(): int
+getAttackPoints(): int Demons

Diagrama UML - Trabalho 3 Programação Orientada a Objetos Lucas Tognoli Munhoz e Rafael Martins de Freitas