

## MODELO RELACIONAL

Map = { mapID }

*mapID: número de identificação de um mapa que pode ser utilizado por um jogador.*

HaveMapRegion = { mapID, coordinates }

*coordinates: código de identificação das coordenadas do jogador;*

Region = { coordinates<sup>\*1</sup>, biome, monument, dangerLevel, monument<sup>\*1o</sup> }

*dangerLevel: nível de perigo que a região apresenta;*

Biomes = { biomesID<sup>\*2</sup>, coordinates<sup>\*1</sup>, resourceAbundance, resourceAvailability, type, climate<sup>\*3</sup> }

*resourceAbundance: quantidade de recursos que um bioma pode apresentar — em questão de estar abundante ou não — a um ou mais jogadores;*

*resourceAvailability: apresenta se há ou não recursos no bioma;*

Flora = { biomes<sup>\*2</sup>, flora }

Fauna = { biomes<sup>\*2</sup>, fauna }

Climate = { climateID<sup>\*3</sup>, temperature, event, statusEffect, visibility }

Characters = { charactersID<sup>\*4</sup>, name, position, characterModel, climate<sup>\*3</sup>, type, item }

GatherYield = { character<sup>\*4</sup>, gatherYield }

EnterCombatCharacters = { enterCombatCharactersID<sup>\*5</sup>, firstCharacter<sup>\*4</sup>, secondCharacter<sup>\*4</sup> }

*firstCharacter: código do primeiro jogador no combate;*

*secondCharacter: código do segundo jogador no combate;*

CombatLog = { enterCombatCharacters<sup>\*5</sup>, indexLog, log }

PlayerCharacters = { charactersID<sup>\*4</sup>, hydration, poisoned, hunger, type, equippedItems1, equippedItems2, equippedItems3, equippedItems4, equippedItems5, backpack<sup>\*13</sup> }

RecruitableCharacters = { charactersID<sup>\*4</sup>, specialization, recruited }

MainCharacter = { charactersID<sup>\*4</sup>, owner<sup>\*6</sup> }

RespawnLocation = { ownerID\*<sup>6</sup>, description, timer, X, Y }

NPCs = { charactersID\*<sup>4</sup>, isAgressive, aggroRange, enemyGrade, type }

*isAgressive: código que apresenta se um NPC é agressivo ou não.*

Animals = { charactersID\*<sup>4</sup>, sound, modelType }

Scientists = { charactersID\*<sup>4</sup>, hasDialogue }

*hasDialogue: código que apresenta se existe a possibilidade de diálogo com o 'scientist'.*

DialogueText = { character\*<sup>4</sup>, dialogue }

Items = { itemsID\*<sup>7</sup>, stackSize, lootGrade\*<sup>8</sup>, name, type, quantity, durability, craftable, type, backpack\*<sup>13</sup>, character\*<sup>4</sup> }

*craftable: código que apresenta se um item pode ser criado a partir da junção de outros itens por parte dos jogadores.*

Ingredients = { items\*<sup>7</sup>, lootGrade\*<sup>8</sup>, ingredient }

Weapons = { itemsID\*<sup>7</sup>, type }

Melee = { itemsID\*<sup>7</sup>, canBeThrown, fleshGatherRate, oreGatherRate, treeGatherRate }

Ranged = { itemsID\*<sup>7</sup>. recoil, attackRange, ammoCapacity, modSlots, fireMode, fireRate, accuracyModifier }

Consumables = { itemsID\*<sup>7</sup>, instantHeal, healOverTime, hidratationYield, bleedingYield, radiationYield, poisonYield, hungerYield, vomitChance, type }

*hidratationYield: código que apresenta a quantidade de hidratação que um consumível oferece ao personagem.*

*bleedingYield: código que apresenta a quantidade de cura de sangramento que um consumível oferece ao personagem.*

*radiationYield: código que apresenta a quantidade de redução ou aumento de radiação que um consumível oferece ao personagem.*

*poisonYield: código que apresenta a quantidade de dano ou cura de envenenamento que um consumível oferece ao personagem.*

*hungerYield: código que apresenta a quantidade de redução de fome que um consumível oferece ao personagem.*

*vomitChance: código que apresenta a chance que um consumível oferece ao personagem de sofrer a êmese.*

Teas = { itemsID<sup>\*7</sup>, statusUpgradeType, upgradePercentage }

Clothing = { itemsID<sup>\*7</sup>, coldResistance, radResistance, explosionResistance, meleeResistance, rangedResistance, biteResistance, equipmentSlot, wetResistance }

*coldResistance: Valor que representa a resistência ao frio concedida pelo equipamento*

*radResistance: Valor que representa a resistência à radiação concedida pelo equipamento.*

*explosionResistance: Valor que representa a resistência à dano de explosão concedida pelo equipamento.*

*meleeResistance: Valor que representa a resistência à dano corpo-a-corpo concedida pelo equipamento.*

*rangedResistance: Valor que representa a resistência à dano a distância concedida pelo equipamento.*

*biteResistance: Valor que representa a resistência à dano de mordidas concedida pelo equipamento.*

*wetResistance: O item concede resistência ao status de molhado. IE: Roupas de radiação, roupa de mergulho.*

Components = { itemsID<sup>\*7</sup> }

Resources = { itemsID<sup>\*7</sup>, isPrimary }

DropCharactersItems = { dropCharactersItemsID, item<sup>\*7</sup>, lootGrade<sup>\*8</sup>, character<sup>\*4</sup> }

PlayerCharactersGeneratesItem = { character<sup>\*4</sup>, items<sup>\*7</sup> }

WeaponsAreComposedOfComponentsResources = { weapons<sup>\*7</sup>, components<sup>\*7</sup>, resources<sup>\*7</sup> }

ConsumablesAreComposedOfComponentsResources = { consumables<sup>\*7</sup>, components<sup>\*7</sup>, resources<sup>\*7</sup> }

ClothingAreComposedOfComponentsResources = { clothing<sup>\*7</sup>, components<sup>\*7</sup>, resources<sup>\*7</sup> }

ResourceNodes = { resourceNodesID<sup>\*9</sup>, nodeType, maxYield, durabilityDamage, biomes<sup>\*2</sup> }

ResourceNodesGenerateItems = { resourceNodes<sup>\*9</sup>, item<sup>\*7</sup> }

Monuments = { name<sup>\*10</sup>, monumentSize, lootGrade, enemyGrade }

regionsMonuments = { name<sup>\*10</sup>, regionMonument }

Structures = { structureID<sup>\*11</sup>, monument<sup>\*10</sup> }

LootCrates = { lootCratesID<sup>\*12</sup>, grade }

StructuresContainsLootCrates = { structure<sup>\*11</sup>, lootCrates<sup>\*12</sup> }

Party = { partyID, character<sup>\*4</sup>, capacity }

partyID: código do grupo de jogadores que estão jogando junto.

Backpack = { ownerID<sup>\*13</sup>, slot(totalSlots - (totalSlots - 1)) ... slot(totalSlots-1),  
slot(totalSlots), totalSlots }

*ownerID: código do dono da 'backpack'.*

*slots: A mochila terá uma quantidade de slots determinada pelo valor de totalSlots, cada slot será um atributo diferente, possuindo um valor específico para o ID item que o ocupa ou 0 caso não haja item.*