

# Trabalho prático 1 - Problema: pesquisas de digievolução de digimon

O jogo eletrônico Digimon Story: Cyber Sleuth, é um jogo do Playstation Vita que faz parte da aclamada série de jogos de Digimon que atualmente é um jogo multiplataforma (PS4, PSVita, PC e Nitendo Switch).

Resumidamente, o jogo consiste de interagir, aperfeiçoar e colecionar Digimons (criaturas, de maioria antropomórfica, baseadas em aspectos históricos e culturais humanos, tal como a mitologia; compostas de dados digitais e vivem no digimundo. originalmente, nascem de digi-ovos e crescem até serem mais poderosas e imponentes nos níveis mais elevados, cujo crescimento acontece em forma de digievolução).

Um digimon no jogo terá 12 características:

Nome, Estágio, Tipo, Atributo, Memória, Espaço de equipamento, Saúde, Espírito, Ataque, Defesa, Inteligência, velocidade.

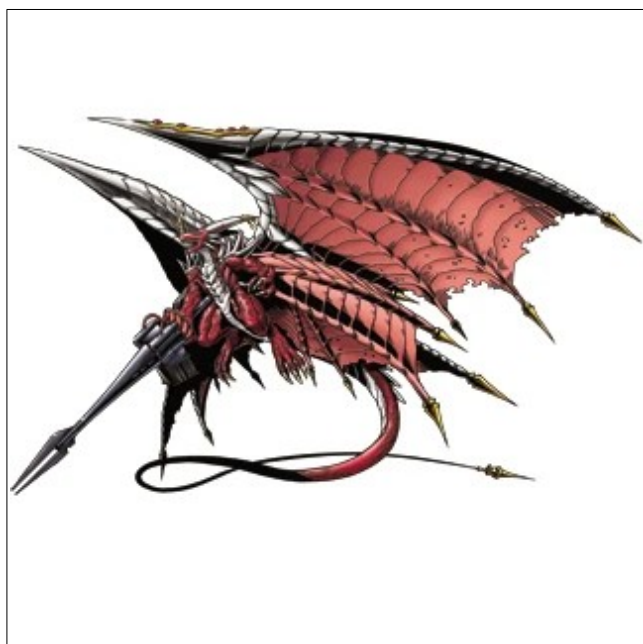


## Exemplo

Name: Botamon  
Stage: Baby  
Type: Free  
Attribute: Neutral  
Memory: 2  
Equip Slots: 0  
HP: 690  
SP: 68  
Atk: 77  
Def: 95  
Int: 76  
Spd: 61

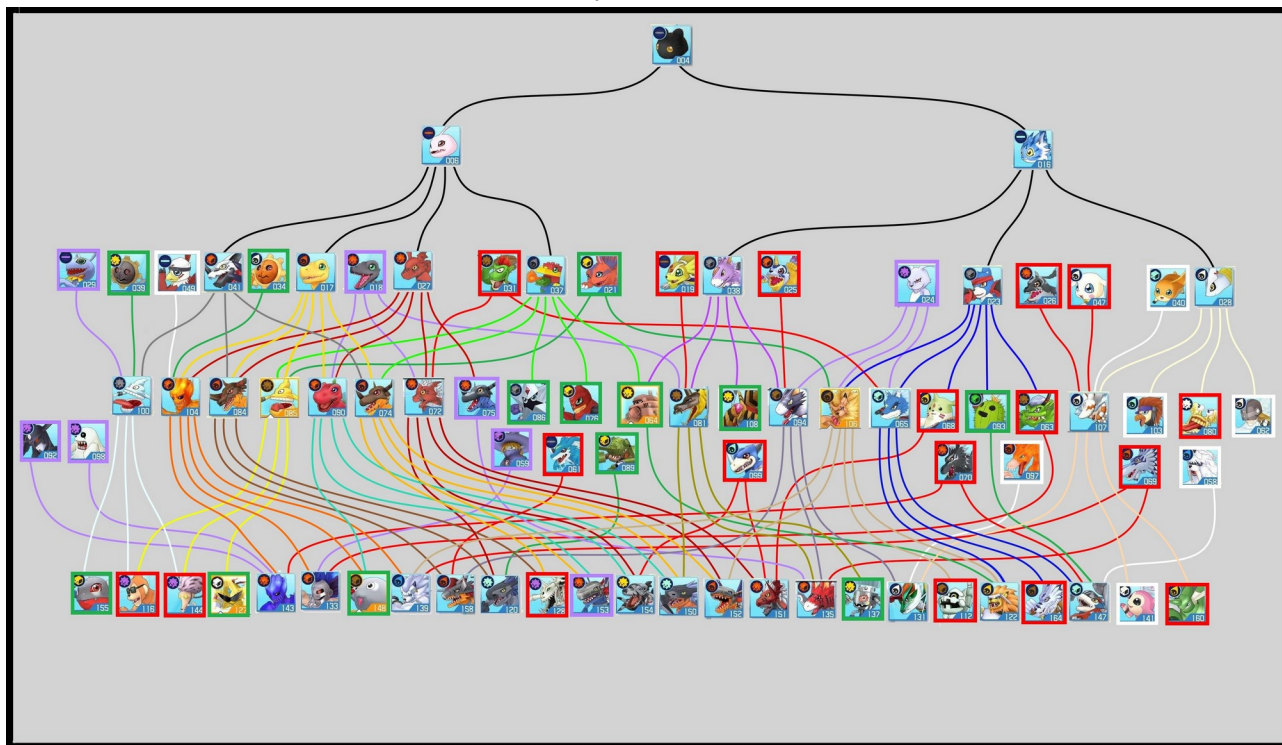
**Na imagem:** ilustração de Botamon, digimon estágio bebê

Uma das maneiras de progressão no jogo são as digievoluções, que nada mais são que o nome do processo que o digimon passa para trocar de forma dada certas circunstâncias. Geralmente ela segue a regra de estágios digiovo » baby » In-training » rookie » champion » Ultimate » Mega » Ultra.



**Na imagem:** ilustração de Examon, uma das possíveis digievoluções estágio Ultra de Botamon

Diferente de jogos como Pokémon, a digievolução não é linear. A partir do estágio baby, um digimon pode tomar diferentes formas, e essas formas também podem se ramificar em diferentes formas.



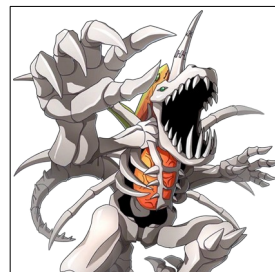
Na imagem: Possíveis caminhos de digievolução do Botamon até o estágio ultimate

Exemplo:



Greymon digievolui para Skullgreymon se:

**Greymon tem :**  
 ATK = 120,  
 DEF = 85,  
 SPD = 70,  
 e seu level é 28.



Greymon digievolui para Metalgreymon se:

**Greymon tem :**  
 ATK = 350,  
 DEF = 100,  
 CAM = 50%,  
 e seu level é 30.



O database criado mapeará todos os digimons do jogo e seus atributos, tal como todas as digievoluções que acontecem entre eles e transformará isso em um grafo (sendo os digimons os vértices e as digievoluções as arestas), e o nosso problema simples será produzir um possível caminho qualquer de um digimon até outro através de digievoluções (exemplo: quais digimons o Botamon deve passar até digievoluir para um HerculesKabuterimon).

## Descrição do trabalho:

**Aluno:** Amaury Luiz Chaves Junior (19111295)

**Banco de dados utilizado:** digimons e digievoluções do jogo "[Digimon Story: Cyber Sleuth](#)"

**Dados:** Digimon.csv, Digivolves\_To.csv → respectivamente a tabela com os dados dos vértices do banco de dados, com exceção da ID do vértice (@rid) e a tabela com os dados das arestas, com exceção da ID da aresta(@rid)

**Servidor escolhido:** OrientDB

**Problema escolhido:** caminho entre dois vértices

**OS Utilizado:** Majaro Linux

**Linguagens utilizadas:** Java, SQL, Javascript, Python

## Referências:

- 1 – [OrientDB - ArchLinux](#)
- 2 – [OrientDB - DockerHub](#)
- 3 - [Maheshwar Ligade - OrientDB docker setup in your local machine](#)
- 4 - [runaman – DB: OrientDB](#)

## Downloads:

**docker**

- Pull docker image

**imagem do orientdb no dockerhub**

- docker pull orientdb

## Container:

**Criando container:**

- \$ docker run -d --name orientdb -p 2424:2424 -p 2480:2480 -e ORIENTDB\_ROOT\_PASSWORD=rootpwd orientdb

**Iniciando o container:**

- \$ docker start orientdb
- localhost:2480

**Encerrando o container:**

- \$ docker stop orientdb

## Console:

**Acessando o Console:**

```
$ docker exec -it orientdb bash
cd $ORIENTDB_HOME/bin
console.sh
```

**Conectando ao localhost:**

```
CONNECT remote:localhost root rootpwd
```

## Comandos:

### **Criar database:**

CREATE DATABASE [nome da tatabase] [usuário] [senha]

### **Deletar database:**

DROP DATABASE [nome da tatabase] [usuário] [senha]

### **Listar as databases:**

LIST DATABASES

### **Abrir database:**

open [nome da database] [usuário] [senha]

### **Criar usuário**

CREATE USER usuário IDENTIFIED BY senha ROLE função\_usuário

### **Criar Vértice (Classe)**

CREATE CLASS vertex EXTENDS V

### **Criar Vértice (Entidade)**

CREATE VERTEX vertex SET atr = "value",...

### **Remover vértice (Entidade)**

DELETE VERTEX vertex [condição]

\* se condição vazia, deleta TODAS as entidades

### **Criar Aresta (Classe)**

CREATE CLASS edge EXTENDS E

### **Criar Aresta (Entidade)**

CREATE EDGE edge FROM vertex\_in TO vertex\_out SET atr = "value",...

### **Remover aresta(Entidade)**

DELETE EDGE edge [condição]

\* se condição vazia, deleta TODAS as entidades

### **Deletar Classe**

DROP CLASS classe

## Processo de criação de Database e solução do problema:

**1** - Arquivos out e out1 gerados através dos programas b.py e c.py (com auxílio da biblioteca matplotlib e pandas) com base na database Digimon.csv, Digivolves\_To.csv.

**2** - Ctrl + C e Ctrl + V do arquivo out e out1 no console (conectado à database).

**3** - Criação do programa do caminho entre dois vértices feito com NodeJS (com auxílio da biblioteca OrientJS)

**index.js** » solução de caminho entre botamon e HerculesKabuterimon

### **Exemplo de queries possíveis:**

SELECT outV().Name, inV().Name FROM Digivolves\_To WHERE outV().Name = "HerculesKabuterimon"

Querie pelas digievoluções do digimon HerculesKabuterimon

SELECT \* FROM Digimon WHERE Name = "Agumon"

Querie pelo digimon agumon