

PHPRPG

Nós temos a tarefa de programar um simples jogo em RPG utilizando os conceitos da orientação a objetos e a linguagem de programação PHP.

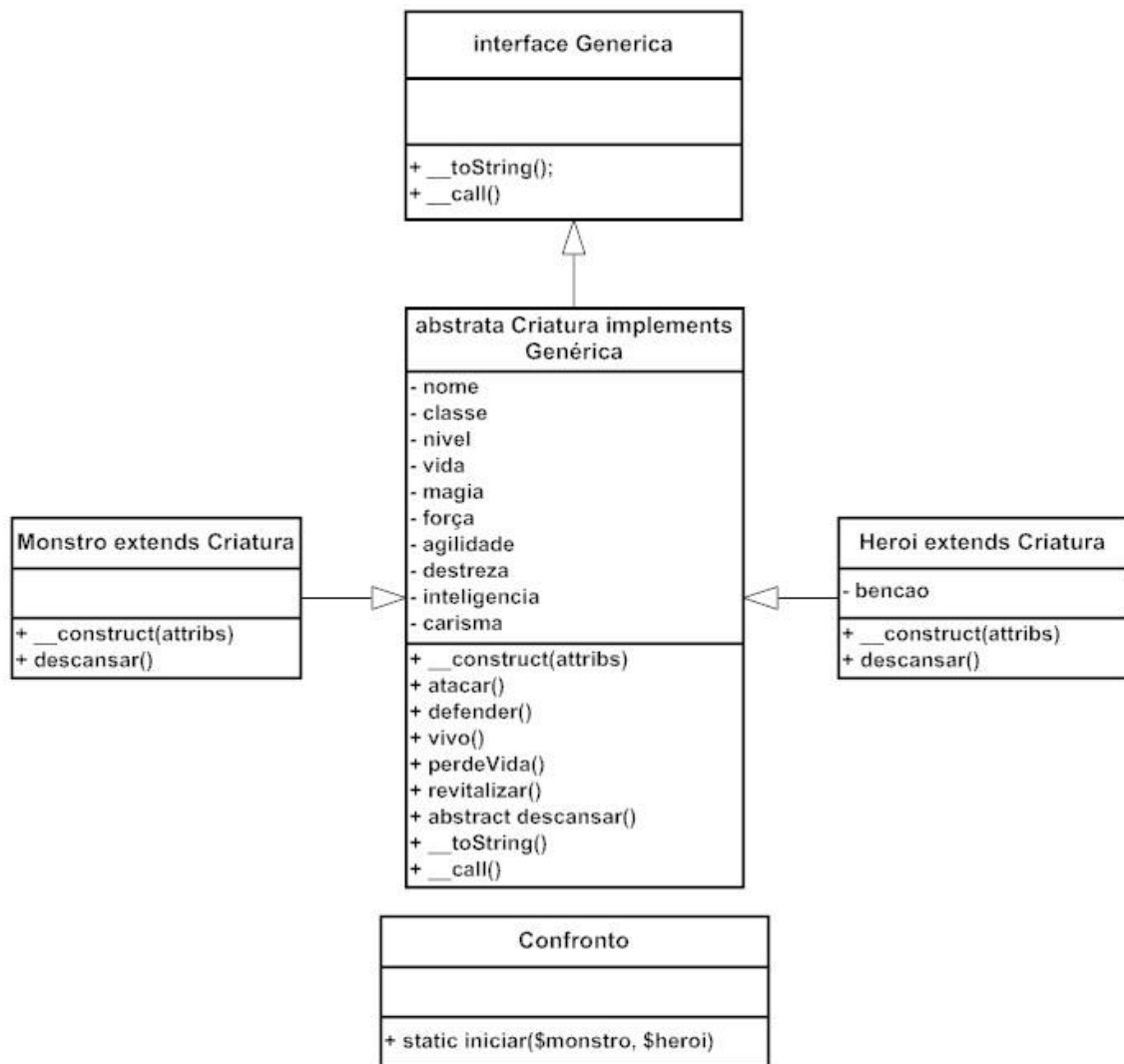


Figura 1 - Diagrama de Classes.

O primeiro passo para isso é implementar as classes de acordo com o diagrama descrito na Figura 1 - Diagrama de Classes.

DESCRIÇÃO DAS CLASSES

| Tipo | Nome | Descrição |
|-----------|-----------|---|
| interface | Generica | É um molde para definir os métodos especiais __call() e __toString() que deverão ser implementados nas demais classes do sistema |
| abstract | Criatura | Define os atributos das criaturas que irão atuar em nosso jogo, deve-se implementar os métodos especiais descritos por sua interface. Cria as ações para as criaturas do jogo e define um método abstrato que será implementado em suas classes herdeiras |
| class | Monstro | Representa um monstro no jogo. Implementa o método abstrato da classe pai |
| class | Heroi | Representa um herói no jogo. Implementa o método abstrato da classe pai. Possui um atributo especial benção, por ser uma criatura do bem |
| class | Confronto | Representa a luta entre dois personagens do jogo, possui apenas um método estático que recebe 2 objetos do tipo criatura como parâmetro, e desenvolve todo o confronto |

ATRIBUTOS DAS CRIATURAS

| Atributo | Descrição |
|--------------|--|
| nome | Nome da criatura |
| classe | Classe da criatura |
| nível | Nível de evolução da criatura. Valores entre (1 e 10) |
| vida | Escala que indica o quanto a criatura está machucada. Valores entre (1 e 100) |
| magia | Escala que indica o quanto essa criatura poderá usar magia em seus combates. Valores entre (1 e 100) |
| força | Força da criatura. Valores entre (1 e 32) |
| agilidade | O quanto a criatura é ágil. Valores entre (1 e 12) |
| destreza | Destreza da criatura. Valores entre (1 e 16) |
| Inteligência | Inteligência da criatura. Valores entre (1 e 20) |
| carisma | O quanto carismática é a criatura. Valores entre (1 e 12) |

Ps. Os atributos devem ser definidos como **protected**.

MÉTODOS DAS CRIATURAS

| Método | Descrição |
|--------------------|--|
| atacar() | Gera um coeficiente que será comparado com a defesa de seu oponente, caso o coeficiente de ataque seja maior que o de defesa de seu oponente o ataque é efetuado |
| defesa() | Gera um coeficiente que será comparado com o ataque de seu oponente, caso o coeficiente de defesa seja maior que o de ataque de seu oponente a defesa é efetuada |
| descansar() | Gera um coeficiente que, se maior que 20 , revitaliza parte da vida e da magia da criatura |
| vivo() | Retorna se uma criatura está viva |
| perdeVida() | Retira a vida que a criatura deve perder |
| revitalizar() | Adiciona as propriedades oferecidas pelo descanso, caso aconteça |
| __construct(attrs) | Construtor da criatura, deve-se passar todos os atributos como parâmetro, e iniciá-los no objeto |

LÓGICA DO JOGO

Devem-se criar vários monstros e heróis que possam combater entre si. A luta só termina quando uma das criaturas estiver morta.

Os métodos **atacar()**, **defender()** e **descansar()** vão gerar um coeficiente entre **0** e **100**.

Esse coeficiente será gerado levando em consideração:

- Composição dos atributos de cada criatura;
- Constante multiplicadora (r).

A constante multiplicadora será um número aleatório entre **1.1** e **1.6** (que irá incrementar a ação de **10 a 60%**).

Abaixo estão descritas as fórmulas para cada ocasião:

Atacar:

$$(nível + força + agilidade + (magia * 0.1)) * r$$

Defender:

$$(agilidade + (destreza * 1.8) + inteligencia) * r + 10$$

Descansar:

$$(vida + nivel + [benção]) * r$$

PS. Se a benção existir é adicionada (herói)

Caso o valor calculado passe do coeficiente máximo (100) essa criatura deverá manter o coeficiente máximo.

CONFRONTO

Em um confronto, o primeiro a atacar é sempre o **monstro**, e a cada turno o ataque deve ser revezado entre herói e monstro.

Quando houver um ataque:

```
criatura1->atacar() // 42: Gera um coeficiente
criatura2->defender() // 45: Gera um coeficiente
```

Se o **coeficiente de ataque da Criatura1** for maior que o coeficiente de defesa da **Criatura2** então a Criatura2 **perderá 8 pontos de vida**. O confronto deve continuar enquanto uma das criaturas estiver viva.

Caso o coeficiente de descanso atinja um nível maior que **20**, então a criatura será revitalizada em **50 pontos de vida e 30 pontos de magia**. Com a única condição de que seus pontos **não ultrapassem os pontos máximos de vida (100)** e de magia (100);

Ao final do confronto **a criatura vencedora poderá ter o direito de descansar antes de partir para um próximo confronto**. Para verificar se a criatura vai ou não descansar, deve-se assim como no ataque() e na defesa() **gerar um novo coeficiente pela fórmula (Descansar)** descrita anteriormente.

Em um dado momento, pela chave do campeonato, dois membros da mesma equipe podem disputar entre si, com o objetivo de haver apenas 1 vencedor.

ALGORÍTMO DE EXEMPLO

```
flag = 1;
enquanto (criatura1->vivo && criatura2->vivo){
    se (flag == 1){
        flag = 2;
        criatura1->atacar();
        criatura2->defender();
        se (ataque funcionou){
            criatura2->perdeVida(8)
        }
    }
    senão{
        flag = 1;
        criatura2->atacar();
        criatura1->defender();
        se (ataque funcionou){
            criatura1->perdeVida(8)
        }
    }
}

se coeficienteDeDescanso > 20 { criaturaVencedora-
>descansar(); }
```

Antes de iniciar o confronto imprima as informações de cada um dos concorrentes:



Nome: **Montaro**
Classe: **Elfo das Trevas**
Nível: **7**
Vida: **100**
Magia: **100**
Força: **10**
Agilidade: **5**
Destreza: **10**
Inteligencia: **15**
Carisma: **10**

X

Bênção: **10**
Nome: **Dark Glorysson**
Classe: **Humano**
Nível: **8**
Vida: **100**
Magia: **100**
Força: **10**
Agilidade: **5**
Destreza: **10**
Inteligencia: **15**
Carisma: **2**



Em seguida, mostre na tela as informações geradas a cada turno, por exemplo:

ROUND 1

Montaro atacou (35.2 x 32) Dark Glorysson

Dark Glorysson perdeu 8 pontos de vida

Vida do herói 92

Vida do monstro 100

ROUND 2

Dark Glorysson atacou (39.6 x 38) Montaro

Montaro perdeu 8 pontos de vida

Vida do herói 92

Vida do monstro 92

ROUND 3

Dark Glorysson defendeu (38 x 35.2) o ataque Montaro

Vida do herói 92

Vida do monstro 92

...

Por fim, mostre quem foi o vencedor, e se o vencedor teve ou não o direito de descansar:

DARK GLORYSSON VENCEU O CONFRONTO

Dark Glorysson pode ***descansar***


HERÓIS

| | |
|---|----------------------|
|  | Nome: Dark Glorysson |
| | Classe: Humano |
| | Nível : 8 |
| | Vida: 100 |
| | Magia: 100 |
| | Força: 10 |
| | Agilidade: 5 |
| | Destreza: 10 |
| | Inteligência: 15 |
| | Carisma: 2 |
| | Benção: 10 |

| | |
|---|-----------------|
|  | Nome: Mellayne |
| | Classe: Humano |
| | Nível : 5 |
| | Vida: 100 |
| | Magia: 80 |
| | Força: 25 |
| | Agilidade: 10 |
| | Destreza: 8 |
| | Inteligência: 2 |
| | Carisma: 20 |
| | Benção: 10 |

| | |
|---|-----------------|
|  | Nome: Gryin |
| | Classe: Anão |
| | Nível : 10 |
| | Vida: 100 |
| | Magia: 50 |
| | Força: 30 |
| | Agilidade: 5 |
| | Destreza: 10 |
| | Inteligência: 5 |
| | Carisma: 1 |
| | Benção: 10 |

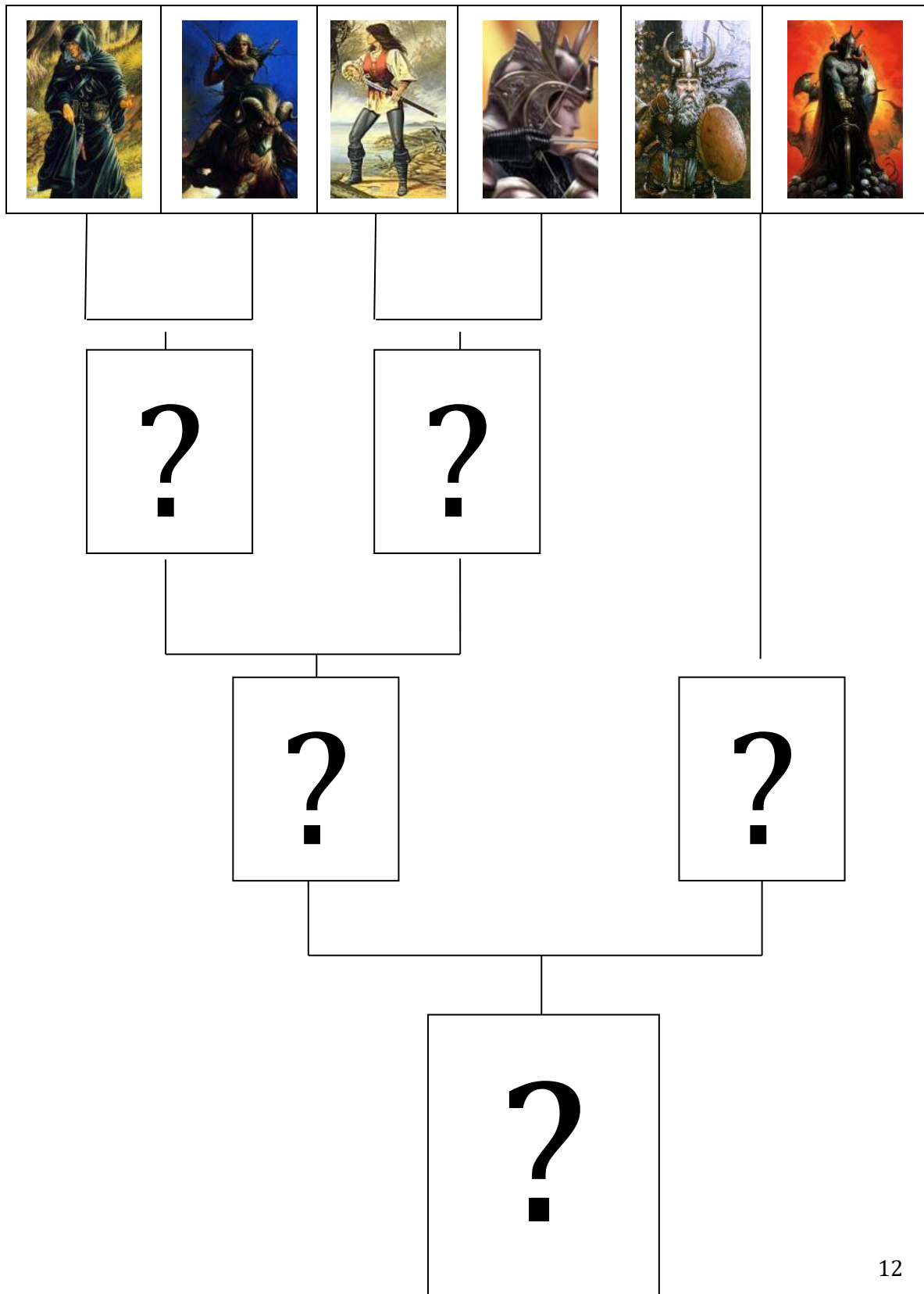
MONSTROS

| | |
|---|-------------------------|
|  | Nome: Montaro |
| | Classe: Elfo das Trevas |
| | Nível : 7 |
| | Vida: 100 |
| | Magia: 100 |
| | Força: 10 |
| | Agilidade: 5 |
| | Destreza: 10 |
| | Inteligência: 15 |
| | Carisma: 10 |

| | |
|--|------------------|
|  | Nome: Lord Black |
| | Classe: Humano |
| | Nível : 10 |
| | Vida: 100 |
| | Magia: 20 |
| | Força: 32 |
| | Agilidade: 12 |
| | Destreza: 10 |
| | Inteligência: 10 |
| | Carisma: 1 |

| | |
|---|------------------|
|  | Nome: Matilda |
| | Classe: Elfo |
| | Nível : 10 |
| | Vida: 100 |
| | Magia: 80 |
| | Força: 10 |
| | Agilidade: 8 |
| | Destreza: 10 |
| | Inteligência: 18 |
| | Carisma: 5 |

CHAVE DO CAMPEONATO



INSTRUÇÕES FINAIS

Grupo: de 3 ou 4 Integrantes (sugestão: o mesmo do trabalho final)

AVALIAÇÃO:

- Estrutura do código;
- Layout;
- POO;
- Funções do PHP;

Todos os integrantes deverão participar na CODIFICAÇÃO do sistema;

A tarefa deverá ser entregue via *pull-request* no *GitHub*.

O repositório para entrega é:

<https://github.com/diogocezar/dctb-utfpr/tree/master/2017/programacao-web-2/tasks/phprpg/>

Deve-se criar uma pasta com o seguinte formado:

phprpg-grupo-x

Por exemplo:

phprpg-grupo-1

Os assets necessários com os avatares dos personagens estão em:

<https://github.com/diogocezar/dctb-utfpr/tree/master/2017/programacao-web-2/tasks/phprpg/assets/>

OBS. 1 Vou olhar quem fez os commits;

OBS 2. Já deixei as pastas criadas para a colocação dos arquivos.

| GRUPO | ALUNOS |
|----------|------------------------------------|
| 1 | ALAN NICOLAS LINS DE ALBUQUERQUE |
| | FABRICIO JHONATA SOUSA DE OLIVEIRA |
| | LUCAS JACINTO DA SILVA |
| 2 | ANDRESSA LEITE HENRIQUES |
| | LUA IVO MACHADO |
| | STEPHANIE LEAO FLORO DA SILVA |
| 3 | TIAGO PEREIRA COELHO |
| | VITOR STELMASCHUK SANTOS |
| | ANDERSON MARQUES SOARES |
| | CARLOS HENRIQUE DOS SANTOS |
| 4 | AMILTON FONTOURA DE CAMARGO JUNIOR |
| | GUILHERME TAVARES TEMPESTA |
| | TIAGO PAGOTTO |
| 5 | LUIZ GUILHERME DEVIDE SPIRITO |
| | RODRIGO MORETTO |
| | LUCAS FERNANDES COELHO |
| 6 | ROBERTO GUIMARAES JUNIOR |
| | JOAO PAULO BERNARDINO MACIEL |