

# Jogo da Forca C++



Alunos:

- **Josué Caleb Avelino dos Santos**
- **Tales Nobre Leite Dias de Oliveira**

Disciplina: Programação Orientada à Objetos

Professora: Adriana Carla Damasceno

# Tema do Projeto

Apresentação inicial da ideia do tema: Implementamos um jogo da forca, com objetivo de gerar uma interação lúdica do usuário com o programa, de modo que seja algo divertido para quem estiver jogando.

Parâmetros para o Jogo da Forca:

- Utiliza caracteres de letras para jogar.
- O usuário pode escolher jogar em 4 dificuldades, sendo elas: Fácil, Médio, Difícil e Customizado ao jogar com outra pessoa
- A dificuldade altera a quantidade de vidas (Fácil = 5 vidas, Médio = 3 vidas, Difícil = 1 vida e Custom o usuário escolhe de 0 até 5 vidas).
- Sorteio da palavra secreta (através de escolha aleatória no banco de palavras do nível).
- Exibe os caracteres errados do usuário e um boneco perdendo as partes do corpo a cada erro.
- Exibe os caracteres certos do usuário em suas posições exatas e as partes do corpo restantes no boneco.
- O usuário pode acrescentar uma nova palavra ao acertar a palavra secreta no seu nível.

# Implementação: hpp

```
1  #include <string>
2  #include <vector>
3
4  using namespace std;
5
6  class Arquivo {
7      public:
8          vector<string> palavras;
9          string nomeArquivo;
10
11          Arquivo();
12
13          void leArquivo(string arquivo);
14          void adicionaPalavra();
15          void salvaArquivo(vector<string> novaLista);
16  };
```

```
18  class Jogo : public Arquivo {
19      private:
20          string palavraSecreta;
21          map<char, bool> chutou;
22          vector<char> chutesErrados;
23
24      public:
25          int errosMax = 5;
26          Jogo(){};
27
28          void sorteiaPalavra();
29          void cabecalho();
30          void boneco();
31
32          bool existeLetra(char chute);
33          bool naoAcertou();
34          bool naoEnforcou();
35
36          void palavra();
37          void erros();
38          void chuta();
39
40          void dificuldade(int dificuldade);
41          void iniciarJogo();
42          void iniciarJogo(int vidas);
43          void iniciarJogo(string palavra);
44  };
```

Exceção



```
46  class ForcaException : public exception {
47      private:
48          char* erro;
49
50      public:
51          ForcaException(char* err) : erro(err) {}
52          char* what() {
53              return erro;
54          }
55  };
```

# Implementação: Iniciando o código



```
22  template <typename T>
23 ▼ void print(T t) {
24      cout << t << endl;
25  };
27  Arquivo::Arquivo() { print("Inicializando..."); };
28
29 ▼ void Arquivo::leArquivo(string nomeArquivo) {
30      ifstream arquivo;
31      this->nomeArquivo = nomeArquivo;
32 ▼   try {
33       arquivo.open(this->nomeArquivo);
34
35       int quantidadePalavras;
36       arquivo >> quantidadePalavras;
37
38       vector<string> palavrasDoArquivo;
39
40 ▼   for (int i = 0; i < quantidadePalavras; i++) {
41       string palavraLida;
42       arquivo >> palavraLida;
43       palavrasDoArquivo.push_back(palavraLida);
44   }
```

# Implementação: Cabeçalho

```
89 ▼ void Jogo::cabecalho() {
90     print("-----");
91     print("|                |");
92     print("|   JOGO DA FORCA   |");
93     print("|                |");
94     print("-----");
95     print("");
96 }
```



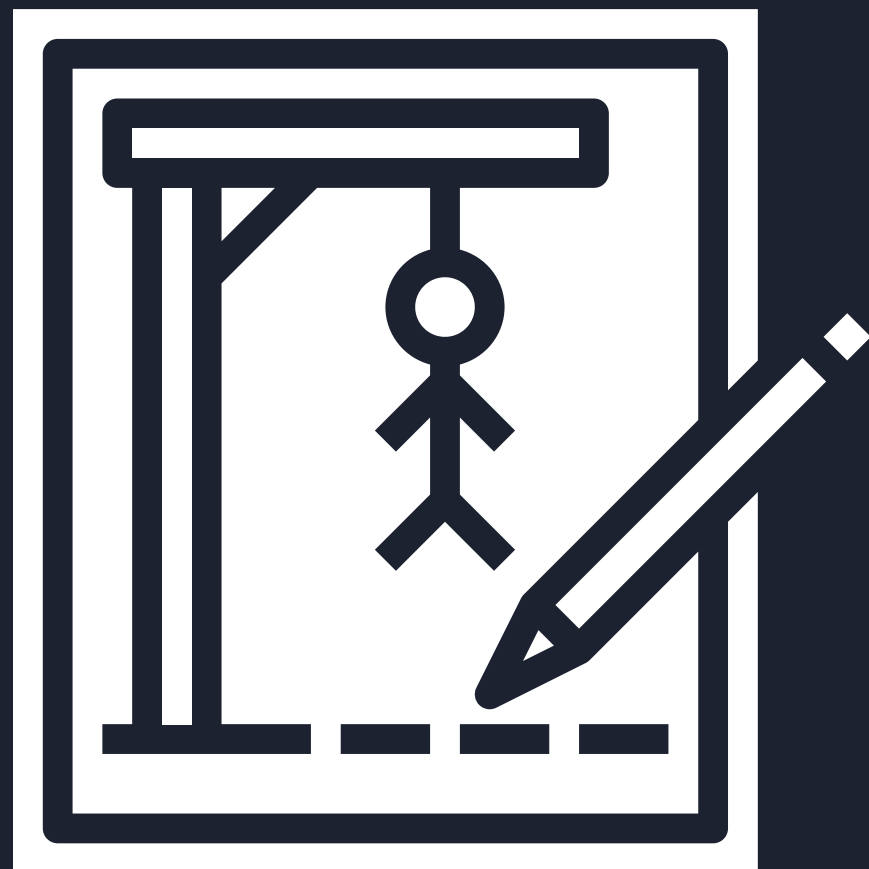
```
98 ▼ void Jogo::boneco() {
99     print("      _____      ");
100     print("|                |      ");
101     print("|                |      ");
102     print("|                |      ");
103     print("|                T      ");
104     print("|      ___|___      ");
105 ▼     if (Jogo::chutesErrados.size() == 0) {
106         print("|      / - - \\      ");
107         print("|      | U ^ U |      ");
108         print("|      \\ W /      ");
109         print("|                T      ");
110         print("|                |      ");
111         print("|              /|\\      ");
112         print("|             / | \\      ");
113         print("|            /  |  \\      ");
114         print("|                |      ");
115         print("|                |      ");
116         print("|             /  \\      ");
117         print("|            /    \\      ");
118         print("|                |      ");
119         print("|_      _|      _|      ");
```

# Implementação: Chutes

```
199 ▼ void Jogo::sorteiaPalavra() {
200     srand(time(NULL));
201     int indiceSorteado = rand() % this->palavras.size();
202
203     this->palavraSecreta = this->palavras[indiceSorteado];
204 };
205
206 ▼ bool Jogo::naoAcertou() {
207     for (char letra : this->palavraSecreta) {
208         if (!chutou[letra])
209             return true;
210     }
211     return false;
212 };
213
214 ▼ bool Jogo::naoEnforcou() {
215     return this->chutesErrados.size() < this->errosMax;
216 };
217
218 ▼ void Jogo::erros() {
219     cout << "Chutes errados: ";
220     for (char letra : this->chutesErrados) {
221         cout << letra << " ";
222     }
223     print("");
224 };
```

```
244 ▼ void Jogo::chuta() {
245     cout << "Seu chute: ";
246     char chute;
247     cin >> chute;
248     chute = toupper(chute);
249
250     chutou[chute] = true;
251
252     if (this->existeLetra(chute)) {
253         print("Você acertou! ;)");
254         print("Essa letra está na palavra!");
255     } else {
256         print("Você errou! :(");
257         print("Essa letra não está na palavra.");
258         this->chutesErrados.push_back(chute);
259     }
260     print("");
261 };
```

# Implementação: Output



```
274 print("Fim de jogo!");
275 print("A palavra secreta era: " + this->palavraSecreta);
276 if (this->naoAcertou())
277     print("Você perdeu! Tente novamente");
278 else {
279     print("Parabéns! :) ");
280     print("Você acertou a palavra!");
281     bool fim;
282     while (!fim) {
283         cout << "Você deseja adicionar uma nova palavra a lista? (S/N) ";
284         char resposta;
285         cin >> resposta;
286         resposta = toupper(resposta);
287
288         if (resposta == 'S') {
289             this->adicionaPalavra();
290             fim = true;
291         } else if (resposta == 'N') {
292             print("Obrigado por Jogar!");
293             fim = true;
294         } else {
295             print("Resposta inválida!");
296         }
297     }
298     fim = false;
299 }
300 };
```

# Implementação: Níveis

```
void Jogo::dificuldade(int dificuldade) {
    if (dificuldade == 0) {
        this->leArquivo("facil.txt");
        print("-----");
        print("|      F Á C I L      |");
        this->iniciarJogo();
    } else if (dificuldade == 1) {
        this->leArquivo("medio.txt");
        print("-----");
        print("|      M É D I O      |");
        this->iniciarJogo(3);
    } else if (dificuldade == 2) {
        this->leArquivo("dificil.txt");
        print("-----");
        print("|    D I F Í C I L    |");
        this->iniciarJogo(1);
    } else if (dificuldade == 3) {
        int vidas;
        string palavra;
        print("Quantas vidas você gostaria de ter?");
        print("(0 a 5)");
        cin >> vidas;
        if (!(0 <= vidas && vidas <= 5)) throw ForcaException("Valor inserido fora da faixa permitida!");
        this->errosMax = vidas;
        print("Qual palavra você quer usar?");
        cin >> palavra;
        print("-----");
        print("|    C U S T O M    |");
        this->iniciarJogo(palavra);
    } else {
        throw ForcaException("Dificuldade não existente!");
    }
};
```

```
int main() {
    Jogo jogo;
    int dif;
    print("Qual dificuldade você deseja jogar?");
    print("0 - Fácil    1 - Médio    2 - Difícil    3 - Custom");
    cin >> dif;

    try {
        jogo.dificuldade(dif);
    } catch (ForcaException ex) {
        cout << ex.what() << endl;
        jogo.dificuldade(0);
    }

    return 0;
};
```



# Implementação: Jogo

Inicializando...

Qual dificuldade você deseja jogar?

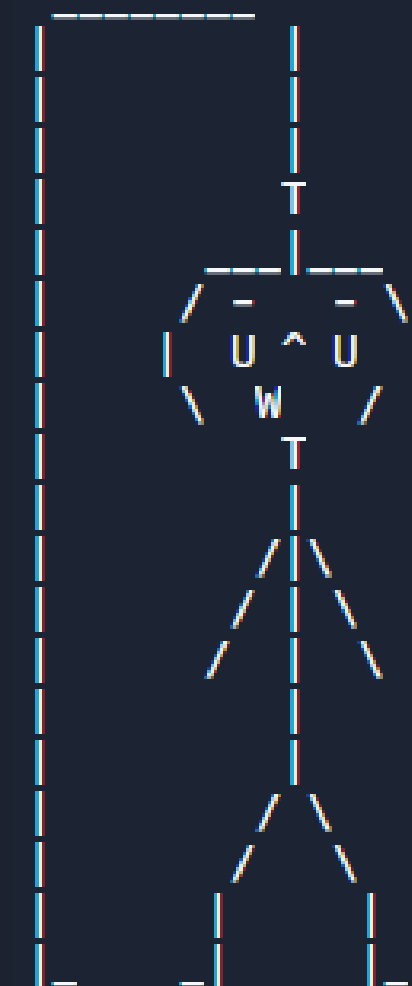
0 - Fácil    1 - Médio    2 - Difícil    3 - Custom  
0

```
-----  
|   F Á C I L   |  
-----  
|   JOGO DA FORCA   |  
-----
```

Chutes errados:

Seu chute: █

Seu chute: T  
Você acertou! ;)   
Essa letra está na palavra!



Chutes errados:  
T \_ T \_  
Seu chute: █

Essa letra está na palavra!



Fim de jogo!  
A palavra secreta era: TESTE  
Parabéns! :)  
Você acertou a palavra!  
Você deseja adicionar uma nova palavra a lista? (S/N) N  
Obrigado por Jogar!

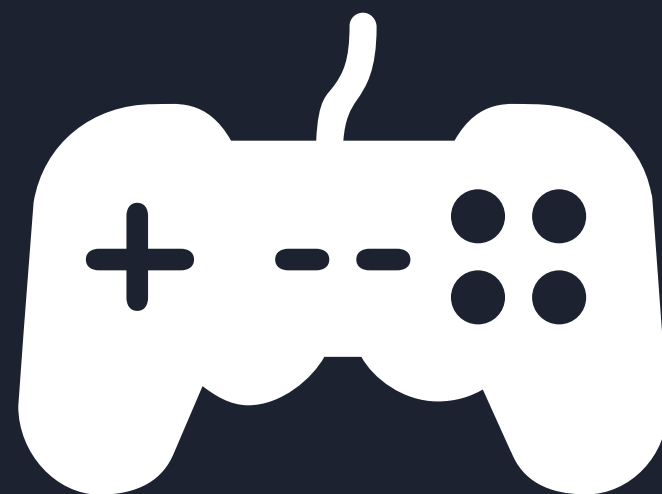
# Implementação: Avançando de Nível

```
Inicializando...
Qual dificuldade você deseja jogar?
0 - Fácil    1 - Médio    2 - Difícil    3 - Custom
1
```

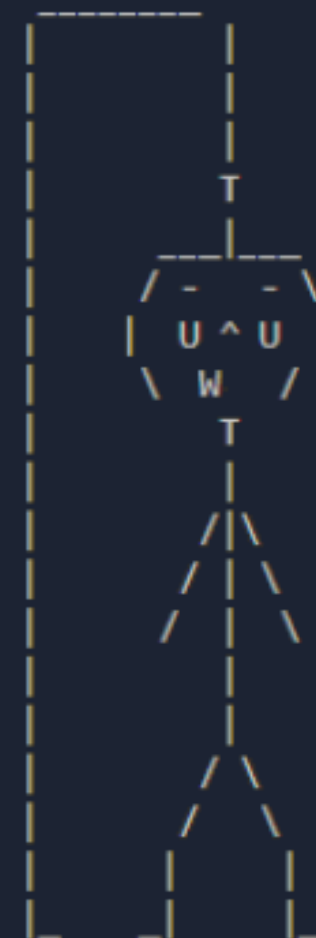
```
-----
|      M É D I O      |
|-----|
|      JOGO DA FORCA  |
|-----|
|-----|
```

Chutes errados:

Seu chute: █



Essa letra esta na palavra!

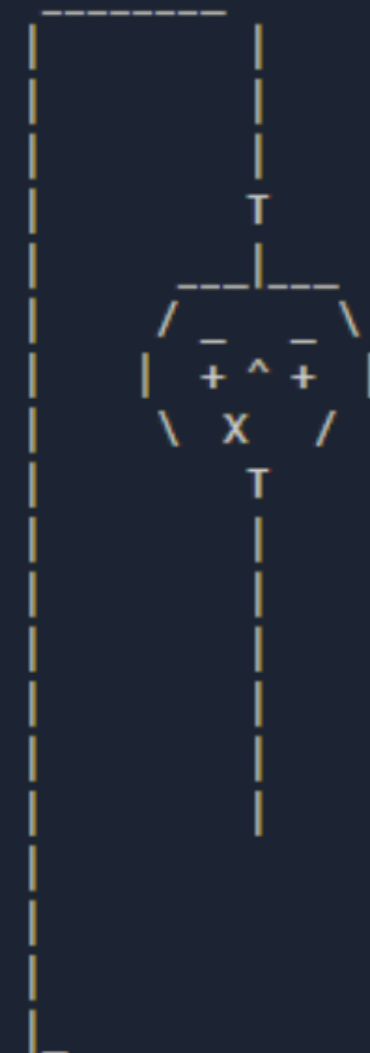


```
Fim de jogo!
A palavra secreta era: ADRIANA
Parabens! :)
Voc|~ acertou a palavra!
Voce deseja adicionar uma nova palavra a lista? (S/N) s
Digite a nova palavra
URUGUAI
```

# Implementação:

GAME  
OVER

```
O N O _ _ T O _ _ _ _  
Seu chute: g  
Voce errou! :(  
Essa letra nao esta na palavra.
```



```
Fim de jogo!  
A palavra secreta era: ONOMATOPEIA  
Voc~ perdeu! Tente novamente
```

# Erros ao longo da implementação

- No início do jogo agora aparece o nível de dificuldade que o usuário deseja jogar, pensamos bastante para fazer isso, na parte anterior do projeto os níveis eram automáticos (na ordem fácil, médio e difícil).
- Tivemos algumas dificuldades no polimorfismo, nossa primeira ideia não foi boa, mas enfim surgiu uma que encaixou com o que queríamos no jogo.

# Diagrama de Classes

