

**Universidade Federal do Piauí**  
**Campus Senador Helvídio Nunes de Barros**  
**Curso: Sistemas de Informação - Período: 1**  
**Disciplina: Algoritmos e Programação 1**  
**Professor(a): Alcilene Dalília De Sousa.**  
**Acadêmicos: Jederilson Sousa Luz e**  
**Mary Cynthia de Sousa.**

## **Trabalho Final**

**Jogo:**

# **Kouhai no Kimyou na Bouken**

**As bizarras aventuras de um novato**

Projeto Final como requisito para a 4ª avaliação da Disciplina Algoritmos e Programação 1 do(a) Professor(a) ALCILENE DALÍLIA DE SOUSA do curso de SISTEMAS DE INFORMAÇÃO do I PERÍODO da UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI, onde foi proposta a criação de um programa com um game do estilo Role-Playing Game (RPG).

## SUMÁRIO:

1. Bibliotecas.....	4
2. Structs.....	5
3. Funções.....	6
4. Funcionamento, Entradas e Saída de dados.....	10

## Bibliotecas:

1. **#include <stdio.h>**: Possui definições de sub-rotinas relativas às operações de entrada/saída, como leitura de dados digitados no teclado e exibição de informações na tela do programa de computador. Também possui numerosas definições de constantes, variáveis e tipos. É um dos cabeçalhos mais populares da linguagem de programação C, intensivamente utilizado tanto por programadores iniciantes como por experientes.

2. **#include <stdlib.h>**: É um arquivo cabeçalho da biblioteca de propósito geral padrão da linguagem de programação C. Ela possui funções envolvendo alocação de memória, controle de processos, conversões e outras. Ela é compatível com C++ e é chamada `cstdlib` em C++. O nome "stdlib" vem de standard library ("standard library" é "biblioteca padrão em inglês").

3. **#include <string.h>**: É um arquivo cabeçalho que fornece funções, macros e definições da biblioteca padrão da linguagem de programação C para manipulação de cadeias de caracteres e regiões de memória. Apesar de fornecer funções portáteis entre plataformas, sabidamente há problemas de segurança que expõe os programas a problemas de transbordamento.

4. **#include <time.h>**: É um arquivo cabeçalho que fornece protótipos para funções, macro e definição de tipos da biblioteca padrão da linguagem de programação C para manipulação de datas e horários de modo padrão.

5. **#include <stdbool.h>**: É um arquivo cabeçalho da biblioteca padrão da linguagem de programação C. É usada para manipular variáveis lógicas, como verdadeiro e falso. Pode ser substituída pela utilização de valores inteiro, sua função é simplesmente facilitar a compreensão do código.

## Structs:

1. **struct Personagem:** Define todos os parâmetros dos personagens criados pelo usuário. Nessa **struct** existem as seguintes variáveis:

**Int:** id, vida, vidaMax, mana, manaMax, nível, EXP (experiência, pontuação que o jogador acumula para passar de nível), EXPMax (guarda o valor de EXP necessário para passar de nível), pontuacao, pontuacaoMax;

**Char:** nome, classe, skills (vetor de string que guarda as habilidades do personagem);

**Float:** forcaFisica, forcaMagica, sorte, agilidade, resistenciaMagica, armadura;

2. **struct Inimigo:** Define todos os parâmetros dos inimigos que o jogador poderá enfrentar. Nessa **struct** existem as seguintes variáveis:

**Int:** id, vida, vidaMax, mana (tipo de energia), manaMax (tipo de energia maxima), nível, EXP (experiência, pontuação que o jogador ganha para passar de nível), ponto;

**Char:** nome (nome do personagem, com espaço para string de tamanho 20);

**Float:** forcaFisica, forcaMagica, sorte, agilidade, resistenciaMagica, armadura;

3. **structs Tabela:** Utilizada para simular uma tabela para facilitar o controle dos personagens. Nessa **struct** existem as seguintes variáveis:

**Personagem:** personagem(vetor com tamanho de string 9);

**Int:** tamanho (essa variável sempre guardará a quantidade de personagens dentro do vetor de personagens);

4. **struct TabelaInimigo:** utilizada para simular uma tabela para facilitar o controle dos inimigos. Nessa **struct** existem as seguintes variáveis:

**Inimigo:** inimigo;

5. **struct Retorno:** Utilizada para múltiplo retorno após o uso de uma skill. Dessa forma será possível retornar as alterações em vários chars ao mesmo tempo através das funções. Nessa **struct** existem as seguintes variáveis:.

**Inimigo:** inimigo, inimigoImagem;

**Personagem:** personagem, personagemImagem;

obs: As “imagens” são uma forma de alterar os status dos inimigos e personagens durante as lutas sem modificá-las diretamente, pois uma vez que os fossem modificados não haveria uma forma eficiente de reverter a alteração. Por exemplo, após o uso de uma habilidade do Guerreiro que aumenta ao máximo suas resistências (Castle of Stone), não seria possível retornar suas resistências para os valores normais no próximo turno. Ao fim da luta os status das imagens sempre retornam a zero.

## **Funções:**

### **1. int escudo (char entrada[2], char msgErro[100]);**

Essa função será usada para evitar entradas indesejadas no programa. Todas as entrada ao longo do código utilizam esta função.

**funcionamento:** Primeiro a função confere se a entrada tem apenas um char (o jogo segue esse padrão de entrada). Caso seja maior, será exibida a msgErro (mensagem de erro) definida no parâmetro e uma nova entrada será requisitada. Caso a entrada tenha um char apenas, será verificado se o valor é '0', caso seja a função retorna 0, se não, a função atoi irá converter o char. Caso o valor após o uso do atoi seja 0, significa que o usuário não digitou um número, então a função retornará -1 para que o valor caia no default dos switchs das funções que usam o escudo.

**retorno:** Após o tratamento da entrada, a função retorna o valor digitado pelo usuário.

### **-> Sessão das funções que manipulam a campanha dentro da dungeon:**

### **2. void start ();**

Essa função pede ao jogador que identifique qual personagem durante a campanha.

**funcionamento:** Pedirá ao usuário que identifique qual personagem deseja usar pelo id, então lança o jogador para a dungeon. No fim da aventura essa função modifica os status dos personagens para consolidar as conquistas.

**retorno:** Função de retorno vazio.

### 3. **Personagem dungeon (Personagem personagem);**

Função que simula a aventura dentro da dungeon.

**funcionamento:** Essa função seta os personagens que serão utilizados durante a luta. Os inimigos serão escolhidos aleatoriamente, e as lutas ocorrem em sucessão até que a vida do personagem chegue a zero. Salva a pontuação do jogador caso seja maior que a anterior.

**retorno:** Retorna o personagem modificado no fim da função.

### ->**Sessão de funções do sistema de batalha:**

### 4. **Personagem batalha (Personagem personagem, Inimigo inimigo);**

Essa função gerencia a mecânica de batalha do jogo.

**funcionamento:** Através de um grande switch, o jogador pode optar por Atacar, Defender, usar Habilidade ou Fugir. Atacar causará dano ao oponente, Defender aumentará as resistências do personagem para que receba menos dano do inimigo, Fugir encerra a luta e Habilidade abre um pequeno menu que permite utilizar uma de suas habilidades, que possuem diversos efeitos variados. Após a escolha do jogador a ação correspondente será efetuado e logo em seguida o inimigo realizará seu ataque. A luta continua até que um dos dois seja derrotado.

**retorno:** Retorna o personagem modificado no fim da função.

### 5. **Retorno skills(Personagem personagem, Personagem imagemP, Inimigo inimigo, Inimigo imagemI, char skill[15]);**

Essa função definirá as habilidades do personagem.

**funcionamento:** Quando o jogador utilizar uma habilidade, essa função irá identificar qual sua classe e qual habilidade está sendo usada. Então os efeitos da habilidade escolhida serão aplicados. Caso a habilidade não seja encontrada uma mensagem dirá ao jogador que o personagem não fez nada e perdeu o turno.

**retorno:** Retorna uma variável do tipo Retorno que contém o personagem, o inimigo e suas imagens.

### ->**Sessão das funções que manipulam o CRUD :**

#### 6. **void save ();**

Guarda um novo personagem criado pelo jogador na tabela.

**funcionamento:** O jogador pode escolher um nome e uma classe para o personagem. Então o novo personagem é alocado na tabela de personagens (tab) na posição igual ao atributo tamanho da tabela. Após a criação do personagem o tamanho da tabela aumenta em 1.

**retorno:** Função de retorno vazio.

#### 7. **void read ();**

Exibe todos os personagens criados.

**funcionamento:** Percorre o vetor de personagens na tabela e apresenta ao jogador com formatação.

**retorno:** Função de retorno vazio.

#### 8. **void del ();**

Remove da tabela o personagem escolhido pelo jogador.

**funcionamento:** Recebe um id identificando qual personagem deve ser removido. Então realoca todos os personagens dessa posição em diante no vetor para uma posição atrás, sobrescrevendo a referência do personagem apagado. No fim da função o tamanho da tabela recebe diminui em 1.

**retorno:** Função de retorno vazio.

#### 9. **void update ();**

Permite que o jogador mude o nome de um personagem.

**funcionamento:** Recebe um id identificando qual personagem deve ser modificado. Caso o id seja válido, o jogador poderá alterar o nome do personagem.

**retorno:** Função de retorno vazio.

**-> Sessão das funções de definição dos atributos de personagens, inimigos e skills:**



**10. Personagem incrementoEXP (Personagem personagem, int exp);**

Adiciona os pontos de EXP ganhos pelo jogador ao personagem.

**funcionamento:** Essa função faz o acréscimo sempre que o personagem ganha EXP. Caso seja o suficiente, a função de aumento de nível será chamada. Caso o EXP seja mais que suficiente para passar de nível, a sobra já será adicionada ao EXP para o próximo nível.

**retorno:** Retorna o personagem com as modificações de EXP e Nível.

**11. Personagem definirAtributos (Personagem personagem, char classe[15]);**

Essa função atribui a um personagem criado atributos base de acordo com sua classe.

**funcionamento:** Através do atributo *classe* do personagem a função irá atribuir valores para seus parâmetros adequados para sua classe. Os slots de habilidades serão todos instanciados como “vazio”.

**retorno:** Retorna o personagem com todas as modificações de parâmetros.

**12. Personagem subirNivel (Personagem personagem);**

Essa função aplica o aumento de nível do personagem.

**funcionamento:** Aumenta o nível do personagem em 1. Cada classe possui uma curva de evolução diferente. Essa função aumenta os status do personagem de acordo com sua classe quando o mesmo sobe o nível. Os parâmetros aumentarão de acordo com uma porcentagem do valor atual.

**retorno:** Retorna o personagem com todas as modificações de parâmetros

**13. void definirInimigos();**

Essa função cria os inimigos.

**funcionamento:** Vários inimigos são instanciados diretamente na tabela de inimigos.

**retorno:** Função de retorno vazio.

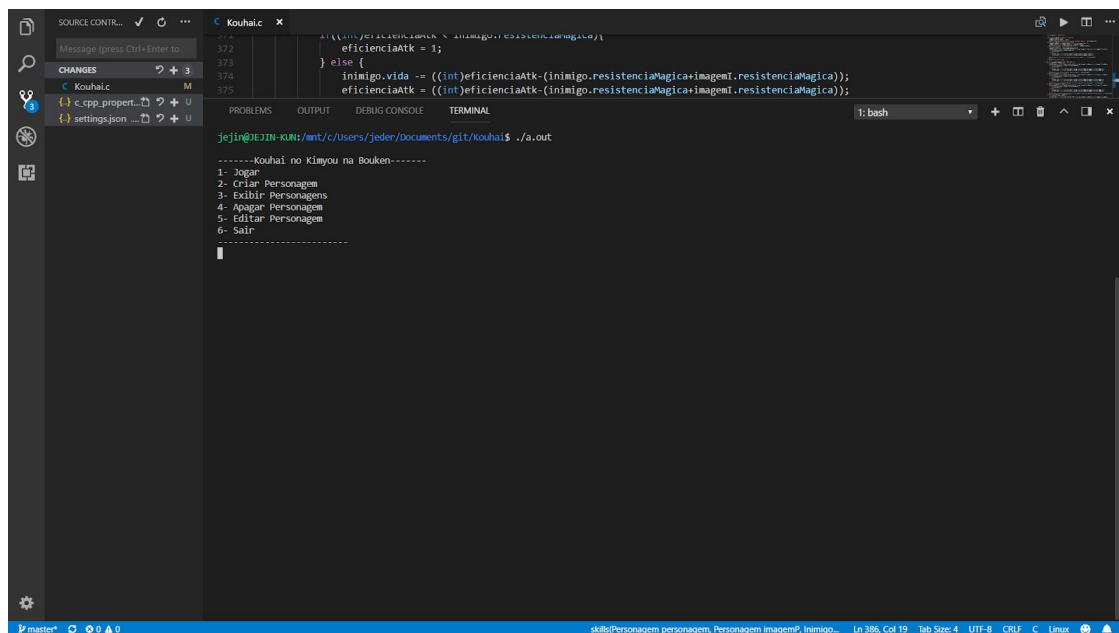
-> **Sessão menu do jogo:**

14. `int main ();`

Função principal, onde haverá as exibições para o usuário e o menu do jogo. Inicia-se as entradas e saídas de dados do programa.

## Funcionamento, Entrada e saída de dados do jogo:

Depois de rodar o programa, será exibido para o usuário um menu com o nome do jogo e as seguintes opções:



The screenshot shows a Visual Studio Code editor with a C++ file named `Kouhaic`. The code defines a function `eficienciaAtk` and a `main` function. The `main` function calls `eficienciaAtk` and then displays a menu in the terminal. The terminal output shows the program's execution, including the menu options: 1- Jogar, 2- Criar Personagem, 3- Exibir Personagens, 4- Apagar Personagem, 5- Editar Personagem, and 6- Sair.

```
372         eficienciaAtk = inimigo.resistenciaMagica;
373     } else {
374         inimigo.vida -= ((int)eficienciaAtk-(inimigo.resistenciaMagica+imagem.resistenciaMagica));
375         eficienciaAtk = ((int)eficienciaAtk-(inimigo.resistenciaMagica+imagem.resistenciaMagica));

jeff@JEJIN-KUN:/mnt/c/Users/jeffr/documents/git/kouhai$ ./a.out
-----Kouhai no Kimyou na Bouken-----
1- Jogar
2- Criar Personagem
3- Exibir Personagens
4- Apagar Personagem
5- Editar Personagem
6- Sair
|
```

**1. Jogar:** Será chamada a função start e se iniciará a campanha.

[illegible]

Durante a campanha vários inimigos irão aparecer em sequência para desafiar o jogador. A cada inimigo derrotado o jogador ganha 10 pontos para o ranking. A campanha termina quando o jogador perder todos os pontos de vida.

The screenshot displays the Visual Studio Code interface with a C++ project open. The left sidebar shows the Explorer view with files like 'c\_cpp\_properties.json' and 'settings.json'. The main editor area shows the source code of 'Kouhaic.cpp', which includes a header guard, namespace 'Kouhaic', and a function 'main' that prints a message about a character's attack. The bottom panel shows the integrated terminal running a bash shell, displaying the program's execution output.

```
#include <iostream>
using namespace std;

namespace Kouhaic {
    void main() {
        printf("Voce atravesa seu oponente com sua espada envolta de eletricidade numa grande velocidade, infligindo %d de dano.", eficiencia);
    }
}
```

Terminal Output:

```
t: bash
-----Batalha-----
Vida: 28/28      Mana: 18/18

Inimigo Vida: 25/25      Mana: 25/25

1- Atacar
2- Defender
3- Habilidades
4- Fugir
-----Turno 1-----
3

-----
[1. Speed Shot] [2. Aim: WeakPoint]
[3. Vazio] [4. Vazio]

Qual Skill deseja usar? 2
Inimigo recebeu 17 de dano.
Voce recebeu 18 de dano.

-----Batalha-----
Vida: 18/28      Mana: 18/18

Inimigo Vida: 8/25      Mana: 25/25

1- Atacar
2- Defender
3- Habilidades
4- Fugir
-----Turno 2-----
3

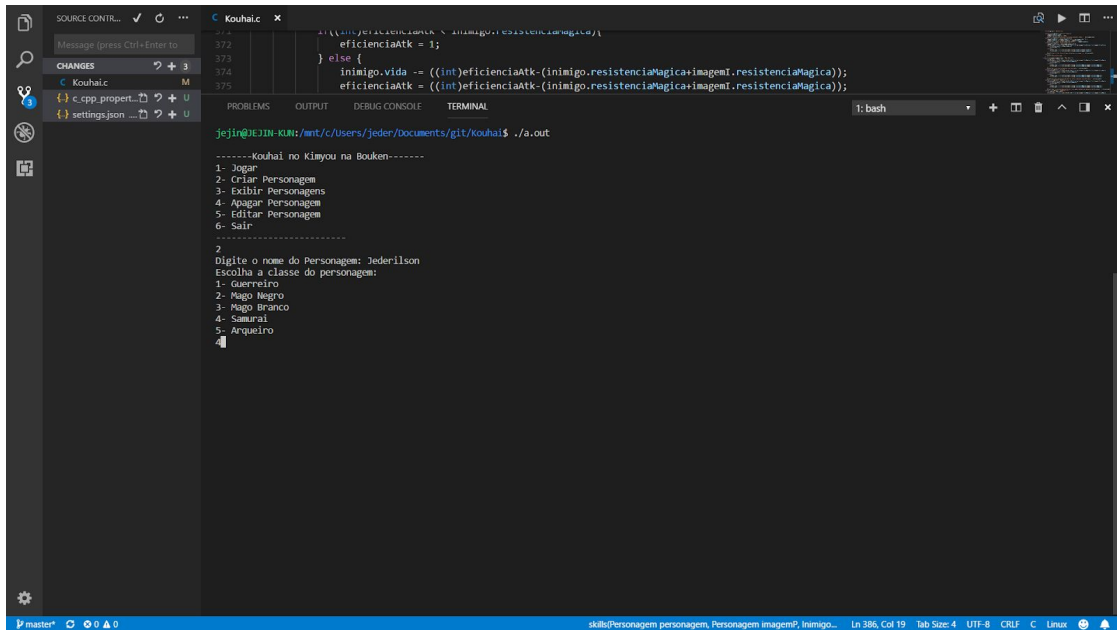
-----
[1. Speed Shot] [2. Aim: WeakPoint]
[3. Vazio] [4. Vazio]

Qual Skill deseja usar? 1
Voce se prepara para o combo...
1- ATQUESE!
Inimigo recebeu 4 de dano.
Descanso. Voce recebeu 13 de dano.

A campanha encorru. Sua pontuacao foi 0.
```

No menu de batalha o jogador pode escolher entre **Atacar** o oponente, se **Defender** para diminuir o dano que o oponente causar no turno, **Fugir** para abandonar a luta e encerrar a campanha e **Habilidades** para escolher uma das habilidades de seu personagem. Cada classe tem suas próprias habilidades com efeitos variados que irão ajudá-lo a derrotar seus inimigos.

2. **Criar Personagem:** Será chamada a função **save** para que seja criado um personagem.



```
372         if (eficienciaAtk < inimigo.resistenciaMagica){
373             eficienciaAtk = 1;
374         } else {
375             inimigo.vida -= ((int)eficienciaAtk-(inimigo.resistenciaMagica+imagemI.resistenciaMagica));
376             eficienciaAtk = ((int)eficienciaAtk-(inimigo.resistenciaMagica+imagemI.resistenciaMagica));
377         }
```

jejin@JEJIN-KIRI:/mnt/c/Users/Jeder/documents/git/kouhai\$ ./a.out

-----Kouhai no Kimyou na Bouken-----

1- Jogar

2- Criar Personagem

3- Exibir Personagens

4- Apagar Personagem

5- Editar Personagem

6- Sair

-----

2

Digite o nome do Personagem: Jederalson

Escolha a classe do personagem:

1- Guerreiro

2- Mago Negro

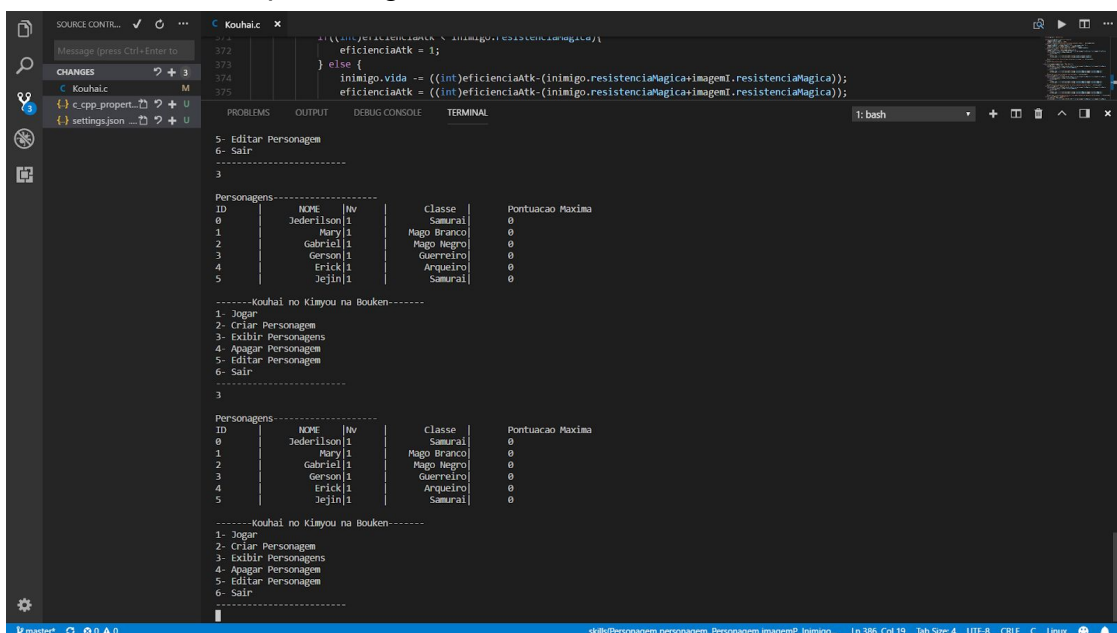
3- Mago Branco

4- Samurai

5- Arqueiro

6

3. **Exibir Personagem:** Será chamada a função **read** para que sejam exibidos os personagem.



```
5- Editar Personagem
6- Sair
-----
3
Personagens-----
ID      NOPE  Inv      Classe      Pontuacao Maxima
0      Jederalson 1      Samurai      0
1      Maryl      1      Mago Branco  0
2      Gabriel    1      Mago Negro   0
3      Gerson     1      Guerreiro    0
4      Erick      1      Arqueiro     0
5      Jejin      1      Samurai      0
-----
-----Kouhai no Kimyou na Bouken-----
1- Jogar
2- Criar Personagem
3- Exibir Personagens
4- Apagar Personagem
5- Editar Personagem
6- Sair
-----
3
Personagens-----
ID      NOPE  Inv      Classe      Pontuacao Maxima
0      Jederalson 1      Samurai      0
1      Maryl      1      Mago Branco  0
2      Gabriel    1      Mago Negro   0
3      Gerson     1      Guerreiro    0
4      Erick      1      Arqueiro     0
5      Jejin      1      Samurai      0
-----
-----Kouhai no Kimyou na Bouken-----
1- Jogar
2- Criar Personagem
3- Exibir Personagens
4- Apagar Personagem
5- Editar Personagem
6- Sair
-----
```

4. **Apagar Personagem:** Será chamada a função **del** para que seja apagado um personagem.

```

372         eficienciaAtk = 1;
373     } else {
374         inimigo.vida -= (((int)eficienciaAtk-(inimigo.resistenciaMagica+imagemI.resistenciaMagica));
375         eficienciaAtk = (((int)eficienciaAtk-(inimigo.resistenciaMagica+imagemI.resistenciaMagica));

```

6- Sair

-----Kouhai no Klayou na Bouken-----

Personagens-----

ID	NOME	Inv	Classe	Pontuacao Maxima
0	Jederilson	1	Samurai	0
1	Mary	1	Mago Branco	0
2	Gabriel	1	Mago Negro	0
3	Gerson	1	Guerreiro	0
4	Erick	1	Arqueiro	0
5	Jejin	1	Samurai	0

Qual o ID do item que deseja apagar?1

Personagem Apagado.

-----Kouhai no Klayou na Bouken-----

1- Jogar

2- Criar Personagem

3- Exibir Personagens

4- Apagar Personagem

5- Editar Personagem

6- Sair

-----Kouhai no Klayou na Bouken-----

Personagens-----

ID	NOME	Inv	Classe	Pontuacao Maxima
0	Mary	1	Mago Branco	0
1	Gabriel	1	Mago Negro	0
2	Gerson	1	Guerreiro	0
3	Erick	1	Arqueiro	0
4	Jejin	1	Samurai	0

-----Kouhai no Klayou na Bouken-----

1- Jogar

2- Criar Personagem

3- Exibir Personagens

4- Apagar Personagem

5- Editar Personagem

6- Sair

5. **Editar Personagem:** Será chamada a função **update** para que seja editado um personagem.

```

372         eficienciaAtk = 1;
373     } else {
374         inimigo.vida -= (((int)eficienciaAtk-(inimigo.resistenciaMagica+imagemI.resistenciaMagica));
375         eficienciaAtk = (((int)eficienciaAtk-(inimigo.resistenciaMagica+imagemI.resistenciaMagica));

```

-----Kouhai no Klayou na Bouken-----

1- Jogar

2- Criar Personagem

3- Exibir Personagens

4- Apagar Personagem

5- Editar Personagem

6- Sair

-----Kouhai no Klayou na Bouken-----

Personagens-----

ID	NOME	Inv	Classe	Pontuacao Maxima
0	Jederilson	1	Samurai	0

Qual o ID do Personagem que deseja atualizar?0

Qual o novo nome?Mary

Personagem Atualizado.

-----Kouhai no Klayou na Bouken-----

1- Jogar

2- Criar Personagem

3- Exibir Personagens

4- Apagar Personagem

5- Editar Personagem

6- Sair

-----Kouhai no Klayou na Bouken-----

Personagens-----

ID	NOME	Inv	Classe	Pontuacao Maxima
0	Mary	1	Samurai	0

Qual o ID do Personagem que deseja atualizar?0

Qual o novo nome?Alcilene

Personagem Atualizado.

-----Kouhai no Klayou na Bouken-----

1- Jogar

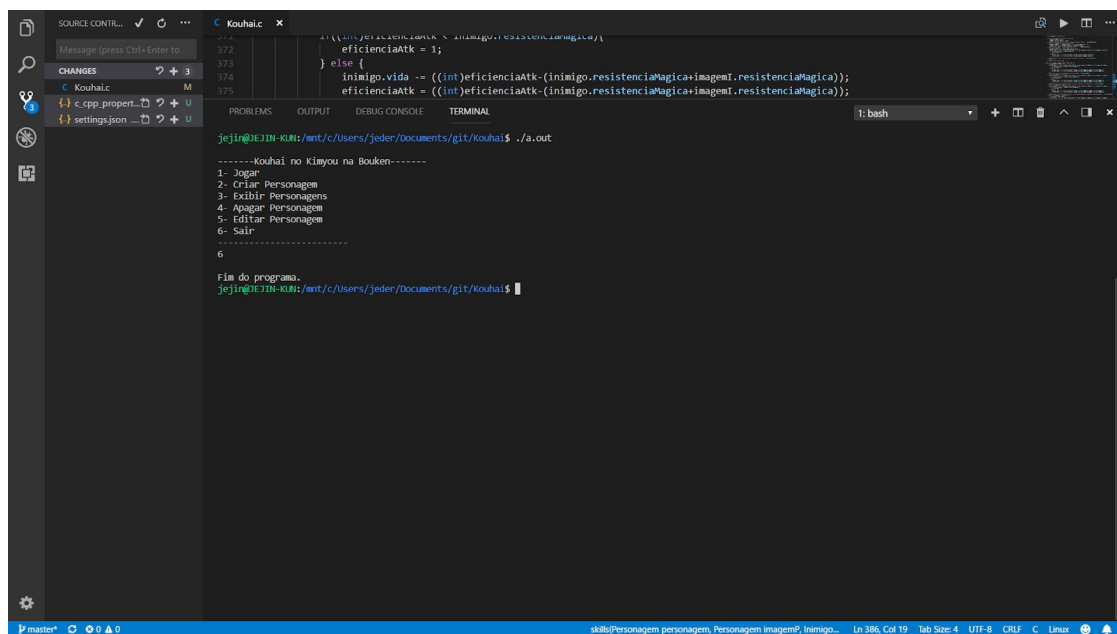
2- Criar Personagem

3- Exibir Personagens

4- Apagar Personagem

5- Editar Personagem

6. **Sair:** O programa será encerrado e será exibido ao usuário a frase “Fim do programa” .



Caso o usuário digite algo que não corresponda às opções do menu, será exibido ao usuário a frase “Escolha uma opção válida!”.

O programa repetirá o menu até que seja digitado a opção **6. Sair**. Então o programa irá encerrar.