```
.data
# File
recebearq:      .asciiz "Informe o nome do arquivo Ex:palavras.txt.\n"
erroarq:        .asciiz "Arquivo não existe. Tente Novamente.\n"
nomearq:        .asciiz ""          # Nome do arquivo que contem as palavras
tamentrada:     .word   1024    # Tamanho maximo do nome do arquivo
Bufferarq:      .space  1024    # Reserva de 1024 bytes para o buffer do arquivo

Palavra:        .space  24      # A Palavra

# Figura
forca:  .asciiz "_______\n|   |  \\|\n       |\n       |\n       |  ",
                "\n       |\n       |\n       |\n      ---\n",
                "_______\n|   |  \\|\n    O   |\n       |\n       |  ",
                "\n       |\n       |\n       |\n      ---\n",
                "_______\n|   |  \\|\n    O   |\n    |  |\n       |  ",
                "\n       |\n       |\n       |\n      ---\n",
                "_______\n|   |  \\|\n    O   |\n    |  |\n    |  |  ",
                "\n       |\n       |\n       |\n      ---\n",
                "_______\n|   |  \\|\n    O   |\n   \\|  |\n    |  |  ",
                "\n       |\n       |\n       |\n      ---\n",
                "_______\n|   |  \\|\n    O   |\n   \\|/  |\n    |  |  ",
                "\n       |\n       |\n       |\n      ---\n",
                "_______\n|   |  \\|\n    O   |\n   \\|/  |\n    |  |  ",
                "\n   /   |\n       |\n       |\n      ---\n",
                "_______\n|   |  \\|\n    O   |\n   \\|/  |\n    |  |  ",
                "\n   / \\  |\n       |\n       |\n      ---\n",

limpatela:      .asciiz "\n\n\n\n\n\n\n\n\n\n\n\n"

# Strings
msgini:         .asciiz "Bem-Vindo ao jogo da forca!\n"
padrao:         .asciiz  " "
Sim:    .asciiz "\nSim!\n"
No:     .asciiz "\nNão!\n"
ja:     .asciiz "\nVocê já tentou esta letra.\n"


escolha:        .asciiz "Escolha uma letra.\n"
teclainvalida:  .asciiz "Tecla invalida. Insira apenas caracteres alfabeticos.\n"

palavracerta:   .asciiz "A palavra correta é: "

Perdeu:                 .asciiz "Você Perdeu!\n"
venceu:                 .asciiz "Você Venceu!\n"
pont:           .asciiz "Você tem "
pont2:          .asciiz " acerto(s) consecutivo(s).\n"
pont3:          .asciiz "\nVocê fez "

jogadenovo:     .asciiz "\nGostaria de Jogar Novamente? (s/n)\n"

msgtchau:       .asciiz "Obrigado por Jogar. Esperamos que tenha se divertido!"
```

```
jaacertou:      .asciiz  "Todas as tentativas: "

adivinhou:      .space  26      # Tentativa

situacaoatual:  .space  26      #s _ s t e _ d

###############################################################################
# Equivalencias
.eqv servico_termina_exec    10
.eqv servico_abre_arquivo    13
.eqv servico_le_arquivo      14
.eqv servico_fecha_arquivo   16
.eqv servico_msg_dialogo     55

.eqv sinaliza_abre_leitura          0
.eqv sinaliza_abre_escrita          1
.eqv sinaliza_abre_anexa_escrita    9

###############################################################################
# Inicia programa
.text

main:
        # Escreve bem vindo
        li      $v0, servico_msg_dialogo
        la      $a0, msgini
        syscall

        # Abre o arquivo
        jal     abrearq

# Roda o jogo
iniciajogo:
        jal     GeraRandom
        la      $a0, adivinhou              # Nova tentativa
        jal     Underline
        la      $a0, situacaoatual
        jal     Underline
        li      $s0, 0              # $s0 irá conter o numero de jogadas.
        li      $s6, 0              # $s6 irá conter o numero de tentativas corretas

        li      $t0, '!'        # Set '!' como t0
        la      $t1, adivinhou      # Carrega a tentativa em t1
        sb      $t0, 0($t1)         # Salva '!' na tentativa

        # Mostra a forca vazia e os espaços em branco
        jal     limpaterminal       # Limpa o terminal
        la      $a1, Palavra
        la      $a0, adivinhou
        la      $a3, situacaoatual
        jal     gerapalavratela             # Retorna todos os espaços em branco
```

```
        jal     mostraforca             # Desenha a forca em branco

        # Roda o jogo
        jal     rodajogo

Underline:
        li      $t1, 0                  # $t1 é o contador
        move    $t0, $a0
LoopUnder:
        li      $t2, 0x0
        sb      $t2, 0($t0)
        addi    $t0, $t0, 1             # Move um palavra
        addi    $t1, $t1, 1             # Incrementa o contador
        bne     $t1, 26, LoopUnder
        jr      $ra

# Arquivo não existe
msgerroarq:
        li      $v0, 4                  # 4 é o codigo da função para printar uma string
        la      $a0, erroarq            # Carrega a string em a0
        syscall                         # Printa a mensagem

#-------[ Subrotinas de acesso ao arquivo ]-----------------------------------------------
#       Pega o nome do arquivo dicionário do usuario
#       lê o arquivo para o buffer

abrearq:
getnomearq:                             # Pega o endereço do arquivo pelo usuario
        # Mostra prompt
        li      $v0, 4                  # 4 é o codigo da função para printar uma string
        la      $a0, recebearq          # Carrega a string em a0
        syscall                         # Printa a mensagem

        # Pega entrada usuario
        li      $v0, 8                  # 8 é o codigo da função para ler um string
        la      $a0, nomearq            # Carrega o nome do arquivo em a0
        lw      $a1, tamentrada             # Carrega o valor do tamanho maximo a1
        syscall                         # Entrada agora salva no nomearq

# Remove o \n da entrada do usuario
corrigeinput:                           # Corrige o input
        li      $t0, 0                  # Loop de contagem
        lw      $t1, tamentrada             # Final do loop
clean:
        beq     $t0, $t1, abreleitura
        lb      $t3, nomearq($t0)
        bne     $t3, 0x0a, Incrementat0     # Não estamos na nova linha
        sb      $zero, nomearq($t0)  # Final Nulo do nomearq
        j       abreleitura
Incrementat0:
        addi    $t0, $t0, 1
        j       clean
```

```
# Abre o arquivo para leitura
abreleitura:
        li      $v0, servico_abre_arquivo
        la      $a0, nomearq            # nomearq é o nome do arquivo
        li      $a1, sinaliza_abre_leitura
        li      $a2, 0                  # Ignora o modo
        syscall                         # Descritor do arquivo retornado em v0
        move    $a0, $v0                # Salva o descritor do arquivo em a0

# Validação se o arquivo foi aberto da forma correta
validaarq:
        li      $t0, -1                         # Set t0 = -1
        beq     $a0, $t0, msgerroarq            # Se o descritor = -1, peça um novo Input

# Leitura do arquivo para o buffer
learq:                                  # Leitura do arquivo
        li      $v0, servico_le_arquivo
        la      $a1, Bufferarq          # Ler para o buffer
        li      $a2, 1024               # Não ler mais que 1024 bytes
        syscall

# Fecha o arquivo
fechaarq:                               # Por boas práticas, deve-se fechar o arquivo
        li      $v0, servico_fecha_arquivo
        syscall

# Retorna o chamador
jr      $ra

#-------[ Gera palavra randômica ]--------------------------------------------------
#       Pega um palavra aleatória
#       Usaremos SEMPRE um arquivo com 50 palavras

GeraRandom:
        li      $v0, 30                 # 30 é o codigo da função para pegar o valor
        syscall

        li      $v0, 40                 # 40 é o código da função para setar o valor randômico
        move    $a1, $a0        # Coloca a parte do tempo que atualiza de segundo em segundo
        li      $a0, 0          # Gerador aleatório 0
        syscall                 # Gerador aleatório é agora um valor aleaório

        li      $v0, 42                 # 42 é o código da função para gerar
        li      $a1, 50         # Nosso numero aleatório é um numero entre 0<=num<50
        li      $a0, 0
        syscall                 # Temos o valor aleatório em $a0

getRandomWord:
        la      $t1, Bufferarq          # Primeiro caracter do Bufferarq
        li      $t0, 0                  # Contador começa com 0
        la      $t2, Palavra            # A Palavra
```

```
        li      $s5, -1                 # Inicializa o numero de caracteres da palavra
getRandomWordLoop:
        lb      $t3, 0($t1)             # Carrega o caracter do Bufferarq
        addi    $t1, $t1, 1             # Próximo caracter
        beq    $t3, 0x0a, randomNext          # Nova linha
        beq     $t0, $a0, rAddletra     # Adiciona a letra na palavra
        j       getRandomWordLoop

randomNext:
        beq    $t0, $a0, finalizeWord # Terminador nulo da palavra
        addi    $t0, $t0, 1             # Incrementa o contador
        j getRandomWordLoop

rAddletra:
        sb      $t3, 0($t2)             # Coloca a letra na palavra
        addi    $t2, $t2, 1             # Proxima letra da palavra
        addi    $s5, $s5, 1             # Pega o numero de caracteres
        j       getRandomWordLoop

finalizeWord:
        li      $t3, 0x00
        sb      $t3, 0($t2)             # Terminador nulo da Palavra Selecionada

        jr      $ra

#       Final
#-------------------------------------------------------------------------------------------------

#-------[ Lógica principal do jogo ]-------------------------------------------------------------
rodajogo:
        jal     solicitaletra           # Pedido de letra
        move    $a2, $v0                # Copia a entrada para a2
        jal     limpaterminal
        la      $a1, Palavra            # Precisamos trocar Palavra com a própria palavra
        la      $a0, adivinhou
        jal     updateGuess             # Verifica se a letra não foi tentada ainda
        jal     mostratentativas        # Mostra as entradas anteriores

        bne     $v0, $0, jaadivinhou  # Continua com o if se a letra já foi tentada
        la      $a3, situacaoatual

        jal     gerapalavratela         # Retorna o layout da palavra
        jal     strContains             # verifica se está correta

        beq     $v0, $0, naotemletra

        # Se foi uma tentativa valida
        li      $v0, 11                         # Mostra o caracter
        li      $a0, 0x1b                       # (ASCII ESC)
        syscall

        la      $a0, Sim
```

```
        syscall
        j       temletra

jaadivinhou:
        la      $a0, ja                 # "Você já tentou esta letra"
        li      $v0, 4
        syscall

temletra:                               # Palavra contêm a letra
        jal     mostraforca
        beq     $s5, $s6, VoceVenceu    # Coloca as letras na palavra
        jal     somacertou
        j       rodajogo

naotemletra:                            # Palavra não contêm a letra
        li      $v0, 11                 # Mostra a letra
        li      $a0, 0x1b               # (ASCII escape)
        syscall

        la      $a0, No
        syscall

        addiu   $s0, $s0, 1             #Incrementa o número de tentativas erradas


        jal     mostraforca
        beq     $s0, 7, Voceperdeu
        jal     somerrou

        j       rodajogo

mostratentativas:
        addi    $sp, $sp, -12           # Aloacação
        sw      $ra, 0($sp)             # Salva o antigo ra
        sw      $a0, 4($sp)             # Salva o antigo a0
        sw      $v0, 8($sp)             # Salva o antigo s0

        move    $t0, $a0
        li      $v0, 4                  # 4 é o código da função para printar uma string
        la      $a0, jaacertou          # "Tentativas: "
        syscall

        la      $t0, adivinhou          # Pega o numero de tentativas
        li      $v0, 11                         # 11 é o código da função para printar um caracter

        lb      $a0, 1($t0)             # Nosso primeiro byte é o "!", então começa no 1
        beq     $a0, 0, tentativafinal  # Tentativa final

        syscall
loopadivinhacao:

        addi    $t0, $t0, 1
```

```
        lb      $a0, 1($t0)
        beq     $a0, 0, tentativafinal
        move    $t1, $a0

        li      $a0, 0x2c               # Printa uma virgula
        syscall

        li      $a0, 0x20               # Printa um espaço
        syscall

        move    $a0, $t1
        syscall
        j       loopadivinhacao

tentativafinal:
        lw      $ra, 0($sp)             # Carrega o antigo ra
        lw      $a0, 4($sp)             # Carrega o antigo a0
        lw      $v0, 8($sp)             # Carrega o antigo s0
        addi    $sp, $sp, 12            # Desaloca

        jr      $ra

limpaterminal:
        li      $v0, 11                 # Printa um caracter
        li      $a0, 0x1b               # Ascii escape
        syscall

        li      $v0, 4                  # Printa uma palavra
        la      $a0, limpatela
        syscall
        jr      $ra

mostraforca:
        li      $v0,11                  # Printa um caracter
        li      $a0,'\n'                # Pula uma linha
        syscall

        li      $t1, 93                 # O personagem possui exatamente 93 caracteres
        mul     $t0, $s0, $t1
        li      $v0, 4                  # Printa uma string
        la      $a0, forca              # Parte superior da figura
        addu    $a0, $a0, $t0           # adiciona o numero de jogadas
        syscall
        addi    $t2, $a0, 50

        la      $a0, situacaoatual      # Printa a situacao atual da palavra
        syscall

        move    $a0, $t2                # Recebe a metade inferior da imagem
        syscall
        jr      $ra
```

```
VoceVenceu:
        li      $v0, 4                  # 4 é o codigo da função que printa uma string
        la      $a0, venceu
        syscall

        addi $s7,$s7,1                  # Soma a pontuação
        li      $v0,4                   # 4 é o codigo da função que printa uma string
        la      $a0,pont
        syscall

        move $a0, $s7                   # Move o a pontuacao para passar como parametro
        li      $v0,1                   # 1 é o codigo da função que printa um inteiro
        syscall

        li      $v0,4                   # 4 é o codigo da função que printa uma string
        la      $a0,pont2
        syscall

        #jal limpalinha

        #jal    mostraforca

        jal     somganhou
        j       Exit

limpalinha:
        li      $v0, 11                 # Printa um caracter
        li      $a0, 0x1b               # ASCII escape
        syscall

        li      $v0, 4
        la      $a0, limpatela
        syscall

        li      $v0, 11                 # Printa um caracter
        li      $a0, 0x1b               # ASCII escape
        syscall

        li      $v0, 11                 # Printa um caracter
        li      $a0, 0x1b               # ASCII escape
        syscall

        li      $v0, 4

        jr      $ra

Voceperdeu:
        li      $v0, 4                  # 4 é o codigo da função que printa uma string
        la      $a0, Perdeu             # "Você Perdeu!\n"
        syscall
```

```
        li      $v0, 11                 # Printa um caracter
        li      $a0, 0x1b               # ASCII escape
        syscall

        li      $v0, 4                  # 4 é o codigo da função que printa uma string
        la      $a0, padrao
        syscall

        la      $a0, palavracerta       # "A palavra correta era:"
        syscall

        la      $a0, Palavra            # A palavra correta
        syscall

        li      $v0,4                   # 4 é o codigo da função que printa uma string
        la      $a0,pont3
        syscall

        move $a0, $s7                   # move a pontuacao para passar como parametro
        li      $v0,1                   # 1 é o codigo da função que printa um inteiro
        syscall

        li      $v0,4                   # 4 é o codigo da função que printa uma string
        la      $a0,pont2
        syscall

        sub $s7,$s7,$s7                 # Zera os acertos consecutivos

        jal     somperdeu

Exit:

        li      $v0, 11                 # Printa um caracter
        li      $a0, 0x1b               # ASCII escape
        syscall

        li      $v0, 4                  # 4 é o codigo da função que printa uma string
        la      $a0, padrao
        syscall

        la      $a0, jogadenovo         # Gostaria de Jogar de novo?
        syscall

        li      $v0, 12                 # 12 é o codigo da função para ler um caracter
        syscall

        #S minusculo ou maiúsculo irá funcionar
        beq     $v0, 0x73, iniciajogo
        beq     $v0, 0x53, iniciajogo

        li      $v0, 4                  # 4 é o codigo da função que printa uma string
        la      $a0, msgtchau           # Mostra msgtchau
```

```
        syscall

        li      $v0, servico_termina_exec
        syscall

#       Final Menu logico do jogo
#-------------------------------------------------------------------------------------------

#-------[ Interação do usuario ]-----------------------------------------------------------
#       Prompt Caracter
solicitaletra:
        addi    $sp, $sp, -12           # Alocação
        sw      $ra, 0($sp)             # Armazena o antigo ra
        sw      $a0, 4($sp)             # Armazena o antigo a0
        sw      $s0, 8($sp)             # Armazena o antigo s0

        li      $v0, 11                 # Printa um caracter
        li      $a0, 0x1b               # ASCII escape
        syscall

        li      $v0, 4                  # 4 é o codigo da função que printa uma string
        la      $a0, padrao
        syscall

        la      $a0, escolha            # Tentativa de uma letra
        syscall

        li      $v0, 12                 # 12 é o código da função para ler um caracter
        syscall                         # v0 contêm o caracter

        bge     $v0, 0x61, retorno      # Recebe letra
        addi    $v0, $v0, 0x20          # Converte para minuscula

retorno:
        blt     $v0, 0x61, letrainvalida    # Verifica se o valor é menor que A, ou seja inválido
        bgt     $v0, 0x7a, letrainvalida    # Verifica se o valor é maior que a, ou seja inválido
        lw      $ra, 0($sp)             # Carrega o antigo ra
        lw      $a0, 4($sp)             # Carrega o antigo a0
        lw      $s0, 8($sp)             # Carrega o antigo s0
        addi    $sp, $sp, 12            # Desaloca
        jr      $ra                     # Retorno

letrainvalida:
        li      $v0, 4                  # 4 é o codigo da função que printa uma string
        la      $a0, teclainvalida      # "Caracter invalido"
        syscall

        j       solicitaletra

#       Final Prompt Caracter
#-------------------------------------------------------------------------
```

```
#-------[ Manipulação da String ]-----------------------------------------------------
#       Checa se a string contêm a string digitada

strContains:
        addi    $sp, $sp, -4    # Alocacao
        sw      $a1, 0($sp)     # Armazena o antigo
        li      $v0, 0

strContainsIter:
        lb      $t0, 0($a1)                 # Carrega um caracter da string
        beq     $t0, $0, strContainsIterBrk # # Para o loop se o final da string é encontrado
        addi    $a1, $a1, 1                 # Incrementa o endereço da string para continuar
escaneando
        beq     $t0, $a2, letraencontrada           # Desvia se os caracter estão certos
        j       strContainsIter                     # Pula para o inicio do loop

letraencontrada:
        addi    $s6, $s6, 1     # Incrementa o número de tentativas corretas
        li      $v0, 1          # Se encontra retorna 1
        j       strContainsIter

strContainsIterBrk:
        lw      $a1, 0($sp)     # Carrega o antigo a0
        addi    $sp, $sp, 4     # Desaloca

        # Retorna para a funcao chamadora
        jr      $ra             # Return

#       Final da Manipulação da String
#-------------------------------------------------------------------------------

#-------[ Lógica das tentativas ]-----------------------------------------------------
# Letra testada

updateGuess:
        addi    $sp, $sp, -8                # Alocaçao
        sw      $a1, 0($sp)                 # Armazena o antigo a0
        sw      $a0, 4($sp)                 # Armazena o antigo a1
        li      $v0, 0                      # Se foi ou não encontrado

updateGuessIter:
        lb $t0, 0($a0)                      # Carrega um caracter da string
        beq $t0, $0, updateGuessIterBrk             # Para o loop se é o final da string
        bne $t0, $a2, letranaoencontrada            # Desvia se o caracter não bate
        li $v0, 1

letranaoencontrada:
        addi $a0, $a0, 1                    # Incrementa o buffer de tentativas
        j updateGuessIter

updateGuessIterBrk:
        sb $a2, 0($a0)                      # Armazena o caracter passado na posição
```

```
        lw $a0, 4($sp)                  # Carrega antigo a1
        lw $a1, 0($sp)                  # Carrega antigo a0
        addi $sp, $sp, 8                # Desaloca
        jr $ra                          # Retorno

# Formação da string ----------------------------------------------------------
#       Gera a palavra sublinhada

gerapalavratela:
        addi    $sp, $sp, -12           # Aloca 4 bytes
        sw      $a0, 0($sp)             # Armazena o antigo a0
        sw      $a1, 4($sp)             # Armazena o antigo a1
        sw      $a3, 8($sp)
        li      $v0, 0                  # Se foi ou não encontrado
        move    $t1, $a0
        li      $t3, 0x5F               # Carrega o caracter sublinhado

loopgerapalavratela:
        lb      $t2, 0($a1)                     # Palavra totalmente correta
        lb      $t0, 0($a0)                     # Todas letras testadas
        beq     $t0, $0, encerraloopgerapalavra # Para o loop se é o final da string
        beq     $t0, $t2, addletra              # Se uma das tentativas é correta
seguegerandopalavra:
        sb      $t3, 0($a3)
        addi    $a0, $a0, 1

        j       loopgerapalavratela

addletra:
        move    $t3, $t2
        j       seguegerandopalavra

encerraloopgerapalavra:
        addi    $a3, $a3, 1     # Vai para o próximo localização no display da palavra

        li      $t4, 0x20
        sb      $t4, 0($a3)
        addi    $a3, $a3, 1

        move    $a0, $t1        # Vai para o incio das letras tentadas
        addi    $a1, $a1, 1     # Vai para a proxima letra na palavra correta
        lb      $t5, 1($a1)     # Se ficar o 0 ele conta um Underline a mais na palavra
        li      $t3, 0x5F       # Carrega o caracter sublinhado
        beq     $t5, $0 finalizagerapalavra
        j       loopgerapalavratela

finalizagerapalavra:
        lw      $a3, 8($sp)
        lw      $a1, 4($sp)     # Carrega antigo a1
        lw      $a0, 0($sp)     # Carrega antigo a0
        addi    $sp, $sp, 12    # Desaloca
        jr      $ra             # Retorno
```

```
# Subrotinas Som -------------------------------------------------------------
somacertou:
	li	$a2, 0		# SOM ID
	li	$a3, 127	# volume
	li	$v0, 33

	li	$a0, 55
	li	$a1, 100
	syscall

	li	$v0, 32
	li	$a0, 65
	syscall

	li	$a2, 0		# SOM ID
	li	$a3, 127	# volume
	li	$v0, 33

	li	$a0, 60
	li	$a1, 100
	syscall

	jr $ra

somerrou:
	li	$a2, 0		# SOM ID
	li	$a3, 127	# volume
	li	$v0, 33

	li	$a0, 60
	li	$a1, 100
	syscall

	li	$v0, 32
	li	$a0, 65
	syscall

	li	$a2, 0		# SOM ID
	li	$a3, 127	# volume
	li	$v0, 33

	li	$a0, 55
	li	$a1, 100
	syscall

	jr $ra

somganhou:
	li	$a2, 0		# SOM ID
	li	$a3, 127	# volume
	li	$v0, 33
```

```
        li      $a0, 69
        li      $a1, 100
        syscall

        li      $a0, 75
        li      $a1, 100
        syscall

        li      $a0, 89
        li      $a1, 100
        syscall

        li      $a0, 78
        li      $a1, 250
        syscall

        li      $a0, 85
        li      $a1, 100
        syscall

        li      $a0, 90
        li      $a1, 250
        syscall

        li      $a0, 85
        li      $a1, 250
        syscall

        li      $a0, 90
        li      $a1, 250
        syscall

        li      $a0, 94
        li      $a1, 500
        syscall

        jr $ra

somperdeu:
        li      $a2, 0          # SOM ID
        li      $a3, 127        # volume
        li      $v0, 33

        li      $a0, 55
        li      $a1, 500
        syscall

        li      $a0, 54
        li      $a1, 500
        syscall
```

```
li      $a0, 51
li      $a1, 500
syscall

li      $a0, 50
li      $a1, 1500
syscall

jr      $ra
```