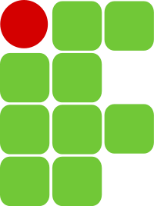
****

**INSTITUTO FEDERAL DE EDUCAÇÂO CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL**

**CAMPUS NOVO HAMBURGO**

**CURSO TECNICO EM MECATRÔNICA**

**GABRIEL MACHADO**

**RAFAEL LIMA**

**MICROCONTROLADORES**

**Display LCD com Arduino**

**Prof.ª Bruna Flesch**

**Novo Hamburgo**

**2019**

**SUMÁRIO**

1. **INTRODUÇÃO...........................................................................................3**
2. **METODOLOGIA........................................................................................3**
3. **RESULTADOS...........................................................................................4**

**3.1 COMPONENTES UTILIZADOS...........................................................4**

**3.2 HARDWARE DO PROJETO................................................................5**

**3.2.A. ANÁLISE DE HARDWARE...................................................6**

**3.3 SOFTWARE DO PROJETO .................................................................6**

1. **CONCLUSÃO............................................................................................16**
2. **REFERÊNCIAS BIBLIOGRÁFICAS .........................................................16**
3. **INTRODUÇÃO**

O trabalho tem como objetivo o desenvolvimento de dois jogos utilizando um display LCD, um microprocessador Arduino e também um teclado matricial de membrana.

Jogo Conversão Binaria: possui três dificuldades, fácil onde seleciona de forma aleatória um número de 1 a 15 na forma binaria e o usuário deve informar seu valor em forma decimal, médio com números de 16 a 63 e difícil com números de 64 a 255.

Jogo da Sequência: também possui três dificuldades, fácil o programa seleciona de forma aleatória um número de 10 a 99 e o jogador tem um tempo para memorizar e em seguida digitar o número, médio com números de 100 a 999 e difícil com números de 1000 a 9999.

Os dois jogos fazem parte de um único programa com menu de escolha do jogo, funções que identificam acertos e erros e possibilitam selecionar outro jogo ou continuar após o game over.

1. **METODOLOGIA**

Para desenvolvimento do projeto foram utilizados componentes como: display LCD 16x2 Keypad shield com botões, microcontrolador Arduino Mega e também um teclado matricial 4x4 com 16 dígitos, nas aulas de microcontroladores foram feitos os estudos da programação do microcontrolador Arduino através da apresentação de conceitos, definições e exercícios práticos e teóricos.

A simulação do projeto foi feita utilizando uma ferramenta online chamada Tinkercad, posteriormente foi montado o circuito na pratica e feito testes.

1. **RESULTADOS**

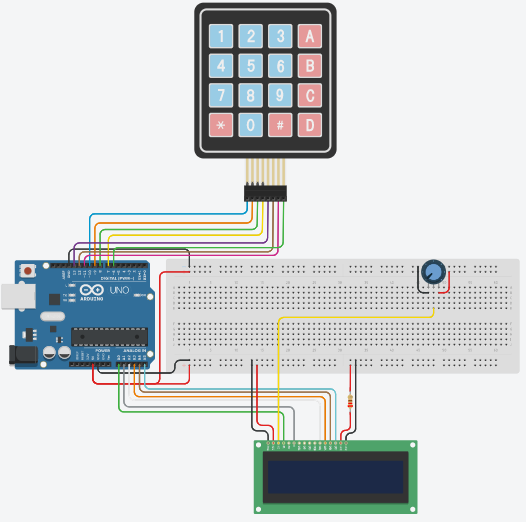
Foi solicitado o uso do display LCD, no trabalho utilizamos novas bibliotecas como: Keypad.h usada para controlar o teclado e também LiquidCrystal.h que possibilita a programação do monitor LCD.

**3.1 COMPONENTES UTILIZADOS**

|  |  |
| --- | --- |
| Microcontrolador Arduino Uno  (Simulador) |  |
| Teclado matricial 4x4 com 16 dígitos  (Simulador e projeto) |  |
| Display LCD 16x2  (Simulador) |  |
| Potenciômetro  (Simulador) |  |
| Arduino Mega 2560  (Projeto) |  |
| Display LCD Keypad Shield  16x2 com teclado  (Projeto) |  |

*Tabela 1 – Componentes utilizados no projeto*

**3.2 HARDWARE DO PROJETO**

****

*Figura 1 – Hardware do projeto simulado (Tinkercad)*

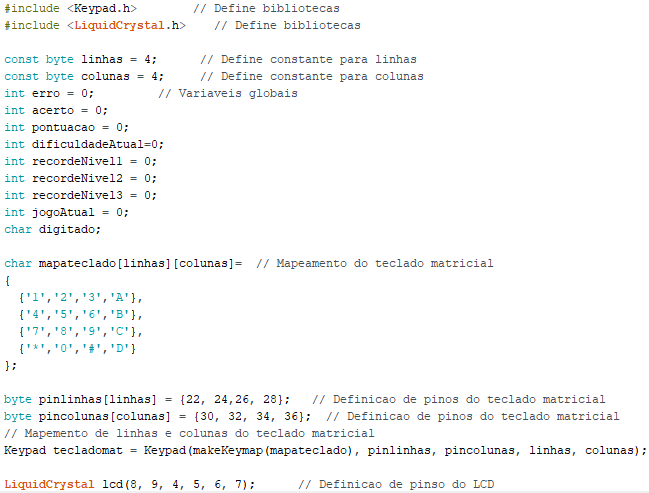
**3.2.A. ANÁLISE DE HARDWARE**

O projeto conta com um microcontrolador Arduino responsável pelo processamento de dados, display LCD que exibe todas as mensagens para o usuário, um teclado para que os dados possam ser enviados ao Arduino e um potenciômetro que regula o brilho da tela.

Para a montagem do projeto na pratica foi utilizado um Arduino mega pois possui mais portas, possibilitando o acoplamento do LCD e do teclado sem precisar utilizar protoboard.

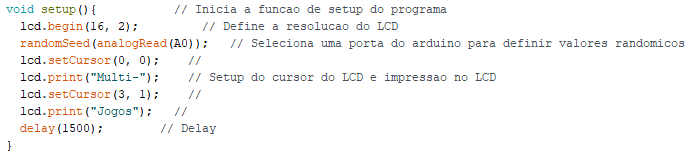
**3.3 SOFTWARE DO PROJETO**

Criação das variáveis de controle, definição dos pinos do hardware e inclusão das bibliotecas.



*Figura 2 – Variáveis do programa*

**Função** **void setup**, é a especificação das configurações iniciais do programa, como; declaração dos valores iniciais das variáveis, definição dos pinos de entrada e saída, definição da velocidade de comunicação.

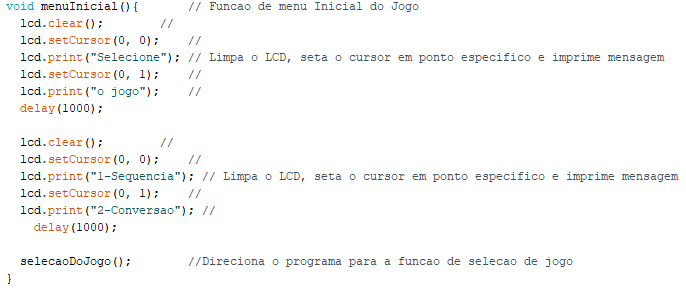


*Figura 3 – Função void setup ()*

**Função loop**: essa função tem o objetivo de repetir uma estrutura de forma continua ou até que um comando “parar” seja enviado ao Arduino.

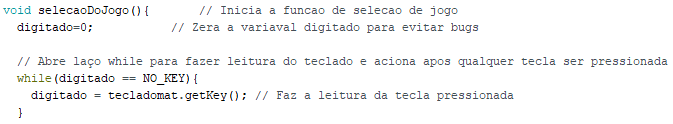
O projeto feito com diferentes funções para minimizar erros e torná-lo mais organizado, será explicado cada uma de forma separada.

**Função menuInicial**: responsável pela seleção do jogo, sequencia ou conversão binaria.



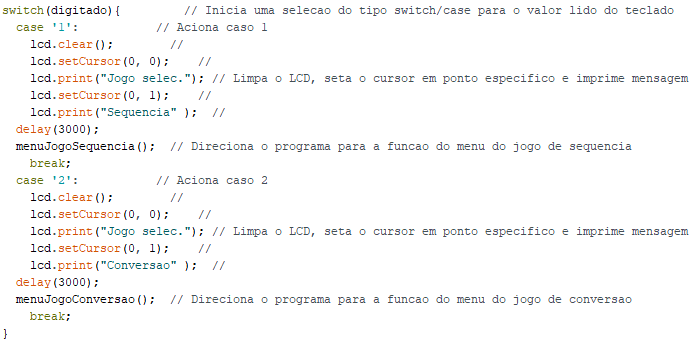
*Figura 4 – Código função menuInicial.*

**Função seleçãoDoJogo:** Aguarda o jogador digitar uma das opções.

**

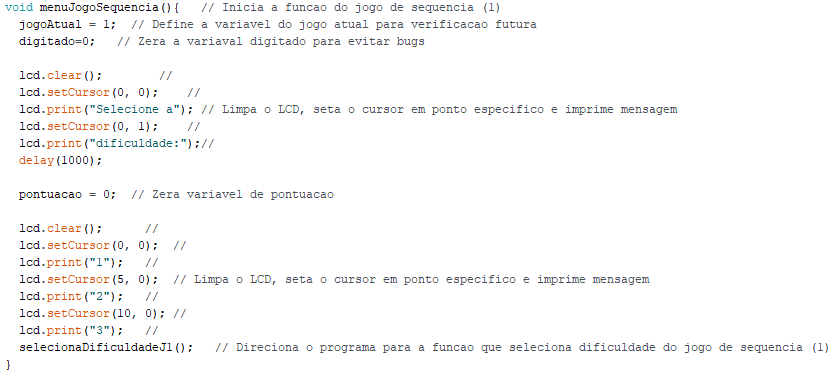
*Figura 5 – Codigo da função seleçãDoJogo.*

**Função switch:** faz um switch case para selecionar o jogo.



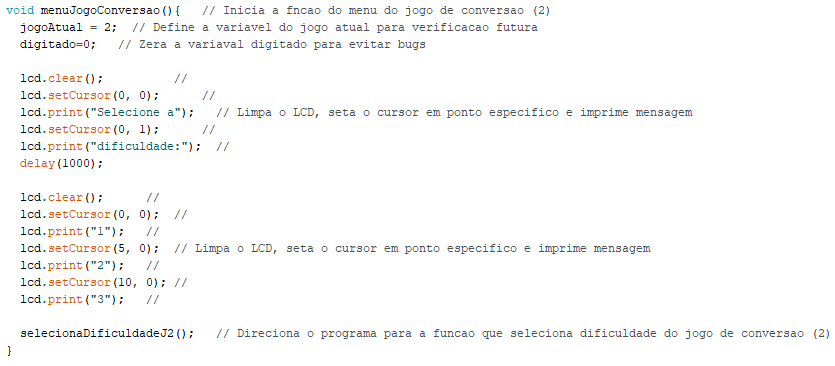
*Figura 6 – Código da função switch.*

**Função menuJogoSequencia:** Inicia o jogo sequência e solicita a dificuldade do jogo.



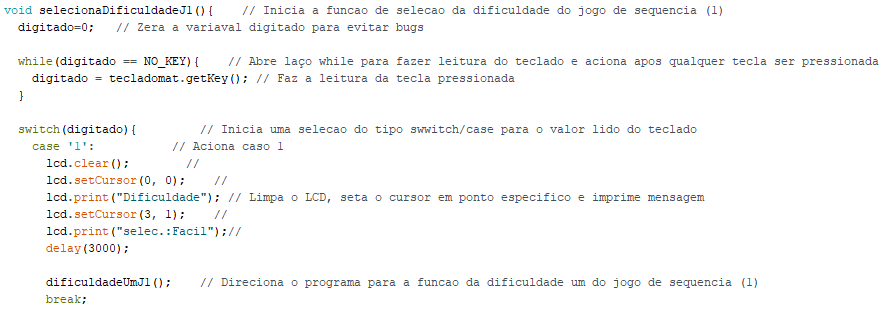
*Figura 7 – Código da função do jogo Sequência.*

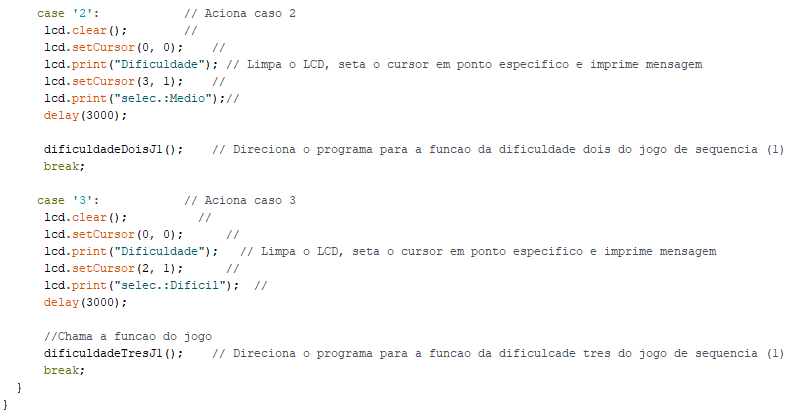
**Função menuJogoConversão:** Inicia o jogo Conversão e solicita a dificuldade do jogo.



*Figura 8 – Codigo que abre a janela no modo manual*

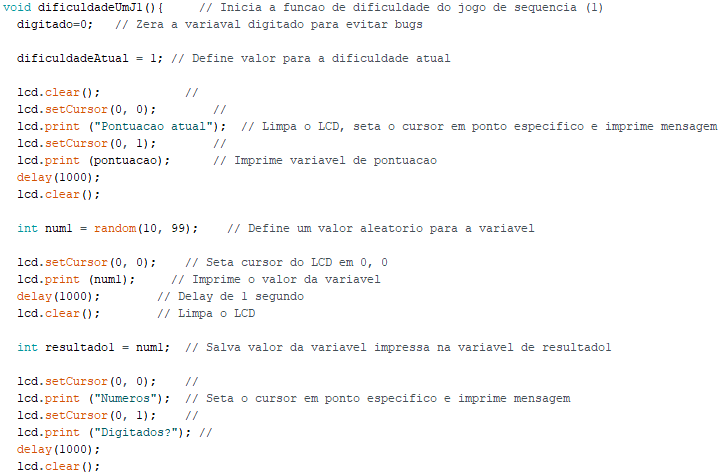
**Função selecionaDificuldadeJ1:** Função que seleciona uma das três dificuldades do jogo Sequencia, a função que seleciona a dificuldade do jogo Conversão é exatamente igual.

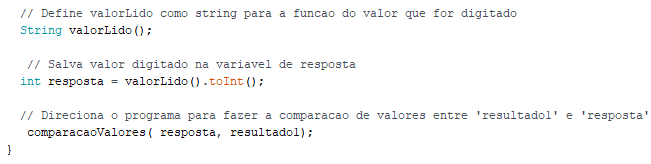


**

*Figura 9 – Código da função selecionaDificuldadeJ1.*

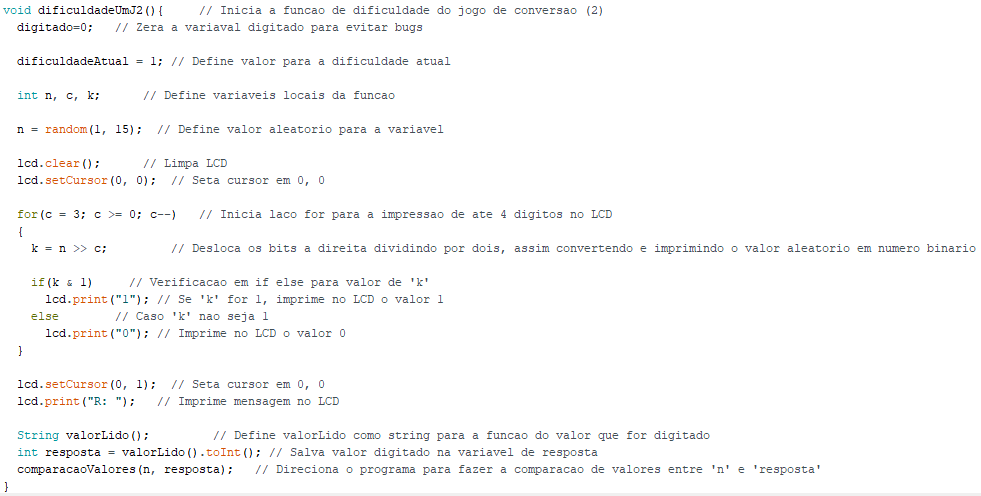
**Função dificuldadeUmJ1:** Executa o jogo na dificuldade fácil, para nível médio e difícil se usa uma função igual, alterando apenas a variável num1 para que o jogo fique mais difícil.

**



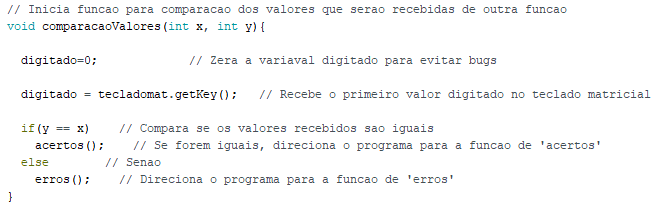
*Figura 10 – Código da função dificuldadeUmJ1.*

**Função dificuldadeUmJ2:** Executa o jogo na dificuldade fácil, para nível médio e difícil se usa uma função igual, alterando apenas a variável n para que o jogo fique mais difícil.

**

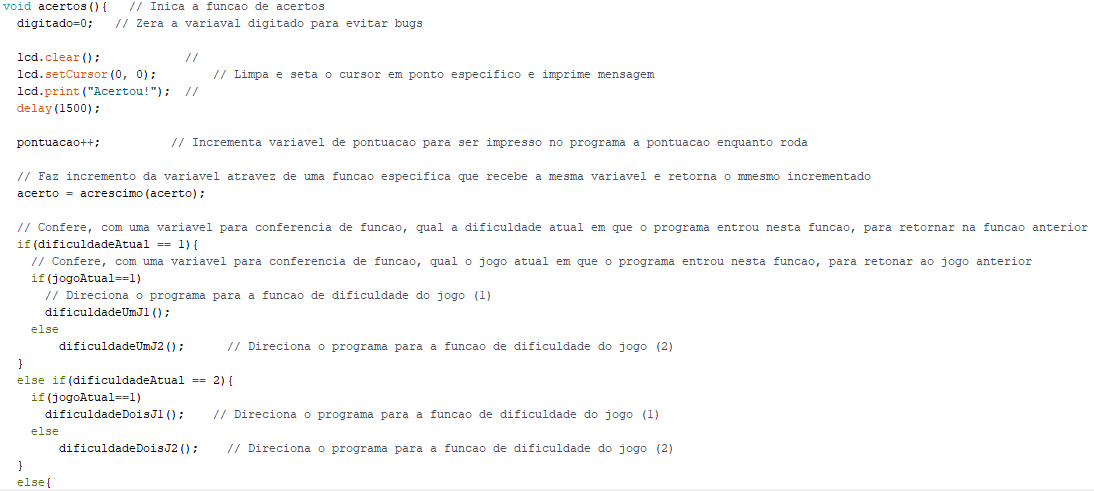
*Figura 11 – Codigo da função dificuldadeUmJ2.*

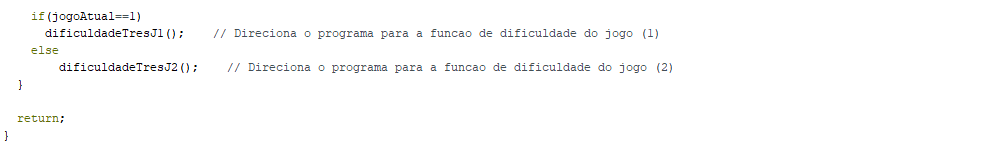
**Função comparacaoValores:** Compara o valor da resposta do jogo com o numero digitado pelo usuário e define se é um acerto ou erro.

**

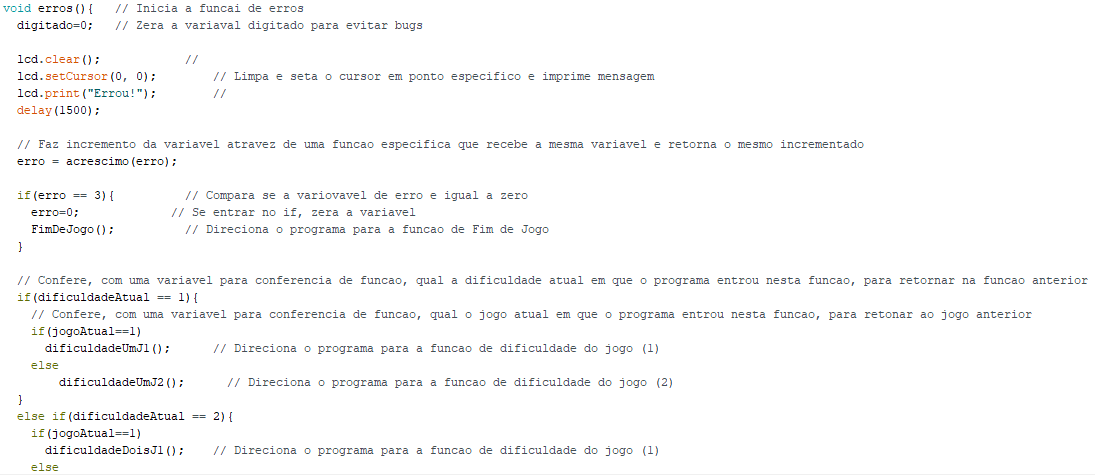
*Figura 12 – Código da função comparacaoValores.*

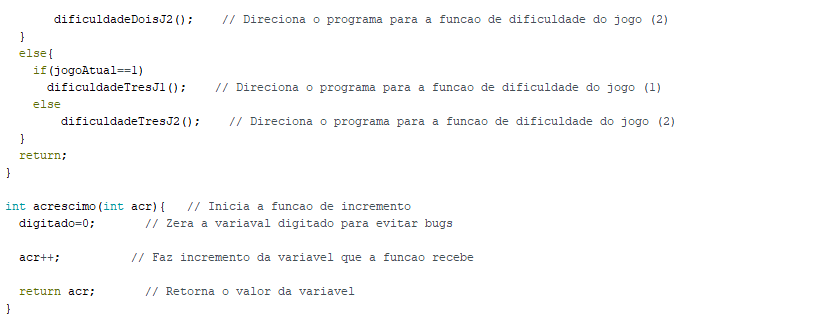
**Função acertos:** Verifica se o jogador acertou a questão, incrementa a pontuação e retorna pra o jogo que está sendo executado no momento.

**

**** *Figura 13 – Código da função acertos.*

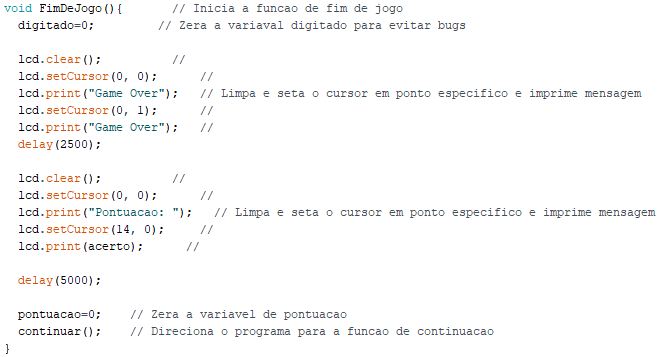
**Função erros:** Verifica se o jogador acertou a reposta e retorna para o jogo e dificuldade atual, também contabiliza a quantidade de erros, sendo três para game over.

****

****

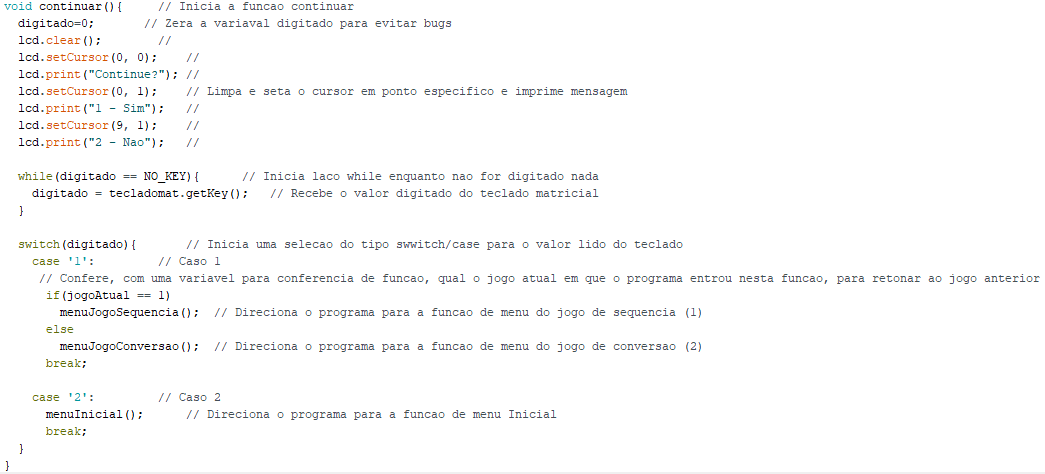
*Figura 14 – Código da função erros.*

**Função FimDeJogo:** Informa o fim de jogo para o jogador e informa a pontuação.

**

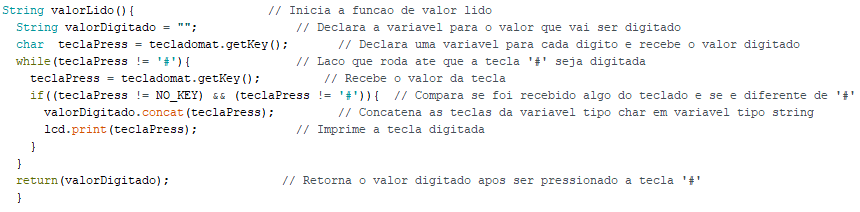
*Figura 15 – Código da função FimDeJogo.*

**Função continuar:** Pergunta ao jogador se quer continuar jogando.

**

*Figura 16 – Código da função continuar.*

**Função valorLido:** Utilizado em todas funções que necessitam a leitura do teclado.

**

*Figura 17 – Código da função valorLido.*

1. **CONCLUSÃO**

O conhecimento adquirido neste trabalho possibilita a criação de inúmeros outros projetos que utilizam os recursos solicitados, como por exemplo o menu de controle de uma máquina, ou a comunicação de operador com máquina.

Pôde-se entender melhor como a comunicação de hardware e software funciona e adquirir uma visão mais ampla sobre projetos.

O trabalho finaliza a segunda etapa da disciplina de microcontroladores.

1. **REFERÊNCIAS BIBLIOGRAFICAS**

FLESCH, Bruna; **LCD, Relé, Servo Motor e Circuitos.** 2019. 19 slides,

TINKERCAD, **Autodesk Tinkercad.** Disponível em: <https://www.tinkercad.com/>. Acesso em 25 de novembro de 2019.