Universidade Federal Fluminense – UFF
Escola de Engenharia – TCE
Curso de Engenharia de Telecomunicações – TGT

Programa de Educação Tutorial – PET
Grupo PET-Tele

Código Morse

Autores: Lúcio Folly Sanches Zebendo (2019)
Alyne Nobre (2019)

Tutor: Alexandre Santos de la Vega

Niterói – RJ
Maio / 2019

1 Introdução

Este documento é consequência de uma pesquisa aprofundada sobre o código Morse. Nele contém alguns projetos realizados pelo Grupo PET Tele usando para além do código referido como base, um resgate sobre suas origens e fatores que levaram à sua criação. Uma ideia central por detrás da abordagem do tema é trazer-lhe uma perspectiva educacional, conhecimento do público e respeito aos assuntos relacionados às telecomunicações.

Como resultado, pretendemos mostrar um dos ramos da engenharia que mais tem se desenvolvido nos últimos anos, de maneira lúdica e didática.

2 Como surgiu?

O código Morse, como seu próprio nome diz, foi inventado por Samuel Finley Breese Morse. Seu surgimento foi necessario em razão sa criaçao do telegrafo elétrico em 1838 -pelo prorio Samuel-.

O aparelho tinha um código base para ser operado e esse código se chama Morse, em homenagem ao seu inventor.

Nesse momento se desenvolveu uma das áreas mais importantes das telecomunicações, mas antes de falar sobre essa área vamos explicar o que é o telégrafo elétrico .

2.1 O telégrafo elétrico

O telégrafo elétrico é um sistema de comunicação que foi inventado antes mesmo do telefone. Contudo, ele passou por numerosas transformações, como o telégrafo mecânico e o telégrafo ótico (desenvolvido por Claude Chappe), até a chegada do telégrafo eletrônico, criado por Samuel Morse.Este último,por sua vez, teve um primeiro modelo não era muito útil devido à sua falta de praticidade, visto que requeria um fio para cada letra do alfabeto.

Apenas alguns anos depois o aparelho ganhou uma forma mais comerciável, contendo apenas um fio, que, por meio de ondas elétricas, transmitia as informações.

A partir disso, a comunicação mundial foi revolucionada e, no ano de 1860, Morse recebeu um prêmio de reconhecimento pelo feito histórico, entregue por Napoleão III.

Figura 1: O dispositivo de transmissão mais conhecido do telégrafo.

3 Funcionamento

A nomenclatura utilizada pelo código baseia-se em pontos e traços, que correspondem a sinais elétricos. O ponto faz o papel de unidade e tem cerca de 1/25 segundos de duração.

Por sua vez, o sinal elétrico relativo ao traço equivale a três pontos.

Cada ponto e traço é seguido por um pequeno silêncio com o tempo referente ao do ponto (0,04 segundos), já entre as letras de uma palavra são separadas por três vezes esse tempo e as palavras por sete vezes esse tempo.

A seguir, é apresentado um pequeno guia sobre o código.

Primeiramente,

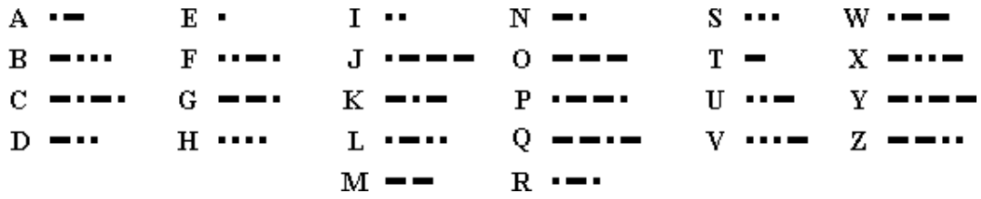

Figura 2: ilustra as principais letras.

Em seguida,

Algarismo	Sinal	Pontuação	Sinal
1	•—————	Ponto	••••••
2	••———	Ponto e vírgula	—•—•—•
3	•••——	Vírgula	•—•—•—
4	••••—	Dois pontos	———•••
5	•••••	Interrogação	••——••
6	—••••	Exclamação	——••—
7	——•••	Apóstrofe	•————•—
8	———••	Traço de união	—•••—
9	————•—	Aspas	•—••—•
0	—————	Parêntesis	—•——•—
		Alínea	•—•—••
		Sublinhado	••——•—
		Duplo traço (=)	—•••—

Figura 3: mostra os números e sinais de pontuação.

Em todas as figuras os valores são traduzidos para o código Morse.

4 Importância

Tendo em vista, que o Telégrafo Eletrônico e o Código Morse transformaram fortemente a área das telecomunicações, hoje podemos dizer que Samuel Finley Breese Morse foi um dos pais da comunicação à distância que conhecemos hoje, pois suas contribuições revolucionaram o meio de comunicação.

4.1 Meio de comunicação

Sendo uma ferramenta usada para permitir a comunicação entre um ou mais indivíduos, realizando assim a disseminação de diversos tipos de informações, o meio de comunicação vem sofrendo ao longo da história da humanidade vários tipos de alterações e evoluções. Surgindo da necessidade dos homens de passarem informações uns aos outros, a comunicação começou por meio de gestos, de sinais e de sons. Logo depois, veio a escrita em 15000 a.c. a partir de desenhos no interior de cavernas africanas.

Com a escrita, as cartas apareceram para se tornar um meio de comunicação bastante usado para o envio de informação a longas distâncias, mas foi entrando em desuso com o aparecimento da telefonia, anos depois.

Ainda antes dos telefones, o telégrafo surgiu, na década de 1790, como uma das primeiras invenções tecnológicas do meio de comunicação à distância.

Entre as décadas de 1830 e 1840, Morse surgiu com telégrafo elétrico (seção 2.1), esse sistema é prático e simples, já que usava o Código Morse, possibilitando o seu uso até a criação do telefone, 32 anos depois. No último século, com o desenvolvimento do setor de telecomunicações os meios de comunicação à distância se desenvolveram drasticamente, usando as tecnologias analógicas (que incluem rádios e telefonia tradicional) e as tecnologias digitais (que incluem as redes de computadores).

Mas apesar de toda essa história, o meio de comunicação sempre compreendeu os três processos da comunicação, e são eles:

4.1.1 Emissor

É a fonte que pretende comunicar uma mensagem (também pode ser chamada de fonte ou de origem). Ele é responsável por dar significado a sua mensagem, ou seja, elaborar à ideia e o conceito que se deseja comunicar, e codificar a mensagem, em outras palavras, constituir um mecanismo pelo qual a mensagem é elaborada para que possa ser transmitida (como exemplo temos o Código Morse).

4.1.2 Mensagem

É a ideia que o emissor deseja transmitir. Dividida em canal, também chamado de veículo, é o espaço situado entre o emissor e o receptor, com por exemplo os fios usados na comunicação da rede de telégrafos, e o ruído, que é a perturbação dentro do processo de comunicação.

4.1.3 Receptor

É a etapa do recebimento da mensagem pelo destinatário.

Nela se decodifica a mensagem, ou seja, é estabelecido mecanismo para decifrar a mensagem, para que assim o receptor a compreenda, e confirme ao emissor que a mensagem foi recebida.

Figura 4

5 Meios de transmissão

Geralmente, as mensagens codificadas em Código Morse são reproduzidas por som ou luz. Por sua vez, são detectadas e para humanos por meio de aparelhos a captam e traduzam (por exemplo o radiotelégrafo e o telégrafo). No entanto, dependendo da criatividade dos usuários do código, ele pode ser transmitido de outras maneiras, já que envolve uma sequência de pontos e traços como visto anteriormente.

Esse meio de transmissão foi muito utilizado por marinheiros no século XIX, tendo o primeiro registro de um resgate marítimo utilizando o Código Morse em 1899, no Estreito de Dover. Ainda no século XIX, a utilização do Código Morse se popularizou rapidamente, alcançando praticamente todos os países europeus.

Em 1865, o Congresso Internacional Telegráfico regulamentou o Código Morse Internacional após algumas alterações no código criado por Samuel Morse, o que proporcionou maior dinamismo às comunicações. Com o surgimento do telefone, no fim do século XIX, o Código Morse caiu em desuso, e o desenvolvimento de novas tecnologias de comunicação mais eficazes desencadeou a substituição do sistema criado por Morse por outros aparelhos.

6 Projetos Desenvolvidos

6.1 Projetos Convencionais

6.1.1 Simulação de Faróis

Esse projeto tem a intenção de simular a comunicação entre dois locais, como é feito nos navios, por exemplo. Ele tem como intuito ultilizar materias de fácil acesso, para que crinças -com auxílio de adultos-, possam criar um brinquedo que simule um tipo de transmissão de informação, e essa, por sua vez, tem como base o Código Morse.

Material ultilizado:

- Dois prolongadores de ralo de 15 cm de diâmetro;
- Dois ralos de 15 cm de diâmetro, com um sistema de abertura manual;
- Dois soquete de lampadas;
- Duas lâmpadas;
- Papel alumínio;
- Dois circulos de plático de 15cm de diâmetro;
- Dois fios (+ tomadas nas pontas) de 3 m(ligados ao bocal);
- Cola quente.

Modo de preparo:

1. Pegue o papel alumínio e revista os prolongadores de ralo por dentro;
2. Faça um furo no círculo, preferencialmente no canto, para o fio passar;
3. Com a cola quente, cole o bocal no canto inferior do prolongador;
4. Encaixe a lâmpada no bocal;
5. Cole o ralo no lado superior do prolongador de ralo;
6. Cole papel aluminio em um dos lados do circulo de plástico;
7. Cole o círculo de plastico com o lado do papel aluminio para o interior do prolongador, lembre de posicionar corretamente o furo do fio;
8. Por fim é so conectar as tomadas.

(a) Projeto visto por cima, sem tampa.

(b) Projeto visto por baixo.

(a) Projeto visto por cima, com tampa.

(b) Fios ligados à lãmpada.

Figura 7: Projeto final.

6.2 Projetos com Arduino

Desde o princípio, foi elaborado a ideia de a protótipo com Arduino, e o uso da placa se autoconfigurou, devido a uma interação ludica com o espectador. Visando essa interação, elaboramosrpo dois projetos principais. O primeiro deles tem a intenção de apenas traduzir um texto para o Morse, já o segundo busca a transmissão de um texto à distância via código Morse.

6.2.1 Tradução de um texto para Morse(LED e Buzzer)

Apesar de simples, o projeto apresenta de maneira intuitiva como transmitir um texto em Morse. O código desse projeto encontra-se no final do documento(anexo 1).

Material utilizado:

- Buzzer;
- LED;
- Protoboard;
- Arduino Uno;
- Cabo USB-B para conexão com Arduino;
- Quatro Jumpers.

6.2.2 Transmissão de um texto via código Morse

O projeto se baseia na emissão de um feixe de luz vermelha através de um laser, cuja absorção será feita utilizando um sensor(LTR-301) controlado por outro arduino. Portanto, o projeto tem como base a transmissão de informação em código morse por um feixe de laser a uma distância aproximada de 2 metros, e depois ser traduzida em português novamente. O código desse projeto encontra-se no final do documento(anexo 2).

Material utilizado no emissor:

- Laser vermelho;
- Protoboard;
- Arduino Uno;

- Cabo para conexão com Arduino;
- 2 Jumpers.

Material utilizado no receptor:

- Sensor LTR-301;
- Protoboard;
- Arduino Uno;
- Cabo para conexão com Arduino;
- 2 Jumpers.

6.3 Projetos com Octave

6.3.1 Desenvolvimento

O código em liguagem computacional Octave, que segue em anexo, foi criado para ilustrar a sonorização do Morse, como sendo um experimento.

código. A ideia era apenas um experimento para ajudar vocês.

Em seguida, para facilitar a depuração, foi anexado uma versão de símbolos textuais para o ponto, para o traço e para o espaço simples.

Nesse código Octave, são implementadas três versões de codificação para o alfabeto Morse: binário, símbolos textuais e sons.

Por binário, deve-se entender que tanto o ponto quanto o traço estão sendo representados por meio de um dos dois símbolos binários, enquanto o espaço simples está sendo representado pelo outro símbolo binário. Na versão atual do código (A2019M07D24), o ponto é representado por "1", o traço por "111" e o espaço simples por "0". Espaços múltiplos do espaço simples são representados por uma quantidade equivalente do símbolo "0".

Por símbolos textuais, deve-se entender que foram atribuídos três símbolos textuais para os três símbolos do código Morse. Dessa forma, pode-se dizer que foi utilizada uma codificação ternária. Na versão atual do código (A2019M07D24), o ponto é representado pela letra "o" ("o"), o traço por um conjunto de três hífens consecutivos ("—"), sem espaço entre eles, e o espaço simples por um espaço em branco (" "). Espaços múltiplos do espaço simples são representados por uma quantidade equivalente do símbolo " ".

Por sons, deve-se entender que tanto o ponto quanto o traço estão sendo representados por meio de um mesmo som, enquanto o espaço simples está sendo representado pelo silêncio. Logo, essa versão também pode ser considerada como uma representação binária. Na versão atual do código (A2019M07D24), o ponto é representado por um vetor de amostras que representa um sinal senoidal com frequência f = 1 kHz, com duração de Tdot = 0.08 s, amostrado com uma frequência Fs = 8 kHz. Deve ser ressaltado que, nas transmissões usuais, é adotado o valor de Tdot = 0.04 s. O traço é representado da mesma forma, porém com uma duração três vezes maior. O espaço simples é representado por um vetor de amostras nulas, com a mesma duração do ponto. Espaços múltiplos do espaço simples são representados por uma quantidade equivalente do vetor nulo. Comparando-se as três versões, pode-se dizer que:

- As versões binário e com símbolos textuais empregam um total de quatro elementos básicos para representar os três símbolos do código Morse, enquanto a versão com sons emprega quatro vetores de pontos, cuja quantidade de pontos depende diretamente de f, de Tdot e de Fs.
- A versão com símbolos textuais é ternária, enquanto as outras duas, em essência, são binárias. Assim, pode-se realizar as seguintes escolhas:
- Adotar a versão binário como uma representação interna e intermediária, que pode ser transformada nas demais.
- Adotar as versões binário e/ou de símbolos textuais (ternária) para uma transmissão digital, baseada em tensão ou corrente. Adotar a versão de sons apenas como versão final, a ser enviada a um circuito gerador de som.

Para ter acesso ao codigo cunsulte o anexo 3.

7 Conclusão

Portanto, a criação do telégrafo e do Código Morse foram a alavanca necessária para o desenvolvimento da tecnologia da comunicação. Podemos observar semelhanças até hoje, com a criação de Morse e dos sinais digitais, que se baseiam em dois elementos de codificação.

Concluímos assim que ainda na era dos computadores e smartphones continuamos a nos comunicar da maneira semelhante aos nossos antepassados.

8 Perguntas frequentes

Qual é a codificação empregada no Código Morse?
O tipo de codificação usado no código e a binária, que consiste em dois pontos: igaldo e desligado (zero e um). Já no caso do Código Morse traço e ponto.
Quais são os tipos de sinais mais usados na transmissão?
Sinais elétricos, sinais sonoros, sinais de luz e sinais de rádio.

Qual é a relação entre a codificação empregada e os sinais elétricos?
Tem a mesma cara de um sinal digital, já que o sinal será é formado por pulsos, causados por um circuito que ora estará, aberto ora fechado, em um determinado intervalo de tempo.

Qual é a relação entre a codificação empregada e os sinais auditivos?
Terá pequenos picos de frequências maiores ou menores, já que o sinal será formado por pulsos, causados por um circuito que ora estará aberto, ora fechado, em um determinado intervalo de tempo.

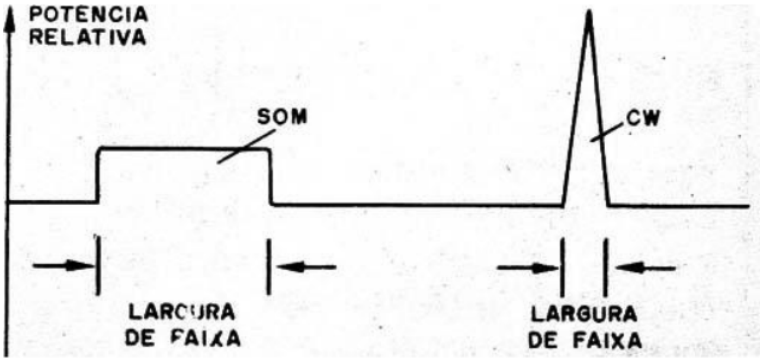

Figura 8: CW = Continuous Wave

Qual é a relação entre a codificação empregada e os sinais visuais?
Terá pequenos flash de luz, já que o sinal será é formado por pulsos, causados por um circuito que ora estará aberto e ora fechado, em um determinado intervalo de tempo.

Todos os símbolos codificados têm um mesmo número de elementos codificadores?
Não, pois tendo em vista que o código Morse tem como elementos codificadores apenas traços e pontos cada símbolo e definido por uma quantidade de elementos diferente dos outros.

É realizada uma uniformização de tamanho para a transmissão dos símbolos codificados?
Tendo em vista que o código Morse tem como elementos codificadores apenas traços e pontos, o número de elementos codificadores varia de símbolo a símbolo depende da quantidade de pontos e traços que o define.

É realizada uma uniformização de tempo para a transmissão dos símbolos codificados?
Não, pois o tempo necessário varia conforme o símbolo, depende da quantidade de pontos e traços que o define.

Que tipos de mensagens são transmitidas? Curtas e simplificadas (tipo telegrama)? Textos completos?
O emissor pode se sentir livre ao enviar a mensagem, desde que tanto o receptor quanto o emissor tenham domínio na codificação e decodificação do código, mas o aconselhado são mensagens curtas para evitar erros, já que toda a codificação é controlada por homens.

Qual é a duração temporal de um ponto?
0.04 segundos.

Qual é a duração temporal de um traço?
0.12 segundos.

Qual é o tempo entre os pontos e os traços para formar os símbolos codificados?

O tempo entre os símbolos de uma palavra equivale à 0,04 segundos(tempo do ponto).

Qual é o tempo entre letras para formar palavras?
O tempo entre as letras de uma palavra equivale à 0,12 segundos(3 vezes o tempo de um ponto).

Qual é o tempo entre palavras para formar frases?
O tempo entre as palavras em uma frase equivale à 0,28 segundos(7 vezes o tempo de um ponto).

Qual é o tempo entre frases para formar parágrafos?
O tempo entre as frases em um parágrafo equivale à 0,32 segundos(uma nova linha é codificada da seguinte forma: .-.-).

Qual é o tempo entre parágrafos para formar textos?
O tempo entre os parágrafos em um texto equivale à 0,36 segundos(um novo parágrafo é codificado da seguinte forma: -...-)

Como diferenciar as mensagens "EEE" e "S"?
As mensagens "EEE" e "S" apesar de serem escritas da mesma forma "..." apresentam leves alterações quando transmitidas. O tempo entre um ponto e outro em "EEE" é de 0,12 segundos, já o tempo entre os pontos da letra "S" é de 0,04 segundos.

Como diferenciar as mensagens "TTT" e "O"?
As mensagens "TTT" e "O" apesar de serem escritas da mesma forma "—" apresentam leves alterações quando transmitidas. O tempo entre um traço e outro em "TTT" é de 0,12 segundos, já o tempo entre os traços da letra "O" é de 0,04 segundos.

Como diferenciar as mensagens "EEEEE" e "5"?
As mensagens "EEEEE" e "5" apesar de serem escritas da mesma forma "....." apresentam leves alterações quando transmitidas. O tempo entre um ponto e outro em "EEEEE" é de 0,12 segundos, já o tempo entre os pontos do número "5" é de 0,04 segundos.

Como diferenciar as mensagens "TTTTT" e "0"?
As mensagens "TTTTT" e "0" apesar de serem escritas da mesma forma "——" apresentam leves alterações quando transmitidas. O tempo entre um traço e outro em "TTTT" é de 0,12 segundos, já o tempo entre os traços do número "0" é de 0,04.

9 Referências bibliográficas

https://www.suapesquisa.com/pesquisa/telegrafo.htm
https://sites.google.com/site/evolucaodacomunicacao15/capitulo-vii—telegrafo
http://minf.ufpa.br/index.php/telegrafo
https://www.infoescola.com/comunicacao/codigo-morse/
https://www.historiadetudo.com/telegrafo
https://ahistoriadacomunicacao.wordpress.com/2013/04/01/a-historia-do-telegrafo/comment-page-1/
https://historiadastransmissoes.wordpress.com/2012/09/23/linha-telegrafica-lisboacoimbra/3-tipos-de-telegrafos/
https://www.estudopratico.com.br/primeira-demonstracao-publica-do-telegrafo/
https://brasilescola.uol.com.br/geografia/codigo-morse.htm
https://mundoeducacao.bol.uol.com.br/geografia/meios-comunicacao.htm
https://www.newtoncbraga.com.br/index.php/projetos-educacionais/10144-sinais-telegraficos-e-codigo-morse-tel144

10 Anexos

10.1 Anexo 1

```
String traduzir(String letra){
  if(letra == "a"){
    return ".-";
  }
  if(letra == "b"){
    return "-...";
  }
  if(letra == "c"){
    return "-.-.";
  }
  if(letra == "d"){
    return "-..";
  }
  if(letra == "e"){
    return ".";
  }
  if(letra == "f"){
    return "..-.";
  }
  if(letra == "g"){
    return "--.";
  }
  if(letra == "h"){
    return "....";
  }
  if(letra == "i"){
    return "..";
  }
  if(letra == "j"){
    return ".---";
  }
  if(letra == "k"){
    return "-.-";
  }
  if(letra == "l"){
    return ".-..";
  }
  if(letra == "m"){
    return "--";
  }
  if(letra == "n"){
    return "-.";
  }
  if(letra == "o"){
    return "---";
  }
  if(letra == "p"){
    return ".--.";
  }
  if(letra == "q"){
    return "--.-";
  }
  if(letra == "r"){
    return "-.-";
  }
```

```
  if(letra == "s"){
    return "...";
  }
  if(letra == "t"){
    return "-";
  }
  if(letra == "u"){
    return "..-";
  }
  if(letra == "v"){
    return "...-";
  }
  if(letra == "w"){
    return ".--";
  }
  if(letra == "x"){
    return "-..-";
  }
  if(letra == "y"){
    return "-.--";
  }
  if(letra == "z"){
    return "--..";
  }
  if(letra == " "){
    return " ";
  }
}

void traduzLed(char codigo){
  delay(40);
  if(codigo == '.'){
    digitalWrite(8, HIGH);
    delay(40);
    digitalWrite(8, LOW);
  }
  if(codigo == ' '){
    digitalWrite(8, HIGH);
    delay(120);
    digitalWrite(8, LOW);
  }
  if(codigo == '-'){
    digitalWrite(8, HIGH);
    delay(120);
    digitalWrite(8, LOW);
  }
}

void traduzBuzzer(char codigo){
  delay(40);
  if(codigo == '.'){
    tone(2, 300);
    delay(40);
    noTone(2);
  }
  if(codigo == ' '){
    delay(120);
  }
  if(codigo == '-'){
```

```
    tone(2, 300);
    delay(120);
    noTone(2);
  }
}

String alfabeto = "abcdefghijklmnopqrstuvwxyz";
String morse = "";

void setup(){
  Serial.begin(9600);
  pinMode(8, OUTPUT); //LED
  pinMode(2, OUTPUT); //buzzer

}

void loop() {
  String entrada;
  int tamanho = 0;
  if (Serial.available() > 0){
     entrada = Serial.readString();
     while(tamanho < entrada.length()){
        morse = morse + traduzir(String(entrada[tamanho]));
        tamanho = tamanho + 1;
        if(tamanho == (entrada.length() - 1)){
          Serial.println(morse);
        }
     }
     tamanho = 0;
     while(tamanho < morse.length()){
        traduzLed(morse[tamanho]);
        tamanho = tamanho + 1;
     }
     tamanho = 0;
     while(tamanho < morse.length()){
        traduzBuzzer(morse[tamanho]);
        tamanho = tamanho + 1;
     }
     tamanho = 0;

  }
  morse = "";
  entrada = "";
  tamanho = 0;
}
```

Anexo 2

```
\\Emissor
String traduzir(String letra){
  if(letra == "a"){
    return ".-";
  }
  if(letra == "b"){
    return "-...";
  }
  if(letra == "c"){
    return "-.-.";
  }
  if(letra == "d"){
```

```
  return "-..";
}
if(letra == "e"){
  return ".";
}
if(letra == "f"){
  return "..-.";
}
if(letra == "g"){
  return "--.";
}
if(letra == "h"){
  return "....";
}
if(letra == "i"){
  return "..";
}
if(letra == "j"){
  return ".---";
}
if(letra == "k"){
  return "-.-";
}
if(letra == "l"){
  return ".-..";
}
if(letra == "m"){
  return "--";
}
if(letra == "n"){
  return "-.";
}
if(letra == "o"){
  return "---";
}
if(letra == "p"){
  return ".--.";
}
if(letra == "q"){
  return "--.-";
}
if(letra == "r"){
  return "-.-";
}
if(letra == "s"){
  return "...";
}
if(letra == "t"){
  return "-";
}
if(letra == "u"){
  return "..-";
}
if(letra == "v"){
  return "...-";
}
if(letra == "w"){
  return ".--";
}
```

```
  if(letra == "x"){
    return "-..-";
  }
  if(letra == "y"){
    return "-.--";
  }
  if(letra == "z"){
    return "--..";
  }
  if(letra == " "){
    return " ";
  }
}

void traduzLaser(char codigo){
  delay(40);
  if(codigo == '.'){
    digitalWrite(7, HIGH);
    delay(40);
    digitalWrite(7, LOW);
  }
  if(codigo == ' '){
    digitalWrite(7, HIGH);
    delay(120);
    digitalWrite(7, LOW);
  }
  if(codigo == '-'){
    digitalWrite(7, HIGH);
    delay(120);
    digitalWrite(7, LOW);
  }
}

String alfabeto = "abcdefghijklmnopqrstuvwxyz";
String morse = "";

void setup(){
  Serial.begin(9600);
  pinMode(7, OUTPUT) // Laser

}

void loop() {
  String entrada;
  int tamanho = 0;
  if (Serial.available() > 0){
     entrada = Serial.readString();
     while(tamanho < entrada.length()){
        morse = morse + traduzir(String(entrada[tamanho]));
        tamanho = tamanho + 1;
        if(tamanho == (entrada.length() - 1)){
          Serial.println(morse);
        }
     }
     tamanho = 0;

  }
  while(tamanho < morse.length()){
        traduzLaser(morse[tamanho]);
```

```
        tamanho = tamanho + 1;
  }
  morse = "";
  entrada = "";
  tamanho = 0;
}

\\Receptor
void calculaCaracter(unsigned long tempo, String mensagem){
  if(tempo < 80){
    mensagem += ".";
  }else{
    mensagem += "-";
  }

}

void calculaEspaco(unsigned long tempo, String mensagem){
  if(tempo > 80){
    mensagem += " ";
  }
}

int pinoReceptor = 9;
int leitura;
int laserOn = 0;
int laserOff = 0;
int inicio = 0;
int ativo = 0;
String mensagem = "";
unsigned long tempoInicio;
unsigned long tempoFim;
unsigned long tempoTotal;


void setup(){
  pinMode(pinoReceptor, INPUT); //DEFINE O PINO COMO SA DA
}

void loop(){
  leitura = digitalRead(pinoReceptor); //VARI VEL RECEBE VALOR LIDO NO RECEPTOR - 1(COM LA
  if(inicio == 0){
     if(leitura <= 500){
        ativo = 1;
        inicio = 1;
      }else{
        continue;
      }
  }else{
    if(leitura <= 500){
        ativo = 1;
        laserOff = 0;
      }else{
        ativo = 0;
      }
    }
    if(ativo == 1 && laserOn == 0){ // SE TIVER LASER
      tempoInicio = millis();
      laserOn = 1;
```

```
      delay(40);
    }else{
      if(ativo == 0 && laserOn == 1){ // 1ms ap s o laser desligar
        tempoFim = millis();
        tempoTotal = tempoFim - tempoInicio;
        calculaCaracter(tempoTotal, mensagem);
        laserOn = 0;
      }else{
        if(ativo == 0 && laserOff == 0){ // SE N O TIVER LASER
          tempoInicio = millis();
          laserOff = 1;
          delay(90);
          continue;
        }else{
          if(ativo == 0 && laserOff == 1){ // 1ms ap s o laser ligar
            mensagem += " "
            laserOff = 0;
           }
        }
      }
    }
  }
}
```

10.2 Anexo 3

```
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
%
% Grupo PET-Tele
%
%   Projeto Codigo Morse
%
%     - Gera alfabeto de padroes.
%     - Gera alfabeto de sons.
%
%     - Executa loop:
%
%       - Requisita texto codificavel em Morse.
%
%       - Codifica texto em padroes.
%       - Codifica texto em sons.
%
%       - Apresenta padrao Morse do texto fornecido.
%       - Executa   sons   Morse do texto fornecido.
%
%     - Ao sair do loop,
%       oferece opcao de salvar os alfabetos.
%
%
% Autor: Alexandre Santos de la Vega
%
% Data : /jul_2019/
%
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%



%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% acoes iniciais
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
```

```
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% limpa ambiente de trabalho
clear all

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%



%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% definicoes iniciais
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% constantes uteis

%
empty_char = '';
%
blank_char = '␣';

%
left_delimiter_char = '{';
%
right_delimiter_char = '}';

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%



%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% gera alfabeto de padroes
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% dot string
dot_char = 'o';

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% trace length
trace_len = 3;  % trace_time = 3 * dot_time

% trace string
trace_str = [empty_char];
for loop_ind = 1:trace_len
  trace_str = [trace_str '-'];
end
clear loop_ind

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
% 1 * dot_time
elem_sp_str   = [ blank_char ];
```

```
%
% 3 * dot_time
symbol_sp_str = [ elem_sp_str elem_sp_str elem_sp_str ];

%
% 7 * dot_time
word_sp_str   = [ elem_sp_str elem_sp_str elem_sp_str ...
                  elem_sp_str elem_sp_str elem_sp_str ...
                  elem_sp_str ];

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% titulo da lista de simbolos
%
%disp(blank_char); disp(blank_char);
%disp('Alfabeto de strings');
%disp(blank_char); disp(blank_char);

% letras
%
A_str = [ dot_char  elem_sp_str trace_str                ];
%
B_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
          dot_char  elem_sp_str dot_char                 ];
%
C_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
          trace_str elem_sp_str dot_char                 ];
%
D_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
          dot_char                                       ];
%
E_str = [ dot_char                                       ];
%
F_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
          trace_str elem_sp_str dot_char                 ];
%
G_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
          dot_char                                       ];
%
H_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
          dot_char  elem_sp_str dot_char                 ];
%
I_str = [ dot_char  elem_sp_str dot_char                 ];
%
J_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
          trace_str elem_sp_str trace_str                ];
%
K_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
          trace_str                                      ];
%
L_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
          dot_char  elem_sp_str dot_char                 ];
%
M_str = [ trace_str elem_sp_str trace_str                ];
%
N_str = [ trace_str elem_sp_str dot_char                 ];
%
O_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
          trace_str                                      ];
```

```
%
P_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
          trace_str elem_sp_str dot_char                ];
%
Q_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
          dot_char  elem_sp_str trace_str               ];
%
R_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
          dot_char                                      ];
%
S_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
          dot_char                                      ];
%
T_str = [ trace_str                                     ];
%
U_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
          trace_str                                     ];
%
V_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
          dot_char  elem_sp_str trace_str               ];
%
W_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
          trace_str                                     ];
%
X_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
          dot_char  elem_sp_str trace_str               ];
%
Y_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
          trace_str elem_sp_str trace_str               ];
%
Z_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
          dot_char  elem_sp_str dot_char                ];

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
disp(blank_char); disp(blank_char);

% base numerica
%
n1_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
           trace_str elem_sp_str trace_str elem_sp_str ...
           trace_str                                    ];
%
n2_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
           trace_str elem_sp_str trace_str elem_sp_str ...
           trace_str                                    ];
%
n3_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
           dot_char  elem_sp_str trace_str elem_sp_str ...
           trace_str                                    ];
%
n4_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
           dot_char  elem_sp_str dot_char  elem_sp_str ...
           trace_str                                    ];
%
n5_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
           dot_char  elem_sp_str dot_char  elem_sp_str ...
           dot_char                                     ];
```

```
%
n6_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
           dot_char  elem_sp_str dot_char  elem_sp_str ...
           dot_char                                    ];
%
n7_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
           dot_char  elem_sp_str dot_char  elem_sp_str ...
           dot_char                                    ];
%
n8_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
           trace_str elem_sp_str dot_char  elem_sp_str ...
           dot_char                                    ];
%
n9_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
           trace_str elem_sp_str trace_str elem_sp_str ...
           dot_char                                    ];
%
n0_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
           trace_str elem_sp_str trace_str elem_sp_str ...
           trace_str                                   ];

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
disp(blank_char); disp(blank_char);

% pontuacao
%
p_pnt_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
              dot_char  elem_sp_str trace_str elem_sp_str ...
              dot_char  elem_sp_str trace_str                 ];
%
p_frc_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
              dot_char  elem_sp_str trace_str elem_sp_str ...
              dot_char                                        ];
%
p_vrg_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
              dot_char  elem_sp_str dot_char  elem_sp_str ...
              trace_str elem_sp_str trace_str                 ];
%
p_2pt_str = [ trace_str elem_sp_str trace_str elem_sp_str ...
              trace_str elem_sp_str dot_char  elem_sp_str ...
              dot_char  elem_sp_str dot_char                  ];
%
p_aps_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
              trace_str elem_sp_str trace_str elem_sp_str ...
              trace_str elem_sp_str dot_char                  ];
%
p_hfn_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
              dot_char  elem_sp_str dot_char  elem_sp_str ...
              dot_char  elem_sp_str trace_str                 ];
%
p_rpr_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
              trace_str elem_sp_str dot_char  elem_sp_str ...
              dot_char  elem_sp_str trace_str                 ];
%
p_lpr_str = [ trace_str elem_sp_str dot_char  elem_sp_str ...
              trace_str elem_sp_str dot_char  elem_sp_str ...
              dot_char  elem_sp_str                           ];
```

```
%
p_asp_str = [ dot_char  elem_sp_str trace_str elem_sp_str ...
              dot_char  elem_sp_str dot_char  elem_sp_str ...
              trace_str elem_sp_str dot_char                 ];
%
p_int_str = [ dot_char  elem_sp_str dot_char  elem_sp_str ...
              trace_str elem_sp_str trace_str elem_sp_str ...
              dot_char  elem_sp_str dot_char                 ];

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%

% gera linhas em branco para terminar
disp(blank_char); disp(blank_char);

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%



%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
% gera alfabeto de sons
%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%

% definicao das frequencias

%
Fs = 8e3;  % sampling   frequency
Fm = 1e3;  % sinusoidal frequency

%
Omega = 2*pi*Fm/Fs;  % Omega = w Ts = 2 pi f Ts = 2 pi f / Fs

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%

% definicao dos tempos

% for testing
dot_time   = 0.08;
%
%
% defined by Alyne/Lucio
%dot_time   = 0.04;
trace_time = 3 * dot_time;
%
% defined by ASV
%morse_symbols_space_time = 2 * dot_time;
%char_space_time          = 4 * dot_time;
%word_sp_vace_time          = 8 * dot_time;

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
% gera alfabeto
%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%

%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%%%
```

```
% construcao do som do ponto

%
% Nbr of samples per dot_time
Ndot = dot_time * Fs;
%
nd = 0:(Ndot-1);
dot_snd = cos(Omega * nd);

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% construcao do som do traco

%
% Nbr of samples per trace_time
Ntrace = trace_time * Fs;
%
nt = 0:(Ntrace-1);
trace_snd = cos(Omega * nt);

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% construcao do som dos elementos de espacamento

%
% 1 * dot_time
elem_sp_snd    = zeros(1,Ndot);

%
% 3 * dot_time
symbol_sp_snd = zeros(1,Ntrace);

%
% 7 * dot_time
word_sp_snd    = zeros(1,7*Ndot);

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
%disp(blank_char); disp(blank_char);

% letras
%
A_snd = [ dot_snd   elem_sp_snd trace_snd                ];
%
B_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
          dot_snd   elem_sp_snd dot_snd                  ];
%
C_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
          trace_snd elem_sp_snd dot_snd                  ];
%
D_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
          dot_snd                                        ];
%
E_snd = [ dot_snd                                        ];
%
F_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
          trace_snd elem_sp_snd dot_snd                  ];
```

```
%
G_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
          dot_snd                                      ];
%
H_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
          dot_snd   elem_sp_snd dot_snd                ];
%
I_snd = [ dot_snd   elem_sp_snd dot_snd                ];
%
J_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
          trace_snd elem_sp_snd trace_snd              ];
%
K_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
          trace_snd                                    ];
%
L_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
          dot_snd   elem_sp_snd dot_snd                ];
%
M_snd = [ trace_snd elem_sp_snd trace_snd              ];
%
N_snd = [ trace_snd elem_sp_snd dot_snd                ];
%
O_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
          trace_snd                                    ];
%
P_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
          trace_snd elem_sp_snd dot_snd                ];
%
Q_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
          dot_snd   elem_sp_snd trace_snd              ];
%
R_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
          dot_snd                                      ];
%
S_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
          dot_snd                                      ];
%
T_snd = [ trace_snd                                    ];
%
U_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
          trace_snd                                    ];
%
V_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
          dot_snd   elem_sp_snd trace_snd              ];
%
W_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
          trace_snd                                    ];
%
X_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
          dot_snd   elem_sp_snd trace_snd              ];
%
Y_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
          trace_snd elem_sp_snd trace_snd              ];
%
Z_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
          dot_snd   elem_sp_snd dot_snd                ];

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
```

```
%
%disp(blank_char); disp(blank_char);

% base numerica
%
n1_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd                                    ];
%
n2_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd                                    ];
%
n3_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd                                    ];
%
n4_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           trace_snd                                    ];
%
n5_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd                                      ];
%
n6_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd                                      ];
%
n7_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd                                      ];
%
n8_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
           dot_snd                                      ];
%
n9_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           dot_snd                                      ];
%
n0_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd elem_sp_snd trace_snd elem_sp_snd ...
           trace_snd                                    ];

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
%disp(blank_char); disp(blank_char);

% pontuacao
%
p_pnt_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
              dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
              dot_snd   elem_sp_snd trace_snd                ];
%
p_frc_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
              dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
              dot_snd                                      ];
```

```
%
p_vrg_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
              dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
              trace_snd elem_sp_snd trace_snd                 ];
%
p_2pt_snd = [ trace_snd elem_sp_snd trace_snd elem_sp_snd ...
              trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
              dot_snd   elem_sp_snd dot_snd                   ];
%
p_aps_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
              trace_snd elem_sp_snd trace_snd elem_sp_snd ...
              trace_snd elem_sp_snd dot_snd                   ];
%
p_hfn_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
              dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
              dot_snd   elem_sp_snd trace_snd                 ];
%
p_rpr_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
              trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
              dot_snd   elem_sp_snd trace_snd                 ];
%
p_lpr_snd = [ trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
              trace_snd elem_sp_snd dot_snd   elem_sp_snd ...
              dot_snd   elem_sp_snd                           ];
%
p_asp_snd = [ dot_snd   elem_sp_snd trace_snd elem_sp_snd ...
              dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
              trace_snd elem_sp_snd dot_snd                   ];
%
p_int_snd = [ dot_snd   elem_sp_snd dot_snd   elem_sp_snd ...
              trace_snd elem_sp_snd trace_snd elem_sp_snd ...
              dot_snd   elem_sp_snd dot_snd                   ];

%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%



%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%
% executa loop de execucao
%   inicio do loop
%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%
%
%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%
do
%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%
%
%
%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%
% requisita texto codificavel em Morse
%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%

%%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%

%
disp(empty_char); disp(empty_char);

%
% interface grafica para leitura do texto
%
```

```
prompt   = {"Texto"};
title    = "Digite o texto a ser codificado";
rowscols = [1, 60];
defaults = {"Digite aqui e pressione OK."};
%
cell_array_str = inputdlg(prompt, title, rowscols, defaults);

% testa de usuario escolheu OK ou CANCEL
if ( isempty(cell_array_str) )
  %
  disp(empty_char); disp(empty_char);
  disp("Aplicativo finalizado corretamente.");
  disp(empty_char); disp(empty_char);
  %
  return
end

%
text_to_translate = cell_array_str{1,1};
%
disp(empty_char); disp(empty_char);
disp('Texto digitado:');
disp([ '{', text_to_translate, '}' ]);
disp(empty_char);

% for testing...
%
%text_to_translate = ...
%    input('Digite o texto a ser codificado:  ');

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%


%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% codifica texto em padroes
% codifica texto em sons
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
%
text_len = length (text_to_translate);

% inicializacao dos vetores de codificacao
%
% inicializacao para padrao textual
morse_str      = [ left_delimiter_char ];
stretch_sp_str = empty_char;
%
% inicializacao para sons
morse_snd      = [];
stretch_sp_snd = [];

%
for ind_s = 1:text_len
  symbol_to_translate = text_to_translate(ind_s);
  %
  if (symbol_to_translate ~= blank_char)
    morse_str = [ morse_str stretch_sp_str ];
    morse_snd = [ morse_snd stretch_sp_snd ];
```

```
end
%
switch (symbol_to_translate)
  %
  case {blank_char}
    morse_str = [ morse_str word_sp_str ];
    morse_snd = [ morse_snd word_sp_snd ];
  %
  case {'a' 'A'}
    morse_str = [ morse_str A_str ];
    morse_snd = [ morse_snd A_snd ];
  case {'b' 'B'}
    morse_str = [ morse_str B_str ];
    morse_snd = [ morse_snd B_snd ];
  case {'c' 'C'}
    morse_str = [ morse_str C_str ];
    morse_snd = [ morse_snd C_snd ];
  case {'d' 'D'}
    morse_str = [ morse_str D_str ];
    morse_snd = [ morse_snd D_snd ];
  case {'e' 'E'}
    morse_str = [ morse_str E_str ];
    morse_snd = [ morse_snd E_snd ];
  case {'f' 'F'}
    morse_str = [ morse_str F_str ];
    morse_snd = [ morse_snd F_snd ];
  case {'g' 'G'}
    morse_str = [ morse_str G_str ];
    morse_snd = [ morse_snd G_snd ];
  case {'h' 'H'}
    morse_str = [ morse_str H_str ];
    morse_snd = [ morse_snd H_snd ];
  case {'i' 'I'}
    morse_str = [ morse_str I_str ];
    morse_snd = [ morse_snd I_snd ];
  case {'j' 'J'}
    morse_str = [ morse_str J_str ];
    morse_snd = [ morse_snd J_snd ];
  case {'k' 'K'}
    morse_str = [ morse_str K_str ];
    morse_snd = [ morse_snd K_snd ];
  case {'l' 'L'}
    morse_str = [ morse_str L_str ];
    morse_snd = [ morse_snd L_snd ];
  case {'m' 'M'}
    morse_str = [ morse_str M_str ];
    morse_snd = [ morse_snd M_snd ];
  case {'n' 'N'}
    morse_str = [ morse_str N_str ];
    morse_snd = [ morse_snd N_snd ];
  case {'o' 'O'}
    morse_str = [ morse_str O_str ];
    morse_snd = [ morse_snd O_snd ];
  case {'p' 'P'}
    morse_str = [ morse_str P_str ];
    morse_snd = [ morse_snd P_snd ];
  case {'q' 'Q'}
    morse_str = [ morse_str Q_str ];
    morse_snd = [ morse_snd Q_snd ];
```

```
case {'r' 'R'}
  morse_str = [ morse_str R_str ];
  morse_snd = [ morse_snd R_snd ];
case {'s' 'S'}
  morse_str = [ morse_str S_str ];
  morse_snd = [ morse_snd S_snd ];
case {'t' 'T'}
  morse_str = [ morse_str T_str ];
  morse_snd = [ morse_snd T_snd ];
case {'u' 'U'}
  morse_str = [ morse_str U_str ];
  morse_snd = [ morse_snd U_snd ];
case {'v' 'V'}
  morse_str = [ morse_str V_str ];
  morse_snd = [ morse_snd V_snd ];
case {'w' 'W'}
  morse_str = [ morse_str W_str ];
  morse_snd = [ morse_snd W_snd ];
case {'x' 'X'}
  morse_str = [ morse_str X_str ];
  morse_snd = [ morse_snd X_snd ];
case {'y' 'Y'}
  morse_str = [ morse_str Y_str ];
  morse_snd = [ morse_snd Y_snd ];
case {'z' 'Z'}
  morse_str = [ morse_str Z_str ];
  morse_snd = [ morse_snd Z_snd ];
%
case {'0'}
  morse_str = [ morse_str n0_str ];
  morse_snd = [ morse_snd n0_snd ];
case {'1'}
  morse_str = [ morse_str n1_str ];
  morse_snd = [ morse_snd n1_snd ];
case {'2'}
  morse_str = [ morse_str n2_str ];
  morse_snd = [ morse_snd n2_snd ];
case {'3'}
  morse_str = [ morse_str n3_str ];
  morse_snd = [ morse_snd n3_snd ];
case {'4'}
  morse_str = [ morse_str n4_str ];
  morse_snd = [ morse_snd n4_snd ];
case {'5'}
  morse_str = [ morse_str n5_str ];
  morse_snd = [ morse_snd n5_snd ];
case {'6'}
  morse_str = [ morse_str n6_str ];
  morse_snd = [ morse_snd n6_snd ];
case {'7'}
  morse_str = [ morse_str n7_str ];
  morse_snd = [ morse_snd n7_snd ];
case {'8'}
  morse_str = [ morse_str n8_str ];
  morse_snd = [ morse_snd n8_snd ];
case {'9'}
  morse_str = [ morse_str n9_str ];
  morse_snd = [ morse_snd n9_snd ];
%
```

```
    case {'.'}
      morse_str = [ morse_str p_pnt_str ];
      morse_snd = [ morse_snd p_pnt_snd ];
    case {'/'}
      morse_str = [ morse_str p_frc_str ];
      morse_snd = [ morse_snd p_frc_snd ];
    case {','}
      morse_str = [ morse_str p_vrg_str ];
      morse_snd = [ morse_snd p_vrg_snd ];
    case {':'}
      morse_str = [ morse_str p_2pt_str ];
      morse_snd = [ morse_snd p_2pt_snd ];
    case {"'"}
      morse_str = [ morse_str p_aps_str ];
      morse_snd = [ morse_snd p_aps_snd ];
    case {'-'}
      morse_str = [ morse_str p_hfn_str ];
      morse_snd = [ morse_snd p_hfn_snd ];
    case {')'}
      morse_str = [ morse_str p_rpr_str ];
      morse_snd = [ morse_snd p_rpr_snd ];
    case {'('}
      morse_str = [ morse_str p_lpr_str ];
      morse_snd = [ morse_snd p_lpr_snd ];
    case {'"'}
      morse_str = [ morse_str p_asp_str ];
      morse_snd = [ morse_snd p_asp_snd ];
    case {"?"}
      morse_str = [ morse_str p_int_str ];
      morse_snd = [ morse_snd p_int_snd ];
    %
    otherwise
      disp(empty_char); disp(empty_char);
      err_msg = [ "Invalid value: <",  ...
                  symbol_to_translate, ...
                  ">  !!!" ]
      error (err_msg);
    %
  endswitch
  %
  if (symbol_to_translate == blank_char)
    stretch_sp_str = empty_char;
    stretch_sp_snd = [];
    else
    stretch_sp_str = symbol_sp_str;
    stretch_sp_snd = symbol_sp_snd;
  end
  %
end

% insere elementos extras
%
% coloca delimitadores,
%   para melhor visualizacao...
morse_str = [ morse_str right_delimiter_char ];
%
% coloca espaco final,
%   para que a placa de som
%   termine o som adequadamente ...
```

```
%     (empiricamente percebido !!!)
morse_snd = [ morse_snd word_sp_snd ];

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%


%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% apresenta padrao Morse do texto fornecido
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
disp(empty_char); disp(empty_char);

%
disp('Texto codificado em Morse:');
disp(morse_str)

%
disp(empty_char); disp(empty_char);

%
title = "Texto codificado em Morse";
msg   = {"Texto original:",
         blank_char,
         ['{', text_to_translate, '}'],
         blank_char,
         blank_char,
         "Texto codificado: ",
         blank_char,
         morse_str
         blank_char,
         blank_char,
         "Pressione OK para ouvir.",
        };
helpdlg (msg, title);

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%


%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% executa sons Morse do texto fornecido
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
soundsc(morse_snd,Fs)

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%


%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% executa loop de execucao
%   fim do loop
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% controle de fim do loop
```

```
%
% BTN = questdlg (MSG, TITLE, BTN1, BTN2, BTN3, DEFAULT)
%
msg     = "O que deseja agora?";
title   = "Codificador Morse";
b_pos   = "Codificar outro texto";
b_neg   = "Sair";
default = b_pos;
%
choice_y_n_y = questdlg (msg, title, b_pos, b_neg, default);

until ( strcmp(choice_y_n_y, b_neg) )

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% oferece opcao de salvar os alfabetos
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
%save morse_padroes_e_sons_dat

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
% finaliza o aplicativo
%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

% informa o final do aplicativo
disp(empty_char); disp(empty_char);
disp("Aplicativo finalizado corretamente.");
disp(empty_char); disp(empty_char);

%%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%

%
% EOF
%
```