

## Gerador de senhas

**Nome dos participantes:** Nathan Furchi Alvisi e Wesley Rodrigues Mendes.

**Nome do projeto:** Gerador de Senhas.

**Objetivo de projeto:** Criar um display que gera senhas aleatórias.

**Justificativa do projeto:** Gerar um captcha aleatoria que será mostrada no display, com finalidade de ser usado em authenticator de contas.

**Ex:** Google Authenticator, Authy, aplicativos que guardam bem a sua conta, impedindo usos indevidos.

**Referência:** <https://tecdicas.com/como-ligar-um-display-lcd-no-arduino-uno/>

## Como ligar um display LCD no Arduino Uno:

### Materiais:

- 1 Arduino Uno Rev 3 com cabo USB;
- 1 Display LCD de fundo azul 16x2;
- 1 Resistor de 1 Kohms;
- 2 Resistores de 220 ohms;
- Cabos jumper Macho – Macho;
- 1 Protoboard de 830 furos;

### Como montar?

Com estes **materiais em mãos** podemos **iniciar a montagem do display LCD no protoboard**, **encaixe** ele deixando um **espaço para os cabos jumpers serem ligados**.

- O **pino 1 (VSS)** é ligado ao **GND**;
- O **pino 2 (VDD)** é ligado ao **5V**;
- O **pino 3 (VO)** é responsável pelo **ajuste do contraste do display**, para isso coloque o **resistor de 1K**, que será **ajustado na medida certa** para **não saturar ou apagar as letras e números do display**;
- O **pino 4 (RS)** é responsável pelo **registrador de seleção do sinal**, é ligado no **pino 12**;
- O **pino 5 (RW)** representa o **sinal de escrita e leitura do display**, é ligado ao **GND**;
- O **pino 6 (E)** **habilita ou desabilita o sinal**, é ligado no **pino 13**;

Os **8 pinos** representados por **B** são responsáveis pelo barramento **Data Bus**, que transportam dados de entrada e saída (I/O) do display em 8 bits.

- Os pinos de 7 ao 10 não são utilizados, pois o **Data Bus** será setado como 4 bits.
- O pino 11 (D4) é ligado no pino 7;

////////////////////////////////////\Z

- O pino 12 (D5) é ligado ao pino 6;
- O pino 13 (D6) é ligado ao pino 5;
- O pino 14 (D7) é ligado ao pino 4;

Os pinos 15 (LED+) e 16 (LED-) são responsáveis por fornecer energia aos LEDs que ficam ao fundo do display para você enxergar o que está escrito nele. O pino 16 é ligado ao **GND** e existem três maneiras de realizar a ligação do pino 15, faça o que você achar mais adequado ao seu display.

1. Utilize os **2 resistores de 220 ohms em paralelo**, fornecendo um total de **110R** ao pino 15.
2. Ligue o pino 15 ao **5V**, retire o resistor de **1K** e coloque um **potenciômetro de 25k** conectado ao **5V**, **GND** e no pino 3 do display. O potenciômetro que também é um resistor, porém variável, ele controla os valores dos LEDs e do contraste.
3. Simplesmente ligue o pino 15 ao **5V** sem potenciômetro de ajuste.

### Programando no Arduino:

Com o display conectado corretamente ao Arduino, agora é possível escrever qualquer palavra no display, ele aceita números e letras em minúsculo e maiúsculo, porém a biblioteca não reconhece a acentuação da nossa linguagem.

1. **Conecte** o seu **Arduino Uno** no computador;
2. **Abra a IDE** oficial do **Arduino**, versão utilizada 1.8.3;
3. Na aba <Ferramentas - Placa:"Arduino/Genuino Uno" - Porta (**Sua Porta**)">;
4. Clique no ícone "**Novo**" para **abrir um novo Sketch**.

A biblioteca utilizada neste projeto foi a **LiquidCrystal**, para incluir ao seu **Sketch** basta acessar o menu de abas <Sketch - Incluir Biblioteca - **LiquidCrystal**>, fazendo este processo ela implementará em seu projeto atual.

Logo em seguida cole este código para controlar o seu display e salve o arquivo .ino:

```

/*
 * Firmware para controlar um display LCD 16x2 by tecnicas
 * Biblioteca/Library : LiquidCrystal
 * 27/10/2017
 */
#include <LiquidCrystal.h> // Adiciona a biblioteca "LiquidCrystal" ao projeto

LiquidCrystal lcd(12, 13, 7, 6, 5, 4); // Pinagem do LCD

void setup()
{
  lcd.begin(16, 2); // Inicia o lcd de 16x2
}

void loop()
{
  lcd.clear();           // Limpa o display
  lcd.setCursor(2, 0);    // 2 = 2 colunas para a direita. 0 = Primeira linha
  lcd.print("Display LCD"); // Imprime um texto
  lcd.setCursor(2, 1);    // 2 = 2 colunas para a direita. 1 = Segunda linha
  lcd.print("* tecnicas.com *");
  delay(5000);           // 5 segundos de delay

  lcd.clear();
  lcd.setCursor(3, 0);
  lcd.print("Ola Mundo!");
  delay(5000);
}

```

## Pra que serve cada linha de código?

1. Utilizamos a **biblioteca "LiquidCrystal"** para **controlar um display de LCD de 16 colunas por 2 linhas**.

```
#include <LiquidCrystal.h>
```

2. Essa linha **cria um objeto "lcd"** do tipo **"LiquidCrystal"** e especifica os **pinos utilizados para a conexão entre o Arduino e o display**. Utilizamos os **pinos (12, 13, 7, 6, 5, 4)**.

```
LiquidCrystal lcd(12, 13, 7, 6, 5, 4);
```

3. A **função "setup()"** é chamada uma vez no **início do programa**. Aqui, é **inicializado o display chamando a função begin(16, 2)**, que **informa ao display que ele possui 16 colunas e 2 linhas**. Também é **utilizada a função "randomSeed(millis)" para iniciar a semente aleatória com base em uma leitura analógica do pino A0**. Isso garante que a **sequência de números aleatórios gerada seja diferente a cada execução do programa**.

```
void setup() {
  lcd.begin(16, 2);
  randomSeed(millis());
}
```

4. A função “loop()” é executada repetidamente após o término da função “setup()”. Aqui, o código realiza as seguintes operações:

- (1) Limpa o display utilizando “lcd.clear()”.
- (2) Define a posição do cursor no display utilizando “lcd.setCursor(3, 0)”. Nesse caso, o cursor é movido para a coluna 3 e linha 0.
- (3) Imprime a mensagem "Seu Token:" utilizando “lcd.print("Seu Token")”.
- (4) Gera uma string aleatória chamando a função “gerarStringAleatoria(6)”, que retorna uma string aleatória com 6 caracteres.
- (5) Define a posição do cursor para a segunda linha do display utilizando “lcd.setCursor(5, 1)”.
- (6) Imprime a string aleatória gerada no passo anterior utilizando “lcd.print(token)”.
- (7) Aguarda 5 segundos antes de repetir o processo (para demonstração).

```
void loop() {
  lcd.clear();

  lcd.setCursor(3, 0);
  lcd.print("Seu Token:");

  String token = gerarStringAleatoria(6);
  lcd.setCursor(5, 1);
  lcd.print(token);

  delay(10000);
}
```

5. Essa função “gerarStringAleatoria(int tamanho)” recebe um parâmetro “tamanho” que indica o número de caracteres que a string aleatória deve ter. A função cria uma string “caracteres” contendo todos os caracteres possíveis que podem ser selecionados aleatoriamente. Em seguida, ela itera “tamanho” vezes, selecionando um índice aleatório da string de caracteres e adicionando o caractere correspondente à string aleatória resultante. Por fim, retorna a string aleatória gerada.

```
String gerarStringAleatoria(int tamanho) {  
    String caracteres = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";  
    String token = "";  
  
    for (int i = 0; i < tamanho; i++) {  
        int indice = random(caracteres.length());  
        token += caracteres.charAt(indice);  
    }  
    return token;  
}
```

No geral, esse código **exibe no display de LCD** uma mensagem "Seu Token:" seguida de uma **sequência aleatória de 6 caracteres**. A cada **10 segundos**, uma **nova sequência é gerada e exibida no display**.