

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

LiquidCrystal lcd(53, 52, 51, 50, 49, 48);

// Basico
byte modo = 0;

// Carga Datos
bool pinAsignado = false;
bool pinLeido = false;
byte pinGuardado[7];

// Nuevo Pin
byte pin = 0;
byte pinMemoria[7];
byte pinX = 0;

void setup() {

    // Inicializa la serial e imprime
    Serial.begin(9600);
    while (!Serial) {
        ;
    }
    lcd.begin(16, 2);
    lcd.blink();
    intro();
}

void loop() {
    if (Serial.available() > 0) {
        modo = Serial.read();
        Serial.println(modo + "\n");

        switch (modo) {
            case 49:
                // Limpia la pantalla e informa al usuario del pedido de nuevo pin
                clearLCD();
                writeLCD(0, 0, "Nuevo PIN:");
                writeLCD(0, 1, ">");

                // Mientras no haya sido asignado
                while (pinAsignado == false) {
                    pinAsignado = nuevoPIN();
                }

                clearLCD();
                intro();
                break;
            case 50:
                while (pinLeido == false) {
                    pinLeido = leerPIN();
                }

                clearLCD();
                intro();

                break;
            default:
                // statements
                break;
        }
    }
}

/*  Funciones extras, no necesarias para el funcionamiento de la caja fuerte
 *    intro - Muestra por serial y por LCD el mensaje de introduccion.
 */

void intro(){
    Serial.println();
    Serial.print(F("Bienvenido a la terminal de la Caja Fuerte.\n"));
    Serial.print(F("Puedes ingresar un numero de 6 digitos como pin.\n"));
    Serial.print(F("Se limpia la pantalla usando la tecla [C].\n"));
    Serial.print(F("Modos:\n"));
    Serial.print(F("[1] - Nuevo PIN\n"));
    Serial.print(F("[2] - Leer PIN\n"));
    delay(100);
    writeLCD(0, 0, "Ingrese modo:");
    writeLCD(0, 1, ">");
}
```

```
}

/* Funciones para control del LCD:
 *   clearLCD   - Limpia todos los caracteres en el LCD.
 *   writeLCD   - Fija posicion y escribe texto.
 *   bwriteLCD  - Identico a writeLCD pero recibe texto en formato ASCII.
 */

void clearLCD() {
  for(byte x = 0; x <= 15; x++) {
    for(byte y = 0; y <= 1; y++) {
      writeLCD(x, y, " ");
    }
  }
}

void writeLCD(byte x, byte y, const char* txt) {
  lcd.setCursor(x, y);
  lcd.write(txt);
}

void bwriteLCD(byte x, byte y, byte txt) {
  lcd.setCursor(x, y);
  lcd.write(txt);
}

/*
 * Funciones para control del PIN
 * nuevoPIN - Utilizado para guardar un nuevo pin en la memoria eeprom.
 * leerPIN  - Accede a la memoria eeprom y despliega el pin almacenado.
 */

bool nuevoPIN() {
  // ¿Existe algun caracter? Si es asi guardarlo en memoria
  if (Serial.available() > 0) {
    pin = Serial.read();

    // ¿Es un numero? Si es asi imprimir en pantalla y serial el numero.
    if ((pin >= 48) && (pin <= 57)) {
      bwriteLCD((pinX + 1), 1, pin);
      pinMemoria[pinX] = pin;
      pinX = pinX + 1;
      if (pinX == 6) {
        for (int addr = 0; addr <= 5; addr++) {
          EEPROM.write(addr, pinMemoria[addr]);
        }
        return true;
      }
    }

    if (pin == 67) {
      for (byte j = 1; j <= 15; j++) {
        writeLCD(j, 1, " ");
      }
      Serial.println("Clear");
    }
  }
  return false;
}

bool leerPIN() {
  Serial.print("PIN: ");
  for (int addr = 0; addr <= 5; addr++) {
    int value = EEPROM.read(addr);
    Serial.print(value, DEC);
    if (addr < 5) {
      Serial.print(" - ");
    }
  }
  Serial.println();
  return true;
}
```