```
//Incluir Bibliotecas Específicas.
#include<SoftwareSerial.h>
#include "SIM900.h"
#include "sms.h"

//Declaração de Classe Global.
SMSGSM sms;

//Declaração de variáveis.
char number[20];
char message[160];
char pos;
char *a1, *a2, *b1, *b2, *c1, *c2, *d1, *d2;
char  security[] ="Houve Uma Tentativa de Acesso Nao Autorizado - Veja o Log"
int error;

//Configuração inical do Arduino.
void setup()
{
 //Selecionandos pino de saida e comunicação serial com PC.
 pinMode(13,OUTPUT);
 pinMode(12,OUTPUT);
 pinMode(11,OUTPUT);
 pinMode(10,OUTPUT);
 Serial.begin(9600);

 //Iniciando Comunicação Serial com a Shield.
 if (gsm.begin(9600))
   Serial.println("\nStatus = PRONTO");
   else Serial.println("\nStatus = ESPERA");

 //Deletando todas as mensagens do SIMcard.
 sms.DeleteSMS(SMS_ALL);
}

//Configurando Loop de Ações.
void loop()
{
 /*Defini a variável pos com a
 posição na memória da mesnagem não lida.*/
 pos = sms.IsSMSPresent(SMS_UNREAD);
 Serial.println("Aguardando Comando...");

 /*Testa se tem sms recebida não lida,
 verificando se o número que enviou é registrado.*/
 if((int)pos>0){
   if(GETSMS_AUTH_SMS == sms.GetAuthorizedSMS((int)pos, number, message, 50, 1
   Serial.print("NOVA MENSAGEM AUTORIZADA, POS=");
   Serial.print((int)pos);
```

```
//Defini o conteúdo da string message como vazio.
message[0]='\0';

//Ler a mensagem não lida e carrega os parametros de comparação.
sms.GetSMS((int)pos,number,message,180);

a1=strstr(message,"OnA");
a2=strstr(message,"OffA");

b1=strstr(message,"OnB");
b2=strstr(message,"OffB");

c1=strstr(message,"OnC");
c2=strstr(message,"OffC");

d1=strstr(message,"OnD");
d2=strstr(message,"OffD");

// Bloco de If/Else para decidir qual comando fazer.
if(a1){
  Serial.println(" - Led On");
  Serial.println("Ambiente A");
  digitalWrite(13,HIGH);}

if(a2){
  Serial.println(" - Led off");
  Serial.println("Ambiente A");
  digitalWrite(13,LOW);}

if(b1){
  Serial.println(" - Led On");
  Serial.println("Ambiente B");
  digitalWrite(12,HIGH);}

if(b2){
  Serial.println(" - Led off");
  Serial.println("Ambiente B");
  digitalWrite(12,LOW);}


if(c1){
  Serial.println(" - Led On");
  Serial.println("Ambiente C");
  digitalWrite(11,HIGH);}

if(c2){
  Serial.println(" - Led off");
  Serial.println("Ambiente C");
```

```
      digitalWrite(11,LOW);}

    if(d1){
      Serial.println(" - Led On");
      Serial.println("Ambiente D");
      digitalWrite(10,HIGH);}

    if(d2){
      Serial.println(" - Led off");
      Serial.println("Ambiente D");
      digitalWrite(10,LOW);}

    if(!a1 && !a2 && !b1 && !b2 && !c1 && !c2 && !d1 && !d2){
      Serial.println(" - Mensagem Error");}

  //Após ação deleta as mensagens, novamente.
  sms.DeleteSMS(SMS_ALL);

  /*Caso o número que enviou a mensagem não está cadsatrado,
  o sitema envia uma mensagem de segurança ao número proprietário.*/
  }else {
   Serial.print("NOVA MENSAGEM NAO AUTORIZADA, POS=");
   Serial.println((int)pos);
   Serial.print("ENVIANDO MENSAGEM DE SEGURANCA PARA: ");
   Serial.println("+556191942518");
   //Função padrão para envio de sms. (numero de destino, string com texto)
   error=sms.SendSMS("+556191942518", security);
     if (error==0){
        Serial.println("SMS NAO ENVIADO");}
        else{
        Serial.println("SMS ENVIADO");}
   sms.DeleteSMS(SMS_ALL);
  }}
  delay(7000);
}
```