



# COMMUNICATION SANS FIL

## SERRURE CONNECTÉE

# SOMMAIRE

- 01 - PRÉSENTATION GÉNÉRALE
- 02 - OBJECTIF PRINCIPALE
- 03 - OBJECTIF SECONDAIRE
- 04 - PLANNING
- 05 - PRÉSENTATION COMPOSANT
- 06 - PROGRAMME
- 07 - PHASE DE TEST
- 08 - CONCLUSION
- 09 - AMÉLIORATIONS



# PRESENTATION GÉNÉRALE

☆ NOTRE PROJET EST UN VERROU ELECTRIQUE DE PORTE A EMPREINTE DIGITALE S'OUVRANT PAR CAPTEUR D'EMPREINTE DIGITALE.

☆ LA SERRURE CONNECTÉE SERA CAPABLE DE COMMUNIQUER PAR MAIL OU PAR MESSAGE POUR L'OUVERTURE, DE PLUS IL EST POSSIBLE DE DEVEROUILLER LA SERRURE DIRECTEMENT PAR SMARTPHONE.

☆ IL SERA POSSIBLE DE COMMUNIQUER AVEC LA SERRURE PAR PLUSIEURS MOYENS DE COMMUNICATION SANS FILS NOTAMMENT PAR LORA, WIFI OU BLUETOOTH.

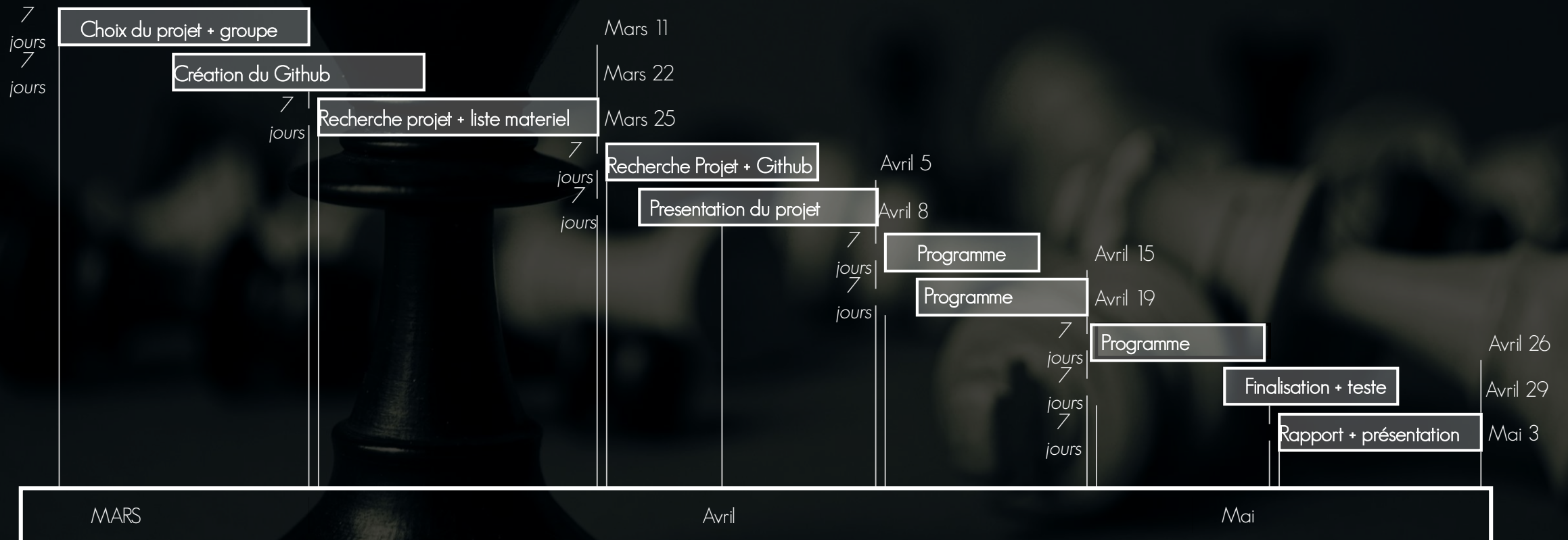
# OBJECTIFS

## SECONDAIRES

- ☆ **UTILISER LE MODULE NFC POUR POUVOIR UTILISER LE SYSTEME DE DEVEROUILLAGE**
- ☆ **UTILISER LE MODULE BLUETOOTH POUR COMMUNIQUER AVEC LE SYSTEME**
- ☆ **UTILISER LE CAPTEUR D'EMPREINTE D'UN SMARTPHONE POUR DEVEROUILLER LA SERRURE**



# PLANNING



# PRÉSENTATION COMPOSANTS



**ESP32 TTGO**

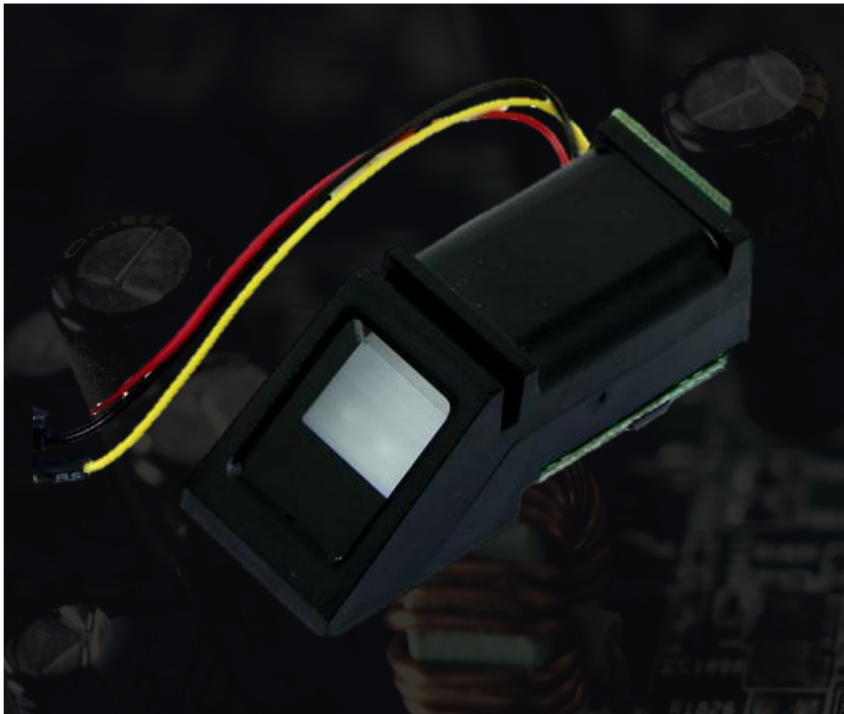
L'ESP32 est une puce combinée Wi-Fi et Bluetooth unique à 2,4 GHz conçue avec le TSMC ultra-basse consommation.

C'est une carte électronique équipée d'un microcontrôleur, l'esp32 peut très bien jouer le rôle d'une carte Arduino.

Ce microcontrôleur dispose d'interfaces Wifi, Bluetooth et Lora.



## MODULE D'IDENTIFICATION DU LECTEUR D'EMPREINTES DIGITALES



Permet la collecte des empreintes digitales, l'enregistrement des empreintes digitales, la comparaison des empreintes digitales et la recherche d'empreintes digitales.

## SERVO MOTEUR



Un servomoteur est un système qui a pour but de produire un mouvement précis en réponse à une commande externe.

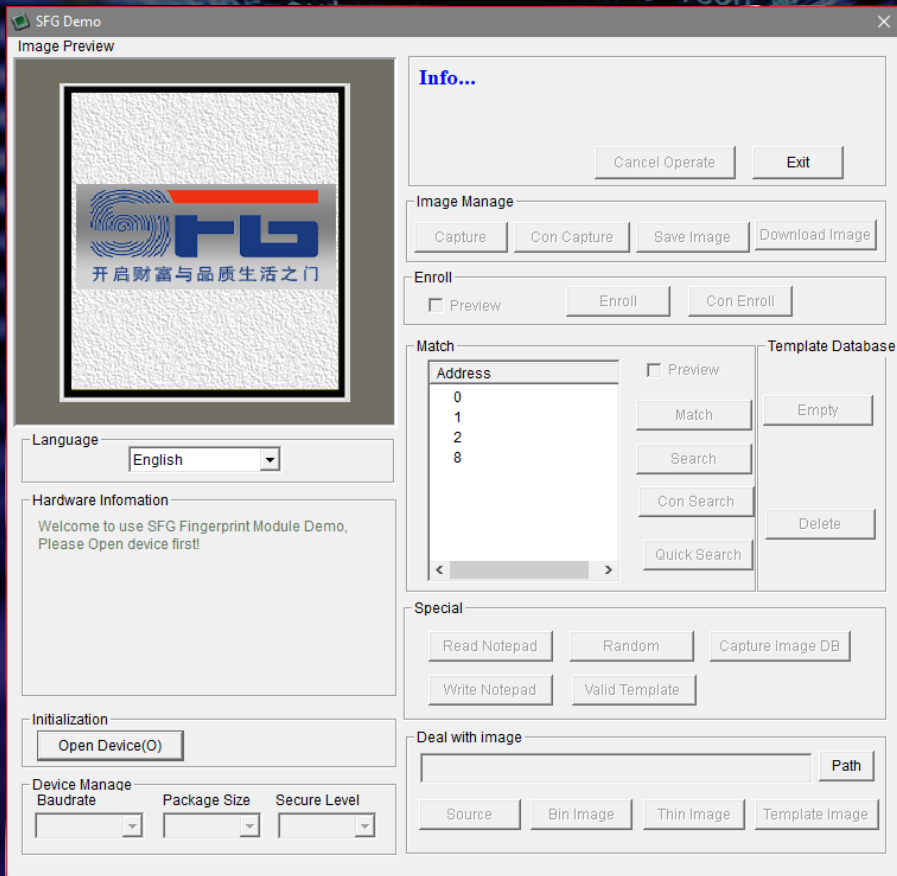
## PILE 9V



Permet d'alimenter la carte Arduino.

# PROGRAMME

## LECTEUR D'EMPREINTE DIGITAL



CONFIGURATION DU LOGICIEL SFG DEMO AVANT DE PROGRAMMER.



LOGICIEL PERMETTANT D'ENREGISTRER, SCANNER ET TESTER LES EMPREINTES DU CAPTEUR.



ENREGISTRER DES EMPREINTES AVANT DE LANCER LE PROGRAMME.



# PROGRAMME

## LECTEUR D'EMPREINTE DIGITAL

fingerprint §

```
#include <SoftwareSerial.h>

#include <Adafruit_Fingerprint.h>

#include <Servo.h>

#include <SoftwareSerial.h>
```

```
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    key = 1;
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;
}
```



INSÉRER LES LIBRAIRIES.



PROGRAMME PERMETTANT DE VERIFIER SI L'EMPREINTE EST CELLE DE L'UTILISATEUR.



INSÉRER LES LIBRAIRIES SI ELLE EST BONNE LE VERROU SE TOURNERA SI ELLE EST MAUVAISE UN MESSAGE D'ERREUR S'AFFICHERA.

# PROGRAMME

## SERVOMOTEUR

```
#include <Servo.h>

Servo serrure; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 180;    // variable to store the servo position
int key = 1;
void setup() {
  serrure.attach(5); // attaches the servo on pin 13 to the servo object
}

void loop() {
  if (key==1) {
    openserrure();
    key = 0;
  }
}

void openserrure(){

  serrure.write(pos);
  delay(2000);
  pos = 90;
  serrure.write(pos);
  delay(2000);
}
```



INSÉRER LES LIBRAIRIES.



PROGRAMME PERMETTANT DE VERIFIER SI L'EMPREINTE EST CELLE DE L'UTILISATEUR.



INSÉRER LES LIBRAIRIES: « SI ELLE EST BONNE LE VERROU SE TOURNERA SI ELLE EST MAUVAISE UN MESSAGE D'ERREUR S'AFFICHERA ».



# PROGRAMME

ESP32(WIFI)

```
discord_test_esp32  arduino_secrets.h  discord.h

#include "arduino_secrets.h"
/*
  Discord WebHook Example for ESP32
*/
#include "discord.h"

void setup() {
  Serial.begin(9600);
  connectWIFI();
  sendDiscord("Hello World!");
}

void loop() {
}
```

PARTIE SERVANT A ENVOYER DES MESSAGES.

```
discord_test_esp32  arduino_secrets.h  discord.h

#define SECRET_SSID "iPhone de Ayoub"
#define SECRET_PASS "ayoub063"
//Ccopy the webhook url here:
#define SECRET_WEBHOOK "/api/webhooks/569826518470098944/ArtR0Qg"
#define SECRET_TTS "true"
```

PARTIE PERMETTANT D'INSÉRER LES DONNÉES  
POUR SE CONNECTE AU WIFI ET A DISCORD.

```
discord_test_esp32  arduino_secrets.h  discord.h

void connectWIFI() {
  WiFiMulti.addAP(ssid, pass);
  WiFi.mode(WIFI_STA);
  Serial.print("[WiFi] Connecting to: ");
  Serial.println(ssid);
  // wait for WiFi connection
  while ((WiFiMulti.run() != WL_CONNECTED)) {
    Serial.print(".");
  }
  Serial.println("[WiFi] Connected");
}

void sendDiscord(String content) {
  WiFiClientSecure *client = new WiFiClientSecure;
  if (client) {
    client -> setCACert(discordappCertificate);
    {
      HTTPClient https;
      Serial.println("[HTTP] Connecting to Discord...");
      Serial.println("[HTTP] Message: " + content);
      Serial.println("[HTTP] TTS: " + discord_tts);
      if (https.begin(*client, discord_webhook)) { // HTTPS
        // start connection and send HTTP header
        https.addHeader("Content-Type", "application/json");
        int httpCode = https.POST("{\"content\":\"" + content + "\",\"tts\":\"" + discord_tts + "\"}");
      }
    }
  }
}
```

PARTIE PERMETTANT DE SE CONNECTER AU WIFI  
ET À DISCORD.

# PROGRAMME

```
class MyServerCallbacks: public BLEServerCallbacks {
    void onConnect(BLEServer* pServer) {
        deviceConnected = true;
    };

    void onDisconnect(BLEServer* pServer) {
        deviceConnected = false;
    }
};

class MyCallbacks: public BLECharacteristicCallbacks {
    void onWrite(BLECharacteristic *pCharacteristic) {
        std::string rxValue = pCharacteristic->getValue();

        if (rxValue.length() > 0) {
            Serial.println("*****");
            Serial.print("Received Value: ");

            for (int i = 0; i < rxValue.length(); i++) {
                Serial.print(rxValue[i]);
            }

            Serial.println();
        }
    }
};
```



INSÉRER LES LIBRAIRIES.



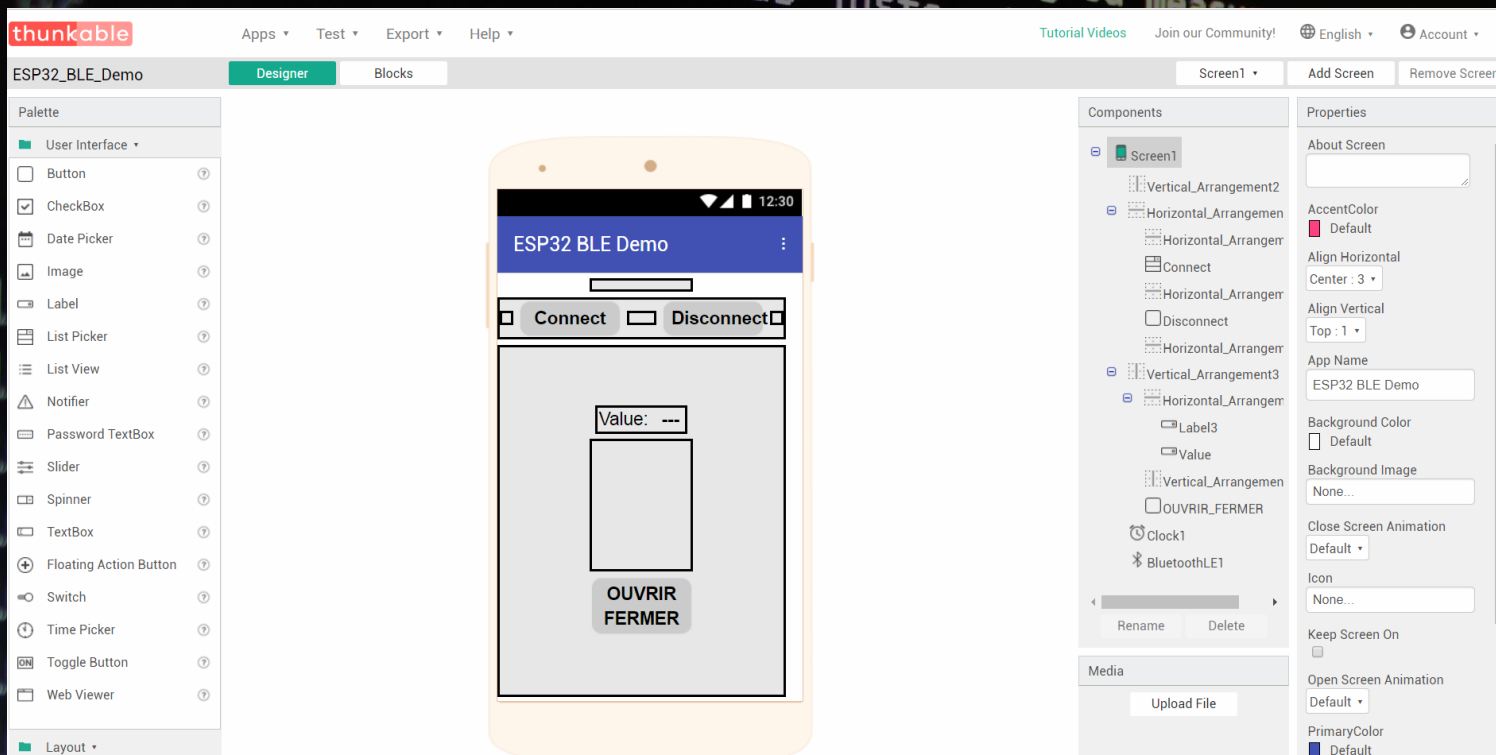
PROGRAMME DE SE CONNECTER EN BLUETOOTH VIA UNE APPLICATION



LE VERROU S'OUVRE ET SE FERME DIRECTEMENT SUR LE TELEPHONE



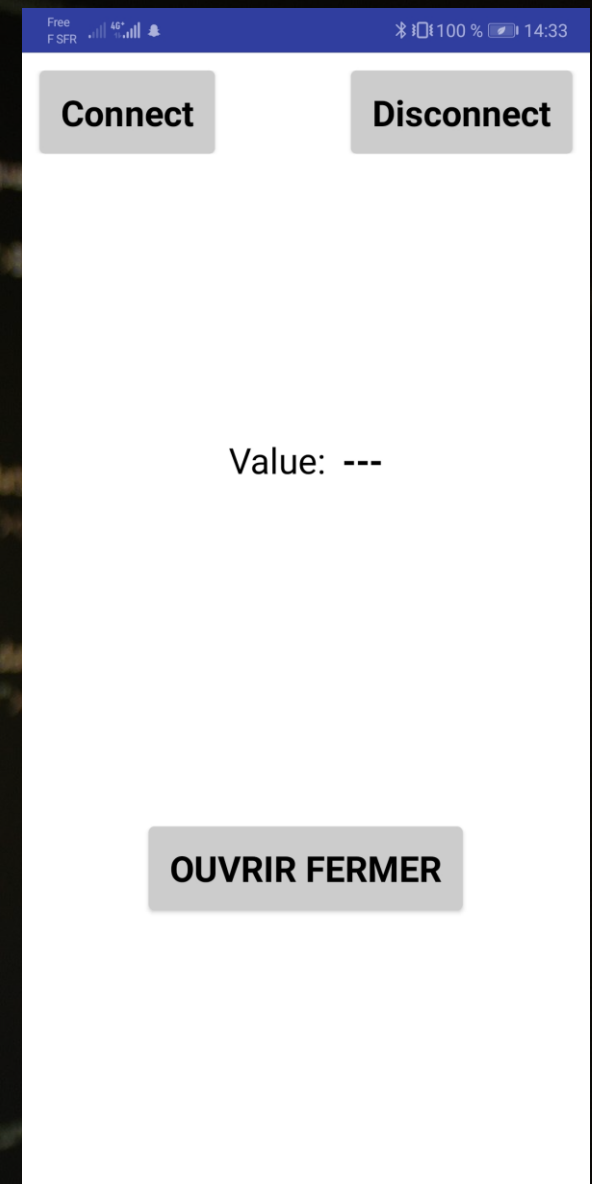
# PROGRAMME



INTERFACE DE CREATION



APPLICATION MOBILE.



SERRURE CONNECTÉE

BENALLAL - BENKHIRA



# PHASE DE TEST



# CONCLUSION



# AMÉLIORATIONS

- ☆ UTILISATION DU CAPTEUR DU TELEPHONE POUR DEVEROUILLER .
- ☆ PROGRAMMER TOUS LES CAPTEURS DANS LA CARTE ARDUINO UNO.
- ☆ CRÉATION D'UNE MAQUETTE (PORTE...).