

[CLIM HTTP] → FLUSSO LOGGO

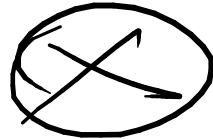
→ DOTTAGLIO

①

(WINDOWS)



PC

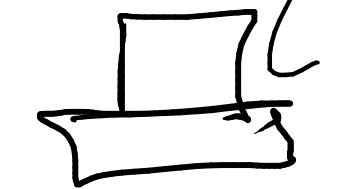


ROUTER

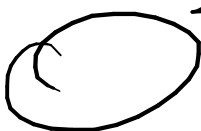
- SSID: "NORONEDS"
- PASSWORD: "PASSWORD"

{ CONNESSIONE - WIFI }

②

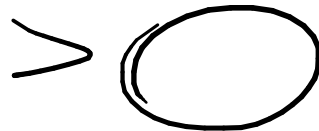


CLIM



WIFI

80 → GET / HTTP.1.1



server

(GOOG65.COM)

{ RICERCA STATALE
server }

***** Comandi fondamentali ClientHTTP *****

Dentro setup():

- WiFi.begin(ssid, password);
// Connessione dei client

- while (WiFi.status() \neq WL_CONNECTED)
// Verifico stato connessione

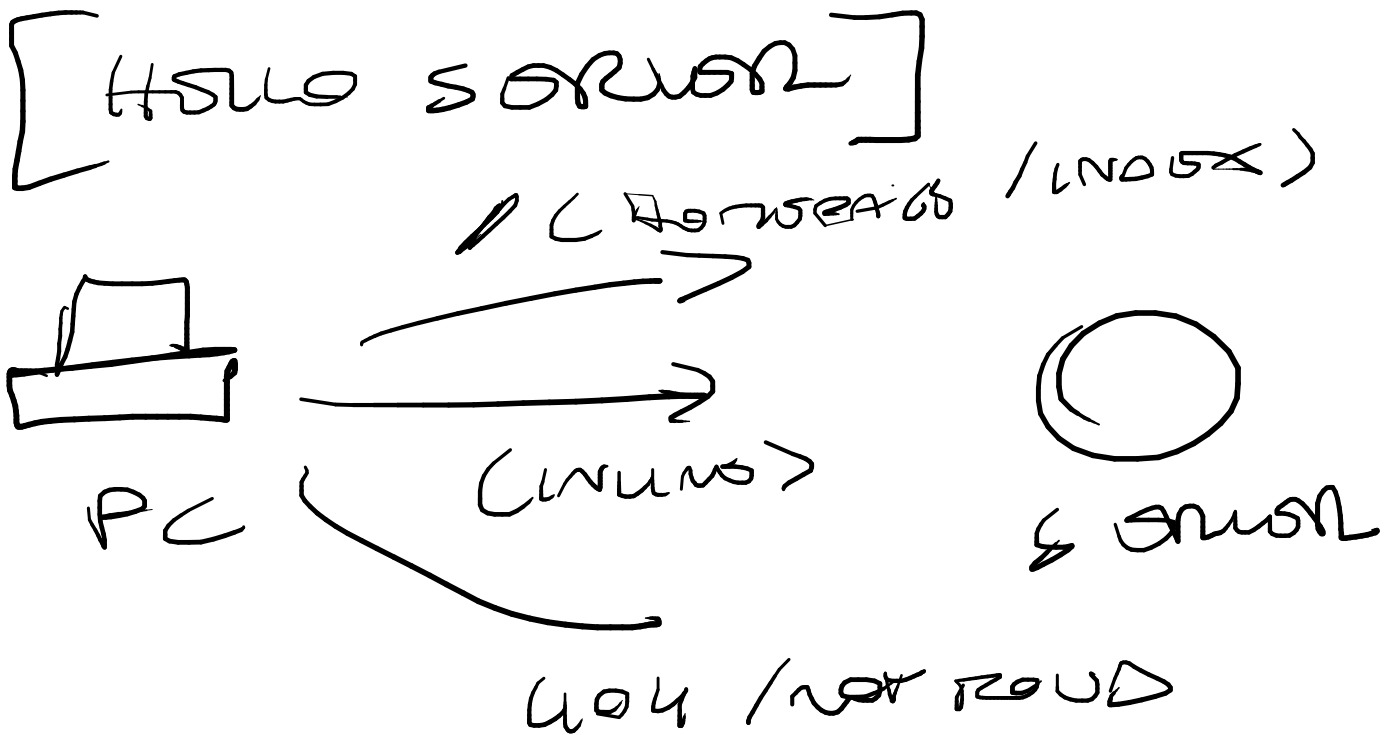
- Serial.println(WiFi.localIP());
// Visualizzazione IP

WiFiClient client; // Crei un oggetto di tipo
- client.connect(host, httpPort) → true/false
// Se connesso oppure no

- client.print("GET / HTTP/1.1\n");
// Prima parte della risposta
→ Connessione? ✓ | ✗

- client.print("Host: www.google.com\r\n
Connection: close\r\n\r\n");
// Seconda parte della risposta
→ Connessione? ✓ | ✗

#include <WiFi.h>



***** Comandi fondamentali HelloServer *****

Dentro setup():

- WiFi.begin(ssid, password);
// Connessione dei client
- while (WiFi.status() \neq WL_CONNECTED)
// Verifico stato connessione
- Serial.println(WiFi.localIP());
// Visualizzazione IP

Utilizzo di modulo DNS (MDNS = Module DNS)

→ ESPmDNS.h (Includiamo "MDNS")

- MDNS.begin("esp32")
// Se richiesta connessione arriva da scheda ESP32

Dentro setup():

```
#include <WebServer.h>
#include <ESPmDNS.h>
```

```
// Creazione server su porta 80
WebServer server(80);
```

Ci serve lanciare sempre il metodo "on".

```
// sito.com/
- server.on("/", handleRoot); // pagina principale - home page

// sito.com/inline
- server.on("/inline",handleInline); // esempio altra pagina

// sito.com/404
- server.onNotFound(handleNotFound); // pagina non trovata

// start del server
- server.begin();
```

Ogni pagina richiede una funzione handle() (che creuani

/ → Root → void handleRoot()

/inline → void handleInline()

/404 → void handleNotFound()

Richieste server:

```
- server.send(200, "text/html", "<a href='/inline'>spegni</a>")
```

Sintassi (comando):

```
send(messaggio HTTP, codifica, contenuto)
```

```
// Dentro loop
server.handleClient();
```

Cose importanti Login.ino:

```
void login(){
    bool usernameAdmin=false;
    bool passwordAdmin=false;
    bool username=false;
    bool password=false;
    for (uint8_t i = 0; i < server.args(); i++) {
        // Se lo username è admin = admin entrerà nelle sue pagine
        if(server.argName(i)=="username" && server.arg(i)=="admin")
            usernameAdmin=true;
        // Se l'admin ha messo la password corretta ("1234")
        if(server.argName(i)=="password" && server.arg(i)=="1234")
            passwordAdmin=true;
        // Stesso per il client → username == "client"
        if(server.argName(i)=="username" && server.arg(i)=="client")
            username=true;
        // Se username ha messo la password corretta ("pass")
        if(server.argName(i)=="password" && server.arg(i)=="pass")
            password=true;
    }
    // Se l'admin ha messo i dati giusti, lo mandiamo alla sua pagina
    if(usernameAdmin==true && passwordAdmin==true){
        server.send(200, "text/html", adminPage);
    }
    else{
        // Se il client ha messo i dati giusti, lo mandiamo alla sua pagina
        if(username==true && password==true){
            server.send(200, "text/html", clientPage);
        }
        else{
            // Errore altrimenti
            server.send(200, "text/html", loginPageError);
        }
    }
}
}
```

