

Università degli Studi della Campania Luigi Vanvitelli - Dipartimento di Ingegneria

Laurea Triennale in Ingegneria Elettronica e Informatica

Disegno by Luca

Laboratorio di Sviluppo di Applicazioni per IoT
a.a. 2023-2024

PHP

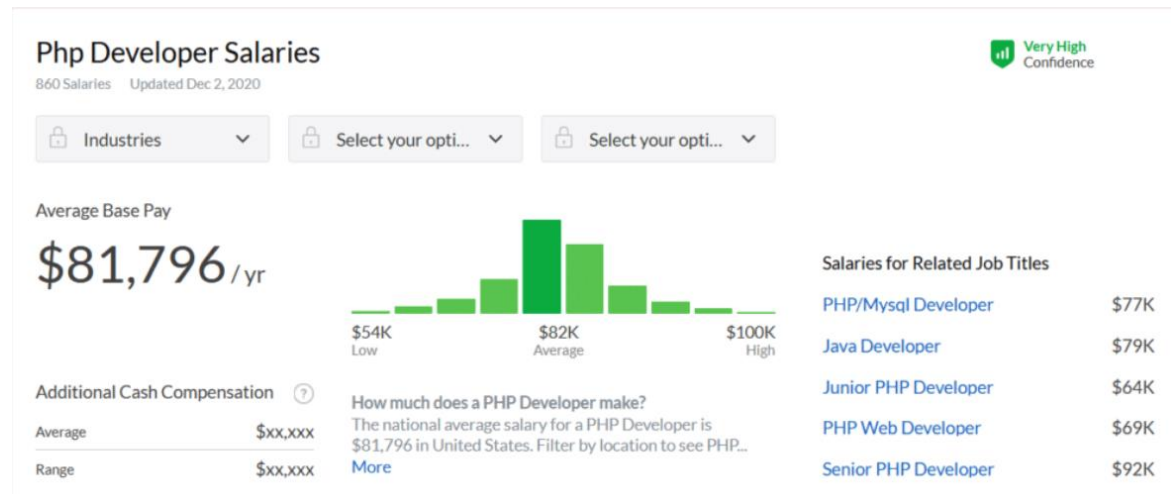
Docente: Carlo Mazzocca
e-mail: carlo.mazzocca@unibo.it

PHP

Processing Hyper Texting

PHP è un linguaggio di scripting open source molto utilizzato per lo sviluppo di applicazioni web

Gli script PHP sono eseguiti sul server che restituisce una risposta al browser sotto forma di pagina HTML

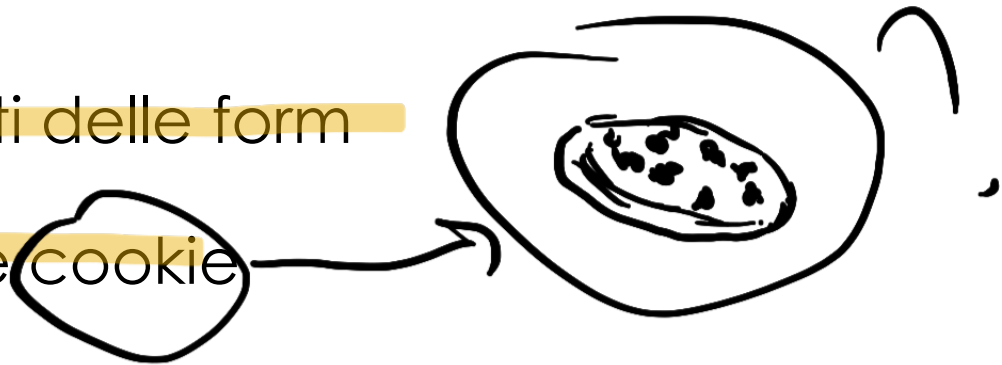


Caratteristiche

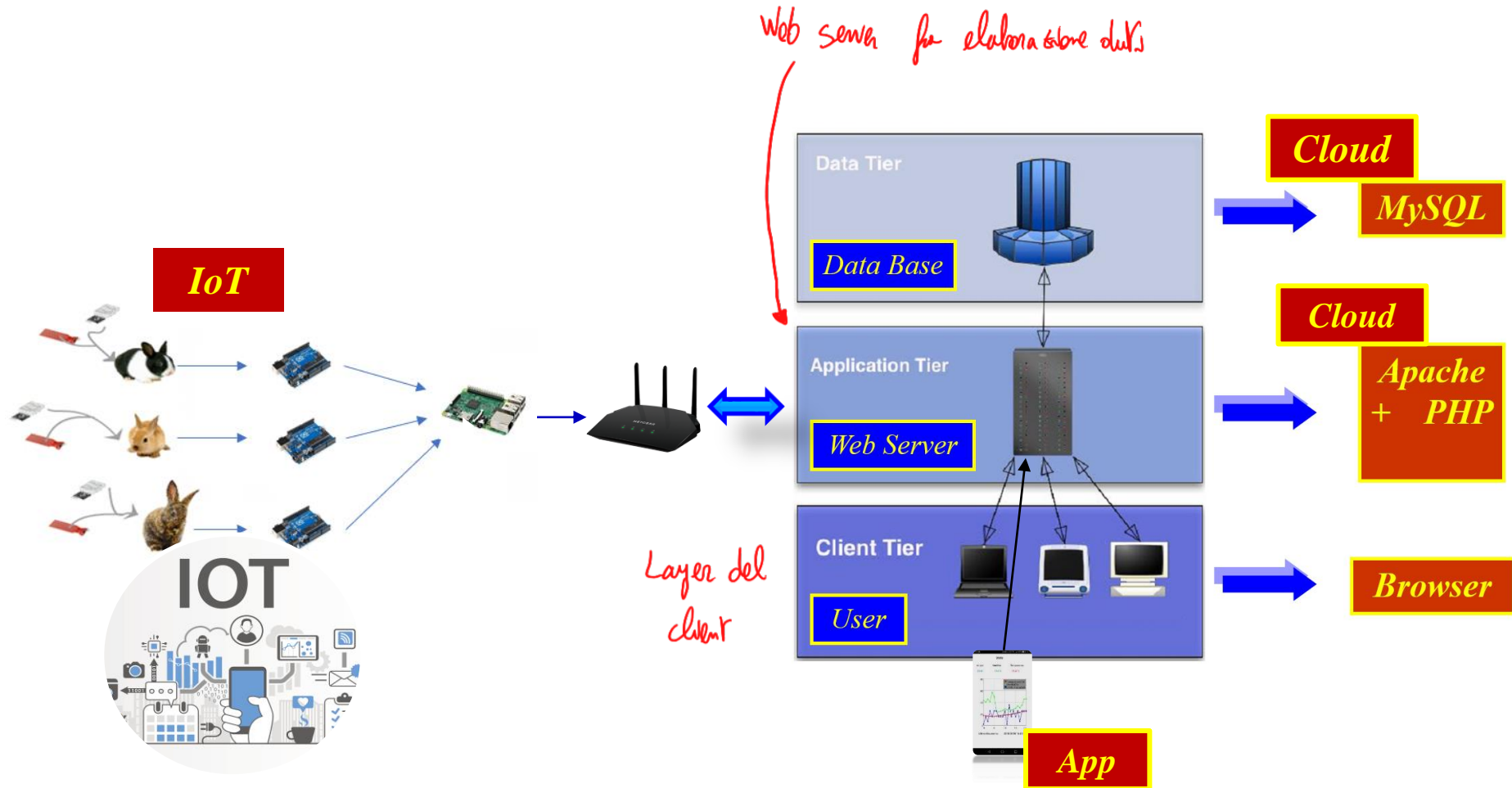
- Linguaggio di scripting interpretato lato server
- Non tipizzato
- PHP viene utilizzato per il back end
- Si integra con altri linguaggi come HTML
- Le variabili sono case sensitive, le funzioni no

Utilizzo

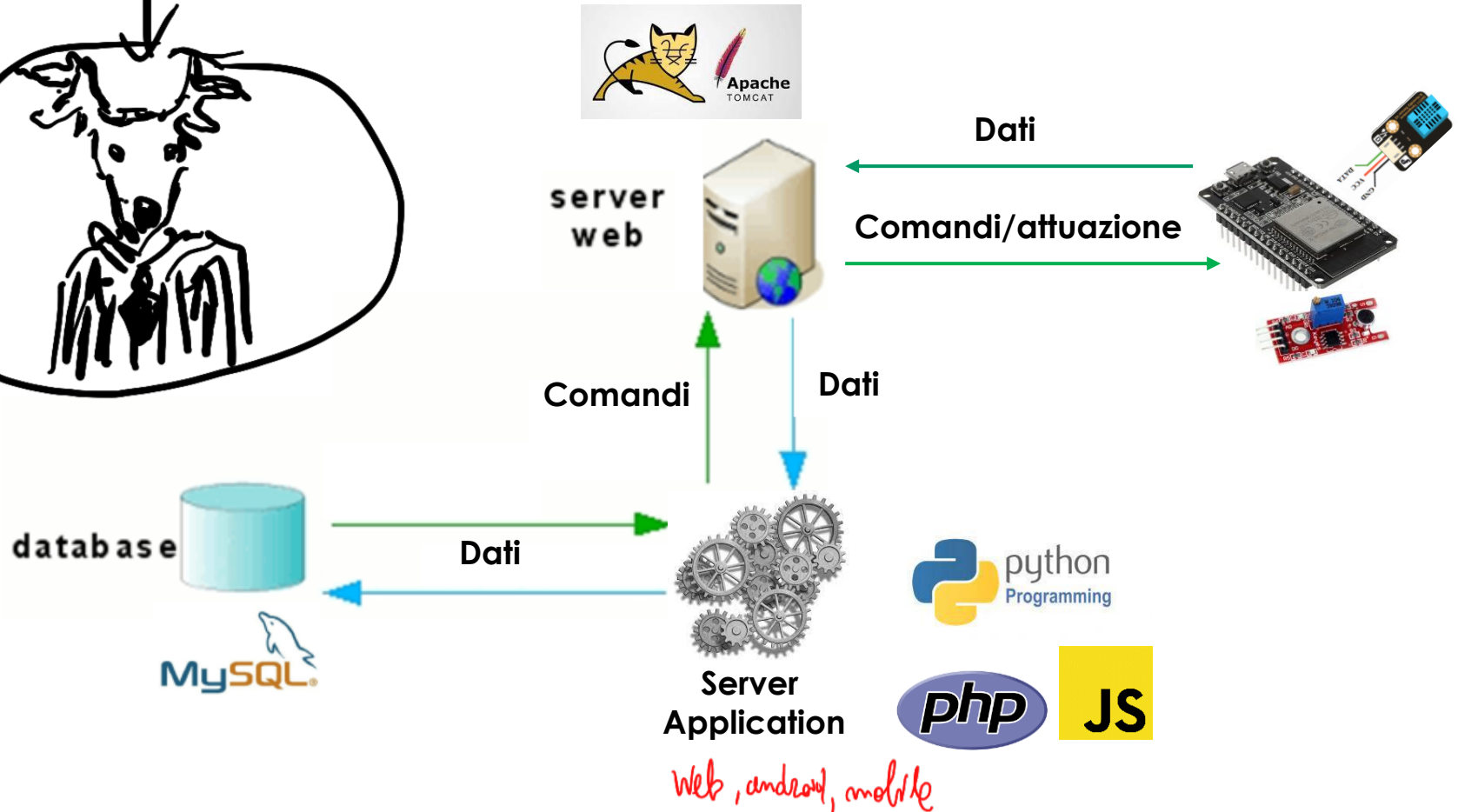
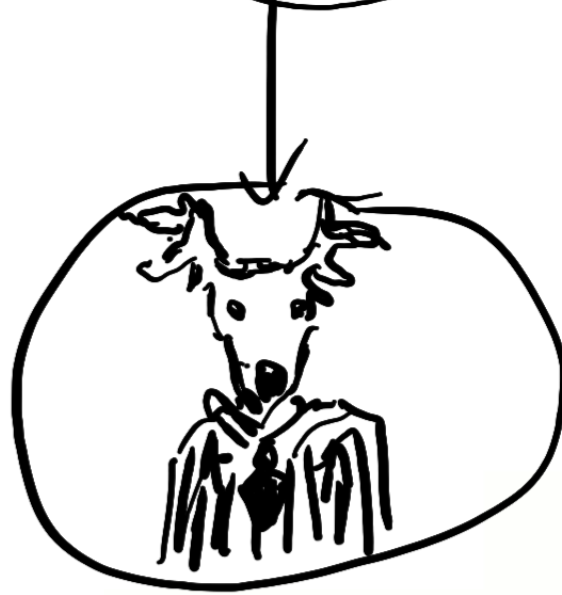
- Generare contenuti dinamici di una pagina web
- Creare, aprire, leggere, scrivere, eliminare e chiudere file
- Raccogliere i dati delle form
- Inviare e ricevere cookie
- Aggiungere, eliminare, modificare i dati nel database
- Controllare l'accesso degli utenti
- Crittografare i dati



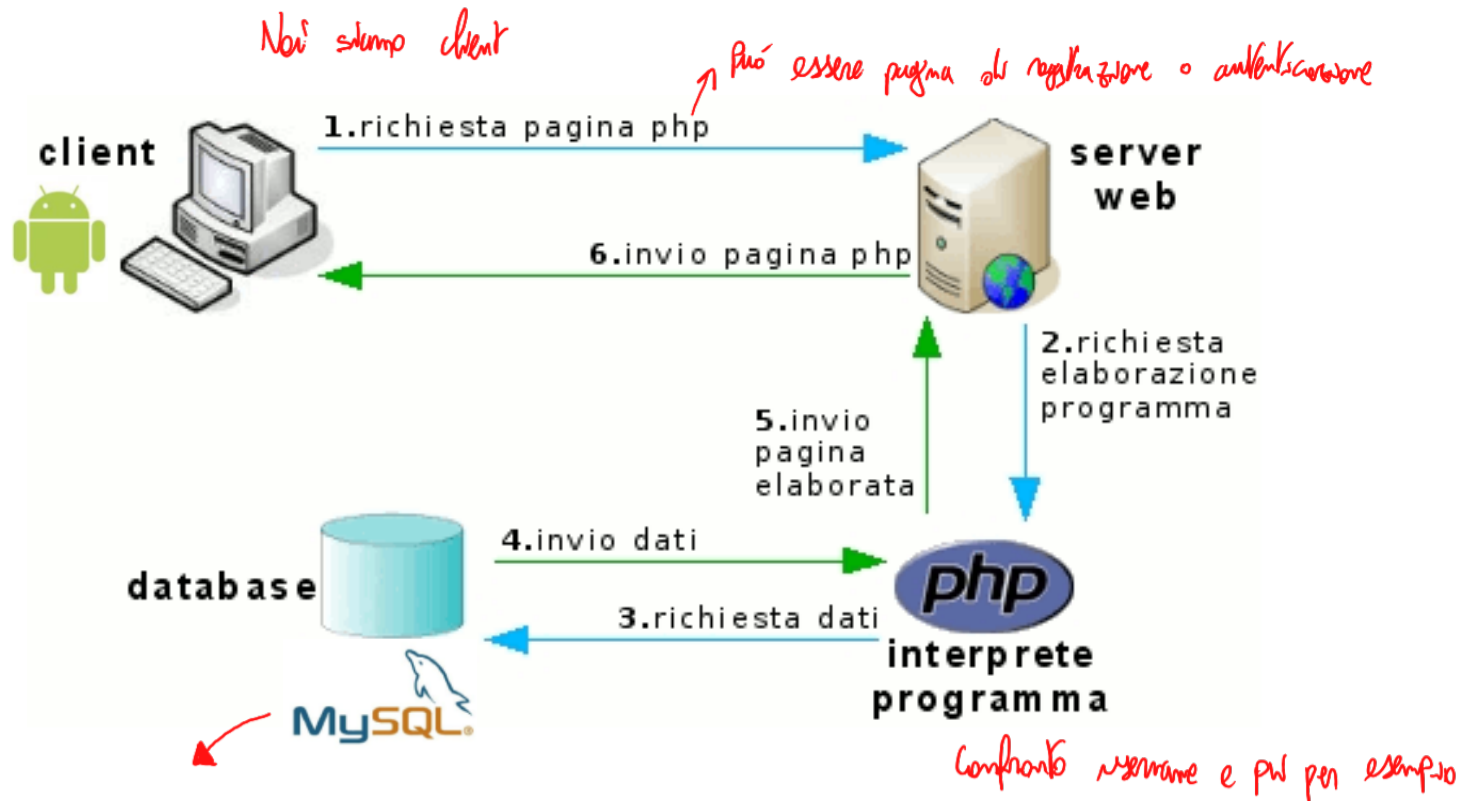
Architettura a Tre Livelli



Client/Server (Smart Sensor)

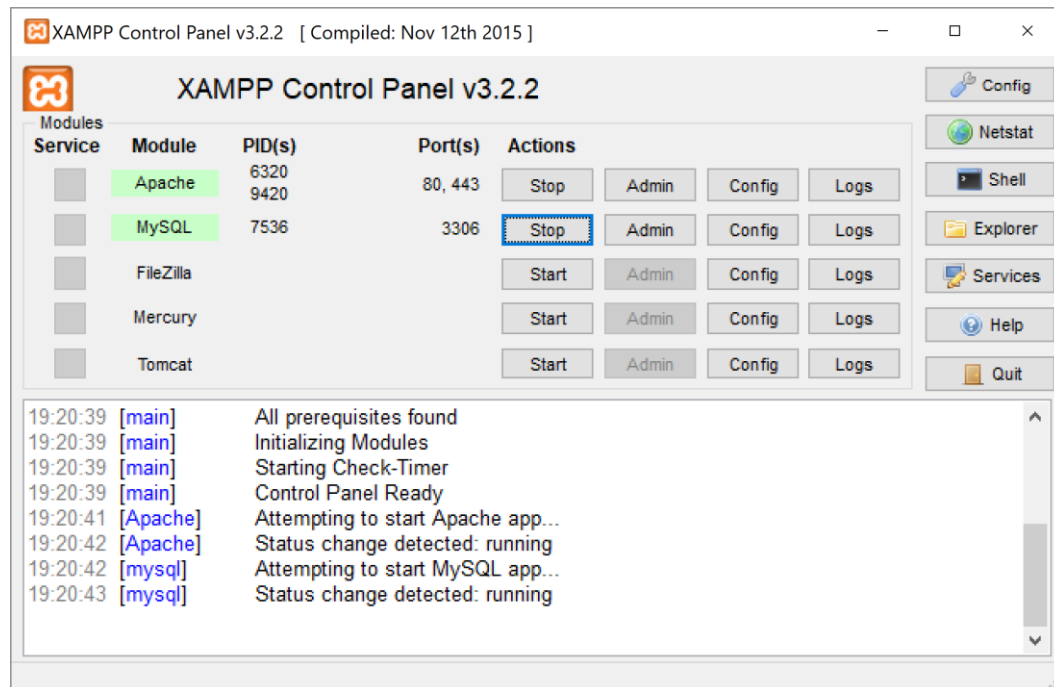


Client/Server (Web Client)

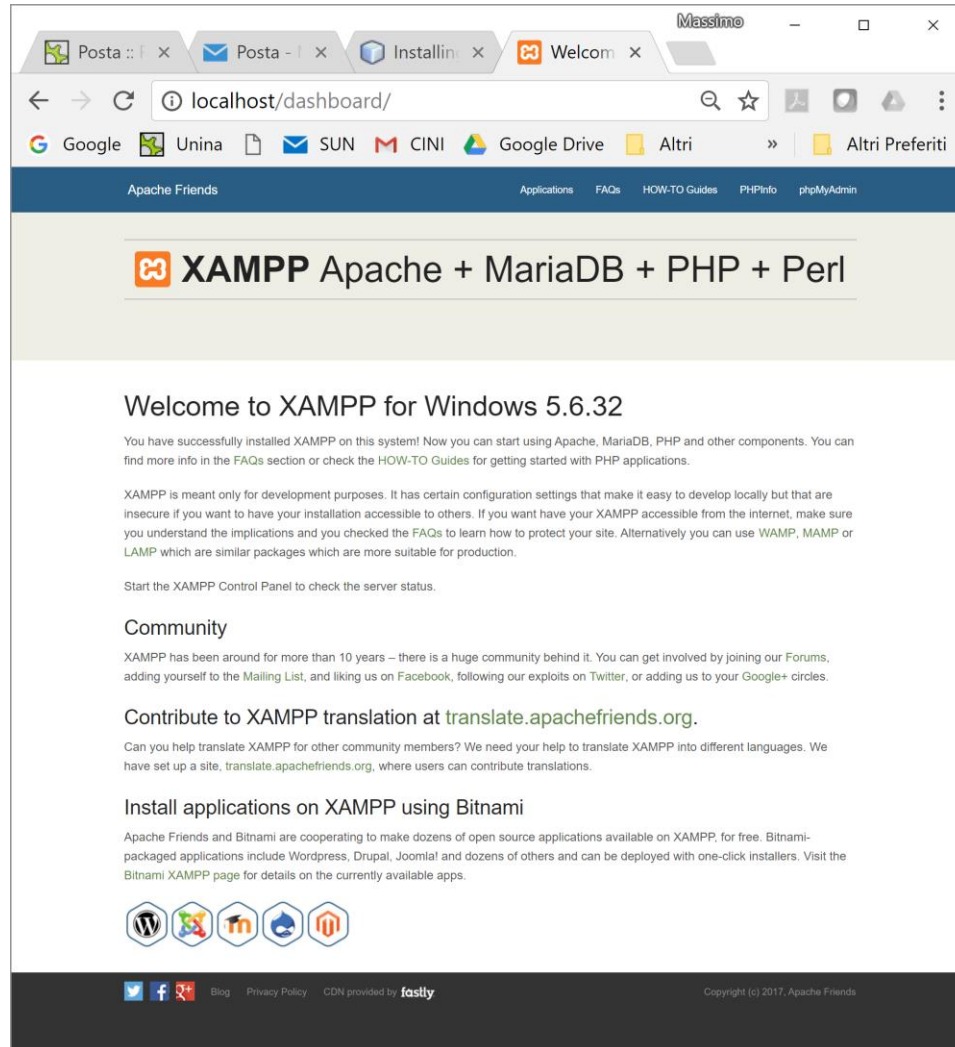


XAMPP

XAMPP è una soluzione web server cross-platform open-source progettata per realizzare applicazioni web in locale



http://localhost



Sintassi PHP

- Uno script PHP inizia con `<?php` e termina con `?>`

```
<?php  
// PHP code goes here  
?>
```

- L'estensione di default dei file PHP è `“.php”`

Esempio di Pagina HTML

Aggiungere file test.php in C:\xampp\htdocs

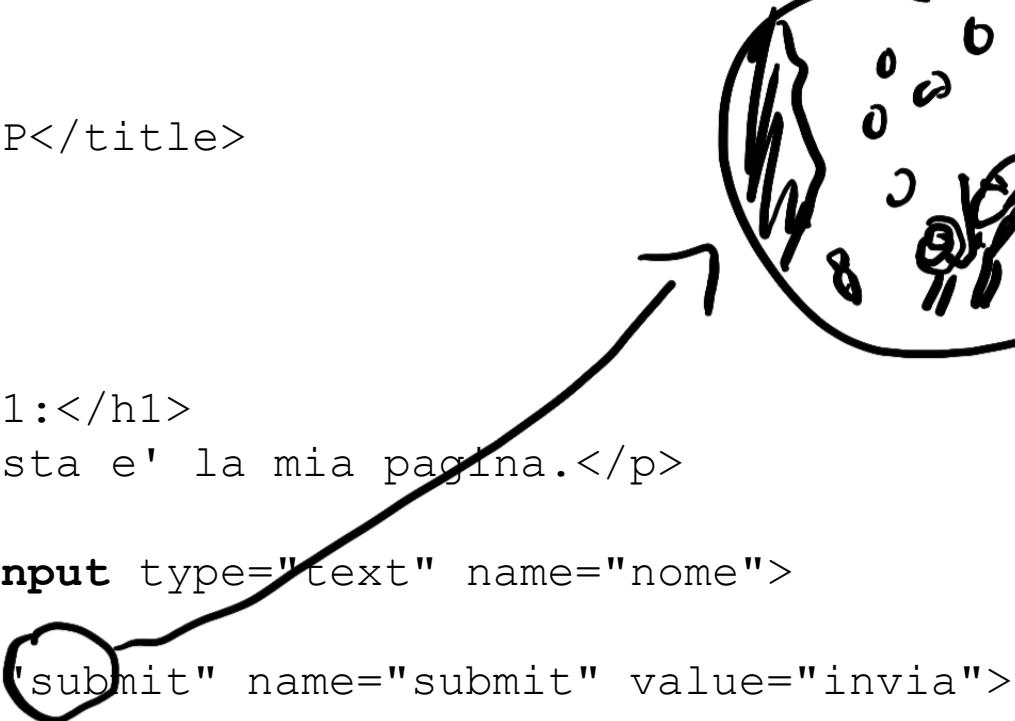
```
<html>
  <head>
    <title>Test PHP</title>
  </head>

  <body>
    <form>
      <h1>Esempio 1:</h1>
      <p>Ciao! Questa e' la mia pagina.</p>

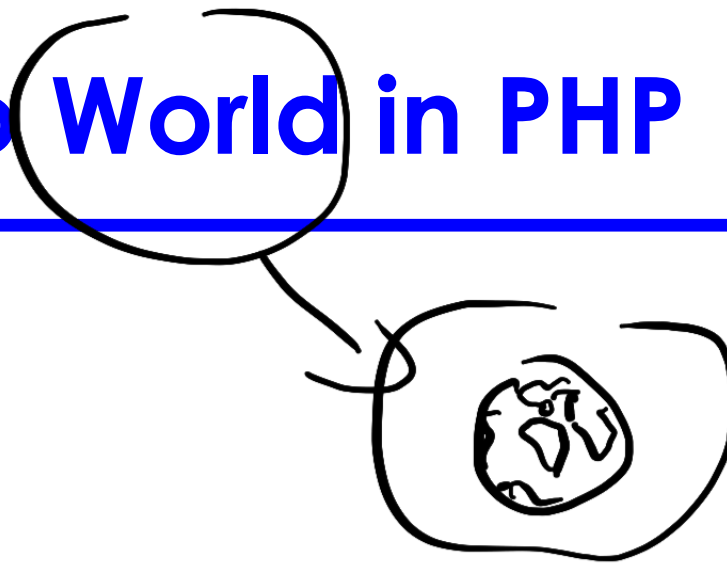
      <br>Nome: <input type="text" name="nome">

      <input type="submit" name="submit" value="invia">

    </form>
  </body>
</html>
```

A hand-drawn sketch of a globe is located in the upper right quadrant of the slide. An arrow originates from the 'submit' attribute of the last input field in the code, circles around it, and points towards the globe. The globe is depicted with simple lines for continents and latitude/longitude lines.

Hello World in PHP



Codice:

```
<?php  
echo "Ciao Mondo!";  
?>
```

Costanti e Variabili

Codice:

```
<?php define("CHEDIRE", "Viva la vita");  
echo "".CHEDIRE.""; //Apparira' la scritta "Viva la vita"  
?>
```

Codice:

```
<?php  
$variabile="valore";  
?>
```

Codice:

```
$variabile=true; // variabile booleana
```

Esempio Codice PHP

Codice:

```
<?php

$eta= 14; //QUI DEFINIAMO BANALMENTE UNA VARIABILE NUMERICA

if ($eta < 18){ //SE IL VALORE DI ETà E' MAGGIORE DI 18

echo "Sei maggiorenne!"; //MOSTRIAMO UN BANALISSIMO TESTO

} else { //altrimenti

echo "Sei minorenne";

}

?>
```

Esempio Programma PHP in XAMPP

Assumere che XAMPP è installato in:

C:\xampp\htdocs

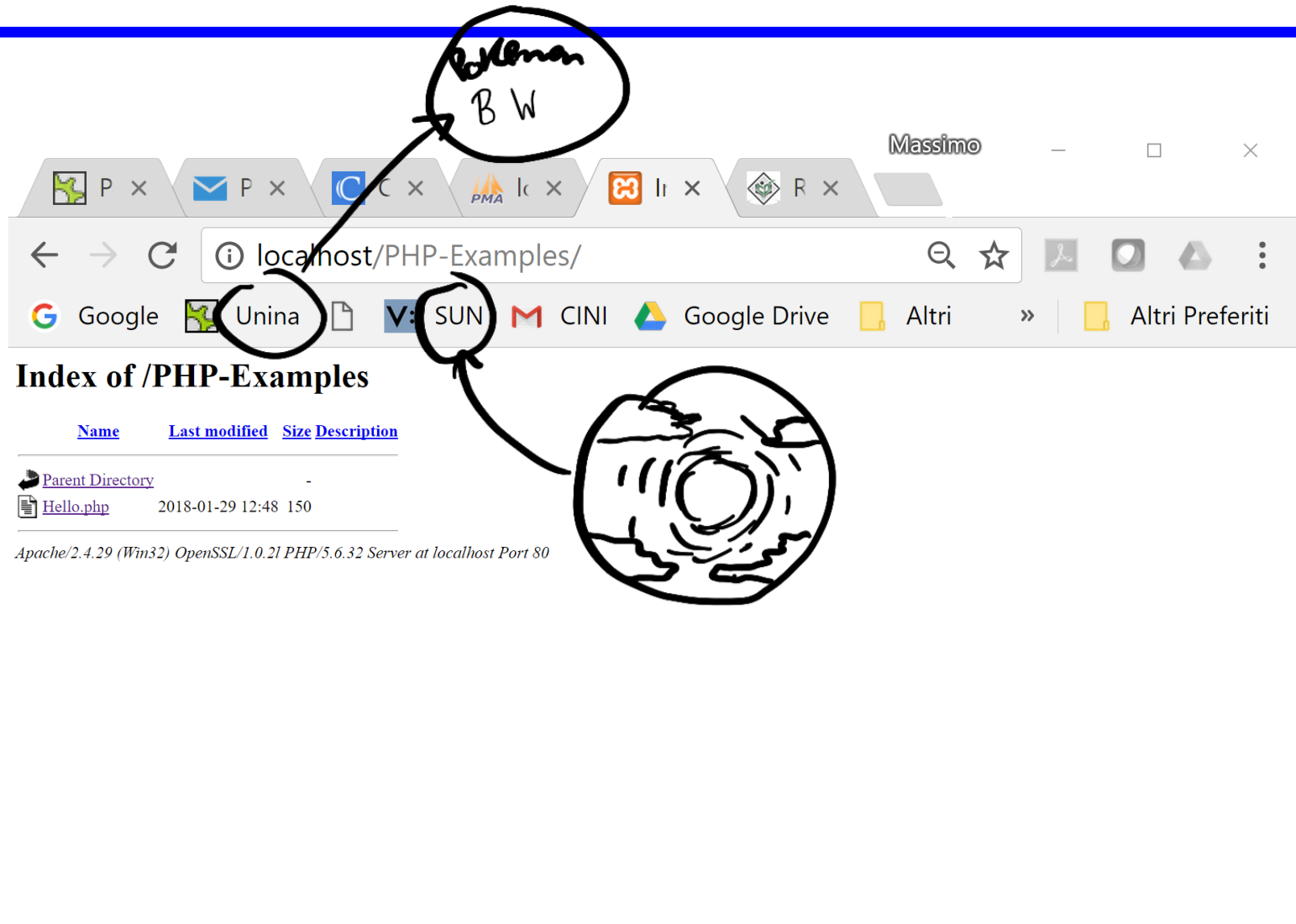
Creare una cartella:

PHP-Examples

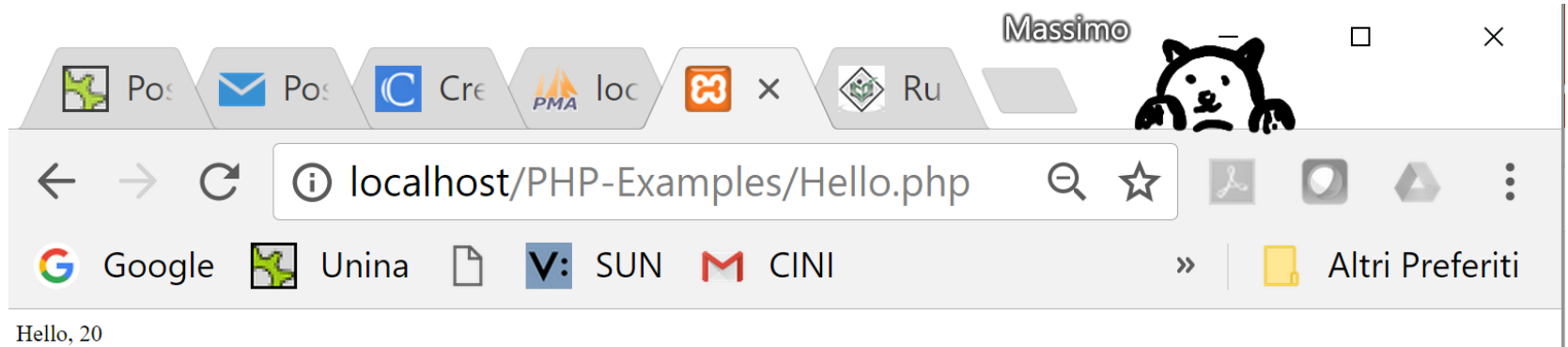
Creare un file php, es. hello.php:

```
<html>
<head><title>PHP Example</title></head>
<body>
  <?php
    $val = 20;
    echo "Hello, $val";
  ?>
</body>
</html>
```

Esempio Programma PHP in XAMPP



Eseguire Programma PHP in XAMPP



Metodo POST

La variabile **\$_POST** serve per ottenere un dato presente in un campo di un form html che abbia come metodo "post". In **2-Account.html** sarà contenuto il seguente codice:

Codice:

```
<form action="pagina.php" method="post">
Eta' : <input type="text" name="eta"/>
<br>
Nome: <input type="text" name="nome"/>
<br>
<input type="submit" name="send" value="Invia Form"/>
</form>
```



Metodo POST

In pagina.php sarà contenuto il seguente codice:

Codice:

↑ identifica contenuti nella post

```
echo $_POST['eta'];

echo "<p></p>"; //Mettiamo questo codice per inserire uno "spazio"
//Qui verrà mostrato ciò che ha inserito l'utente nel campo di nome eta.

echo $_POST['nome'];

//Qui verrà mostrato ciò che ha inserito l'utente nel campo nome
```

Metodo POST

Per verificare se la variabile vuota

Codice:

```
if (empty($nomevariabile)) {
```

Per verificare se la variabile esiste

Codice:

```
if (isset($nomevariabile)) {
```

Metodo POST

Codice:

```
if ($_POST['send']){

if (!empty($_POST['eta'])) {
echo $_POST['eta'];
} else {
echo "Hai lasciato vuoto il campo età";
}

echo "<p></p>"; //Mettiamo questo codice per inserire uno "spazio"
//Qui verrà mostrato ciò che ha inserito l'utente nel campo di nome eta.

if (empty($_POST['nome'])) {
echo "Il campo nome è vuoto!";
} else {
echo $_POST['nome'];
}
//Qui verrà mostrato ciò che ha inserito l'utente nel campo nome
}
```

Metodo GET

\$_GET viene utilizzato per gestire dei dati tramite "url"

Per passare in paramentro: *pagina.php?nome=Max*

Codice:

```
<?php  
echo "Ciao";  
  
echo $_GET['nome'];  
?>
```

Variabile Predefinite

Codice:

```
echo $_SERVER['HTTP_COOKIE']; //Mostra il valore di tutti i cookie
echo $_SERVER['HTTP_HOST']; // Mostra il nome dell'host su quale risiede
il server web
echo $_SERVER['REMOTE_ADDR']; // Mostra l'indirizzo ip dell'utente
echo $_SERVER['PHP_SELF']; // Mostra il nome file della pagina corrente
echo $_SERVER['SCRIPT_FILENAME']; //Mostra il nome e il percorso completo
dello script corrente
echo $_SERVER['SERVER_NAME']; //Mostrail nome del server web
echo $_SERVER['HTTP_USER_AGENT']; //Mostra il nome univoco del browser
utilizzato dall'utente
```

Array

Codice:

```
<?php
$nomearray = array ("giorgio","nicola","filippo"); //definiamo questa
variabile/array
?>
```

Codice:

```
<?php
echo $nomearray[0] ; //Verrà mostrata la scritta giorgio
echo $nomearray[1]; //Verrà mostrata la scritta nicola
echo $nomearray[2]; //Verrà mostrata la scritta filippo e cosi' via..
?>
```


Array

Codice:

```
<?php
$estensione="gif";
$estensioni_accettate = array('bmp', 'jpg', 'gif');
if( in_array($estensione, $estensioni_accettate) )
{
echo "$estensione";
}
else
{
echo "no";
}
?>
```

Creare una Funzione in PHP

Codice:

```
<?php
$testo2 = "Alessandro";

function stampa_testo($testo1){
global $testo2; //LA VARIABILE $TESTO2 LA DEFINISCO FUORI DALLA FUNZIONE,
QUINDI PER POTERLA USARE NELLA FUNZIONE DEVO SCRIVERE GLOBAL $TESTO2;

echo $testo1;
echo ", ";
echo $testo2;
}

echo stampa_testo("Matteo"); //Verrà mostra la scritta Matteo, Alessandro
?>
```

File

- **fopen**(\$filedaaprire, \$mode); // Aprire
- **fread**(\$filedaleggere); //Leggere
- **fwrite** (\$testodascrivere, \$fileincuiscrivere); // Scrivere
- **unlink**(\$daeliminare); // Eliminare
- **file_exists**(\$dacontrollare); // Controllare esistenza
- **is_writable**(\$dacontrollare) // Controllare se riscrivibile
- **is_readable**(\$dacontrollare) // Controlla se leggibile
- **file_get_contents** // Leggere un file, si puo' usare al posto di fread, ma non ha bisogno di usare fopen prima(a differenza di fread)
- **fclose**(\$filedachiusure) // Chiudere

File

Codice:

```
<?php

$fp = fopen("nomefile.txt", "w"); //Per prima cosa apriamo il file,
scrivendo come modalità "w", in modo da poterlo sia leggerlo che
scriverlo

fwrite($fp, "Ciao");

fclose($fp); //Chiudiamo il file, serve più che altro per motivi di
sicurezza.

?>
```

Cartelle

- **mkdir();** // Creare
- **rmdir();** // Eliminare una cartella
- **is_dir();** // Controllare esistenza
- **opendir();** // Aprire
- **readdir ();** // Leggere una cartella, per usare questa funzione bisogna mettere opendir all'inizio e closedir alla fine
- **closedir();** //Chiudere

Sessioni e Cookie

Le sessioni e i cookie sono entrambe variabili speciali in cui è possibile memorizzare dati temporaneamente

- **setcookie**("nomecookie", "valorecookie", "tempocookie"); // Creare un cookie

Per le sessioni si usano alcune funzioni:

- **session_start()**; // Bisogna usarla prima di settare una funzione
- **session_destroy()**; // Elimina tutte le sessioni
- **\$_SESSION['nomesessione']** = "valore"; // Creare sessione

Sessioni e Cookie

Codice:

```
<?php

session_start(); //Funzione che "crea un terreno" adatto alla definizione
delle sessioni

$_SESSION['username'] = "Matteo"; //Stabiliamo la sessione di nome
username

echo $_SESSION['username']; //Mostriamo il valore della sessione username

?>
```

Sessioni e Cookie

```
<form action="pagina.php" method="post">
<input type="text" name="username"/>
<input type="password" name="pass"/>
<input type="submit" name="login" value="Login"/>
</form>

<?php

if ($_POST['login']){

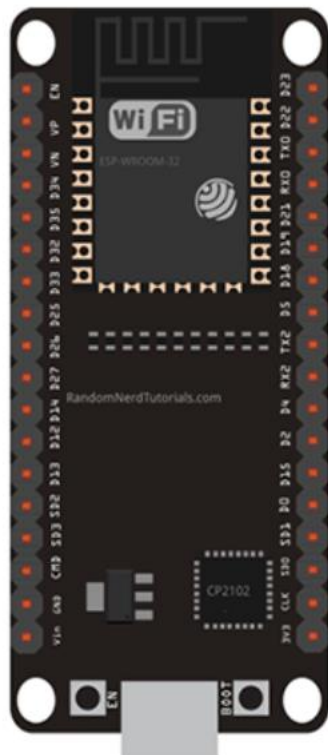
$password="123"; //Password per il login

if ($_POST['pass'] == $password){
//Se il dato proveniente dal campo username del form e il dato è uguale
alla variabile username e il dato //proveniente dal campo pass del form è
uguale alla variabile password
```


Sessioni e Cookie

```
setcookie("username", $_POST['username'], time() + 3600);  
//Crea un cookie di nome username, che contenga l'username inserito nel  
campo username del form  
  
setcookie("login", "ok", time() + 3600);  
//Crea un cookie di nome login, di valore "ok" e che duri 3600 secondi  
  
} else { //Altrimenti  
echo "Username e/o Password errati. Riprova"; //Fai apparire questo  
messaggio  
}  
  
}
```

POST da ESP32



HTTP POST

/update-sensor

- 1 URL encoded
- 2 JSON object
- 3 Plain text

HTTP Response

Status 200 (OK)

**Web Service
or API**

POST da ESP32 - Sketch

```
1  #include <WiFi.h>
2  #include <HTTPClient.h>
3
4  const char* ssid = "router name";
5  const char* password = "password";
6
7  //Your Domain name with URL path or IP address with path
8  const char* serverName = "http://192.168.1.84/myapp/test_sensor.php/";
9
10 // the following variables are unsigned longs because the time, measured in
11 // milliseconds, will quickly become a bigger number than can be stored in an int.
12 unsigned long lastTime = 0;
13 // Timer set to 10 minutes (600000)
14 //unsigned long timerDelay = 600000;
15 // Set timer to 5 seconds (5000)
16 unsigned long timerDelay = 5000;
17
18 void setup() {
19     Serial.begin(115200);
20
21     WiFi.begin(ssid, password);
22     Serial.println("Connecting");
23     while(WiFi.status() != WL_CONNECTED) {
24         delay(500);
25         Serial.print(".");
26     }
27     Serial.println("");
28     Serial.print("Connected to WiFi network with IP Address: ");
29     Serial.println(WiFi.localIP());
30
31     Serial.println("Timer set to 5 seconds (timerDelay variable), it will take 5 seconds before publishing the first reading.");
32 }
```

POST da ESP32 - Sketch

```
34 void loop() {
35     //Send an HTTP POST request every 10 minutes
36     if ((millis() - lastTime) > timerDelay) {
37         //Check WiFi connection status
38         if(WiFi.status()== WL_CONNECTED){
39             WiFiClient client;
40             HTTPClient http;
41
42             // Your Domain name with URL path or IP address with path
43             http.begin(client, serverName);
44
45             // If you need Node-RED/server authentication, insert user and password below
46             //http.setAuthorization("REPLACE_WITH_SERVER_USERNAME", "REPLACE_WITH_SERVER_PASSWORD");
47
48             // Specify content-type header
49             http.addHeader("Content-Type", "application/x-www-form-urlencoded");
50             // Data to send with HTTP POST
51             String httpRequestData = "api_key=tPmAT5Ab3j7F9&sensor=BME280&value1=24.25&value2=49.54&value3=1005.14";
52             // Send HTTP POST request
53             int httpResponseCode = http.POST(httpRequestData);
54
55             // If you need an HTTP request with a content type: application/json, use the following:
56             //http.addHeader("Content-Type", "application/json");
57             //int httpResponseCode = http.POST("{\"api_key\":\"tPmAT5Ab3j7F9\",\"sensor\":\"BME280\",\"value1\":\"24.25\",\"value2\":\"49.54\",\"value3\":\"1005.14\"}");
58
59             // If you need an HTTP request with a content type: text/plain
60             //http.addHeader("Content-Type", "text/plain");
61             //int httpResponseCode = http.POST("Hello, World!");
62
63             Serial.print("HTTP Response code: ");
64             Serial.println(httpResponseCode);
65
66             // Free resources
67             http.end();
68         }
69         else {
70             Serial.println("WiFi Disconnected");
71         }
72         lastTime = millis();
73     }
74 }
```

POST da ESP32 - PHP

```
1  <?php
2  // Check if API key is provided
3  if(isset($_POST['api_key'])) {
4      // Check if sensor data is provided
5      if(isset($_POST['sensor']) && isset($_POST['value1']) && isset($_POST['value2']) && isset($_POST['value3'])) {
6          // Get the sensor data
7          $sensor = $_POST['sensor'];
8          $value1 = $_POST['value1'];
9          $value2 = $_POST['value2'];
10         $value3 = $_POST['value3'];
11
12         // Open or create the file for appending
13         $file = 'sensor_data.txt';
14         $handle = fopen($file, 'a');
15
16         // Write the sensor data to the file
17         fwrite($handle, "Sensor: $sensor, Value1: $value1, Value2: $value2, Value3: $value3\n");
18
19         // Close the file
20         fclose($handle);
21
22         // Echo a success message
23         echo "Sensor data saved successfully.";
24     } else {
25         // If sensor data is not provided, return an error message
26         echo "Error: Incomplete sensor data";
27     }
28 } else {
29     // If API key is not provided, return an error message
30     echo "Error: No API key provided";
31 }
32 ?>
```