

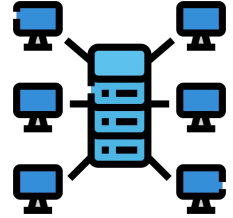
---

# Segunda Etapa P.I

— Daniel Van der Laat, Lisandro Sandi,  
Sebastian Venegas, Marlon Murillo —

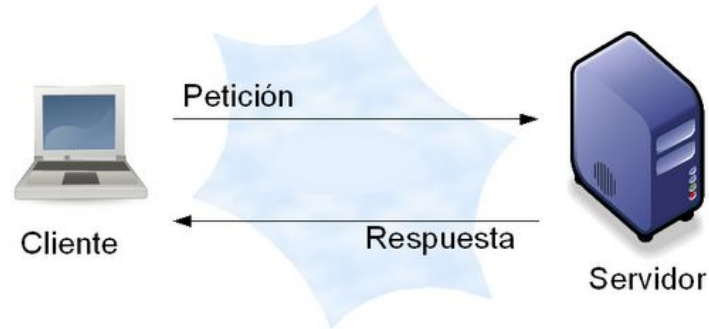
---

# Protocolo



- Sockets
- Servidor

- Cliente



# Código

```
nlohmann::json json_data;
input_file >> json_data;
input_file.close();
std::string json_string = json_data.dump();
nlohmann::json jfigure = json_data[buf];
std::string hmtl = "<!DOCTYPE html> \
<html> \
  <head> \
    <title>Lista de Palabras</title>\
  </head>\
  <body>\|
";
std::string nameFigure = buf;
hmtl+= "<h1>" + nameFigure + "</h1>\
<ul> ";
for (auto i = jfigure.begin(); i != jfigure.end(); ++i) {
  hmtl += "<li>" + i.key() + "</li>";
}
hmtl += "\
</html>";

sendResponse(client, hmtl.c_str());
```

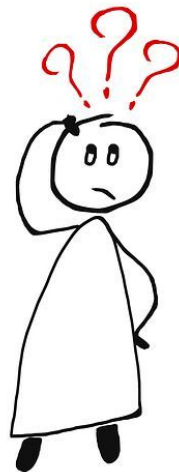
# Código

```
void browser(char* buf, std::vector<std::string>& html_vector) {  
    get_Regex(buf, "GET /([A-Za-z]+) HTTP/1.1", std::ref(html_vector));  
}
```

```
void Service( Socket * client ) {  
    char buf[ 1024 ] = { 0 };  
    int sd, bytes;  
  
    std::string directorio = "./jsons/";  
    std::vector<std::string> archivos;
```

# Dificultades encontradas

- Aprender a utilizar archivos .json y su librería
- Inyectar la información del .json en HTML para el navegador
- El servidor “se cae” cuando el navegador es el cliente



# Gracias

