Breve explicação de como criar um servidor e um cliente em Prolog para processamento de JSON

Servidor:
========

1. Declaração de um tipo JSON (apenas necessário se usar predicados de conversão de JSON para Prolog como o prolog_to_json/2 e json_to_prolog/2. No exemplo anexo optou-se por usar dicionários não sendo obrigatória esta declaração.

```
:- json_object student(name:string, number:integer).
```

2. Associação de um endereço à execução de um predicado, neste caso (para a porta 5000):

```
http://localhost:5000/processa_json       % faz executar o predicado p_json/1.

:- http_handler('/processa_json', p_json, []).
```

3. Criação do servidor a escutar em Port

```
server(Port) :-
        http_server(http_dispatch, [port(Port)]).
```

4. Implementação do servidor que aguarda pela recepção de um JSON contendo um campo *set_user* que pode ser acedido com JSON.set_user, cria um novo termo JSON usando o nome João e o número enviado, por exemplo {"name":"joao", "number":3000} e responde ao cliente com este JSON.

```
p_json(Request) :-
        http_read_json(Request, JSON, [json_object(dict)]),
        % R = json([name=joao,number=3000]),
        % alternativa para não ter que declarar :-json_object
        R = student("joao",JSON.set_user),
        prolog_to_json(R, JSONObject),
        reply_json(JSONObject, [json_object(dict)]).
```

Cliente:
======

Cria um termo JSON (ex. {set_user: 3000}) e envia-o ao servidor, recebe a resposta como um JSON em formato dicionário (ex: para o json recebido {"name":"joao", "number":3000}, podemos aceder aos valores com Reply.name e Reply.number).

```
client(Number):-
        Term = json([set_user = Number]),
        http_post('http://localhost:5000/processa_json', json(Term), Reply, [json_object(dict)]),
        write('Client: '),write(Reply.name),nl,
        write('Client: ‘),write(Reply.number),nl.
```

Exemplo de utilização:

?- server(5000). % inicia servidor na porta 5000

?- client(1000). % cria JSON com o argumento, envia ao servidor e mostra a resposta.