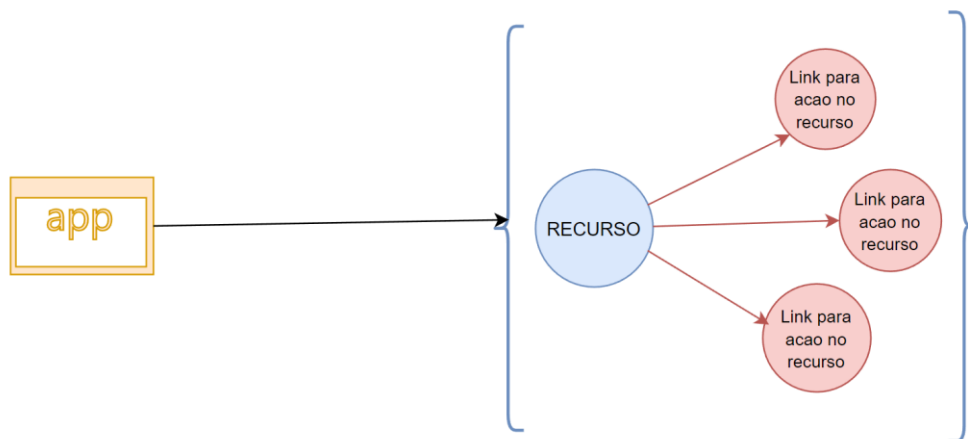




Hypermedia como motor da aplicação – HATEOAS

HATEOAS é uma das constraints que ajudam a atingir o patamar de RESTFUL em uma API e significa **Hypermedia as the engine of application state**, a ideia é **devolver junto com a resposta de uma requisição** os **links** que o cliente pode utilizar para fazer uma ação em um determinado recurso:



Spring HATEOAS – HATEOAS

O Spring tem um projeto que ajuda a aplicar esse padrão de HATEOAS:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-hateoas</artifactId>
</dependency>
```

A maneira mais simples de construir uma resposta que segue o HATEOAS é usando uma **entityModel<T>** para envelopar o **recurso**:



Os links serão adicionados no EntityModel

```
@GetMapping("/{idPedido}")
public ResponseEntity<EntityModel<PedidoDTOResponse>> buscarPedido(@PathVariable Long idPedido){
    var dtoResponsePedido = new PedidoDTOResponse(pedidoCrudService.buscarPedidoPorId(idPedido));
    var entityModelPedidoDTO : EntityModel<PedidoDTOResponse> = EntityModel.of(dtoResponsePedido);
```

Agora podemos construir links com o **WebMvcLinkBuilder**, aqui temos a facilidade de *usar métodos de uma classe*, então eu posso **construir um link com uma referência direta para outro método do**

Controller:

Os métodos de construção também são do WebMvcLinkBuilder

```
WebMvcLinkBuilder linkToAddItem = linkTo(methodOn(this.getClass()).adicionarItem(dtoResponsePedido.getId(), itemPedidoDTO: null));  
WebMvcLinkBuilder linkToRealizarPedido = linkTo(methodOn(this.getClass()).realizarPedido(dtoResponsePedido.getId()));  
WebMvcLinkBuilder linkToCancelarPedido = linkTo(methodOn(this.getClass()).cancelarPedido(dtoResponsePedido.getId()));
```

como estou construindo esses links na classe de controller, posso usar uma referência a classe com this

Agora é só adicionar os métodos criados no EntityModel especificando suas relações:

```
entityModelPedidoDTO.add(linkToGetPedido.withSelfRel());  
entityModelPedidoDTO.add(linkToAddItem.withRel("adicionar um item"));  
entityModelPedidoDTO.add(linkToRealizarPedido.withRel("realizar o pedido"));  
entityModelPedidoDTO.add(linkToCancelarPedido.withRel("cancelar o pedido"));  
  
return ResponseEntity.status(HttpStatus.CREATED).body(entityModelPedidoDTO);
```



```
{  
  "_links": {  
    "self": {  
      "href": "http://localhost:8080/pedidos/9"  
    },  
    "adicionar um item": {  
      "href": "http://localhost:8080/pedidos/9/adicionar-item"  
    },  
    "realizar o pedido": {  
      "href": "http://localhost:8080/pedidos/9/realizar"  
    },  
    "cancelar o pedido": {  
      "href": "http://localhost:8080/pedidos/9/cancelar"  
    }  
  }  
}
```