

# API REST

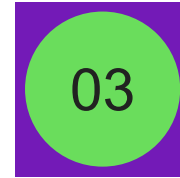




UPDATE



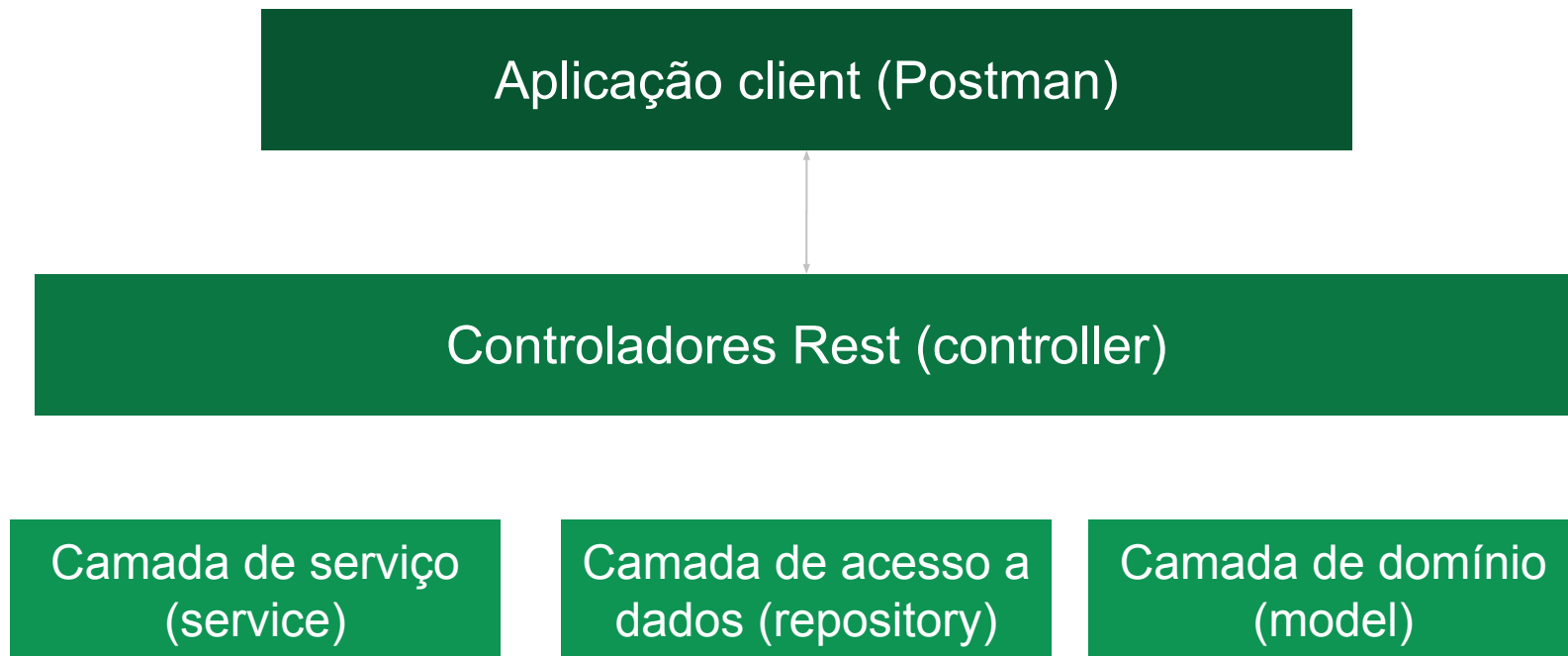
POST



DELETE

# Estrutura do projeto

---



# UPDATE

---

## **UsuarioService:**

```
public Usuario update(Integer id, Usuario obj) {  
    Usuario newObj = findById(id);  
    newObj.setNome(obj.getNome());  
    newObj.setSenha(obj.getSenha());  
    return repository.save(obj);  
}
```

## **UsuarioController:**

```
@PutMapping(value =("/{id}")
```

```
    public ResponseEntity<Usuario> update(@PathVariable Integer id, @RequestBody  
    Usuario obj) {  
        Usuario newObj = service.update(id, obj);  
        return ResponseEntity.ok().body(newObj);  
    }
```

# DELETE

---

## **UsuarioService:**

```
public void delete(Integer id) {  
    findById(id);  
    repository.deleteByld(id);  
}
```

## **UsuarioController:**

```
@DeleteMapping(value = "/{id}")  
public ResponseEntity<Void> delete(@PathVariable Integer id) {  
    service.delete(id);  
    return ResponseEntity.noContent().build();  
}
```

**.build()** = Construa a entidade de resposta sem corpo.

# POST

---

## **UsuarioService:**

```
public Usuario create(Usuario obj) {  
    obj.setId(null);  
    return repository.save(obj);  
}
```

## **UsuarioController:**

### **@PostMapping**

```
public ResponseEntity<Usuario> create(@RequestBody Usuario obj){  
    Usuario newObj = service.create(obj);  
    URI uri =  
    ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(newObj.getId()).toUri();  
    return ResponseEntity.created(uri).build();  
}
```

# POST

---

URI = import java.net.URI

ServletUriComponentsBuilder.*fromCurrentRequest()* = Prepare um construtor copiando o esquema, host, porta, caminho e string de consulta de um HttpServletRequest.

.path() = caminho

.buildAndExpand() = Construtor padrão.

VAMOS PRATICAR?