

```

public class CategoriaController {

    private CategoriaRepository categoriaRepository;

    public CategoriaController(CategoriaRepository categoriaRepository){
        this.categoriaRepository = categoriaRepository;
    }

    @GetMapping("/{id}")
    public Categoria getById(@PathVariable int id){
        return categoriaRepository.findById(id)
            .orElseThrow(() -> new ResponseStatusException(
                HttpStatus.NOT_FOUND, reason: "Perfil não encontrado."));
    }

    @PostMapping
    @ResponseStatus(HttpStatus.CREATED)
    public Categoria save(@RequestBody Categoria categoria) {
        return categoriaRepository.save(categoria);
    }

    @DeleteMapping("/{id}")
    @ResponseStatus(HttpStatus.NO_CONTENT)
    public void delete (@PathVariable int id){
        categoriaRepository.findById(id)
            .map(categoria -> {
                categoriaRepository.delete(categoria);
                return categoria;
            })
            .orElseThrow(() -> new ResponseStatusException(
                HttpStatus.NOT_FOUND, reason: "Perfil não encontrado"));
    }

    @PutMapping("/{id}")
    @ResponseStatus(HttpStatus.NO_CONTENT)
    public void update (@PathVariable int id,
                        @RequestBody Perfil perfil){
        categoriaRepository.findById(id)
            .map(categoriaExistente ->{
                perfil.setId(categoriaExistente.getId_categoria());
                categoriaRepository.save(categoriaExistente);
                return categoriaExistente;
            })
            .orElseThrow(() -> new ResponseStatusException(
                HttpStatus.NOT_FOUND, reason: "Perfil não encontrado."));
    }

    @GetMapping
    public List<Categoria> find(Categoria filtro){
        ExampleMatcher matcher = ExampleMatcher.matching()
            .withIgnoreCase()
            .withStringMatcher(

```