

## Dicas úteis:

## 1. @CreationTimestamp / @UpdateTimestamp

<https://thorben-janssen.com/persist-creation-update-timestamps-hibernate/>

## 2. Cors

```
@Configuration
public class ConfigurationCors {

    @SuppressWarnings({ "rawtypes", "unchecked" })
    @Bean
    public FilterRegistrationBean corsFilter() {
        UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
        CorsConfiguration config = new CorsConfiguration().applyPermitDefaultValues();
        config.addAllowedMethod(HttpMethod.PUT);
        config.addAllowedMethod(HttpMethod.DELETE);
        source.registerCorsConfiguration("/**", config);
        FilterRegistrationBean bean = new FilterRegistrationBean(new CorsFilter(source));
        bean.setOrder(0);
        return bean;
    }
}
```

## 1. @CreationTimestamp / @UpdateTimestamp

<https://thorben-janssen.com/persist-creation-update-timestamps-hibernate/>

## 2. Cors

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/CORS>

<https://spring.io/blog/2015/06/08/cors-support-in-spring-framework>

[https://github.com/aopaixao/residencia\\_api\\_restful/tree/main/configuration](https://github.com/aopaixao/residencia_api_restful/tree/main/configuration)

## 3. Insertable e Updatable no Mapeamento da Entidade

```
@ManyToOne
@JsonBackReference
@JoinColumn(name = "orderid", referencedColumnName = "orderid", insertable = false, updatable = false)
private Orders orders;

@Column(name = "orderid")
private Integer orderId;
```

## 4. Query e Native Query

```
Class ClienteEnderecoVO{  
  
    private String nomeCliente;  
    private Integer codigoCliente;  
    private String logradouro;  
    private Integer logradouroNumero;  
    private String logradouroComplemento;  
    private String logradouroBairro;  
    private String logradouroCidade;  
    private String logradouroUf;  
  
}
```

```
@Entity  
@Table(name = "cliente")  
public class Cliente{  
  
    @Id  
    @Column(name="codigo_cliente")  
    private Integer id;  
  
    @Column(name="nome_cliente")  
    private String nome;  
  
}
```

```
Class ClienteEnderecoVO{
```

```
    private String nomeCliente;  
    private Integer codigoCliente;  
    private String logradouro;  
    private Integer logradouroNumero;  
    private String logradouroComplemento;  
    private String logradouroBairro;  
    private String logradouroCidade;  
    private String logradouroUf;
```

```
}
```

```
@Entity
```

```
@Table(name = "cliente")
```

```
public class Cliente{
```

```
    @Id
```

```
    @Column(name="codigo_cliente")
```

```
    private Integer id;
```

```
    @Column(name="nome_cliente")
```

```
    private String nome;
```

```
}
```

```
public interface ClienteRepository extends JpaRepository<Cliente,Integer> {
```

```
    @Query(value= "SELECT" +
```

```
        "c.nome AS nomeCliente, c.id_cliente AS codigoCliente, " +
```

```
        "e.logradouro, e.logradouroNumero, e.logradouroComplemento, " +
```

```
        "e.logradouroBairro, e.logradouroCidade, e.logradouroUf " +
```

```
        "FROM cliente c INNER JOIN endereco e " +
```

```
        "ON (c.codigoCliente = e.codigoCliente) " +
```

```
        "WHERE c.codigoCliente = :codigoCliente ", nativeQuery = true)
```

```
    ClienteEnderecoVO listarEnderecoCliente(@Param("codigoCliente") TypedParameterValue id);
```

```
    @Query(value= "SELECT nome, id From Cliente Where id = ?1")
```

```
    Cliente listCodigoAndNome(Integer codigoCliente)
```

```
}
```