

Generation Brasil – Projeto Integrador

Projeto: Plataforma WEB – ColaborArt.

Tipo: E-commerce.

Representantes: Alana Rodrigues Teixeira / Bruno Falango / Gustavo Broch /

Laíse Lima Galvão de Souza / Vitor Carneiro Borela.

Back-End

Framework: Spring Boot

Link: <https://start.spring.io/>

Versão: 2.4.1

Projeto: Maven Project

Linguagem: Java

Versão: 8

Dependências Adicionadas:

- Spring Web
- Spring Boot DevTools
- Spring Data JPA
- Validation
- MySQL Driver

Banco de Dados: MySQL

Model

Usuário - Atributos:

- **private long idUsuario;**

Número de identificação do usuário dentro da *tb_usuario*.

(Exemplo: 1, 2, 3, etc.)

- **private String nome_completo;**

Nome completo do usuário na *tb_usuario*.

(Exemplo: Ana Claudia Vax)

- **private String email;**

Email do usuário na *tb_usuario*.

(Exemplo: anaclaudiavax@hotmail.com)

- **private BigDecimal valor;**

Valor do Produto.

(Exemplo: R\$22,50)

- **private String senha;**

Senha do usuário na *tb_usuario*.

(Exemplo: emprego123)

Usuário – Métodos:

- **getters / setters dos atributos do model;**

```
public long getIdUsuario() {  
    return idUsuario;  
}  
  
public void setIdUsuario(long idUsuario) {  
    this.idUsuario = idUsuario;  
}  
  
public String getNome_completo() {  
    return nome_completo;  
}  
  
public void setNome_completo(String nome_completo) {  
    this.nome_completo = nome_completo;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getSenha() {  
    return senha;  
}
```

```
public void setSenha(String senha) {  
    this.senha = senha;  
}
```

Repository

```
1 package br.com.colaborart.ecommerce.repository;  
2  
3 import org.springframework.data.jpa.repository.JpaRepository;  
4  
5  
6  
7 public interface UsuarioRepository extends JpaRepository<Usuario, Long> {  
8  
9 }  
10
```

Interface – UsuarioRepository

Criação do repositório do usuário, estendendo da Classe *JpaRepository*, onde recebe a **Usuário** Model e o tipo primitivo **Long**.

Controller

```
@Autowired  
private UsuarioRepository repositorio;
```

Atributo para implementação do repositório, através da injeção de dependências (@Autowired).

```
@GetMapping()  
public ResponseEntity<List<Usuario>> buscarTodosUsuarios() {  
    return ResponseEntity.ok(repositorio.findAll());  
}
```

Retorna todos os usuários encontrados no banco de dados.

```
@GetMapping("/{id}")  
public ResponseEntity<Produto> buscarProdutoPorId(@PathVariable long id){  
    return repositorio.findById(id).map(resposta -> ResponseEntity.ok(resposta))  
        .orElse(ResponseEntity.notFound().build());  
}
```

Retorna o Usuário de acordo com Id fornecido pelo usuário.

```

@PostMapping()
public ResponseEntity<Usuario> inserirUsuario(@RequestBody Usuario usuario) {
    return ResponseEntity.status(HttpStatus.CREATED).body(repositorio.save(usuario));
}

```

Inserir informações do Usuário no banco de dados.

```

@PutMapping()
public ResponseEntity<Usuario> atualizarUsuario(@RequestBody Usuario usuario){
    return ResponseEntity.status(HttpStatus.OK).body(repositorio.save(usuario));
}

```

Atualiza informações do Usuário no banco de dados, de acordo com o Id.

```

@DeleteMapping("/{id}")
public void deleteUsuario(@PathVariable long id) {
    repositorio.deleteById(id);
}

```

Apaga informações do Usuário no banco de dados.

CRUD – Postman

GET

<http://localhost:8080/usuario>

```

{
  "idUsuario": 1,
  "nome_completo": "Gustavo BC",
  "email": "guzinho93@hotmail.com",
  "senha": "Gu9319"
},
{
  "idUsuario": 2,
  "nome_completo": "Bruno FG",
  "email": "bruninhobruno43@hotmail.com",
  "senha": "Bru9819"
},

```

GET

<http://localhost:8080/usuario/5>

Body ▾ 200 OK 20 ms 349 B Save Response ▾

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "idUsuario": 5,
3   "nome_completo": "Alana RD",
4   "email": "analacrazyrt@pop.com.br",
5   "senha": "20bananas"
6 }
```

POST

<http://localhost:8080/usuario/>

Params Auth Headers (8) Body ● Pre-req. Tests Settings Cookies Code

raw JSON Beautify

```
1 {
2   "nome_completo": "Simba Lion King",
3   "email": "kingimba@hotmail.com",
4   "senha": "kimbinha123"
5 }
6
7
```

Body ▾ 201 Created 680 ms 362 B Save Response ▾

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "idUsuario": 6,
3   "nome_completo": "Simba Lion King",
4   "email": "kingimba@hotmail.com",
5   "senha": "kimbinha123"
6 }
```

PUT

<http://localhost:8080/usuario/>

Params Auth Headers (8) Body ● Pre-req. Tests Settings Cookies Code

raw JSON Beautify

```
1 {
2   "idUsuario": 6,
3   "nome_completo": "Simba Lion King",
4   "email": "kingimba@hotmail.com",
5   "senha": "kimbinha123"
6 }
7
```

Body ▾ 200 OK 929 ms 357 B Save Response ▾

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "idUsuario": 6,
3   "nome_completo": "Simba Lion King",
4   "email": "kingimba@hotmail.com",
5   "senha": "kimbinha123"
6 }
```

DELETE

<http://localhost:8080/usuario/6>

