

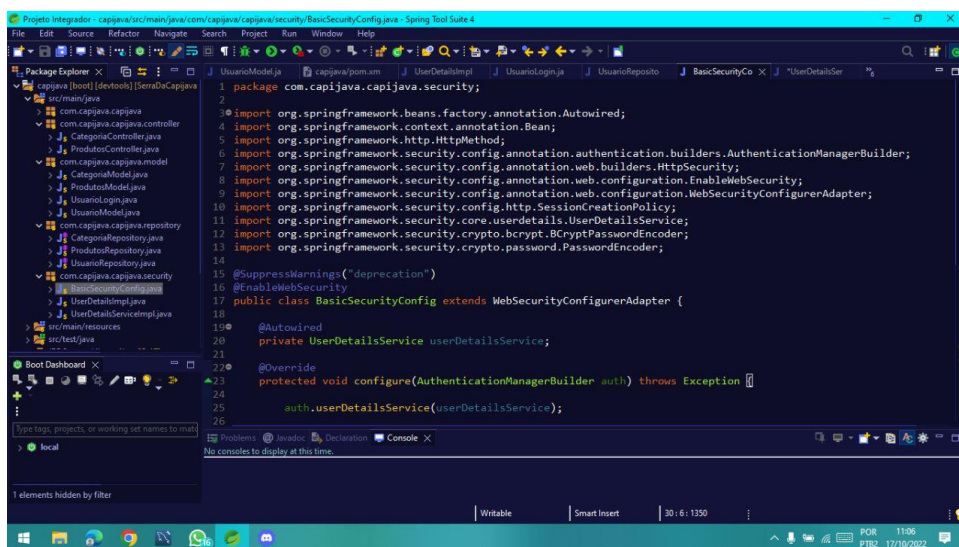
# PROJETO INTEGRADOR GRUPO 2 – SERRA DA CAPIJAVA

17/10/2022

HOJE FORAM CRIADAS AS CAMADAS : SECURITY , SERVICE E ATUALIZAÇÃO DAS CAMADAS JÁ EXISTENTES.

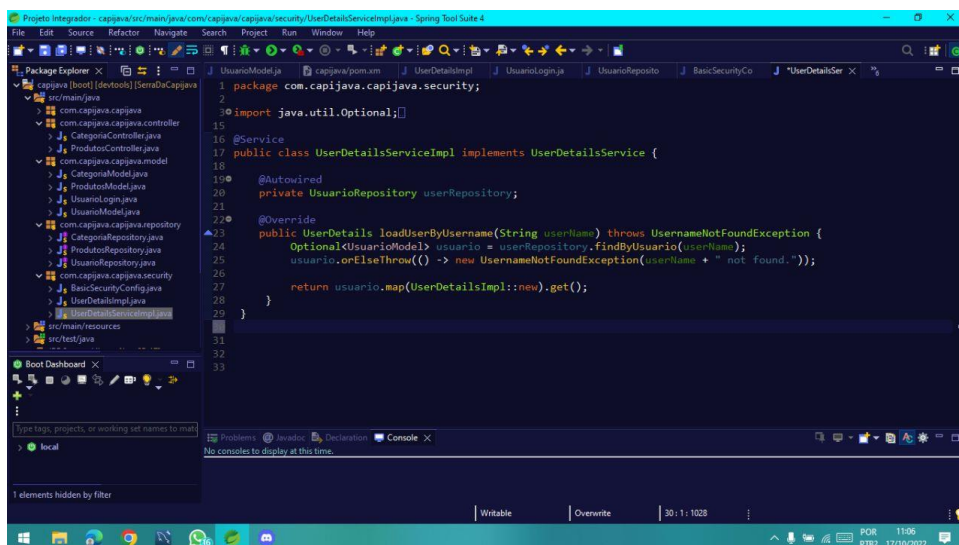
- CAMADA SECURITY (NOVA) - NESSA CAMADA FORAM CRIADAS AS CLASSES :

*BasicSecurityConfig*



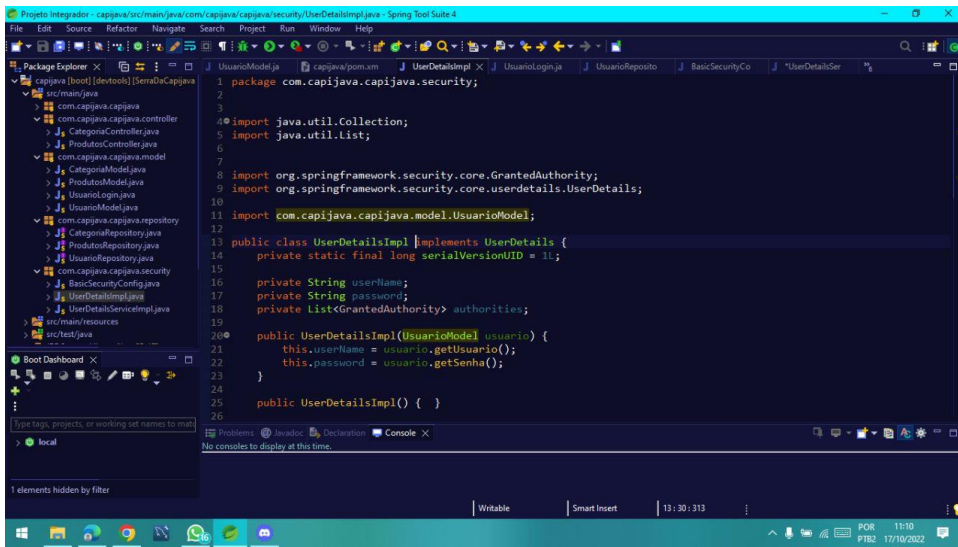
```
1 package com.capijava.capijava.security;
2
3
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.context.annotation.Bean;
6 import org.springframework.http.HttpMethod;
7 import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
8 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
9 import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
10 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
11 import org.springframework.security.config.http.SessionCreationPolicy;
12 import org.springframework.security.core.userdetails.UserDetailsService;
13 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
14 import org.springframework.security.crypto.password.PasswordEncoder;
15
16 @SuppressWarnings("deprecation")
17 @EnableWebSecurity
18 public class BasicSecurityConfig extends WebSecurityConfigurerAdapter {
19
20     @Autowired
21     private UserDetailsService userDetailsService;
22
23     @Override
24     protected void configure(AuthenticationManagerBuilder auth) throws Exception {
25
26         auth.userDetailsService(userDetailsService);
27     }
28 }
```

*UserDetailsServiceImpl*



```
1 package com.capijava.capijava.security;
2
3 import java.util.Optional;
4
5
6 @Service
7 public class UserDetailsServiceImpl implements UserDetailsService {
8
9     @Autowired
10     private UsuarioRepository usuarioRepository;
11
12     @Override
13     public UserDetails loadUserByUsername(String userName) throws UsernameNotFoundException {
14         Optional<UsuarioModel> usuario = usuarioRepository.findByUsername(userName);
15         usuario.orElseThrow(() -> new UsernameNotFoundException(userName + " not found."));
16
17         return usuario.map(UserDetailsServiceImpl::new).get();
18     }
19 }
```

```
UserDetailsImpl
```



- **CAMADA SERVICE (NOVA)**

### UsuarioService.java

[illegible]

## UserServiceImplService.java

```
1 package com.capjava.capjava.security;
2
3 import java.util.Optional;
4
5
6
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.security.core.userdetails.UserDetails;
9 import org.springframework.security.core.userdetails.UserDetailsService;
10 import org.springframework.security.core.userdetails.UsernameNotFoundException;
11 import org.springframework.stereotype.Service;
12
13 import com.capjava.capjava.model.UsuarioModel;
14 import com.capjava.capjava.repository.UsuarioRepository;
15
16 @Service
17 public class UserDetailsServiceImpl implements UserDetailsService {
18
19     @Autowired
20     private UsuarioRepository usuarioRepository;
21
22     @Override
23     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
24         Optional<UsuarioModel> usuario = usuarioRepository.findByUsername(username);
25         usuario.orElseThrow(() -> new UsernameNotFoundException(usuarioName + " not found."));
26
27         return usuario.map(UserDetailsImpl::new).get();
28     }
29 }
30
31
```

## UserServiceImplService.java

```
1 package com.capjava.capjava.security;
2
3 import java.util.Optional;
4
5
6
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.security.core.userdetails.UserDetails;
9 import org.springframework.security.core.userdetails.UserDetailsService;
10 import org.springframework.security.core.userdetails.UsernameNotFoundException;
11 import org.springframework.stereotype.Service;
12
13 import com.capjava.capjava.model.UsuarioModel;
14 import com.capjava.capjava.repository.UsuarioRepository;
15
16 @Service
17 public class UserDetailsServiceImpl implements UserDetailsService {
18
19     @Autowired
20     private UsuarioRepository usuarioRepository;
21
22     @Override
23     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
24         Optional<UsuarioModel> usuario = usuarioRepository.findByUsername(username);
25         usuario.orElseThrow(() -> new UsernameNotFoundException(usuarioName + " not found."));
26
27         return usuario.map(UserDetailsImpl::new).get();
28     }
29 }
30
31
```

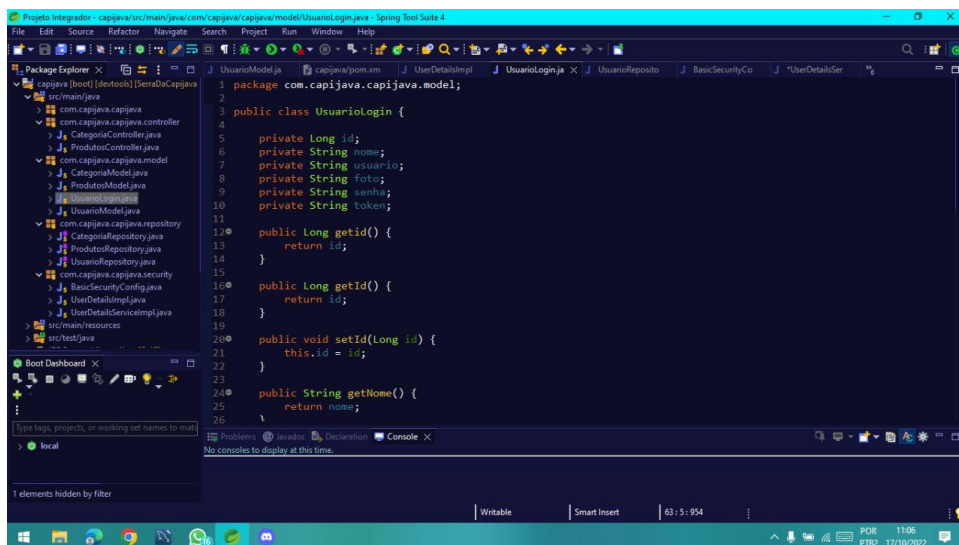
- CAMADA CONTROLLER (ATUALIZADA) - NESSA ATUALIZAÇÃO DA CAMADA , FOI CRIADA A CLASSE USUÁRIOCONTROLLER

## Classe UsuarioController.java

```
1 package com.capijava.capijava.controller;
2
3 import java.util.List;
4 import java.util.Optional;
5
6 import javax.validation.Valid;
7
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.http.HttpStatus;
10 import org.springframework.http.ResponseEntity;
11 import org.springframework.web.bind.annotation.CrossOrigin;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.PathVariable;
14 import org.springframework.web.bind.annotation.PostMapping;
15 import org.springframework.web.bind.annotation.PutMapping;
16 import org.springframework.web.bind.annotation.RequestBody;
17 import org.springframework.web.bind.annotation.RequestMapping;
18 import org.springframework.web.bind.annotation.RestController;
19
20 import com.capijava.capijava.model.UsuarioLogin;
21 import com.capijava.capijava.model.UsuarioModel;
22 import com.capijava.capijava.repository.UsuarioRepository;
23 import com.capijava.capijava.service.UsuarioService;
24
25 @RestController
26 @RequestMapping("/usuarios")
27 @CrossOrigin(origins = "*", allowedHeaders = "**")
28 public class UsuarioController {
29
30     @Autowired
31     private UsuarioService service;
32
33     @Autowired
34     private UsuarioRepository repository;
35
36     @RequestMapping("/all")
37     public ResponseEntity<List<UsuarioModel>> getAll() {
38         return ResponseEntity.ok(repository.findAll());
39     }
40
41     @RequestMapping("/{id}")
42     public ResponseEntity<UsuarioModel> getById(@PathVariable Long id) {
43         return repository.findById(id);
44     }
45
46     @PostMapping("/login")
47     public ResponseEntity<UsuarioLogin> autenticarUsuario(
48         @RequestBody Optional<UsuarioLogin> usuario) {
49         return service.login(usuario);
50     }
51
52     @PutMapping("/{id}")
53     public ResponseEntity<UsuarioModel> atualizarUsuario(
54         @PathVariable Long id,
55         @RequestBody UsuarioModel usuario) {
56         return service.atualizarUsuario(id, usuario);
57     }
58
59     @PostMapping("/register")
60     public ResponseEntity<UsuarioModel> criarUsuario(
61         @RequestBody UsuarioModel usuario) {
62         return service.criarUsuario(usuario);
63     }
64
65     @DeleteMapping("/{id}")
66     public ResponseEntity<Void> deletarUsuario(
67         @PathVariable Long id) {
68         return service.deletarUsuario(id);
69     }
70 }
```

- CAMADA MODEL (ATUALIZADA) - NESSA ATUALIZAÇÃO FORAM CRIADAS AS CLASSES - USUÁRIO E USUÁRIOLOGIN

## UsuarioLogin

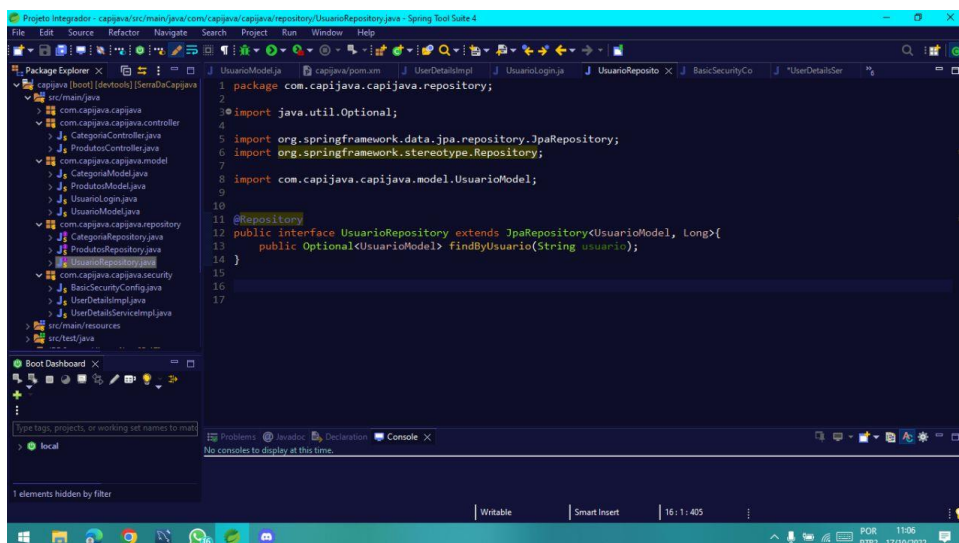


## UsuarioModel.java

[illegible]

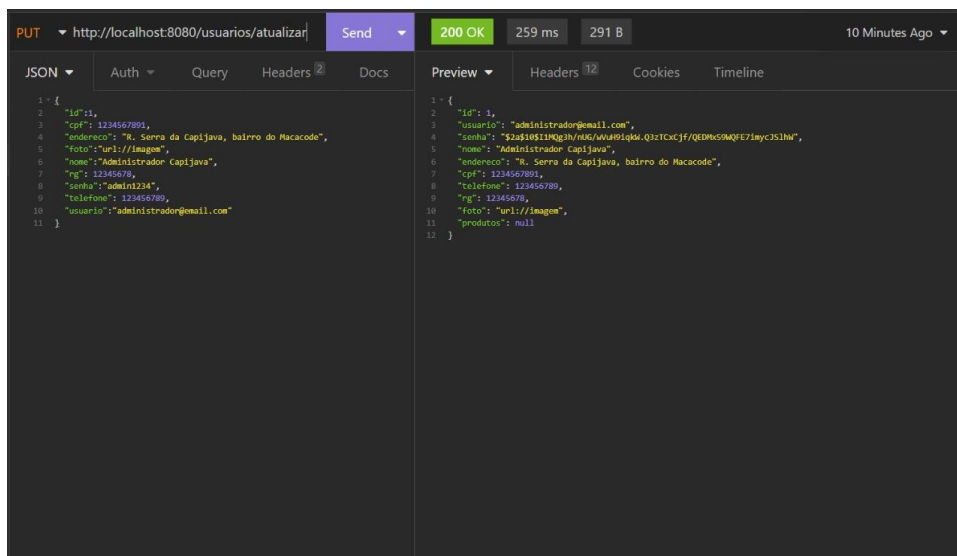
- **CAMADA REPOSITORY (ATUALIZADA) - NESSA ATUALIZAÇÃO FOI CRIADA A CLASSE**

### UsuarioRepository



## FORAM REALIZADOS TESTES NO INSOMNIA

### PUT

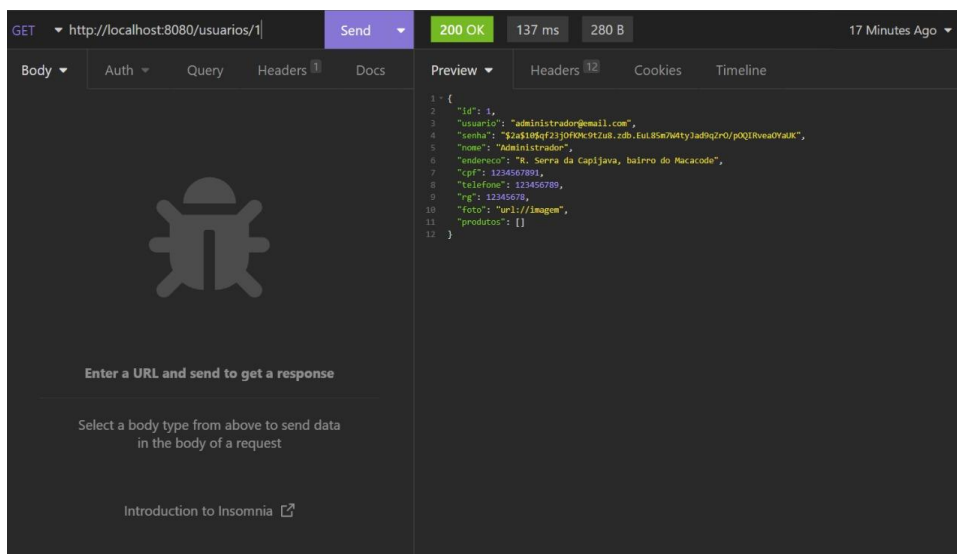


The screenshot shows the Insomnia interface for a PUT request. The URL is `http://localhost:8080/usuarios/atualizar/`. The status is **200 OK** with a response time of 259 ms and a body size of 291 B. The JSON body is displayed in the Preview tab.

```
1 {
2   "id": 1,
3   "cpf": 123456789,
4   "endereco": "R. Serra da Capijava, bairro do Macacode",
5   "foto": "url://imagem",
6   "nome": "Administrador Capijava",
7   "rg": 12345678,
8   "senha": "admin1234",
9   "telefone": 123456789,
10  "usuario": "administrador@email.com"
11 }
```

```
1 {
2   "id": 1,
3   "usuario": "administrador@email.com",
4   "senha": "$2a$10$11Rq3h/m4G/wU401qk4.Q3zTCVCjf/QEDH6S0MQFE7ImyC3S1ha",
5   "nome": "Administrador Capijava",
6   "endereco": "R. Serra da Capijava, bairro do Macacode",
7   "cpf": 123456789,
8   "telefone": 123456789,
9   "rg": 12345678,
10  "foto": "url://imagem",
11  "produtos": null
12 }
```

### GET - USUARIOS



The screenshot shows the Insomnia interface for a GET request. The URL is `http://localhost:8080/usuarios/1`. The status is **200 OK** with a response time of 137 ms and a body size of 280 B. The JSON body is displayed in the Preview tab.

```
1 {
2   "id": 1,
3   "usuario": "administrador@email.com",
4   "senha": "$2a$10$cf23j0f0Kc9tZu8.zdb.EuL85m7Nety7adnqzr0/poQIRveaOYakM",
5   "nome": "Administrador",
6   "endereco": "R. Serra da Capijava, bairro do Macacode",
7   "cpf": 123456789,
8   "telefone": 123456789,
9   "rg": 12345678,
10  "foto": "url://imagem",
11  "produtos": []
12 }
```

### GET – LOCALIZANDO USUARIOS

GET http://localhost:8080/usuarios/all Send 200 OK 439 ms 282 B 18 Minutes Ago

Body Auth Query Headers 1 Docs

Preview Headers 12 Cookies Timeline

```

1 [
2 {
3   "id": 1,
4   "usuario": "administrador@email.com",
5   "senha": "$2a$10$qf23j0f0K9tZu8.zdb.Eul85w74tYJad9qZr0/p0Q1Rvea0YaUK",
6   "nome": "Administrador",
7   "endereco": "R. Serra da Capijava, bairro do Macacoe",
8   "cpf": 123456789,
9   "telefone": 123456789,
10  "rg": 12345678,
11  "foto": "url://imagem",
12  "produtos": []
13 }
14 ]

```

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

[Introduction to Insomnia](#)

## POST – LOGIN USUARIO

POST http://localhost:8080/usuarios/login Send 200 OK 107 ms 265 B 21 Minutes Ago

JSON Auth Query Headers 1 Docs

Preview Headers 12 Cookies Timeline

```

1 {
2   "usuario": "administrador@email.com",
3   "senha": "admin1234"
4 }

```

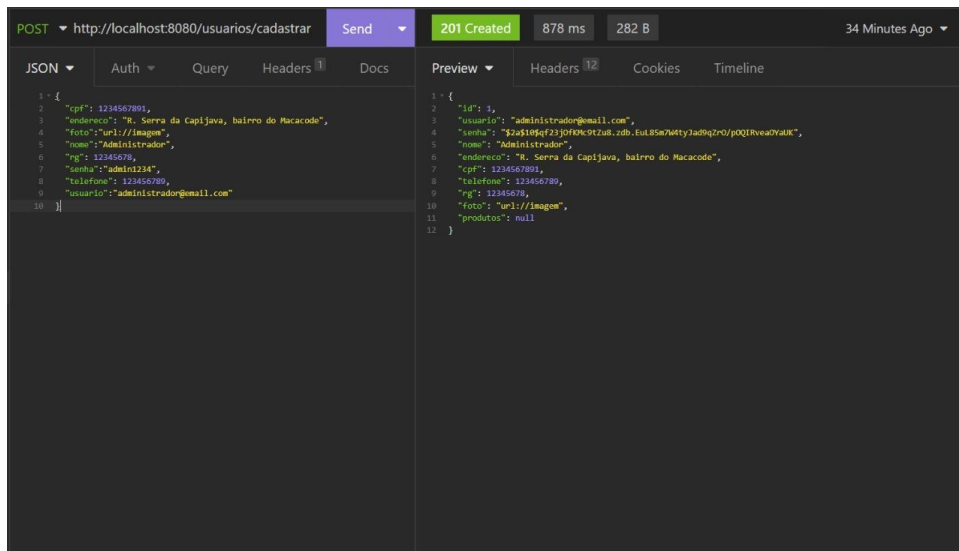
```

1 {
2   "id": 1,
3   "cpf": 0,
4   "rg": 0,
5   "telefone": 0,
6   "endereco": null,
7   "nome": "Administrador",
8   "usuario": "administrador@email.com",
9   "foto": "url://imagem",
10  "senha": "$2a$10$qf23j0f0K9tZu8.zdb.Eul85w74tYJad9qZr0/p0Q1Rvea0YaUK",
11  "token": "Basic YWltak5pc3hyYWRvck81bW9pbC5jb280YWRtaWwtaW9pM0"
12 }

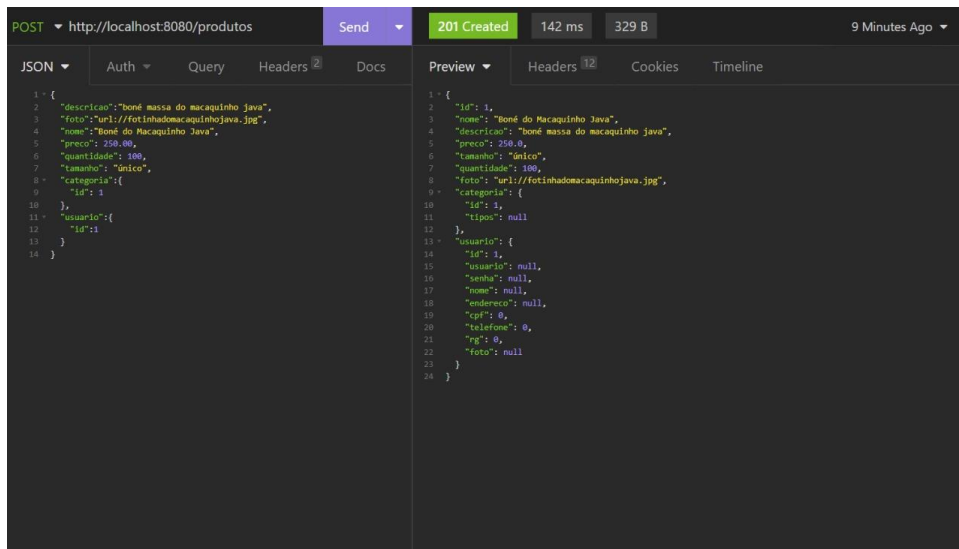
```

## POST – CADASTRO USUÁRIO





## POST – PRODUTOS



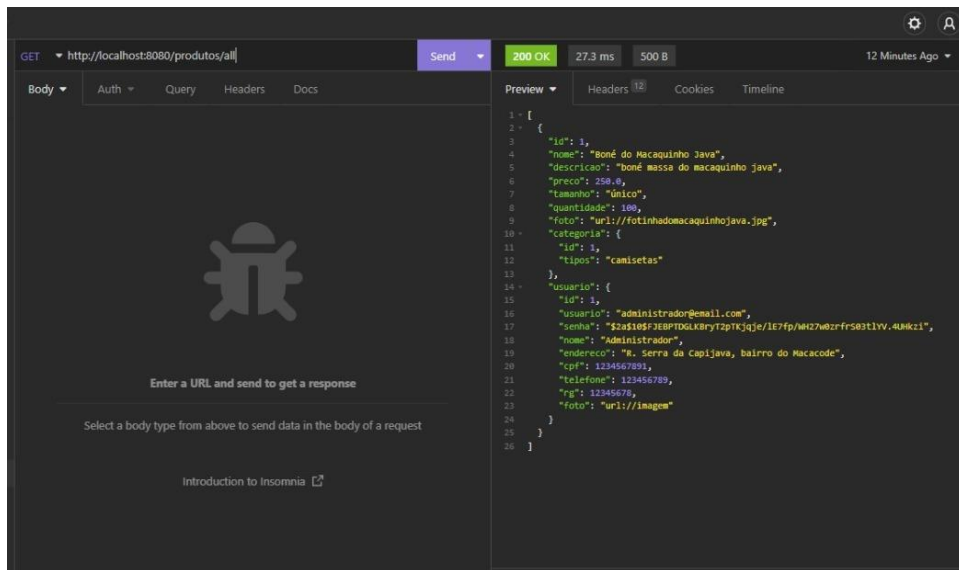
## REALIZADA AUTORIZAÇÃO PARA QUE OS USUÁRIOS POSSAM TER AUTONIMA NO LOGIN !

```

37● @Override
38 protected void configure(HttpSecurity http) throws Exception {
39
40     http.authorizeRequests().antMatchers("/usuarios/login").permitAll().antMatchers("/usuarios/cadastrar")
41         .permitAll().antMatchers(HttpMethod.GET, "/produtos/all").permitAll().
42         antMatchers(HttpMethod.OPTIONS).permitAll().anyRequest().authenticated().and().httpBasic()
43         .and().sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS).and().cors().and()
44         .csrf().disable();
45
46
47

```





NA DATA DE HOJE TUDO FOI REALIZADO COM SUCESSO , DESDE INCLUSÃO DE NOVAS, CAMADAS, CLASSES E TESTES INSOMNIA .