

Dentro do Spring, alterar a model de Usuario e de UsuarioLogin, colocando os dados de "tipo" e "foto"

---Lembrar de gerar os getters e setters

```
private String foto;  
private String tipo;
```

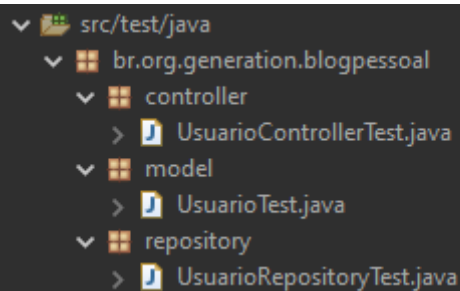
preencher os dados novos dentro do construtor:

```
public Usuario(long id, String nome, String usuario, String senha, LocalDate dataNascimento, String tipo, String foto) {  
    this.id = id;  
    this.nome = nome;  
    this.usuario = usuario;  
    this.senha = senha;  
    this.dataNascimento = dataNascimento;  
    this.tipo = tipo;  
    this.foto = foto;  
}
```

Dentro da UsuarioService, colocar os novos campos para persistirem no momento que o usuário logar:

```
public Optional<UsuarioLogin> loginUsuario(Optional<UsuarioLogin> usuarioLogin) {  
    BCryptPasswordEncoder encoder = new BCryptPasswordEncoder();  
    Optional<Usuario> usuario = usuarioRepository.findByUsuario(usuarioLogin.get().getUsuario());  
  
    if (usuario.isPresent()) {  
        if (encoder.matches(usuarioLogin.get().getSenha(), usuario.get().getSenha())) {  
  
            String auth = usuarioLogin.get().getUsuario() + ":" + usuarioLogin.get().getSenha();  
            byte[] encodedAuth = Base64.encodeBase64(auth.getBytes(Charset.forName("US-ASCII")));  
            String authHeader = "Basic " + new String(encodedAuth);  
  
            usuarioLogin.get().setId(usuario.get().getId());  
            usuarioLogin.get().setNome(usuario.get().getNome());  
            usuarioLogin.get().setSenha(usuario.get().getSenha());  
            usuarioLogin.get().setTipo(usuario.get().getTipo());  
            usuarioLogin.get().setFoto(usuario.get().getFoto());  
            usuarioLogin.get().setToken(authHeader);  
  
            return usuarioLogin;  
        }  
    }  
}
```

Colocar os novos dados do usuário dentro dos arquivos de teste:



```
src/test/java  
├── br.org.generation.blogpessoal  
│   ├── controller  
│   │   └── UsuarioControllerTest.java  
│   ├── model  
│   │   └── UsuarioTest.java  
│   └── repository  
│       └── UsuarioRepositoryTest.java
```

Vai ficar desta forma:

```
LocalDate dataAdmin = LocalDate.parse("1990-07-22", DateTimeFormatter.ofPattern("yyyy-MM-dd"));
usuarioAdmin = new Usuario(0L, "Administrador", "admin@email.com.br", "admin123", dataAdmin, "admin", "foto");
```

Depois desses passos executados no STS, refazer o deploy para o Heroku, com os comandos **git add. / git commit -m "mensagem de commit" / git push heroku master**

Pelo site do Heroku, deletar o banco de dados e recriar ele, pois foram alteradas colunas na estrutura do banco.

