## CAMADA BASIC SECURITY

### MODEL:

Dentro da camada MODEL, adicionamos uma classe UsuarioLogin.

### REPOSITORY

Editamos a interface UsuarioRepository que já havia sido criada, e alteramos o método por um FindByEmail.

### SEGURANÇA

Criamos 3 classes: (i)BasicSecuritConfig, (ii) UserDetailsImpl , e (iii) UserDetailsServiceImpl.

A BasicSecuritConfig faz a configuração básica de acesso, isto é, confirma se o acesso é ou não seguro.

Por meio dessa classe foi possível determinais quais acessos são liberados para o cliente sem que seja necessário usar um token.

A UserDetailsImpl criou uma autorização de acesso ao usuário (relativos ao e-mail e senha).

Finalmente, a UserDetailsServiceImpl criamos um método para fazer uma **consulta** ao banco de dados, para checar se já existe um e-mail igual cadastrado no branco de dados. Se o e-mail não existir, retornará uma autorização e solicitação para salvar o e-mail digitado.

### SERVICE

Criamos a classe UsuarioService, responsável pela criação das regras de negócio. Ou seja, é uma classe de configuração.

Dentro dela, criamos dois métodos: um que se refere ao campo cadastrar, e outro, ao campo logar.

No método cadastrar, criamos uma condição que impossibilita a criação de dois usuários com com dados que já foram inseridos no banco de dados.

No método logar, criamos a condição que criptografa as senhas inseridas pelos usuários (no campo senha). Ressaltamos que é um padrão de segurança básico (US-ASCII).

### CONTROLLER

Na classe UsuarioController, adicionamos alguns métodos além dos já existentes, quais sejam: a criação de um método post com o end-point “/cadastrar”, o qual faz um try-catch para verificar

se o usuário já existe no banco de dados (este método usa o método CadastrarUsuário criado na camada de service).

Já o método post com o end-point “login” faz uma autenticação da senha, ou seja: verifica se a senha digitada pelo usuário no momento do login é igual à senha utilizada para cadastro (a senha salva no banco de dados).

Caso a senha digitada pelo usuário não seja a mesma da senha cadastrada no banco de dados, o login não será autorizado. Se a senha conferir com a do banco de dados, o login será autorizado e devolverá um token criptografado (com um prefixo basic).

**JSON QUE DEVE SER INSERIDO PARA CADASTRO:**


{

"nome":"Bilbo Bolseiro",

"email":"bilbobolseiro@sustentart.com",

"senha":"123564556",

"telefone":"123456789101"

}


**JSON QUE DEVERÁ SER RETORNADO NO LOGIN:**


{

"email": "bilbobolseiro@sustentart.com",

"senha": "$2a$10$Z6cQfXiWahBL6iOuJj0W6e90oYVu1lpB2ujrQYsQRbOheA0LHwNJW",
"token": "Basic YmlsYm9AazoxMjM1NjQ1NTY="

}