Índice

[Definições 2](#_Toc66882827)

[Motivação 2](#_Toc66882828)

[Classe 2](#_Toc66882829)

[Repositórios 2](#_Toc66882830)

[Funções 2](#_Toc66882831)

[Multi Língua 2](#_Toc66882832)

[[bool] ValidateRequest ( DtoLoginInformation ) 3](#_Toc66882833)

[[bool] Exec 3](#_Toc66882834)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data | Tarefa | Versão | Responsável | Motivo |
| 12/03/2021 | 3012 | 1 | R. Groff | Criação do documento |

# Definições

## Motivação

Este serviço processa a autorização de usuários na plataforma de serviços.

# Classe

## Repositórios

Para acesso ao banco de dados, teremos interfaces para:

* CartaoRepository;
* EmpresaRepository;

# Funções

## Multi Língua

Para mensagens ao usuário, teremos as listas abaixo:

new List<string> // english

{

"Ops, something went wrong", /\* 0 \*/

"Invalid user authentication", /\* 1 \*/

},

new List<string> // spanish

{

"Ops, algo salió mal", /\* 0 \*/

"Autenticación de usuario no válida", /\* 1 \*/

},

new List<string> // pt-br

{

"Ops, algo deu errado", /\* 0 \*/

"Autenticação inválida", /\* 1 \*/   
},

## [bool] ValidateRequest ( DtoLoginInformation )

Inicio ValidateRequest ( **DtoLoginInformation** dto )  
 Se dto.userType == **TipoUsuario**.Associado  
 Se dto.empresa == null OU  
 dto.matricula == null OU  
 dto.venc == null OU  
 dto.codAcesso == null OU  
 dto.senha == null  
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se   
 Fim Se  
Fim ValidateRequest

## [bool] Exec

Inicio Exec (...)  
 **LocalNetwork** network  
 **DtoLoginInformation** dto  
 ref **DtoAuthenticatedUser** loggedUser

Se ValidateRequest (dto) == FALSO  
 retornar FALSO  
 Fim Se  
  
 var **db** = GetConnection(network)

Se dto.userType == **TipoUsuario**.Associado   
 Então  
 var **t\_emp** = empresaRepository.GetEmpresaNum ( db, dto.empresa )  
 Se t\_emp == null   
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se  
 Se t\_emp.bBlocked == VERDADEIRO  
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se  
 var **titularidade** = 1  
 var **t\_associado** = cartaoRepository.GetCartao ( db, t\_emp.id, dto.matricula, titularidade) Se t\_associado == null   
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se

var **senhaDes** = DESdeCript ( t\_associado.stSenha )  
 Se senhaDes <> dto.senha  
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se  
 Se t\_associado.nuStatus <> CartaoStatus.Habilitado  
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se  
 Se t\_associado.stVenctoCartao <> dto.venc  
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se  
 var **codAcesso** = ObterCodigoAcesso (...)  
 t\_emp.nuEmpresa  
 t\_associado.nuMatricula  
 titularidade  
 t\_associado.nuViaCartao  
 t\_associado.stCpf  
   
   
 Se codAcesso <> dto.codAcesso  
 Então  
 Erro = mensagem de erro 1  
 retornar FALSO  
 Fim Se

loggedUser.\_id = t\_associado.id  
 loggedUser.\_type = **TipoUsuario**.Associado   
 loggedUser.email = “”  
 loggedUser.terminal = “”  
 loggedUser.nome = t\_associado.stNome  
 loggedUser.empresa = t\_emp.nuEmpresa  
 loggedUser.matricula = t\_associado.nuMatricula  
   
Fim Se

Caso Exceção  
 Erro = mensagem de erro 1  
 retornar FALSO  
Fim Exceção

Fim Exec