

## **Issues Authenticator**

### **Casos de Teste de Unidade**

## Histórico da Revisão

| Data       | Versão | Descrição                  | Autor                    | Revisor |
|------------|--------|----------------------------|--------------------------|---------|
| 02/12/2017 | 1.0    | Criação dos casos de teste | Alexandre, Iuri, Rodrigo |         |
|            |        |                            |                          |         |
|            |        |                            |                          |         |
|            |        |                            |                          |         |
|            |        |                            |                          |         |
|            |        |                            |                          |         |
|            |        |                            |                          |         |
|            |        |                            |                          |         |

### CASOS DE TESTE SERVIDOR

Caso de Teste 1: Obter preferências de laboratório

#### Identificação

**Classe** PreferenciasLaboratorio

**Método** obter(usuario\_lab\_id)

#### Descrição:

O método recebe como entrada o id do usuário do laboratório, e retorna uma lista de objetos da classe PreferenciasLaboratorio com todas as preferências do usuário, que estão salvas no banco, como atributos para cada laboratório.

#### Pré-condições para o caso:

Usuário de id 'usuario\_lab\_id' registrado no banco de dados.

**Conjunto de valores:** (id\_usuario, id\_lab ,temp\_min, temp\_max, ,iluminacao\_min, iluminacao\_max, umidade\_min, umidade\_max).

| Conjunto de Valores |                       |  |   |  |
|---------------------|-----------------------|--|---|--|
|                     | Usuário não existente | Usuário possui preferências para 1 laboratório | Usuário possui preferências para 2 laboratórios                   | Usuário possui preferências para 3 laboratórios  |
| usuario_lab_id      | 0                     | 111  | 222   | 333  |
| Resultado Esperado  |                       | [111, 1, 20, 30, 0, 1, 37, 40]                 | [222, (1, 2), (20,20), (30,30), (0,0), (1, 1), (37, 37), (40,40)] | [333, (1, 2, 3), (20,20, 20), (30,30, 30), (0,0, 0), (1, 1, 1), (37, 37, 37), (40,40, 40)] |

|                         |         |                                |   |  |
|-------------------------|---------|--------------------------------|---|--|
| <b>Resultado Obtido</b> |         | [123, 1, 20, 30, 0, 1, 37, 40] | [222, (1, 2), (20,20), (30,30), (0,0), (1, 1), (37, 37), (40,40)] | [333, (1, 2, 3), (20,20, 20), (30,30, 30), (0,0, 0), (1, 1, 1), (37, 37, 37), (40,40, 40)] |
| <b>Sucesso/Falha</b>    | Sucesso | Sucesso                        | Sucesso   | Sucesso  |

**Caso de Teste 2: Salvar preferências do laboratório****Identificação****Classe PreferenciasLaboratorio****Método salvar()****Descrição:**

O método salvar não recebe nenhuma entrada, utilizando somente atributos da classe PreferenciasLaboratorio ('usuario\_lab\_id', 'lab\_id', 'temperatura\_min', 'temperatura\_max', 'luminosidade\_min', 'luminosidade\_max', 'umidade\_min' e 'umidade\_max'). Caso este objeto exista no banco de dados, ele é atualizado com os valores de cada atributo da classe PreferenciasLaboratorio, caso contrário é criado um objeto com estes valores.

**Pré-condições para o caso:**

Nenhuma

| Conjunto de Valores       |                                |                                |
|---------------------------|--------------------------------|--------------------------------|
|                           | Objeto já existe no banco      | Objeto não existe no banco     |
| <b>usuario_lab_id</b>     | 123                            | 124                            |
| <b>lab_id</b>             | 1                              | 1                              |
| <b>temperatura_min</b>    | 20                             | 20                             |
| <b>temperatura_max</b>    | 30                             | 30                             |
| <b>luminosidade_min</b>   | 0                              | 0                              |
| <b>luminosidade_max</b>   | 1                              | 1                              |
| <b>umidade_min</b>        | 37                             | 37                             |
| <b>umidade_max</b>        | 40                             | 40                             |
| <b>Resultado Esperado</b> | [123, 1, 20, 30, 0, 1, 37, 40] | [124, 1, 20, 30, 0, 1, 37, 40] |
| <b>Resultado Obtido</b>   | [123, 1, 20, 30, 0, 1, 37, 40] | [124, 1, 20, 30, 0, 1, 37, 40] |
| <b>Sucesso/Falha</b>      | Sucesso                        | Sucesso                        |

**Caso de Teste 3: Recuperar preferências****Identificação****Função** `get_preferencias()`**Descrição:**

Não possui entradas, retorna um json com todas as preferências do usuário cujo id ('usuario\_lab\_id') está no header Authorization.

**Pré-condições para o caso:**

O usuário em questão existe no banco de dados.

| Conjunto de Valores       |                |                    |
|---------------------------|----------------|--------------------|
|                           | Usuário existe | Usuário não existe |
| <b>usuario_lab_id</b>     | 123            | 124                |
| <b>Resultado Esperado</b> |                | []                 |
| <b>Resultado Obtido</b>   |                | []                 |
| <b>Sucesso/Falha</b>      | Sucesso        | Sucesso            |

**Caso de Teste 4: Inserir preferências****Identificação****Função** `post_preferencias(id)`**Descrição:**

Recebe o id do laboratório como entrada ('id'), e um id de usuário ('usuario\_lab\_id', através do header Authorization) com suas preferências ('data', através de um arquivo json), retornando 'Bad Request' caso nenhum json seja enviado. Caso as preferências estejam incompatíveis com os valores da zona de conforto, é retornado um json com os mesmos. Caso não haja nenhum erro, as preferências são salvas e é retornada uma string vazia.

**Pré-condições para o caso:**

O usuário e laboratório em questão existem no banco

| Conjunto de Valores   |   |   |                                     |                               |
|-----------------------|---|---|-------------------------------------|-------------------------------|
|                       | Usuário não existe, mas laboratório sim | Laboratório não existe, mas usuário sim | Nem usuário nem laboratório existem | Usuário e laboratório existem |
| <b>id</b>             | 123                                     | 124                                     | 123                                 | 124                           |
| <b>usuario_lab_id</b> | 111                                     | 222                                     | 222                                 | 111                           |
| <b>data</b>           | Bad Request                             | Bad Request                             | Bad Request                         |                               |

|                           |             |             |             |         |
|---------------------------|-------------|-------------|-------------|---------|
| <b>Resultado Esperado</b> | Bad Request | Bad Request | Bad Request |         |
| <b>Resultado Obtido</b>   | Bad Request | Bad Request | Bad Request |         |
| <b>Sucesso/Falha</b>      | Sucesso     | Sucesso     | Sucesso     | Sucesso |

**Caso de Teste 5: Obter lista de laboratórios autorizados****Identificação:****Classe Laboratorio****Método obter\_laboratorios\_autorizados(user\_id)****Descrição:**

Recebe o id do usuário como entrada, retorna uma lista('data') vazia caso não existam laboratórios autorizados para o usuário ou uma lista('result') com o id e nome('lab\_id', 'lab\_nome') dos laboratórios autorizados.

**Pré-condições para o caso:**

O usuário e laboratório em questão existem no banco

| Conjunto de Valores       |                                 |                                     |                                 |
|---------------------------|---------------------------------|-------------------------------------|---------------------------------|
|                           | Laboratório nem usuário existem | Usuário existe, mas laboratório não | Existe um ou mais laboratórios  |
| <b>user_id</b>            | 124                             | 124                                 | 123                             |
| <b>result</b>             | -                               | -                                   | ['111','Laboratório de Dexter'] |
| <b>data</b>               | []                              | []                                  | -                               |
| <b>Resultado Esperado</b> | []                              | []                                  | ['111','Laboratório de Dexter'] |
| <b>Resultado Obtido</b>   | []                              | []                                  | ['111','Laboratório de Dexter'] |
| <b>Sucesso/Falha</b>      | Sucesso                         | Sucesso                             | Sucesso                         |

**Caso de Teste 6: Obter dados da zona de conforto****Identificação****Classe Laboratorio****Método obter\_zona\_de\_conforto(self)****Descrição:**

Recebe a id do laboratório('lab\_id') como entrada, e retorna uma lista('d') com os dados da zona de conforto do mesmo ou uma lista('data') vazia caso não encontre dados.

**Pré-condições para o caso:**

Laboratório com id 'lab\_id' existe no banco de dados

| Conjunto de Valores |                        |  |  |
|---------------------|------------------------|--|--|
|                     | Laboratório não existe | Laboratório existe mas não possui zona de conforto | Laboratório e zona de conforto existem   |
| lab_id              | 111                    | 222  | 333  |
| data                | []                     | []   | -  |
| d                   | -                      | -  | [333, (1, 2, 3), (20,20, 20), (30,30, 30), (0,0, 0), (1, 1, 1), (37, 37, 37), (40,40, 40)] |
| Resultado Esperado  | []                     | []   | [333, (1, 2, 3), (20,20, 20), (30,30, 30), (0,0, 0), (1, 1, 1), (37, 37, 37), (40,40, 40)] |
| Resultado Obtido    | []                     | []   | [333, (1, 2, 3), (20,20, 20), (30,30, 30), (0,0, 0), (1, 1, 1), (37, 37, 37), (40,40, 40)] |
| Sucesso/Falha       | Sucesso                | Sucesso  | Sucesso  |

**Caso de Teste 7: Obter preferências do laboratório****Identificação:****Classe Laboratorio****Método obter\_do\_laboratorio(usuario\_lab\_id, lab\_id)****Descrição:**

Recebe como entrada o id do usuário('usuario\_lab\_id') e o id do laboratório('lab\_id'), e retorna um objeto da classe PreferenciasLaboratorio com as preferências do usuário para o laboratório em questão ou uma lista vazia('data') caso usuário ou laboratório não existam.

**Pré-condições para o caso:**

Usuário com id 'usuario\_lab\_id' e laboratório com id 'lab\_id' existem no banco de dados.

| Conjunto de Valores |                                   |                                    |                               |
|---------------------|-----------------------------------|------------------------------------|-------------------------------|
|                     | Usuário e laboratório não existem | Usuário ou laboratório não existem | Usuário e laboratório existem |
| usuario_lab_id      | 111                               | 222                                | 333                           |
| lab_id              | 123                               | 124                                | 125                           |
| data                | []                                | []                                 | -                             |
| Valor de saída      | []                                | []                                 | [333, (1, 2, 3), (20,20, 20), |

|                         |         |         |  |
|-------------------------|---------|---------|--|
|                         |         |         | (30,30, 30), (0,0, 0), (1, 1, 1), (37, 37, 37), (40,40, 40)]                               |
| <b>Resultado Obtido</b> | []      | []      | [333, (1, 2, 3), (20,20, 20), (30,30, 30), (0,0, 0), (1, 1, 1), (37, 37, 37), (40,40, 40)] |
| <b>Sucesso/Falha</b>    | Sucesso | Sucesso | Sucesso  |

## CASOS DE TESTE APLICATIVO

### Caso de Teste 8: Salvar credenciais

#### Identificação:

**Função salvar()**

#### Descrição:

A função salvar recebe as entradas de login e senha do usuário, salva-os e envia-os através de API para o servidor. Caso o login e a senha sejam iguais aos dos registrados, o usuário tem acesso liberado.

#### Pré-condições para o caso:

Nenhuma.

| Conjunto de Valores       |                                 |                                    |                                    |
|---------------------------|---------------------------------|------------------------------------|------------------------------------|
|                           | Login existe e senha correta    | Login existe e senha errada        | Login não existe                   |
| <b>Resultado Esperado</b> | Entrada registrada com sucesso. | Não foi possível registrar entrada | Não foi possível registrar entrada |
| <b>Resultado Obtido</b>   | Entrada registrada com sucesso. | Não foi possível registrar entrada | Não foi possível registrar entrada |
| <b>Sucesso/Falha</b>      | Sucesso                         | Sucesso                            | Sucesso                            |

### Caso de Teste 9: Registrar entrada

#### Identificação:

**Função entrada()**

**Descrição:**

A função entrada faz uso do login e da senha armazenados e envia-os através de API para o servidor para ser registrado como log. Caso o login e a senha sejam iguais aos dos registrados, o usuário tem acesso liberado.

**Pré-condições para o caso:**

As credenciais do usuário (login e senha) devem estar salvas no aplicativo.

| Conjunto de Valores       |                                 |                                    |                                    |
|---------------------------|---------------------------------|------------------------------------|------------------------------------|
|                           | Login existe e senha correta    | Login existe e senha errada        | Login não existe                   |
| <b>Resultado Esperado</b> | Entrada registrada com sucesso. | Não foi possível registrar entrada | Não foi possível registrar entrada |
| <b>Resultado Obtido</b>   | Entrada registrada com sucesso. | Não foi possível registrar entrada | Não foi possível registrar entrada |
| <b>Sucesso/Falha</b>      | Sucesso                         | Sucesso                            | Sucesso                            |

**Caso de Teste 3: Registrar saída**

**Identificação:**

**Função saída()**

**Caso de Teste :**

**Descrição:**

Faz uso do login e senha armazenados e envia-os através de API para o servidor para ser registrado como log.

**Pré-condições para o caso:**

As credenciais do usuário (login e senha) devem estar salvas no aplicativo.

| Conjunto de Valores       |                                  |  |
|---------------------------|----------------------------------|--|
|                           | Credenciais salvas no aplicativo | Credenciais não estão salvas no aplicativo |
| <b>Resultado Esperado</b> | Saída registrada com sucesso     | Não foi possível registrar saída           |
| <b>Resultado Obtido</b>   | Saída registrada com sucesso     | Não foi possível registrar saída           |
| <b>Sucesso/Falha</b>      | Sucesso                          | Sucesso                                    |



**Caso de Teste 10: Mostrar preferências**

**Identificação:**

**Função** preferencias()

**Descrição:**

O aplicativo envia as credenciais do usuário (login e senha), através de API, para o servidor. A função, após receber as preferências em json, mostra-as para o usuário, caso as tenha encontrado.

**Pré-condições para o caso:**

As credenciais do usuário (login e senha) devem estar salvas no aplicativo.

| Conjunto de Valores       |   |                                     |
|---------------------------|---|-------------------------------------|
|                           | Preferências encontradas  | Preferências não encontradas        |
| <b>Resultado Esperado</b> | [("Laboratório de Dexter", "Laboratório de Engenharia de Software"), (20, 20), (30, 30), (0, 0), (1, 1), (37, 37), (40,40)] | Não foi possível obter preferencias |
| <b>Resultado Obtido</b>   | [("Laboratório de Dexter", "Laboratório de Engenharia de Software"), (20, 20), (30, 30), (0, 0), (1, 1), (37, 37), (40,40)] | Não foi possível obter preferencias |
| <b>Sucesso/Falha</b>      | Sucesso   | Sucesso                             |

**Caso de Teste 11: Mostrar preferências dos laboratórios**

**Identificação:**

**Função** select\_nome\_laboratorio()

**Descrição:**

O aplicativo envia as credenciais do usuário (login e senha), através de API, para o servidor. A função, após receber o nome em json de todos os laboratórios do usuário, mostra-as para o usuário, caso as tenha encontrado.

**Pré-condições para o caso:**

As credenciais do usuário (login e senha) devem estar salvas no aplicativo.

| Conjunto de Valores       |   |                                     |
|---------------------------|---|-------------------------------------|
|                           | Laboratórios encontrados  | Laboratórios não encontrados        |
| <b>Resultado Esperado</b> | [("Laboratório de Dexter", "Laboratório de Engenharia de Software"), (20, 20), (30, 30), (0, 0), (1, 1), (37, 37), (40,40)] | Não foi possível obter preferencias |

|                         |  |                                     |
|-------------------------|--|-------------------------------------|
|                         | de Software”]  |                                     |
| <b>Resultado Obtido</b> | [“Laboratório de Dexter”, “Laboratório de Engenharia de Software”] | Não foi possível obter preferencias |
| <b>Sucesso/Falha</b>    | Sucesso  | Sucesso                             |

#### Caso de Teste 12: Mostrar notificações

##### Identificação

##### Função notificacoes()

##### Descrição:

O aplicativo envia as credenciais do usuário (login e senha), através de API, para o servidor. A função, após receber as notificações em json, mostra-as para o usuário, caso as tenha encontrado.

##### Pré-condições para o caso:

As credenciais do usuário (login e senha) devem estar salvas no aplicativo.

| Conjunto de Valores       |  |                                     |
|---------------------------|--|-------------------------------------|
|                           | Notificações encontradas   | Notificações não encontradas        |
| <b>Resultado Esperado</b> | [‘Não foi possível recuperar identificação para obter laboratorios’] | Não foi possível obter notificações |
| <b>Resultado Obtido</b>   | [‘Não foi possível recuperar identificação para obter laboratorios’] | Não foi possível obter notificações |
| <b>Sucesso/Falha</b>      | Sucesso  | Sucesso                             |