



CRUD User

POST TU70-admin-adicionar-dentista-valido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ADMIN, a fim de cadastrar um novo dentista.

Resultado: Status 200 OK. ADMIN consegue cadastrar um novo dentista.

- 01/06: Status 200 Ok. Admin consegue cadastrar um novo dentista e alocá-lo em uma equipe

HEADERS

Content-Type

application/json

Authorization

```
{{authAdminH}}
```

BODY raw

```
{
  "email": "dentista5@dentista.com",
  "senha": "dentista",
  "papel": "DENTISTA",
  "nome": "Dentista 5 da Silva",
  "idEquipe": 2
}
```

Example Request

TU70-admin-adicionar-dentista-valido



```
--data-raw '{
  "email": "dentista5@dentista.com",
  "senha": "dentista",
  "papel": "DENTISTA",
  "nome": "Dentista 5 da Silva",
```

[View More](#)

POST TU71-admin-adicionar-dentista-email-repetido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ADMIN, a fim de adicionar um novo dentista com um e-mail já utilizado

Resultado: Status 400. Não é possível criar um novo dentista com um e-mail já utilizado.

HEADERS

Content-Type

application/json

Authorization

```
{{authAdminH}}
```

BODY raw

```
{
  "email": "dentista3@dentista.com",
  "senha": "dentista",
  "papel": "DENTISTA",
  "nome": "Dentista 3 da Silva"
}
```

Example Request

TU71-admin-adicionar-dentista-email-repetido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
```



```
"senha": "dentista",  
"papel": "DENTISTA",  
"nome": "Dentista 3 da Silva"  
'
```

POST *TU72-admin-adicionar-dentista-email-formato-nome-nulo

```
{{sbms-usuario}}
```

Um administrador consegue adicionar um dentista com e-mail sem formato pré-definido (@ e .com). Também pode adicionar um usuário sem nome. (Formato sera verificado no front)

HEADERS

Content-Type

application/json

Authorization

```
{{authAdminH}}
```

BODY raw

```
{  
  "email": "dentistadentista",  
  "senha": "dentista",  
  "papel": "DENTISTA",  
  "nome": "Dentista da Silva"  
}
```

Example Request

*TU72-admin-adicionar-dentista-email-formato-nome-nulo

```
curl --location -g --request POST '{{sbms-usuario}}' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: {{authAdminH}}' \  
--data-raw '{  
  "email": "dentistadentista",  
  "senha": "dentista",  
  "papel": "DENTISTA",  
  "nome": "Dentista da Silva"  
}'
```



POST TU73-admin-adicionar-coordenador-valido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ADMIN, a fim de cadastrar um novo coordenador.

Resultado: Status 200 OK. ADMIN consegue cadastrar um novo coordenador.

HEADERS

Content-Type

application/json

Authorization

```
{{authAdminH}}
```

BODY raw

```
{
  "email": "coordenador6@coordenador.com",
  "papel": "COORDENADOR",
  "senha": "coordenador",
  "nome": "Coordenador 6 da Silva"
}
```

Example Request

TU73-admin-adicionar-coordenador-valido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminH}}' \
--data-raw '{
  "email": "coordenador6@coordenador.com",
  "papel": "COORDENADOR",
  "senha": "coordenador",
  "nome": "Coordenador 6 da Silva"
}'
```



POST TU74-admin-adicionar-coordenador-email-repetido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ADMIN, a fim de adicionar um novo coordenador com um e-mail já utilizado.

Resultado: Status 400. Não é possível criar um novo coordenador com um e-mail já utilizado.

HEADERS

Content-Type

application/json

Authorization

```
{{authAdminH}}
```

BODY raw

```
{
  "email": "coordenador1@coordenador.com",
  "papel": "COORDENADOR",
  "senha": "coordenador",
  "nome": "Coordenador 1 da Silva"
}
```

Example Request

TU74-admin-adicionar-coordenador-email-repetido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminH}}' \
--data-raw '{
  "email": "coordenador1@coordenador.com",
  "papel": "COORDENADOR",
  "senha": "coordenador",
  "nome": "Coordenador 1 da Silva"
}'
```



POST TU75-admin-adicionar-arrolador-valido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ADMIN, a fim de cadastrar um novo arrolador.

Resultado: Status 200 OK. ADMIN consegue cadastrar um novo arrolador.

HEADERS

Content-Type

application/json

Authorization

```
{{authAdminH}}
```

BODY raw

```
{
  "nome": "Arrolador 3 da Silva",
  "email": "arrolador3@arrolador.com",
  "papel": "ARROLADOR",
  "senha": "arrolador"
}
```

Example Request

TU75-admin-adicionar-arrolador-valido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminH}}' \
--data-raw '{
  "nome": "Arrolador 3 da Silva",
  "email": "arrolador3@arrolador.com",
  "papel": "ARROLADOR",
  "senha": "arrolador"
}'
```



POST TU76-admin-adicionar-arrolador-email-repetido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ADMIN, a fim de adicionar um novo arrolador com um e-mail já utilizado.

Resultado: Status 400. Não é possível criar um novo arrolador com um e-mail já utilizado.

HEADERS

Content-Type

application/json

Authorization

Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoiQURNSU4iLCJpYXQiOiJlY2NTA5OTQ3NzYsImV4cCI6MTY1MTA4MTE3Niwic3ViljoiMSJ9.IRPdHBvZdbDw3FPg1eE_-Kk4wSldMzCb4FPw9fi984

BODY raw

```
{
  "email": "arrolador1@arrolador.com",
  "papel": "ARROLADOR",
  "senha": "arrolador",
  "nome": "Arrolador 1 da Silva"
}
```

Example Request

TU76-admin-adicionar-arrolador-email-repetido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoiQURNSU4iLCJpYXQiOiJlY2NTA5OTQ3NzYsImV4cCI6MTY1MTA4MTE3Niwic3ViljoiMSJ9.IRPdHBvZdbDw3FPg1eE_-Kk4wSldMzCb4FPw9fi984' \
--data-raw '{
  "email": "arrolador1@arrolador.com",
  "papel": "ARROLADOR",
  "senha": "arrolador",
  "nome": "Arrolador 1 da Silva"
}'
```




POST TU78-admin-adicionar-assistente-valido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ADMIN, a fim de cadastrar um novo assistente.

Resultado: Status 200 OK. ADMIN consegue cadastrar um novo assistente.

HEADERS

Content-Type

application/json

Authorization

{{authAdminH}}

BODY raw

```
{
  "nome": "Assistente 3 da Silva",
  "email": "assistente3@assistente.com",
  "papel": "ASSISTENTE",
  "senha": "assistente"
}
```

Example Request

TU78-admin-adicionar-assistente-valido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminH}}' \
--data-raw '{
  "nome": "Assistente 3 da Silva",
  "email": "assistente3@assistente.com",
  "papel": "ASSISTENTE",
  "senha": "assistente"
}'
```




POST TU80-coordenador-adicionar-papel-invalido

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário COORDENADOR, a fim de cadastrar um novo coordenador.

Resultado: Status 401 Unauthorized. COORDENADOR não consegue cadastrar um novo COORDENADOR ou ADMIN.

HEADERS

Content-Type

application/json

Authorization

Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJyLCJpYXQiOiJlE2NTI3MzM4NzgsImV4cCI6MTY2MTI4NzQ3OH0.wjt1MaUCqMVRlZgrLJGnpJxYRJFkDPfII3p3UjDPryl

BODY raw

```
{
  "email": "coordenador5@coordenador.com",
  "papel": "COORDENADOR",
  "senha": "coordenador",
  "nome": "Coordenador 5 da Silva"
}
```

Example Request

TU80-coordenador-adicionar-papel-invalido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJyLCJpYXQiOiJlE2NTI3MzM4NzgsImV4cCI6MTY2MTI4NzQ3OH0.wjt1MaUCqMVRlZgrLJGnpJxYRJFkDPfII3p3UjDPryl' \
--data-raw '{
  "email": "coordenador5@coordenador.com",
  "papel": "COORDENADOR",
  "senha": "coordenador",
  "nome": "Coordenador 5 da Silva"
}'
```


POST TU82-coordenador-adicionar-arrolador-valido

{{sbms-usuario}}

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users' juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário COORDENADOR, a fim de cadastrar um novo arrolador.

Resultado: Status 200 OK. Coordenador consegue cadastrar um novo arrolador.

HEADERS

Content-Type
application/json

Authorization
{{authCoordH}}

BODY raw

```
{
  "email": "arrolador8@arrolador.com",
  "papel": "ARROLADOR",
  "senha": "arrolador",
  "nome": "Arrolador 9 da Silva",
  "idEquipe": 2
}
```

Example Request

TU82-coordenador-adicionar-arrolador-valido

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authCoordH}}' \
--data-raw '{
  "email": "arrolador8@arrolador.com",
  "papel": "ARROLADOR",
  "senha": "arrolador",
  "nome": "Arrolador 9 da Silva",
  "idEquipe": 2
}'
```

[View More](#)

POST TU84-dentista-adicionar-usuario

```

{{sbms-usuario}}

```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users', juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário DENTISTA, a fim de criar um novo usuário.

Resultado: Status 401: Unauthorized. Dentista não consegue adicionar um novo usuário.

HEADERS

Content-Type

```
application/json
```

Authorization

Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoieVVOVElTVEEiLCJpYXQiOiJlE2NTIwMzQ0NTQsImV4cCI6MTY1MjlyMDg1NCwic3ViljoICI9.IWGLVwMZGCJsNYdqJ6afJdyUDnCzbg8SlpvgjU2NgY

BODY raw

```
{
  "email": "assistente11@assistente.com",
  "papel": "ASSISTENTE",
  "senha": "assistente11",
  "nome": "Assistente 11 da Silva"
}
```

Example Request

TU84-dentista-adicionar-usuario

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2x1IjoiREV0VElTVEEiLCJpYXQiOiIyMDI0LTA1LTA1IiwiaWF0IjE1ODU0MjUwMDAwfQ.' \
--data-raw '{
    "email": "assistente11@assistente.com",
    "papel": "ASSISTENTE",
    "senha": "assistente11",
}
```



POST TU85-assistente-adicionar-usuario

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users', juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ASSISTENTE, a fim de criar um novo usuário.

Resultado: Status 401: Unauthorized. Assistente não consegue adicionar um novo usuário.

HEADERS

Content-Type

application/json

Authorization

Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xiIjoiaWNTSVNURU5URSImlhdCI6MTY1MDk5NjEwMywiZXhwIjoxNjUxMDgyNTAzLCJzdWIiOiI2In0.3dmlWzfVbb9hmO9edfDqfQZZ3cdsbTM8BozOAZ_zOMI

BODY raw

```
{
  "email": "assistente10@assistente.com",
  "papel": "ASSISTENTE",
  "senha": "assistente10",
  "nome": "Assistente 10 da Silva"
}
```

Example Request

TU85-assistente-adicionar-usuario

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xiIjoiaWNTSVNURU5URSImlhdCI6MTY1MDk5NjEwMywiZXhwIjoxNjUxMDgyNTAzLCJzdWIiOiI2In0.3dmlWzfVbb9hmO9edfDqfQZZ3cdsbTM8BozOAZ_zOMI' \
--data-raw '{
  "email": "assistente10@assistente.com",
  "papel": "ASSISTENTE",
  "senha": "assistente10",
  "nome": "Assistente 10 da Silva"
}'
```




POST TU86-arrolador-adicionar-usuario

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users', juntamente com o JSON preenchido com os dados: email, senha, papel e nome além do token de autenticação do usuário ARROLADOR, a fim de criar um novo usuário.

Resultado: Status 401: Unauthorized. Arrolador não consegue adicionar um novo usuário.

HEADERS

Content-Type

application/json

Authorization

Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoiQVJST0xBRE9SIiwiaWF0IjoxNjUxMDAwMjg4LCJleHAiOiE2NTEwODY2ODgsInN1Yil6Ijg1QXpQsfsDwhjmm9BBbcF-QaP3NVgSE2w6JU5oRPwjx6_o

BODY raw

```
{
  "email": "assistente10@assistente.com",
  "papel": "ASSISTENTE",
  "senha": "assistente10",
  "nome": "Assistente 10 da Silva"
}
```

Example Request

TU86-arrolador-adicionar-usuario

```
curl --location -g --request POST '{{sbms-usuario}}' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoiQVJST0xBRE9SIiwiaWF0IjoxNjUxMDAwMjg4LCJleHAiOiE2NTEwODY2ODgsInN1Yil6Ijg1QXpQsfsDwhjmm9BBbcF-QaP3NVgSE2w6JU5oRPwjx6_o' \
--data-raw '{
  "email": "assistente10@assistente.com",
  "papel": "ASSISTENTE",
  "senha": "assistente10",
  "nome": "Assistente 10 da Silva"
}'
```



GET TU87-admin-mostrar-todos-usuarios

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/users', necessário token de autenticação do usuário ADMIN, a fim de verificar a lista de todos os usuários do sistema.

Resultado: Status 200: OK. Administrador consegue ver todos os usuários do sistema.

HEADERS

Authorization

```
{{authAdminH}}
```

Example Request

TU87-admin-mostrar-todos-usuarios

```
curl --location -g --request GET '{{sbms-usuario}}' \
--header 'Authorization: {{authAdminH}}'
```

GET TU88-mostrar-usuarios-por-equipe

```
{{sbms-usuario}}
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/users', necessário token de autenticação do usuário COORDENADOR, a fim de verificar a lista dos usuários de sua própria equipe.

Resultado: Status 200: OK. Coordenador consegue ver lista de sua própria equipe. Demais usuários não conseguem ver essa lista.

HEADERS

Authorization

```
{{authCoordH}}
```



Example Request

TU88-mostrar-usuarios-por-equipe

```
curl --location -g --request GET '{{sbms-usuario}}' \
--header 'Authorization: {{authCoordH}}'
```

POST TU89-admin-desativar-usuario

```
{{sbms-usuario}}/13/deactivate
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/users/9/deactivate' informando o token de autenticação do usuário ADMIN, a fim de desativar um usuário.

Resultados:

- 02/05: Status 200 OK. Admin consegue desativar um usuário já existente.

HEADERS**Authorization**

```
{{authAdmin}}
```

Example Request

TU89-admin-desativar-usuario

```
curl --location -g --request POST '{{sbms-usuario}}/13/deactivate' \
--header 'Authorization: {{authAdmin}}'
```

POST TU90-coordenador-desativar-usuario

```
{{sbms-usuario}}/13/deactivate
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Descrição: É requisitado a url 'localhost:3000/users/12/deactivate' informando o token de autenticação do usuário COORDENADOR, a fim de desativar um usuário.

Resultados:



- 16/05: Status 404: Not Found. Quando um Coordenador tenta desativar um usuário de outra equipe, aparece que este usuário não existe.

HEADERS

Authorization

{{authCoordH}}

Example Request

TU90-coordenador-desativar-usuario

```
curl --location -g --request POST '{{sbms-usuario}}/13/deactivate' \
--header 'Authorization: {{authCoordH}}'
```

GET TU91-admin-mostrar-lista-equipes

localhost:3000/teams

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/teams' juntamente com o token de autenticação do usuário ADMIN, a fim de mostrar a lista de equipes criadas.

Resultado:

- 23/05: Status 200 Ok. Exibe lista de equipes criadas com o nome dos coordenadores de cada uma delas, além do ID destas.

HEADERS

Authorization

{{authAdminH}}

Example Request

TU91-admin-mostrar-lista-equipes

```
curl --location --request GET 'localhost:3000/teams' \
```



POST TU92-ativar-usuario

```
{{sbms-usuario}}/18/activate
```

Headers: São as variáveis que indicam o tipo de conteúdo (JSON) e o token de autenticação.

Body: São dados, o "corpo" do teste, no nosso caso o JSON referente aos dados que serão requisitados e testados.

Descrição: É requisitado a url 'localhost:3000/users/:id/activate' juntamente com o token de autenticação do usuário ADMIN, a fim de ativar um usuário desativado.

Resultado:

- 25/05: Status 200 Ok. Mostra os dados do usuário e ativa-o novamente. Testado com usuário Administrador e usuário Coordenador e ambos podem ativar/desativar usuários. Foi testado também com os papéis Arrolador, Assistente e Dentista e todos deram permissão negada.

HEADERS

Authorization

```
{{authAdminH}}
```

Example Request

TU92-ativar-usuario

```
curl --location -g --request POST '{{sbms-usuario}}/18/activate' \  
--header 'Authorization: {{authAdminH}}'
```

POST heroku-adicionar-papeis-validos

```
{{heroku}}users
```

Foram executados os testes TU70, TU73, TU75, TU78, TU81, TU82, TU83 nesse caso utilizando a URL do Heroku (<http://saude-bucal-backend.herokuapp.com/>) e foi possível criar usuários com os papéis descritos nos testes.

HEADERS

Content-Type

{{authAdminH}}

BODY raw

```
{
  "email": "assistente5@assistente.com",
  "senha": "assistente5",
  "papel": "ASSISTENTE",
  "nome": "Assistente 5 da Silva",
  "idEquipe": 3
}
```

Example Request

heroku-adicionar-papeis-validos

```
curl --location -g --request POST '{{heroku}}users' \
--header 'Content-Type: application/json' \
--header 'Authorization: {{authAdminH}}' \
--data-raw '{
  "email": "assistente5@assistente.com",
  "senha": "assistente5",
  "papel": "ASSISTENTE",
  "nome": "Assistente 5 da Silva",
  "idEquipe": 3
}'
```

View More

POST heroku-ativar-usuario

```
{{heroku}}users/18/activate
```

Foi executado o teste TU92, nesse caso utilizando a URL do Heroku (<http://saude-bucal-backend.herokuapp.com/>) e foi possível ativar um usuário previamente criado e desativado.

HEADERS

Authorization

{{authAdminH}}

Example Request

heroku-ativar-usuario

POST heroku-desativar-usuario

{{heroku}}users/1/deactivate

Foram executados os testes TU89 e TU90, nesse caso utilizando a URL do Heroku (<http://saude-bucal-backend.herokuapp.com/>) e foi possível desativar um usuário previamente criado.

HEADERS

Authorization

Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImIhdCI6MTY1NDU0MTA4MSwiZXhwIjoxNjU0NjI3NDgxfQ.mmQBP9BdLhtKHnD2hmkEadHzHfjE-DaB4zi_4c7u6AI

Example Request

heroku-desativar-usuario

curl --location -g --request POST '{{heroku}}users/1/deactivate' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImIhdCI6MTY1NDU0MTA4MSwiZXhwIjoxNjU0NjI3NDgxfQ.mmQBP9BdLhtKHnD2hmkEadHzHfjE-DaB4zi_4c7u6AI'

GET heroku-admin-mostrar-lista-equipres

{{heroku}}teams

Foi executado o teste TU91, nesse caso utilizando a URL do Heroku (<http://saude-bucal-backend.herokuapp.com/>) e foi possível visualizar a lista de equipes com o usuário Administrador.

HEADERS

Authorization

{{authAdminH}}

Example Request

heroku-admin-mostrar-lista-equipres



```
curl --location -g --request GET '{{heroku}}teams' \
```