**Tabla de contenidos**

[1. Usuarios y niveles de autorización 3](#_Toc30961963)

[2. Login 4](#_Toc30961964)

[2.1. Gestor 4](#_Toc30961965)

[2.2. Cliente 4](#_Toc30961966)

[3. Métodos para el gestor 5](#_Toc30961967)

[3.1. Obtener gestores 5](#_Toc30961968)

[3.2. Obtener gestor por id 6](#_Toc30961969)

[3.3. Obtener gestor por nombre de usuario 7](#_Toc30961970)

[3.4. Obtener clientes 8](#_Toc30961971)

[3.5. Obtener cliente por id 9](#_Toc30961972)

[3.6. Obtener cliente por nombre de usuario 10](#_Toc30961973)

[3.7. Obtener mensajes recibidos 11](#_Toc30961974)

[3.8. Obtener mensajes enviados 11](#_Toc30961975)

[3.9. Obtener mensaje por id 12](#_Toc30961976)

[3.10. Obtener transferencias 13](#_Toc30961977)

[3.11. Obtener transferencia por id 14](#_Toc30961978)

[3.12. Agregar gestor 15](#_Toc30961979)

[3.13. Agregar cliente 16](#_Toc30961980)

[3.14. Enviar mensaje 17](#_Toc30961981)

[3.15. Eliminar gestor por id 18](#_Toc30961982)

[3.16. Eliminar cliente por id 19](#_Toc30961983)

[3.17. Eliminar mensajes 20](#_Toc30961984)

[3.18. Eliminar mensaje por id 20](#_Toc30961985)

[4. Métodos para el cliente 21](#_Toc30961986)

[4.1. Obtener información 21](#_Toc30961987)

[4.2. Obtener transferencias 22](#_Toc30961988)

[4.3. Obtener transferencia por id 23](#_Toc30961989)

[4.4. Realizar transferencia 24](#_Toc30961990)

# Usuarios y niveles de autorización

|  |  |
| --- | --- |
| **Usuario** | **Nivel de autorización** |
| gestor | 3 |
| cliente | 2 |
| público | 1 |

# Login

## Gestor

|  |  |
| --- | --- |
| **Breve descripción** | Permite autenticar a un gestor. |
| **Nivel de autorización** | 1 |
| **Ruta** | /login/gestor/ |
| **Método HTTP** | POST |
| **Cabecera HTTP** |  |
| **Cuerpo de la petición** | Par clave/valor utilizando el tipo de contenido (x-www-form-urlencoded:   1. usuario (string): nombre del gestor a autenticar. 2. password (string): password del gestor. |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee los siguientes campos si la autenticación es correcta:    1. token: token de autenticación para el gestor. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/login/gestor/>  Cuerpo de la petición (x-www-form-urlencoded):  usuario: gestor1  password: gestor1 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "token": " node012fxec1s005vf1e5uciv3khme30"      },      "ok": true  } |
| **Posibles errores** | 1. Autenticación incorrecta. 2. Error interno del servidor. |

## Cliente

|  |  |
| --- | --- |
| **Breve descripción** | Permite autenticar a un cliente. |
| **Nivel de autorización** | 1 |
| **Ruta** | /login/cliente/ |
| **Método HTTP** | POST |
| **Cabecera HTTP** |  |
| **Cuerpo de la petición** | Par clave/valor utilizando el tipo de contenido (x-www-form-urlencoded:   1. usuario (string): nombre del cliente a autenticar. 2. password (string): password del cliente. |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee los siguientes campos si la autenticación es correcta:    1. token: token de autenticación para el cliente. |
| **Ejemplo de petición correcta** | URL: [http://127.0.0.1:8085/login/client](http://127.0.0.1:8085/login/cliente)e/  Cuerpo de la petición (x-www-form-urlencoded):  usuario: cliente1  password: cliente1 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "token": " node012fxec1s005vf1e5uciv3khme30"      },      "ok": true  } |
| **Posibles errores** | 1. Autenticación incorrecta. 2. Error interno del servidor. |

# Métodos para el gestor

## Obtener gestores

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener la lista de todos los gestores. |
| **Nivel de autorización** | 3 |
| **Ruta** | /gestores/ |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (array): posee un array de objetos JSON con la información de cada uno de los gestores. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/gestores>/  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": [          {              "correo": "gestor1@mail.com",              "usuario": "gestor1",              "id": 1          },          {              "correo": "gestor2@mail.com",              "usuario": "gestor2",              "id": 2          },          {              "correo": "gestor3@mail.com",              "usuario": "gestor3",              "id": 3          }      ],      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes 2. Error interno del servidor |

## Obtener gestor por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener un gestor a partir de su id |
| **Nivel de autorización** | 3 |
| **Ruta** | /gestores/:id |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información del gestor asociado a la id pasada. En caso no de existir el gestor, se devuelve un objeto JSON vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/gestores>/1  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "correo": "gestor1@mail.com",          "usuario": "gestor1",          "id": 1      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener gestor por nombre de usuario

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener un gestor a partir de su nombre de usuario |
| **Nivel de autorización** | 3 |
| **Ruta** | /gestores/:usuario |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información del gestor asociado a la id pasada. En caso no de existir el gestor, se devuelve un objeto JSON vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/gestores>/gestor1  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "correo": "gestor1@mail.com",          "usuario": "gestor1",          "id": 1      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener clientes

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener la lista de todos los clientes. |
| **Nivel de autorización** | 3 |
| **Ruta** | /clientes/ |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (array): posee un array de objetos JSON con la información de cada uno de los clientes. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/clientes>/  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | {      "msg": "",      "data": [          {              "correo": "cliente1@mail.com",              "id\_gestor": 1,              "usuario": "cliente1",              "id": 1,              "saldo": 200          },          {              "correo": "cliente2@mail.com",              "id\_gestor": 2,              "usuario": "cliente2",              "id": 2,              "saldo": 300          },          {              "correo": "cliente3@mail.com",              "id\_gestor": 3,              "usuario": "cliente3",              "id": 3,              "saldo": 200          }      ],      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener cliente por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener un cliente a partir de su id |
| **Nivel de autorización** | 3 |
| **Ruta** | /clientes/:id |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información para el cliente asociado a la id pasada. En caso no de existir el cliente, se devuelve un objeto JSON vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/clientes/1>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "correo": "cliente1@mail.com",          "id\_gestor": 1,          "usuario": "cliente1",          "id": 1,          "saldo": 200      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener cliente por nombre de usuario

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener un cliente a partir de su nombre de usuario |
| **Nivel de autorización** | 3 |
| **Ruta** | /clientes/:usuario |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información para el cliente cliente a la id pasada. En caso no de existir el cliente, se devuelve un objeto JSON vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/clientes>/cliente1  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "correo": "cliente1@mail.com",          "id\_gestor": 1,          "usuario": "cliente1",          "id": 1,          "saldo": 200      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener mensajes recibidos

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener la lista de todos los mensajes recibidos para el gestor autenticado. |
| **Nivel de autorización** | 3 |
| **Ruta** | /mensajes/recibidos |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (array): posee un array de objetos JSON con la información de todos los mensajes recibidos para el gestor autenticado. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/mensajes/recibidos>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | {      "msg": "",      "data": [          {              "texto": "adiós gestor1",              "fecha": "2020-01-16T00:00",              "id\_origen": 3,              "id": 3          },          {              "texto": "adiós gestor1",              "fecha": "2020-01-16T00:00",              "id\_origen": 2,              "id": 4          }      ],      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener mensajes enviados

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener la lista de todos los mensajes enviados para el gestor autenticado. |
| **Nivel de autorización** | 3 |
| **Ruta** | /mensajes/enviados |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (array): posee un array de objetos JSON con la información de todos los mensajes enviados para el gestor autenticado. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/mensajes/enviados>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | {      "msg": "",      "data": [          {              "texto": "hola gestor2",              "fecha": "2020-01-15T00:00",              "id\_destino": 2,              "id": 1          },          {              "texto": "hola gestor3",              "fecha": "2020-01-15T00:00",              "id\_destino": 3,              "id": 2          }      ],      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener mensaje por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener un mensaje a partir de su id para el gestor autenticado (siempre y cuando sea un mensaje recibido o enviado del propio gestor autenticado). |
| **Nivel de autorización** | 3 |
| **Ruta** | /mensajes/:id |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información del mensaje asociado a la id pasada. En caso no de existir el mensaje, se devuelve un objeto JSON vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/mensajes/1>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "texto": "hola gestor2",          "fecha": "2020-01-15T00:00",          "id\_origen": 1,          "id\_destino": 2,          "id": 1      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener transferencias

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener la lista de todas las transferencias. |
| **Nivel de autorización** | 3 |
| **Ruta** | /transferencias/ |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (array): posee un array de objetos JSON con la información de todas las transferencias de todos los clientes. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/transferencias>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | {      "msg": "",      "data": [          {              "fecha": "2020-01-28T00:00",              "concepto": "transferencia1",              "id\_beneficiario": 2,              "id": 1,              "importe": 200,              "id\_ordenante": 1          },          {              "fecha": "2020-01-23T00:00",              "concepto": "transferencia2",              "id\_beneficiario": 1,              "id": 2,              "importe": 50,              "id\_ordenante": 2          }      ],      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener transferencia por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener una transferencia a partir de su id. |
| **Nivel de autorización** | 3 |
| **Ruta** | /transferencias/:id |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información de la transferencia asociada a la id pasada. En caso no de existir el mensaje, se devuelve un objeto JSON vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/transferencias/1>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "fecha": "2020-01-28T00:00",          "concepto": "transferencia1",          "id\_beneficiario": 2,          "id": 1,          "importe": 200,          "id\_ordenante": 1      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Agregar gestor

|  |  |
| --- | --- |
| **Breve descripción** | Permite agregar un nuevo gestor |
| **Nivel de autorización** | 3 |
| **Ruta** | /gestores/ |
| **Método HTTP** | POST |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** | * Objeto JSON compuesto por los siguientes campos:  1. usuario (string): nombre del nuevo gestor. 2. password (string): contraseña del nuevo gestor. 3. correo (string): correo electrónico del nuevo gestor. |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/gestores/>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30  Cuerpo de la petición:  {      "password": "gestor4",      "correo": "gestor4@mail.com",      "usuario": "gestor"  } |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. Datos enviados con formato incorrecto. 2. Falta algún campo o algún tipo es incorrecto. 3. Ya existe un gestor con el mismo nombre de usuario. 4. Ya existe un gestor con el mismo correo. 5. Privilegios insuficientes. 6. Error interno del servidor. |

## Agregar cliente

|  |  |
| --- | --- |
| **Breve descripción** | Permite agregar un nuevo cliente. |
| **Nivel de autorización** | 3 |
| **Ruta** | /clientes/ |
| **Método HTTP** | POST |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** | * Objeto JSON compuesto por los siguientes campos:  1. usuario (string): nombre del nuevo cliente. 2. id\_gestor (number): identificador del gestor asociado. 3. password (string): contraseña del nuevo cliente. 4. correo (string): correo electrónico del nuevo cliente. 5. saldo (number): saldo del nuevo cliente. |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/clientes/>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30  Cuerpo de la petición:  {      "password": "cliente4",      "correo": "cliente4@mail.com",      "id\_gestor": 1,      "usuario": "cliente4",      "saldo": 200  } |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. Datos enviados con formato incorrecto. 2. Falta algún campo o algún tipo es incorrecto. 3. Ya existe un cliente con el mismo nombre de usuario. 4. Ya existe un cliente con el mismo correo. 5. El gestor asociado no existe. 6. Privilegios insuficientes. 7. Error interno del servidor. |

## Enviar mensaje

|  |  |
| --- | --- |
| **Breve descripción** | Permite enviar un mensaje a otro gestor |
| **Nivel de autorización** | 3 |
| **Ruta** | /mensajes/ |
| **Método HTTP** | POST |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** | * Objeto JSON compuesto por los siguientes campos:  1. id\_destino (number): identificador del gestor destino. 2. text (string): texto del mensaje. |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/mensajes/>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30  Cuerpo de la petición:  {      "texto": "hola gestor2",      "id\_destino": 2  } |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. Datos enviados con formato incorrecto. 2. Falta algún campo o algún tipo es incorrecto. 3. Gestor destino no existe. 4. Privilegios insuficientes. 5. Error interno del servidor. |

## Eliminar gestor por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite eliminar un gestor a partir de su id |
| **Nivel de autorización** | 3 |
| **Ruta** | /gestores/:id |
| **Método HTTP** | DELETE |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/gestores>/1  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. El gestor no existe. 2. No se puede eliminar el gestor porque existen datos asociados (clientes y/o mensajes). 3. Privilegios insuficientes. 4. Error interno del servidor. |

## Eliminar cliente por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite eliminar un cliente a partir de su id. |
| **Nivel de autorización** | 3 |
| **Ruta** | /clientes/:id |
| **Método HTTP** | DELETE |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/clientes>/:id  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. El cliente no existe. 2. No se puede eliminar el cliente porque existen datos asociados (transferencias y/o préstamos). 3. Privilegios insuficientes. 4. Error interno del servidor. |

## Eliminar mensajes

|  |  |
| --- | --- |
| **Breve descripción** | Permite eliminar todos los mensajes enviados o recibidos por el gestor autenticado. |
| **Nivel de autorización** | 3 |
| **Ruta** | /mensajes/ |
| **Método HTTP** | DELETE |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/mensajes>/  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Eliminar mensaje por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite eliminar un mensaje enviado o recibido por el gestor autenticado a partir del id del mensaje. |
| **Nivel de autorización** | 3 |
| **Ruta** | /mensajes/:id |
| **Método HTTP** | DELETE |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/mensajes>/:id  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. El mensaje no existe. 2. Privilegios insuficientes. 3. Error interno del servidor. |

# Métodos para el cliente

## Obtener información

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener información del cliente autenticado. |
| **Nivel de autorización** | 2 |
| **Ruta** | /cliente/ |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información del cliente autenticado. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/cliente>/  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "correo": "cliente1@mail.com",          "id\_gestor": 1,          "usuario": "cliente1",          "id": 1,          "saldo": 200      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

* 1. **Realizar ingreso de dinero**

|  |  |
| --- | --- |
| **Breve descripción** | Permite realizar un ingreso de dinero para el cliente autenticado. |
| **Nivel de autorización** | 2 |
| **Ruta** | /cliente/ingreso/:cantidad |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: http://127.0.0.1:8085/cliente/ingreso/2500  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. Cantidad introducida incorrecta. 2. Privilegios insuficientes. 3. Error interno del servidor. |

## Obtener transferencias

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener la lista de todas las transferencias para el cliente autenticado. |
| **Nivel de autorización** | 2 |
| **Ruta** | /cliente/transferencias/ |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (array): posee un array de objetos JSON con la información de las transferencias realizadas por el cliente autenticado. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/cliente/transferencias>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | {      "msg": "",      "data": [          {              "fecha": "2020-01-28T00:00",              "concepto": "transferencia1",              "id\_beneficiario": 2,              "id": 1,              "importe": 200,              "id\_ordenante": 1          },          {              "fecha": "2020-01-23T00:00",              "concepto": "transferencia2",              "id\_beneficiario": 1,              "id": 2,              "importe": 50,              "id\_ordenante": 2          }      ],      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Obtener transferencia por id

|  |  |
| --- | --- |
| **Breve descripción** | Permite obtener una transferencia a partir de su id. |
| **Nivel de autorización** | 2 |
| **Ruta** | /cliente/transferencias/:id |
| **Método HTTP** | GET |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** |  |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): posee la información de la transferencia asociada a la id pasada. En caso de no de existir la transferencia, se devuelve un objeto JSON vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/cliente/transferencias/1>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30 |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {          "fecha": "2020-01-28T00:00",          "concepto": "transferencia1",          "id\_beneficiario": 2,          "id": 1,          "importe": 200,          "id\_ordenante": 1      },      "ok": true  } |
| **Posibles errores** | 1. Privilegios insuficientes. 2. Error interno del servidor. |

## Realizar transferencia

|  |  |
| --- | --- |
| **Breve descripción** | Permite realizar una transferencia de dinero a un cliente desde el cliente autenticado. |
| **Nivel de autorización** | 2 |
| **Ruta** | /cliente/transferencias/ |
| **Método HTTP** | POST |
| **Cabecera HTTP** | El token de autenticación es establecido en la cabecera HTTP Authentication siguiendo el método de autenticación de acceso básica:  Authorization: Basic <token> |
| **Cuerpo de la petición** | * Objeto JSON compuesto por los siguientes campos:  1. id\_beneficiario (number): identificador del cliente beneficiario 2. importe (number): cantidad de la transferencia. 3. concepto (string): concepto de la transferencia. |
| **Parámetros de la respuesta** | * Objeto JSON compuesto por los siguientes campos:  1. ok (boolean): toma el valor true cuando la petición se ha efectuado correctamente y false en caso de error. 2. msg (string): es una cadena vacía cuando la petición se ha efectuado correctamente y en caso de error contiene el mensaje de error asociado. 3. data (object): objeto vacío. |
| **Ejemplo de petición correcta** | URL: <http://127.0.0.1:8085/cliente/transferencias/>  Cabecera de la petición:  Authorization: Basic node012fxec1s005vf1e5uciv3khme30  Cuerpo de la petición:  {      "id\_beneficiario": 2,      "concepto": "transferencia1",      "importe": 100  } |
| **Ejemplo de respuesta correcta** | Respuesta:  {      "msg": "",      "data": {},      "ok": true  } |
| **Posibles errores** | 1. Datos enviados con formato incorrecto. 2. Falta algún campo o algún tipo es incorrecto. 3. El cliente beneficiario no existe. 4. El cliente ordenante y el cliente beneficiario no pueden ser el mismo. 5. Saldo insuficiente. 6. Privilegios insuficientes. 7. Error interno del servidor. |