

SmartMood - Pruebas de funcionamiento



Instituto Tecnológico de Estudios Superior de Zamora
Ingeniería en tecnologías de la información y comunicaciones
Alumno: Ricardo Díaz Márquez

Servicio Usuarios

Tipo de servicio: Entidad

Responsable: Ricardo Díaz

Operaciones expuestas:

1.- Crear usuario:

Actores: Usuarios, Administradores

Url: <http://SmartMood/Api/Usuarios>

Método HTTP: Post

Lógica de negocio: Permite crear usuarios con un ID que podrán crear etiquetas y entradas asociadas con dicho ID.

POST

/usuarios/ Crear usuario

Parameters

No parameters

Request body required

application/json

```
{  "nombre": "Ricky Diaz",  "contrasena": "Tato189",  "correo": "rickywatch180@gmail.com",  "edad": 23,  "fechaCreacion": "2025-05-13T22:45:27.446655"}  
```

Execute

Responses

Curl

```
curl -X 'POST' \  'http://127.0.0.1:8000/usuarios/' \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "nombre": "Ricky Diaz",  "contrasena": "Tato189",  "correo": "rickywatch180@gmail.com",  "edad": 23,  "fechaCreacion": "2025-05-13T22:45:27.446655"}  '
```

Request URL

http://127.0.0.1:8000/usuarios/

Server response

Code	Details
200	<div>Response body</div> <div><pre>{ "mensaje": "Usuario insertado con éxito" }</pre></div> <div>Download</div>

Response headers

```
_id: ObjectId('68241a49efd82cf71201c0c0')
nombre: "Ricky Diaz"
contrasena: "Tato189"
correo: "rickywatch180@gmail.com"
edad: 23
FechaCreacion: "2025-05-13T22:45:27.446655"
```

2.- Eliminar usuario

Actores: Usuarios, Administradores

Url: <http://SmartMood/Api/Usuarios/{idUsuario}>

Método HTTP: Delete

Lógica de negocio: Permite eliminar usuarios de la base de datos. Antes de ejecutarse, tiene que pasar por un proceso de validación que confirme que el usuario introdujo su contraseña correctamente o que es ejecutado por un administrador.

DELETE /usuarios/{idUsuario}/eliminar Eliminar usuario

Parameters

Cancel

Reset

Name	Description
idUsuario * required	68241a49efd82cf71201c0c0
string (path)	

Request body required

application/json

```
{
  "contrasena": "Tato189"
}
```

Responses

Curl

```
curl -X 'DELETE' \
  'http://127.0.0.1:8000/usuarios/68241a49efd82cf71201c0c0/eliminar' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "contrasena": "Tato189"
  }'
```

Request URL

```
http://127.0.0.1:8000/usuarios/68241a49efd82cf71201c0c0/eliminar
```

Server response

Code	Details
200	<div>Response body<div><pre>{ "mensaje": "Usuario eliminado con exito" }</pre></div></div>

3.- Cambiar contraseña

Actores: Usuarios

Url: <http://SmartMood/Api/Usuarios/{idusuario}>

Método HTTP: Put

Lógica de negocio: Permite cambiar la contraseña de un usuario. Antes de ejecutarse, tiene que pasar por un proceso de validación que confirme que el usuario introdujo su contraseña correctamente o que es ejecutado por un administrador.

PUT

/usuarios/{idUsuario}/cambiarcontrasena

Cambiarcontrasena

Parameters

Cancel

Name	Description
idUsuario ★ required	
string (path)	6822c62f0b4c54e3cd6c935d

Request body required

applic

```
{  "contrasenaAnterior": "Mamutt192",  "contrasenaNueva": "ZorroTecZamora2d"}
```

Responses

Curl

```
curl -X 'PUT' \
'http://127.0.0.1:8000/usuarios/6822c62f0b4c54e3cd6c935d/cambiarcontrasena' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "contrasenaAnterior": "Mamutt192",
  "contrasenaNueva": "ZorroTecZamora2d"
}'
```

Request URL

```
http://127.0.0.1:8000/usuarios/6822c62f0b4c54e3cd6c935d/cambiarcontrasena
```

Server response

Code	Details
200	<div>Response body<pre>{ "mensaje": "Contraseña actualizada con éxito."}</pre></div>

4.- Cambiar datos usuario

Actores: Usuarios

Url: <http://SmartMood/Api/Usuarios/{IdUsuario}/cambiardatos>

Método HTTP: Put

Los campos que contengan algo serán actualizados en el usuario con el id indicado. Los campos que no contengan nada o contengan "Null" serán dejados como están.

PUT **/usuarios/{idUsuario}/cambiardatos** Cambiardatos

Parameters

Name	Description
idUsuario * required string (path)	6822c62f0b4c54e3cd6c935d

Request body required

```
{
  "contrasena": "ZorroTecZamora20",
  "nombre": "Nuevo nombre",
  "correo": "nuevocorreo@gmail.com",
  "edad": 50
}
```

Responses

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/usuarios/6822c62f0b4c54e3cd6c935d/cambiardatos' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "contrasena": "ZorroTecZamora20",
    "nombre": "Nuevo nombre",
    "correo": "nuevocorreo@gmail.com",
    "edad": 50
  }'
```

Request URL

```
http://127.0.0.1:8000/usuarios/6822c62f0b4c54e3cd6c935d/cambiardatos
```

Server response

Code	Details
200	Response body <pre>{ "mensaje": "Datos cambiados con éxito." }</pre>

5.- Ver usuarios

Actores: Administradores

Url: <http://SmartMood/Api/Usuarios>

Método HTTP: Get

Lógica de negocio: Permite ver todos los usuarios de la base de datos.

The screenshot shows a REST client interface. At the top, there is a button labeled 'GET' and the URL '/usuarios/' with the text 'Consultar usuarios' next to it. Below this, there is a section titled 'Parameters' with the text 'No parameters'. A large blue button labeled 'Execute' is positioned below the parameters section. The response area shows a status code '200' and the label 'Response body'. The response body is a JSON object containing a success message and a list of three user objects.

```
{
  "mensaje": "Consulta exitosa. Listado de usuarios:",
  "usuarios": [
    {
      "nombre": "PepitoElChiquito",
      "contrasena": "Frutodrilo19",
      "correo": "Bluey1802@gmail.com",
      "edad": 23,
      "FechaCreacion": "2025-05-11T22:49:09.679070",
      "idUsuario": "682172c1ce9135701bd1f5ce"
    },
    {
      "nombre": "Mordecai",
      "contrasena": "RowRow23",
      "correo": "chicaSexy@gmail.com",
      "edad": 29,
      "FechaCreacion": "2025-05-12T20:47:49.507177",
      "idUsuario": "6822abd7a8a2f021d01eee22"
    },
    {
      "nombre": "Nuevo nombre",
      "contrasena": "ZorroTecZamora20",
      "correo": "nuevocorreo@gmail.com",
      "edad": 50,
      "FechaCreacion": "2025-05-12T23:09:04.685669",
      "idUsuario": "6822c62f0b4c54e3cd6c935d"
    }
  ]
}
```

Servicio Actividades

Tipo de servicio: Entidad

Responsable: Ricardo Díaz

Operaciones expuestas:

1.- Crear actividad:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades>

Método HTTP: Post

Lógica de negocio: Permite crear actividades para asignarlas a entradas de diario. Esta operación asume que fue realizada desde un end point donde el usuario inició sesión.

POST **/actividades/** Crear actividad

Parameters

No parameters

Request body required

```
{
  "idUsuario": "6822abd7a8a2f021d01eee22",
  "nombre": "Tejer",
  "descripcion": "Pasar hilo por otro hilo",
  "icono": "Tejedura"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/actividades/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "idUsuario": "6822abd7a8a2f021d01eee22",
    "nombre": "Tejer",
    "descripcion": "Pasar hilo por otro hilo",
    "icono": "Tejedura"
  }'
```

Request URL

`http://127.0.0.1:8000/actividades/`

Server response

Code	Details
200	<div>Response body</div> <pre>{ "mensaje": "Actividad insertada con exito" }</pre>

2.- Eliminar Actividad:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades/{ActividadId}/Eliminar>

Método HTTP: Delete

Lógica de negocio: Permite eliminar actividades del usuario

DELETE	/actividades/{idActividad}/eliminar	Eliminar actividad						
Parameters								
<table><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>idActividad * required</td><td></td></tr><tr><td>string (path)</td><td>682395c6c466a26450cc30d8</td></tr></tbody></table>			Name	Description	idActividad * required		string (path)	682395c6c466a26450cc30d8
Name	Description							
idActividad * required								
string (path)	682395c6c466a26450cc30d8							
Request body required								
<pre>{ "contrasena": "ZorroTecZamora20" }</pre>								

Responses					
Curl					
<pre>curl -X 'DELETE' \ 'http://127.0.0.1:8000/actividades/682395c6c466a26450cc30d8/eliminar' \ -H 'accept: application/json' \ -H 'Content-Type: application/json' \ -d '{ "contrasena": "ZorroTecZamora20" }'</pre>					
Request URL					
<pre>http://127.0.0.1:8000/actividades/682395c6c466a26450cc30d8/eliminar</pre>					
Server response					
Code	Details				
200	<table><tr><td>Response body</td></tr><tr><td><pre>{ "mensaje": "Actividad eliminada con éxito." }</pre></td></tr><tr><td>Response headers</td></tr><tr><td><pre>content-length: 45 content-type: application/json date: Wed, 14 May 2025 04:36:26 GMT server: unicorn</pre></td></tr></table>	Response body	<pre>{ "mensaje": "Actividad eliminada con éxito." }</pre>	Response headers	<pre>content-length: 45 content-type: application/json date: Wed, 14 May 2025 04:36:26 GMT server: unicorn</pre>
Response body					
<pre>{ "mensaje": "Actividad eliminada con éxito." }</pre>					
Response headers					
<pre>content-length: 45 content-type: application/json date: Wed, 14 May 2025 04:36:26 GMT server: unicorn</pre>					

3.- Modificar actividad:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades/{ActividadId}/Modificar>

Método HTTP: Put

Lógica de negocio: Los campos que contengan algo serán actualizados en la actividad con el id indicado. Los campos que no contengan nada o contengan "Null" serán dejados como están.

PUT `/actividades/{idActividad}/cambiardatos` Cambiardatos

Parameters

Name	Description
idActividad * required string (path)	<input type="text" value="682395ebc466a26450cc30d9"/>

Request body required

```
{  
  "nombre": "Pasear",  
  "descripcion": "Salir a caminar",  
  "icono": "Personita"  
}
```

Responses

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/actividades/682395ebc466a26450cc30d9/cambiardatos' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "nombre": "Pasear",
    "descripcion": "Salir a caminar",
    "icono": "Personita"
  }'
```

Request URL

<http://127.0.0.1:8000/actividades/682395ebc466a26450cc30d9/cambiardatos>

Server response

Code	Details
------	---------

200	
-----	--

Response body

```
{
  "mensaje": "Datos cambiados con éxito."
}
```

Response headers

4.- Ver actividades del usuario:

Actores: Usuarios

Url: <http://SmartMood/Api/Actividades/{Id usuario}>

Método HTTP: Get

Lógica de negocio: Permite ver las etiquetas del usuario.

Name	Description
idUsuario * required string (path)	<input type="text" value="6822c62f0b4c54e3cd6c935d"/>

Execute

Request URL

`http://127.0.0.1:8000/actividades/6822c62f0b4c54e3cd6c935d`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "mensaje": "Consulta exitosa. Listado de actividades:", "actividades": [{ "idUsuario": "6822c62f0b4c54e3cd6c935d", "nombre": "Pasear", "descripcion": "Salir a caminar", "icono": "Personita", "idActividad": "682395ebc466a26450cc30d9" }, { "idUsuario": "6822c62f0b4c54e3cd6c935d", "nombre": "Desayunar", "descripcion": "Huevos y jamón", "icono": "Huevo", "idActividad": "6823960cc466a26450cc30da" }] }</pre>

5.- Ver todas las actividades

Actores: Desarrollador

Url: <http://SmartMood/Api/Actividades>

Método HTTP: Get

Lógica de negocio: Permite ver todas las actividades.

GET	/actividades/ Consultaractividades
<p>Parameters</p> <hr/>	
<p>No parameters</p>	

200

Response body

```
{
  "mensaje": "Consulta exitosa. Listado de actividades:",
  "actividades": [
    {
      "idUser": "6822c62f0b4c54e3cd6c935d",
      "nombre": "Pasear",
      "descripcion": "Salir a caminar",
      "icono": "Personita",
      "idActividad": "682395ebc466a26450cc30d9"
    },
    {
      "idUser": "6822c62f0b4c54e3cd6c935d",
      "nombre": "Desayunar",
      "descripcion": "Huevos y jamón",
      "icono": "Huevo",
      "idActividad": "6823960cc466a26450cc30da"
    },
    {
      "idUser": "6822abd7a8a2f021d01eee22",
      "nombre": "Tejer",
      "descripcion": "Pasar hilo por otro hilo",
      "icono": "Tejedura",
      "idActividad": "68241c8fef82cf71201c0c1"
    }
  ]
}
```

Servicio Emociones

Tipo de servicio: Entidad

Responsable: Ricardo Díaz

Operaciones expuestas:

1.- Ver todas las emociones

Actores: Desarrollador, Usuarios

Url: <http://SmartMood/Api/Emociones>

Método HTTP: Get

Lógica de negocio: Permite ver todas las emociones.

Emociones

GET

/emociones/ Consultaremociones

Parameters

No parameters

200

Response body

```
{
  "mensaje": "Consulta exitosa. Listado de emociones:",
  "emociones": [
    {
      "nombre": "Feliz",
      "descripcion": "Default",
      "color": "Amarillo",
      "idEmocion": "68240f703f6830992c99b105"
    },
    {
      "nombre": "Triste",
      "descripcion": "Default",
      "color": "Azul",
      "idEmocion": "68240f703f6830992c99b106"
    },
    {
      "nombre": "Enojado",
      "descripcion": "Default",
      "color": "Rojo",
      "idEmocion": "68240f703f6830992c99b107"
    },
    {
      "nombre": "Incomodo",
      "descripcion": "Default",
      "color": "Rosa",
      "idEmocion": "68240f703f6830992c99b108"
    }
  ]
}
```

Servicio Entradas (Pendiente)

Tipo de servicio: Entidad

Responsable: Ricardo Díaz

Operaciones expuestas:

1.- Crear entrada:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas>

Método HTTP: Post

Lógica de negocio: Permite crear entradas asociadas con el ID de un usuario.

2.- Eliminar entrada:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{EntradaId}/Eliminar>

Método HTTP: Delete

Lógica de negocio: Permite eliminar entradas de la base de datos.

3.- Modificar entrada:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{IdEntrada}/Modificar>

Método HTTP: Put

Lógica de negocio: Los campos que contengan algo serán actualizados en la entrada con el id indicado. Los campos que no contengan nada o contengan "Null" serán dejados como están.

4.- Ver entradas:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas>

Método HTTP: Get

Lógica de negocio: Permite ver todas las entradas.

5.- Ver entradas del usuario:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{IdUsuario}>

Método HTTP: Get

Lógica de negocio: Permite ver todas las entradas que pertenezcan a un usuario.

6.- Ver entrada por ID:

Actores: Usuarios

Url: <http://SmartMood/Api/Entradas/{IdEntrada}>

Método HTTP: Get

Lógica de negocio: Permite ver una entrada con el id especificado.