



Angelica Landazabal

## ► Documentación Técnica - SocialMeli

### Objetivo:

En el siguiente documento se detalla los aspectos a considerar para las pruebas del aplicativo desarrollado en la herramienta Spring Boot.

### Pre-requerimientos del sistema:

**Create Buyer:** Se debe de cargar los usuarios a través del siguiente endpoint.

#### Sign:

Method	SIGN
POST	/users/followed/create
Ejemplo: /users/followed/create	
PAYLOAD	{ "user_id": null, "user_name": "User Test" }
Response	Status Code 200 (todo OK) Status Code 400 (Bad Request)
More Info	Se retornará un JSON con los datos almacenados del usuario. En caso contrario, se retornarán datos nulos.

**Create Seller:** Se debe de cargar los usuarios a través del siguiente endpoint.

#### Sign:



Method	SIGN
POST	/users/followers/create
Ejemplo: /users/followers/create	
PAYLOAD	{ "user_id": null, "user_name": "User Test" }
Response	Status Code 200 (todo OK) Status Code 400 (Bad Request)
More Info	Se retornará un JSON con los datos almacenados del usuario. En caso contrario, se retornarán datos nulos.

Los id de los usuarios (compradores y vendedores) se incrementan automáticamente, por eso es necesario dejar la clave **"user\_id"** como **null**.

Los usuarios serán los únicos que tendrán un identificador exclusivo dentro de la aplicación. En el caso de los productos y post, estos podrán repetirse sin repercusiones.

## Requerimientos del sistema:

**US 0001:** Poder realizar la acción de "Follow" (seguir) a un determinado vendedor.

### Sign:

Method	SIGN
POST	/users/{user_id}/follow/{user_id_to_follow}
Ejemplo: /users/1235/follow/1569	
Response	Status Code 200 (todo OK) -> "Usuarios seguidos." Status Code 400 (Bad Request) -> "Los usuarios no pueden seguirse."

**More Info**

- Se retornará un mensaje en cualquiera de los dos casos.
- Si los usuarios ya se seguían, se retornará un **Status Code 400**.

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica al usuario actual
user_id_to_follow	int	Número que identifica al usuario a seguir

**US 0002:** Obtener el resultado de la cantidad de usuarios que siguen a un determinado vendedor

**Sign:**

Method	SIGN
GET	/users/{user_id}/followers/count
<b>Ejemplo:</b> /users/1569/followers/count	
Response	{ "user_id": 1569, "user_name": "vendedor1", "followers_count": 35 }
More Info:	En caso que el usuario no exista, se devolverán los valores como <b>null</b> .

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
user_name	String	Nombre de usuario asociado a la user_id



**US 0003:** Obtener un listado de todos los usuarios que siguen a un determinado vendedor (¿Quién me sigue?)

**Sign:**

Method	SIGN
GET	/users/{user_id}/followers/list
<b>Ejemplo:</b> /users/1569/followers/list	
<b>Response</b>	<pre>{   "user_id": 1569,   "user_name": "vendedor1",   "followers": [     {       "user_id": 4698,       "user_name": "usuario1"     },     {       "user_id": 1536,       "user_name": "usuario2"     },     {       "user_id": 2236,       "user_name": "usuario3"     }   ] }</pre>
<b>More Info</b>	<ul style="list-style-type: none"><li>• En caso de que el usuario no tenga <b>followers</b>, el valor será <b>null</b>.</li><li>• En caso de que el usuario no exista, no se retornará ningún valor.</li></ul>

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
user_name	String	Nombre de usuario asociado a la user_id



**US 0004:** Obtener un listado de todos los vendedores a los cuales sigue un determinado usuario (¿A quién sigo?)

**Sign:**

Method	SIGN
GET	/users/{user_id}/followed/list
Ejemplo: /users/4698/followed/list	
Response	<pre>{   "user_id": 4698,   "user_name": "usuario1",   "followed": [     {       "user_id": 1569,       "user_name": "vendedor1"     },     {       "user_id": 6932,       "user_name": "vendedor2"     },     {       "user_id": 6631,       "user_name": "vendedor3"     }   ] }</pre>
More Info	<ul style="list-style-type: none"><li>• En caso de que el usuario no tenga <b>followed</b>, el valor será <b>null</b>.</li><li>• En caso de que el usuario no exista, no se retornará ningún valor.</li></ul>

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
user_name	String	Nombre de usuario asociado a la user_id



**US 0005:** Dar de alta una nueva publicación (Solo se podrá publicar un producto por post.)

**Sign:**

Method	SIGN
POST	/products/post
PAYLOAD:	{ "user_id": 1235, "id_post": 18, "date": "29-04-2021", "detail": { "product_id": 1, "product_name": "Silla Gamer", "type": "Gamer", "brand": "Racer", "color": "Red & Black", "notes": "Special Edition" }, "category": 100, "price": 1500.50 }
RESPONSE	Status Code 200 (todo OK) -> "Post guardado con éxito." Status Code 400 (Bad Request) -> "Ha ocurrido un error al guardar el post."
More Info	Se retornará un mensaje en cualquiera de los dos casos.

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
id_post	int	Número identificadorio de cada una de las publicaciones
date	LocaleDate	Fecha de la publicación en formato dd-MM-yyyy
product_id	int	Número identificadorio de cada uno de los productos asociados a una publicación
product_name	String	Cadena de caracteres que representa el nombre de un



		producto
type	String	Cadena de caracteres que representa el tipo de un producto
brand	String	Cadena de caracteres que representa la marca de un producto
color	String	Cadena de caracteres que representa el color de un producto
notes	String	Cadena de caracteres para colocar notas u observaciones de un producto
category	int	Identificador que sirve para conocer la categoría a la que pertenece un producto. Por ejemplo: 100: Sillas, 58: Teclados
price	double	Precio del producto



**US 0006:** Obtener un listado de las publicaciones realizadas por los vendedores que un usuario sigue en las últimas dos semanas (para esto tener en cuenta ordenamiento por fecha, publicaciones más recientes primero).

**Sign:**

Method	SIGN
GET	/products/followed/{user_id}/list
Ejemplo: /products/followed/4698/list	
RESPONSE:	<pre>{   "user_id": 4698,   "posts": [{     "id_post": 32,     "date": "01-05-2021",     "detail": {       "product_id": 62,       "product_name": "Headset RGB Inalámbrico",       "type": "Gamer",       "brand": "Razer",       "color": "Green with RGB",       "notes": "Sin Batería"     },     "category": 120,     "price": 2800.50   }, {     "id_post": 18,     "date": "29-04-2021",     "detail": {       "product_id": 1,       "product_name": "Silla Gamer",       "type": "Gamer",       "brand": "Racer",       "color": "Red &amp; Black",       "notes": "Special Edition"     },     "category": 100,     "price": 15000.50   } }</pre>





	}
More Info	En caso de que el usuario no exista, se mostrarán los valores como nulos.

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario

**US 0007:** Poder realizar la acción de "Unfollow" (dejar de seguir) a un determinado vendedor.

**Sign:**

Method	SIGN
POST	/users/{user_id}/unfollow/{user_id_to_unfollow}
<b>Ejemplo:</b> /users/1569/unfollow/1235	
<b>RESPONSE</b>	Status Code 200 (todo OK) -> "Los usuarios se han dejado de seguir." Status Code 400 (Bad Request) -> "Ha ocurrido un error en la acción."
More Info	Se retornará un mensaje en cualquiera de los dos casos.

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica al usuario actual
user_id_to_unfollow	int	Número que identifica al usuario a dejar de seguir

**US 0008:** Ordenamiento alfabético ascendente y descendente**Sign:**

Method	SIGN
GET	<b>Ejemplos:</b> /users/{user_id}/followers/list?order=name_asc /users/{user_id}/followers/list?order=name_desc /users/{user_id}/followed/list?order=name_asc /users/{user_id}/followed/list?order=name_desc
Response	<pre>{   "user_id": 4698,   "user_name": "usuario1",   "followed": [     {       "user_id": 1569,       "user_name": "vendedor1"     },     {       "user_id": 6932,       "user_name": "vendedor2"     },     {       "user_id": 6631,       "user_name": "vendedor3"     }   ] }</pre>
More Info	<ul style="list-style-type: none"><li>• En caso de que el usuario no tenga <b>followed/followers</b>, el valor será <b>null</b>.</li><li>• En caso de que el usuario no exista, no se retornará ningún valor.</li></ul>

order	Description
name_asc	Alfabético ascendente.
name_desc	Alfabético descendente.

**US 0009:** Ordenamiento por fecha ascendente y descendente**Sign:**

Method	SIGN
GET	<b>Ejemplos:</b> /products/followed/{user_id}/list?order=date_asc /products/followed/{user_id}/list?order=date_desc
<b>RESPONSE:</b>	<pre>{   "user_id": 4698,   "posts": [{     "id_post": 32,     "date": "01-05-2021",     "detail": {       "product_id": 62,       "product_name": "Headset RGB Inalámbrico",       "type": "Gamer",       "brand": "Razer",       "color": "Green with RGB",       "notes": "Sin Batería"     },     "category": 120,     "price": 2800.50   }, {     "id_post": 18,     "date": "29-04-2021",     "detail": {       "product_id": 1,       "product_name": "Silla Gamer",       "type": "Gamer",       "brand": "Racer",       "color": "Red &amp; Black",       "notes": "Special Edition"     },     "category": 100,     "price": 15000.50   } ] }</pre>
<b>More Info</b>	En caso de que el usuario no exista, se mostrarán los valores como nulos.



order	Description
date_asc	Fecha ascendente (de más antigua a más nueva)
date_desc	Fecha descendente (de más nueva a más antigua)



## Extra Bonus

**US 0010:** Llevar a cabo la publicación de un nuevo producto en promoción (Solo se podrá publicar un producto por post.)

### Sign:

Method	SIGN
POST	/products/promo-post
PAYLOAD:	{ "user_id": 1569, "id_post": 18, "date": "29-04-2021", "detail": { "product_id": 1, "product_name": "Silla Gamer", "type": "Gamer", "brand": "Racer", "color": "Red & Black", "notes": "Special Edition" }, "category": "100", "price": 1500.5, "has_promo": true, "discount": 0.25 }
RESPONSE	Status Code 200 (todo OK) -> "Post guardado con éxito." Status Code 400 (Bad Request) -> "Ha ocurrido un error al guardar el post."
More Info	Se retornará un mensaje en cualquiera de los dos casos.

### Filtros/Parámetros:

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
id_post	int	Número identificadorio de cada una de las publicaciones
date	LocaleDate	Fecha de la publicación en formato <b>dd-MM-yyyy</b>
product_id	int	Número identificadorio de cada uno de los productos



		asociados a una publicación
product_name	String	Cadena de caracteres que representa el nombre de un producto
type	String	Cadena de caracteres que representa el tipo de un producto
brand	String	Cadena de caracteres que representa la marca de un producto
color	String	Cadena de caracteres que representa el color de un producto
notes	String	Cadena de caracteres para colocar notas u observaciones de un producto
category	int	Identificador que sirve para conocer la categoría a la que pertenece un producto. Por ejemplo: 100: Sillas, 58: Teclados
price	double	Precio del producto
has_promo	boolean	Campo true o false para determinar si un producto está en promoción o no
discount	double	En caso de que un producto estuviese en promoción ,establece el monto de descuento.



**US 0011:** Obtener la cantidad de productos en promoción de un determinado vendedor

**Sign:**

Method	SIGN
GET	/products/{userId}/promo-post/count
Response	{ "user_id" : 1569, "user_name": "vendedor1", "promo_products_count": 23 }
More Info:	En caso que el usuario no exista, se devolverán los valores como <b>null</b> .

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
user_name	String	Cadena de caracteres que representa el nombre del usuario
promo_products_count	int	Cantidad numérica de productos en promoción de un determinado usuario.



**US 0012:** Obtener un listado de todos los productos en promoción de un determinado vendedor

**Sign:**

Method	SIGN
GET	/products/{userId}/list
RESPONSE:	<pre>{   "user_id": 1569,   "user_name": "vendedor1",   "posts": [{     "id_post": 18,     "date": "29-04-2021",     "detail": {       "product_id": 1,       "product_name": "Silla Gamer",       "type": "Gamer",       "brand": "Racer",       "color": "Red &amp; Black",       "notes": "Special Edition"     },     "category": "100",     "price": 15000.5,     "has_promo": true,     "discount": 0.25   }, {     "id_post": 32,     "date": "01-05-2021",     "detail": {       "product_id": 62,       "product_name": "Headset RGB Inalámbrico",       "type": "Gamer",       "brand": "Razer",       "color": "Green with RGB",       "notes": "Sin Batería"     },     "category": "120",     "price": 2800.69,     "has_promo": true,     "discount": 0.1   } }] }</pre>



**More Info**

En caso de que el usuario no exista, se mostrarán los valores como nulos.

**Filtros/Parámetros:**

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
user_name	String	Cadena de caracteres que representa el nombre del usuario
id_post	int	Número identificador de cada una de las publicaciones
date	LocaleDate	Fecha de la publicación en formato <b>dd-MM-yyyy</b>
product_id	int	Número identificador de cada uno de los productos asociados a una publicación
product_name	String	Cadena de caracteres que representa el nombre de un producto
type	String	Cadena de caracteres que representa el tipo de un producto
brand	String	Cadena de caracteres que representa la marca de un producto
color	String	Cadena de caracteres que representa el color de un producto
notes	String	Cadena de caracteres para colocar notas u observaciones de un producto
category	int	Identificador que sirve para conocer la categoría a la que pertenece un producto. Por ejemplo: 100: Sillas, 58: Teclados
price	double	Precio del producto
has_promo	boolean	Campo true o false para determinar si un producto está en promoción o no
discount	double	En caso de que un producto estuviese en promoción, establece el monto de descuento.