



Programación Java. Sprint 1

▣ Equipo 2:

Requerimientos técnicos funcionales

A. Requerimientos Iniciales

US 0001: Poder realizar la acción de “Follow” (seguir) a un determinado vendedor

Sign:

Method	SIGN
POST	/users/{userId}/follow/{userIdToFollow}
Ejemplo: /users/123/follow/234	
Response	<p>Status Code 200 (OK):</p> <pre>{ "user_id": 3, "followed_user_id": 1 }</pre> <p>Status Code 400 (Bad Request):</p> <pre>{ "message": "User 5 is not a seller", "status_code": "BAD_REQUEST" }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 15 not found", "status_code": "NOT_FOUND" }</pre> <p>Status Code 400 (Bad Request):</p> <pre>{</pre>



	<pre>"message": "User 1 can't follow himself", "status_code": "BAD_REQUEST" }</pre> <p>Status Code 400 (Bad Request):</p> <pre>{ "message": "User 3 already follows user 1", "status_code": "BAD_REQUEST" }</pre>
--	---

Filtros/Parámetros:

Parámetros	Tipo	Descripción/Ejemplo
userId	int	Número que identifica al usuario actual
userIdToFollow	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/{userId}/followers/count
Ejemplo: /users/234/followers/count/	
Response	<p>Status Code 200 (OK):</p> <pre>{ "user_id": 234, "user_name": "vendedor1", "followers_count": 35 }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre>



```
Status Code 400 (Bad Request):  
{  
  "message": "User 4 is not a seller",  
  "status_code": "BAD_REQUEST"  
}
```

Filtros/Parámetros:

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



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

Sign:

Method	SIGN
GET	/users/{userId}/followers/list
Ejemplo: /users/234/followers/list	
Response	<p>Status Code 200 (OK):</p> <pre>{ "user_id": 234, "user_name": "vendedor1", "followers": [{ "user_id": 4698, "user_name": "usuario1" }, { "user_id": 1536, "user_name": "usuario2" }, { "user_id": 2236, "user_name": "usuario3" }] }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre> <p>Status Code 400 (Bad Request):</p> <pre>{ "message": "User 4 is not a seller", "status_code": "BAD_REQUEST" }</pre>

Filtros/Parámetros:



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



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

Sign:

Method	SIGN
GET	/users/{userId}/followed/list
Ejemplo: /users/4698/followed/list	
Response	<p>Status Code 200 (OK):</p> <pre>{ "user_id": 4698, "user_name": "usuario1", "followed": [{ "user_id": 234, "user_name": "vendedor1" }, { "user_id": 6932, "user_name": "vendedor2" }, { "user_id": 6631, "user_name": "vendedor3" }] }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre>

Filtros/Parámetros:

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

**US 0005:** Dar de alta una nueva publicación**Sign:**

Method	SIGN
POST	/products/post
PAYLOAD:	<pre>{ "user_id": 123, "date": "29-04-2021", "product": { "product_id": 1, "product_name": "Silla Gamer", "type": "Gamer", "brand": "Racer", "color": "Red & Black", "notes": "Special Edition" }, "category": 100, "price": 1500.50 }</pre>
RESPONSE	<p>Status Code 200 (OK):</p> <pre>{ "date": "29-04-2021", "category": 100, "price": 1500.5, "user_id": 1, "product": { "type": "Gamer", "brand": "Racer", "color": "Red & Black", "notes": "Special Edition", "product_id": 1, "product_name": "Silla Gamer" } }</pre> <p>Status Code 400 (Bad Request):</p> <pre>{ "message": "Product cannot be empty", "status_code": "BAD_REQUEST" }</pre>



	<p>Status Code 400 (Bad Request):</p> <pre>{ "message": "Invalid category", "status_code": "BAD_REQUEST" }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre>
--	--

Filtros/Parámetros:

Parámetros	Tipo	Descripción/Ejemplo
user_id	int	Número que identifica a cada usuario
date	LocalDate	Fecha de la publicación en formato dd-MM-yyyy
product_id	int	Número identificador de un producto asociado 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/{userId}/list
Ejemplo: /products/followed/4698/list	
RESPONSE:	<p>Status Code 200 (OK):</p> <pre>{ "user_id": 4698, "posts": [{ "user_id": 123, "post_id": 32, "date": "01-05-2021", "product": { "product_id": 62, "product_name": "Headset RGB Inalámbrico", "type": "Gamer", "brand": "Razer", "color": "Green with RGB", "notes": "Sin Batería" }, "category": 120, "price": 2800.69 }, { "user_id": 234, "post_id": 18, "date": "29-04-2021", "product": { "product_id": 1, "productName": "Silla Gamer", "type": "Gamer", "brand": "Racer", "color": "Red & Black", "notes": "Special Edition" }, "category": 100,</pre>



	<pre>"price": 15000.50 }] }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre>
--	--

Filtros/Parámetros:

Parámetros	Tipo	Descripción/Ejemplo
userId	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/{userId}/unfollow/{userIdToUnfollow}
Ejemplo: /users/234/unfollow/123	
RESPONSE	<p>Status Code 200 (OK):</p> <pre>{ "user_id": 5, "followed_user_id": 1 }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 5 does not follow user 1", "status_code": "NOT_FOUND" }</pre> <p>Status Code 404 (Not Found):</p> <pre>{</pre>



```
{  
  "message": "User 99 not found",  
  "status_code": "NOT_FOUND"  
}
```

Filtros/Parámetros:

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

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

Method	SIGN
GET	Ejemplos: /users/{UserID}/followed/list?order=name_asc /users/{UserID}/followed/list?order=name_desc
RESPONSE	<p>Status Code 200 (OK):</p> <pre>{ "followed": [{ "user_id": 9, "user_name": "Andres" }, { "user_id": 5, "user_name": "Hernan" }, { "user_id": 3, "user_name": "Jose" }, { "user_id": 7, "user_name": "Lucas" }], "user_id": 2, "user_name": "Juan" }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre> <p>Status Code 400 (Bad Request):</p> <pre>{ "message": "Cannot order by cualquier_verdura",</pre>



	<pre>"status_code": "BAD_REQUEST" }</pre>
--	---

GET	Ejemplos: /users/{UserID}/followers/list?order=name_asc /users/{UserID}/followers/list?order=name_desc
RESPONSE	<p>Status Code 200 (OK):</p> <pre>{ "followers": [{ "user_id": 5, "user_name": "Hernan" }, { "user_id": 3, "user_name": "Jose" }, { "user_id": 11, "user_name": "Luis" }, { "user_id": 4, "user_name": "Oscar" }], "user_id": 2, "user_name": "Juan" }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre> <p>Status Code 400 (Bad Request):</p> <pre>{ "message": "Cannot order by cualquier_verdura", "status_code": "BAD_REQUEST" }</pre>



	}
--	---

order	Description
name_asc	Alfabético ascendente.
name_desc	Alfabético descendente.

***Nota:** Este ordenamiento aplica solo para US-003 y US-004.

US 0009: Ordenamiento por fecha ascendente y descendente

Sign:

Method	SIGN
GET	Ejemplos: /products/followed/{userId}/list?order=date_asc /products/followed/{userId}/list?order=date_desc
RESPONSE	Status Code 200 (OK): { "posts": [{ "date": "25-09-2024", "product": { "type": "Gamer", "brand": "Racer", "color": "Red & Black", "notes": "Special Edition", "product_id": 1, "product_name": "Silla Gamer" }, "category": 0, "price": 1500.5, "user_id": 1, "post_id": 1 }, { "date": "29-09-2024",



```
"product": {
  "type": "Gamer",
  "brand": "Racer",
  "color": "Red & Black",
  "notes": "Special Edition",
  "product_id": 1,
  "product_name": "Silla Gamer"
},
"category": 0,
"price": 1500.5,
"user_id": 2,
"post_id": 2
},
{
  "date": "29-09-2024",
  "product": {
    "type": "Gamer",
    "brand": "Racer",
    "color": "Red & Black",
    "notes": "Special Edition",
    "product_id": 1,
    "product_name": "Silla Gamer"
  },
  "category": 0,
  "price": 1500.5,
  "user_id": 1,
  "post_id": 4
}
],
"user_id": 3
}
```

Status Code 404 (Not Found)

```
{
  "message": "User 99 not found",
  "status_code": "NOT_FOUND"
}
```

Status Code 400 (Bad Request):

```
{
  "message": "Cannot order by cualquier_verdura_v2",
  "status_code": "BAD_REQUEST"
}
```



	}
--	---

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

***Nota:** Este ordenamiento aplica solo para la US-0006

US 0010: Llevar a cabo la publicación de un nuevo producto en promoción

Sign:

Method	SIGN
POST	/products/promo-post
PAYLOAD:	<pre>{ "user_id": 234, "date": "29-04-2021", "product": { "product_id": 1, "product_name": "Silla Gamer", "type": "Gamer", "brand": "Racer", "color": "Red & Black", "notes": "Special Edition" }, "category": 100, "price": 1500.50, "has_promo": true, "discount": 0.25 }</pre>
Response	Status Code 200 (OK): <pre>{ "date": "01-04-2023", "category": 100, "price": 1500.5,</pre>



```
"discount": 0.25,  
"user_id": 1,  
"product": {  
  "type": "Gamer",  
  "brand": "Racer",  
  "color": "Red & Black",  
  "notes": "Special Edition",  
  "product_id": 1,  
  "product_name": "Silla Gamer"  
},  
"has_promo": true  
}
```

Status Code 404 (Not Found):

```
{  
  "message": "User 99 not found",  
  "status_code": "NOT_FOUND"  
}
```

Status Code 400 (Bad request):

```
{  
  "message": "Product cannot be empty",  
  "status_code": "BAD_REQUEST"  
}
```

Status Code 400 (Bad request):

```
{  
  "message": "Discount cannot be zero",  
  "status_code": "BAD_REQUEST"  
}
```

Status Code 400 (Bad request):

```
{  
  "message": "Date cannot be empty",  
  "status_code": "BAD_REQUEST"  
}
```

Filtros/Parámetros:

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



date	LocalDate	Fecha de la publicación en formato dd-MM-yyyy
product_id	int	Número identificador de un producto asociado 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/promo-post/count?user_id={userId}
Response	Status Code 200 (OK): { "user_id" : 234, "user_name": "vendedor1", "promo_products_count": 23 }



	<p>Status Code 400 (Bad request):</p> <pre>{ "message": "User 3 is not a seller", "status_code": "BAD_REQUEST" }</pre> <p>Status Code 404 (Not Found):</p> <pre>{ "message": "User 99 not found", "status_code": "NOT_FOUND" }</pre>
--	--

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.

B. Bonus

US 0012: Obtener un listado de top 10 vendedores según la cantidad de seguidores y desempatan por la cantidad de publicaciones.

Sign:

Method	SIGN
GET	/users/top-10
RESPONSE:	<p>Status Code 200 (OK):</p> <pre>[{ "user_id": 7, "user_name": "Lucas",</pre>



```
"followers_count": 11,
"posts_count": 1
},
{
  "user_id": 9,
  "user_name": "Andres",
  "followers_count": 9,
  "posts_count": 4
},
{
  "user_id": 5,
  "user_name": "Hernan",
  "followers_count": 8,
  "posts_count": 2
},
{
  "user_id": 2,
  "user_name": "Juan",
  "followers_count": 6,
  "posts_count": 11
},
{
  "user_id": 3,
  "user_name": "Jose",
  "followers_count": 6,
  "posts_count": 1
},
{
  "user_id": 4,
  "user_name": "Oscar",
  "followers_count": 5,
  "posts_count": 5
},
{
  "user_id": 6,
  "user_name": "Santiago",
  "followers_count": 4,
  "posts_count": 4
},
{
  "user_id": 1,
  "user_name": "Pepe",
```



```
"followers_count": 2,  
"posts_count": 6  
},  
{  
  "user_id": 8,  
  "user_name": "Carlos",  
  "followers_count": 1,  
  "posts_count": 4  
},  
{  
  "user_id": 11,  
  "user_name": "Luis",  
  "followers_count": 1,  
  "posts_count": 2  
}  
]
```