

No	Caso de Prueba	Datos de la prueba	Pasos	Resultado esperado	Resultado obtenido	Estatus	Enlace al defecto
1	Verificar que el usuario pueda añadir un producto existente a un kit.	Path params: id=6 <pre>{ "productsList": [{ "id": 3, "quantity": 3 }] }</pre>	1. El usuario selecciona el método POST y hace una solicitud a: URL + api/v1/kits/:id/products 2. El usuario envía una petición usando un ID de producto existente en los parámetros y el cuerpo de la solicitud. 3. El usuario añade productos existentes al kit. 4. El usuario obtiene un código de estado: 200 OK en la respuesta.	- Se añaden productos existentes al kit - El estado de la solicitud es 200 OK	- Se añaden productos existentes al kit - El estado de la solicitud es 200 OK	Passed	
2	Agregar 1 producto válido a un kit existente	<pre>{ "productsList": [{ "id": 1, "quantity": 1 }] }</pre>	1. El usuario selecciona el método POST y hace una solicitud a: /api/v1/kits/6/products. 2. El usuario envía en el cuerpo una lista con un producto existente (id=1, quantity=1). 3. El usuario añade el producto al kit. 4. El usuario obtiene un código de estado 200 OK en la respuesta.	- Se añaden productos existentes al kit - El estado de la solicitud es 200 OK	- Se añaden productos existentes al kit. - El estado de la solicitud es 200 OK.	Passed	
3	Agregar varios productos válidos (menos de 30)	<pre>{ "productsList": [{ "id": 1, "quantity": 1 }, { "id": 2, "quantity": 2 }] }</pre>	1. El usuario selecciona el método POST y hace una solicitud a: /api/v1/kits/6/products. 2. El usuario envía una lista con varios productos válidos (id=1, id=2). 3. El sistema agrega los productos al kit. 4. El usuario recibe 200 OK como respuesta.	- Se agregan varios productos válidos al kit - El estado de la solicitud es 200 OK	- Se añaden todos los productos existentes al kit. - El estado de la solicitud es 200 OK.	Passed	
4	Agregar 30 productos únicos	<pre>{ "productsList": [{ "id": 1, "quantity": 1 }, { "id": 2, "quantity": 1 }, ... { "id": 30, "quantity": 1 }] }</pre>	1. El usuario selecciona el método POST hacia /api/v1/kits/6/products. 2. El usuario envía una lista con 30 productos únicos. 3. El sistema agrega todos los productos al kit. 4. El usuario obtiene 200 OK.	- Se agregan 30 productos únicos correctamente - El estado de la solicitud es 200 OK	- Se muestra el mensaje de error: "message": "No más de 30 artículos por conjunto". - El estado de la solicitud es 400 Bad Request. - El límite de productos está mal definido (rechaza 30 en lugar de aceptar).	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-1
5	Intentar agregar 31 productos únicos	<pre>{ "productsList": [{ "id": 1, "quantity": 1 }, { "id": 2, "quantity": 1 }, ... { "id": 31, "quantity": 1 }] }</pre>	1. El usuario selecciona POST y envía 31 productos únicos. 2. El sistema valida el límite de productos. 3. El sistema rechaza la solicitud. 4. El usuario recibe 400 Bad Request con el mensaje: "No más de 30 artículos por conjunto".	- El sistema rechaza la solicitud al exceder el límite de 30 productos - El estado de la solicitud es 400 Bad Request - Se muestra el mensaje "No más de 30 artículos por conjunto"	- Se muestra el mensaje: "message": "No más de 30 artículos por conjunto". - El sistema rechaza correctamente la solicitud. - El estado de la solicitud es 400 Bad Request.	Passed	
6	Enviar un ID de producto inexistente	<pre>{ "productsList": [{ "id": 9999, "quantity": 1 }] }</pre>	1. El usuario selecciona POST a /api/v1/kits/9999/products. 2. El sistema busca el kit. 3. No encuentra el ID del kit. 4. El usuario recibe 404 Not Found	- El sistema detecta que el producto no existe - El estado de la solicitud es 400 Bad Request	- Se devuelve una estructura completa de kit con productos reales, aunque el ID enviado no existe. - El sistema no valida correctamente los IDs de productos. - El estado de la solicitud es 200 OK. - Se muestra un contenido inesperado en la respuesta (datos de otros productos).	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-2

7	Enviar un ID de kit inexistente	Path params: id=9999 { "productsList": [{ "id": 1, "quantity": 2 }] }	1. El usuario selecciona el método POST y hace una solicitud a /api/v1/kits/9999/products. 2. El usuario envía un cuerpo con productos válidos (id=1, quantity=2). 3. El sistema busca el kit correspondiente. 4. El sistema no encuentra el ID del kit. 5. El usuario recibe un código de estado 404 Not Found.	- El sistema no encuentra el kit especificado - El estado de la solicitud es 404 Not Found	- El sistema valida la existencia del kit. - El sistema rechaza la solicitud. - El estado de la solicitud es 404 Not Found.	Passed	
8	Enviar productsList vacío	Path params: id=6 { "productsList": [] }	1. El usuario selecciona POST y envía un productsList vacío. 2. El sistema valida el contenido del array. 3. El sistema rechaza la solicitud. 4. El usuario obtiene 400 Bad Request.	- El sistema rechaza la solicitud porque productsList está vacío - El estado de la solicitud es 400 Bad Request	- El sistema valida el contenido del campo productsList. - El sistema rechaza la solicitud. - El estado de la solicitud es 400 Bad Request.	Passed	
9	Enviar productsList con formato incorrecto	Path params: id=6 { "productsList": { "id": 1, "quantity": 2 } }	1. El usuario selecciona POST y envía productsList como objeto en lugar de array. 2. El sistema valida la estructura del body. 3. El sistema detecta formato incorrecto. 4. El usuario recibe 400 Bad Request.	- El sistema rechaza la solicitud porque productsList no es un array - El estado de la solicitud es 400 Bad Request	- El sistema valida la estructura incorrecta del body. - El sistema rechaza la solicitud. - El estado de la solicitud es 400 Bad Request.	Passed	
10	Enviar quantity = 0	Path params: id=6 { "productsList": [{ "id": 1, "quantity": 0 }] }	1. El usuario hace una solicitud POST con quantity=0. 2. El sistema valida que quantity sea mayor a 0. 3. El sistema rechaza la solicitud. 4. El usuario recibe 400 Bad Request.	- El sistema rechaza la solicitud porque quantity es 0 - El estado de la solicitud es 400 Bad Request	- El sistema genera un error interno al procesar el valor. - El estado de la solicitud es 500 Internal Server Error. - El sistema no maneja correctamente el valor límite.	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-3
11	Enviar quantity negativo	Path params: id=6 { "productsList": [{ "id": 1, "quantity": -1 }] }	1. El usuario selecciona POST y envía un producto con quantity=-1. 2. El sistema valida el valor del campo. 3. El sistema rechaza la solicitud. 4. El usuario recibe 400 Bad Request.	- El sistema rechaza la solicitud porque quantity es negativo - El estado de la solicitud es 400 Bad Request	- El sistema acepta el valor negativo y procesa la solicitud. - El estado de la solicitud es 200 OK. - No se valida correctamente la cantidad nega	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-4
12	Enviar quantity como texto	Path params: id=6 { "productsList": [{ "id": 1, "quantity": "dos" }] }	1. El usuario selecciona POST y envía quantity como texto ("dos"). 2. El sistema valida el tipo de dato. 3. El sistema rechaza la solicitud. 4. El usuario obtiene 400 Bad Request.	- El sistema rechaza la solicitud porque quantity tiene un tipo de dato incorrecto - El estado de la solicitud es 400 Bad Request	- El sistema acepta el valor como texto sin mostrar error. - El estado de la solicitud es 200 OK. - No se valida el tipo de dato del campo quantity.	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-5
13	Enviar id de producto como texto	Path params: id=6 { "productsList": [{ "id": "uno", "quantity": 1 }] }	1. El usuario selecciona POST y envía id del producto como texto ("uno"). 2. El sistema valida el tipo de dato. 3. El sistema rechaza la solicitud. 4. El usuario recibe 400 Bad Request.	- El sistema rechaza la solicitud porque id tiene un tipo de dato incorrecto - El estado de la solicitud es 400 Bad Request	- El sistema genera un error interno al intentar convertir el valor de id. - Se muestra el mensaje: "invalid input syntax for type integer". - El estado de la solicitud es 500 Internal Server Error. - Falta validación de tipo para el campo id.	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-6

14	Enviar cuerpo sin campo productsList	{ "items": [{ "id": 1, "quantity": 2 }] }	<p>1. El usuario selecciona POST y envía un cuerpo sin el campo productsList.</p> <p>2. El sistema valida los campos requeridos.</p> <p>3. El sistema detecta la falta del campo obligatorio.</p> <p>4. El usuario recibe 400 Bad Request.</p>	<p>- El sistema rechaza la solicitud por falta del campo obligatorio productsList</p> <p>- El estado de la solicitud es 400 Bad Request</p>	<p>- El sistema valida la estructura del body.</p> <p>- El sistema rechaza la solicitud.</p> <p>- El estado de la solicitud es 400 Bad Request.</p>	Passed	
15	Enviar JSON malformado	Path params: id=6 { "productsList": [{ "id": 1, "quantity": 2 }] }	<p>1. El usuario selecciona POST y envía un JSON malformado.</p> <p>2. El sistema intenta leer el cuerpo de la solicitud.</p> <p>3. El sistema encuentra error de sintaxis.</p> <p>4. El usuario obtiene 400 Bad Request.</p>	<p>- El sistema rechaza la solicitud debido a un formato JSON malformado</p> <p>- El estado de la solicitud es 400 Bad Request</p>	<p>- El sistema no maneja correctamente la sintaxis inválida.</p> <p>- Devuelve una página HTML con el mensaje "Internal Server Error".</p> <p>- El estado de la solicitud es 500 Internal Server Error.</p>	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-7
16	Agregar producto duplicado con diferente cantidad	Path params: id=6 { "productsList": [{ "id": 1, "quantity": 1 }, { "id": 1, "quantity": 3 }] }	<p>1. El usuario selecciona POST y envía dos productos con el mismo id y diferente quantity.</p> <p>2. El sistema valida IDs duplicados.</p> <p>3. El sistema combina las cantidades sin contarlos como IDs únicos adicionales.</p> <p>4. El usuario recibe 200 OK.</p>	<p>- Se actualiza la cantidad del producto repetido sin duplicar su ID</p> <p>- El estado de la solicitud es 200 OK</p>	<p>- El sistema genera un error interno al procesar productos repetidos.</p> <p>- El estado de la solicitud es 500 Internal Server Error.</p> <p>- Falta validación para IDs duplicados dentro del array.</p>	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-8
17	Enviar ID de kit no numérico	Path params: id=abc { "productsList": [{ "id": 1, "quantity": 1 }] }	<p>1. El usuario selecciona POST a /api/v1/kits/abc/products.</p> <p>2. El sistema valida el formato del ID en la URL.</p> <p>3. El sistema rechaza la solicitud.</p> <p>4. El usuario recibe 400 Bad Request.</p>	<p>- El sistema rechaza la solicitud porque el ID del kit no tiene un formato válido</p> <p>- El estado de la solicitud es 400 Bad Request</p>	<p>- El sistema genera un error interno al procesar el parámetro : id.</p> <p>- El estado de la solicitud es 500 Internal Server Error.</p> <p>- Falta validación de tipo para el ID del kit.</p>	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-9
19	deliveryTime = 7 (debajo del rango permitido)	{ "deliveryTime": 7, "productsCount": 10, "productsWeight": 3 }	<p>1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery.</p> <p>2. El usuario envía un valor de deliveryTime menor al rango permitido (7).</p> <p>3. El sistema valida el parámetro deliveryTime.</p> <p>4. El sistema devuelve la respuesta.</p>	<p>- No es posible realizar la entrega fuera del horario permitido.</p> <p>- El estado de la solicitud es 200 OK</p>	<p>- No es posible realizar la entrega fuera del horario permitido.</p> <p>- El estado de la solicitud es 200 OK</p>	Passed	
20	deliveryTime = 8 (inicio del rango válido)	{ "deliveryTime": 8, "productsCount": 10, "productsWeight": 3 }	<p>1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery.</p> <p>2. El usuario envía un valor de deliveryTime válido (8).</p> <p>3. El sistema valida el parámetro deliveryTime.</p> <p>4. El sistema devuelve la respuesta.</p>	<p>- La entrega es posible dentro del horario permitido.</p> <p>- El estado de la solicitud es 200 OK</p>	<p>- La entrega es posible dentro del horario permitido.</p> <p>- El estado de la solicitud es 200 OK</p>	Passed	
21	deliveryTime = 23 (fuera del rango permitido)	{ "deliveryTime": 23, "productsCount": 10, "productsWeight": 3 }	<p>1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery.</p> <p>2. El usuario envía un valor de deliveryTime fuera del rango permitido (23).</p> <p>3. El sistema valida el parámetro deliveryTime.</p> <p>4. El sistema devuelve la respuesta.</p>	<p>- No es posible realizar la entrega fuera del horario permitido.</p> <p>- El estado de la solicitud es 200 OK</p>	<p>- No es posible realizar la entrega fuera del horario permitido.</p> <p>- El estado de la solicitud es 200 OK</p>	Passed	

22	deliveryTime con texto	{ "deliveryTime": "texto", "productsCount": 10, "productsWeight": 3 }	1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery. 2. El usuario envía un valor no numérico para deliveryTime. 3. El sistema valida el formato del parámetro. 4. El sistema devuelve la respuesta.	- El sistema rechaza la solicitud por formato incorrecto. - El estado de la solicitud es 400 Bad Request	- El sistema acepta el valor no numérico en deliveryTime. - El estado de la solicitud es 200 OK. - La validación del tipo de dato para deliveryTime no se está aplicando correctamente.	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-10?atlOrigin=eyJpIjoiNDkwYTMxNGM0MGRhNDMxMzhkOTIjMDk0ZWYyIjEzNGEiLCJwIjoiajI9
23	productsCount = 0 (mínimo permitido)	{ "deliveryTime": 10, "productsCount": 0, "productsWeight": 3 }	1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery. 2. El usuario envía un valor de productsCount igual a 0. 3. El sistema valida el parámetro productsCount. 4. El sistema devuelve la respuesta.	- No es posible procesar una entrega con cantidad de productos igual a cero. - El estado de la solicitud es 200 OK	- No es posible procesar una entrega con cantidad de productos igual a cero. - El estado de la solicitud es 200 OK	Passed	
24	Enviar productsCount = 1 (inicio del rango válido)	{ "deliveryTime": 10, "productsCount": 15, "productsWeight": 3 }	1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery. 2. El usuario envía un valor de productsCount válido (1). 3. El sistema valida el parámetro productsCount. 4. El sistema devuelve la respuesta.	- La entrega es posible con cantidad mínima válida. - El estado de la solicitud es 200 OK	- La entrega es posible con cantidad mínima válida. - El estado de la solicitud es 200 OK	Passed	
25	productsCount = 16 (fuera del rango permitido)	{ "deliveryTime": 10, "productsCount": 16, "productsWeight": 3 }	1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery. 2. El usuario envía un valor de productsCount mayor al rango permitido (16). 3. El sistema valida el parámetro productsCount. 4. El sistema devuelve la respuesta.	- No es posible procesar una entrega con esa cantidad de productos. - El estado de la solicitud es 200 OK	- No es posible procesar una entrega con esa cantidad de productos. - El estado de la solicitud es 200 OK	Passed	
26	productsCount con símbolo	{ "deliveryTime": 10, "productsCount": "@", "productsWeight": 3 }	1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery. 2. El usuario envía en el body un valor no numérico para productsCount ("@"). 3. El sistema valida el tipo de dato del parámetro productsCount. 4. El sistema rechaza la solicitud.	- El sistema rechaza la solicitud por formato incorrecto en productsCount. - El estado de la solicitud es 400 Bad Request.	- El sistema acepta el valor no numérico en productsWeight. - El estado de la solicitud es 200 OK. - No se genera el error esperado al validar el tipo de dato de productsWeight.	Failed	https://jero2007ortega-1761589531416.atlassian.net/browse/UG-11?atlOrigin=eyJpIjoiNGU1MTk4NmE3NzVjNDIhNTNmMzk2YWlyYTUzNzA5MjIjMDk0ZWYyIjEzNGEiLCJwIjoiajI9
27	productsWeight = 0 (mínimo permitido)	{ "deliveryTime": 10, "productsCount": 10, "productsWeight": 0 }	1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery. 2. El usuario envía un valor de productsWeight igual a 0. 3. El sistema valida el parámetro productsWeight. 4. El sistema devuelve la respuesta.	- La entrega es posible con peso total de 0 kg. - El estado de la solicitud es 200 OK	- La entrega es posible con peso total de 0 kg. - El estado de la solicitud es 200 OK	Passed	
28	productsWeight = 3.1 (inicio del segundo rango válido)	{ "deliveryTime": 10, "productsCount": 10, "productsWeight": 3.1 }	1. El usuario selecciona el método POST y hace una solicitud a: URL + /order-and-go/v1/delivery. 2. El usuario envía un valor de productsWeight igual a 3.1. 3. El sistema valida el parámetro productsWeight. 4. El sistema devuelve la respuesta.	- La entrega es posible con peso dentro del segundo rango permitido. - El estado de la solicitud es 200 OK	- La entrega es posible con peso dentro del segundo rango permitido. - El estado de la solicitud es 200 OK	Passed	

[illegible]