

ID	Título del caso de prueba	Pasos de la prueba	Datos de prueba (Cuerpo de la solicitud)	Resultado esperado	Resultado actual	Estado	Enlace al informe de errores
Requisito 1: /api/v1/kits/:id/products							
1	Agregar producto existente al kit	Enviar POST a /api/v1/kits/:id/products con un ID de kit en la URL y un ID existente de producto en el cuerpo de la solicitud.	{ "ID": 2, "productsList": [{"id": 1, "quantity": 2}] } }	200 OK	200 OK	APROBADO	
2	Agregar producto a kit con un ID de producto ya existente en el kit	Enviar POST a /api/v1/kits/:id/products con un ID de kit válido y producto ya agregado	{ "ID": 2, "productsList": [{"id": 1, "quantity": 1}] } }	200 OK	200 OK	APROBADO	
3	Agregar producto a kit con ID del Kit en formato string ("abc")	Enviar POST a /api/v1/kits/:id/products con un ID de kit="abc" y producto válido	{ "ID": "abc", "productsList": [{"id": 1, "quantity": 2}] } }	404 Not Found	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-1
4	Agregar producto a kit con ID del Kit negativo (-5)	Enviar POST a /api/v1/kits/:id/products con un ID de kit=-5 y producto válido	{ "ID": -5, "productsList": [{"id": 1, "quantity": 1}] } }	404 Not Found	501 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-2
5	Agregar producto a kit con ID del Kit vacío ""	Enviar POST a /api/v1/kits/:id/products con un ID de kit="" y producto válido	{ "ID": "", "productsList": [{"id": 1, "quantity": 1}] } }	404 Not Found	405 Method not Allowed	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-3
6	Agregar producto omitiendo el parámetro de ID del Kit aunque se especifique un ID válido de producto y cantidad válida	Enviar POST a /api/v1/kits/:id/products sin un ID de kit pero con ID de producto válido	{ "productsList": [{"id": 1, "quantity": 1}] } }	404 Not Found	406 Method not Allowed	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-4
7	Agregar 5 unidades de un producto válido a un kit	Enviar POST a /api/v1/kits/:id/products con un ID de kit válido y ID de producto existente con quantity=5	{ "ID": 2, "productsList": [{"id": 1, "quantity": 5}] } }	200 OK	200 OK	APROBADO	
8	Agregar Producto con Id de producto en formato string ("x123")	Enviar POST a /api/v1/kits/:id/products con un ID de kit válido y ID de Producto ="x123"	{ "ID": 2, "productsList": [{"id": "x123", "quantity": 1}] } }	400 Bad Request	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-5
9	Agregar producto a kit con ID de Producto negativo (-1)	Enviar POST a /api/v1/kits/:id/products con un ID de kit válido y ID de Producto = -1	{ "ID": 2, "productsList": [{"id": -1, "quantity": 1}] } }	400 Bad Request	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-6
10	Agregar producto a kit con un ID de Producto vacío ""	Enviar POST a /api/v1/kits/:id/products con un ID de kit válido y ID de Producto = ""	{ "ID": 2, "productsList": [{"id": "", "quantity": 2}] } }	400 Bad Request	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-7

11	Agregar producto a kit con ID válido omitiendo el parámetro de ID del Producto	Enviar POST a /api/v1/kits/:id/products con un ID de kit válido pero sin enviar ID de Producto.	{ "ID": 2, "productsList": [{ "quantity": 2] }	400 Bad Request	200 OK	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-8
12	Agregar producto con su cantidad en formato string ("cinco")	Enviar POST a /api/v1/kits/:id/products con cantidad ="cinco"	{ "ID": 2, "productsList": [{ "id": 1, "quantity": "cinco"]] }	400 Bad Request	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-9
13	Agregar producto con su cantidad en decimal (1.5)	Enviar POST a /api/v1/kits/:id/products con cantidad = 1.5	{ "ID": 2, "productsList": [{ "id": 1, "quantity": 1.5}] }	400 Bad Request	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-10
14	Agregar producto con una cantidad negativa (-2)	Enviar POST a /api/v1/kits/:id/products con cantidad = -2	{ "ID": 2, "productsList": [{ "id": 1, "quantity": -2}] }	400 Bad Request	200 OK	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-11
15	Agregar producto con su cantidad indicando un valor vacío ""	Enviar POST a /api/v1/kits/:id/products con cantidad = ""	{ "ID": 2, "productsList": [{ "id": 1, "quantity": ""}] }	400 Bad Request	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-12
16	Agregar producto a kit con ID válido, pero omitiendo el parámetro de Cantidad	Enviar POST a /api/v1/kits/:id/products con un ID de kit válido pero sin especificar el parámetro de cantidad.	{ "ID": 2, "productsList": [{ "id": 1}] }	400 Bad Request	500 Internal Server Error	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-13
17	Agregar producto con su cantidad indicando un valor igual a cero	Enviar POST a /api/v1/kits/:id/products con cantidad = 0	{ "ID": 2, "productsList": [{ "id": 1, "quantity": 0}] }	400 Bad Request	200 OK	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-14
18	Agregar 29 productos a un kit	1) Crear un nuevo kit sin productos. 2) Enviar POST a /api/v1/kits/:id/products con ese mismo ID de Kit y agregar 29 productos distintos	{ "ID": 7, "productsList": [{"id": 1, "quantity": 1}, {"id": 2, "quantity": 1}, {"id": 3, "quantity": 1}, {"id": 4, "quantity": 1}, {"id": 5, "quantity": 1}, {"id": 6, "quantity": 1}, {"id": 7, "quantity": 1}, {"id": 8, "quantity": 1}, {"id": 9, "quantity": 1}, {"id": 10, "quantity": 1}, {"id": 11, "quantity": 1}, {"id": 12, "quantity": 1}, {"id": 13, "quantity": 1}, {"id": 14, "quantity": 1}, {"id": 15, "quantity": 1}, {"id": 16, "quantity": 1}, {"id": 17, "quantity": 1}, {"id": 18, "quantity": 1}, {"id": 19, "quantity": 1}, {"id": 20, "quantity": 1}, {"id": 21, "quantity": 1}, {"id": 22, "quantity": 1}, {"id": 23, "quantity": 1}, {"id": 24, "quantity": 1}, {"id": 25, "quantity": 1}, {"id": 26, "quantity": 1}, {"id": 27, "quantity": 1}, {"id": 28, "quantity": 1}, {"id": 29, "quantity": 1}]	200 OK	200 OK	APROBADO	

19	<p>Agregar 30 productos a un kit</p>	<p>1) Crear un nuevo kit sin productos. 2) Enviar POST a /api/v1/kits/:id/products con ese mismo ID de Kit y agregar 30 productos distintos</p>	<pre>{ "ID": 7, "productsList": [{ "id": 1, "quantity": 1 }, { "id": 2, "quantity": 1 }, { "id": 3, "quantity": 1 }, { "id": 4, "quantity": 1 }, { "id": 5, "quantity": 1 }, { "id": 6, "quantity": 1 }, { "id": 7, "quantity": 1 }, { "id": 8, "quantity": 1 }, { "id": 9, "quantity": 1 }, { "id": 10, "quantity": 1 }, { "id": 11, "quantity": 1 }, { "id": 12, "quantity": 1 }, { "id": 13, "quantity": 1 }, { "id": 14, "quantity": 1 }, { "id": 15, "quantity": 1 }, { "id": 16, "quantity": 1 }, { "id": 17, "quantity": 1 }, { "id": 18, "quantity": 1 }, { "id": 19, "quantity": 1 }, { "id": 20, "quantity": 1 }, { "id": 21, "quantity": 1 }, { "id": 22, "quantity": 1 }, { "id": 23, "quantity": 1 }, { "id": 24, "quantity": 1 }, { "id": 25, "quantity": 1 }, { "id": 26, "quantity": 1 }, { "id": 27, "quantity": 1 }, { "id": 28, "quantity": 1 }, { "id": 29, "quantity": 1 }, { "id": 30, "quantity": 1 }] }</pre>	200 OK	200 OK	APROBADO	
20	<p>Agregar 31 productos a un kit</p>	<p>1) Crear un nuevo kit sin productos. 2) Enviar POST a /api/v1/kits/:id/products con ese mismo ID de Kit y agregar 31 productos distintos</p>	<pre>{ "ID": 7, "productsList": [{ "id": 1, "quantity": 1 }, { "id": 2, "quantity": 1 }, { "id": 3, "quantity": 1 }, { "id": 4, "quantity": 1 }, { "id": 5, "quantity": 1 }, { "id": 6, "quantity": 1 }, { "id": 7, "quantity": 1 }, { "id": 8, "quantity": 1 }, { "id": 9, "quantity": 1 }, { "id": 10, "quantity": 1 }, { "id": 11, "quantity": 1 }, { "id": 12, "quantity": 1 }, { "id": 13, "quantity": 1 }, { "id": 14, "quantity": 1 }, { "id": 15, "quantity": 1 }, { "id": 16, "quantity": 1 }, { "id": 17, "quantity": 1 }, { "id": 18, "quantity": 1 }, { "id": 19, "quantity": 1 }, { "id": 20, "quantity": 1 }, { "id": 21, "quantity": 1 }, { "id": 22, "quantity": 1 }, { "id": 23, "quantity": 1 }, { "id": 24, "quantity": 1 }, { "id": 25, "quantity": 1 }, { "id": 26, "quantity": 1 }, { "id": 27, "quantity": 1 }, { "id": 28, "quantity": 1 }, { "id": 29, "quantity": 1 }, { "id": 30, "quantity": 1 }, { "id": 31, "quantity": 1 }] }</pre>	400 Bad Request	400 Bad Request	APROBADO	
ID	Título del caso de prueba	Pasos de la prueba	Datos de prueba (Cuerpo de la solicitud)	Resultado esperado	Resultado actual	Estado	Enlace al informe de errores
Requisito 2: /order-and-go/v1/delivery							
1	<p>Validar isItPossibleToDeliver=false cuando deliveryTime=7 y productsCount/productsWeight en rango mínimo</p>	<p>Enviar POST a /order-and-go/v1/delivery con deliveryTime=7 y productsCount/productsWeight en rango mínimo.</p>	<pre>{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }</pre>	<pre>200 OK { "name": "Order and Go", "isItPossibleToDeliver": false, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }</pre>	<pre>{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 5 }</pre>	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG-1-33

2	Validar isItPossibleToDeliver=true cuando deliveryTime=8 y productsCount/productsWeight en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=8 y productsCount/productsWeight en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
3	Validar isItPossibleToDeliver=true cuando deliveryTime=9 y productsCount/productsWeight en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=9 y productsCount/productsWeight en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
4	Validar isItPossibleToDeliver=true cuando deliveryTime=21 y productsCount/productsWeight en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=21 y productsCount/productsWeight en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
5	Validar isItPossibleToDeliver=true cuando deliveryTime=22 y productsCount/productsWeight en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=22 y productsCount/productsWeight en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
6	Validar isItPossibleToDeliver=false cuando deliveryTime=23 y productsCount/productsWeight en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=23 y productsCount/productsWeight en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": false, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-34
7	Validar respuesta 400 Bad Request cuando deliveryTime es string y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime="texto" (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-15
8	Validar respuesta 400 Bad Request cuando deliveryTime es negativo y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=-1 (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 5 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-16

9	Validar respuesta 400 Bad Request cuando deliveryTime = 0 y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=0 (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 5 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-17
10	Validar respuesta 400 Bad Request cuando deliveryTime está vacío y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime="" (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 5 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-18
11	Validar respuesta 400 Bad Request cuando productsCount es string no numérico y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsCount="texto" (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-19
12	Validar respuesta 400 Bad Request cuando productsCount es negativo y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsCount=-1 (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-20
13	Validar respuesta 400 Bad Request cuando productsCount es 0 y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsCount=0 (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-21
14	Validar respuesta 400 Bad Request cuando productsCount está vacío y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsCount="" (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-22
15	Validar hostDeliveryCost=3 y clientDeliveryCost=0 cuando productsWeight=0 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10 y productsCount/productsWeight en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	

16	Validar hostDeliveryCost=3 y clientDeliveryCost=0 cuando productsWeight=0.1 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=0.1 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
17	Validar hostDeliveryCost=3 y clientDeliveryCost=0 cuando productsWeight=2.9 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=2.9 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
18	Validar hostDeliveryCost=3 y clientDeliveryCost=0 cuando productsWeight=3 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=3 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
19	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsWeight=3.1 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=3.1 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-23
20	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsWeight=3.2 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=3.2 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-24
21	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsWeight=5.9 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=5.9 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
22	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsWeight=6 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=6 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	

23	Validar hostDeliveryCost=5 y clientDeliveryCost=9 cuando productsWeight=6.1 y deliveryTime/productsCount en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=6.1 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 9 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-25
24	Validar respuesta 400 Bad Request cuando productsWeight es string y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight="texto" (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-26
25	Validar respuesta 400 Bad Request cuando productsWeight es negativo y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=-1 (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-27
26	Validar respuesta 400 Bad Request cuando productsWeight =0 y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight=0 (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-28
27	Validar respuesta 400 Bad Request cuando productsWeight está vacío y las otras variables en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsWeight="" (valor inválido) y las otras variables en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	400 Bad Request { "error": "Bad Request", "status": 400 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-29
28	Validar hostDeliveryCost=3 y clientDeliveryCost=0 cuando productsCount=0 y productsWeight=0.5 con deliveryTime en rango mínimo	Enviar POST a /order-and-go/v1/delivery con productsCount=0 y productsWeight=0.5 y deliveryTime/productsCount en rango mínimo.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
29	Validar hostDeliveryCost=3 y clientDeliveryCost=0 cuando productsCount=8 y productsWeight=3 con deliveryTime en rango mínimo	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=8 y productsWeight=3.0.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	

30	Validar hostDeliveryCost=3 y clientDeliveryCost=0 cuando productsCount=5 y productsWeight=2.0 con deliveryTime en rango	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=5 y productsWeight=2.0.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
31	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsCount=5 y productsWeight=3.1 con deliveryTime en rango	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=5 y productsWeight=3.1.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 3, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-30
32	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsCount=8 y productsWeight=6.0 con deliveryTime en rango	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=8 y productsWeight=6.0.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
33	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsCount=9 y productsWeight=2.0 con deliveryTime en rango	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=9 y productsWeight=2.0.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
34	Validar hostDeliveryCost=5 y clientDeliveryCost=0 cuando productsCount=15 y productsWeight=3.0 con deliveryTime en rango	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=15 y productsWeight=3.0.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	APROBADO	
35	Validar hostDeliveryCost=5 y clientDeliveryCost=9 cuando productsCount=5 y productsWeight=6.1 con deliveryTime en rango	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=5 y productsWeight=6.1.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 9 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 0 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-31
36	Validar hostDeliveryCost=5 y clientDeliveryCost=9 cuando productsCount=16 y productsWeight=2.0 con deliveryTime en rango	Enviar POST a /order-and-go/v1/delivery con deliveryTime=10, productsCount=16 y productsWeight=2.0.	{ "deliveryTime": 7, "productsCount": 0, "productsWeight": 0 }	200 OK { "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 9 }	{ "name": "Order and Go", "isItPossibleToDeliver": true, "hostDeliveryCost": 5, "toBeDeliveredTime": { "min": 20, "max": 25 }, "clientDeliveryCost": 5 }	NO APROBADO	https://urbanroutetest.atlassian.net/browse/UG1-32