

ID	Test Scenario	Test Case	Precondición	Pasos	Datos	RE
1	Api UsuarioCompra Recurso = /mocks/user	Crear usuarios	Que no exista el usuario	1.1.- Registrar usuario.	1.1 Información valida de usuario (usuario, region, email)	Mensaje "Usuario creado correctamente" Codigo Respuesta: 201
					1.2 Información vacia de usuario (usuario, region, email)	Mensaje "Debe ingresar todos los datos requeridos del usuario "
					1.3 Información invalida de usuario (usuario, region, email)	Mensaje "Debe ingresar datos validos del usuario " Codigo Respuesta: "400"
					1.4 Email de usuario existente	Mensaje "Ese email ya está registrado"
2		Mostrar informacion de usuarios	Que exista el usuario	Paso 1.1 2.1 - Consultar usuario	No se requiere datos especificos para listar todos los usuarios	Mostrar la infomación de todos los usuarios si existen (usuario, region, email, id) "Codigo Respuesta: 200" Si no hay usuarios Mnsj "No existen usuarios registrados" "Codigo Respuesta: ???"
					2.1 Identificacion valida de usuario (Id valido)	Mostrar la infomacion (usuario, region, email, id) de un usuario especifico. "Codigo Respuesta: 200"
					2.2 Identificacion invalida de usuario (Id invalido)	Mensaje "Usuario no registrado" Codigo Respuesta: "400"
3		Modificar informacion de usuario	Que exista el usuario	Paso 2.1 3.2.- Modificar la informacion	Dato 2.1 y Dato 1.1	Que modifique la información del usuario Codigo Respuesta: "200"
					Dato 2.1 y Dato 1.3	Mensaje "Debe ingresar datos validos del usuario" Codigo Respuesta: "400"
					Dato 2.1 y Dato 1.2	Mensaje "Debe ingresar todos los datos requeridos del usuario" Codigo Respuesta: "400"
4		Eliminar Usuario	Que exista el usuario	Paso 2.1 4.2.- Eliminar usuario	Dato 2.1	Que elimine el usuario seleccionado Codigo Respuesta: "204"
					Dato 2.2	Mensaje "Usuario no existente" Codigo Respuesta: "400"
5		Crear productos	Que no exista el producto	5.1.- Registrar producto.	5.1 Informacion valida del producto (nombre, descripcion, precio)	Mensaje "Producto creado correctamente" Codigo Respuesta: 201
					5.2 Informacion vacia del producto (nombre, descripcion, precio)	Mensaje "Debe ingresar todos los datos requeridos del producto"
					5.3 Informacion invalida del producto (nombre, descripcion, precio)	Mensaje "Debe ingresar datos validos del producto" Codigo Respuesta: "400"
					5.4 Nombre del producto existente	Mensaje "Ese nombre del producto ya existe"

6	Api UsuarioCompra Recurso = /mocks/product	Mostrar informacion del producto	Que exista el producto	Paso 5.1 6.1.- Consultar producto	No se requiere datos especificos para listar todos los productos	Mostrar la infomación de todos los productos si existen (nombre, descripcion, precio,id) "Codigo Respuesta: 200" Si no hay productos Mnsj "No existen productos registrados" "Codigo Respuesta: 200"
					6.1 Identificacion valida de producto (Id valido)	Mostrar la infomacion (nombre, descripcion, precio,id) de un producto especifico. "Codigo Respuesta: 200"
					6.2 Identificacion invalida de producto (Id invalido)	Mensaje "Producto no registrado" Codigo Respuesta: "404"
7		Modificar informacion de producto	Que exista el producto	Paso 6.1 7.2.- Modificar la informacion	Dato 6.1 y Dato 5.1	Que modifique la información del producto Codigo Respuesta: "200"
					Dato 6.1 y Dato 5.3	Mensaje "Debe ingresar datos validos del producto" Codigo Respuesta: "400"
					Dato 6.1 y Dato 5.2	Mensaje "Debe ingresar todos los datos requeridos del producto" Codigo Respuesta: "400"
					Dato 6.2 y Dato 5.1	Mensaje "El producto a modificar no existe" Codigo Respuesta: "404"
8		Eliminar Usuario	Que exista el producto	Paso 6.1 8.2.- Eliminar producto	Dato 6.1	Que elimine el producto seleccionado Codigo Respuesta: "204"
					Dato 6.2	Mensaje "Producto no existente" Codigo Respuesta: "400"
9		Crear purchase	Que exista usuario y que exista producto	9.1.- Registrar el purchase.	Dato 2.1 - Dato 5.1 fecha actual	Mensaje "Compra exitosa" Codigo Respuesta: 201
					Dato 2.1 y Datos de producto existente	Mensaje "Ya tiene ese producto asociado"
					Datos 2.2 y Dato 5.1	Mensaje "Usuario no existente"
					Datos 2.1 y Dato 5.3	Mensaje "Producto no existente"

10	Api UsuarioCompra Recurso = /mocks/purchase	Mostrar informacion del purchase	Que exista usuario, que exista producto y que exista un purchase	Paso 1.1 Paso 5.1 Paso 9.1 10.1.- Consultar purchase	No se requiere datos especificos para listar todos los purchase	Mostrar la infomación de todos los purchase si existen (id de usuario, usuario, producto, precio, fecha) "Codigo Respuesta: 200" Si no hay purchase Mnsj "No existen compras" "Codigo Respuesta: ???"
					Datos 2.1	Mostrar la infomacion (id de usuario, usuario, producto, precio, fecha) de la compra del usuario especifico si tiene compras. o Mensaje "El usuario no tiene compras asociadas"
					Datos 2.2	Mensaje "Usuario no existente" Codigo Respuesta: "400"
11		Modificar informacion de purchase	Que exista el purchase	Paso 6.1 11.1.- Modificar la informacion	Dato 2.1 - Dato 5.1 fecha actual	Que modifique la información de la compra Codigo Respuesta: "200"
					Dato 2.1 y Dato 5.3	Mensaje "Debe ingresar datos validos de la compra" Codigo Respuesta: "400"
					Dato 2.2 y Dato 5.3	Mensaje "Usuario no existente" Codigo Respuesta: "400"
12	Eliminar purchase	Que exista el purchase	Paso 6.1 12.1.- Eliminar producto	Identificación valida de purchase (Valor único)	Mensaje "La compra fue eliminada exitosamente" Codigo Respuesta: "204"	
				Identificación invalida de purchase	Mensaje "La compra no existe" Codigo Respuesta: "400"	
13	Api UsuarioCompra Recurso = /mocks/profile	Mostrar informacion de profile	Que exista usuario, Que exista producto y Que exista un purchase	Paso 1.1 Paso 5.1 Paso 9.1 13.1.- Consultar profile	Datos 2.1	Mostrar la infomacion (id de usuario, usuario, región,email, producto) de la compra del usuario especifico, si tiene compras. o Mensaje "El usuario no tiene compras asociadas"
					Datos 2.2	Mensaje "Usuario no existente" Codigo Respuesta: "400"
14	Integridad de Datos	Validar la integridad del Usuario	Que exista usuarios registrados	Paso 2.1	Dato 2.1	Que el tipo de dato de los Usuarios sea el correspondiente { "id": alfanumerico, "username": String, "region": String, "email": "String@dominio.com"}
		Validar la integridad del Producto	Que exista productos registrados	Paso 6.1	Dato 6.1	Que el tipo de dato de los Productos sea el correspondiente { "id": alfanumerico, "productname": string, "descripcion": string, "precio": float}
		Validar la integridad del Purchase	Que exista Compras registradas	Paso 10.1	Datos 2.1 - Datos 6.1	Que el tipo de dato de la Compra sea el correspondiente {"id": alfanumerico, "username": String, "productname": String, "precio": Float, "fecha": dd/mm/aaaa}
		Validar la integrida del Profile	Que exista usuario, Que exista producto y Que exista un purchase	Paso 13.1	Datos 2.1 y Datos 6.1	Que el tipo de dato del profile sea la correspondiente { "userid": alfanumerico, "username": String, "region": String, "email": string@dominio.com", "productname": ["string"]}
14	Formato de Respuesta	Validar de respuesta de todas las API {Usuario, Producto, Purchase, profile}	Que exista usuarios registrados	Hacer petición GET de cada API para mostrar el contenido de cada una	URL de cada API	Que el tipo de respuesta sea: Json

RO	Estatus	Criticidad	Estado	Version
	OK/Fail	Alta/Baja/Media	Ejecutado/ En proceso/En Espera	
* No muestra mensaje de creación exitosa (responde el Body) Código Respuesta: "201"				
* Si se envía la misma información de los parámetros requeridos, la duplica cambiando el ID				
* Si no se le pasa ningún parámetro, no muestra mensaje, respondiendo la información de todos los productos existentes				
* Cuando se le envía información inválida la crea, Código de respuesta 201				
* Si se le envía información vacía, crea un producto con información null				
* Si se le envía información de un producto existente duplica el registro cambiando el ID				

* Si hay productos, muestra la información de todos los Productos ("Codigo Respuesta: 200") * Si no hay productos muestra un arreglo vacío [ ] ("Codigo Respuesta: 200")			
* Muestra la información del Producto específico			
* Con caracteres especiales muestra información * Con alfanúmericos muestra el mensaje que no existe el producto (Código Respuesta 404) * Con cualquier carácter incluyendo "/", "\", muestra un html y Código de respuesta 404 (500?) * Con cualquier carácter incluyendo "i" no se conecta a la API			
* Modifica toda la información siempre y cuando se ingrese todos los campos requeridos (Código de respuesta: "202")	Ok		
* Si falta algún campo en específico lo modifica a NULL (Código de respuesta: "202") * Si los campos van vacíos, modifica la información a vacío (Código de respuesta: "202") * Si en el campo precio se le envía información incorrecta la modifica a Cero (Código de respuesta: "202")			
* Si el Id es válido y los campos van vacíos, modifica la información a vacío (Código de respuesta: "202")			
* Mensaje "El producto a modificar no existe" Codigo Respuesta: "404" * Cuando el Id es vacío, responde html (Código de respuesta: "405")	Ok Fail		
* Cuando se le ingresa un Id válido lo elimina, pero no muestra el mensaje, "Código de respuesta 204"			
* Mensaje: "El producto a eliminar no existe" Código de respuesta: 404 * Cuando se le envía un Id vacío responde un html Código de respuesta 405			

[illegible]