



## **Documentación tarea integradora #1**

**Algoritmos y programación 2**

**Grupo: 1**

**Kevin Alejandro Mera Castillo**

**Docente: Juan Manuel Reyes**

**Semestre 2020 – 2**

## ÍNDICE

#		Página
01	Requerimientos funcionales	3 - 4
02	Diagrama de clases	5 - 8
03	Diseño de casos de prueba	9 - 10

## 1. Requerimientos funcionales

**RF1:** Registrar un restaurante. Añade un restaurante a la lista de restaurantes, requiere de un nombre, un código NIT y el nombre del administrador. Si las condiciones se cumplen (que el NIT sea único) el restaurante es registrado satisfactoriamente.

**RF2:** Registrar productos. Añade un producto a la lista de productos, requiere de un nombre, código único, descripción, costo y el NIT del restaurante al cual va a ser añadido. Finalmente, el producto es agregado.

**RF3:** Registrar un cliente. Añade un cliente a la lista de clientes ordenadamente por su nombre y apellido de forma descendente, requiere de un nombre, apellido, número del documento, tipo de documento, teléfono y dirección. Finalmente, el cliente es agregado.

**RF4:** Registrar una orden. Añade una orden a la lista de órdenes, su código es autogenerado, requiere el número de documento del cliente quien hace el pedido y el NIT del restaurante al que solicita. Asimismo, se crea una lista de productos a pedir con su respectivo código y cantidad. Finalmente, la orden es agregada.

**RF5:** Actualizar la información de los restaurantes. Actualiza la información de los restaurantes al buscar por su NIT. Finalmente, se actualiza el atributo que el usuario decidió.

**RF6:** Actualizar la información de los productos. Actualiza la información de los productos al buscar por su código. Finalmente, se actualiza el atributo escogido por el usuario.

**RF7:** Actualizar la información de los clientes. Actualiza la información de los clientes al buscar por su número de identificación. Finalmente, se actualiza el atributo escogido por el usuario.

**RF8:** Actualizar la información de la orden. Actualiza la información de la orden como también su estado al buscar por su código. Finalmente, se actualiza el atributo escogido por el usuario.

**RF9:** Serializar la información del programa automáticamente. Guarda en archivos serializados la información del programa después de agregar o actualizar información.

**RF10:** Exportar información de órdenes. Exporta información de las ordenes en formato .csv ordenado por el nit del restaurante ascendente, documento del cliente descendente, fecha del pedido ascendente y código del producto ascendente. Se le solicita el usuario el separador a usar. Finalmente, se exporta la información de las ordenes.

**RF11:** Mostrar la información de los restaurantes. Despliega la lista con la información de los restaurantes ordenadamente de forma alfabéticamente ascendente.

**RF12:** Mostrar la información de los clientes. Despliega la lista con la información de los clientes tal cual como fueron añadidos ordenadamente.

**RF13:** Mostrar la información de los clientes ordenada por sus números telefónicos. Despliega la lista con la información de los clientes ordenadamente por sus números de teléfono descendente.

**RF14:** Ordenar la lista de clientes. Mantiene ordenada la lista de clientes de forma alfabéticamente descendente.

**RF15:** Mostrar la información de los productos. Despliega la lista con la información los productos.

**RF16:** Mostrar la información de las órdenes. Despliega la lista con la información de las órdenes.

**RF17:** Buscar cliente. Realiza la búsqueda binaria en la lista de clientes por su nombre completo. Finalmente, muestra el tiempo que se demoró realizando la búsqueda.

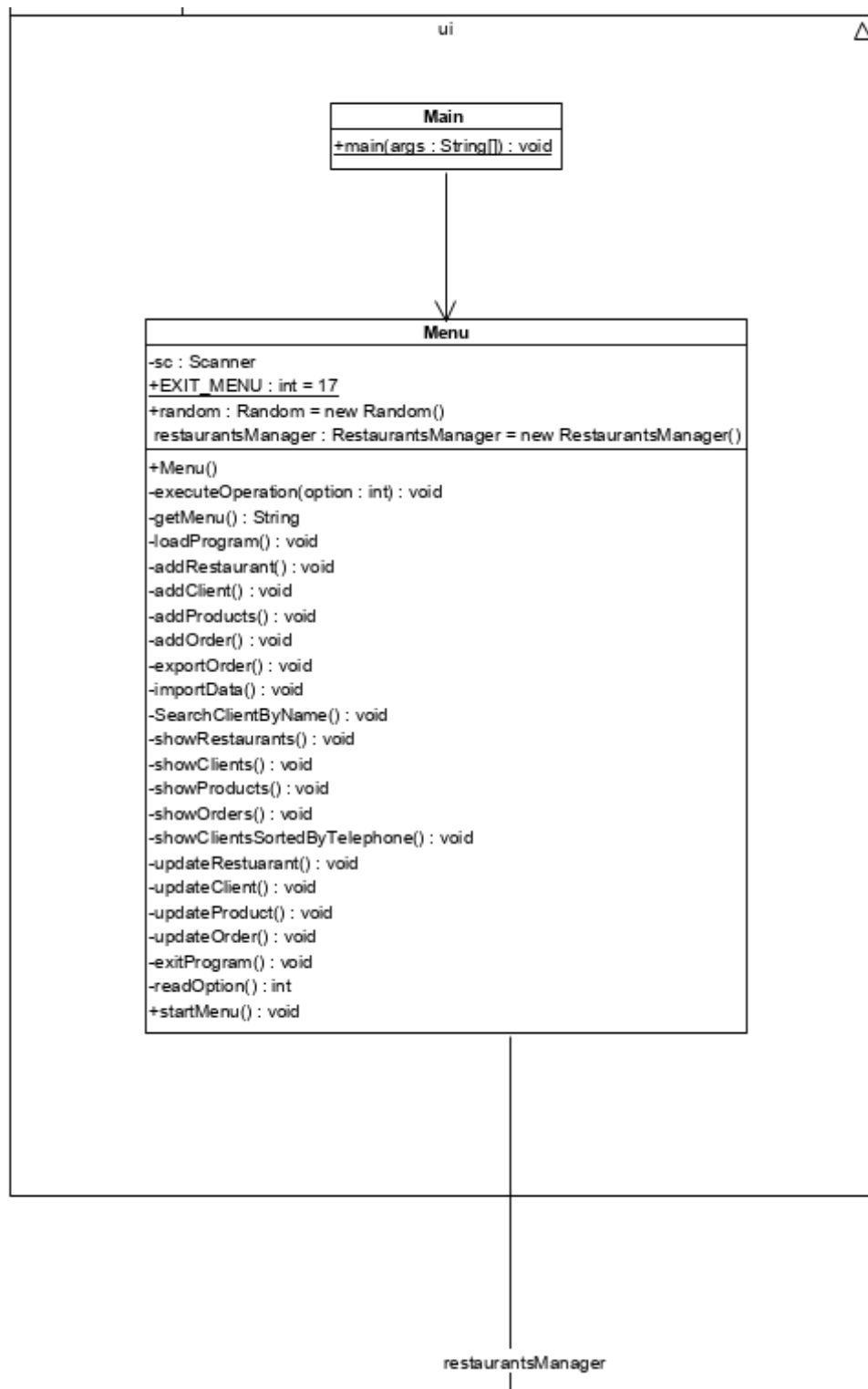
**RF18:** Importar información de restaurantes. Importa la información de restaurantes desde un archivo con formato .csv Finalmente, la información es cargada en el programa.

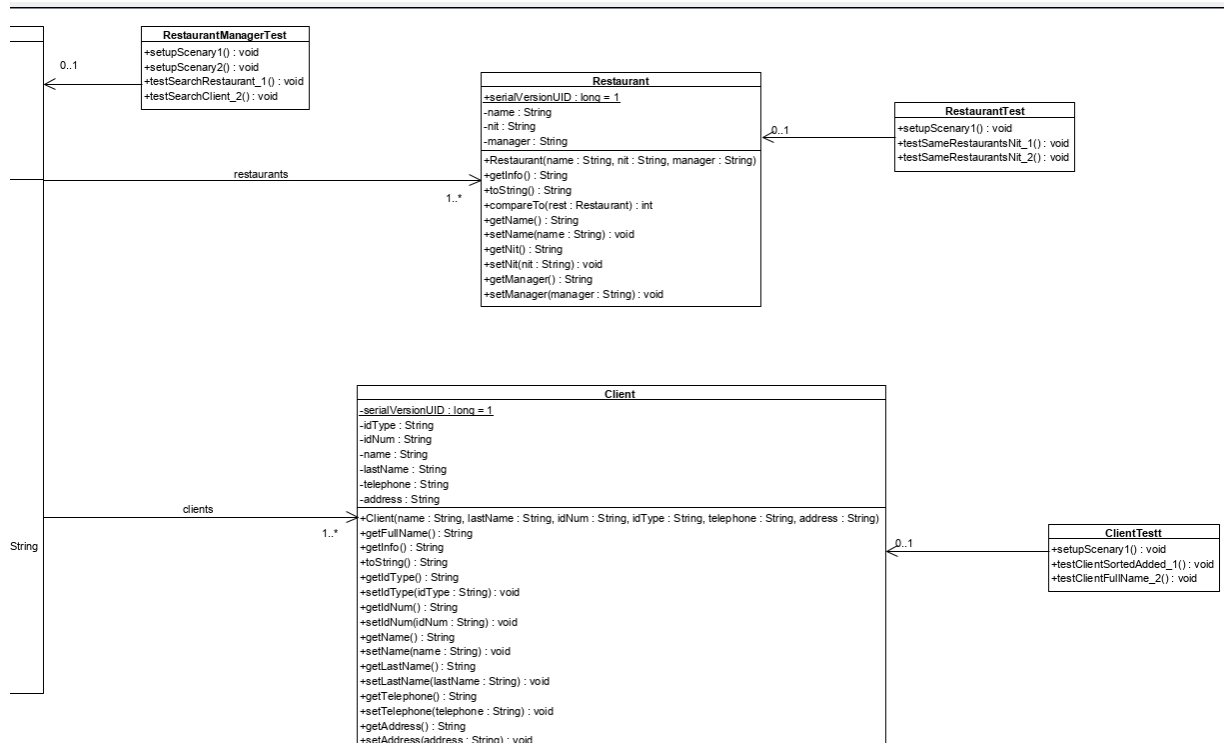
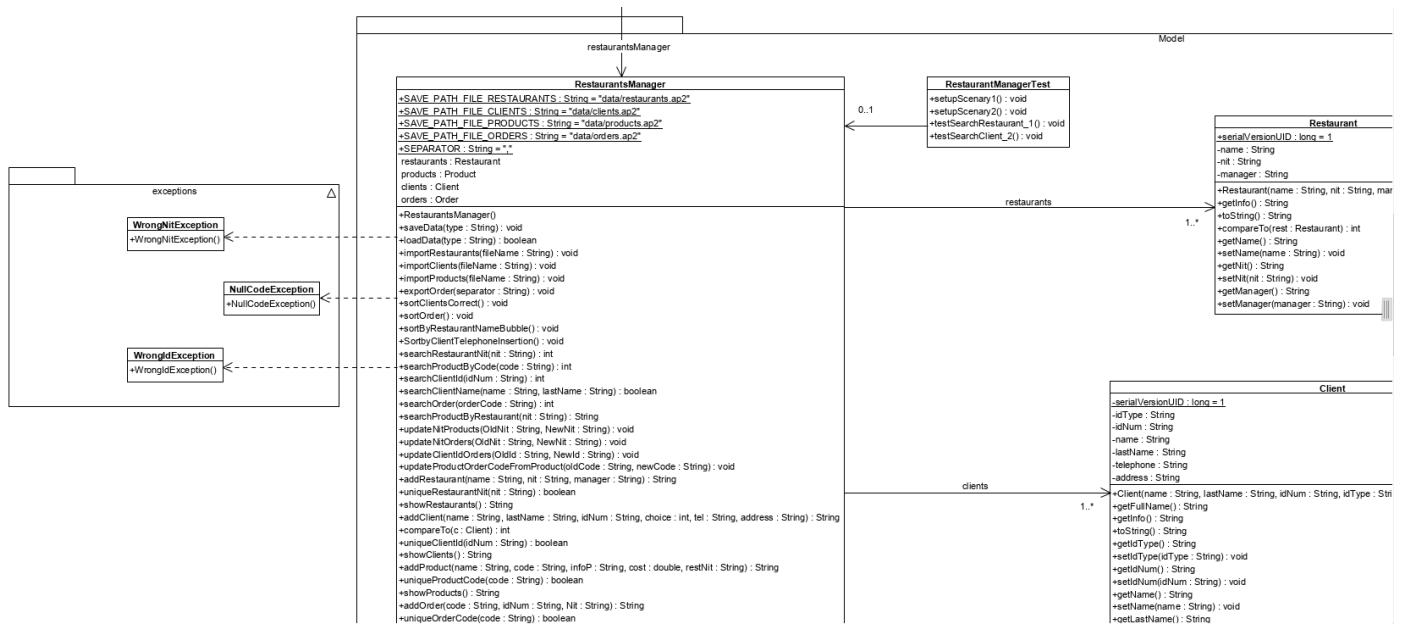
**RF19:** Importar información de clientes. Importa la información de clientes desde un archivo con formato .csv Finalmente, la información es cargada en el programa.

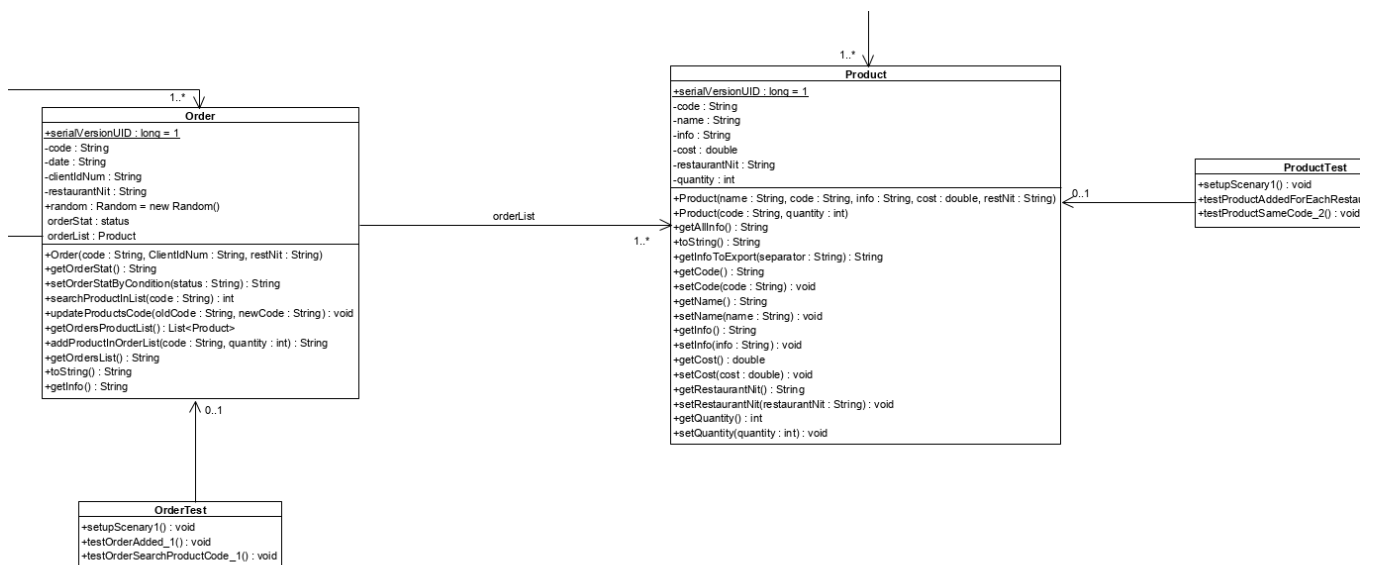
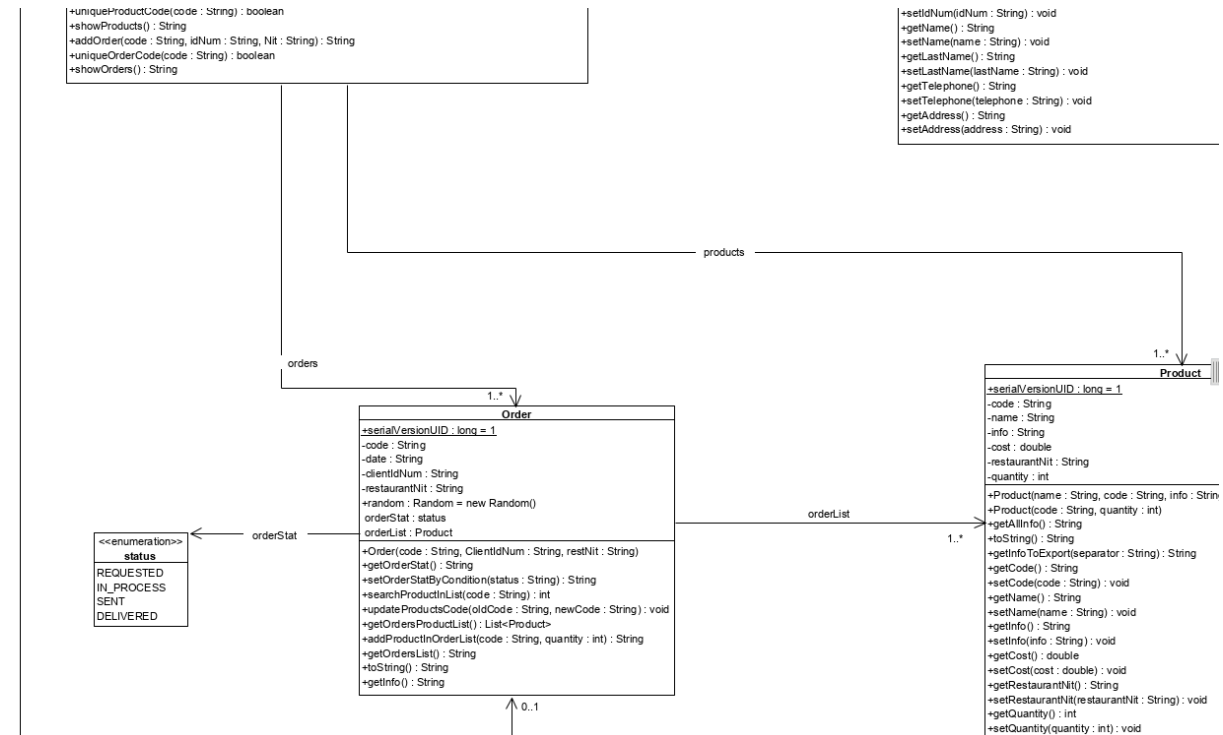
**RF20:** Importar información de productos. Importa la información de productos desde un archivo con formato .csv Finalmente, la información es cargada en el programa.

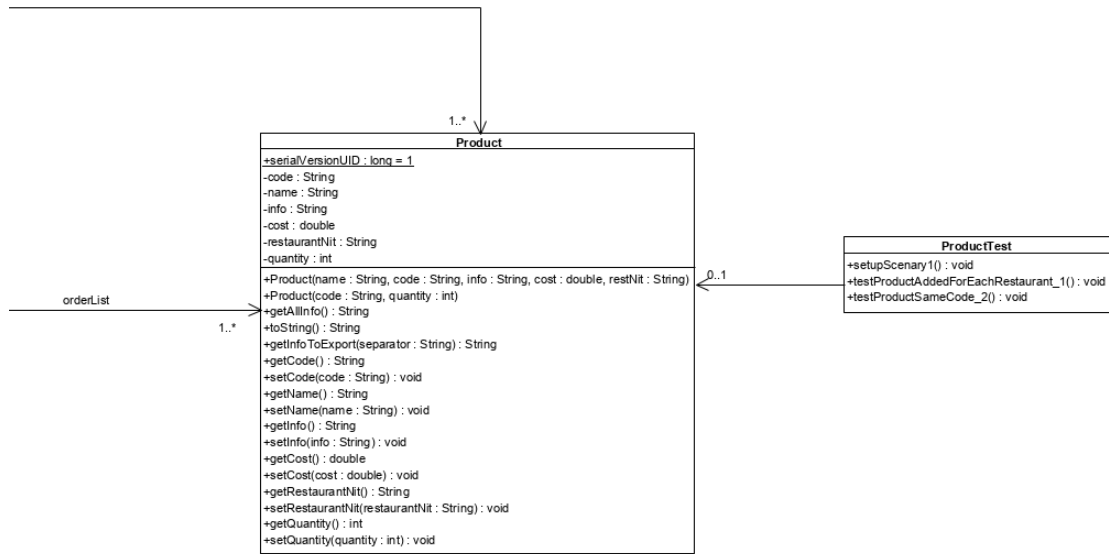
**RF21:** Importar información de órdenes. Importa la información de ordenes desde un archivo con formato .csv Finalmente, la información es cargada en el programa.

## 2. Diagrama de clase











### 3. Diseño de casos de prueba

DISEÑOS DE CASOS DE PRUEBA					
OBJETIVO DE LA PRUEBA: Verificar que un cliente se pueda añadir correctamente, ordenadamente y devuelva su número de documento					
CLASE	MÉTODO	ESCENARIO	VALORES DE ENTRADA		RESULTADO
Client	getIdNum	setupScenary_1	Cliente: Name = "Andrea" LastName = "Corrales" IdNum = "2154531341" idType = 1 Telephone = "31254867624" Address = "Cali"	True. Agrega correctamente y ordenadamente ya que al realizar la prueba en la clase Client y retornar el IdNum de la posición 1 del array corresponde a "2154531341"	
OBJETIVO DE LA PRUEBA: Verificar que un cliente se pueda añadir correctamente, ordenadamente devuelva su nombre completo					
Client	getFullName	setupScenary_1	Cliente: Name = "Camilo" LastName = "Ramirez" IdNum = "1548225645" idType = 4 Telephone = "3124865794" Address = "Cali"	True. Agrega correctamente y ordenadamente ya que al realizar la prueba en la clase Client y retornar el nombre completo de la posición 0 del array corresponde a "Camilo Ramirez"	
OBJETIVO DE LA PRUEBA: Verificar que una orden se añada correctamente y retorne el tamaño esperado					
Order	getOrders	setupScenary_1	Cliente Andrea: igual que el de la prueba anterior. Restaurante: Name = "Las delicias de Gallo" Nit = "12345" Manager = "Gallo" Restaurante: Name = "Las delicias de Colombia" Nit = "1234" Manager = "Camilo" Producto: name = "Pollo" Code = "4673" Infor = "Pollo por libra" Cost = 12000 RestNit = "1234" Producto: name = "Carne de res" Code = "4675" Infor = "Carne de res por libra" Cost = 8500 ResttNit = "1234" Orden: Client ID="2154531341" ResttNit = "12345" Orden: Client ID="2154531341" ResttNit = "12345" Producto en lista de orden: 2*code = 4673 quantity = 2, quantity = 3 Producto en lista de orden: 2*code = 2153 quantity = 2	True. Agrega las dos ordenes válidas y devuelve el tamaño 2 del array de ordenes	
OBJETIVO DE LA PRUEBA: Verificar que una orden se añada correctamente y busque la orden por su código					
Order	searchProductInList	setupScenary_1	Cliente Andrea: igual que el de la prueba anterior. Restaurante: Name = "Las delicias de Gallo" Nit = "12345" Manager = "Gallo" Restaurante: Name = "Las delicias de Colombia" Nit = "1234" Manager = "Camilo" Producto: name = "Pollo" Code = "4673" Infor = "Pollo por libra" Cost = 12000 RestNit = "1234" Producto: name = "Carne de res" Code = "4675" Infor = "Carne de res por libra" Cost = 8500 ResttNit = "1234" Orden: Client ID="2154531341" ResttNit = "12345" Orden: Client ID="2154531341" ResttNit = "12345" Producto en lista de orden: 2*code = 4673 quantity = 2, quantity = 3 Producto en lista de orden: 2*code = 2153 quantity = 2	True. Agrega las dos ordenes válidas y devuelve la posicion #1 de la orden buscada	
OBJETIVO DE LA PRUEBA: Verificar que el producto retorne el Nit del restaurante al que pertenece					
Product	getRestaurantNit	setupScenary_1	Restaurante: Name = "Las delicias de Gallo" Nit = "12345" Manager = "Gallo" Restaurante: Name = "Las delicias de Colombia" Nit = "1234" Manager = "Camilo" 2*(Producto: name = "Arepa" Code = "2153" Infor = "	True. El producto pertenece al restaurante registrado con NIT = "12345"	

Product	getRestaurantNit	setupScenary_1	Restaurante: Name = "Las delicias de Gallo" Nit = "12345" Manager = "Gallo" Restaurante: Name = "Las delicias de Colombia" Nit = "1234" Manager = "Camilo" 2*(Producto: name = "Arepa" Code = "2153" Infor = "Arepa con queso" Cost = 1500 ResttNit = "12345")	True. El producto sólo se añadió la primera vez, el tamaño del array es 1
OBJETIVO DE LA PRUEBA: Verificar que se añada un restaurante correctamente y lo busque				
RestaurantManager	searchRestaurantNit	setupScenary_1	Restaurante: Name = "Las delicias de Gallo" Nit = "12345" Manager = "Gallo" Restaurante: Name = "Las delicias de Colombia" Nit = "1234" Manager = "Camilo"	True. El restaurante buscado por su NIT = "1234" se encuentra en la posición 1 del array
OBJETIVO DE LA PRUEBA: Verificar que se añada un cliente correctamente y lo busque por su nombre y apellido				
RestaurantManager	searchClientName	setupScenary_2	Cliente: Name = "Camilo" LastName = "Ramirez" IdNum = "1548225645" idType = 4 Telephone = "3124865794" Address = "Cali" Cliente: Name = "Andrea" LastName = "Corrales" IdNum = "2154531341" idType = 1 Telephone = "31254867624" Address = "Cali"	True. El restaurante buscado por su NIT = "1234" se encuentra en la posición 1 del array