

Giovanni Mosquera Diazgranados

A00365672

Tarea Integradora 1

Requerimientos funcionales:

Nombre:	R1: Add a client
Entrada:	Name, Identification type, Identification number, Phone, Address
Salida:	Client added successfully
Descripción:	Add a client to the software, it will be used to created orders

Nombre:	R2: Add a restaurant
Entrada:	Name, nit, name admin
Salida:	Restaurant added successfully
Descripción:	Add a restaurant to the software, it will be used to created orders

Nombre:	R3: Add a product
Entrada:	Name, description, price, restaurantNit, code
Salida:	Product added successfully or can't be added because a restaurant with this nit don't exists
Descripción:	Add a product to the software, it will be used to created orders

Nombre:	R4: Create an order
Entrada:	RestaurantNit, clientCode
Salida:	Orde created successfully or can't be created because a restaurant or client with these codes don't exists
Descripción:	Add a client to the software, it will be used to created orders

Nombre:	R5: Save data
Entrada:	Object to be saved
Salida:	Data saved successfully
Descripción:	Save the application data in binary files with serialization

Nombre:	R6: Load data
Entrada:	Object to be loaded
Salida:	Data loaded successfully
Descripción:	Load data to be used in the application, this data will be stay in binary files transparent to the users

Nombre:	R7: Import data from CSV File
----------------	-------------------------------

Entrada:	Object to be imported
Salida:	Data imported successfully
Descripción:	Import data from a CSV file including clients, restaurants, orders and products

Nombre:	R8: Create a report in CSV File using a custom separator
Entrada:	Separator
Salida:	Report created successfully
Descripción:	Create a custom report with orders, restaurant and client's data using a custom separator selected by the user

Nombre:	R9: Edit data
Entrada:	Object to be edited, data to be replaced
Salida:	Object edited successfully
Descripción:	Edit an object with new data, in the order case, the status can't be placed in past

Nombre:	R10: Show sorted information
Entrada:	ArrayList to be sorted and the order
Salida:	Print in console the data sorted
Descripción:	Sort data in ArrayLists using different algorithm sort types

Nombre:	R11: Search a client with name
Entrada:	Client name
Salida:	Print the finded user if it exists
Descripción:	Use binary search to select a user using her name

Pruebas Unitarias:

Nombre	Clase	Escenario
SetupStage1()	Client	A object of type Client with typelidentity = 3, IdentityNumber = 1006055396, name = "Giovanni Mosquera", Phone = "3225034089", Address = "Cra 1C 2 # 60-18"

Nombre	Clase	Escenario
SetupStage1()	Software	Objeto de la clase Software vacío

Nombre	Clase	Escenario
SetupStage2()	Software	Objeto de la clase Software con un cliente añadido: name = Giovanni Mosquera identityNumber= 1006055396 identityType = 1 Phone = 4880416 Address = Cra 1C 2 #60-18

Nombre	Clase	Escenario
SetupStage1()	Order	Objeto de la clase Order: clientCode = 1006055396 restaurantNit = 1006
Nombre	Clase	Escenario
SetupStage1()	Product	Objeto de la clase Product: name = mcBurger Code = 1001 price = 12000 Description = Burgen with extra cheese RestaurantNit = 94503
Nombre	Clase	Escenario
SetupStage1()	Restaurant	Objeto de la clase Restaurant: Name = McDolands nameAdmin = "Giovanni Mosquera RestaurantNit = 94503

Pruebas:

Objetivo de la Prueba:.	Comprobar que el tipo de identificacion se convierte a entero y corresponde al entregado por parámetro			
Clase	Método	Escenario	Valores de Entrada	Resultado
Client	getIdentificationTypeForExport()	SetupStage1()	Ninguno	El tipo de identificacion que retorna en entero es 3

Objetivo de la Prueba:.	Comprobar que cambia el numero de identificacion			
Clase	Método	Escenario	Valores de Entrada	Resultado
Client	setIdentificationNumber()	SetupStage1()	IdentificationNumber = 94311890	El tipo numero de identificación

				del usuario es: "94311890"
--	--	--	--	-------------------------------

Order:

Objetivo de la Prueba:.	Comprobar que cambia el numero de identificacion			
Clase	Método	Escenario	Valores de Entrada	Resultado
Order	getStatus()	SetupStage1()	Ninguno	El estado de la orden es 1

Objetivo de la Prueba:.	Cambiar el estado de una orden en formato entero			
Clase	Método	Escenario	Valores de Entrada	Resultado
Order	setStatus()	SetupStage1()	Status = 4	El nuevo estado de la orden es 4,"Delivered"

Product:

Objetivo de la Prueba:.	Cambiar el nombre de un producto			
Clase	Método	Escenario	Valores de Entrada	Resultado
Product	setName()	SetupStage1()	Name = McCheese	El nuevo nombre del producto es = McCheese

Objetivo de la Prueba:.	Consultar el código de un producto			
Clase	Método	Escenario	Valores de Entrada	Resultado
Product	getCode()	setupStage1()	Ninguno	Product code = 1001

Restaurant

Objetivo de la Prueba:.	Cambiar el nombre de un restaurant			
Clase	Método	Escenario	Valores de Entrada	Resultado
Restaurant	setName()	SetupStage1()	Name = CheeseMcBurger	El nuevo nombre del restaurante es = CheeseMcBurger

