

Integrantes: Giovanni Mosquera A00365672 - Juan Pablo Sanin A00296776 - Juan Camilo Zorrilla A00365972

Requerimientos Funcionales:

- **RF1** Agregar un nuevo usuario vendedor, debe permitir crear un nuevo usuario con los atributos correspondientes los cuales son nombre, apellido, número de identificación, correo electrónico, contraseña, nombre de usuario, ruta de la foto, calificación del usuario, historial de ventas y artículos que vende. **(Funcionando)**
- **RF2** Agregar artículo para vender, debe permitir crear un nuevo artículo a la lista de artículos que ofrece el usuario vendedor. Debe tener los atributos correspondientes que son nombre del artículo, código, precio, descripción, foto, cantidad. **(Funcionando)**
- **RF3** Visualizar lista de artículos, permite mostrar en pantalla la lista de artículos que el vendedor posee actualmente. Debe tener los atributos correspondientes que son nombre del artículo, código, precio, descripción, foto y cantidad.
- **RF4** Agregar artículo al historial de ventas, debe permitir agregar un nuevo artículo al historial de ventas de manera automática en el momento en el que un usuario comprador efectúa esta compra.
- **RF5** Visualizar historial de ventas, permite mostrar en pantalla la lista de las ventas de artículos que el vendedor ha efectuado hasta la fecha. Debe tener los atributos correspondientes que son nombre del artículo, código, precio, descripción, foto y cantidad.
- **RF6** modificar estado de la orden, debe permitir al usuario vendedor modificar el estado de una orden.
- **RF7** Agregar un nuevo usuario comprador, debe permitir crear un nuevo usuario con los atributos correspondientes los cuales son nombre, apellido, número de identificación, correo electrónico, contraseña, nombre de usuario, ruta de la foto, canasta de compras y historial de compras. **(Funcionando)**
- **RF8** Agregar un nuevo artículo a la canasta, debe permitir agregar un nuevo artículo al carrito de compras, este artículo debe contener los atributos correspondientes que son nombre del artículo, código, precio, descripción, foto y cantidad.
- **RF9** Eliminar un artículo de la canasta, debe permitir eliminar un nuevo artículo del carrito de compras.
- **RF10** Visualizar artículos de la canasta, debe permitir mostrar en pantalla los artículos guardados en la canasta en ese instante, estos artículos deben contener los atributos correspondientes que son nombre del artículo, código, precio, descripción, foto y cantidad.
- **RF11** Realizar una nueva compra, debe permitir al usuario comprador realizar una nueva compra creando una orden con código de orden, lista de artículos y fecha de pedido.
- **RF12** Calificar al vendedor, debe permitir al usuario comprador calificar al vendedor justo después de haber realizado una compra.
- **RF13** Agregar compra a historial de compras, al realizar una compra se debe agregar esta compra al historial de compras del usuario comprador de manera automática.

- **RF14** Visualizar historial de compras, debe permitir al usuario comprador visualizar el total de las compras que ha realizado.
- **RF15** Agregar un nuevo usuario administrador, debe permitir crear un nuevo usuario con los atributos correspondientes los cuales son nombre, apellido, número de identificación, correo electrónico, contraseña, nombre de usuario, ruta de la foto, lista de usuarios y lista de todos los artículos.
- **RF16** Visualizar usuarios, debe permitir al administrador visualizar en pantalla el total de usuarios a partir de un criterio de orden que él seleccione.
- **RF17** Visualizar artículos, debe permitir al administrador visualizar en pantalla el total de artículos a partir de un criterio de orden que él seleccione.
- **RF18** Importar usuarios, debe permitir al administrador importar una lista de usuarios al programa.
- **RF19** Importar artículos, debe permitir al administrador importar una lista de artículos al programa.
- **RF20** Banear vendedor, debe permitir al administrador realizar el baneo a un vendedor.

Requerimientos no funcionales:

1. El programa debe ser programado en Java.
2. El programa debe tener una interfaz gráfica de usuario hecha utilizando scene builder e implementando javafx.
3. El programa debe tener información persistente.
4. El diseño de las interfaces debe ser minimalista.
5. La paleta de colores debe ser de azules claros y oscuros.
6. El programa debe tener un logo de la empresa.
7. El programa debe usar íconos.
8. El programa debe manejar una iconografía constante.

Pruebas Unitarias:

Nombre	Clase	Escenario
setupStage1	Article	Vacío
setupStage1	UserBuyer	Vacío
setupStage1	UserSeller	Vacío
setupStage1	User	Vacío
setupStage1	Administratior	Vacío
setupStage1	Order	Se crea un objeto de tipo Articulo
setupStage1	Technology	Vacío

setupStage1	CellPhone	Vacío
setupStage1	TV	Vacío
setupStage1	HomeAppliances	Vacío
setupStage1	Fridge	Vacío
setupStage2	Fridge	Un objeto creado, que si es Smart y si es froze
setupStage3	Fridge	Un objeto creado, que no es Smart y no es froze
setupStage1	Stove	Vacío
setupStage2	Stove	3 objetos Stove creados, cada uno con difirente tipo de estufa (1-Gas,2-Elctrica,3.gas y eléctrica)
setupStage1	SkyMarket	Se crea un objeto de SkyMarket vacío
setupStage2	SkyMarket	Se crea un objeto de SkyMarket con 4 usuarios añadidos por defecto

Clase	Método	Escenario	Valores de Entrada	Resultado
Article	Article(Constructor)	setupStage1	Name: "Y9" Code: "12345" Price: 1000000 Description:" Celulargama media" Picture = "data/PictureHuaweiY9" Quantity = 3 NextArticle = null	El método constructor funciona correctamente
Article	Setters y getters	setupStage1	Name: "Mate 20" Code: "1234" Price: 3000000 Description:" Celular con muy buena camara" Picture = "data/Picture1" Quantity = 2 NextArticle = null	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
User	User(Constructor)	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul"	El método constructor funciona correctamente.
User	Setters y getters	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul"	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
UserBuyer	UserBuyer(Constructor)	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul" Basket = null History = null	El método constructor funciona correctamente
UserBuyer	Setters y getters	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul" Basket = null History = null	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
-------	--------	-----------	--------------------	-----------

UserSeller	UserSeller(Constructor)	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul"	El método constructor funciona correctamente
UserSeller	Setters y getters	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul" Calification = 0 History = null sellArticles = null	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
Administratior	Administratior(Constructor)	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul"	El método constructor funciona correctamente
Administratior	Setters y getters	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul"	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
Administratior	Administratior(Constructor)	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul"	El método constructor funciona correctamente
Administratior	Setters y getters	setupStage1	Name: "Raul" lastName: "Gomez" Identification: "1005040032" Email : "raulg@gmail.com" Password: "raulg10" Username: "raulito20" Picture = "data/PictureRaul"	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
Order	Order(Constructor)	setupStage1	orderCode; 2341 articles : ObjectArticle	El método constructor funciona correctamente
Order	Setters y getters	setupStage1	orderCode; 123332 articles : ObjectArticle	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
Technology	Technology(Constructor)	setupStage1	Name: "Y9" Code: "12345" Price: 1000000 Description:" Celular gama media"	El método constructor funciona correctamente

			Picture = "data/PictureHuaweiY9" Quantity = 3 NextArticle = null PrevArticle = null BatteryWats = 2000 ScreenSize =20 Ram = 4 Processor = Hisilicon Kirin	
Technology	Setters y getters	setupStage1	Name: "Y9" Code: "12345" Price: 1000000 Description:" Celular gama media" Picture = "data/PictureHuaweiY9" Quantity = 3 NextArticle = null PrevArticle = null BatteryWats = 2000 ScreenSize =20 Ram = 4 Processor = Hisilicon Kirin	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
CellPhone	Technology(Constructor)	setupStage1	Name: "Y9" Code: "12345" Price: 1000000 Description:" Celular gama media" Picture = "data/PictureHuaweiY9" Quantity = 3 NextArticle = null PrevArticle = null BatteryWats = 2000 ScreenSize =20 Ram = 4 Processor = Hisilicon Kirin	El método constructor funciona correctamente

			Number sims = 2 Number cameras = 3	
CellPhone	Setters y getters	setupStage1	Name: "Mate 20" Code: "1234" Price: 3000000 Description: "Celular con muy buena camara" Picture = "data/Picture1" Quantity = 2 NextArticle = null PrevArticle = null BatteryWats = 2000 ScreenSize =20 Ram = 4 Processor = Hisilicon Kirin Number sims = 2 Number cameras = 3	Los getters y setters realizan su trabajo correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HomeAppliances	HomeAppliances(Constructor)	setupStage1	Name: "Nevera LG" Code: "12346" Price: 2000000 Description: "Nevera con alto nivel de congelamiento" Picture = "data/PictureNeveraLG" Quantity = 4 NextArticle = null PrevArticle = null Weight =100 Capacity =1000 WattsConsum =200 Height = 200 Width = 100	El método constructor funciona correctamente

HomeAppliances	Setters y getters	setupStage1	Name: "Nevera LG" Code: "12346" Price: 2000000 Description: "Nevera con alto nivel de congelamiento" Picture = "data/PictureNeveraLG" Quantity = 4 NextArticle = null PrevArticle = null Weight =100 Capacity =1000 WattsConsum =200 Height = 200 Width = 100	Los getters y setters realizan su trabajo correctamente.
----------------	-------------------	-------------	--	--

Clase	Método	Escenario	Valores de Entrada	Resultado
Fridge	Fridge(Constructor)	setupStage1	Name: "Nevera LG" Code: "12346" Price: 2000000 Description: "Nevera con alto nivel de congelamiento" Picture = "data/PictureNeveraLG" Quantity = 4 NextArticle = null PrevArticle = null Weight =100 Capacity =1000 WattsConsum =200 Height = 200 Width = 100 Smart = true frost = False	El método constructor funciona correctamente
Fridge	Setters y getters	setupStage1	Name: "Nevera LG" Code: "12346" Price: 2000000 Description: "Nevera con alto nivel de congelamiento"	Los getters y setters realizan su trabajo correctamente.

			Picture = “data/PictureNeveraLG” Quantity = 4 NextArticle = null PrevArticle = null Weight =100 Capacity =1000 WattsConsum =200 Height = 200 Width = 100 Smart = true frost =true	
Fridge	testIsSmartStringAndIsNoFrozeStringTrue	setupStage2	Smart = true frost = true	Los metodos retornan “SI” de manera correcta
Fridge	testIsSmartStringAndIsNoFrozeStringFalse	setupStage3	Smart = false frost = false	Los metodos retornan “NO” de manera correcta

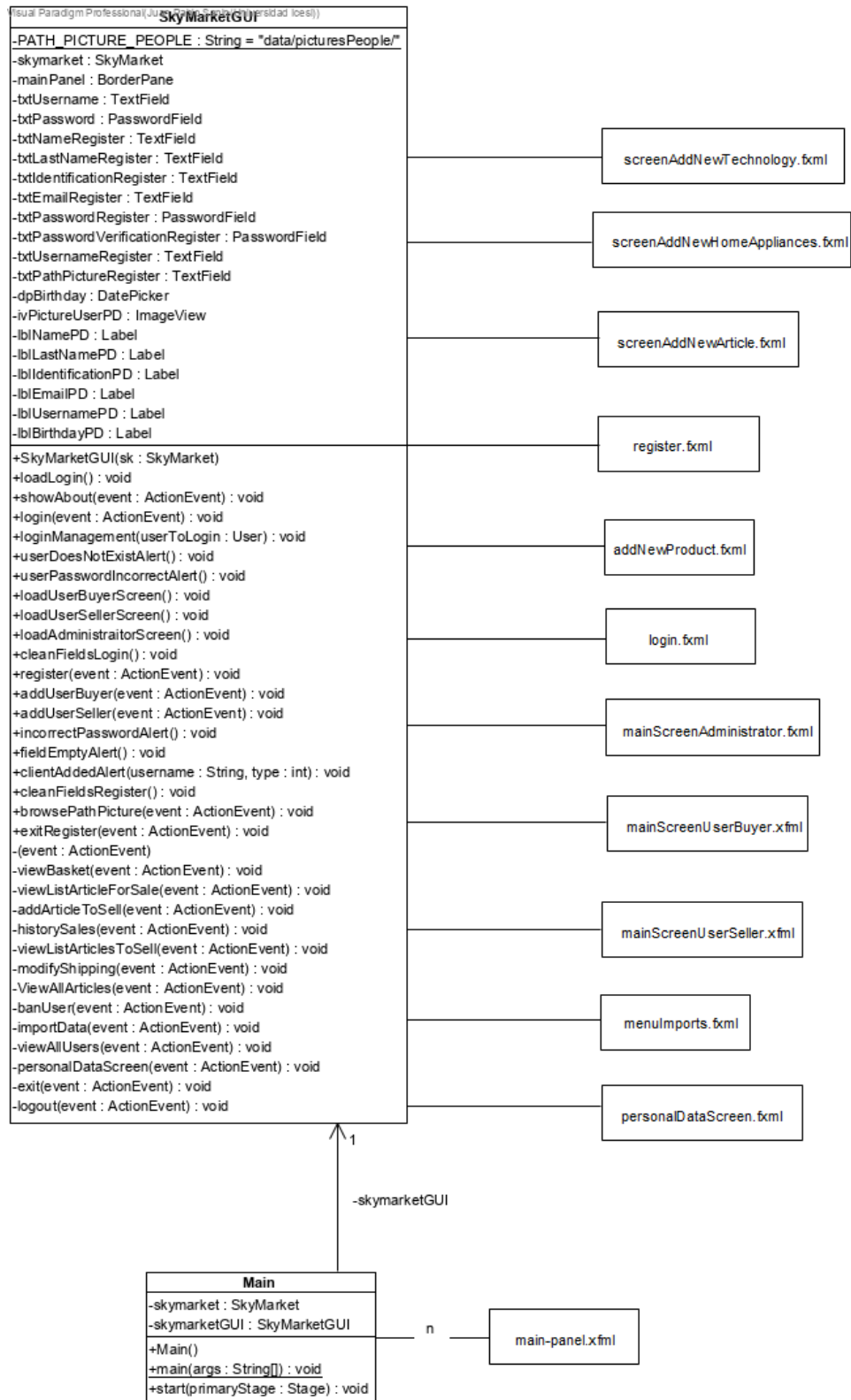
Clase	Método	Escenario	Valores de Entrada	Resultado
Stove	Stove(Constructor)	setupStage1	Name: “Estufa mabe” Code: “1223” Price: 700000 Description: “Estufa elegante”	El método constructor funciona correctamente

			Picture = "data/PathPictureEstufa" Quantity = 2 NextArticle = null PrevArticle = null Weight =80 Capacity =500 WattsConsum =1000 Height = 1 Width = 1 NumberOfNozzles = 4 typeStove = 3	
Stove	Setters y getters	setupStage1	Name: "Estufa mabe" Code: "1223" Price: 700000 Description: "Estufa elegante" Picture = "data/PathPictureEstufa" Quantity = 2 NextArticle = null PrevArticle = null Weight =80 Capacity =500 WattsConsum =1000 Height = 1 Width = 1 NumberOfNozzles = 4 typeStove = 3	Los getters y setters realizan su trabajo correctamente.
Stove	getTypeStoveString	setupStage2	Nada	El metodo retorna los string correctos.
Clase	Método	Escenario	Valores de entrada	Resultado
SkyMarket	newUser	setupStage1	Name: "Giovanni" Lastname: "Mosquera" Identification: "1006055396" Email: Giovanni2414g@gmail.com Password: "1234" Username: "xGiovanni" Image: "images/" Date: LocalDate.now()	Un nuevo usuario ha sido añadido al arrayList de usuarios

			UserType = 0	
SkyMarket	binarySearchUser	setupStage2	Se buscan los nombres: Username: "xGiovanni" Username: "Estela" Con el método binarySearchUser	El método debe retornar el usuario "xGiovani" y reportar como nulo si se busca el usuario "Estela"
SkyMarket	Login	setupStage1	Name: "Giovanni" Lastname: "Mosquera" Identification: "1006055396" Email: Giovanni2414g@gmail.com Password: "1234" Username: "xGiovanni" Image: "images/" Date: LocalDate.now() UserType = 0	La variable currentUser no debe estar nula
SkyMarket	Logout	setupStage1	Name: "Giovanni" Lastname: "Mosquera" Identification: "1006055396" Email: Giovanni2414g@gmail.com Password: "1234" Username: "xGiovanni" Image: "images/" Date: LocalDate.now() UserType = 0	Deberá de loguearse correctamente el usuario y verificar el currentUser no es nulo para verificar el login, luego de llamar al método logout la variable currentUser deberá estar con valor nulo
Clase	Método	Escenario	Valores de entrada	Resultado
SkyMarket	Getters y Setters	setupStage1		Los getters y setters realizan su trabajo correctamente.

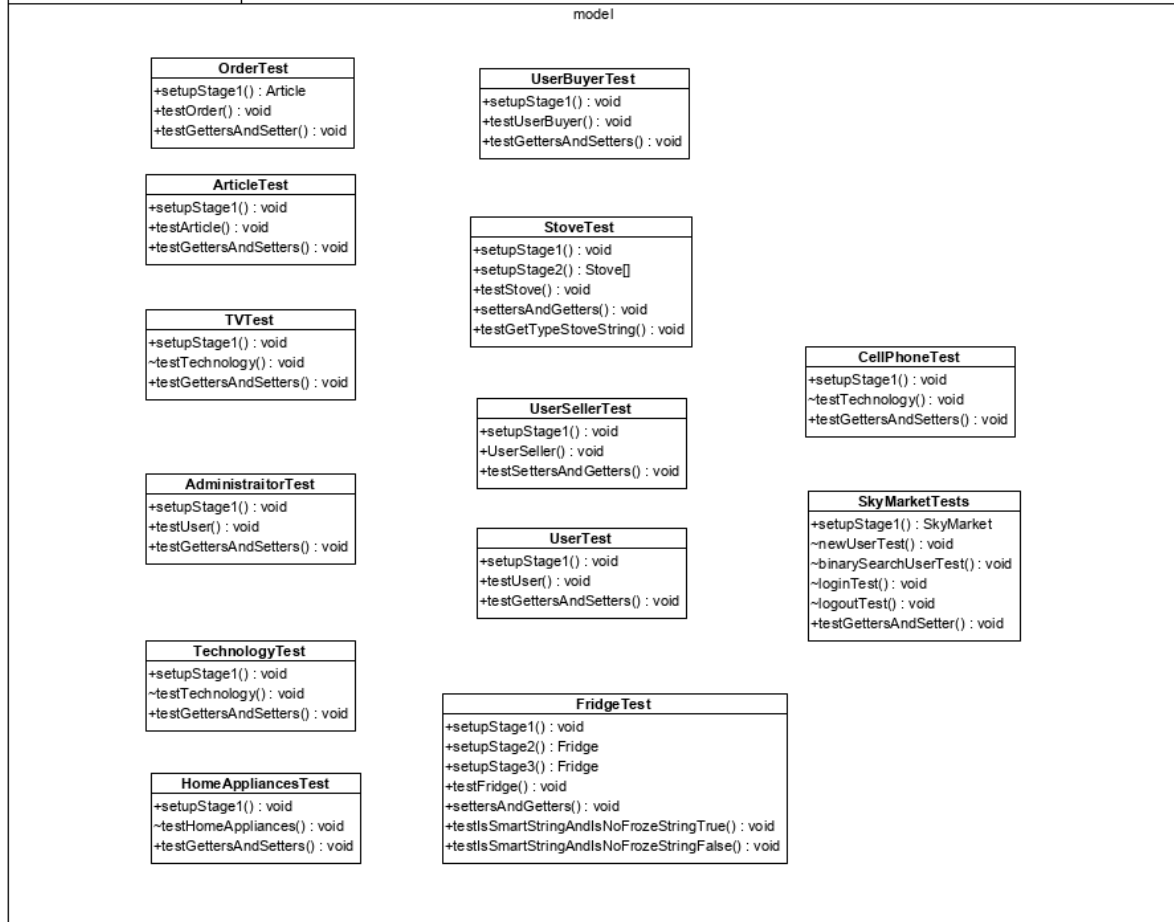
Diagramas de Clases:

1. GUI:

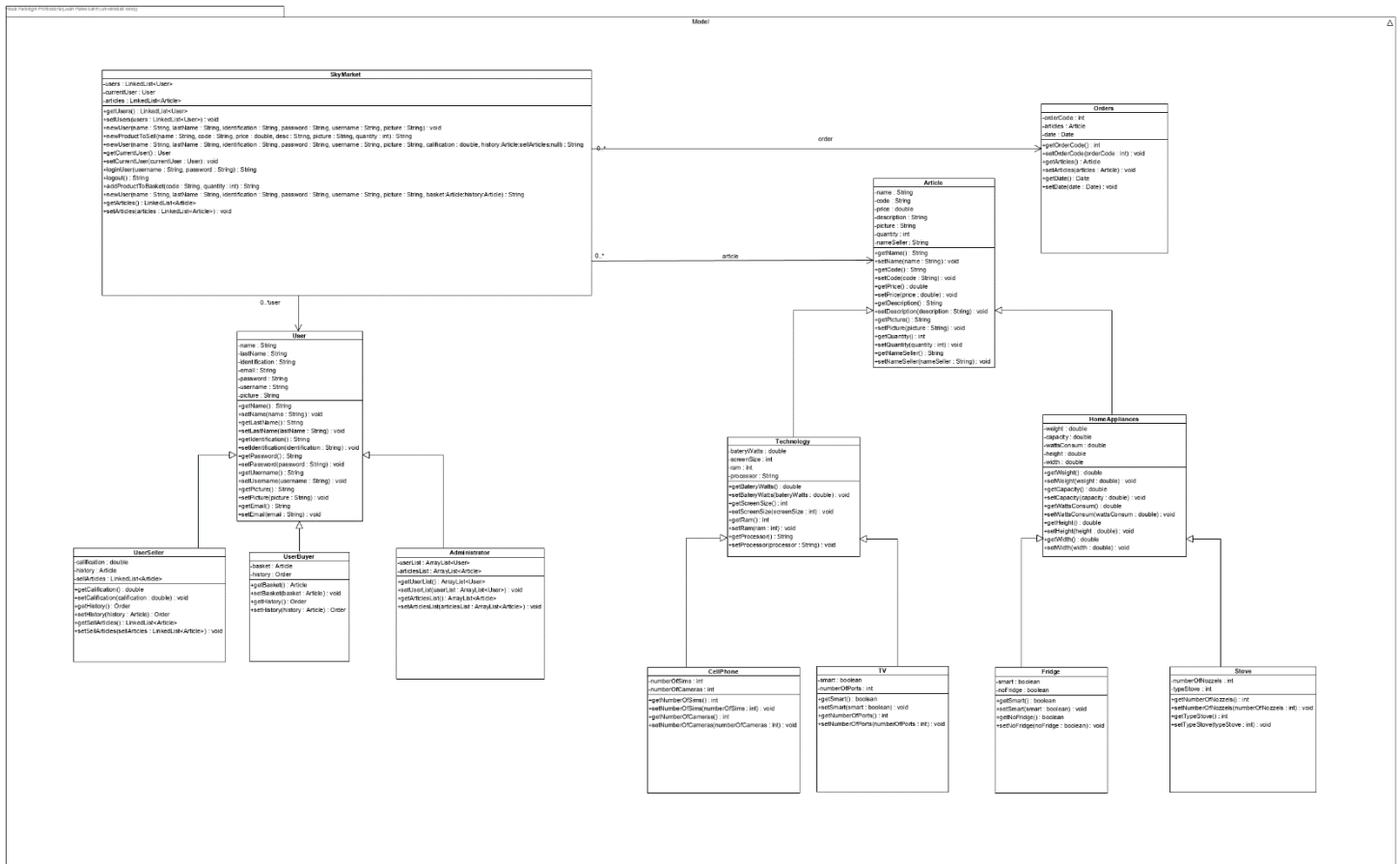


2. Tests:

Visual Paradigm Professional (Juan Pablo Sanin) (Universidad Icesi)



3. Model:



Para ver los diagramas de clases con mayor resolución, en el proyecto en la carpeta docs en la carpeta class diagram images estarán las imágenes por separado.