

## Tarea Integradora 3

Tienda Monarca

Integrantes:

Sebastian Navia Ramirez<sup>1</sup>

Gianni Benavides García<sup>2</sup>

Dirigido a:

Juan Reyes

Santiago de Cali, 18 de mayo de 2021

Algoritmos y programación II, Facultad de Ingeniería

Ingeniería Telemática<sup>1</sup> y Ingeniería de Sistemas<sup>2</sup>.



## TIENDA MONARCA:

### **Enunciado:**

Tienda Monarca, es un pequeño supermercado el cual se ha caracterizado por ser muy flexible con sus clientes al momento de cancelar sus compras, permitiéndoles llevar productos fiados.

En la tienda monarca podemos comprar una variedad de productos, que van desde elementos de la canasta familiar, dulces, gaseosas etc. Todos estos se venden en paquetes y de forma individual.

El dueño de la tienda siempre ha tenido que llevar sus apuntes en cuadernos, pero se ha visto en la necesidad de optimizar su tiempo y orden. Por lo tanto se hace indispensable el desarrollo de un programa, el cual le permita agilizar, ordenar, hacer validación de datos y de seguridad. El programa debe estar en la capacidad de gestionar los productos, clientes, inventarios.

El usuario principal se creara con valores definidos como (usuario: admin y clave: 123), también podrá ser editable.

En caso de los productos estos deben tener un nombre, una cantidad en existencia, un precio y un proveedor.

El proveedor debe contar con el nombre, su identificación, la dirección de su oficina y un número de contacto.

Los clientes tendrán un nombre, número de identificación y un número de contacto. Únicamente se creara un cliente como deudor cuando este solicite los productos fiados, este tendrá un nombre, un número de identificación, número de contacto y fecha de compra. Los datos serán OBLIGATORIOS para todo comprador con el fin de llevar un orden ante la Dian.

El inventario debe mostrar cuantos elementos hay de cada producto y debe informar cuando se venda la última unidad, cambiando así su estado de disponible a agotado.

El programa debe mostrar en orden descendente el listado de clientes deudores por su nombre. También debe permitir cambiar el orden por fecha y por la cantidad que debe. Al igual que debe permitir abonar a la deuda de cada deudor, buscando a este por su número de identificación. Cuando este pague la totalidad de la deuda avisar y borrar de la lista de deudores y solo guardarlo como cliente.

Debido a la contingencia que vive actualmente el mundo por el covid el programa debe estar en la capacidad de evaluar qué días pueden hacer las compras los clientes, dependiendo del día calendario y el último número de su cedula. Solo podrán entrar los días pares los que su identificación termine en número par y viceversa con los impares, se le debe negar la posibilidad de comprar a personas que no posean cedula, pasaporte o cedula extranjera.

### **Requerimientos Funcionales.**

**Req 1:** permitir la creación, la modificación, y la manipulación de los datos de los clientes.  
Nombre, identificación, dirección, teléfono, tipo de documento.

**Req 2.1:** Agregar cliente.

**Req 2.2:** Eliminar cliente.

**Req 2.3:** Ordenar la lista de cliente alfabéticamente, por identificación

**Req 2.4:** Mostrar la Información de un cliente.

**Req 2:** Gestionar los Productos que se ofrecen al consumidor. Como el nombre, una cantidad en existencia, un precio y un proveedor, valor de compra (unitario) y valor de venta (unitario).

**Req 2.1:** Agregar producto.

**Req 2.2:** Eliminar producto.

**Req 2.3:** Modificar la cantidad.

**Req 2.4:** Mostrar la disponibilidad.

**Req 2.5:** Mostrar la información del producto.

**Req 3:** Gestionar los clientes deudores, nombre del cliente, identificación, dirección, teléfono, monto de la deuda, fecha de la compra.

**Req 3.1:** Agregar cliente.

**Req 3.2:** Eliminar cliente.

**Req 3.3:** Ordenar la lista de cliente alfabéticamente, por identificación o deuda

**Req 3.4:** Mostrar la Información de un cliente deudor.

**Req 3.5:** Abonar a la deuda.

**Req 3.6:** Cambiar el estado de deudor a solo cliente.

**Req 4:** Gestionar Inventario, mostrar todos los productos con su respectiva cantidad y su proveedor y valor de compra.

**Req 4.1:** Mostrar los productos en el inventario.

**Req 4.2:** Ordenar lista de inventario por nombre.

**Req 4.3:** Modificar la cantidad de elementos en inventario.

**Req 5:** Gestionar al usuario principal, nombre de usuario y contraseña.

**Req 5.1:** hacer uso del usuario principal o crear otro.

**Req 7.** Incluir permanentemente en el programa la hora y la fecha actual.

**Req 8.** Generar un archivo CSV de clientes, con los datos del nombre, dirección y teléfono.

**Req 9:** El programa debe guardar toda su información a través de la serialización de sus objetos en archivos. Este guardado debe ser transparente para el usuario del programa, es decir, cada vez que se registre o actualice información, esta se guardará en los archivos serializados.

**Req 10:** Los clientes solo podrán comprar productos los días permitidos según su identificación.

**Req 11:** Gestionar las ventas de la tienda.

**Req 11.1:** Los clientes pueden compara en efectivo o tarjeta de crédito.

**Req 12:** Registrar los proveedores, pueden ser a crédito o de contado.

**Req 13:** Gestionar la información de los usuarios, (administradores de las cuentas. (Empleados)).

**Req 13.1** Añadir una cuenta de usuario.

## **Justificación**

Es necesario la participación de los dos para optimizar nuestro tiempo y avanzar con mayor agilidad en la codificación del proyecto. Además que gracias a los dos podemos hacer una lluvia de ideas donde podamos escoger adecuadamente soluciones para los obstáculos que se puedan presentar a lo largo del trabajo.

La interfaz gráfica se necesita de diferentes opiniones para crear una interfaz más agradable y fácil para su utilización.

## **README**

### **Tienda Monarca.**

"Tienda Monarca", is a program created with the intention of saving and manipulating all the data and information of the products for sale, the customers.

It also has a very special section, the store "Tienda Monarca", is very flexible with payments, so it allows to trust customers. The development is focused on this last idea.

functional for Windows, Linux and Mac OS computer operating systems.

Snake\_and\_Ladders

\*Project made with:Eclipse IDE For java developers

\*Language: Java 8

\*Operating system:Windows 10

\*Project carried out by:

<https://github.com/Sebastianavia>

<https://github.com/G20-00>

Enlace: [https://github.com/Sebastianavia/Tienda\\_Monarca/tree/main/docs](https://github.com/Sebastianavia/Tienda_Monarca/tree/main/docs)

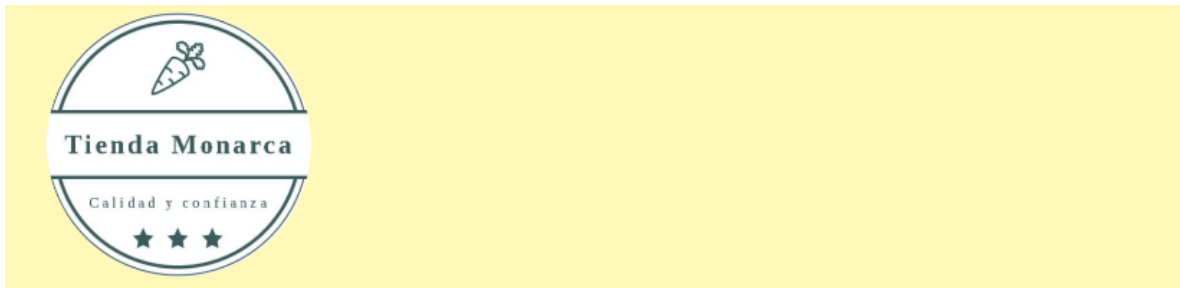
-CHANGELOG\_1

### Changed

add....

1. Functional Requirements (we modify requirement 5, 11 and add requirement 13)
2. Non-Functional Requirements.
3. Model Class Diagram including inheritance and polymorphism (abstract classes, non-abstract superclasses and interfaces).
4. Class Diagram of the User Interface.
5. Design of Unit Tests.
6. Class Diagram of Unit Tests.

## Mockups



join





00.00.00

dd/mm/aa

## Menu

Cientes

Deudores

Inventario

Producto

Venta

Proveedores



## Cientes

Nombre	Identificacion	Direccion	Telefono
Tabla sin contenido			



## Añadir clientes

Nombre

Identificacion

Telefono

Direccion

Añadir

salir



## Deudores

Nombre	Identificacion	Direccion	Telefono	Deuda	Fecha
Tabla sin contenido					



## Productos

1

Nombre	precio	Diponibilidad	Proveedor
Tabla sin contenido			

Añadir

Salir



## Inventario

Nombre producto	Valor compra	Proveedor	Cantidad
Tabla sin contenido			

Salir



## Venta

productos	ganancia
Tabla sin contenido	

Añadir

Salir



## Añadir Proveedores

Nombre

Telefono

Direccion

Añadir

salir



## Añadir Producto

Nombre

Precio

Proveedor

Disponibilidad

Cantidad

Añadir

salir

## Buscador de clientes

Buscador:

Add

Exit

Nombre

Id

Dirección

Celular

Tabla sin contenido

Ciente. Sin asignar

Buscar

Nombre y apellido separado por

Busqueda binaria

Cantidad:

Ingrese la cantidad

Pagó

Ingrese la cantidad

Debe pagar

Sin asignar

Atras

Crear venta

Lista de Productos:

Nombre	Precios	Disponibilidad
Tabla sin contenido		

Añadir

Eliminar

Nombre	Cantidad	Precio
Tabla sin contenido		

Debe

\$ Sin asignar Pesos

## Requerimientos no funcionales.

### Eficiencia.

1. El sistema debe ser capaz de procesar y calcular los procesos dictados por el administrador en menos de dos segundos, a excepción de las búsquedas binarias y al empezar el programa que se demora 5 segundos para entrar.
2. Al ser un programa contenido en el computador, no necesita internet, y además, solo puede ser usado una vez cuando se esta operando para evitar fallos de guardados de datos.
3. Serializar la información de productos.
4. Serializar la información de Clientes.
5. Serializar la información de proveedores

### Seguridad lógica y de datos

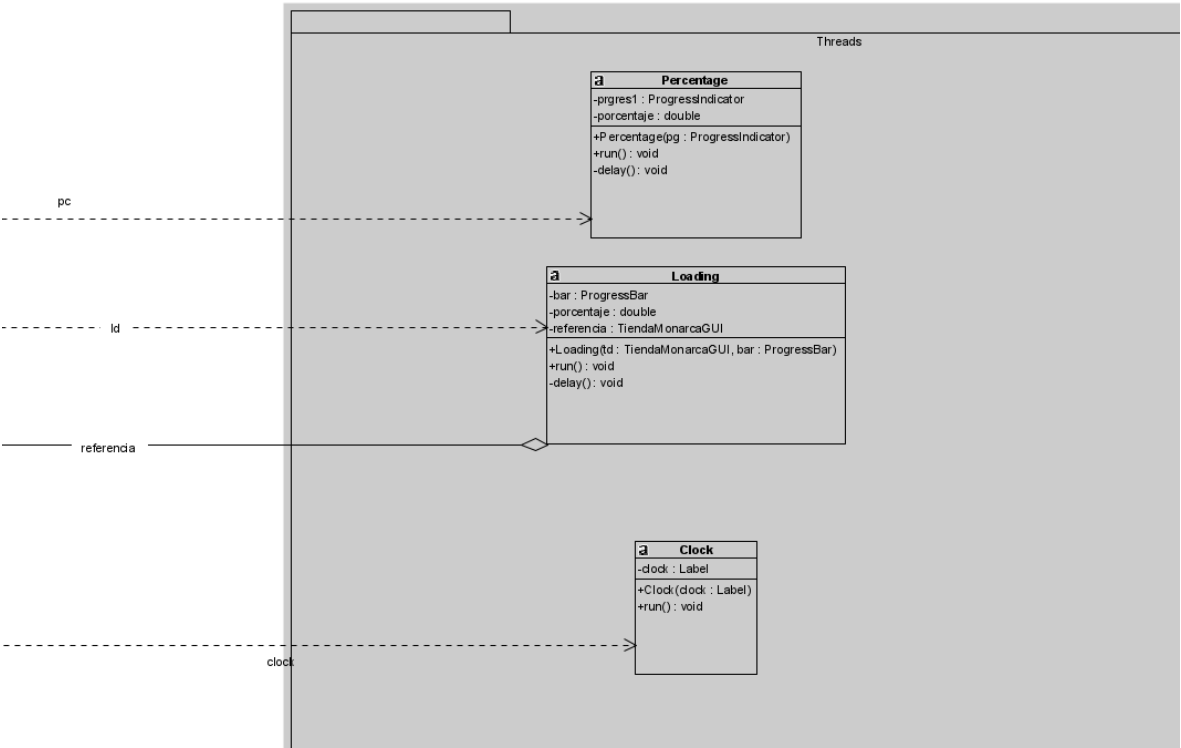
6. Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
7. Solo el administrador podrá hacer uso del programa.

### Usabilidad

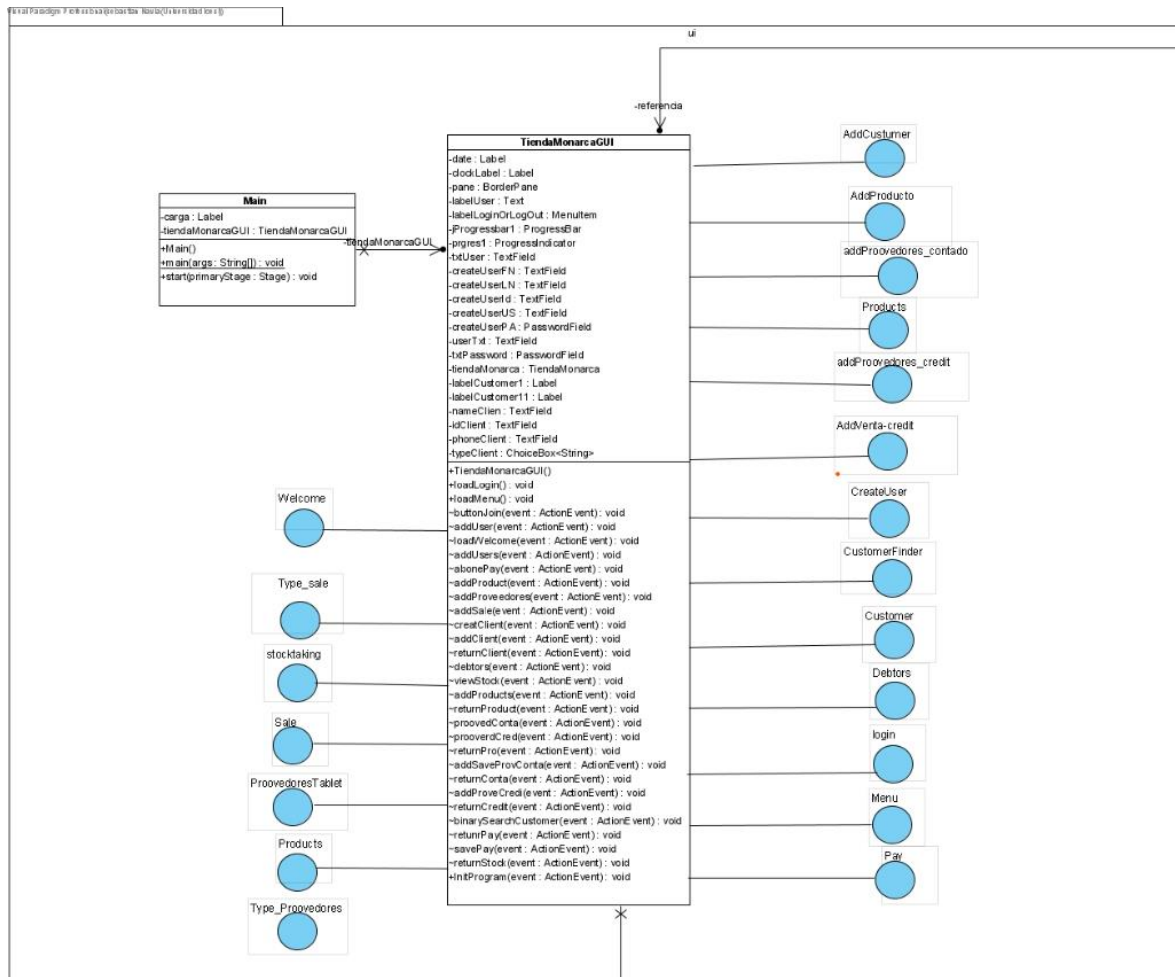
8. El sistema debe contar con manuales de usuario estructurados adecuadamente.
9. El sistema debe poseer interfaces gráficas bien formadas.

Diagrama de Clases

Threads



## Diagrama de Clases UI





## Diagrama de las pruebas unitarias

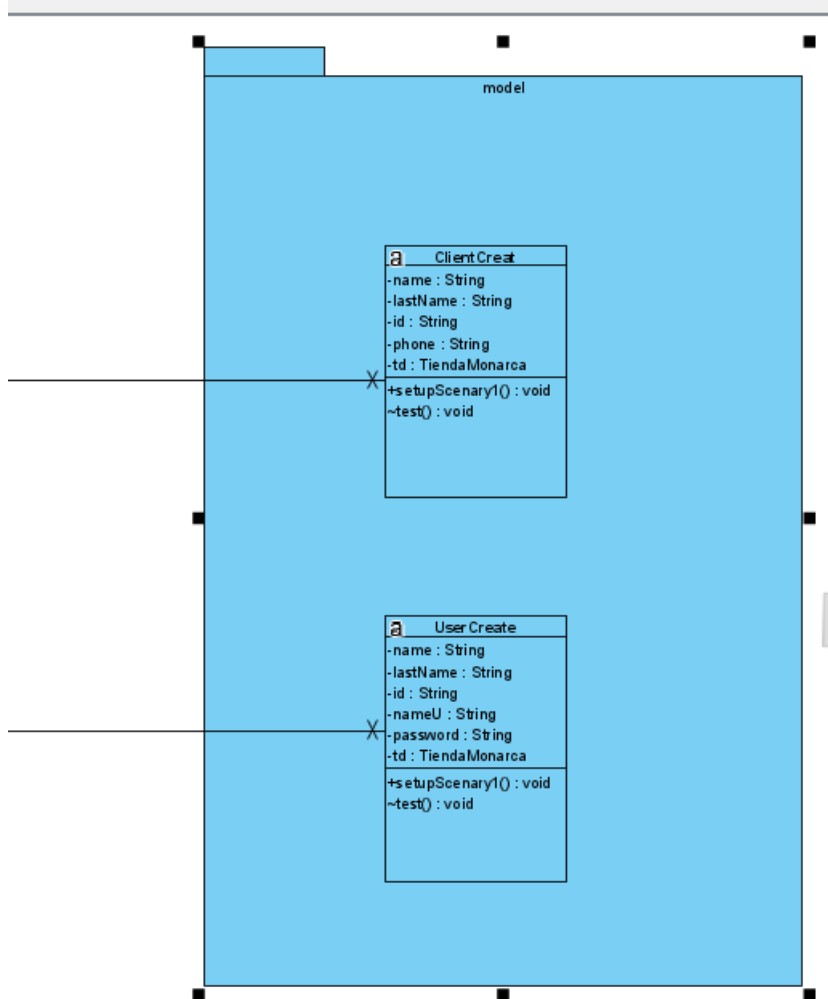


Diagrama de clases model

El diagrama model, es muy grande y es difícil q se observe correctamente te dejaremos el link para q lo puedas explorar de mejor manera

[https://github.com/Sebastianavia/Tienda\\_Monarca/blob/main/docs/diagram/diagram%20class \(all\).jpg](https://github.com/Sebastianavia/Tienda_Monarca/blob/main/docs/diagram/diagram%20class%20(all).jpg)

