

## REQUERIMIENTOS FUNCIONALES

**1) Poder ingresar el perfil de los trabajadores:** el cual contiene el nombre, apellido, salario, numero de teléfono, el cargo asignado dentro del supermercado. (Gerente, Empleados de sección, Cajeros, Almacén). Aquí también se puede elegir si el empleado requiere de crear una cuenta de acceso al sistema. Los empleados que pueden tener una cuenta de acceso al sistema son los CAJEROS, personal del INVENTARIO y el GERENTE.

**2) Iniciar sesión en el programa:** El programa tendrá la capacidad de permitirle el acceso a los empleados que tengan una cuenta de acceso.

**3) Poder generar en un archivo plano la nomina a pagar cada mes,** de acuerdo al salario de los empleados, se debe ver en el archivo cuanto se le paga a cada empleado, y al final del archivo el total a pagar de nomina.

**4) Texto en movimiento en la pantalla:** Se desea que en ventana principal haya un mensaje que se mueva de izquierda a derecha, este mensaje será proporcionado por el gerente para que sea visible a todos los empleados que usen el software.

**5) Guardar el programa:** El programa debe tener una opción de guardar todo el programa para después volver a abrirlo.

Habrà una sección de **INVENTARIO** con las siguientes funcionalidades requeridas:

**6) Agregar productos al inventario:** Poder tener la opción de agregar productos al inventario del supermercado, con su respectivo nombre, cantidad, código, precio, medida (kg, g, L, ml etc.), marca (Bimbo, Coca Cola etc.), Categoría( Alimentos, Limpieza, Bebidas), y una subcategoría la cual se distribuirá para cada diferente Categoría así:

Alimentos:

-Otros

-Fruta

Limpieza:

-Liquido

-Polvo

-Herramientas

Bebidas:

-Gaseosas

-Jugos

**7) Eliminar productos del inventario:** Poder tener la opción de eliminar algún producto ingresando el código del producto y la cantidad a eliminar.

**8) Generar en un archivo plano todo el inventario del supermercado:** El programa debe tener la posibilidad de poder generar en un archivo todos los productos que se encuentren en el inventario del supermercado.

Habrà una sección de **CAJERO** con las siguientes funcionalidades requeridas:

**9)Agregar productos al carrito:** El programa podrá añadir productos, ingresando la cantidad y el código del producto a agregar, estos productos se deben encontrar en el inventario del programa, luego se añadirán a una lista de productos para comprar por el cliente.

**10)Eliminar productos del carrito:** El programa estará en la capacidad de poder quitar algunos productos de la lista de compra del usuario, ingresando la cantidad y el código del producto.

**11)Generar factura de compra:** El programa debe lograr “vender” los productos y generar una factura en una factura en un archivo plano, con los productos vendidos, y el total a pagar por el consumidor.

## REQUERIMIENTOS NO FUNCIONALES

- 1) El programa tendrá total control solamente por el empleado que tenga cargo de GERENTE.
- 2) El programa siempre que vaya a buscar en alguna estructura de datos tendrá que hacer esta búsqueda de forma recursiva.
- 3) Para buscar algún producto en el inventario se requerirá hacerlo con una búsqueda binaria siendo encontrado por el código del producto.
- 4) Se tendrán excepciones propias para los siguientes posibles problemas:
  - Falte información del empleado cuando se intente agregar.
  - Se intente agregar al carrito del cliente un producto que no esté en el sistema, de igual manera para cuando se intente eliminar del inventario un producto inexistente.
  - Falten datos principales al agregar un nuevo producto, tales como (cantidad, nombre, Medida)
- 5) Se deben tener algunos ordenamientos;
  - Cuando se genere el archivo plano con las nominas y la nomina total, los empleados deben ser ordenados por nombre y después por salario, mediante el ordenamiento por burbuja para que en el archivo generado quede de mayor a menor.
  - Los empleados que estén en la TableView de la sección GERENTE estarán ordenados por inserción.
  - Cuando se genere la factura, los productos estarán ordenados de el más costoso al menos costoso.
- 6) Las cuentas de acceso serán guardadas en una lista doblemente enlazada.
- 7) La lista de productos agregados al carrito de compra del cliente estarán en una lista doblemente enlazada.
- 8) Cada vez que se genere una factura de venta, esta se agregará a un árbol binario, para tener cuenta de las ventas, cual ha sido la mayor venta, y la menor.
- 9) El texto en movimiento debe ser concurrente, mientras que el software esté abierto.
- 10) Al iniciar sesión el programa debe lanzar alertas si el nombre de usuario no está registrado en las cuentas de acceso, o si la contraseña es incorrecta.

11) Cuando se esté agregando los productos al inventario, este será inteligente y se irá actualizando a medida que se va indicando que categoría de alimento es, y que subcategoría de esa categoría es, después de seleccionar la categoría y su subcategoría al oprimir un botón con un “+” este dependiendo de qué categoría y subCategoría sea, mostrará un pequeño panel donde pedirá la información adicional que requiera este producto (Los que no necesiten información adicional no despegará este panel), por cada vez que se muestre el panel diferente se hará una clase hija de la clase padre “Product”.

12) Los productos del inventario estarán en un árbol binario por su código.

13) El empleado tendrá como foto de perfil una figura geométrica y un color, (Si hay alguien con la misma foto de perfil, el sistema no podrá dejarlo elegir esta).

14) La foto de perfil del empleado que se mostrará en la pagina principal en la parte superior derecha será concurrente y estará rotando durante todo el programa.

Para esta entrega se necesita que al menos dos requerimientos funcionales estén terminados, por lo que a continuación se listan.

**-El registro de empleados y la creación de su cuenta de acceso:** La puede encontrar en la pagina inicial del programa, luego le da clic en registrar empleado, rellena todos los campos y crea una cuenta de acceso para continuar.

**-El mensaje en movimiento del Gerente:** Esto se podrá ver en la parte superior de la ventana, también podrá modificar su contenido, yendo a la sección del gerente y en la parte inferior derecha escribir el mensaje deseado.

**IMPORTANTE:** Para ingresar a la página principal del Software y poder acceder a todas las funcionalidades puede ingresar con:

username: gustavo  
password: 123

O bien, puede registrar su propia cuenta, **TENGA EN CUENTA QUE** debe registrarse como GERENTE, para poder tener acceso a todo el programa.

# DIAGRAMA DE CLASES DEL MODELO

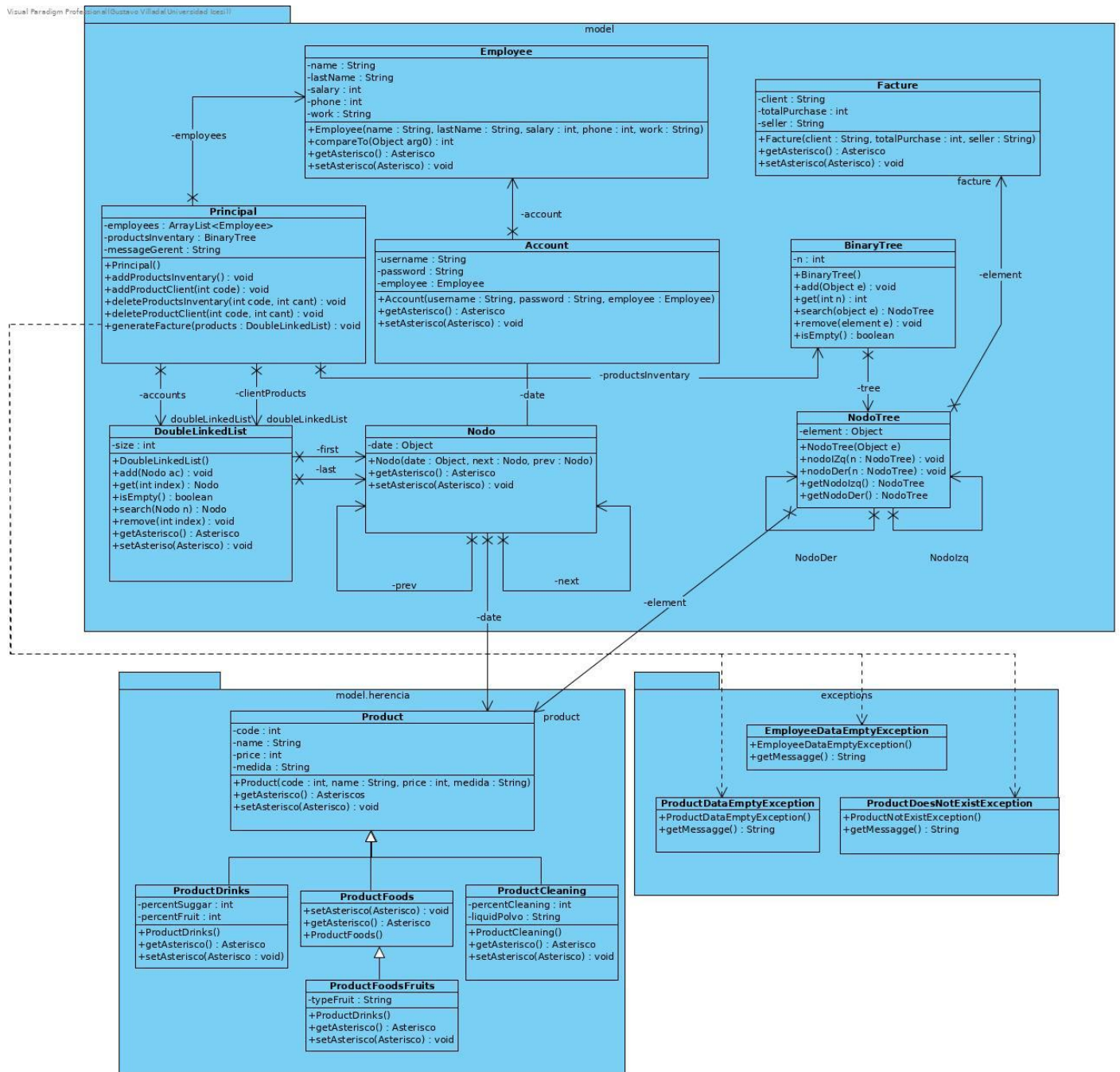
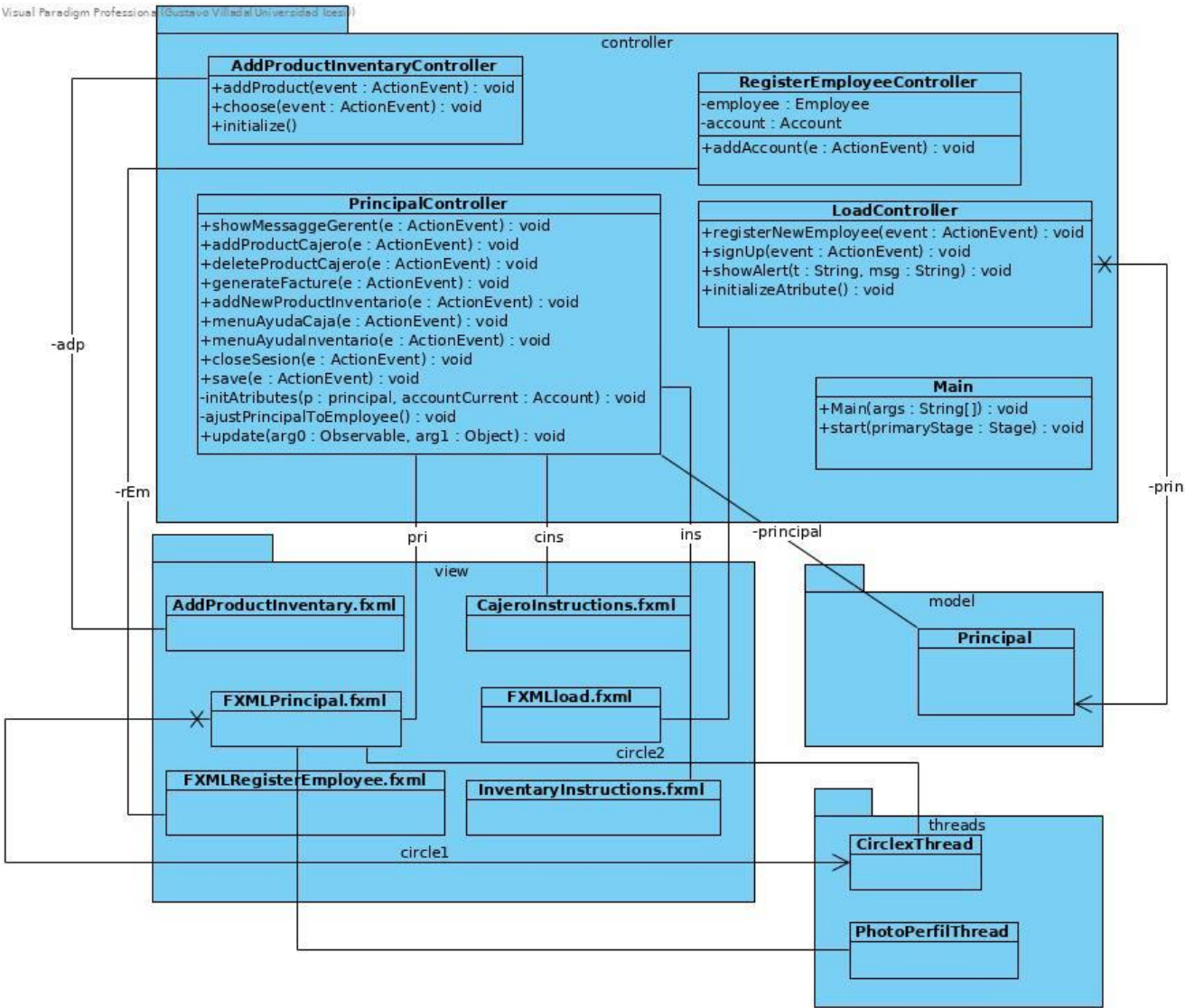


DIAGRAMA DE CLASES CON LA INTERFAZ DE USUARIO.



## DISEÑO DE CASOS DE PRUEBA

### Configuración de los escenarios.

Nombre	Clase	Escenario
setUp1	PrincipalTest	Se inicializa una DoubleLinkedList.
setUp2	PrincipalTest	Hay una DoubleLinkedList con varios elementos agregados de tipo ProductCleaning.
setUp3	PrincipalTest	Hay una DoubleLinkedList con varios elementos de tipo Product, ProductCleaning, ProductDrinks, ProductFoods.
setUp4	PrincipalTest	Hay una DoubleLinkedList destinada para las cuentas de acceso (Clase Account) esta lista doblemente enlazada se encuentra con una cuenta de acceso, con el username="gustavo" y una password="123".
setUp5	PrincipalTest	Hay una DoubleLinkedList destinada para cuentas de acceso, se encuentra vacía.

Objetivo de la Prueba: Verificar que el método addProductsInventory() de la clase Principal funcione de manera correcta cuando, 1) Esté vacía 2) Hayan elementos hijos de un mismo padre 3) Hayan elementos hijos de un padre.

Clase	Método	Escenario	Valores entrada	Resultado
Principal	addProductsInventory()	setUp1	Un objeto de la clase product con todos sus valores ingresados correctamente.	-Se ha agregado en el inicio de la lista doble enlazada el objeto que ha entrado como parámetro.
Principal	addProductsInventory()	setUp2	Un objeto de la clase ProductCleaning con todos los valores ingresados correctamente.	-Se ha agregado correctamente al inicio de la lista doblemente enlazada el objeto, y se han corrido una posición los demás elementos.
Principal	addProductsInventory()	setUp3	Un objeto de la clase ProductFoodsFruit con todos los valores correspondientes ingresados correctamente.	-El objeto de la clase ProductsFoodsFruit se ha agregado correctamente junto a los demás objetos de las clases ProductsDrinks e ProductsCleaning.

Objetivo de la Prueba: Verificar que el método SignUp() de la clase Principal funcione de manera correcta cuando, 1) Esté vacía 2) Hayan cuentas de acceso 3) El username esté registrado pero ingrese mal la contraseña.				
Clase	Método	Escenario	Valores entrada	Resultado
Principal	signUp()	setUp5	Un String que representará el username="gustavo" y otro String que representará la password="123"	-Se lanzará un error, ya que no se ha encontrado el username ingresado.
Principal	signUp()	setUp4	Un String que representará el username="gustavo" y otro String que representará la password="123"	-Se iniciará correctamente la sesión, y se abrirá la página principal del programa.
Principal	signUp()	setUp4	Un String que representará el username="gustavo" y otro String que representará la password="321"	-Aparecerá una alerta que notifique que la contraseña está errada.

Objetivo de la Prueba: Verificar que el método deleteProductsInventory() de la clase Principal funcione de manera correcta cuando, 1) Esté vacía 2) Hayan elementos				
Clase	Método	Escenario	Valores entrada	Resultado
Principal	deleteProductsClient()	SetUp5	Una variable de tipo int para representar el código del producto que se quiere eliminar, y otra variable de tipo int que representará la cantidad a eliminar.	-Se lanzará ProductDoesNotExist Exception , ya que no se encontró el producto solicitado.
Principal	deleteProductsClient()	setUp4	Una variable de tipo int para representar el código del producto que se quiere eliminar, y otra variable de tipo int que representará la cantidad a eliminar.	-Se eliminaran de la lista doblemente enlazada los productos indicados, y aparecerá una alerta notificando que la acción fue exitosa.

La gran mayoría de métodos son a partir de los métodos de la Lista Enlazada, por lo que si se cumplen estos, entonces se cumplen en gran parte los demás.