

MEMORIA ESCRITA TIENDA VIRTUAL

DAVID ANDRÉS CANO GONZÁLEZ

GARCÍA BLANDÓN DAVID

JUAN PABLO BUITRAGO DÍAZ

JULIAN ESTEBAN FERNÁNDEZ MONTOYA

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN

FACULTAD DE MINAS

PROGRAMACIÓN ORIENTADA A OBJETOS

MEDELLÍN

2020

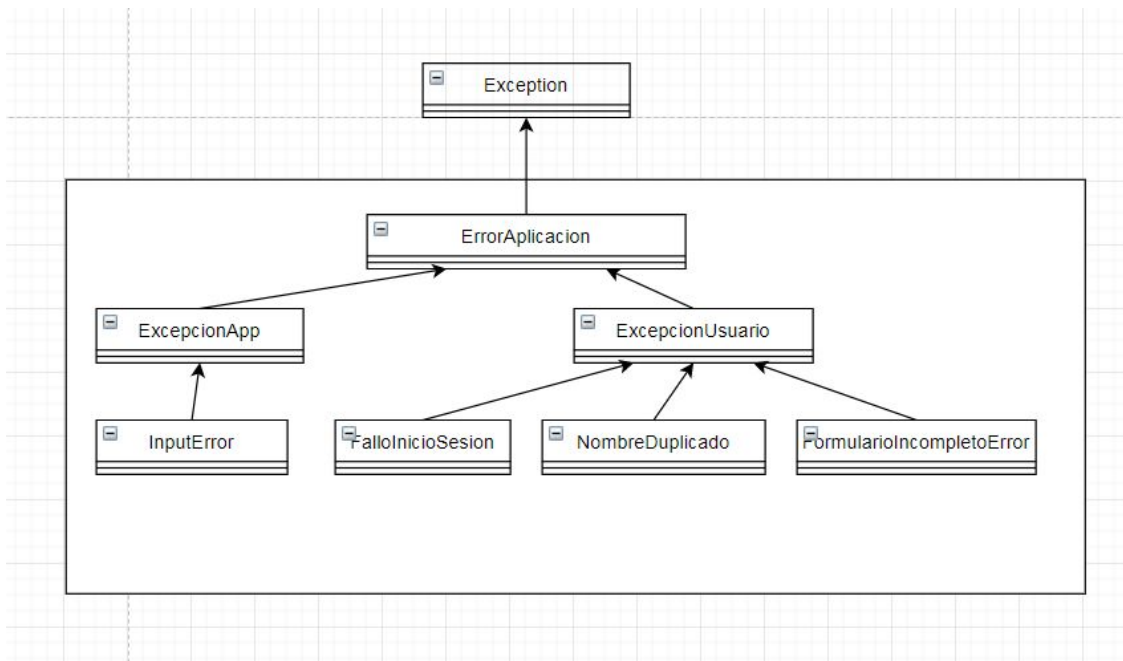
1. Descripción General

El análisis que hicimos estuvo enfocado al modelo de una tienda virtual, un sistema en el que un usuario pudiera buscar y ver productos, para posteriormente comprarlos y así mismo, debía existir un administrador con capacidad de manejar los productos y las opciones del usuario. Debido a que se necesita un usuario y una cuenta para ingresar en el sistema, también diseñamos una cuenta de invitado; quien podría ver las características generales del sistema, para luego acceder a ellas por medio de una cuenta de usuario.

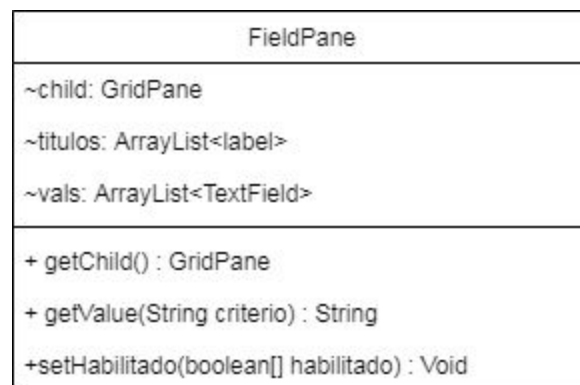
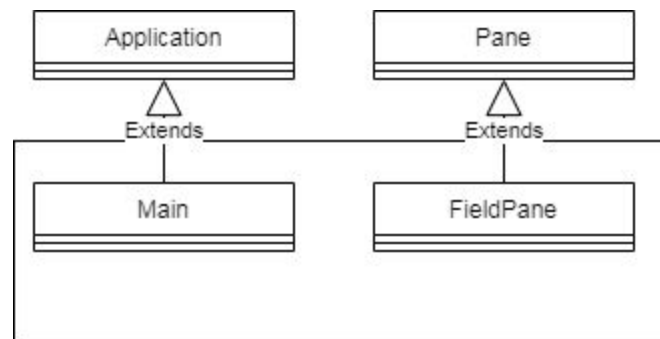
El sistema está basado en un par de inventarios individuales que representarán los productos que existen en la tienda y los que serán almacenados temporalmente por el usuario para ser comprados. Para asegurar el funcionamiento de estos, implementamos una serie de métodos de búsqueda y modificación de atributos que permitirían hallar los productos y cambiar sus características por las que fuesen pertinentes dependiendo de las acciones realizadas.

2. Diagrama de clases

Paquete Excepciones

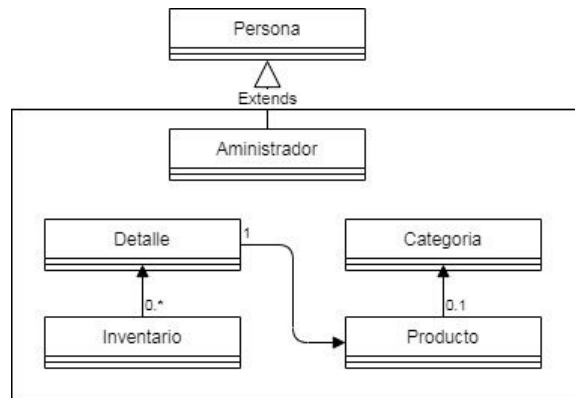


Paquete Ulmain



Main
<u>+usuarios: ArrayList<Persona></u> <u>+categorias: ArrayList<categoria></u> <u>+productos: ArrayList<productos></u> <u>+estadisticos: Estadistica</u> <u>+inventario: Inventario</u> <u>+nivel: int</u> <u>+usuario: Persona</u> <u>mainStage: Stage</u> <u>PrincipalAdministrador: BorderPane</u> <u>PrincipalInicial: GridPane</u> <u>PrincipalInvitado: BorderPane</u> <u>PrincipalUsuario: BorderPane</u> <u>SceneAdministrador: Scene</u> <u>SceneInicial: Scene</u> <u>SceneInvitado: Scene</u> <u>SceneUsuario: Scene</u>
<u>+main(): Void</u> <u>+Inicio(): Void</u> <u>+montarDB(): Void</u> <u>+menuAdministrador(): Menubar</u> <u>+menuInvitado(): Menubar</u> <u>+menuUsuario(): Menubar</u> <u>+MostrarCarro(): Void</u> <u>+mostrarCategorias(): Void</u> <u>+isNumeric(String s): Boolean</u> <u>+archivo(): Void</u> <u>+ayuda(): Void</u> <u>+BuscarCategoria(): Void</u> <u>+BuscarNombre(): Void</u> <u>+CargarDB(): void</u> <u>+Categoricas(int j): FieldPane</u> <u>+crearCategoria(): void</u> <u>+crearProducto(): void</u> <u>+finalizar(): void</u> <u>+inventateEsta(): FieldPane</u> <u>+mostrarInventario(): Void</u> <u>+mostrarProductos(): Void</u> <u>+productor(): FieldPane</u> <u>+Escenas(): void</u> <u>+start(Stage arg0): void</u>

Paquete Administrador



Administrador
<code>+busqueda(String nombreProducto) : int</code> <code>+busqueda(int indexCategoria) : ArrayList<Producto></code> <code>+crearProducto(String name, String description, double oPrice, double sPrice) : Void</code> <code>+crearCategoria(String name, String description) : Void</code> <code>+añadirDetalle(Producto obj1, int amount) : Void</code> <code>+ModificarNombreProducto(int index, String name) : Void</code> <code>+ModificarDescripcionProducto(int index, String description) : Void</code> <code>+addCategoriaProducto(int index, Categoria cat1) : Void</code> <code>+delCategoriaProducto(int index, Categoria cat1) : Void</code> <code>+modificarPrecioCompra(int index, Double price) : Void</code> <code>+modificarPrecioVenta(int index, Double price) : Void</code> <code>+resetCalificacion(int index) : Void</code> <code>+modificarNombreCategoria(int index,String name) : Void</code> <code>+modificarDescripcionCategoria(int index,String description) : Void</code> <code>+addProductoInventario(int index,Detalle detail) : Void</code> <code>+delProductoInventario(int index,Detalle detail) : Void</code> <code>+modificarProductoInventario(Detalle detail,int amount) : Void</code> <code>+modificarCantidadProducto(int index, int amount) : Void</code> <code>+eliminarProducto(int i) : Void</code> <code>+eliminarCategoria(int i) : Void</code> <code>+eliminarDetalle(int i) : Void</code> <code>+verEstadisticas() : Void</code> <code>+CrearUsuario(String nombre,boolean genero, int edad, String usuario,String contraseña,String pregunta, String respuesta) : Void</code> <code>+toString() : String</code> <code>+mostrarProductos() : String</code>

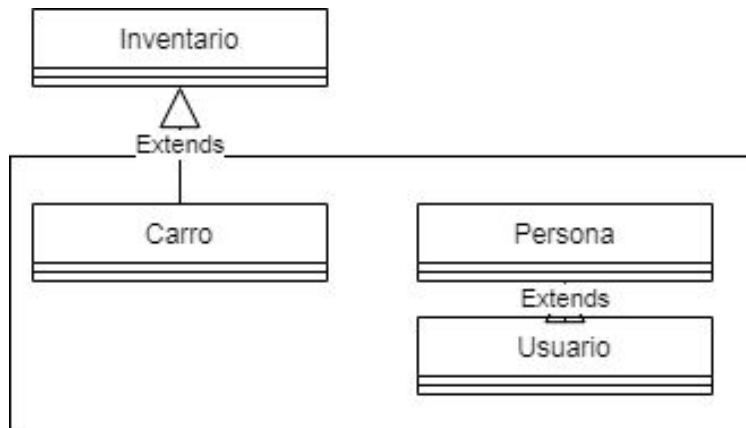
Detalle
-producto : Producto -cantidad : Int
+ getProducto() : Producto + getCantidad() : Int ~ setProducto() : Void ~ setCantidad() : Void + restarCantidad(int amount) : Void + toString() : String + toString2() : String

<i>Categoria</i>
-nombre : String -descripción : String
+GetDescripcion() : String ~SetDescripcion() : void +GetNombre() : String ~SetNombre() : void +toString() : String

<i>Producto</i>
-nombre : String -descripcion : String -categorias : ArrayList<Categoria> -precioVenta : double -precioCompra : double
+GetNombre() : String +GetDescripcion() : String +GetCategoria() : ArrayList<Categoria> +GetCategorialndividual() : Categoria ~GetPrecioCompra() : Int +GetPrecioVenta() : Int +GetCalificacion() : Float ~SetNombre(String s) : Void ~SetDescripcion(String s) : Void ~SetPrecioVenta(double i) : Void ~SetPrecioCompra(double i) : Void +toString() : String +toString2() : String

<i>Inventario</i>
Protected Inventario : ArrayList<Detalle>
+AddInventario(Detalle d) : void +AddInventario(Detalle d,int i) : Void +DellInventario(Detalle d) : Void +DellInventario(int i) : Void +RealizarBusqueda(String n) : int +RealizarBusqueda(int indexCategoria) : ArrayList<Producto> +toString() : String +toString2() : String +getInventario(): ArrayList<Detalle> +getInventario(int i): Detalle

Paquete Usuario



Usuario
-saldo : Double -carroCompra : Carro
+getSaldo() : Double +reducirSaldo(double valorAPagar) : void +actualizarSaldo(double aumento) : void +getSubtotal() : Double +getCarroToString() : String +getCarro() : Carro +reducirSaldo(double valorAPagar) : void +añadirProducto(int indexInventario) : void +eliminarProducto(int indexCarro) : void +busqueda(String nombreProducto) : int +busqueda(int indexCategoria) : ArrayList<Producto> +comprar() : void +mostrarProductos() : String +toString() : String

<i>Carro</i>
-numObjetos : int
-subTotal: double
+GetSubTotal() : Double
+getNumObjetos() : int
~AddProducto(Detalle d) : Void
~AddProducto(Detalle d,int i) : Void
~DelProducto(Detalle d) : Void
~DelProducto(int i) : Void
+actualizar() : Void
+GenerarFactura() : String
+toString() : String
+getNumObjetos(): int

<i>Persona</i>
-nombre : String
-genero : Boolean
-edad : Int
-usuario : String
-contrasena : String
-preguntaRecuperacion: String
-respuestaRecuperacion: String
+GetNombre() : String
+GetGenero() : Boolean
+GetEdad() : Int
+GetUsuario() : String
+GetContrasena() : String
-SetNombre() : Void
-SetUsuario() : Void
-SetContrasena() : Void
-SetGenero() : Byte
-SetEdad() : Int
+ComprobarContrasena(String contraseña) : Boolean
+ComprobarRespuesta(String nombre) : Boolean
+RecuperarContraseña(String respuesta, String contraseñaNueva) : Void
+static mostrarCategorias(): String
+abstract mostrarProductos(): String
+abstract busqueda(): int
+abstract busqueda(int indexCategoria): int

3. Manual de Usuario y Funcionalidades

IMPORTANTE: Antes de verificar las funcionalidades es importantes conocer lo siguiente:

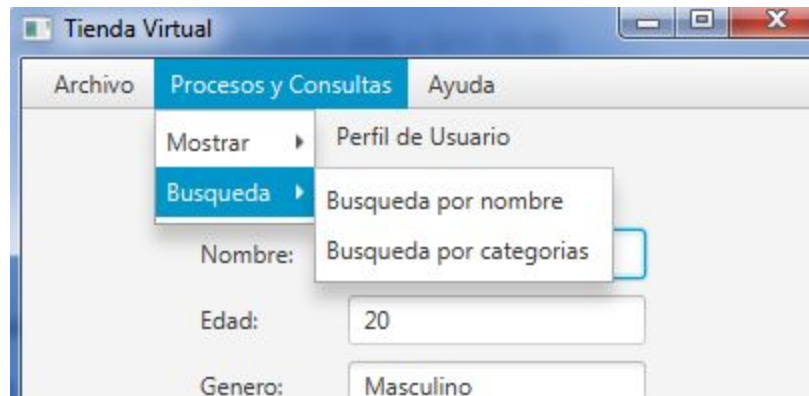
Administrador usuario: root **Respuesta de recuperación:** root

contraseña: root

Usuario usuario: user **Respuesta de recuperación:** user

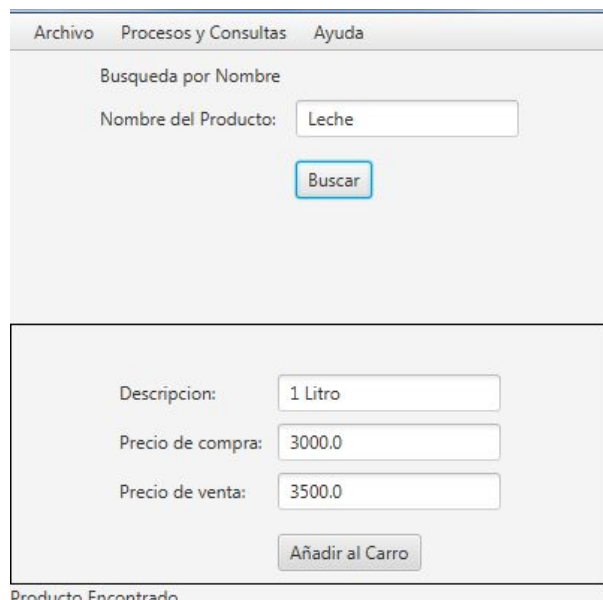
contraseña: user

***Búsqueda:** Mediante esta función podrá buscar los productos disponibles de la tienda. Primero, deberá estar dentro de la tienda, ya sea como usuario, invitado o administrador.

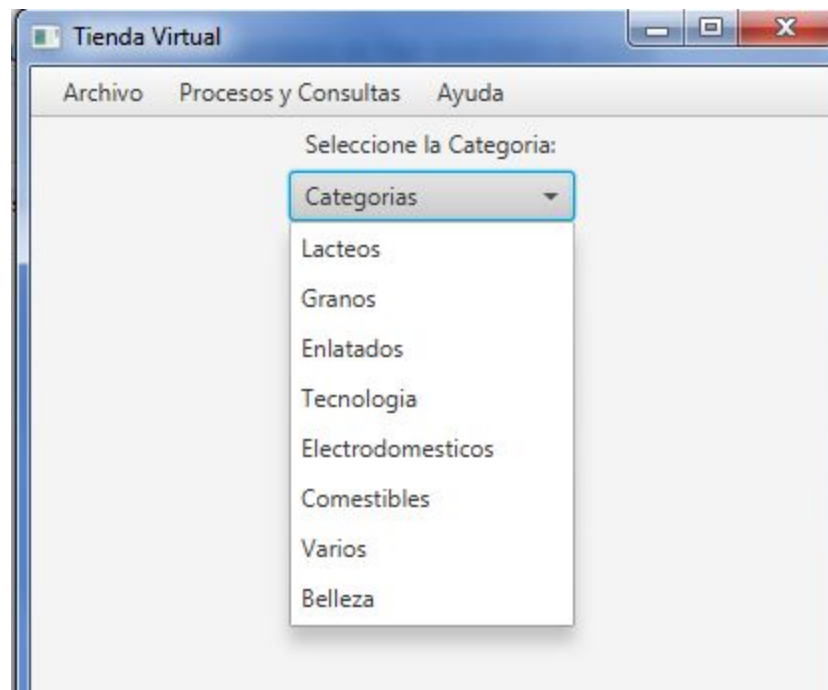


Al dar click en “Búsqueda”, se desplegarán 2 opciones: “Búsqueda por nombre” o “Búsqueda por categorías”.

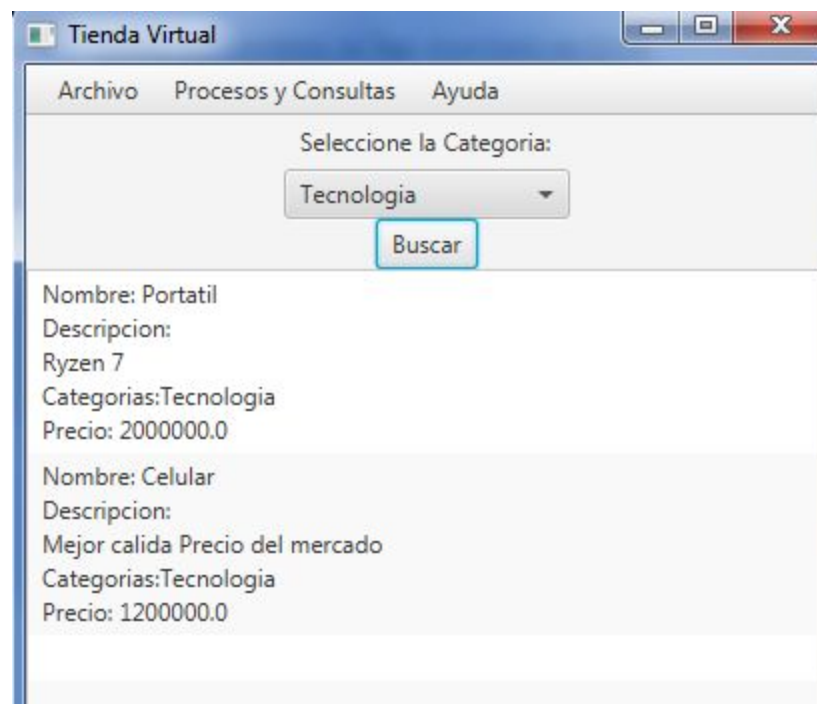
Al **buscar por nombre**, aparecerá un espacio donde podrá poner un texto, y si el texto coincide con algo de los nombres de los productos en la tienda, este aparecerá.



Al **buscar por categoría**, aparecerá una opción desplegable donde se encontrarán todas las categorías disponibles.

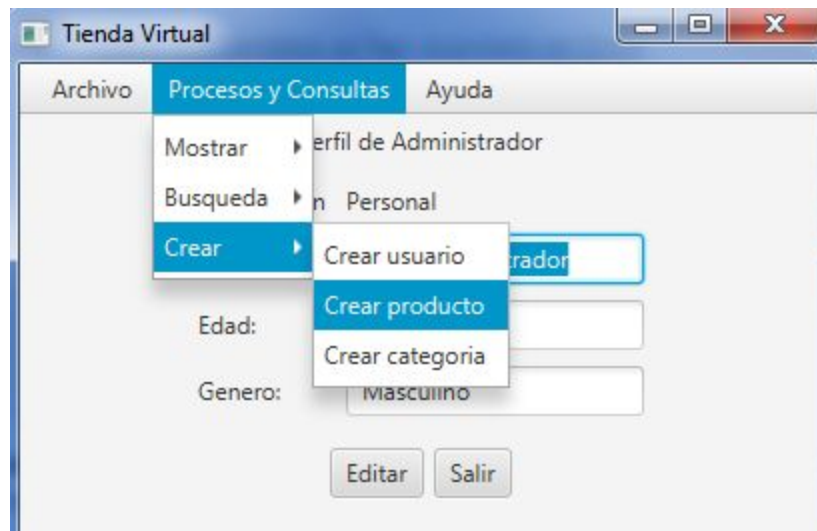


al seleccionar una categoría, y pulsando el botón buscar, aparecerán todos los productos que encajan en la categoría seleccionada.



***Gestión de productos:** Al ingresar a la tienda como usuario, tendrá la opción de crear y editar productos de la tienda.

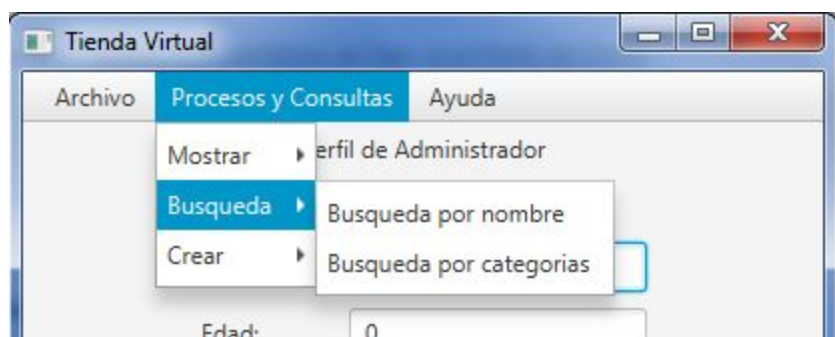
Para **crear un producto** clickee la opción “Procesos y consultas”, luego “Crear”, y posteriormente, “Crear producto”.



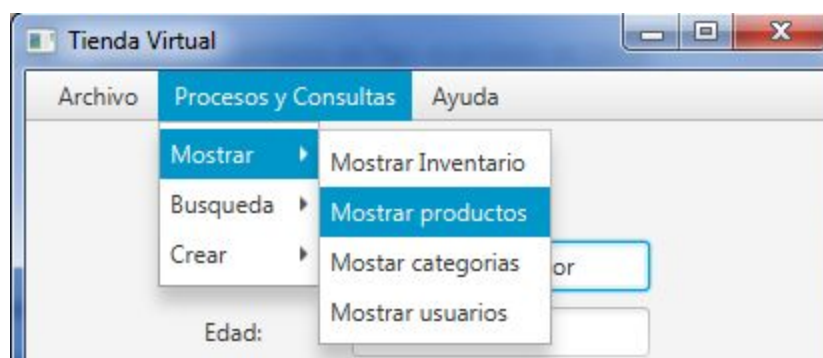
Se abrirá un formulario donde podrá rellenar la información de producto en cuestión, finalmente, para crearlo, clickee “Crear”.

A screenshot of the 'Crear producto' form within the 'Tienda Virtual' application. The form has a title bar 'Tienda Virtual' and a menu bar with 'Archivo', 'Procesos y Consultas', and 'Ayuda'. The main content area is titled 'Crear producto'. It contains several input fields: 'Nombre' (with 'Peras' entered), 'Descripcion' (with 'De un olmo' entered), 'Precio original' (with '50' entered), 'Precio de venta' (with '100' entered), 'Categoria' (a dropdown menu with 'Comestibles' selected), and 'Cantidad' (with '10' entered). At the bottom, there are two buttons: 'Salir' and 'Crear'.

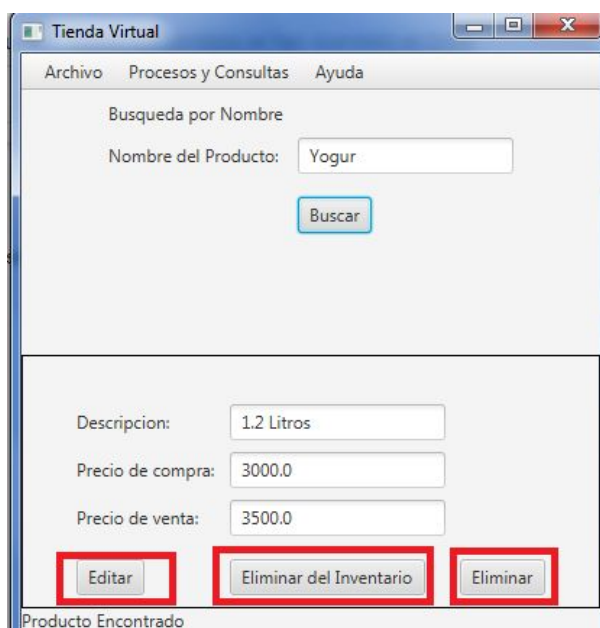
Para **editar un producto**, primero deberá buscarlo; ya sea, por “Búsqueda”.



ó por “mostrar productos”.

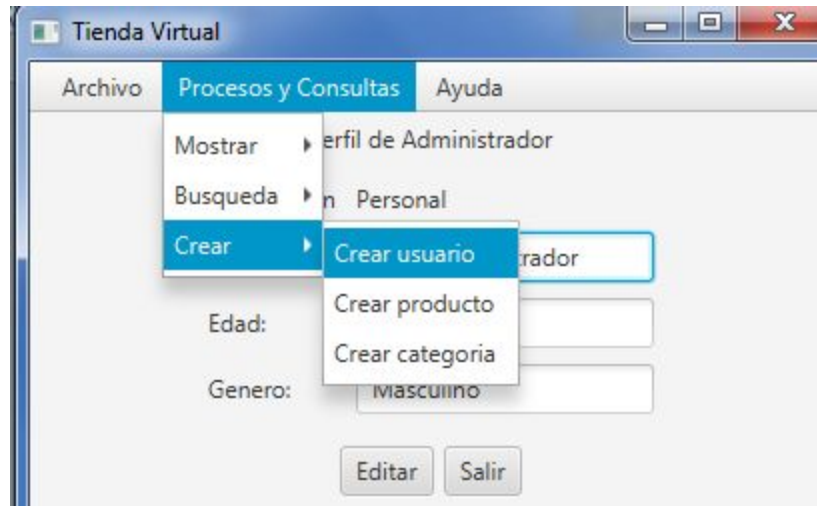


Una vez encuentre el producto, tendrá 3 opciones: “Editar”, que le permitirá editar los precios y la descripción del producto. “Eliminar del inventario”, que borrará las existencias de ese producto en el inventario. Y “Eliminar”, que borrará el producto como tal del inventario.

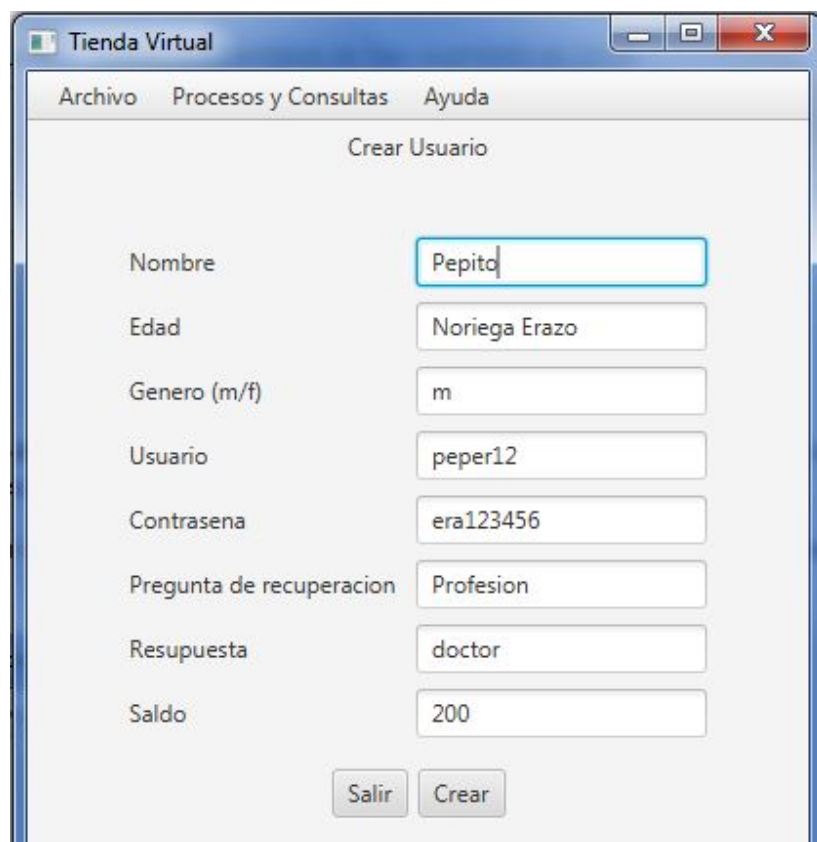


***Gestión de Usuarios:** Al entrar a la tienda como administrador, usted podrá crear y eliminar usuarios.

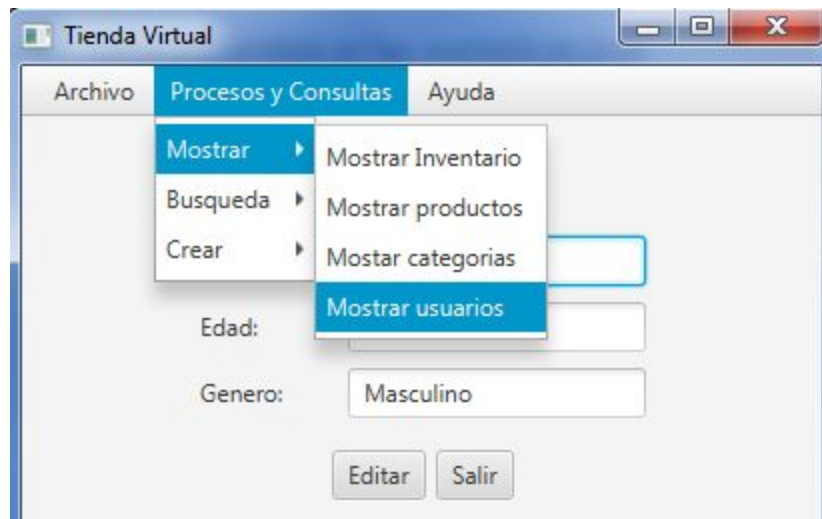
Para **crear un usuario** diríjase a “Procesos y consultas”, luego a “Crear”, y posteriormente a “Crear usuario”.



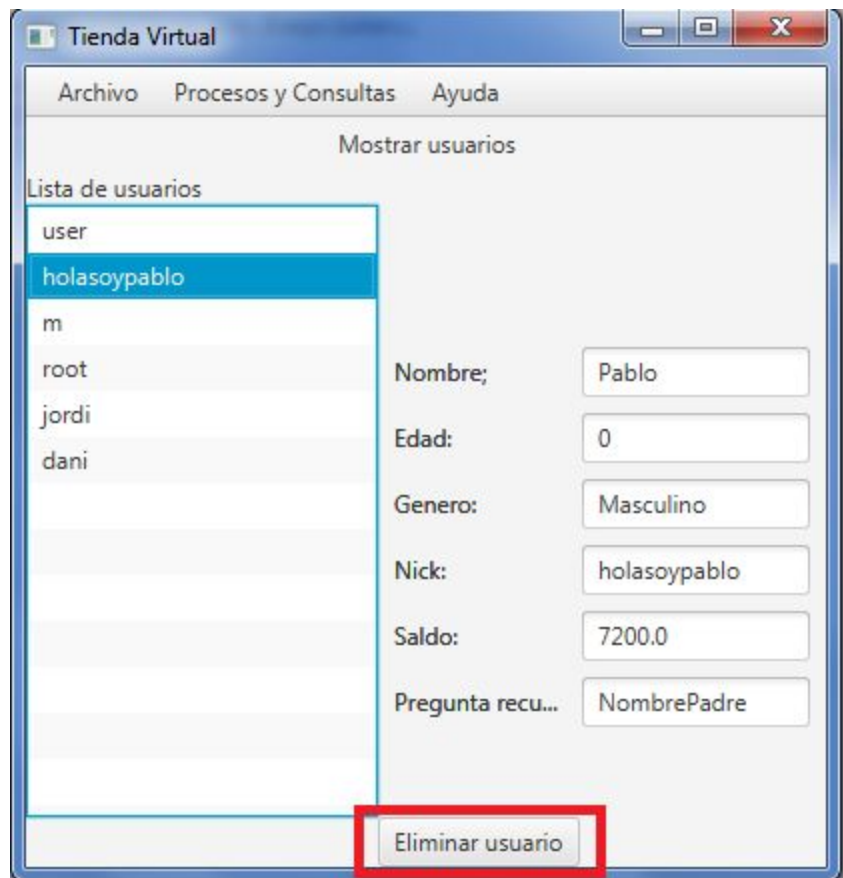
Le aparecerá un formulario donde podrá introducir la información del nuevo usuario. Finalmente, para crear el nuevo usuario presione “Crear”.

A screenshot of the 'Tienda Virtual' application window showing the 'Crear Usuario' form. The menu bar is the same as the previous image. The form contains the following fields and values: 'Nombre' (Pepito), 'Edad' (Noriega Erazo), 'Genero (m/f)' (m), 'Usuario' (peper12), 'Contraseña' (era123456), 'Pregunta de recuperacion' (Profesion), 'Resupuesta' (doctor), and 'Saldo' (200). At the bottom of the form are two buttons: 'Salir' and 'Crear'.

Para **eliminar un usuario**, primero diríjase a “Procesos y consultas”, luego a “Mostrar”, y finalmente a “Mostrar usuarios”.

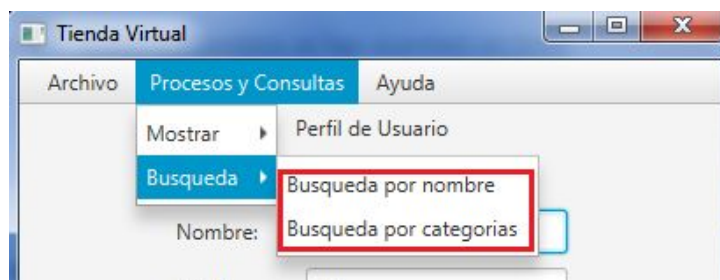


Luego, seleccione el usuario que desea eliminar y pulse “Eliminar usuario”.

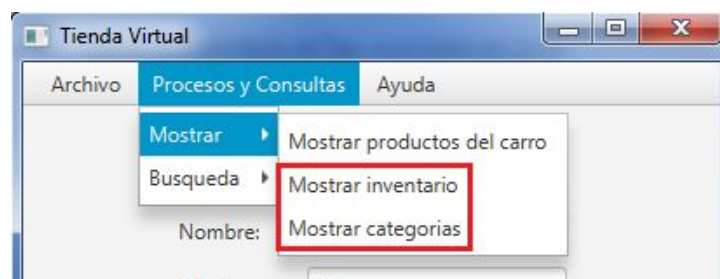


***Comprar productos:** Como usuario, tendrá la opción de comprar objetos de la tienda.

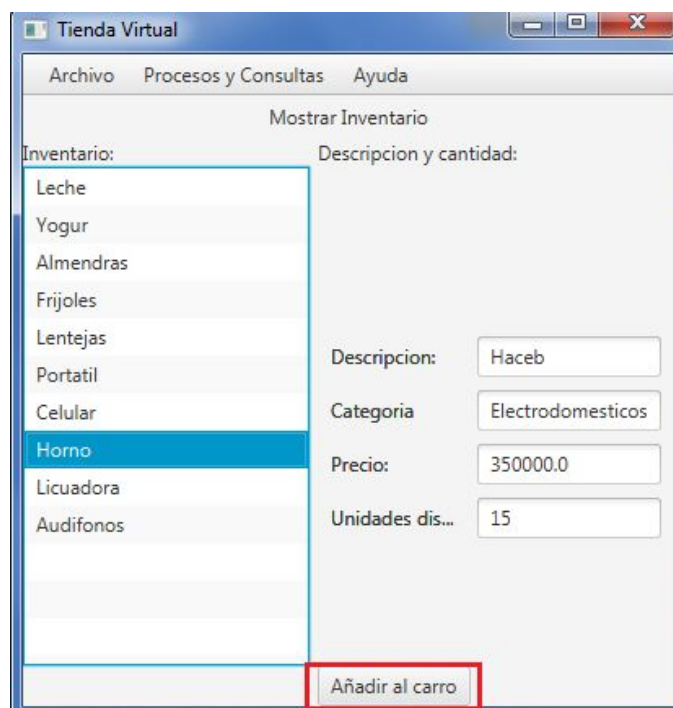
Primero, deberá agregar productos a su carrito de compras. para eso deberá buscarlos por nombres o categorías:



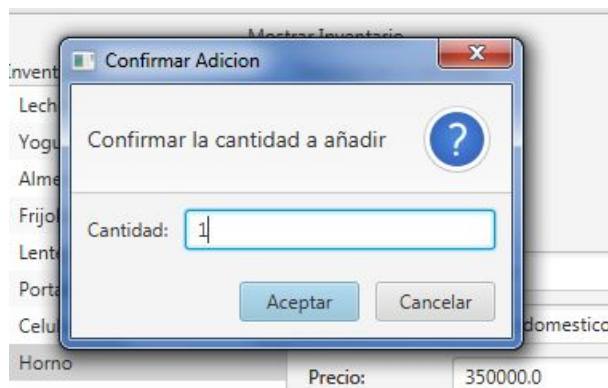
o mostrando todo el inventario, ó filtrándolo por categorías.



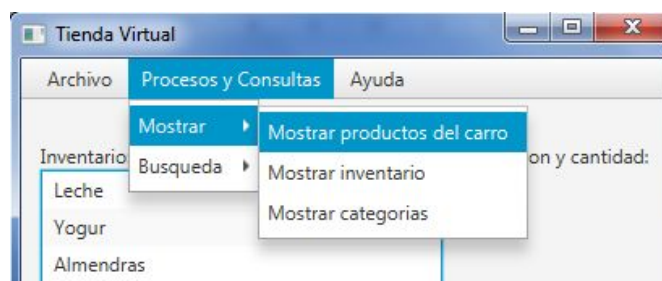
deberá seleccionar un producto, y posteriormente hacer clic en “añadir al carrito”.



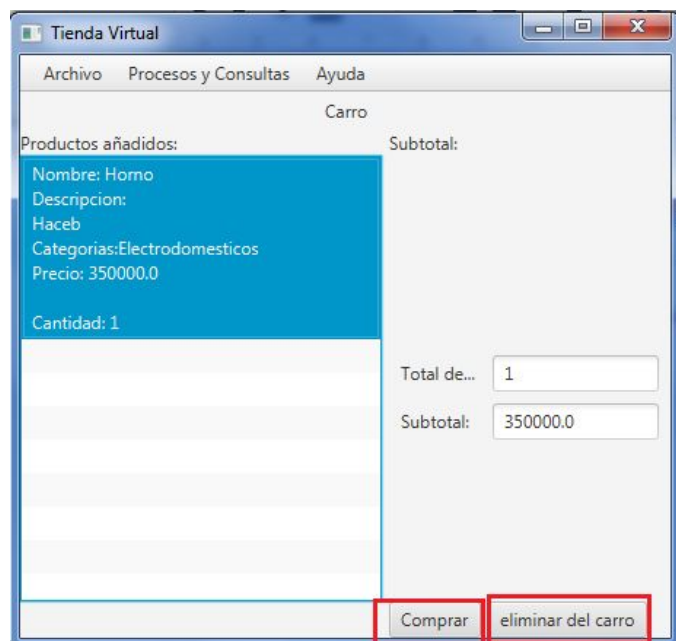
se le pedirá que ingrese la cantidad del producto y luego dé clic en “Aceptar”.



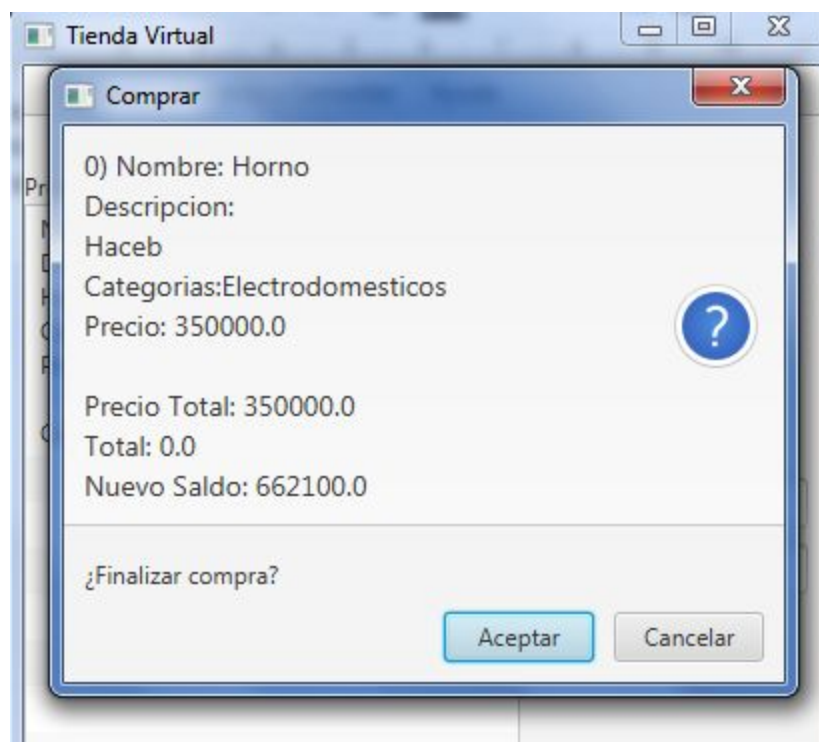
Usted podrá ver lo que se encuentra en el carrito al ir a “Procesos y consultas”, “Mostrar”, y “Mostrar productos del carro”



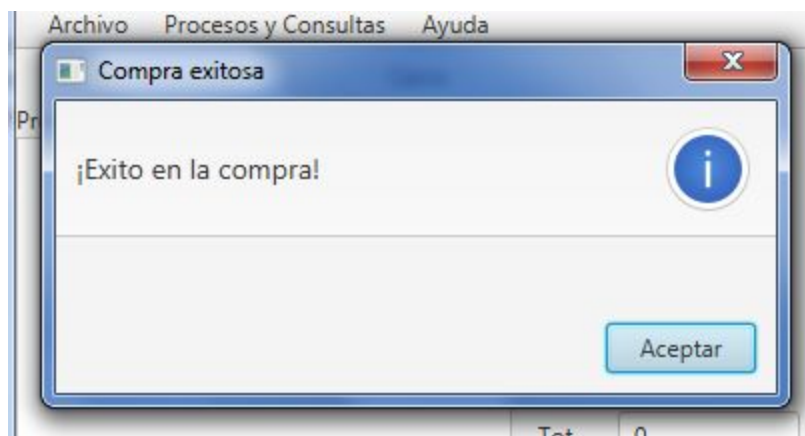
Una vez en el carrito, podrá ver 2 opciones: “Comprar” y “Eliminar del carrito”. la última le permitirá eliminar ese producto del inventario. Y si decide comprar lo que está en el carrito entonces pulse “Comprar”.



Al pulsar “comprar”, le aparecerá la factura de compra.

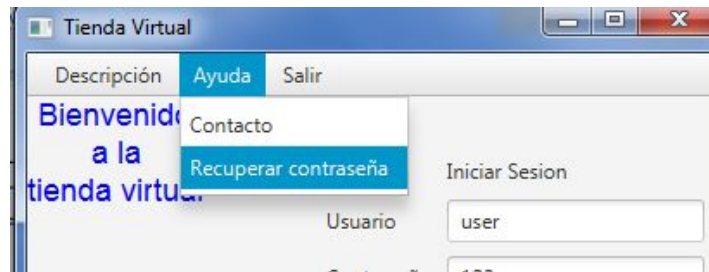


Al aceptar le aparecerá un mensaje de confirmación.

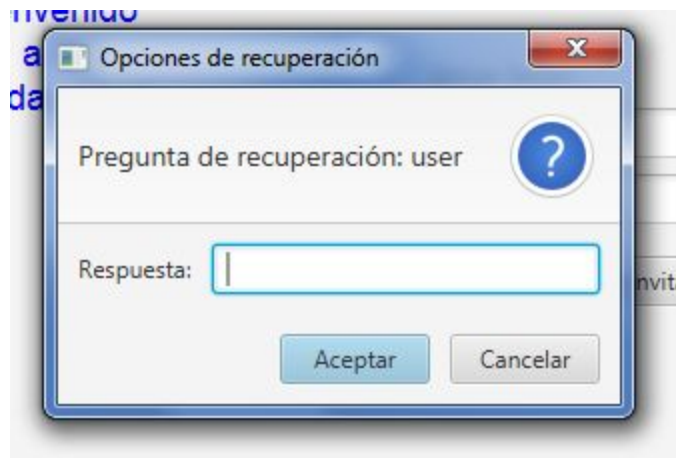


***Recuperar contraseña:** Si en algún momento se llega a olvidar la contraseña, puede recuperarla respondiendo a una pregunta de seguridad.

Primero vamos a “Ayuda”, luego a “Recuperar contraseña”.



Deberá ingresar su nombre de usuario.



deberá responder a la pregunta. Si lo hace correctamente, entonces le pedirá que ingrese una nueva contraseña.

