

Convenciones de programación

- Clases

Cedula	Clase que permitirá verificar si la cedula del usuario registrado es correcta o no.
Factura	Implementa métodos para realizar los cálculos para mostrar el valor a pagar del usuario en una factura con sus respectivos datos.
Jarabe	Clase que hereda de Medicina que permite calcular valores a pagar por medicamentos que son exclusivamente del tipo jarabe.
ListaProducto	Clase que almacena los productos que tenemos a la disposición en la farmacia.
Medicina	Clase abstracta que hereda de Producto y separa los productos en jarabes o pastillas.
Pastilla	Clase que hereda Medina que permite calcular valores a pagar por medicamentos que son exclusivamente del tipo pastilla.
Persona	Clase que contiene todos los atributos que puede tener un usuario.
PoliMed	Clase que representa la farmacia.
Producto	Clase abstracta que contiene todos los atributos de los productos de la farmacia.
TextPrompt	Clase para hacer un Placeholder.
Usuario	Subclase que tiene como clase principal o clase padre a Persona.
LeerArchivos	Clase con métodos para validar usuarios.

- Interfaces

GUI_Login	Clase que contiene métodos de la clase LeerArchivo y valida un usuario registrado previamente.
GUI_Registro	Clase que contiene métodos para guardar usuarios en archivos .txt para luego verificarlos en la cla GUI_Login.
GUI Producto	Clase que contiene métodos para cargar y muestra de Datos de los productos.
GUI Factura	Clase que contiene métodos para mostrar la factura de compra de productos y datos del usuario registrado.

- Métodos

mostrarFecha();	Método void que toma la fecha y hora del ordenador donde se esté ejecutando la aplicación.
calcularTotalPagar();	Método void que se encarga de calcular el valor a pagar por las compras.
toString();	Método String que muestra los datos del usuario que realizo la compra y valores a pagar por las compras.
esValida(String cedulaNumero);	Método boolean que se encarga de verificar que el número de cedula sea válido.
Persona(String nombre, String Apellido, String cedula, Int Edad);	Método que almacena cada una de las variables de la clase Persona.
habilitarBotonoes();	Método void que permite habilitar los botones si y solo si se han

	llenado todos los campos correspondientes.
esCorreoValido(String correo);	Método boolean que se encarga de verificar que el correo electrónico ingresado sea válido.
Medicina(String nombre, String Apellido, String cedula, Int Edad);	Método que almacena los síntomas que posee la persona que después se despliegan los métodos de pastillas o jarabes
iniciarProductos();	Método en el cual se guardan todos los medicamentos.
Producto(String codigoProducto, String nombreProducto, double precioProducto);	Método que almacena cada una de las variables de la clase Producto.
Pastillas(String codigoProducto, String nombre, String síntomas, double precioProducto);	Método que almacena toda la información sobre el producto pastillas.
Jarabe(String codigoProducto, String nombre, String síntomas, double precioProducto);	Método que almacena toda la información sobre el producto Jarabes.
userData();	Método void para extraer el correo y la clave y validar con usuario ya registrado
crearUsuario();	Método void para crear archivo .txt y guardar datos del usuario nuevo.
ListaUsuarios();	Método constructor crea una lista de usuarios.
ListaProductos();	Método en el cual se añaden los productos o medicamentos.
añadirUsuario(Usuario usuario);	Método void para añadir usuario a

	un arreglo de usuarios.
validarUsuario(String usuarios[], String user, String pass);	Método boolean que retornara un verdadero o falso si es usuario logeado existe en el registro o no.
buscarUsuario(String correo);	Método int que devuelve un índice de la posición del usuario encontrado en el arreglo.
calcularPrecio(int cantidadProducto);	Método double para calcular el precio a pagar por un solo medicamento.
añadirProducto(Producto producto);	Método void para añadir productos a la farmacia.
añadirProductoComprado(Producto producto, int cantidad);	Método int para añadir y devolver la cantidad total de los productos comprados
cargarDatos();	Método que muestra los datos almacenados en el archivo .txt.
limpiarElementos();	Elimina todas las estancias completadas en la GUI_Registro
Usuario(String nombre, String apellido, String cedula, String edad, Int edad, String contraseña);	Método que guarda todas las variables de la clase Usuario
TextPrompt(text: String, component: JTextComponent)	Método para nombrar cada campo que el usuario debe llenar
habilitarBotones()	Habilita los botones de la interfaz registro, cuando se completan todos los campos del GUI_registro.

- Variables

int anio;	Variable de instancia publica en la clase Factura que guarda el valor
-----------	---

	del año establecido en el dispositivo donde se ejecuta la aplicación.
int dia;	Variable de instancia publica en la clase Factura que guarda el valor del día establecido en el dispositivo donde se ejecuta la aplicación.
int mes;	Variable de instancia publica en la clase Factura que guarda el valor del mes establecido en el dispositivo donde se ejecuta la aplicación.
int hora;	Variable de instancia publica en la clase Factura que guarda el valor de la hora establecido en el dispositivo donde se ejecuta la aplicación.
int minuto;	Variable de instancia publica en la clase Factura que guarda el valor de los minutos establecidos en el dispositivo donde se ejecuta la aplicación.
int segundos	Variable de instancia publica en la clase Factura que guarda el valor de los segundos establecidos en el dispositivo donde se ejecuta la aplicación.
double total	Variable de instancia publica en la clase Factura que guarda el precio final de los productos que el usuario compro.
String nombre;	Variable de instancia estática final en la clase Persona para

	almacenar el nombre del usuario registrado
String apellido;	Variable de instancia estática final en la clase Persona para almacenar el apellido del usuario registrado
String direccion;	Variable de instancia estática final en la clase Persona para almacenar la dirección del usuario registrado
String cedula;	Variable de instancia estática final en la clase Persona para almacenar la cédula del usuario registrado
int edad;	Variable de instancia estática final en la clase Persona para almacenar la edad del usuario registrado
ArrayList<Producto> productosFarmacia;	Arreglo que contiene todos los productos de la farmacia disponibles para comprar.
ArrayList<Producto> productosComprados;	Arreglo que contiene todos los productos comprados.
ArrayList<Object []>productosVector;	Arreglo que contiene todos los tipos de productos disponibles (pastillas, jarabes)..
String correo;	Variable de instancia estática de la clase Usuario la cual hereda de Persona y almacena el correo del usuario registrado.
String contraseña;	Variable de instancia estática de la clase Usuario la cual hereda de

	Persona y almacena la contraseña del usuario registrado.
String codigoProducto;	Variable de instancia privada de la clase abstracta Producto, almacena el código del Producto
String nombreProducto;	Variable de instancia privada de la clase abstracta Producto, almacena el nombre del Producto
double precioProducto;	Variable de instancia privada de la clase abstracta Producto, almacena el precio del Producto
ArrayList<Producto> listaProductos;	Arreglo que contiene y almacena la lista de los productos disponibles.
String sintoma;	Variable de instancia privada de la clase Medicina la cual hereda de Producto y almacena el sintoma del usuario registrado.
Text Field txtUsuario;	Variable de instancia privada de la interfaz GUI_Login la cual almacena el usuario.
Password Field pssClave;	Variable de instancia privada de la interfaz GUI_Login la cual almacena la contraseña del usuario.
Button btnIngresar;	Variable de instancia privada de la interfaz GUI_Login realiza la acción de ingresar al aplicativo.
Button btnRegistro;	Variable de instancia privada de la interfaz GUI_Login realiza la acción de registrarse e ingresar al aplicativo.

Text Field txtNombre;	Variable de instancia privada de la interfaz GUI_Registro en la cual se ingresa el nombre del usuario.
Text Field txtApellido;	Variable de instancia privada de la interfaz GUI_Registro en la cual se ingresa el Apellido del usuario.
Text Field txtCorreo;	Variable de instancia privada de la interfaz GUI_Registro en la cual se ingresa el correo del usuario.
Text Field txtCedula	Variable de instancia privada de la interfaz GUI_Registro en la cual se ingresa la cedula del usuario.
Text Field txtEdad;	Variable de instancia privada de la interfaz GUI_Registro en la cual se ingresa el edad del usuario.
Button btnGuardar;	Variable de instancia privada de la interfaz GUI_Registro la cual permite guardar los datos del usuario..
Button btnCancelar;	Variable de instancia privada de la interfaz GUI_Registro la cual permite cancelar el registro del usuario.
Button btnRegresar;	Variable de instancia privada de la interfaz GUI_Registro la cual permite regresar a la primera interfaz de GUI_Login