Instituto Tecnológico de Costa Rica

Proyecto #1

Programación Orientada a Objetos (POO)

Profesor: Cristian Campos Agüero

Integrantes:

* Juan David Mora Acuña / 2024126028
* Noé López Durón / 2024234500
* Steven Daniel Umaña López / 2024253314

Verano 2024

Índice

[Manual de usuario 3](#_Toc187028079)

[Pruebas de funcionalidad 29](#_Toc187028080)

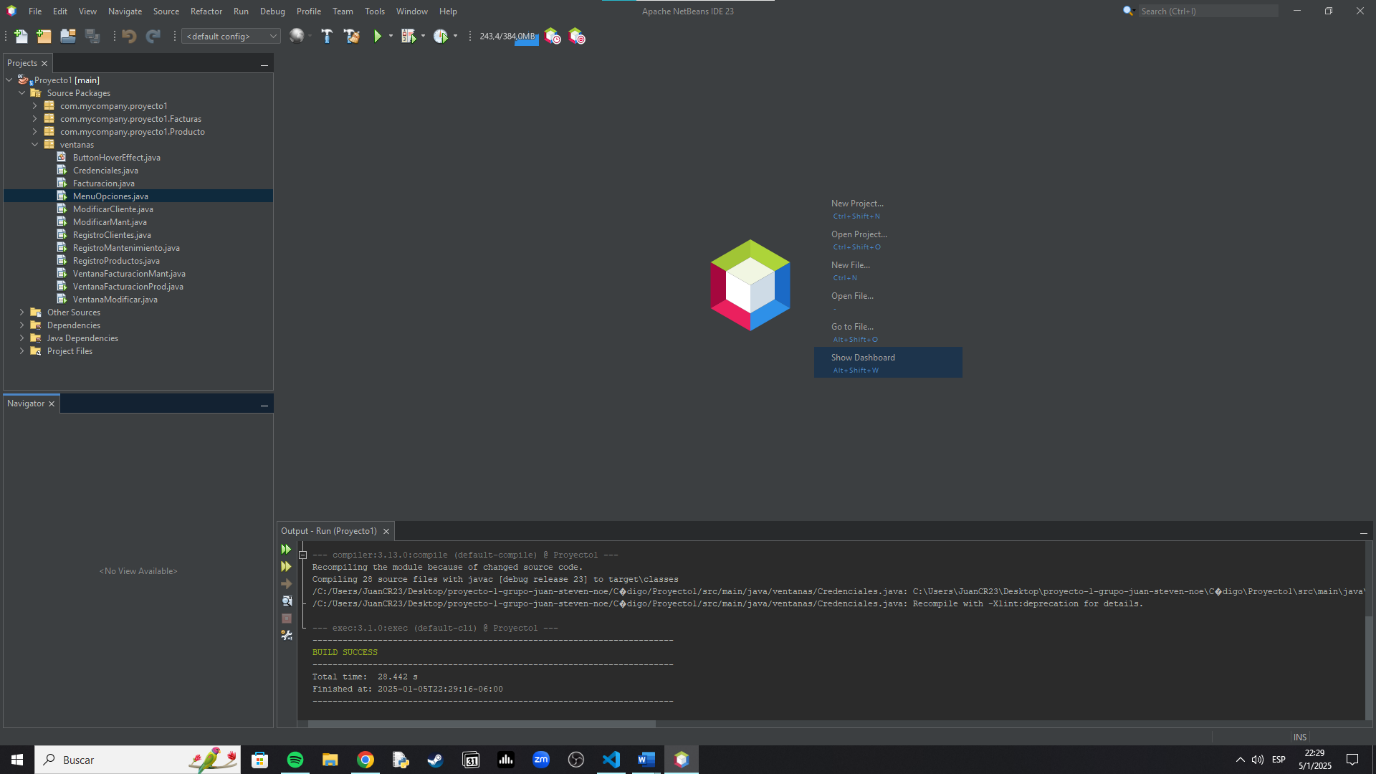
[Descripción del problema 30](#_Toc187028081)

[Diseño del programa 31](#_Toc187028082)

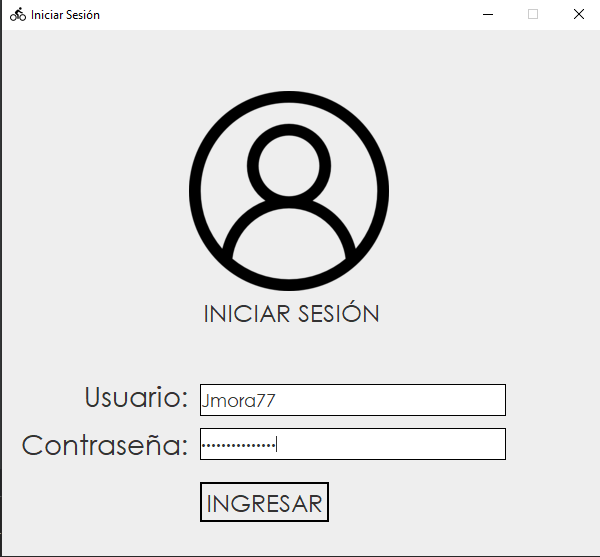
[Librerias usadas 35](#_Toc187028083)

[Análisis de resultados 36](#_Toc187028084)

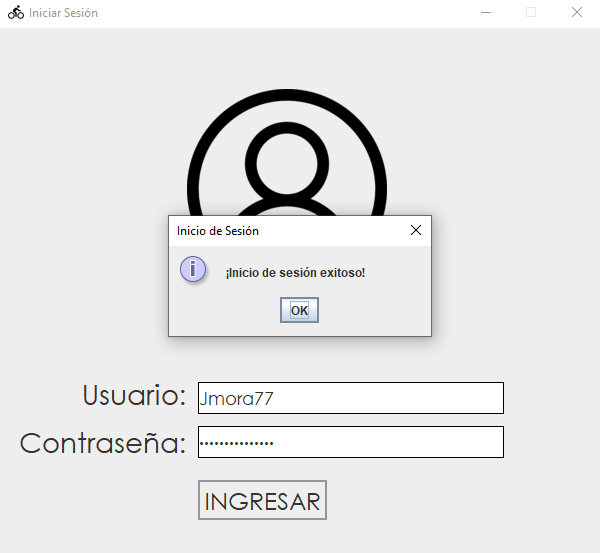
# Manual de usuario



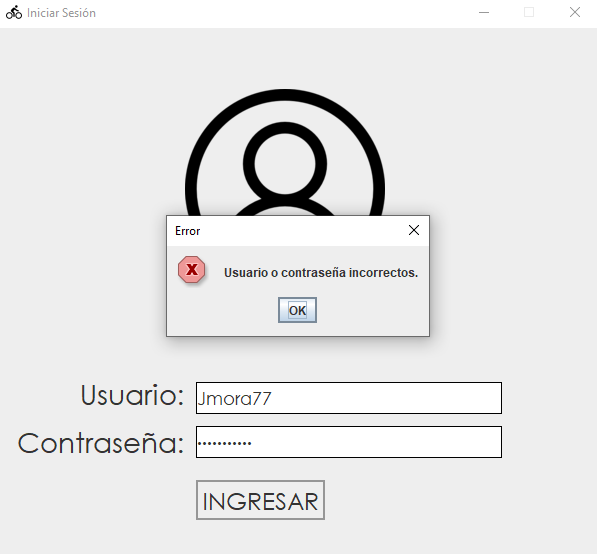
Se inicia el programa dándole click al botón run o de inicio ubicado en la parte superior de la pantalla.



Posteriormente se le desplegará al usuario una ventana de inicio de sesión donde deberá colocar un usuario y contraseña validados para el inicio de sesión.



Se muestra un mensaje de inicio de sesión aceptado. En caso de ingresar datos incorrectos se mostrará de la siguiente manera.

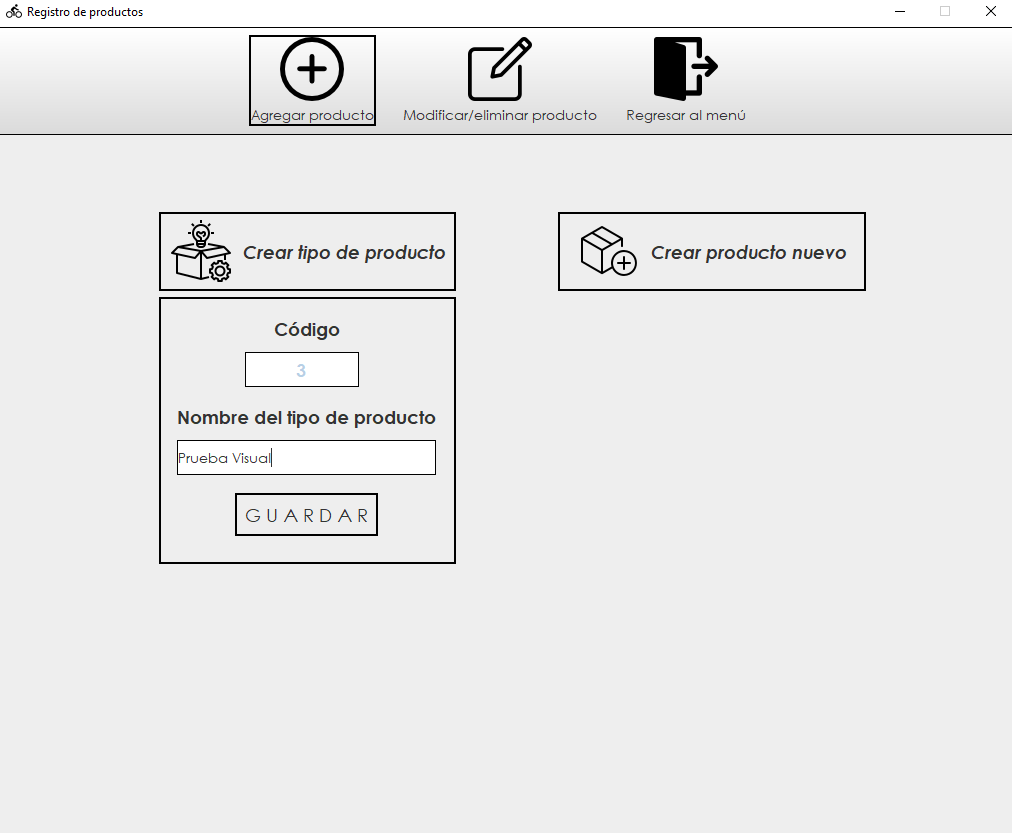




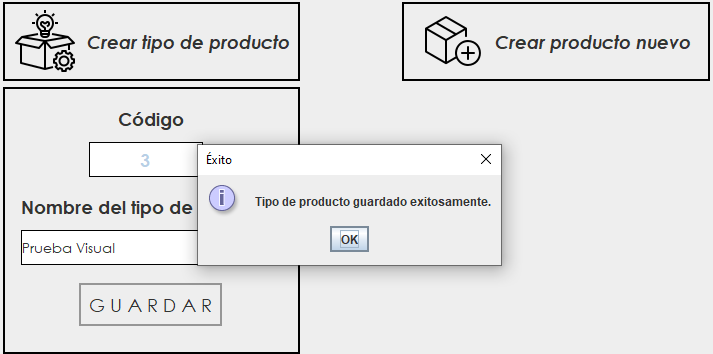
Luego de un inicio de sesión exitoso se le mostrará al usuario la ventana del menú principal del programa, donde se observarán distintas opciones de administración (Productos, Clientes, Mantenimientos, Facturación). También se muestra un logo en la parte superior izquierda donde el usuario puede cerrar la sesión.



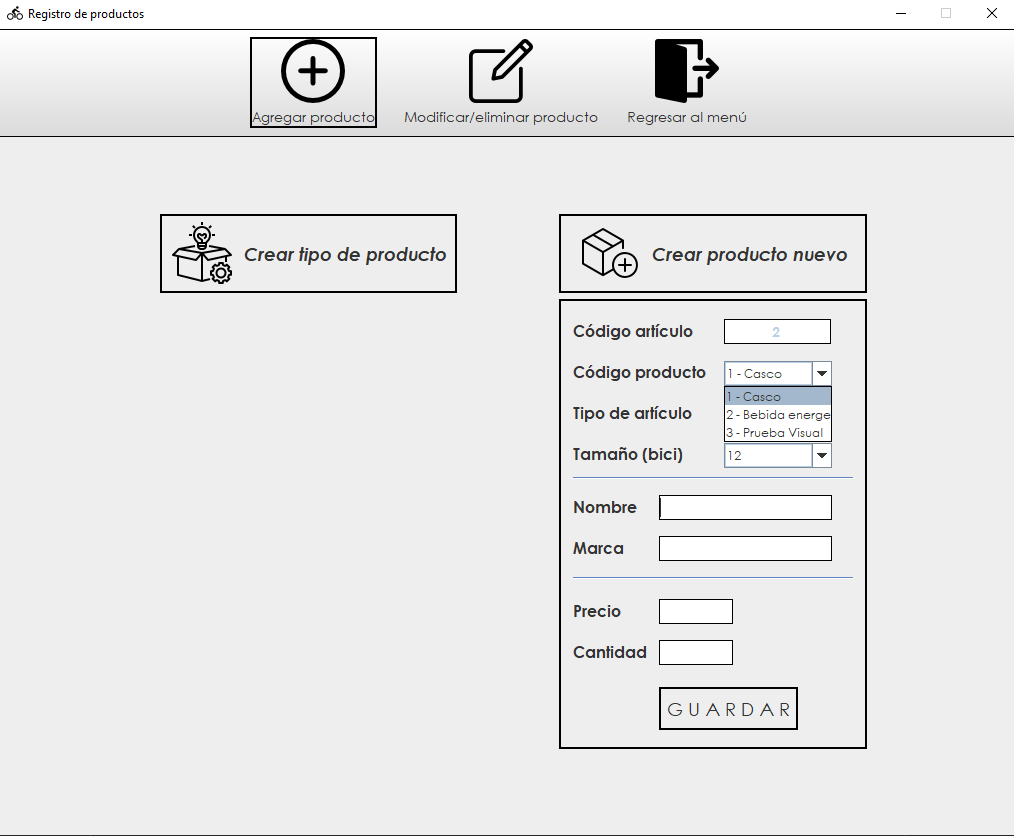
Al seleccionar el módulo de “Registro de productos” se desplegará esta ventana con mensaje de bienvenida al usuario. En la parte superior se presentan 3 botones (Añadir producto, Modificar o eliminar producto y regresar al menú principal).



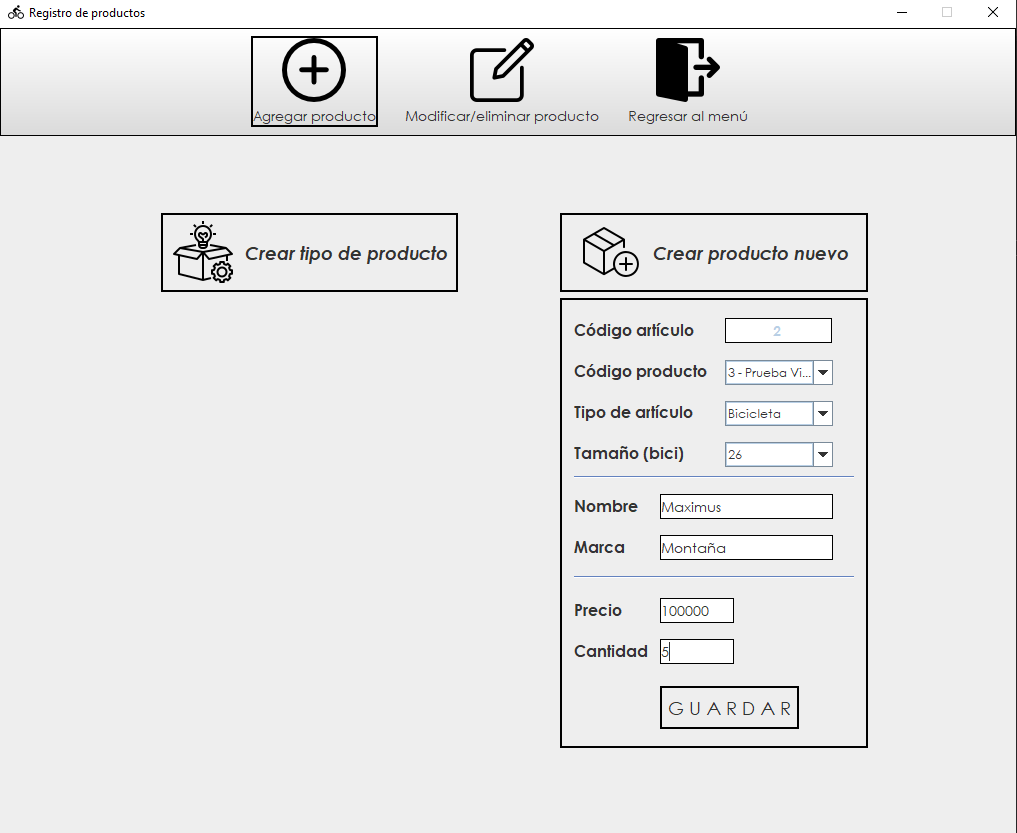
Al seleccionar el botón de “Añadir producto” se desplegarán dos botones adicionales que le permitirán al usuario crear un tipo de producto o crear un producto nuevo. Al seleccionar “Crear tipo de producto” el usuario podrá crear un nuevo tipo que le permitirá crear productos de ese tipo posteriores a su creación. (Cada tipo posee un código exclusivo que se creará de forma automática).



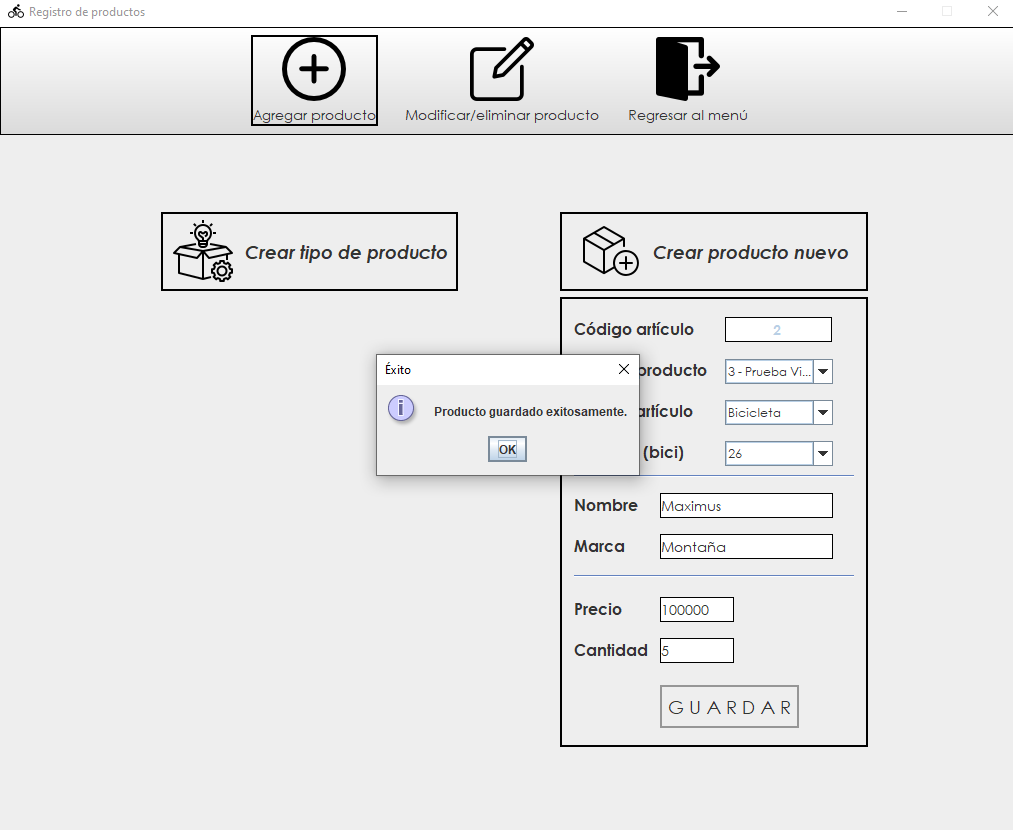
Luego de la creación del tipo de producto se muestra un mensaje de confirmación al usuario.



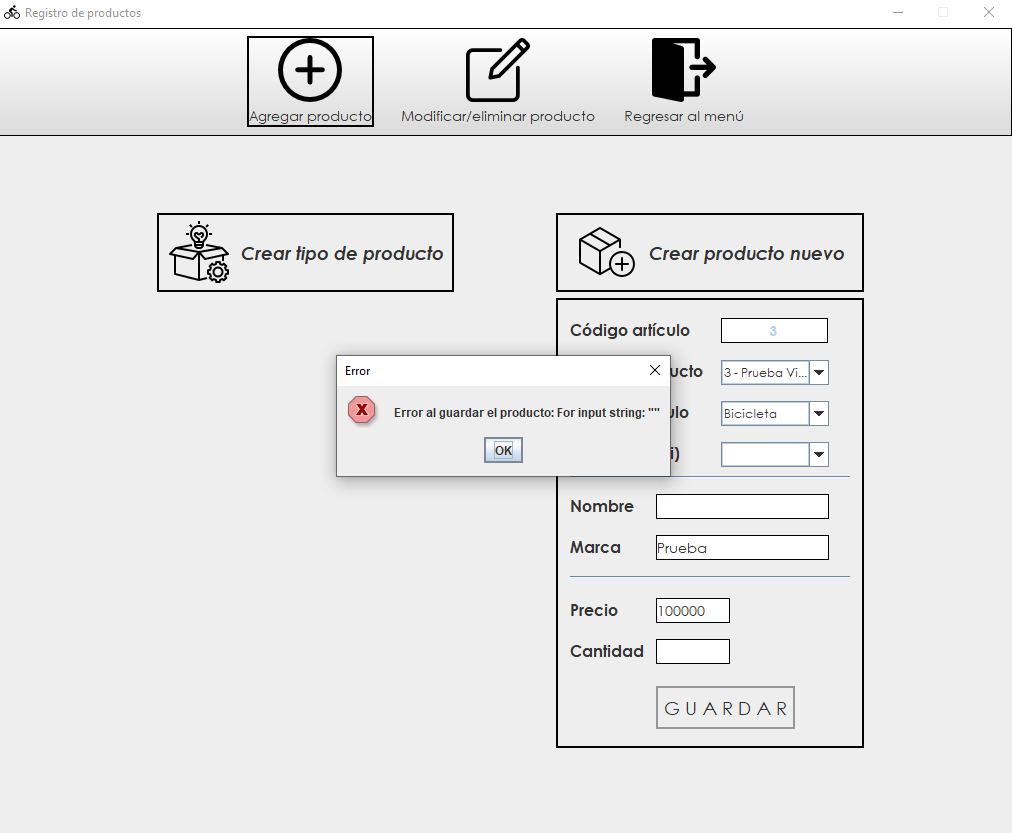
El botón “Crear producto nuevo” se le permitirá al usuario crear productos nuevos de un tipo creado anteriormente, como se observa en la imagen.



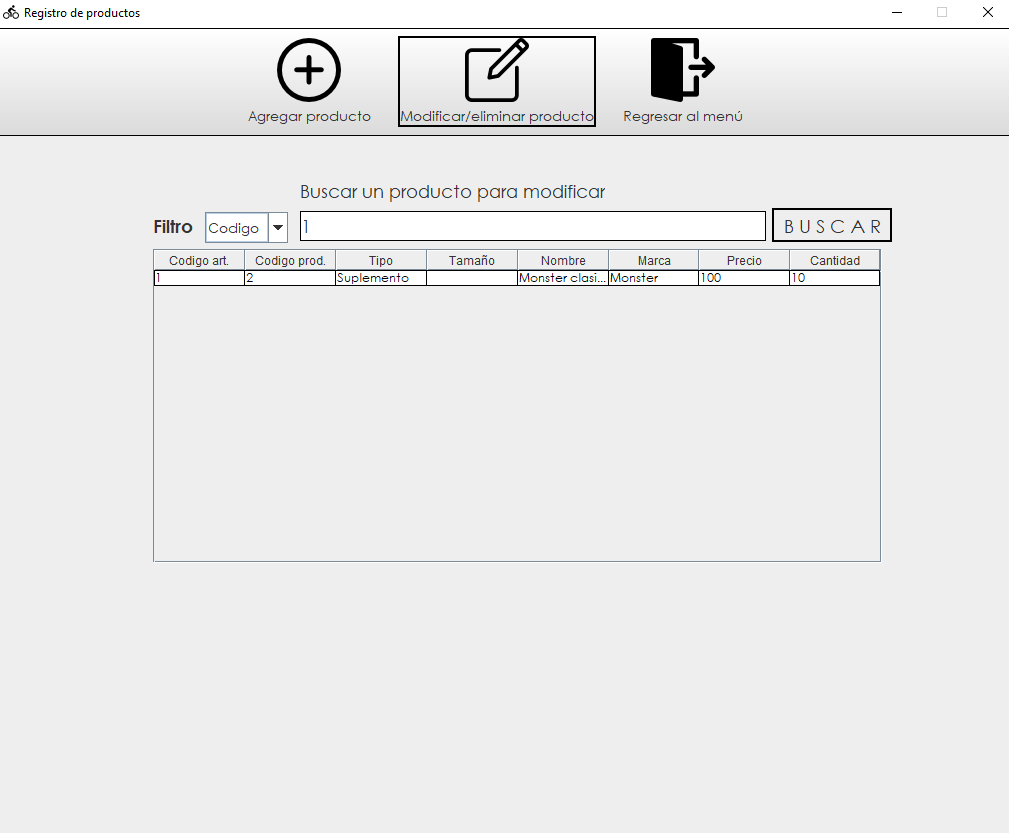
Al igual que los tipos de productos, cada producto poseerá un código de artículo exclusivo, luego se seleccionará algún código de producto creado previamente, se selecciona el tipo de artículo (Bicicleta, Accesorio o Suplemento), en caso de haber seleccionado una bicicleta se marca el tamaño de esta. Luego se le ingresan el nombre, marca, precio y cantidad del producto a crear.



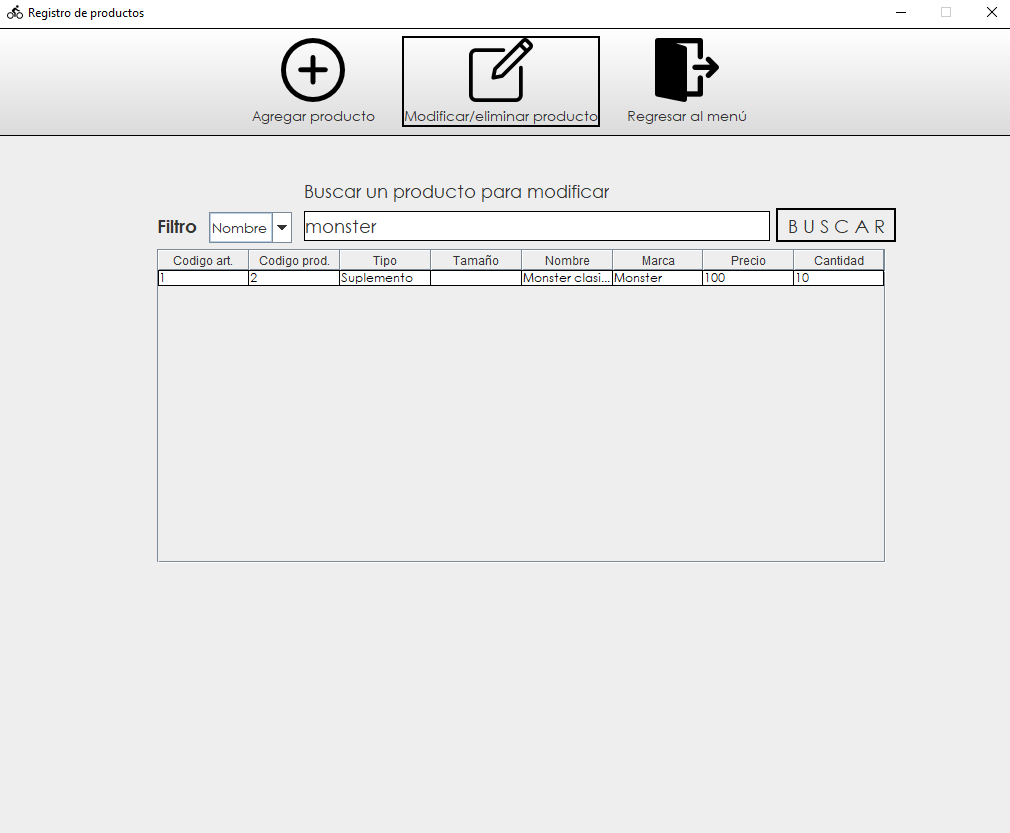
Si se completaron todos los espacios solicitados el producto se creará de forma exitosa.



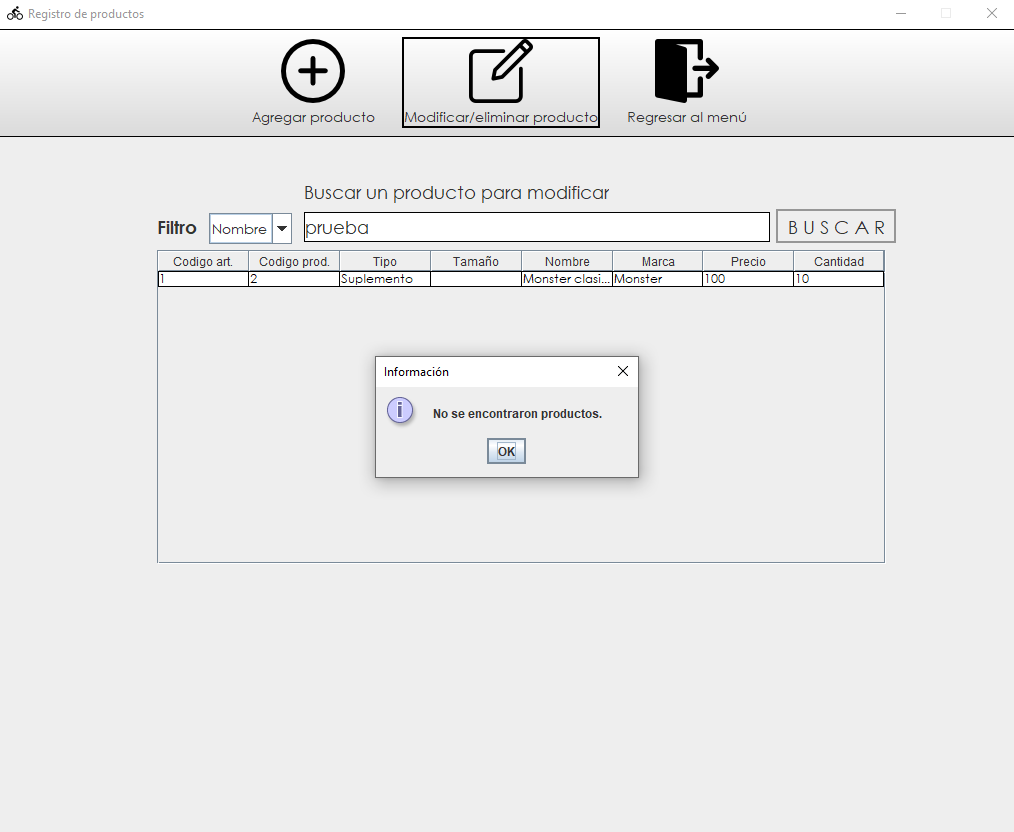
En caso de no completar los datos solicitados se le mostrara al usuario un mensaje de error.



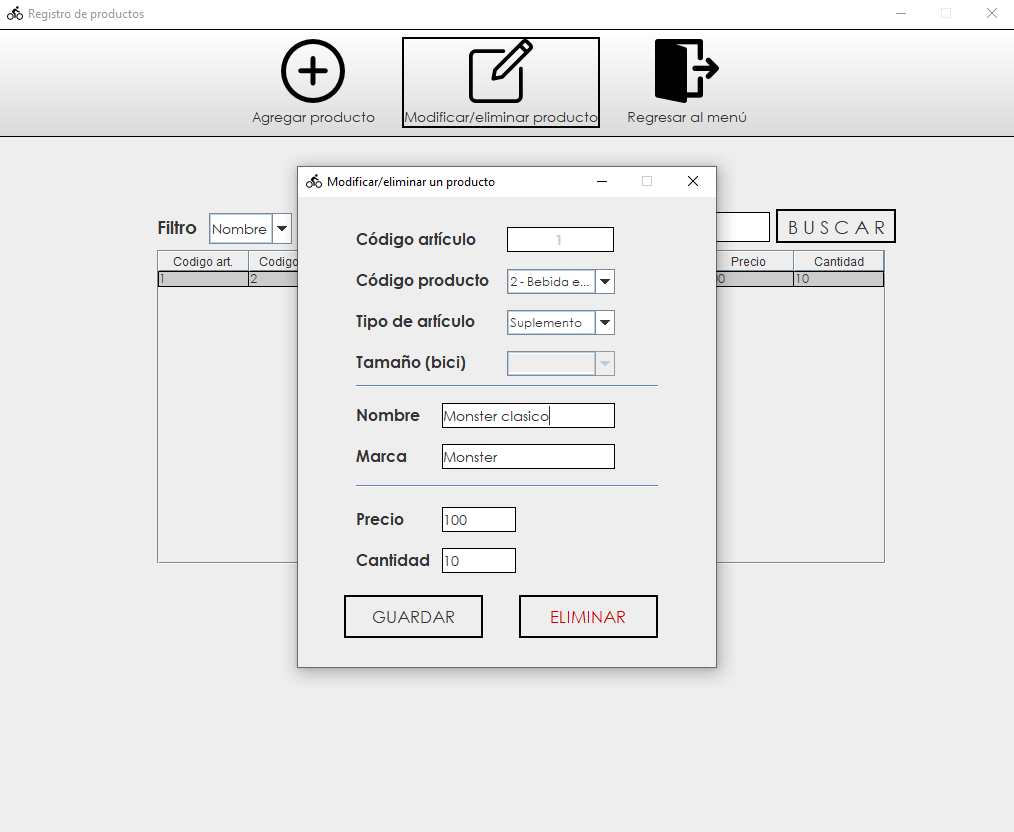
En el botón “Modificar/eliminar producto” se le mostrará al usuario una tabla donde podrá buscar los productos registrados por código o nombre. (Se observa por código).



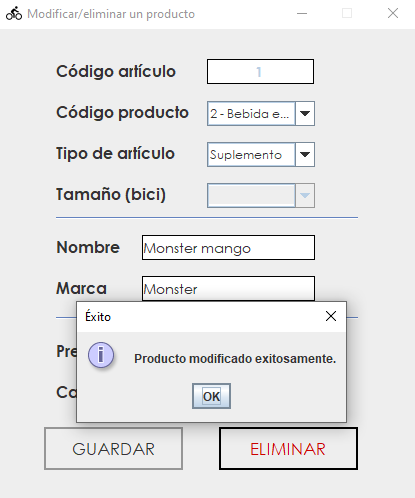
(Se observa por nombre).



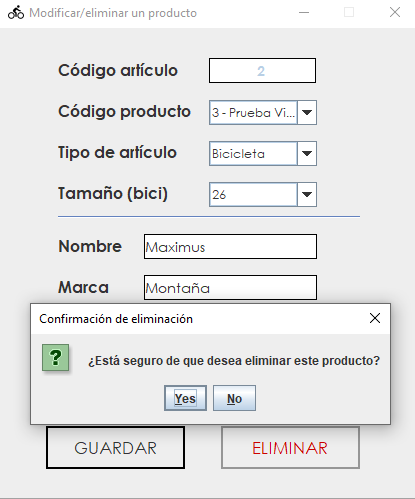
En caso de ingresar algún código o nombre inexistente se le mostrará al usuario un mensaje.



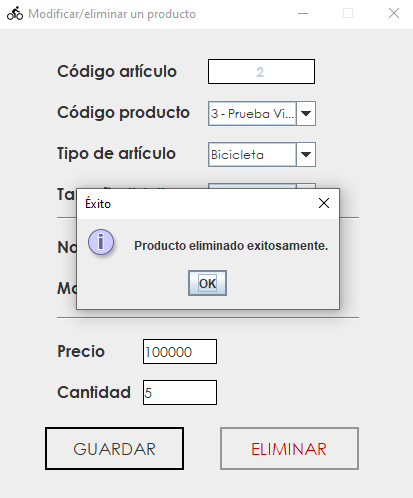
Al dar doble click sobre un producto, este abrirá una ventana que le permite al usuario modificar o eliminar el producto.



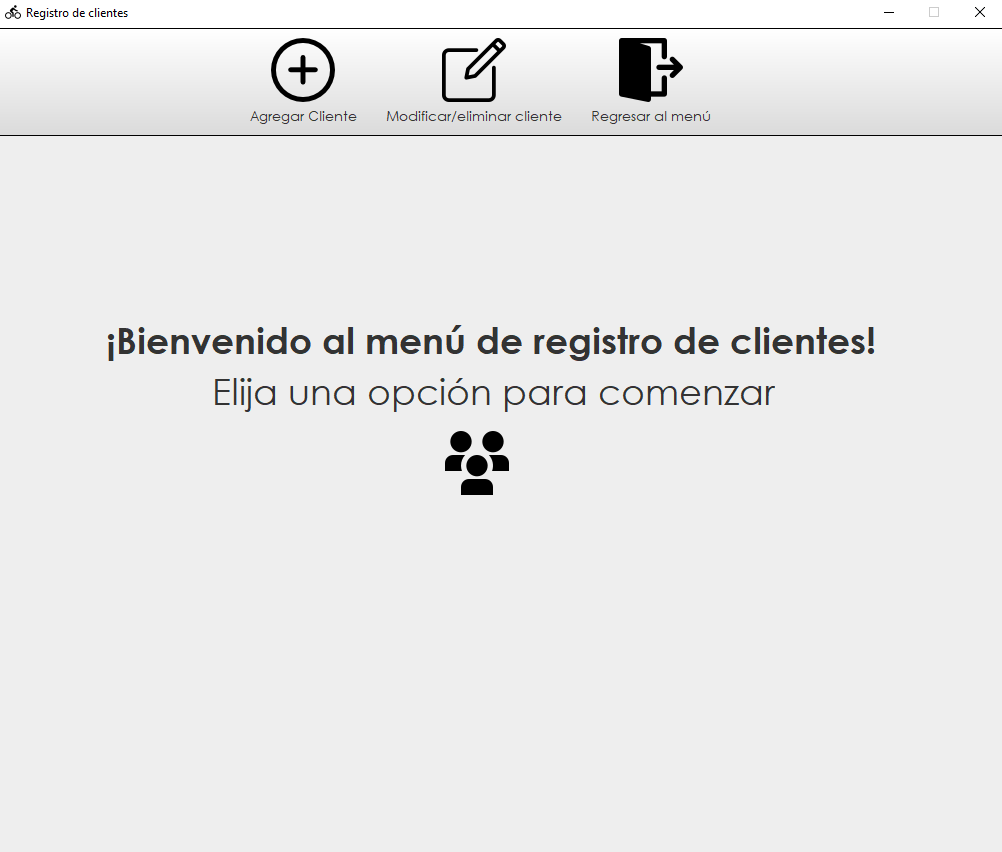
Se modifica el producto.



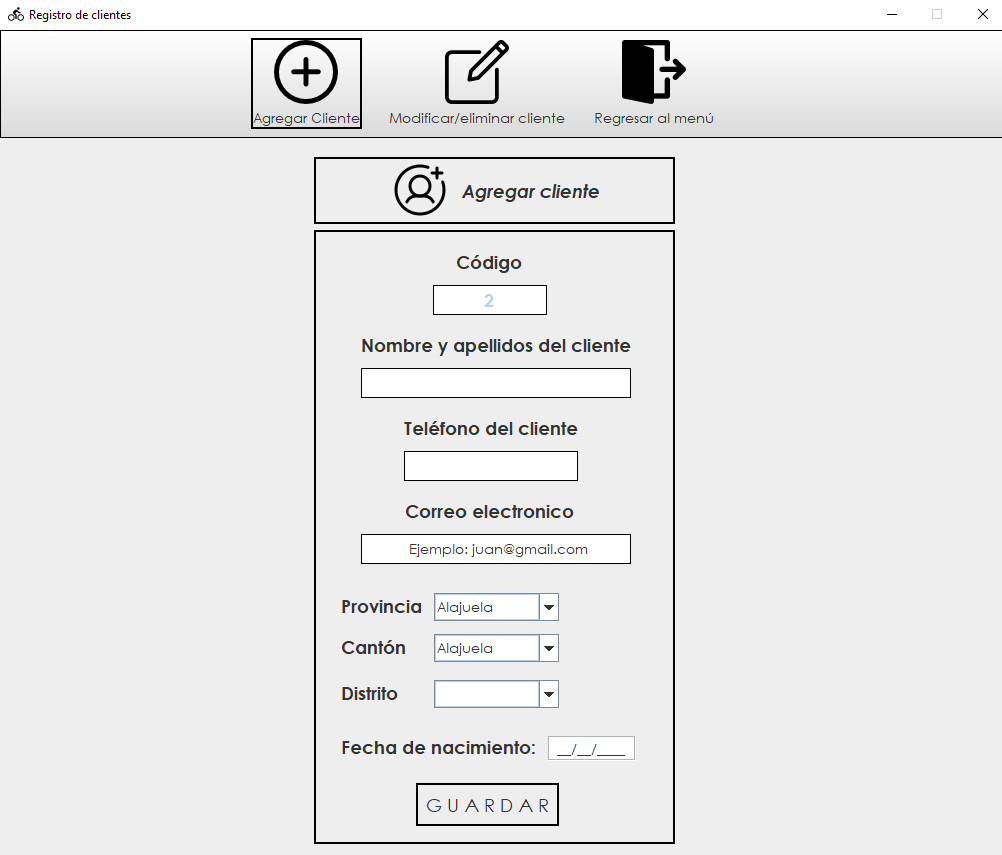
Antes de eliminar un producto se le muestra al usuario una ventana de confirmación.



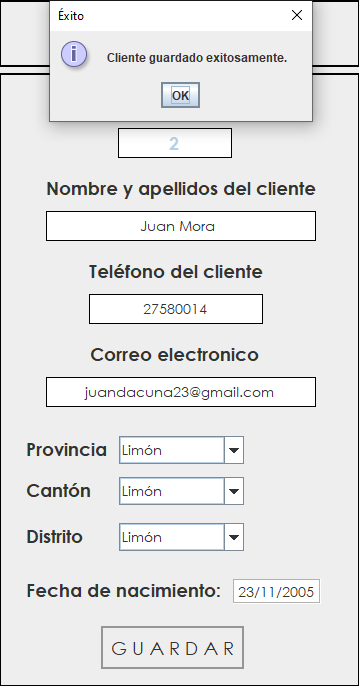
Al presionar “Yes” se le muestra al usuario que el producto se eliminó exitosamente.



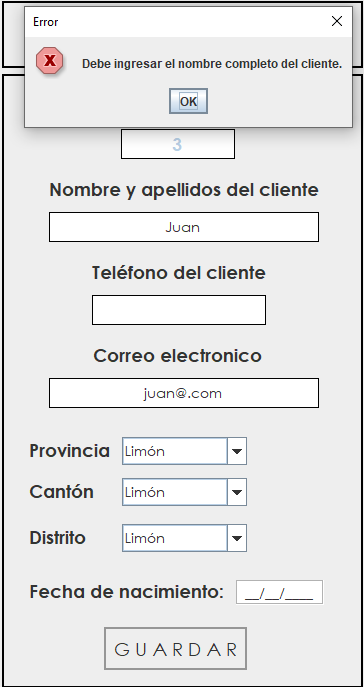
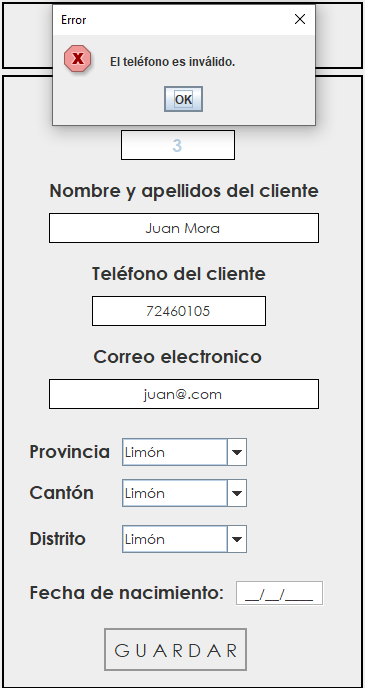
Al seleccionar el módulo de “Registro de clientes” se desplegará esta ventana con mensaje de bienvenida al usuario. En la parte superior se presentan 3 botones (Añadir Cliente, Modificar o eliminar cliente y regresar al menú principal).



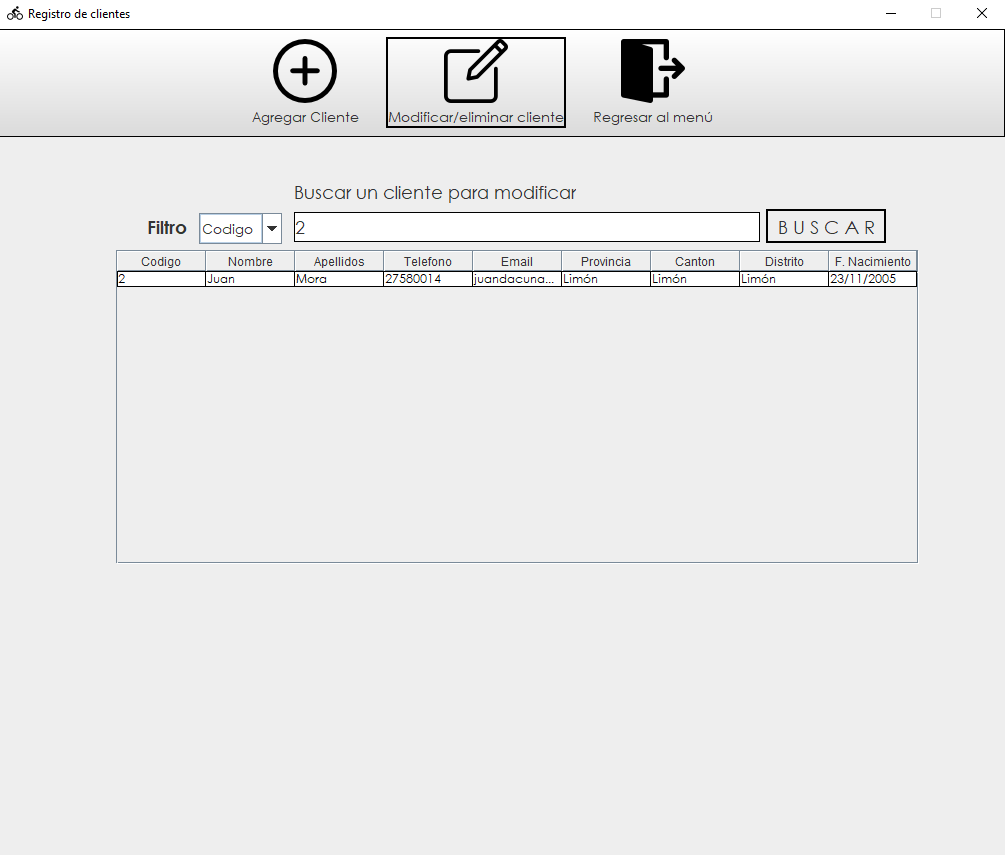
Al seleccionar el botón de “Añadir cliente” se desplegará un botón adicional que le permitirá al usuario crear un nuevo cliente. Donde el usuario deberá ingresar los datos personales del cliente como: nombre, apellidos, número de teléfono, correo electrónico, provincia, cantón, distrito y fecha de nacimiento.



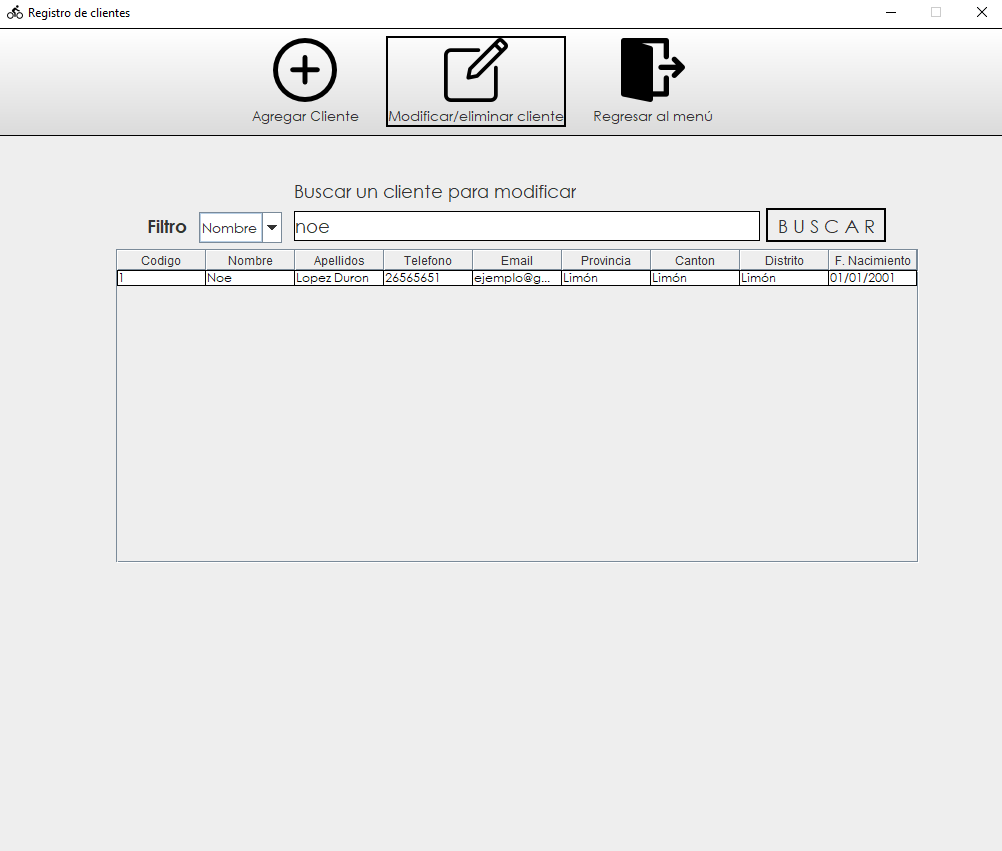
Una vez completado todos los espacios solicitados se habrá creado correctamente el cliente mostrando un mensaje de confirmación.

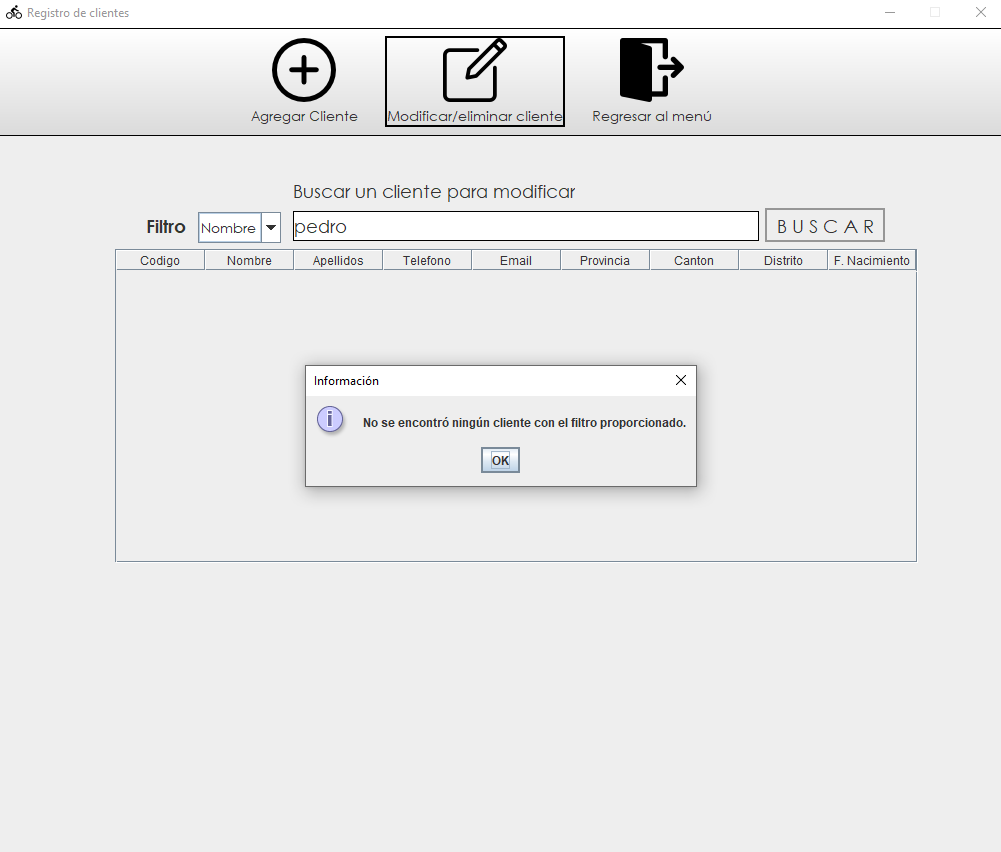
En caso de no ingresar los datos completos o correctos, el programa no permitirá crear un cliente y le mostrará un mensaje de error al usuario.



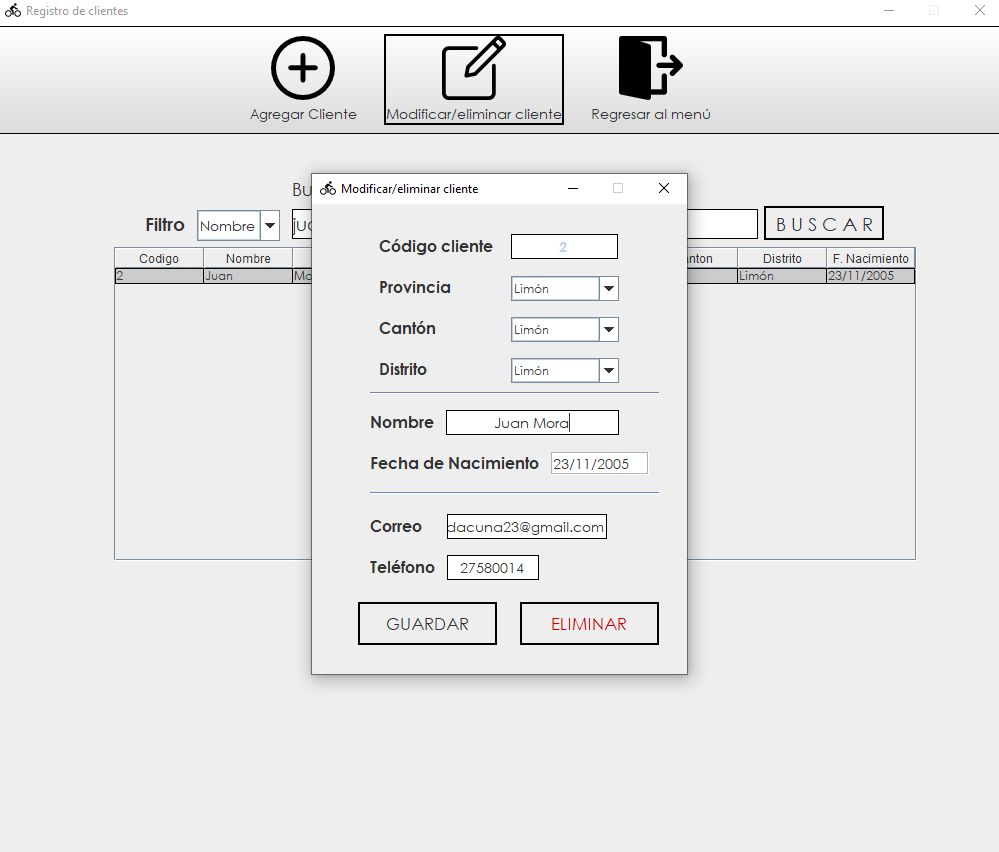
En el botón “Modificar/eliminar cliente” se le mostrará al usuario una tabla donde podrá buscar los clientes registrados por código o nombre. (Se observa por código).



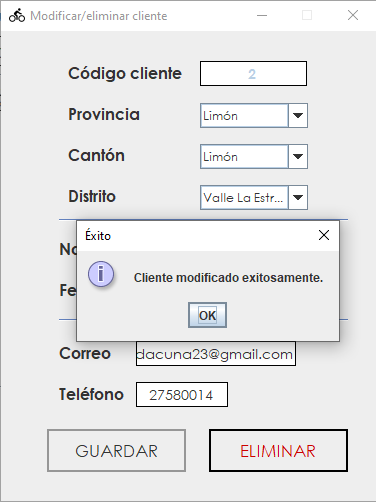
(Se observa por nombre).



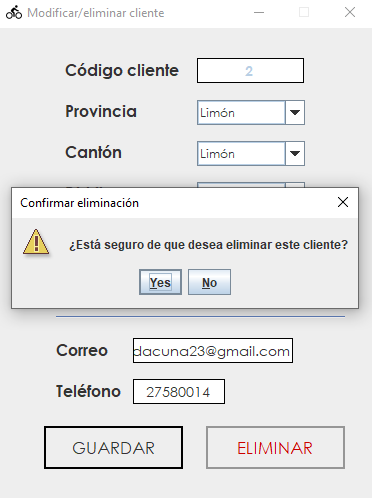
En caso de ingresar algún código o nombre inexistente se le mostrará al usuario un mensaje.



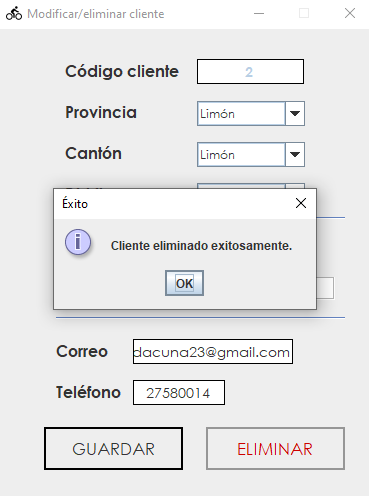
Al dar doble click sobre un cliente, este abrirá una ventana que le permite al usuario modificar o eliminar el cliente.



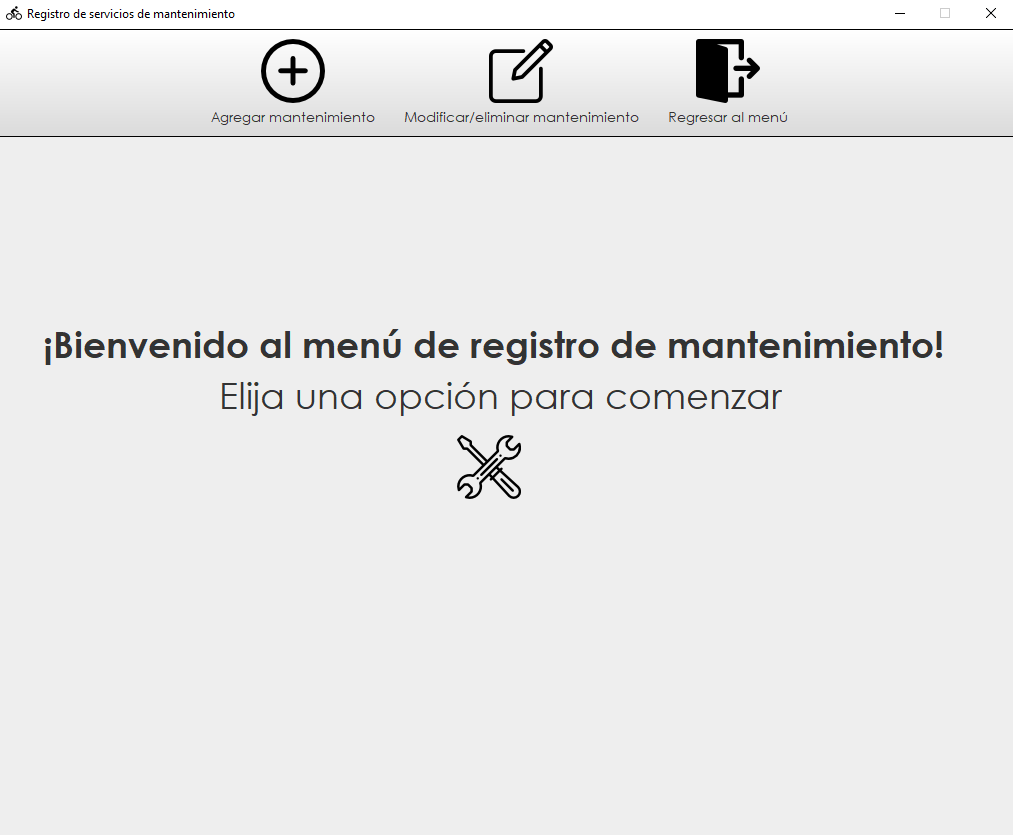
Se modifican los datos del cliente.



Antes de eliminar un cliente se le muestra al usuario una ventana de confirmación.



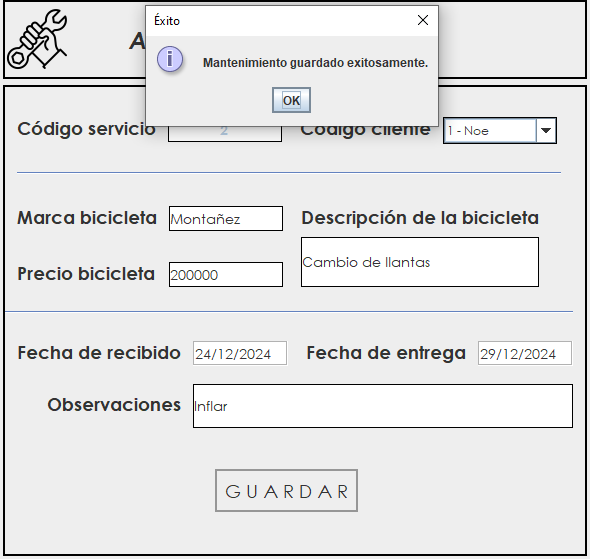
Al presionar “Yes” se le muestra al usuario que el cliente se eliminó correctamente.



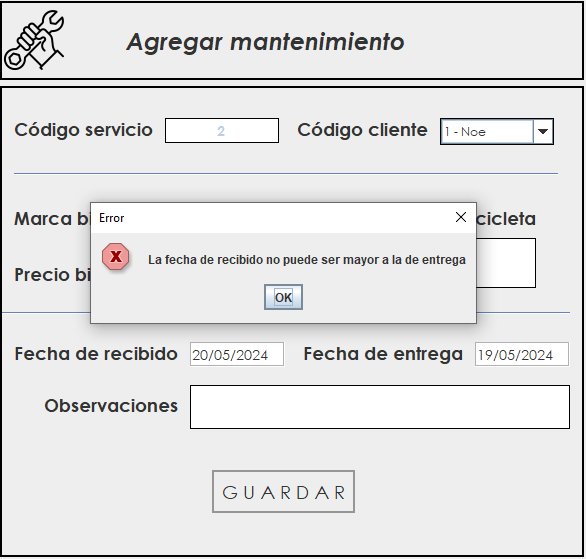
Al seleccionar el módulo de “Registro de mantenimiento” se desplegará esta ventana con mensaje de bienvenida al usuario. En la parte superior se presentan 3 botones (Añadir mantenimiento, Modificar o eliminar mantenimiento y regresar al menú principal).



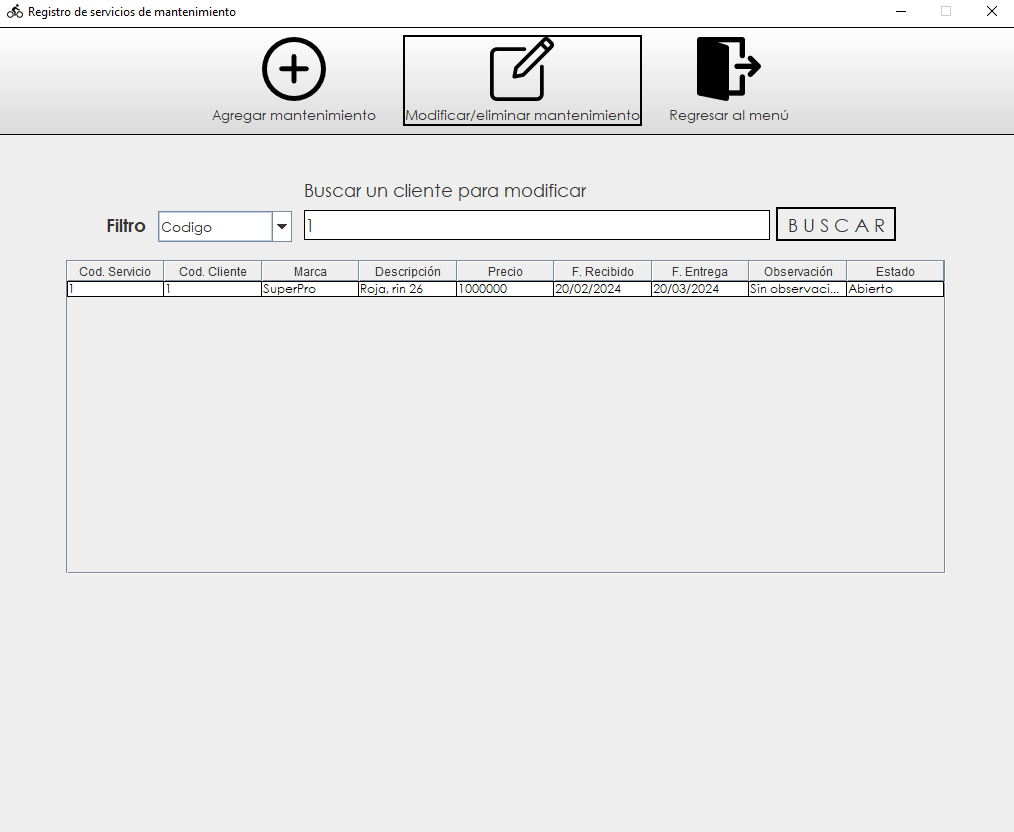
Al seleccionar el botón de “Añadir mantenimiento” se desplegará un botón adicional que le permitirá al usuario crear un nuevo servicio de mantenimiento. Donde el usuario deberá ingresar datos necesarios para brindar el servicio, como el código del cliente que solicita el mantenimiento, la marca de la bicicleta y su precio, descripción de esta, la fecha en la que se recibió la bicicleta y la fecha de entrega, y observaciones. (El código de servicio se crea automáticamente y es exclusivo).



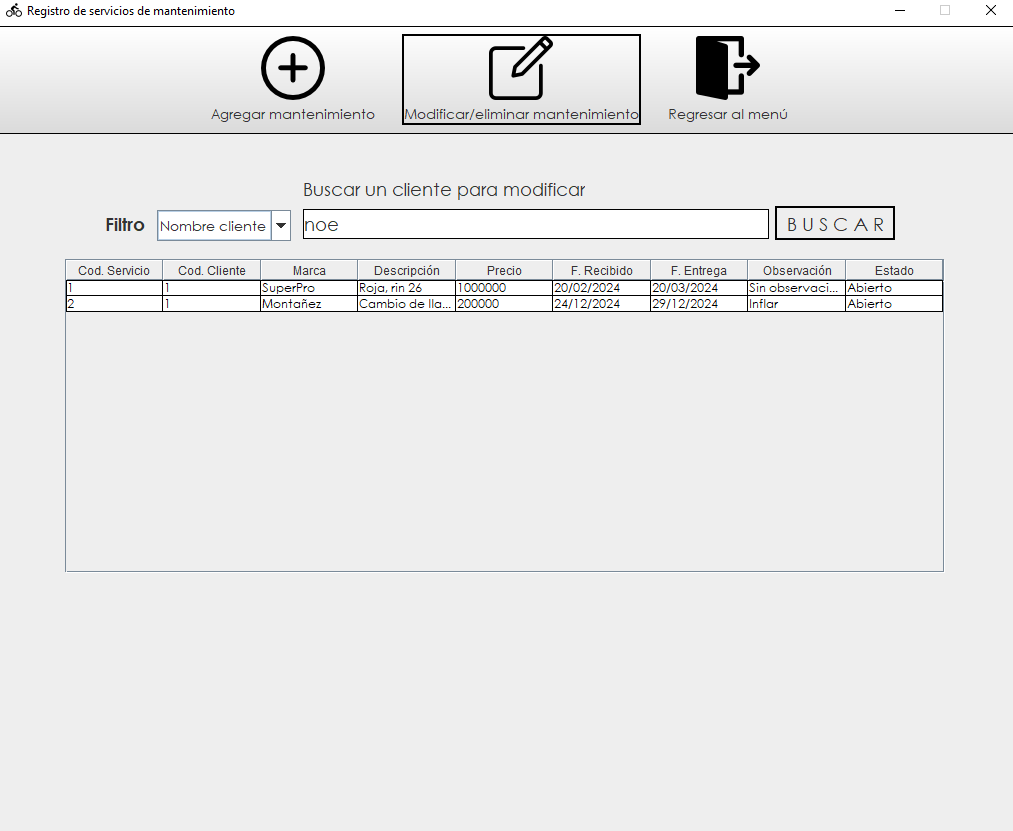
Una vez completado todos los espacios solicitados se habrá creado correctamente el servicio de mantenimiento mostrando un mensaje de confirmación.



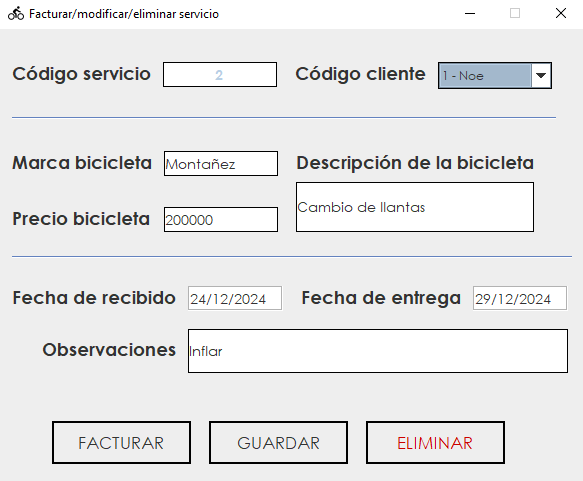
En caso de no ingresar los datos completos o correctos, el programa no permitirá crear un servicio de mantenimiento y le mostrará un mensaje de error al usuario.



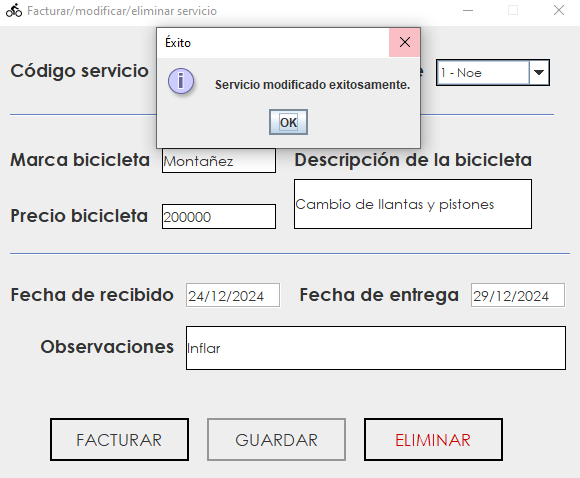
En el botón “Modificar/eliminar mantenimiento” se le mostrará al usuario una tabla donde podrá buscar los servicios de mantenimiento registrados por código o nombre de cliente. (Se observa por código).



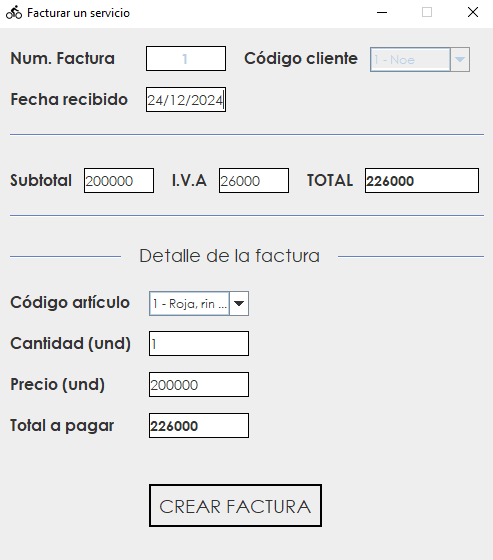
(Se observa por nombre de cliente).



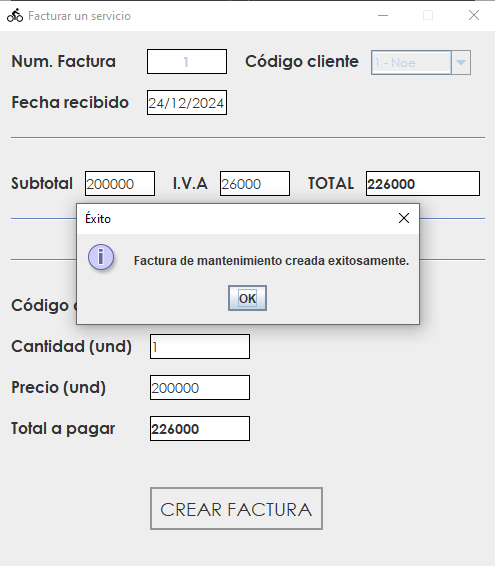
Al dar doble click sobre un servicio de mantenimiento, este abrirá una ventana que le permite al usuario modificar, eliminar o facturar el mantenimiento.



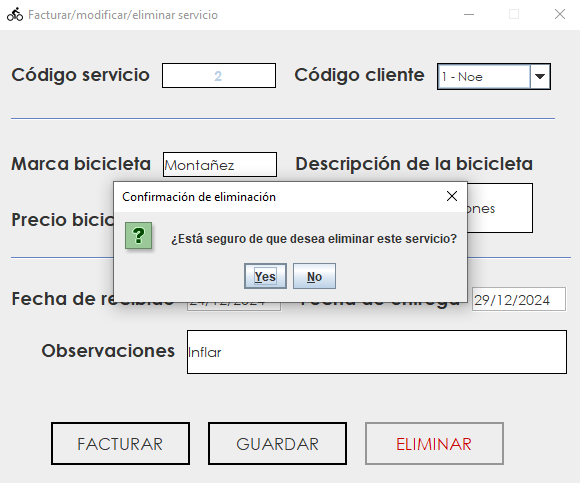
Se modifican los datos del servicio de mantenimiento.



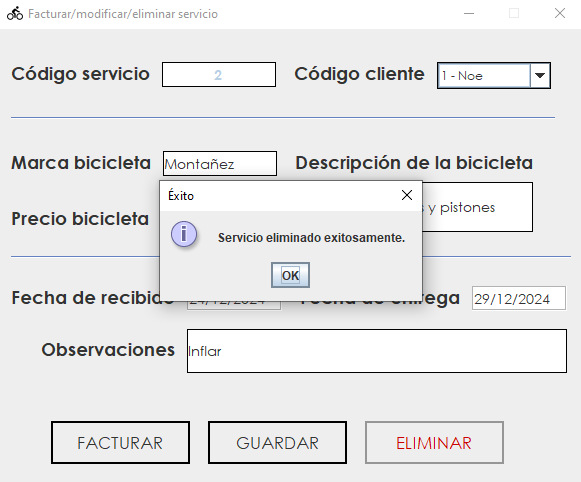
Al seleccionar el botón “FACTURAR” se muestra una ventana donde se procede a realizar la factura del servicio. Se muestra la fecha del recibido, precio subtotal, I.V.A y el Total, la cantidad de artículos.



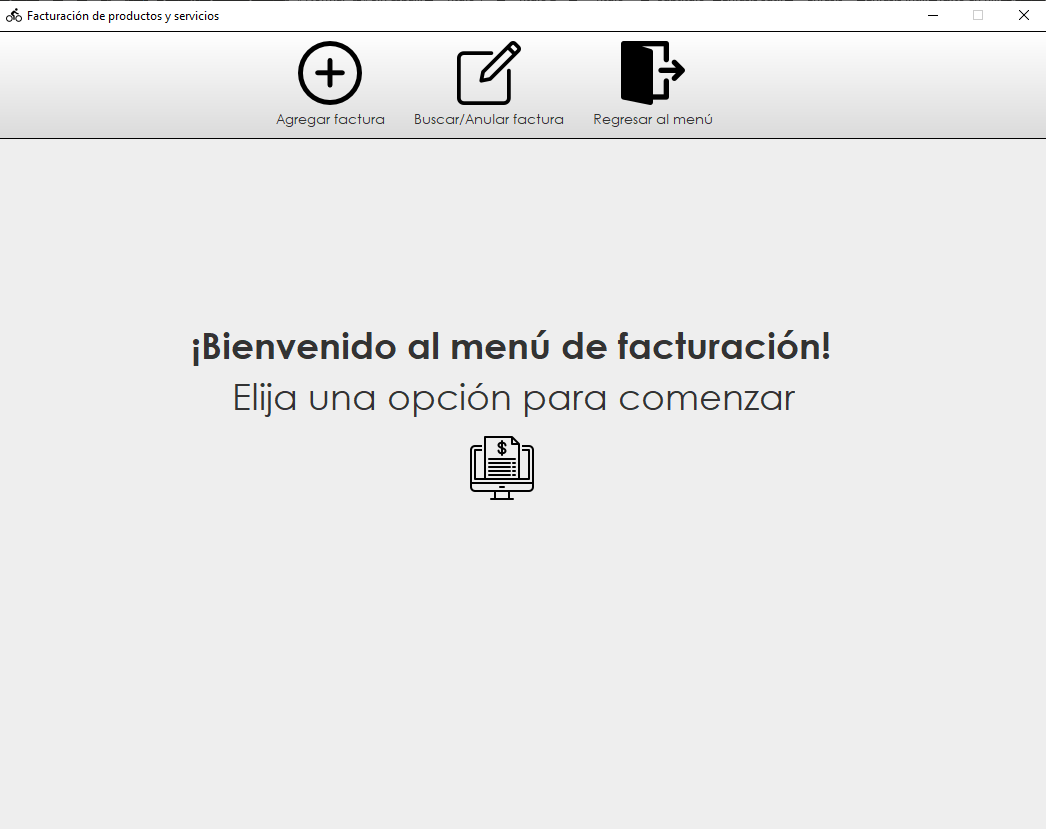
Al presionar el botón “CREAR FACTURA” se le muestra al usuario una confirmación de la factura.



Antes de eliminar un servicio de mantenimiento se le muestra al usuario un mensaje de confirmación.



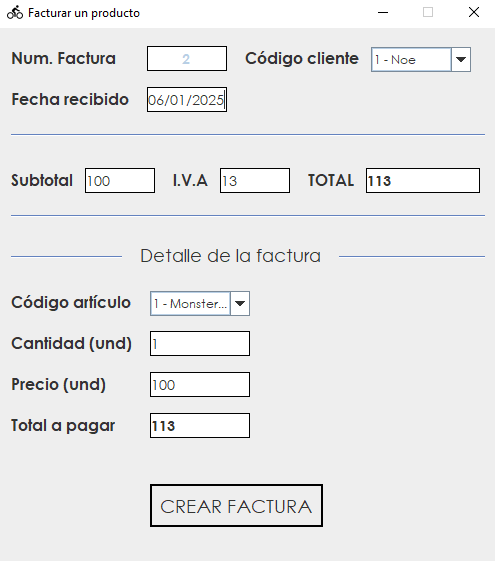
Al presionar “Yes” se le muestra al usuario que el servicio de mantenimiento se eliminó correctamente.



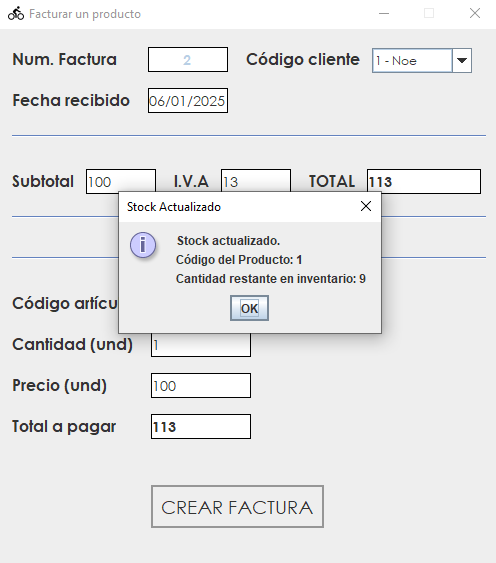
Al seleccionar el módulo de “Facturación” se desplegará esta ventana con mensaje de bienvenida al usuario. En la parte superior se presentan 3 botones (Añadir factura, Buscar o anular factura y regresar al menú principal).



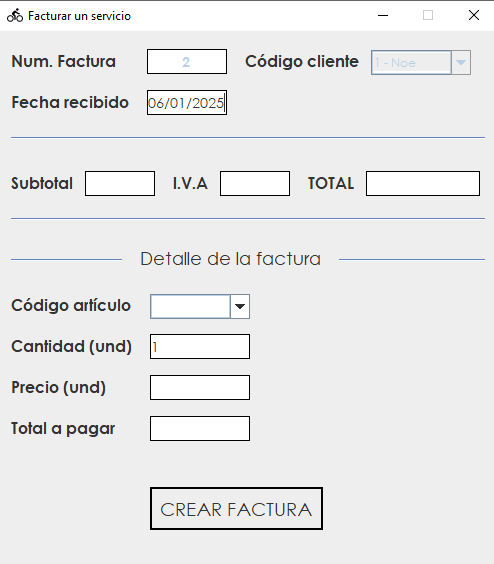
Al seleccionar el botón de “Añadir factura” se desplegarán dos botones adicionales que le permitirán al usuario facturar un producto o facturar un servicio de mantenimiento.



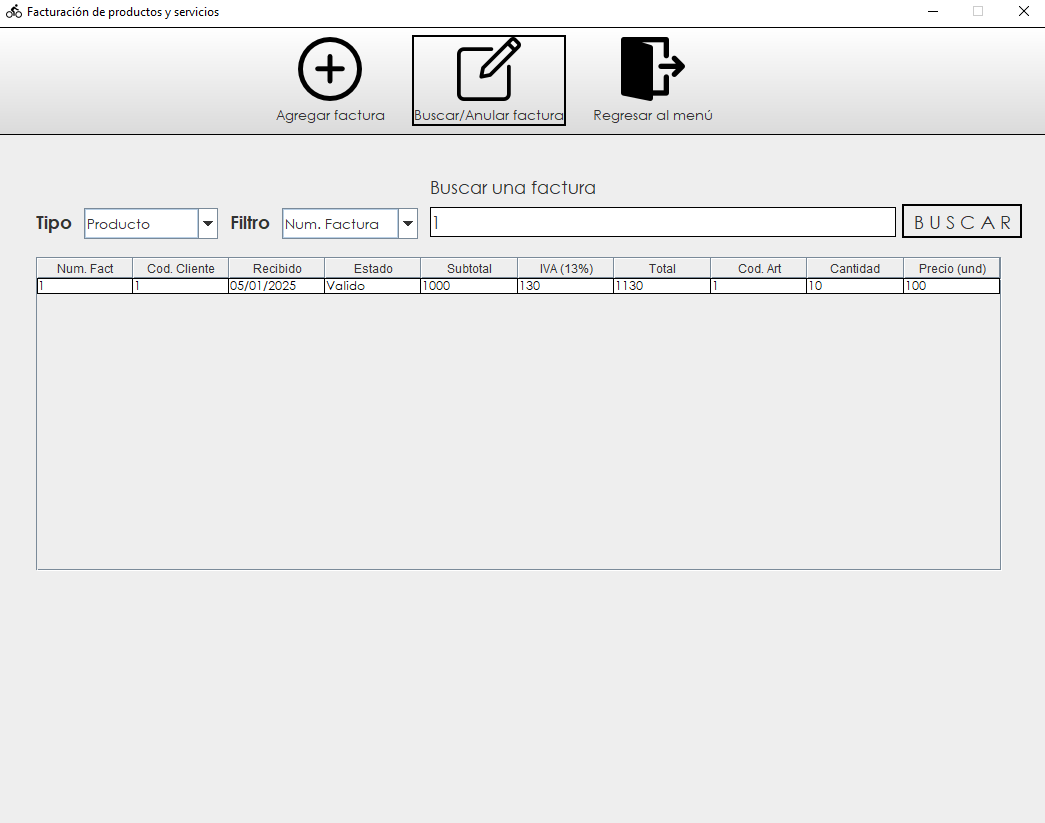
Al presionar el botón “Facturar un producto” el programa permite facturar productos para un cliente. Donde se creará un número exclusivo de factura, el código del cliente a facturar su producto, la fecha de recibido y los datos monetarios, además de los detalles de factura (Código de artículo, la cantidad, el precio y el total a pagar).



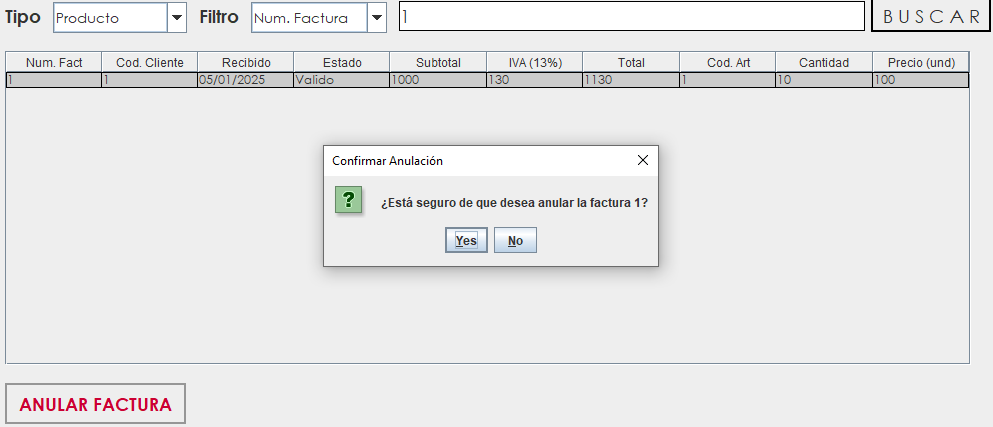
Al crear factura se actualiza el stock y se guarda la factura, mostrándole un mensaje de confirmación al usuario.



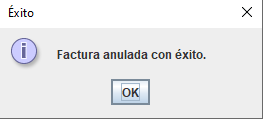
Al presionar el botón “Facturar un servicio” el programa permite facturar un servicio de mantenimiento para un cliente. Donde se creará un número exclusivo de factura, el código del cliente a facturar su producto, la fecha de recibido y los datos monetarios, además de los detalles de factura (Código de artículo, la cantidad, el precio y el total a pagar).



En el botón “Buscar/Anular factura” se le mostrará al usuario una tabla donde podrá buscar las facturas de los servicios de mantenimiento o productos registrados por número de factura, nombre de cliente o fecha.



Al dar doble click sobre una factura, se mostrará un botón adicional que le permite al usuario anular la factura, y se le da un mensaje de confirmación si se presiona esta, permitiendo anular la factura seleccionada.



Al presionar “Yes” el estado de la factura cambia a "Anulado”

# Pruebas de funcionalidad

Enlace video YouTube: <https://youtu.be/jXEyCusbAuA>

# Descripción del problema

Crear un sistema para la gestión y administración de una Tienda de Ciclismo. Se debe permitir la automatización y organización de las operaciones relacionadas con la venta de productos, registro de clientes, servicio de mantenimiento y facturación. En la actualidad este tipo de tareas se puede realizar de forma manual, pero resulta ineficiente, lo que puede llevar a la tienda a errores, perdida de tiempo y dificultad para mantener el seguimiento de las transacciones e información.

Se utiliza el lenguaje de Java con NetBeans para el empleo de este proyecto, empleando los principios de la Programación Orientada a Objetos (POO) creando una interfaz gráfica que sea amigable y fácil de entender para el usuario y una lógica eficiente y funcional.

Permitiendo a la administración de la tienda el Control de Acceso a los datos administrativos, la gestión de productos, gestión de clientes, gestión de servicios de mantenimiento y facturación de los productos o servicios vendidos.

# Diseño del programa

**Clase Archivo:** Clase encargada de la gestión de archivos, permitiendo su almacenamiento y lectura en formato JSON utilizando la biblioteca Jackson.

**Clase Cliente:** Clase que representa un cliente con datos personales y validaciones básicas. Esta clase encapsula la información de un cliente, incluyendo su código, nombre, apellidos, teléfono, correo y ubicación geográfica. Además, proporciona validaciones mínimas para garantizar la integridad de los datos.

**Clase Mantenimiento:** Clase que representa un servicio de mantenimiento de bicicletas. Esta clase encapsula los detalles de un mantenimiento, incluyendo el código del servicio, el cliente asociado, la información de la bicicleta, el precio, las fechas relevantes, observaciones y el estado del mantenimiento.

**Clase Proyecto1:** Clase principal del proyecto. Se encarga de inicializar la interfaz gráfica del programa, mostrando la ventana de credenciales como primer punto de interacción con el usuario. El método “main” utiliza la cola de eventos de AWT para garantizar que la interfaz gráfica se ejecute en el hilo de despacho de eventos de Swing, evitando posibles problemas de concurrencia.

**Clase Usuario:** Clase que representa un usuario con credenciales de acceso. Esta clase proporciona métodos para gestionar el nombre de usuario y contraseña, además de validar credenciales contra un archivo JSON.

**Clase Utilidades:** Clase de utilidades que proporciona métodos auxiliares para la gestión de productos, clientes y mantenimientos en la aplicación. Incluye funciones para cargar datos en combo boxes y tablas, realizar búsquedas y abrir ventanas de modificación.

**Clase Validador:** Clase que proporciona validaciones para entradas de datos en un sistema. Incluye métodos para validar entradas alfabéticas, numéricas, alfanuméricas, correos electrónicos, teléfonos y fechas en formato específico.

**Clase ButtonHoverEffect:** Clase encargada de los efectos hover.

**Clase Credenciales:** Clase que representa la ventana de credenciales de acceso. Esta ventana permite a los usuarios ingresar sus credenciales para autenticarse en el sistema. Cuenta con efectos visuales en los botones y establece el botón de ingreso como el predeterminado.

**Clase Facturación:** Clase que representa la ventana de facturación en el sistema. Esta interfaz permite gestionar facturas, incluyendo la facturación de productos y servicios, la búsqueda y anulación de facturas, y la visualización de detalles. Se implementan efectos visuales en los botones y la navegación mediante un “CardLayout”.

**Clase MenuOpciones:** Clase que representa el menú de opciones principal del sistema. Desde este menú, los usuarios pueden acceder a diferentes módulos como el registro de productos, clientes, mantenimientos y facturación. Además, se implementan efectos visuales en los botones para mejorar la experiencia de usuario.

**Clase ModificarCliente:** Clase que representa la ventana de modificación de clientes. Permite modificar la información de un cliente existente, incluyendo su nombre, teléfono, correo, y ubicación (provincia, cantón y distrito). Cuenta con validaciones de entrada y configuración de límites en los campos de texto.

**Clase ModificarMant:** Clase que representa la ventana de modificación de un mantenimiento. Permite modificar la información de un servicio de mantenimiento existente, incluyendo el código del cliente, marca de bicicleta, precio, descripción, fechas de recepción y entrega, y observaciones. Se implementan validaciones de entrada y configuración de límites en los campos de texto.

**Clase RegistroClientes:** Clase que representa la ventana de registro de clientes. Permite registrar, modificar y gestionar clientes en la base de datos. Incluye validaciones de entrada, configuración de límites en los campos de texto y efectos visuales en los botones.

**Clase RegistroMantenimiento:** Clase que gestiona el registro y modificación de mantenimientos. Permite agregar, modificar y listar servicios de mantenimiento, así como asignarlos a clientes específicos. Incluye validaciones en los campos de entrada y efectos visuales en la interfaz gráfica.

**Clase RegistroProductos:** Clase que representa la interfaz de registro y modificación de productos. Permite registrar nuevos productos y tipos de productos, así como modificarlos. Cuenta con validaciones de entrada, límites de caracteres y efectos visuales en los botones.

**Clase VentanaFacturacionMant:** Clase que representa la interfaz para la facturación de mantenimientos. Permite generar facturas de servicios de mantenimiento, asociando un cliente con un mantenimiento previamente registrado. Incluye funcionalidades para la asignación automática del número de factura, la carga de clientes y mantenimientos abiertos, y la configuración del campo de fecha.

**Clase VentanaFacturacionProd:** Clase que representa la interfaz para la facturación de productos. Permite generar facturas de productos, asociando un cliente con un producto previamente registrado. La clase se encarga de manejar la creación automática de números de factura, la carga de clientes y productos en los comboBox, y la validación de los campos de fecha.

**Clase VentanaModificar:** La clase VentanaModificar permite modificar los datos de un producto registrado. Proporciona funcionalidades para modificar el nombre, marca, precio y cantidad del producto, además de configurar los efectos visuales en los botones y validar las entradas de usuario.

**Clase Producto:** Clase que representa un producto en el sistema. Incluye información detallada del producto, como su código, tipo, tamaño, marca, precio y cantidad en stock. Implementa la interfaz “ConCodigo” para proporcionar un identificador único.

**Clase TipoProducto:** Clase que representa un tipo de producto dentro del sistema. Contiene un identificador único y un nombre que describe el tipo de producto. Implementa la interfaz “ConCodigo” para proporcionar un identificador estándar.

**Clase BuscarFactura:** Clase que permite la búsqueda de facturas de productos o de mantenimiento en archivos JSON almacenados. Proporciona un método para buscar facturas según un criterio específico.

**Clase DetalleFactura:** Clase que representa el detalle de una factura. Contiene información sobre el artículo facturado, incluyendo su código, número de factura, cantidad, precio unitario y total.

**Clase EncabezadoFactura:** Clase que representa el encabezado de una factura. Contiene información general de la factura, incluyendo el número de factura, código del cliente, fecha de recepción, estado de la factura, subtotal, impuesto y total.

**Clase Factura:** Clase que representa una factura con un encabezado y una lista de detalles. Contiene información de la factura, incluyendo su encabezado y los detalles de los productos o servicios facturados. Además, permite agregar detalles y recalcular los totales automáticamente. Implementa la interfaz “ConCodigo” para permitir la obtención de un identificador único basado en el número de factura.

**Clase GuardarFactura:** Clase encargada de la gestión y almacenamiento de facturas. Proporciona métodos para guardar facturas de productos y facturas de mantenimiento, asegurando que los datos sean almacenados correctamente en archivos JSON.

**Clase UtilidadesFacturas:** Clase que proporciona utilidades para la gestión de facturas. Incluye métodos para cargar mantenimientos abiertos, cargar productos y obtener productos por código.

# Librerías usadas

**Librería Swing:** Se utilizó para la interfaz gráfica.

**Librería Jackson:** Se usó para el manejo de archivos .JSON

**Archivos creados:**

* **facturas\_mantenimiento:** Guarda las facturas de los servicios de mantenimiento.
* **facturas\_productos:** Guarda las facturas de los productos.
* **mantenimiento:** Guarda el registro de los servicios de mantenimiento.
* **productos:** Lleva el registro de los productos creados.
* **Provincias\_Cantones:** Se usa para el orden y la selección a la hora de crear el espacio geográfico de un cliente.
* **registroClientes:** Guarda el registro de los clientes.
* **tipoProductos:** Guarda los tipos de productos.
* **usuarios:** Usuarios con su respectiva contraseña para el ingreso administrativo.

# Análisis de resultados

**Objetivos alcanzados:**

Se logró la completitud de lo solicitado en el proyecto, la estética e interfaz del trabajo fue valorada por todos los integrantes del grupo y todos estuvimos de acuerdo con ella. La lógica se terminó por completo logrando el pleno funcionamiento del programa esperado.

\*NO HUBIERON OBJETIVOS NO ALCANZADOS\*