Instituto Tecnológico de Costa Rica

Profesor: Cristian Paz Campos Agüero

Curso: Programación Orientada a Objetos (POO) IC-2101

**PROYECTO PROGRAMADO 1: TIENDA DE CICLISMO**

Estudiantes: Jose Mario Jiménez Vargas - 2023102334

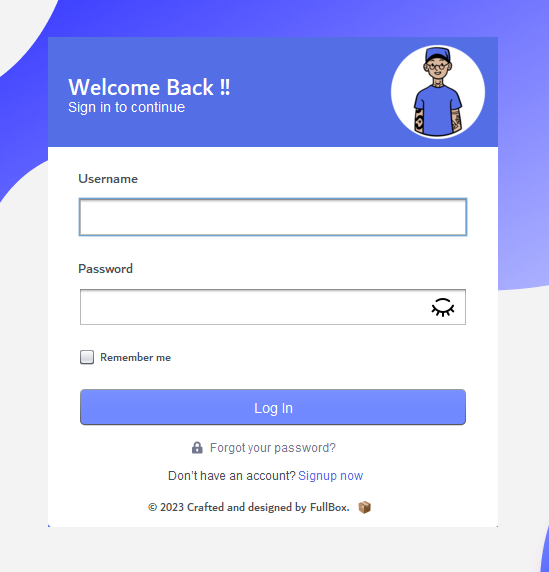
John Sebastián Ceciliano Piedra - 2023267790

Saymon Porras Briones - 2023172692

Fecha de entrega: miércoles 20 de diciembre del 2023

# Manual de usuario

***Ingresar Usuario***



***????***

***Restablecer Contraseña***

***Registrar Nuevo Usuario***

***Validar Campos***

***Recordar Contraseña***

***Ingresar Contraseña***

***Ver Contraseña***

Esta ventana constituye la primera interfaz que se presenta al iniciar el programa. Desde aquí, se brinda la posibilidad de autenticar el usuario mediante la introducción de sus credenciales, incluyendo nombre de usuario y contraseña. Asimismo, se ofrece la opción de habilitar la función de recordar la contraseña para futuras sesiones.

Además de las funciones básicas de inicio de sesión, la ventana incluye opciones relevantes como la posibilidad de registrarse como nuevo usuario. También se presenta un mecanismo para la recuperación de contraseña en caso de olvido, asegurando una experiencia de usuario completa y segura.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Nueva Contraseña***

***Correo De Recuperación***

***Salir***

***Validar Campos***

***Validar Contraseña***

***Nuevo Usuario***

Esta interfaz de registro ofrece la posibilidad de crear un nuevo usuario, completando el proceso con la asignación de una contraseña y la inclusión de una dirección de correo electrónico para fines de recuperación de cuenta.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***¿Olvidaste tu contraseña?  
Sigue los siguientes pasos.***

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Click

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

***Ventana Productos***

***Ventana Clientes***

***Ventana Servicios***

Icono

Descripción generada automáticamente

***Recordar***

***Regresar Al   
Inicio De Sesión***

***Ventana Impresión***

***Ventana Facturacion***

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

***Recordar***

***No Recordar***

En esta ventana, se han establecido cinco secciones distintas para realizar registros, adaptándose a las diversas necesidades del usuario. Estas secciones permiten la creación de nuevos productos e ítems, la incorporación de nuevos clientes, la introducción de nuevos servicios, la facturación de las operaciones mencionadas y la posibilidad de imprimir dicha documentación, lo cual se considera esencial para el funcionamiento integral del programa. Por último, se encuentra el botón "Cerrar Sesión", que facilita el retorno a la pantalla inicial de inicio de sesión.

***Borrar Producto***

***Modificar Producto***

***Buscar Producto***

***Crear Producto***

***Regresar***

Gráfico

Descripción generada automáticamente con confianza media

***Aceptar***

***Cancelar***

***Buscar***

***Borrar Artículo***

***Modificar Artículo***

***Cancelar***

***Buscar***

***Crear Artículo***

***Aceptar***

***Crear Artículo***

En esta interfaz, se llevan a cabo simultáneamente dos tipos de registros: productos y artículos. Cada sección cuenta con sus propios controles CRUD (Crear, Leer, Actualizar, Eliminar) y botones de aceptar y cancelar. La disposición conjunta de estas secciones en una misma ventana responde a la interdependencia entre los ítems y los productos, ya que los primeros están directamente vinculados a las registradas en los segundos. Esta implementación no solo optimiza la practicidad de la operación, sino que también contribuye a lograr un diseño estético y funcional en la interfaz del programa.

***Modificar Cliente***

***Buscar Cliente***

***Regresar***

***Borrar Cliente***

***Agregar Cliente***

Tabla

Descripción generada automáticamente con confianza baja

***Aceptar /Buscar***

***Restablecer Campos***

***Cancelar***

En esta ventana, se lleva a cabo el ingreso de nuevos clientes mediante funciones CRUD, donde cada operación (Crear, Leer, Actualizar, Eliminar) facilita la gestión de la información. Una vez completados y validados todos los campos, el nuevo cliente se registra de manera eficiente en la tabla ubicada en el lado derecho de la interfaz. Esta disposición resulta práctica para futuras modificaciones y su utilización en otras secciones del programa.

***Modificar Servicio***

***Buscar Servicio***

***Regresar***

***Borrar Servicio***

***Crear Servicio***

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Aceptar***

***Cancelar***

***Buscar***

Esta ventana es la de ingreso de servicios. La implementación de las funciones CRUD en esta sección garantiza una gestión eficaz de la información relacionada con los servicios. La disposición práctica de los elementos en la interfaz facilita futuras modificaciones y asegura su utilidad en otras secciones del programa.

***Anular Factura***

***Buscar Factura***

***Crear Factura***

***Borrar Producto***

***Regresar***

Interfaz de usuario gráfica

Descripción generada automáticamente

***Aceptar Factura   
Servicio***

***Aceptar Factura Articulo***

Esta interfaz guarda una estrecha relación con varias ventanas previas, tales como las de artículos, servicios y clientes, ya que para generar una factura es imprescindible contar con la información proveniente de estos apartados. En la interfaz, se implementan funciones Create (crear), Search (buscar) y Remove (eliminar), dado que la modificación no está permitida.

Cabe destacar que esta interfaz fue diseñada con la finalidad de integrar ambos estilos de factura en una misma ventana, buscando así maximizar la comodidad y practicidad del usuario. La inclusión de ambas opciones facilita la adaptabilidad a distintos contextos y necesidades, proporcionando una experiencia más completa y eficiente.

Gráfico

Descripción generada automáticamente

***Manual De Usuario***

***Documentación***

***Creadores***

***Nuevo Detalle***

***Aceptar Factura***

***Regresar***

Esta interfaz tiene la función de extraer información de las facturas generadas y llevar a cabo la impresión correspondiente en un archivo PDF. Su operación se centra en la búsqueda de facturas a través de su identificador (ID). Una vez identificada la factura por su ID, se procede a exhibir los campos necesarios para la impresión, para finalmente generar el archivo PDF que representa la factura completa.

# Pruebas de funcionalidad

En el enlace siguiente a este texto encontrará un link que lo dirigirá a un video colgado en la plataforma YouTube, que contiene las pruebas de funcionalidad del programa: \*INSERTAR LINK\*

# Descripción del problema

Este sistema se encargará en gestionar la venta de productos relacionado al ciclismo como también ofrecer servicios de taller.

Un sistema de gestión es usado para obtener información, realizar transacciones y hacer pedidos para algún servicio, como por ejemplo la gestionar los trabajos en un taller mecánico.

El desarrollo de este programa debe ser Orientado a Objetos por lo que las diferentes clases que forman parte de este enunciado han de desarrollarse. La información debe de guardarse en un archivo plano con separadores (CSV), JSON o XML.

# Diseño del programa

|  |  |
| --- | --- |
| **Clase y descripción** | **Relaciones con otras clases** |
| Main Class: La clase principal desempeña un papel esencial en el programa al encargarse de la inicialización del ArrayList con la información de los objetos registrados. | Esta desempeña un papel crucial al establecer relaciones fundamentales con las clases subyacentes (ítems, products,customers,maintenance,ids,users). En particular, esta clase se encarga de inicializar las listas de objetos utilizando la estructura ArrayList. Cada instancia de estas listas almacena objetos generados por las respectivas clases. Gracias a esta implementación, se logra un manejo eficiente y sin complicaciones de los objetos creados, proporcionando así una base sólida para la gestión integral de las instancias de las clases secundarias. |
| Category Class: La clase de Categoría de Producto juega un papel central en el programa, ya que se utiliza para la creación de objetos de dicha categoría y para llevar a cabo modificaciones en su información. | Esta clase desempeña un papel fundamental y comparte una relación esencial con la clase de artículos (item class). Su función principal radica en generar las categorías que deben obligatoriamente asociarse a cada artículo. En otras palabras, esta clase facilita la creación y gestión de las categorías que definen y enriquecen la información de cada artículo, contribuyendo así a la estructura integral del sistema. |
| Item Class: Es la clase de artículos, se utiliza para crear los objetos de Artículos y hacer modificaciones en la información de ellos. | Esta clase mantiene una relación fundamental con la clase de Facturación (Bill class), ya que, al generarse ítems en esta clase, es imperativo su uso en el proceso de facturación. La conexión estrecha entre ambas clases asegura que los artículos seleccionados por el cliente se reflejen de manera precisa y completa en el proceso de facturación, contribuyendo así a una experiencia de transacción coherente y eficiente. |
| Customer Class: Es la clase de clientes, se utiliza para crear los objetos de clientes y hacer modificaciones en la información de ellos. | Esta clase establece una relación estrecha tanto con la clase de Mantenimiento (Maintenance class) como con la clase de Facturación (Billing class). La interconexión entre estas clases se hace evidente, ya que ambas requieren un cliente previamente registrado como atributo. Este cliente, a su vez, se asigna a un servicio en el caso de la clase de Mantenimiento o a un artículo en el contexto de la clase de Facturación. Esta relación facilita una integración fluida entre las operaciones de mantenimiento y facturación, asegurando que los servicios o artículos se asocien correctamente con un cliente registrado, contribuyendo así a la coherencia y eficiencia del sistema en su conjunto. |
| Maintenance Class: Es la clase de mantenimiento, se utiliza para crear los objetos para registrar los servicios de mantenimiento y hacer modificaciones en la información registrada. | Esta clase establece una relación significativa con la clase de Facturación (Billing class), ya que su propósito fundamental es registrar servicios que, en etapas posteriores, serán objeto de facturación. Al utilizar esta clase para la captura y registro de servicios, se sienta la base para la posterior creación de facturas en la clase de Facturación. Esta conexión directa asegura que los servicios prestados se documenten de manera precisa y completa, facilitando así un proceso de facturación eficiente y preciso en la clase correspondiente. |
| Bill Class: Es la clase de facturación, se utiliza para crear los objetos de facturas y hacer modificaciones en la información de estas. | Esta clase mantiene una conexión directa y esencial con la clase de Detalles (Detail class), ya que los datos generados en esta clase están destinados a ser cargados en la clase de Detalles para su posterior impresión. La interacción efectiva entre ambas clases garantiza que la información relevante se capture y presente de manera eficiente en los documentos de detalle. |
| Detail Class: | Esta clase finaliza el proceso al generar la factura que el cliente verá. Con información recopilada previamente, como detalles de servicios y productos, organiza estos datos de manera coherente en un documento de factura. La coordinación efectiva con clases anteriores garantiza que la factura refleje con precisión las transacciones, brindando así una experiencia satisfactoria al cliente. |
| Id Class: Es la clase de conteo de id’s, se utiliza para hacer aumentos en los id’s de las diferentes opciones del programa que requieren llevar un aumento 1 en 1. | Esta clase no interactúa directamente con otras clases, pero está relacionada con los atributos de algunas de ellas al facilitar un conteo claro de la cantidad de objetos generados en cada instancia. Su función es proporcionar una visión consolidada y organizada del número de objetos creados en el contexto de diferentes ventanas o instancias del programa. Aunque no participa activamente en la interacción con otras clases, contribuye significativamente al seguimiento y la administración efectiva de la cantidad de instancias generadas en el sistema. |
| VerifyUsers: Es la clase para la verificación de usuarios en el programa, se utiliza para la creación de objetos de verificación y comprobar que los usuarios o personas que quieren ingresar al programa son aceptados en este. | Esta clase no tiene relación con otras clase ya que esta desempeña un papel integral al autenticar las credenciales del usuario y gestionar la persistencia de la opción "Recordar contraseña". |
| Register Users: Es la clase para el registro de nuevos usuarios, se utiliza para la almacenar un nuevo usuario, contraseña y si el cliente desea recordar posteriormente su contraseña. | Esta clase, centrada en la validación de nuevos usuarios y contraseñas, opera de manera independiente. Su función primordial es garantizar que la información ingresada al crear un nuevo usuario cumpla con los criterios de validez establecidos. Aunque no interactúa directamente con otras clases, desempeña un papel clave en la seguridad y coherencia del sistema. |

# Librerías usadas

# Commons-logging

La librería commons-logging-1.2 es parte del proyecto Apache Commons Logging, proporcionando una interfaz independiente de la implementación para el registro de eventos en aplicaciones Java. Permite cambiar entre diferentes sistemas de registro sin modificar el código. Incluye adaptadores para varios frameworks, soporta una jerarquía de registradores, es liviana y funciona en diversos entornos. Aunque ha sido ampliamente usado, algunos desarrolladores prefieren sistemas más modernos como SLF4J con implementaciones como Logback o Log4j2

.

**Commons-Lang**

La librería commons-lang3-3.8.1 es parte de Apache Commons Lang, que proporciona utilidades para mejorar las clases estándar de Java. Ofrece funciones para trabajar con cadenas, arrays, números, fechas, objetos, validaciones y reflexión.

**javax.mail y activation**

En nuestro código, aprovechamos estas bibliotecas para implementar funciones de seguridad y facilitar el envío de correos electrónicos de forma segura. Las clases y métodos proporcionados por javax.mail y activation permiten una gestión robusta de las operaciones de correo electrónico, brindando una interfaz intuitiva para construir, enviar los correos, y asegurando la comunicación electrónica se realice de manera fiable y segura.

**iText**

# iText es una biblioteca de Java para la manipulación de documentos PDF. Permite crear, modificar y gestionar archivos PDF en aplicaciones Java, facilitando tareas como la generación de informes, la manipulación de contenido y la gestión de formularios PDF. La versión específica 2.0.8 puede tener características particulares.

# Análisis de resultados

**Diseño Original y Estético de la Interfaz.**

Desde el inicio del proyecto, el equipo demostró un profundo interés en desarrollar una interfaz gráfica excepcional. Siempre mantuvimos un equilibrio del 50/50 entre la estética visual y la lógica del programa, construyendo gradualmente cada aspecto con atención y tiempo. Nos esforzamos por lograr que la aplicación se presentara como un programa diseñado de manera integral para su ciclo de vida. Al concluir el proyecto, experimentamos una gran satisfacción al entregar un producto en el que cada miembro contribuyó significativamente, desde los colores y las imágenes hasta las interacciones con los botones. Consideramos que estas adiciones aportaron un toque especial y pulido a nuestro proyecto.

**Generación de Tablas para Facilitar la Interacción del Usuario.**

Desarrollamos tablas para mejorar la interacción del usuario, siguiendo la sugerencia del profesor, quien propuso una idea atractiva y lógica. La implementación de estas tablas facilita la visualización de ingresos y la búsqueda de objetos específicos, brindando comodidad a los usuarios. Aunque fue un desafío, con tiempo y el respaldo del equipo logramos crear una herramienta efectiva y útil.

**Funciones Adicionales en Todo el Programa.**

Dedicamos tiempo a pensar lógicamente en cómo un usuario interactuaría con el software, identificando posibles vacíos y seleccionando herramientas para una experiencia de usuario más fluida. Priorizamos la integración de todas las funciones en un único espacio, evitando que el usuario tenga que navegar entre múltiples ventanas. Esta decisión no solo hace que el programa sea atractivo, sino también intuitivo y sencillo de usar. Quisimos asegurarnos de que todas las acciones del usuario se realicen de manera cohesionada y eficiente en un mismo entorno.

**Representación Gráfica de Detalles en la Facturación.**

La implementación de esta funcionalidad resultó desafiante, ya que no estábamos familiarizados con la creación de documentos desde Java. La dificultad principal radicó en lograr que la factura quedara bien proporcionada y que todos sus atributos fueran comprensibles. A pesar de esto, dedicamos esfuerzos considerables para superar estos desafíos y garantizar una representación clara y precisa en la generación de facturas.

**Reutilización Eficiente de Métodos en el Código.**

La reutilización eficiente de métodos en el código fue fundamental para optimizar nuestra implementación. Nos enfocamos en crear funciones versátiles que pudieran ser utilizadas en múltiples contextos, lo que no solo simplificó el desarrollo sino también mejoró la mantenibilidad del código.

**Lógica Principal del Programa de Forma Clara y Coherente.**

La lógica principal del programa se diseñó con claridad y coherencia como prioridad. Establecimos una estructura lógica que resultara intuitiva para el equipo y fácil de seguir, lo que facilitó la comprensión y modificación del código.

**Funcionalidades de Fácil Comprensión e Interacción.**

Diseñamos las funcionalidades con el objetivo de que fueran fácilmente comprensibles e interactivas. Nos esforzamos por hacer que cada función resultara llamativa y accesible para el usuario, promoviendo así una experiencia de usuario intuitiva y atractiva.