******

Universidad Latinoamericana de ciencia y tecnología

Curso Fundamentos de Programación

Proyecto Final

Manual técnico

Material elaborado por:

Karl Aase Blanco

Marjorie Davis

Carlos Gómez Iraheta

Profesor: Fernando Salas

# Índice

Contenido

[Índice 2](#_Toc16964875)

[Introducción 3](#_Toc16964876)

[Diagramas de Flujo 4](#_Toc16964877)

[Diccionarios de datos y explicaciones 5](#_Toc16964880)

[Clase: Contabilidad (Conta) 5](#_Toc16964881)

[Clase: Facturación (Fact) 7](#_Toc16964882)

[Clase: Clientes (cli) 11](#_Toc16964883)

# Introducción

Para este Proyecto, se debe hacer un programa con un uso productivo para alguna empresa real. Por dicha razón nos dimos a la tarea de buscar una empresa que requiera de un programa de software, encontramos una empresa conocida como “Estética canina S.O.S” la cual cuenta con dificultades a la hora de realizar la facturación para sus clientes. por lo que decidimos desarrollar un Sistema de facturación para esta empresa.

El objetivo principal es programar un sistema usando los temas vistos en clase. Adicionalmente debía tener las siguientes tareas:

* Menú principal
* Función para ingresar elementos diccionario.
* Función de ingreso datos
* Función para anular movimientos.
* Función para reporte

Todo debe realizarse usando objetos y clases, podrá almacenarse en archivos de texto.

Este documento estará subdividido entre cada una de las clases, adicionalmente habrá una sección donde se verán los diagramas de flujo.

Por último los requisitos de este manual técnico son:

* Realizar en Word.
* Un solo documento.
* Portada con los integrantes y nombre de la aplicación.
* Debe indicar los diagramas UML de la estructura utilizada y de los procesos dados.
* Debe detallar las variables y datos usados. (diccionario de datos)

# Diagramas de Flujo

## 

## 

# Diccionarios de datos y explicaciones

## Clase: Contabilidad (Conta)

**Variables (Diccionario de datos):**

* **static int Grooming\_parcial;**

Se utiliza para almacenar el dato de perros que atendió en la modalidad de “Grooming parcial “para después usar el método CalculoP\_parcial realizar el cálculo y así luego transferirle el dato de la ganancia a la variable Grooming\_parcialtotal;

* **static int Grooming\_T;**

Se utiliza para almacenar el dato de perros que atendió en la modalidad de “Grooming Total “para después usar el método CalculoP\_total realizar el cálculo y así luego transferirle el dato de la ganancia a la variable Grooming\_Ttotal;

* **static int monto\_adicional;**

Se utiliza para almacenar el dato del dinero adicional que pudo haber recibido en el método CalculoP\_Monto\_adicional para así transferirle el dato de la ganancia a la variable monto\_adicionaltotal;

* **static int gastos;**

Se utiliza para almacenar el dato del gasto que pudo haber recibido en el método CalculoP\_Gastos, para así transferirle el dato del gasto a la variable gastostotal;

* **static int Grooming\_parcialtotal;**

Se encarga de recibir y almacenar el dato del método CalculoP\_parcial para luego ser parte del cálculo en CalculoP\_ingreso\_total

* **static int Grooming\_Ttotal;**

Se encarga de recibir y almacenar el dato del método CalculoP\_total para luego ser parte del cálculo en CalculoP\_ingreso\_total

* **static int monto\_adicionaltotal;**

Se encarga de recibir y almacenar el dato del método CalculoP\_Monto\_adicional para luego ser parte del cálculo en CalculoP\_ingreso\_total

* **static int gastostotal;**

Se encarga de recibir y almacenar el dato del método CalculoP\_Gastos para luego ser parte del cálculo en CalculoP\_ingreso\_total

* **static int ingreso\_total;**

Se encarga de recibir y almacenar el dato del método CalculoP\_ingreso\_total

**Codigo (Metodos):**

* **CalculoP\_parcial()**

Se encarga de multiplicar la variable Grooming\_parcial \* 5000 y le transfiere el resultado a Grooming\_parcialtotal

* **CalculoP\_total()**

Se encarga de multiplicar la variable Grooming\_T \* 10000 y le transfiere el resultado a Grooming\_Ttotal

* **CalculoP\_Monto\_adicional()**

Se encarga de transferir el monto de la variable monto\_adicional a monto\_adicionaltotal

* **CalculoP\_Gastos()**

Se encarga de transferir el monto de la variable gastos a gastostotal

* **CalculoP\_ingreso\_total()**

Se encarga de realizar el cálculo sumando las variables Grooming\_parcialtotal, Grooming\_Ttotal, monto\_adicionaltotal y le restamos la variable gastostotal para luego retornar la variable ingreso\_total.

## Clase: Facturación (Fact)

**Variables (Diccionario de datos):**

* **static String strFecha;**

Esta variable se usara para ingresar la fecha de manera automática usando las librerías de “java.text.SimpleDateFormat;” y “java.util.Date;”.

* **static String Lugar;**

Esta variable se usara para almacenar el lugar de emisión de la factura.

* **static String Nombre;**

Esta variable se usara para almacenar el nombre del cliente al que se le hace la factura.

* **static String apellido;**

Esta variable se usara para almacenar el nombre del cliente al que se le hace la factura.

* **static int Numero\_cedula\_física\_o\_jurídica;**

Esta variable se usara para almacenar el número de cedula del cliente al que se le hace la factura.

* **static int Telefono;**

Esta variable se usara para almacenar el número de teléfono del que emite la factura.

* **static String Servicio;**

Esta variable se usara para almacenar el servicio que se le dio al cliente al que se le hace la factura.

* **static int costo;**

Esta variable se usara para almacenar el precio a cobrar al cliente al que se le hace la factura.

* **static double calculo\_iva;**

Esta variable se usara para almacenar el valor del impuesto IVA del cliente al que se le hace la factura.

* **static double Total;**

Esta variable se usara para almacenar el precio final a cobrar al cliente al que se le hace la factura.

* **static int opcion;**

Esta variable se usara para controlar la opción que selecciona el cliente.

**Codigo (Metodos):**

* **Stiempo()**

Este método solicita el tiempo mediante el uso de la librería “time” y “date”

* **Scorreo()**

Este método solicita el correo del usuario y lo registra en la variable correo

* **Sapellido()**

Este método solicita el apellido del usuario y lo registra en la variable apellido

* **SNombre()**

Este método solicita el nombre del usuario y lo registra en la variable Nombre

* **SNumero\_cedula\_física\_o\_jurídica()**

Este método solicita el número de cedula del usuario y lo registra en la variable cedula\_física\_o\_jurídica.

* **STelefono()**

Este método solicita el teléfono del usuario y lo registra en la variable Telefono

* **SLugar()**

Este método solicita el lugar de emisión de la factura y lo registra en la variable Lugar

* **solicitar\_datosServicio ();**

Este método solicita el servicio dado al cliente, y le da las dos opciones. Estas son:

Grooming parcial

Grooming Total

Dependiendo de lo que seleccione usara la variable costo y le dara un dato de tipo int especifico y cargara el servicio seleccionado como tipo string a la variable Servicio

* **SCalculoiva()**

Dependiendo cual servicio seleccionado en solicitar\_datosServicio se realizara un cálculo el precio del IVA y lo almacenara en double calculo\_iva.

* **SCalculofinal()**

Se encarga de sumar el costo de las variables servicio y el calculo\_IVA correspondiente para dar una suma total y almacenar ese dato en double Total.

* **generar\_factura()**

Este método se encarga de introducir los datos con la estructura dada en imprimir\_factura() mediante la librería Printwritter.

* **Imprimir\_factura();**

Este método imprime la factura. Junta todos los datos solicitados anteriormente y los une en un formato de factura especifico y lo introduce en un JTextArea para luego si el usuario decide que esta correcta pasar a usar el método pregenerar\_factura() y generar\_factura().

## Clase: Clientes (cli)

**Variables (Diccionario de datos):**

* **String[][] matriz = new String[20][6]**

Esta matriz se llama “lista” en la mayoría de los métodos que se verán mas adelante. Solo en el caso del menú, se llama “matrix”. En los demás métodos, se llama “lista”.

**Codigo (Metodos):**

* **Agregar (String cedula, String nombre, String domicilio, String tel, String email);**

Este es el método se usa al seleccionado botón “Agregar”, introduce datos de un Jtextfield en el Jtable. Este Recorre el Jtable e introduce los datos en un espacio vacío.

* **modificar(int index, String cedula, String nombre, String domicilio, String tel, String email);**

Este es el método se usa al seleccionado botón “Modificar”, introduce datos de un Jtextfield en el Jtable sí reconoce que este está en la tabla. Este Recorre el Jtable e introduce los nuevos datos.

* **eliminar(String cedula);**

Este es el método se usa al seleccionado botón “Eliminar”, Busca el usuario en la lista recorriéndola y dejando el espacio de los datos del usuario en vacío para que esta se oculte.

* **cargarMatriz(String lista[][]);**

Este método se usa para cargar una matriz además de cargar los clientes que se tenían antes de usar el programa. El usuario no tiene acceso a este método. Esto se puede cambiar usando bases de datos o cambiándolo en el código fuente.