



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA  

---

SEDE MEDELLÍN

## **Manual De Usuario Práctica 2**

### **Programación Orientada A Objetos**

#### **Autores:**

Leidy Daniela Alzate Flórez  
Juan Daniel Bula Isaza  
Óscar Andrés Úsuga Nanclares

#### **Entregado A:**

Jaime Alberto Guzmán Luna  
Alejandro Jiménez

4 de febrero de 2022

## Contenido

<b>1</b>	<b>DESCRIPCIÓN GENERAL DE LA SOLUCIÓN .....</b>	<b>2</b>
1.1	ANÁLISIS .....	2
1.2	DISEÑO .....	2
1.3	IMPLEMENTACIÓN .....	2
1.3.1	Capa de persistencia de datos .....	2
1.3.2	Capa lógica .....	3
1.3.3	Capa asociada a la interfaz del usuario .....	4
<b>2</b>	<b>DESCRIPCIÓN DEL DISEÑO ESTÁTICO DEL SISTEMA EN LA ESPECIFICACIÓN UML .....</b>	<b>5</b>
2.1	REALIZAR UNA VENTA.....	7
2.2	REALIZAR DEVOLUCIÓN.....	7
2.3	EJECUTAR CONTRATO DE CLIENTE .....	7
2.4	FINALIZAR CONTRATO DE CLIENTE.....	8
2.5	ENVIAR INFORMACIÓN MASIVA.....	8
<b>3</b>	<b>MANUAL DE USUARIO .....</b>	<b>8</b>
3.1	CÓMO ABRIR EL ARCHIVO EJECUTABLE .....	8
3.2	EJECUCIÓN DEL PROGRAMA .....	8
<b>4</b>	<b>BIBLIOGRAFÍA .....</b>	<b>12</b>

## **1 Descripción general de la solución**

### **1.1 Análisis**

El equipo 11 propone un aplicativo tipo CRM, que básicamente busca administrar clientes, mejorar la relación entre ellos o vender productos de forma sistematizada, bien sea por contratos o ventas individuales. La elección de dicha propuesta se hace efectiva al evidenciar el potencial y gran variedad de actores que presenta el dominio del problema, por ende, hacen que sea un dominio propenso para aplicar los diferentes aspectos de la programación orientada a objetos.

La traducción de CRM al español es: Gestión de relación con el cliente. No obstante, una mejor descripción del concepto puede ser: Software para la administración de la relación con los clientes, que también es utilizada para la organización y administración de los contactos de una empresa. Tiene como función principal ayudar a sus usuarios a centralizar toda la información de sus clientes, también organizar las actividades que deben cumplir con los mismos y recopilar información estadística sobre su desempeño. Claramente, la implementación de este tipo de aplicativos, solucionan diversos problemas en nuestra cotidianidad, y lo mejor, es que son problemas que se pueden atacar con la programación orientada a objetos, por esto es su gran importancia y su significativo auge en la actualidad.

### **1.2 Diseño**

Tenemos en cuenta las siguientes reglas de negocio:

- Los productos que se venderán serán de tipo tecnología y electrodomésticos.
- Habrá un número fijo de 10 posibles productos.
- El envío de mensajes será por medio de EMAIL, SMS o FAX.
- Las operaciones que ejecutar por medio del software son: Realizar venta, realizar devolución, ejecutar contrato de cliente, finalizar contrato de cliente y enviar información masiva.

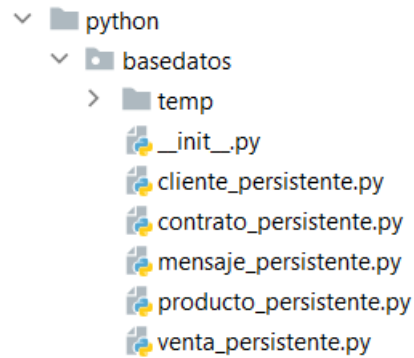
### **1.3 Implementación**

El programa fue implementado en el lenguaje Python para su compilación y ejecución. Para el desarrollo del código fuente se utilizó el IDE Pycharm, también, cabe resaltar que, para la resolución del trabajo colaborativo, se utilizó la tecnología de Git para repositorios locales y el sitio web de GitHub para el repositorio remoto.

El software está dividido en 3 aspectos importantes los cuales son:

#### **1.3.1 Capa de persistencia de datos**

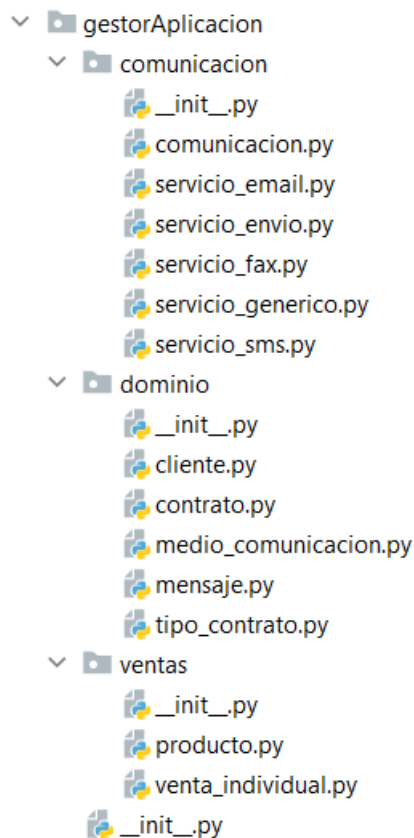
Una capa de persistencia encapsula el comportamiento necesario para mantener los objetos, en otras palabras, se encarga de leer, escribir y borrar objetos en el almacenamiento persistente (base de datos). La persistencia de la información es la parte más crítica en una aplicación de software. En esta capa encontraremos archivos **.txt** donde por medio de serialización de objetos (interfaz serializable de python) se guardan listas de estos. Esta capa se encuentra estructurada de la siguiente forma:



- El paquete temp contiene los archivos .txt donde se guardarán los streams de bytes de los objetos respectivos.
- Se desglosan los paquetes clientes y productos, donde están nombrados por cédula y código de producto respectivamente, también se guardan los mensajes en el paquete mensajes.txt.

### 1.3.2 Capa lógica

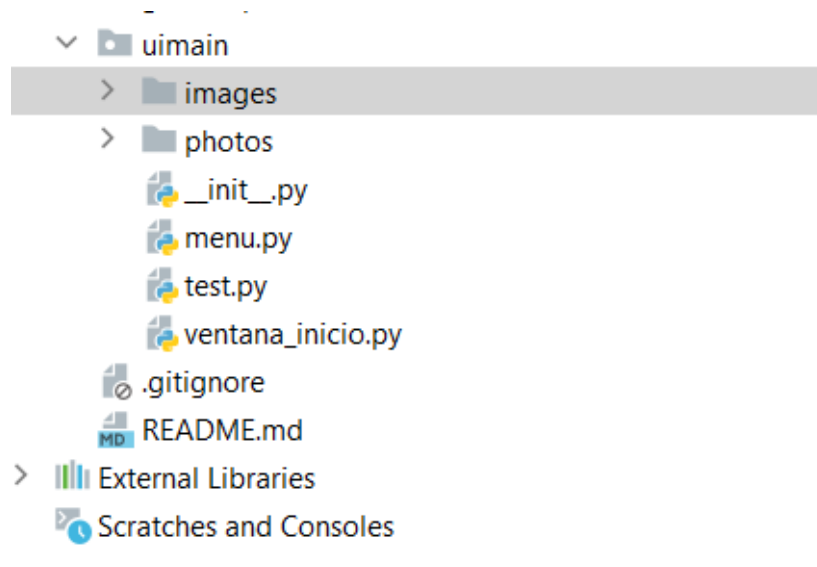
Esta capa ocupa un lugar primordial en la construcción de una infraestructura de software que nos permitirá el crecimiento y la extensión de servicios para todas las aplicaciones. Contiene todas las clases relevantes para el modelo de negocio, la cual está especificada en el diagrama UML que se encuentra más adelante del documento. Las clases mencionadas, contienen los atributos y métodos necesarios para cumplir con la tarea de nuestro CRM, su esquema es el siguiente:



De la figura anterior, podemos observar como la capa lógica está contenida en el paquete **gestorAplicacion**, que posteriormente se divide en tres subpaquetes, **comunicacion**, **dominio** y **ventas**.

### 1.3.3 Capa asociada a la interfaz del usuario

En esta etapa en particular, como característica principal de las interfaces, se especifica qué se debe hacer, pero no su implementación. Serán las clases las que se encargan de implementar estas interfaces, las cuales describen la lógica del comportamiento de los métodos. Esta sección, permite como tal, la interacción entre el programa y el usuario, utilizando el paquete “tkinter”, el cual es una fina capa orientada a objetos, proporciona un conjunto de herramientas robusto e independiente de la plataforma para administrar ventanas, la estructura de dicha capa esta estructurada de la siguiente maner:



Allí, se pueden observar 2 carpetas adicionales, una llamada “images” y la otra “photos”, las cuales contienen imágenes que fueron cargadas en la interface gráfica de usuario, el archivo `ventana_inicio.py` contiene los códigos asociados al paquete `tkinter`, que se encargan de ejecutar la ventana de inicio y la ventana principal de la aplicación CRM.

## 2 Descripción del diseño estático del sistema en la especificación UML

En el siguiente diagrama de clases, se encuentran especificados cada uno de los atributos asociados a su respectiva clase, junto con los métodos que implementan. Se evidencian 8 clases con múltiples relaciones y cardinalidades.

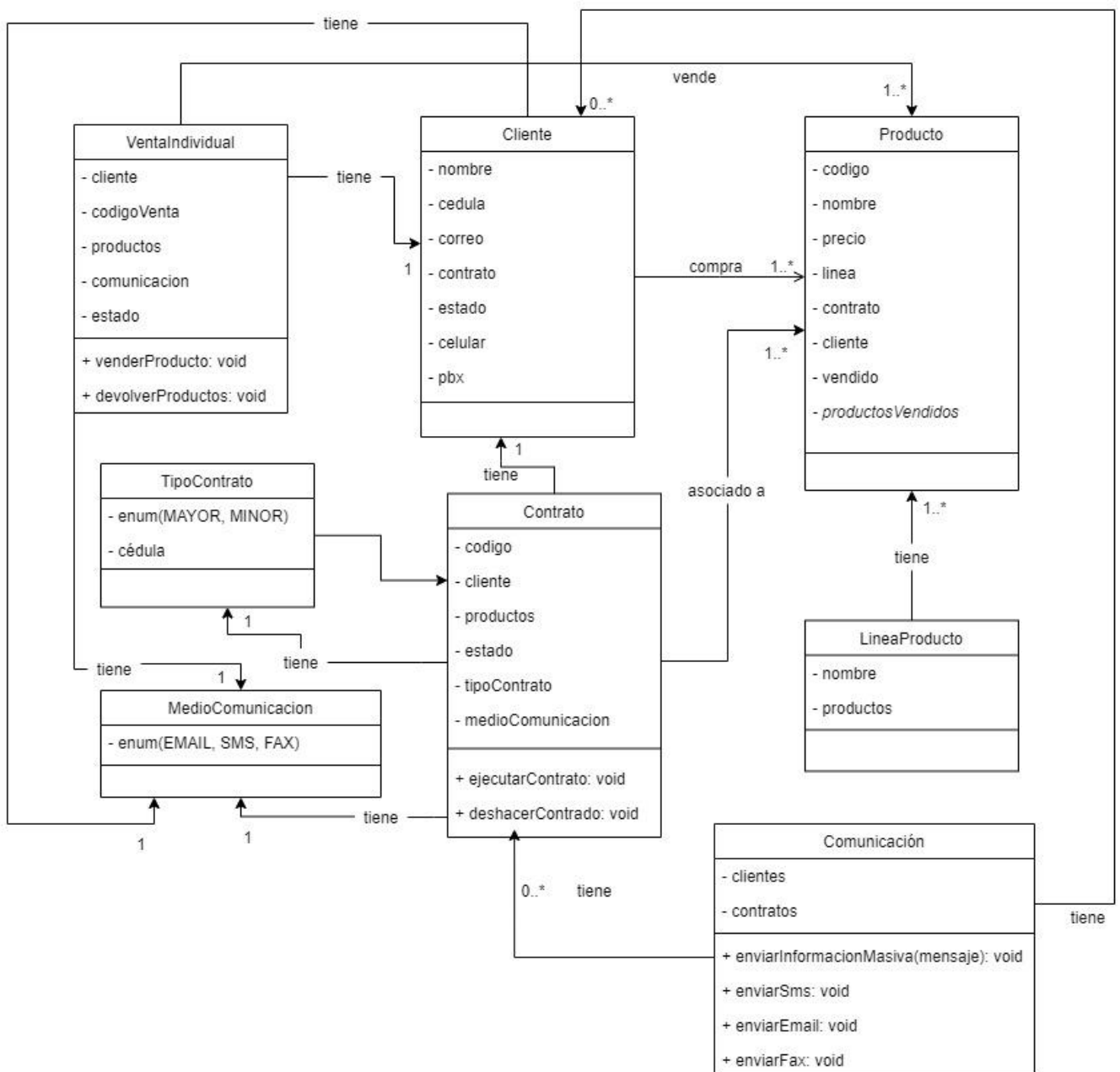


Diagrama de clases para el CMR en consideración

## ***Descripción de las funcionalidades implementadas***

### **2.1 Realizar una venta**

La venta, se define como el proceso de intercambio mediante el cual una parte, el vendedor, satisface la necesidad de otra, el comprador, en contrapartida de una suma de dinero o crédito. Para el caso de nuestro CRM, serán compras de artículos electrodomésticos y electrónicos, para ello, se requiere la interacción de objetos de tipo cliente y objeto, pues sin alguno de ellos, sería imposible realizar dicha función. No obstante, también se le asigna un código de venta y un estado, calificando si esta “pendiente” o “finalizada”.

En este procedimiento interactúan los objetos de la siguiente manera:

- Tenemos un objeto llamado ProductoPersistente que consulta todos los productos que están persistidos en los archivos.
- Tenemos un objeto llamado ClientePersistente que consulta desde archivos el cliente al cual se le ejecutará la venta.
- Creamos el objeto VentaIndividual al que se le asignan los productos y el cliente, se asigna el estado y queda con la información requerida para una venta.
- Tenemos el objeto VentaPersistente que almacena la información de la venta en archivos.

### **2.2 Realizar devolución**

Es la remesa, generalmente de mercancías, devuelta por un cliente a un proveedor, dicha funcionalidad, es muy común tenerla en cuenta en un CRM. Para la realización efectiva de esta función, se requieren de objetos tipo producto, cliente, además de características adicionales, si se desea asignar código único a la devolución, fecha, entre otros.

En nuestro sistema los objetos que interactúan son:

- VentaPersistente consulta la información de la venta que se quiere devolver.
- ProductoPersistente actualiza la información en inventario de los productos que están siendo devueltos dentro de esta venta.
- VentaIndividual ejecuta su método de instancia devolverProducto que elimina el producto de la lista existente, cambia el estado de la venta en caso de ser necesario y deja la devolución lista para ser almacenada.
- VentaPersistente actualiza el estado de la venta en el archivo.

### **2.3 Ejecutar contrato de cliente**

Contrato mediante el cual una persona se obliga con respecto a otra a realizar una serie de servicios a cambio de un bien (agregar a espacio de memoria). Para aplicación de este método, se deben tener en cuenta objetos de tipo cliente, objeto, contrato.

La interacción de los objetos se da de la siguiente manera:

- ClientePersistente obtiene el cliente para el cual se le asociará el contrato.
- ProductoPersistente obtiene los productos que serán asociados a este contrato.
- Contrato ejecuta su método de instancia ejecutarContrato que realiza la asociación del cliente con el contrato de forma bidireccional, asocia los productos y asigna el estado EJECUTADO para dicho contrato.

- ContratoPersistente guarda la información del contrato en archivos.
- ClientePersistente actualiza la información del cliente con el nuevo contrato asociado.

## 2.4 Finalizar contrato de cliente

Eliminar del espacio de memoria un contrato de cliente que ya fue realizado, liberar ese espacio para un posible futuro contrato. Para la ejecución de esta funcionalidad son necesarios objetos de tipo cliente, producto y contrato.

Los objetos en esta funcionalidad interactúan de la siguiente manera:

- ContratoPersistente obtiene la información del contrato a ser finalizado.
- Contrato ejecuta su método interno deshacerContrato el cual desliga al cliente existente en el contrato, también desliga al contrato del cliente y cambia el estado del contrato para que quede FINALIZADO.
- ContratoPersistente guarda la información del contrato.
- ClientePersistente actualiza la información del cliente desligando el contrato.

## 2.5 Enviar información masiva

Funcionalidad ligada a la comunicación, aquí se envían mensajes informativos a los clientes y contratos del sistema. Aquí interactúan los objetos ClientePersistente y ContratoPersistente para obtener toda la información de los clientes y contratos registrados en el sistema. Luego se recorre cada uno y la clase Comunicación hace un envío de los mensajes según el medio de comunicación que tenga cada cliente/contrato. Entre los medios de comunicación están FAX, EMAIL, SMS, TODOS y con base en estos medios de comunicación se envía un mensaje con un origen distinto. Finalmente se guardan estos mensajes en un archivo de texto para que el sistema tenga trazabilidad de los mensajes que han sido enviados durante la ejecución del sistema.

# 3 Manual de Usuario

## 3.1 Cómo abrir el archivo ejecutable

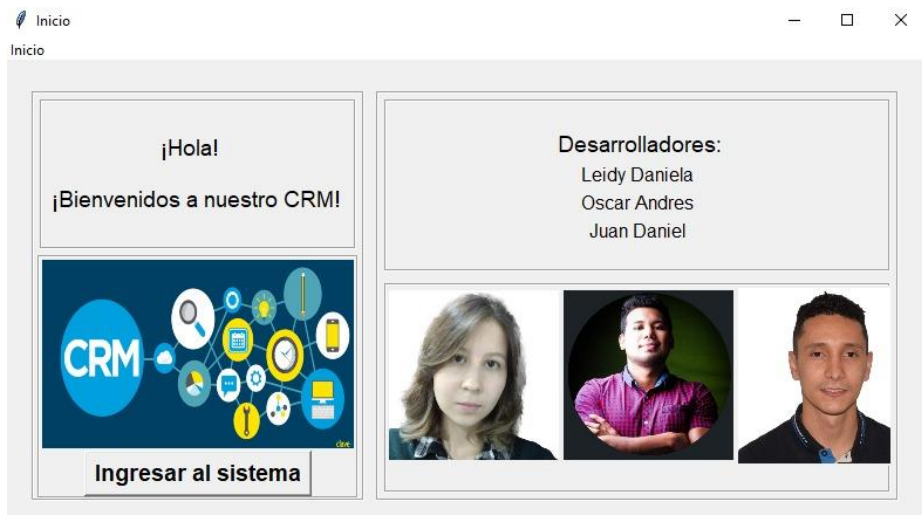
**Paso 1:** El usuario del aplicativo tendrá en su poder un archivo de extensión .zip, el cual debe descomprimir en alguna ruta de su computadora utilizando el descompresor de su preferencia.

**Paso 2:** Una vez los archivos han sido descomprimidos, busque al interior de la carpeta resultante el archivo ejecutable de nombre **start** y de doble clic sobre él para ejecutarlo.

## 3.2 Ejecución del programa

Una vez ejecutado el aplicativo, se tendrá la siguiente visualización de la ventana de inicio de nuestro CRM:





Observe que la ventana de inicio esta dividida en 4 sesiones, donde los dos cuadrantes del sector derecho dan a conocer una breve presentación de los desarrolladores del CRM, con las fotografías de ellos, cabe resaltar que al dar clic sobre el recuadro de los nombres, se despliega una breve biografía de cada uno de ellos, de la siguiente manera:

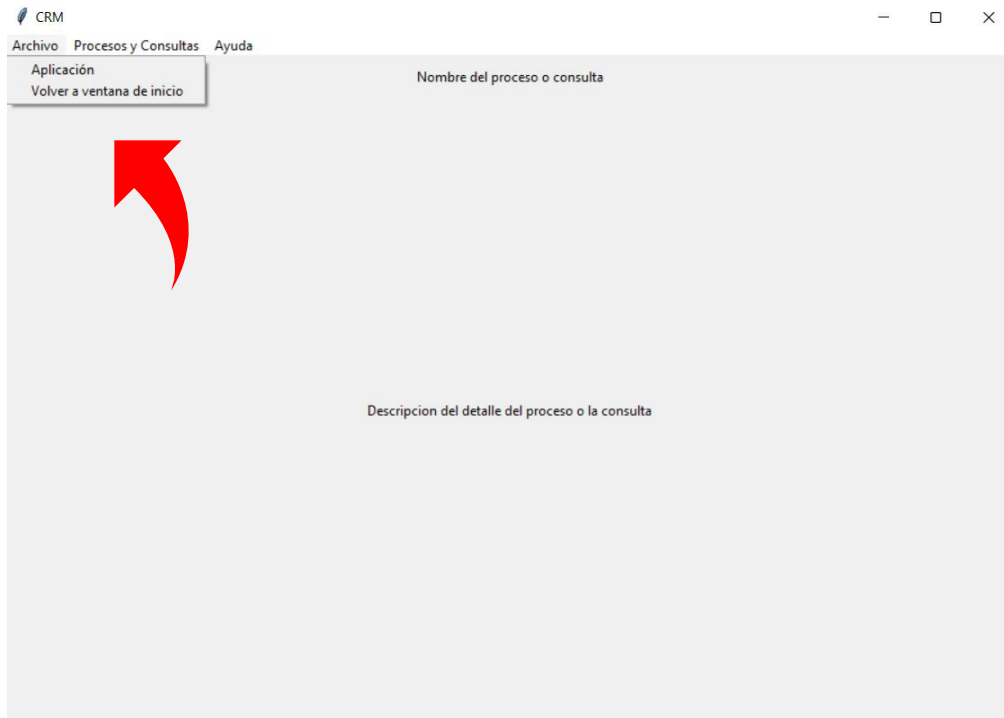


El menú de inicio que está presente en esta primera ventana despliega 2 opciones, las cuales son las siguientes:

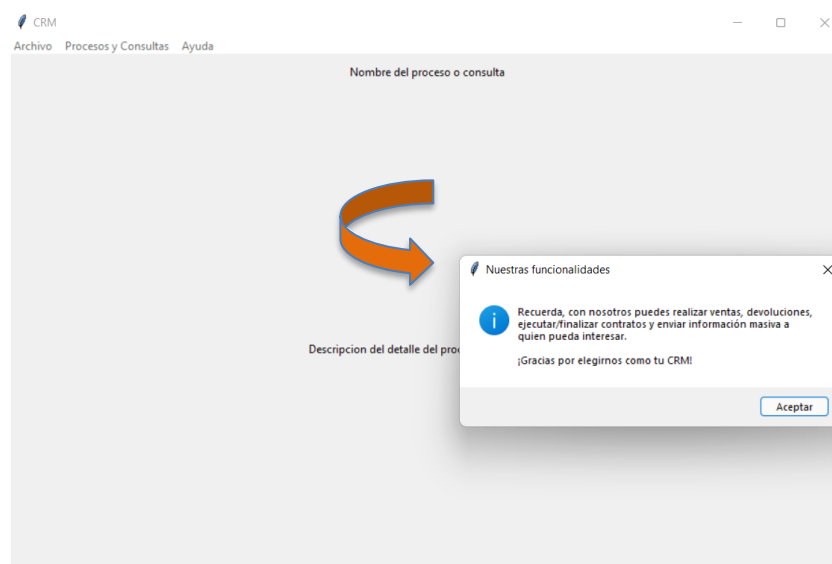


Donde al dar clic en “Nosotros”, se presentará una pequeña descripción de lo que puede realizar nuestro CRM, también allí, se encuentra la opción para salir de la aplicación.

Si se da clic al botón ingresar sistema, automáticamente se abrirá la ventana principal de la aplicación, donde encontrarás lo siguiente:

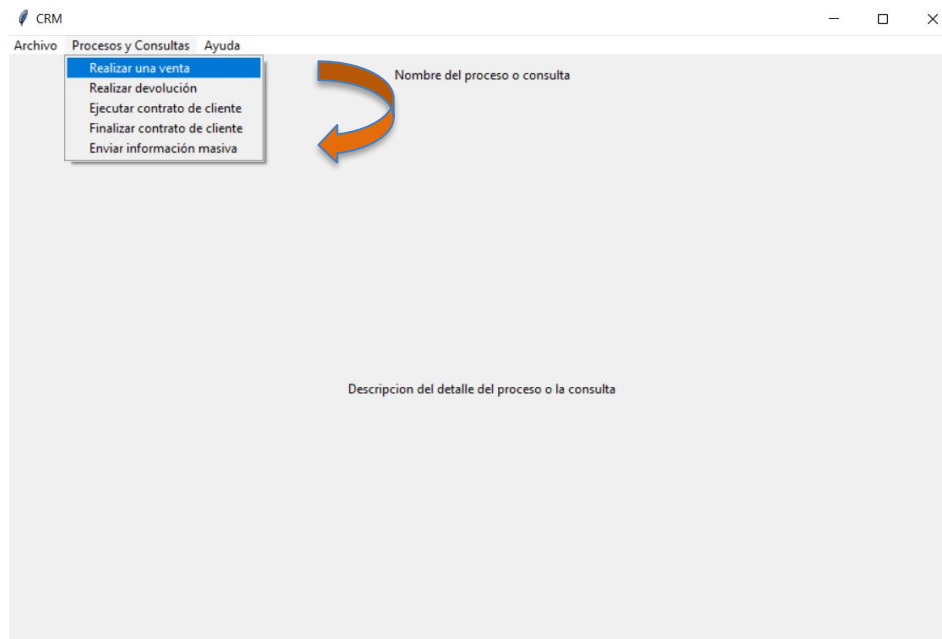


Donde se tendrá acceso al menú de la aplicación, al dar clic en la sección de archivo, saldrá un mensaje que recuerda las funcionalidades de nuestro sistema:



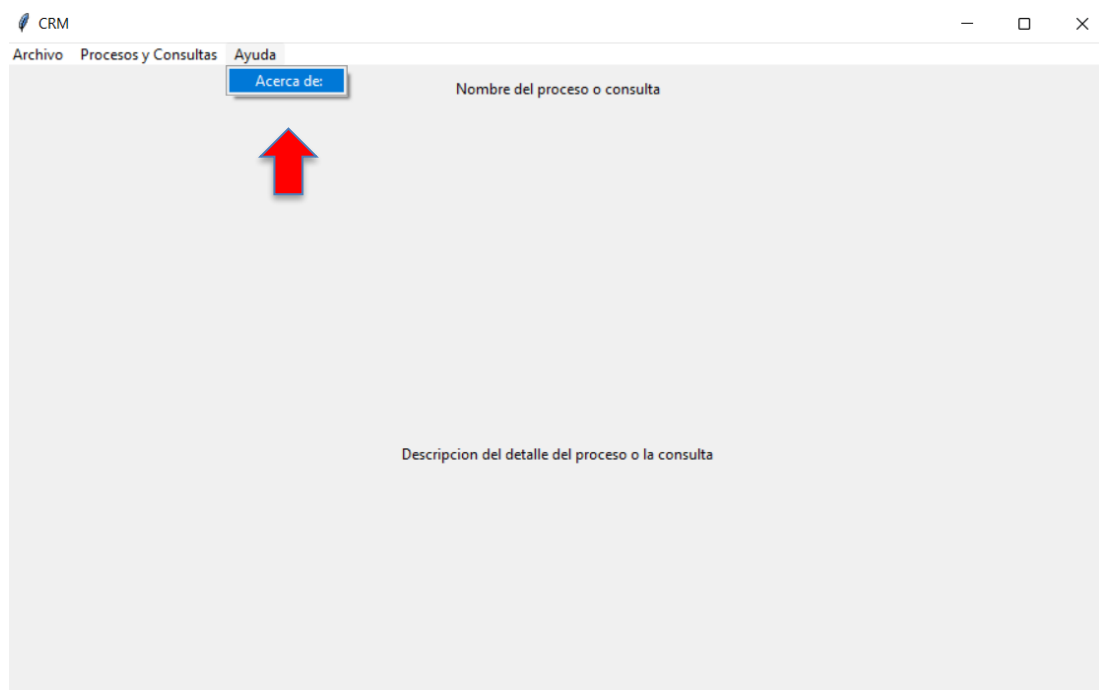
La otra opción de la sección archivo, es cuando el usuario desea volver a la ventana de inicio.

En la siguiente sección del menú, se encuentra procesos y consultas, allí es donde se ejecutarán las funcionalidades del CRM, dicha sección se presenta de la siguiente manera:



Donde se pueden observar, las 5 funcionalidades que pueden ser seleccionadas.

En la última sección del menú de la aplicación CRM, se encuentra la opción de ayuda, donde al dar clic, se despliega un ítem llamado “acerca de” la cual muestra una ventana de diálogo con los nombres de los autores de la aplicación, esto se puede visualizar de la siguiente manera:



De esta forma, termina nuestro manual, donde, en resumidas cuentas, se cuenta con 2 ventanas, una de inicio y otra de aplicación, que desarrollan una diversidad de acciones para interactuar con el usuario, pues este es el gran objetivo de las interfaces gráficas, con una metodología mucho más amigable, entrar en interacción con el usuario de forma directa, con el fin de ejecutar ciertas actividades.

#### **4 Bibliografía**

- GARCIA; I, (2001). CRM: Gestión de la relación con los clientes. Fc editorial. España.
- BRUNETTA; B, (2019). La experiencia del cliente: De la estrategia a la implementación. México
- Kurniawan, B. (2015). Java: A Beginner's Tutorial (4th Edition). BrainySoftware.