

**Tecnológico Superior de Jalisco**

**Unidad Académica Mascota**

**Ingeniería en Sistemas Computacionales**

**Materia:**

Fundamentos en Ingeniería de Software

**Actividad:**

Proyecto Integrador

**Nombre:** Nahum Isaias Hernandez Romero

**No. Control:**210110891

**Nombre del profesor:**

Alonso Rafael Peña Lara

**Fecha:**

30/11/2023

**Nombre de la aplicación:**

Software de Gestión de juguetería ElNayus's

### **Descripción de proyecto**

Las razones por la cual elegí hacer este software de gestión de juguetería son porque el mercado hay un alto nivel de competitividad por lo tanto las empresas se ven obligados a tener un software de gestión que esta es la mejor solución para tener un gestionamiento de recursos empresariales todo esto bajo el modelo de software como servicio y esto tiene muchos beneficios que aportaran a la empresa.

### **Justificación**

Dicho anteriormente con el alto nivel de competitividad las empresas se ven obligados a un software de gestión por lo tanto ayudar a una empresa con este software a que este le dé mejores beneficios también tener un mejor gestionamiento y también porque un software como estos para una empresa de este estilo le ayudara a reducir costes y aumentar al máximo su productividad.

### **Objetivos Generales**

Como primero el software será para el gestionamiento de una empresa de juguetería teniendo en cuenta el alto nivel de competitividad mi idea es hacer este software de gestionamiento para esta empresa que se dedica a los juguetes con esto ayudando reduzca costes y aumentar su productividad y aplicar los beneficios que estos da algunos beneficios serían los siguientes:

- Menor duración de los tiempos de fabricación.
- Reducción de costes.
- Optimización de los procesos internos.
- Optimización de la planificación de la fabricación.
- Previsión de necesidades.

### **Objetivos Particulares**

El objetivo de este software es crear varios apartados ya que ser un software para una empresa la misma empresa nos pedirá como es que desea su software mi idea es tener una forma de pago en línea con esto también tener proveedores para que esto pueda hacer a la empresa y por lo tanto en un futuro llegar a mejorarlo. Lo que hare es que primero hacer una sección para los clientes esta sección será como vender en línea una vez que el cliente tenga sus artículos enviarlo al software para que procesa al pago se creara una ficha de cliente para así poderlo pagar poner toda la información que se le pida y hacer la compra. Y ya en un futuro hacer un apartado para proveedores para que en este caso compren artículos como para reabastecer su tienda etc. Todo esto a un precio accesible para que la empresa pueda crecer.

### **Levantamiento de requerimientos**

Buscando en algunas páginas llegue a encontrar que con el alto nivel de competitividad las empresas hacer un software de gestión en este caso es de una juguetería pues sabiendo esto y entonces buscando llegue a tomar estos requerimientos ya que son muy importantes para este software hay que tener en cuenta cada uno de estos requerimientos para poder llegar acabo el software y así poderlo hacer bien y hacerlo a comodidad de los cliente también hay algunas cosas que se pueden cambiar ya que los requerimientos pueden variar depende a lo que ocupe las empresas.

### **Requerimientos funcionales**

#### **1. Gestión de Inventarios:**

- Registro de productos con detalles como nombre, descripción, categoría, proveedor, precio de compra, precio de venta, etc.
- Actualización automática de inventario después de cada venta.
- Alertas de stock bajo para productos con inventario bajo.

#### **2. Ventas:**

- Procesamiento de ventas, incluyendo la capacidad de escanear códigos de barras.
  - Registro de ventas para llevar un historial.
  - Generación de facturas y recibos.
3. Clientes:
- Registro de clientes con información como nombre, dirección, número de teléfono, correo electrónico, etc.
  - Seguimiento de historial de compras de clientes.
  - Programa de lealtad para recompensar a clientes frecuentes.
4. Proveedores:
- Registro de proveedores con detalles de contacto.
  - Seguimiento de pedidos y entregas de proveedores.
5. Reportes:
- Informes de ventas diarias, mensuales y anuales.
  - Informes de inventario, incluyendo productos más vendidos y menos vendidos.
  - Informes de rentabilidad.
6. Gestión de Empleados:
- Creación de perfiles de empleados con roles y permisos.
  - Registro de actividades de los empleados, como ventas procesadas o cambios en el inventario.
7. Seguridad:
- Control de acceso para garantizar que solo usuarios autorizados puedan realizar ciertas funciones.
  - Respaldo regular de datos para evitar pérdidas de información.

8. Integración con Canales de Venta Online:

- Si la juguetería vende productos en línea, integración con plataformas de comercio electrónico.

9. Gestión de Promociones y Descuentos:

- Capacidad para aplicar promociones y descuentos en productos.
- Seguimiento del rendimiento de promociones.

10. Gestión de Devoluciones:

- Proceso para manejar devoluciones de productos.
- Actualización automática del inventario después de una devolución.

11. Gestión de Cuentas por Pagar y Cuentas por Cobrar:

- Seguimiento de las cuentas por pagar a proveedores.
- Registro y seguimiento de las cuentas por cobrar de los clientes.

### **Requerimientos No Funcionales**

1. Usabilidad:

- La interfaz de usuario debe ser intuitiva y fácil de usar para los empleados.
- El sistema debe proporcionar capacitación fácilmente accesible para los nuevos usuarios.

2. Rendimiento:

- El sistema debe ser capaz de manejar un volumen específico de transacciones simultáneas, especialmente durante períodos de alta demanda, como días festivos o temporadas de compras.

- Los tiempos de respuesta del sistema deben ser rápidos para garantizar una experiencia fluida para los usuarios.
3. Escalabilidad:
- El software debe ser capaz de adaptarse y crecer para manejar un aumento en el número de productos, clientes y transacciones.
4. Seguridad:
- Debe haber medidas de seguridad robustas para proteger la información confidencial de los clientes, empleados y la propia empresa.
  - Se debe implementar un sistema de control de acceso para garantizar que solo usuarios autorizados puedan acceder a ciertas funciones.
5. Disponibilidad:
- El sistema debe estar disponible durante las horas de operación comercial, con un tiempo de inactividad mínimo para mantenimiento programado.
6. Confidencialidad y Privacidad:
- El software debe cumplir con las regulaciones de privacidad y protección de datos.
  - Debe haber controles de acceso adecuados para garantizar que la información confidencial esté protegida.
7. Compatibilidad:
- El software debe ser compatible con diversos dispositivos y sistemas operativos utilizados por la empresa.
8. Mantenibilidad:
- El sistema debe ser fácil de mantener y actualizar sin interrupciones significativas en las operaciones diarias.

- Deben documentarse los procedimientos de mantenimiento.
9. Portabilidad:
- El software debe ser capaz de ejecutarse en diferentes plataformas y entornos sin pérdida significativa de funcionalidad.
10. Cumplimiento Normativo:
- El sistema debe cumplir con las regulaciones y estándares legales aplicables a la industria y al manejo de datos.
11. Resiliencia:
- El software debe ser resistente a fallas, con la capacidad de recuperarse rápidamente en caso de interrupciones.
12. Costo:
- Debe haber un equilibrio adecuado entre las capacidades del software y los costos asociados con su implementación y mantenimiento.

### **Requerimientos de dominio**

1. Categorización de Productos:
- El sistema debe admitir la clasificación de productos por categorías específicas, como juguetes educativos, muñecas, juegos de construcción, etc.
2. Gestión de Edades Recomendadas:
- La capacidad de asignar edades recomendadas a los productos para ayudar en la segmentación y marketing.
3. Promociones Específicas para Juguetes:
- Funcionalidades específicas para gestionar promociones y descuentos en juguetes, considerando temporadas específicas como Navidad, cumpleaños, etc.
4. Compatibilidad con Normativas de Seguridad de Juguetes:

- El sistema debe tener en cuenta y permitir la gestión de la información relacionada con las normativas de seguridad de juguetes vigentes en la región.
5. Manejo de Tendencias y Ciclos de Juguetes:
    - La capacidad de gestionar el ciclo de vida de los juguetes, identificando productos de moda, tendencias y aquellos que están llegando al final de su popularidad.
  6. Gestión de Kits y Conjuntos:
    - Si la juguetería vende kits o conjuntos de juguetes, el sistema debe ser capaz de manejar la relación entre productos individuales y conjuntos.
  7. Información sobre Licencias y Personajes:
    - La capacidad de asociar productos con personajes de licencia y gestionar la información relacionada con acuerdos de licencia.
  8. Registro de Juguetes con Funciones Específicas:
    - Para juguetes educativos o tecnológicos, la capacidad de registrar información sobre las funciones específicas, como habilidades de aprendizaje, tecnologías incorporadas, etc.
  9. Seguimiento de Temporadas Específicas:
    - Funcionalidades para gestionar el inventario y las ventas durante temporadas específicas, como la vuelta al colegio o las temporadas de regalos.
  10. Personalización de Embalajes y Etiquetas:
    - Si se ofrecen servicios de personalización de embalajes o etiquetas, el sistema debe permitir la inclusión de esta información en las transacciones.
  11. Gestión de Garantías y Devoluciones Específicas para Juguetes:
    - Funcionalidades específicas para manejar garantías y devoluciones de productos, considerando las peculiaridades de los juguetes.
  12. Compatibilidad con Juguetes Multilingües:

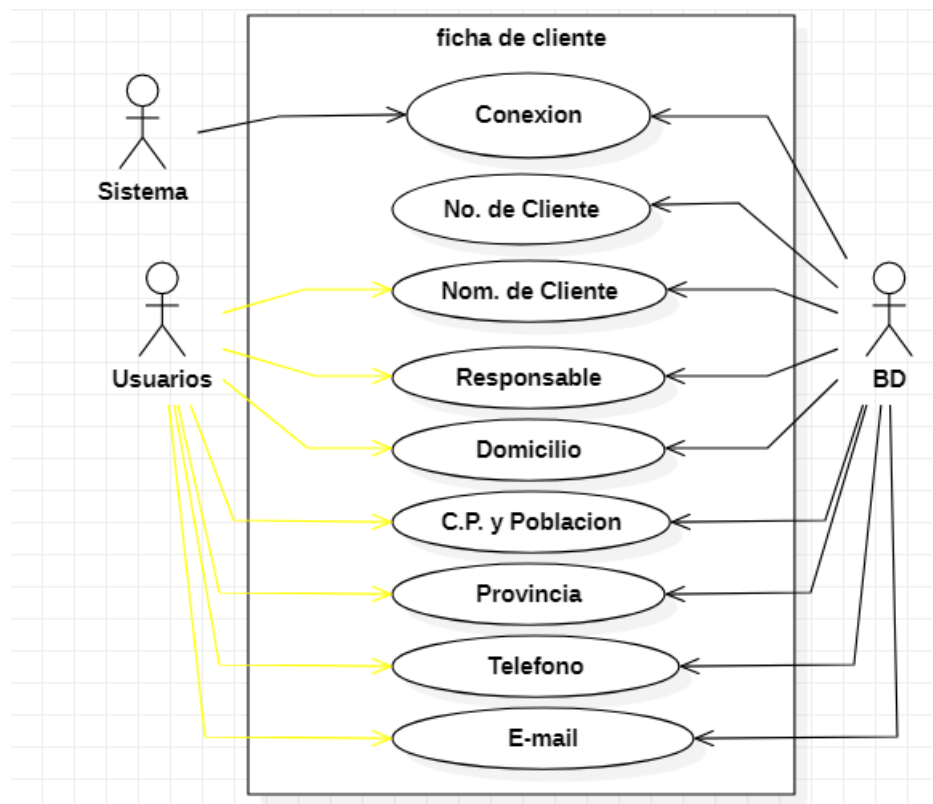


- Si la juguetería vende juguetes con información en varios idiomas, el sistema debe ser capaz de gestionar esta variabilidad lingüística.

### Diagrama de caso de usos

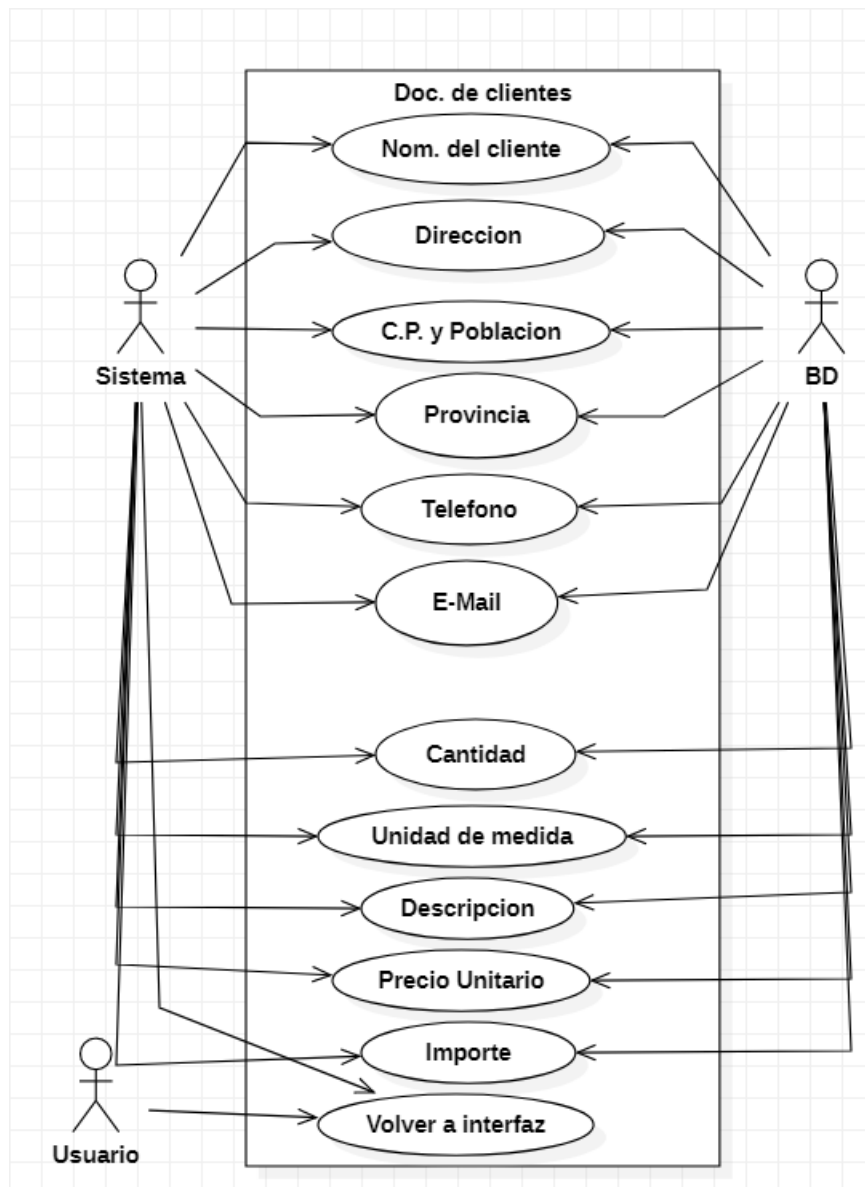
El diagrama de casos de uso es una forma de diagrama de comportamiento en lenguaje de modelado unificado (UML, del inglés Unified Modelling Language), con la que se representan procesos empresariales, así como sistemas y procesos de programación orientada a objetos. Por lo tanto, UML no es un lenguaje de programación, sino un lenguaje de modelado, es decir, un método estandarizado para representar sistemas planificados o ya existentes. En este diagrama, todos los objetos involucrados se estructuran y se relacionan entre sí. (lonos, 2020)

Diagrama de caso de uso Ficha de Cliente



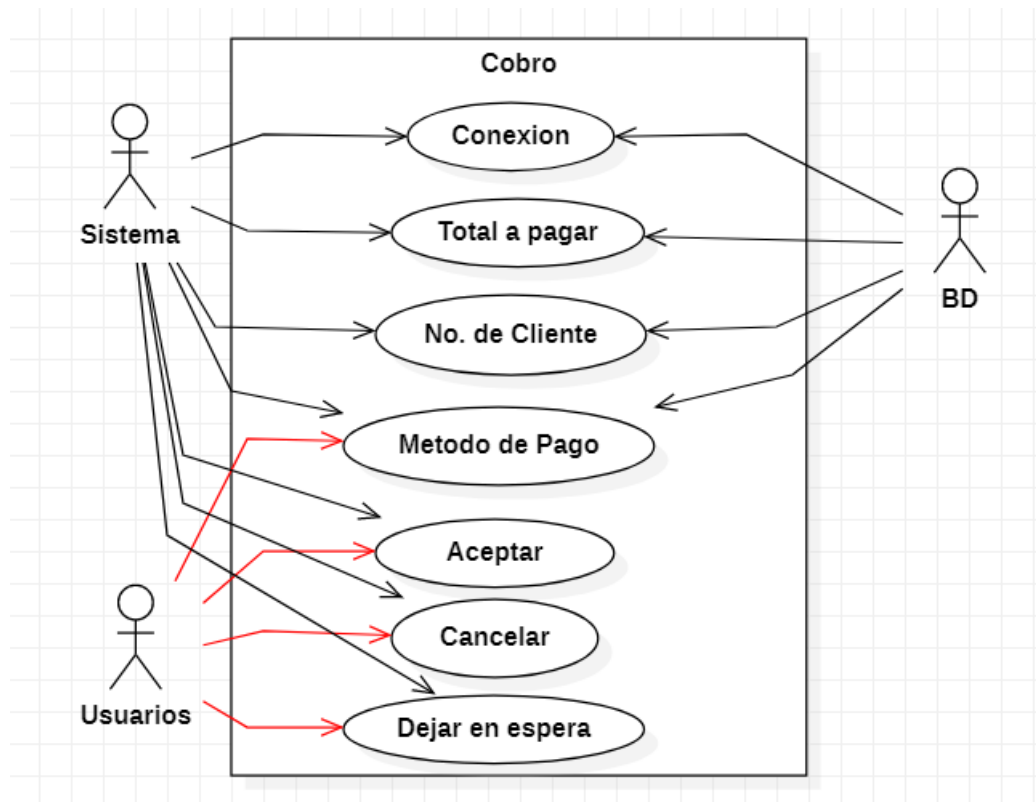
En ficha de cliente es por así decirlo donde el cliente se da de alta aquí es donde colocara todos sus para esto necesitamos tener la conexión con la base de datos ya que en esta es donde se guardarán los datos para usarlos en los apartados que siguen esto se tendrá que hacer antes de hacer otra acción ya que sin hacer esto nuestro software no va jalar a la perfección y por lo tanto algunos apartados saldrán en blanco.

Diagrama de caso de uso Doc. De clientes



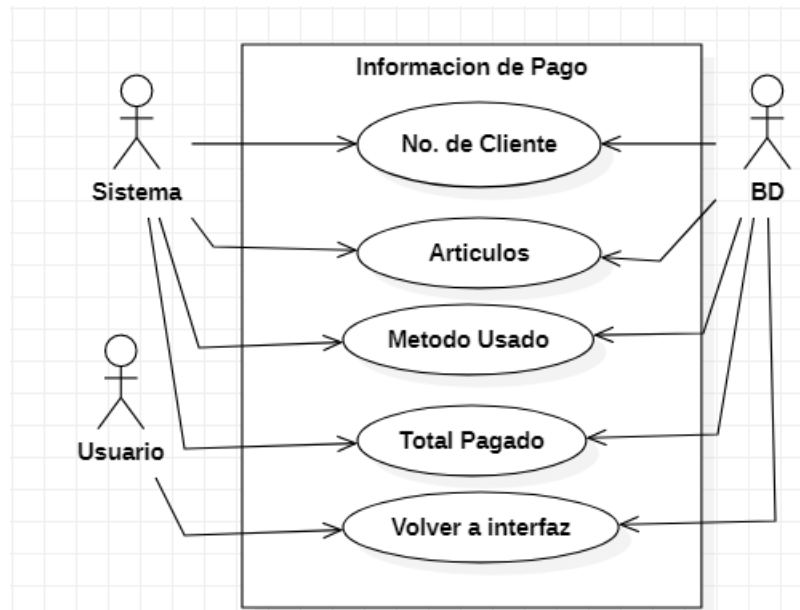
Después de que el usuario ponga sus datos necesitara revisar el documento de cliente para verificar que todos sus datos personales estén correctos y también verificar que los productos que vaya a comprar se encuentren aquí y que todo este perfecto como consecuente que todo este correcto estará un botón que nos regresara a la interfaz principal y aquí continuar con el cobro

Diagrama de caso de uso cobro



Como en todos necesitamos la conexión la base de datos nos dará la información sobre el total a pagar y el No. De cliente El usuario confirmará el Método de Pago como consecuente el usuario aquí decidirá si Aceptar Cancelar o Dejarlo en espera.

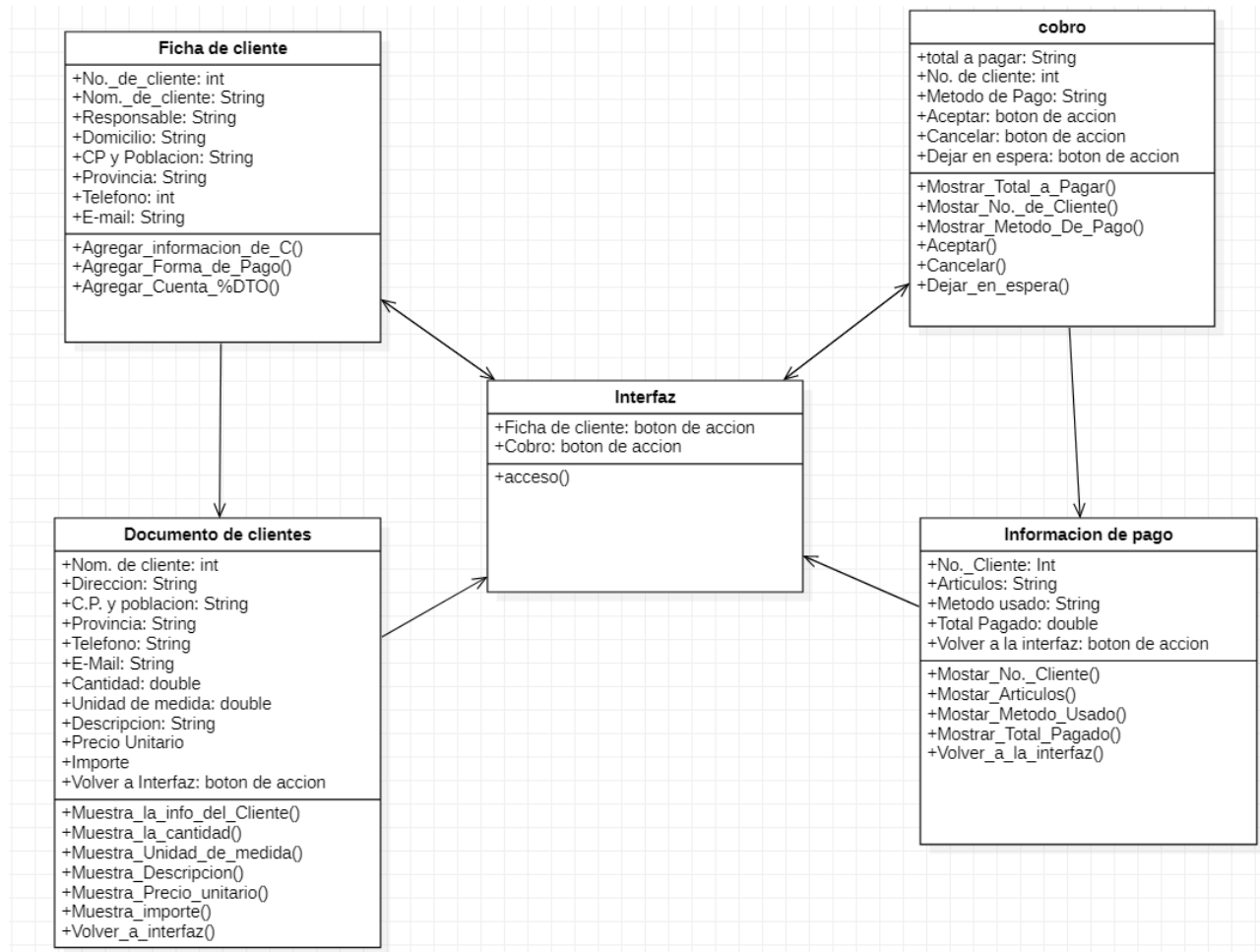
Diagrama de caso de uso Información de Pago



Por último la información de pago es donde nos darán todos los datos Como el número de nuestro cliente tanto como los artículos que el compro el método de pago que el uso el total que pago por los artículos y por ultimo un botón que su acción es que lo regrese a la interfaz.

### Diagrama de Clases

Los diagramas de clases son uno de varios tipos de diagramas estructurales de UML. Los diagramas estructurales muestran la estructura estática de un sistema, en lugar de mostrar cómo cambia un objeto con el tiempo. (miro, s.f.)

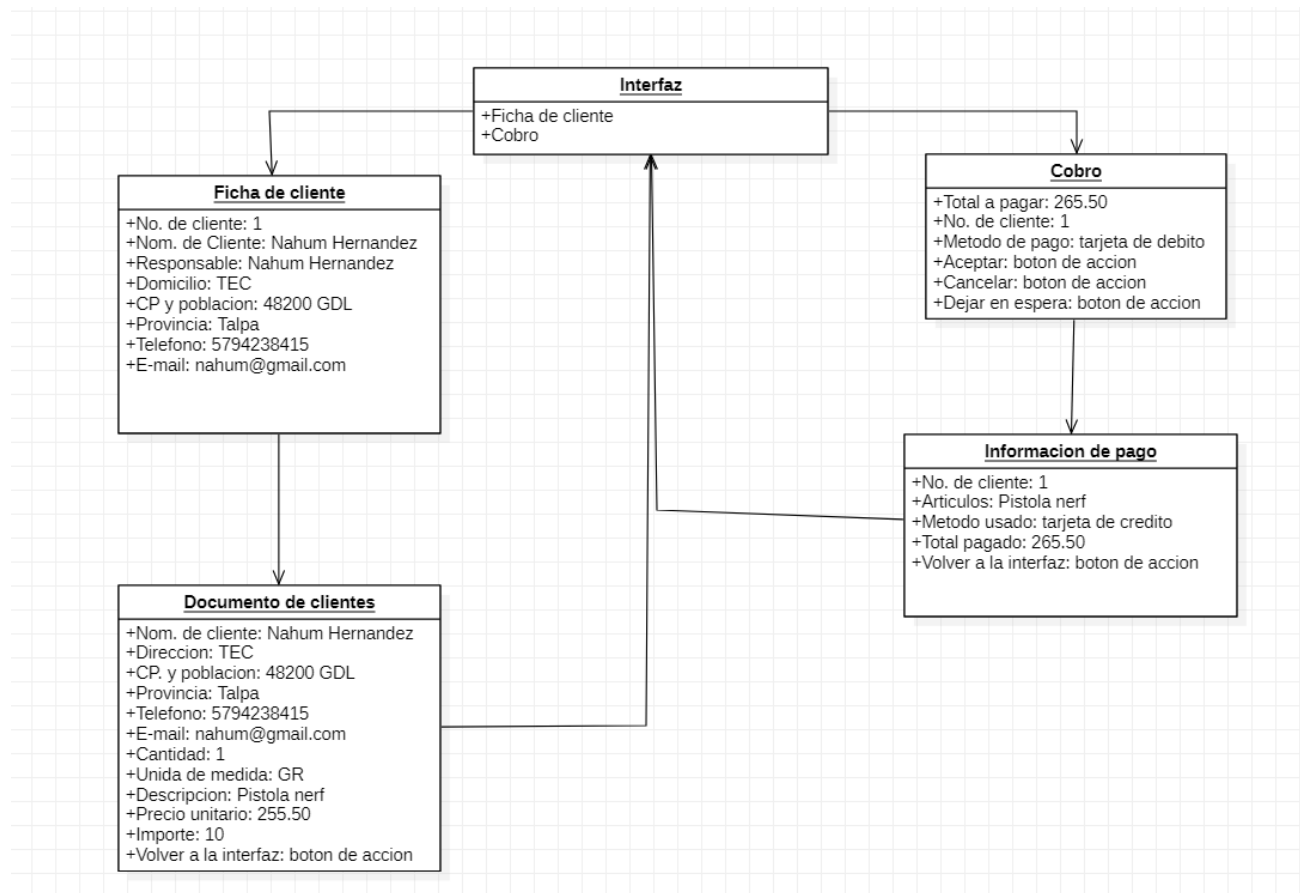


El software funciona de la siguiente manera Cuando ejecutemos o iniciemos el software lo primero nos aparece el apartado interfaz con esto ya antes había seleccionado el o los artículos que decidió comprar al entrar a la interfaz el usuario tendrá 2 botones que son ficha de cliente y cobro como primero el cliente tiene que darse de alta en ficha de cliente y ver que su información este correcta en documentos de clientes y de aquí regresar a la interfaz y seguir con el cobro sin el usuario no se dio de alta en la ficha de cliente el botón cobro no estará disponible sabiendo esto continuamos con el cobro que aquí nos aparecerá el total a pagar de los productos que eligió, elegirá un método de pago y aquí decidirá si cancela su compra la deja en espera o la compra, en caso de dar cancelar o dejar en espera nos regresara a la interfaz y en caso de seleccionar el botón aceptar se hará la

compra y como consecuente nos mandara a información pago que aquí estará toda la información de la compra como ultimo le apretara al botón de volver a la interfaz y seria todo.

## Diagrama de Objetos

El diagrama de objetos fue definido en la ahora obsoleta especificación UML 1.4.2 como «Un gráfico de instancias, incluyendo objetos y valores de datos. Cada diagrama de objetos representa una instancia de un diagrama de clase; muestra una fotografía del estado detallado de un sistema en un punto específico del tiempo». Esta misma especificación también afirma que el diagrama de objetos es «un diagrama de clase con objetos y no clases» (diagramas UML, s.f.)



## Conclusión

Esto es un trabajo que vamos a seguir usando en futuros semestres en este caso el profe nos pidió que hagamos un software de gestiones en el que nosotros íbamos a elegir de que iba a ser nuestros software y lo vamos estar trabajando hasta que esté listo en este caso fueron como los cimientos del software por así decirlos para en siguientes semestres complementarlo mas hasta poder lograr el objetivo plantado con este software puede ser que durante la creación haya cambios de opinión para que el software este más completo.

Como conclusión de la materia siento que reforcé y aprendí cosas que pueden a llegar a servir en un futuro como en el caso de hacer un programa ya tenemos como algo referente al que podemos ayudarnos para hacerlo como serian los diagramas de clase de objetos o los de casos de uso todo esto para tener la aplicación mejor planificada.

## Referencias

*diagramas UML*. (s.f.). Obtenido de <https://diagramasuml.com/objetos/>

*Ionos*. (24 de julio de 2020). Obtenido de <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>

*miro*. (s.f.). Obtenido de <https://miro.com/es/diagrama/que-es-diagrama-clases-uml/>