



Universidad
Rafael Landívar
Tradición Jesuita en Guatemala

Universidad Rafael Landívar
Facultad de Ingeniería
Análisis y Diseño II
Inge. Dhaby Eugenio Xiloj Curruchiche

Tema:

Informe de Proyecto Final Empresa
“RESTAURANTE DE COMIDA TÍPICA “CALLE REAL””

- Alex Omar Tzul Tax 1587217
- Elmer Gustavo Pú Tzunux 1535017
- Dominga Del Rosario Gómez Sac. 1643417
- Fredy Mat Xhun Mateo Juan 1571617

Repositorio:

<https://github.com/elmergustavo/SistemaPuntoVenta>

Quetzaltenango, 12/07/ 2021

INTRODUCCIÓN

En las empresas y microempresas poseen necesidades ya que a pasar el tiempo se van ampliando por lo que ya no les da abasto la forma en la que se venía trabajando para controlar todos sus servicios que prestan y por ende sus ganancias ya que dichas necesidades se vuelven la causa de muchos problemas. Una empresa grande posee problemas grandes.

Para empezar, se debe de realizar un análisis a la empresa en general con la única finalidad de estudiar, analizar y proponer nuevos métodos, cambiando procesos o implementando herramientas que ayuden en el desempeño de la empresa que se estudia y solucionar dichos problemas.

En el presente informe se realizó el análisis para la empresa Restaurante de comida típica “CALLE REAL” la cual está ubicada en 4ta calle D12-09 zona 1 Quetzaltenango, sector terciario o de servicio al cual se pertenece, dicha empresa se catalogó como empresa en crecimiento y desarrollo el propietario accedió a la propuesta planteada por los estudiantes del curso de análisis y Diseño.

Dichos estudiantes realizaron la correspondiente entrevista al propietario para conocer la empresa en si su funcionamiento en las actividades que se realiza a diario y mensual para poder analizar lo que realmente se necesita para implementar y así automatizar sus actividades para luego realizar los estudios correspondientes según el temario del curso.

DATOS GENERALES

Nombre de la empresa

Restaurante de comida típica Calle Real

Ubicación

4ta calle D12-09 zona 1 Quetzaltenango, sector terciario o de servicio al cual se pertenece

Contacto

7725-0397, Julio Francisco, Chef a cargo

Inscripción

Pequeño contribuyente

Descripción general

Calle Real es un restaurante ubicado en la zona 1 de la ciudad de Quetzaltenango, con dos años de funcionamiento desde su fundación, inspirado en las raíces guatemaltecas su entorno es totalmente colonial, con decoraciones preponderantes regionales, mostrando algo de cultura nacional y gastronómica. Actualmente el menú es estacionario y variado, tomando gastronomía extranjera y nacional, pero como eje principal la nacional, dando platillos preponderantes de la región aledaña del departamento, tomando como base recetas caceras y brindando una atención como si estuviera en el comedor de su hogar. El nombre restaurante y comida típica quiso dar a conocer que en dicha empresa no solo se pueden encontrar platillos regionales guatemaltecos (Comida típica), sino también una variedad de platillos demandantes que no son tradicionales dentro de la gastronomía meramente guatemalteca, por ello lleva al frente del nombre la palabra "Restaurante"

Breve Descripción de que tratará el proyecto

Nos enfocaremos en crear un sistema para el control, gestión y operación del Restaurante Calle Real, el cual se encuentra ubicado en 4ta calle D12-09 zona 1 Quetzaltenango, este sistema tiene como objetivo el poder ser utilizado para uso diario del restaurante mientras desarrolla sus operaciones, desde el registro del pedido ya sea para llevar o consumir en el local, así mismo el cobro de ellos y el apartado de facturación, también el registro y control de compras, almacenar la información de clientes, proveedores, ingredientes, recetas, así mismo realizar búsquedas de criterios según campos como direcciones, nombres, teléfonos, el sistema también podrá mostrar o actualizar el inventario. Para ingresar a dicho software y un apartado para administrar a los usuarios (crear, modificar o eliminar).

Personas que interactúa con el sistema:

Administrador/Dueño

Módulos en que se divide el proyecto y explicación sobre su función de cada uno:

Módulo 1: Ventas

En este módulo se realizarán las ventas en la caja, tanto las ventas rápidas como las ventas del servicio para llevar en el restaurante

Módulo 2: Inventario

Llevaremos a cabo el control de inventario de la empresa sobre el producto que ingresa, así como los avisos cuando falten productos

Módulo 3: Proveedores

Llevaremos el control de los proveedores y sus respectivos pagos, si estamos al día o no, también realizaremos pedidos

Módulo 4: Cotizaciones

El cual consiste en poder crear un documento con la información para el solicitante acerca de los precios especiales y/o servicio de catering

Módulo 5: Administración Financiera

Controlar los cobros recibidos y gestionar los impuestos pagados o por pagar

Módulo 6: Reportes y Estadísticos

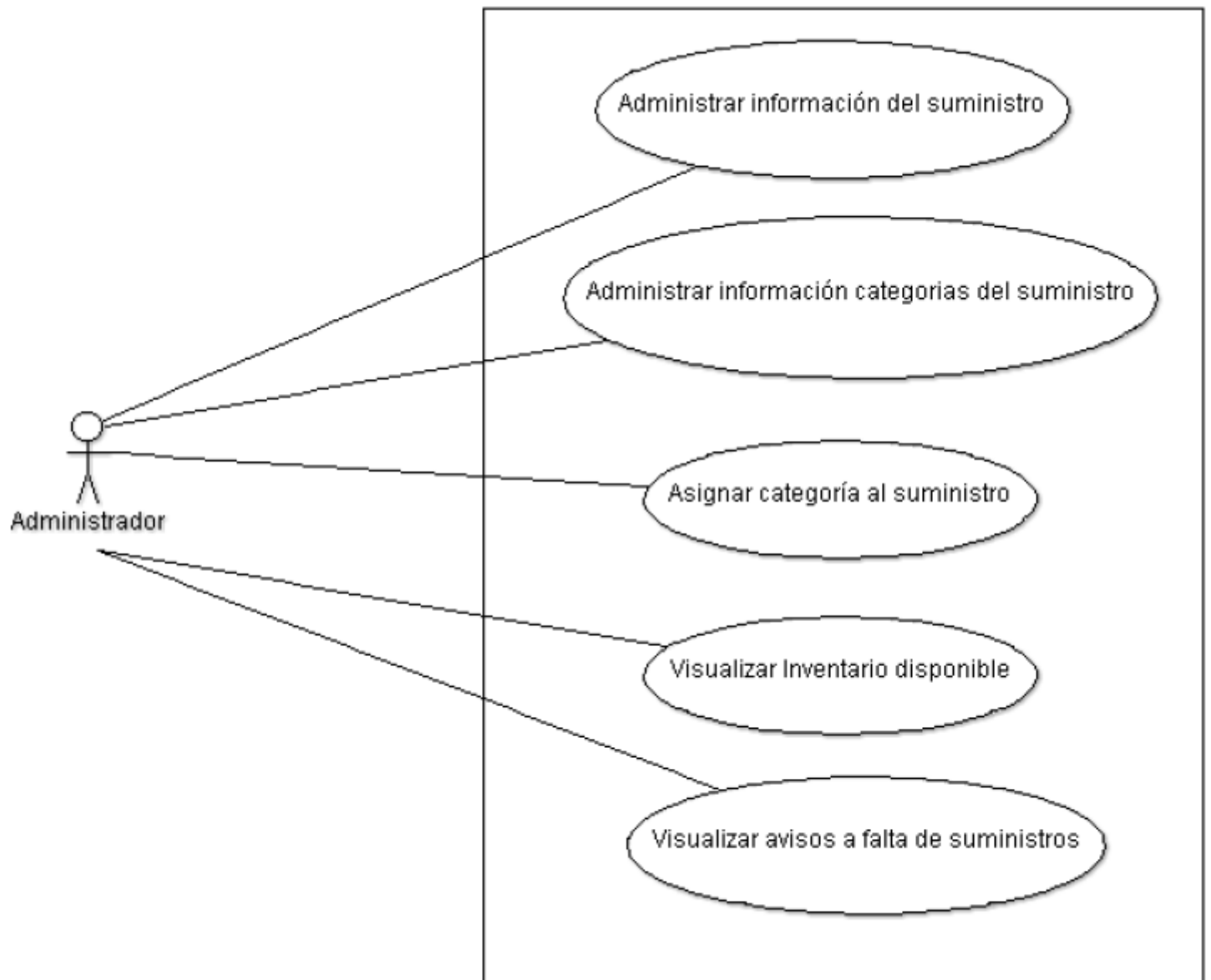
Poder generar reportes de ventas o compras de fechas en específico, así mismos reportes de platillos más vendidos, o generar estadísticos de estos.

DIAGRAMAS DE CASO DE USO

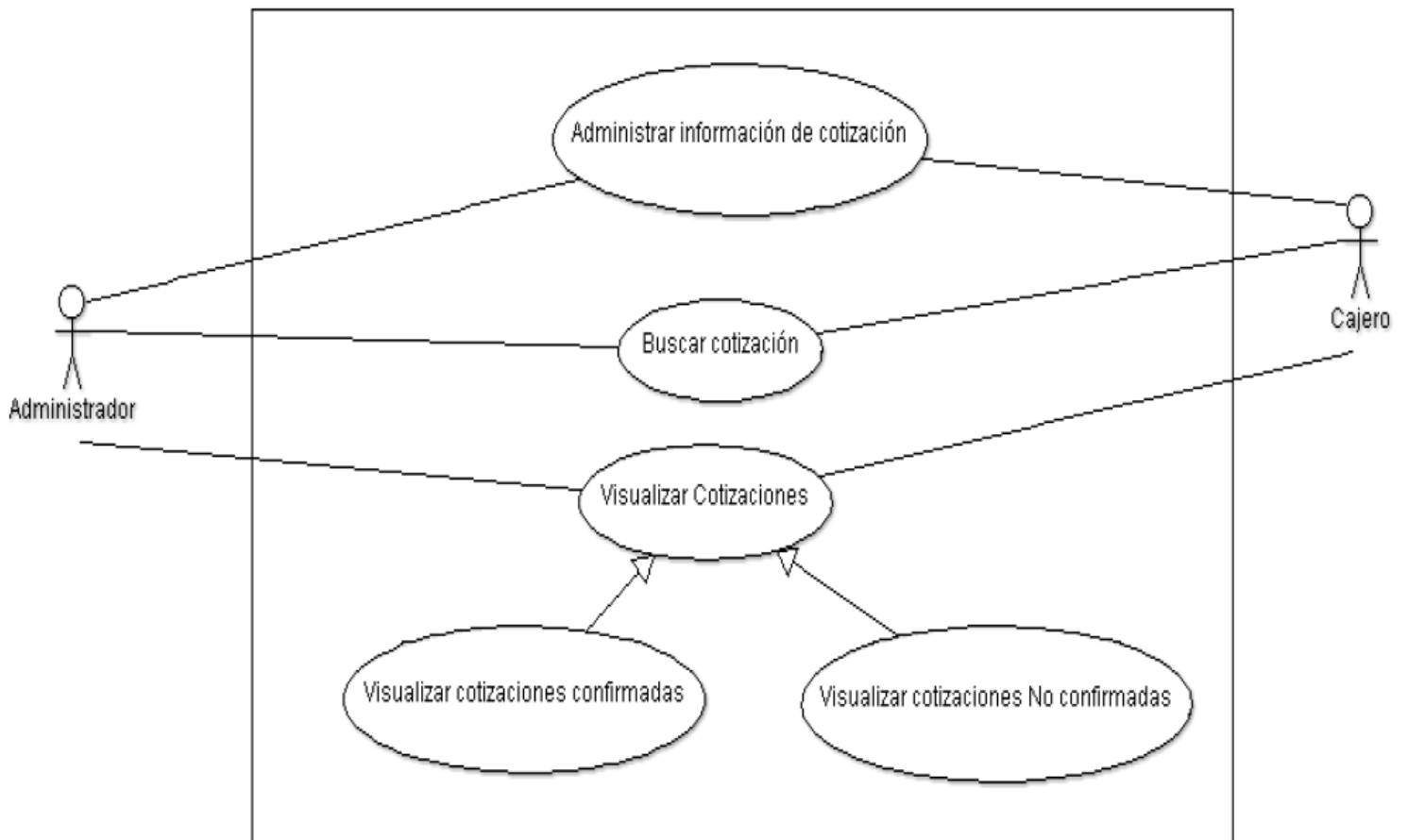
Caso de Uso: Modulo Venta



Caso de Uso: Modulo Inventario



Caso de Uso: Modulo Cotización



Caso de Uso: Modulo Administración Financiera

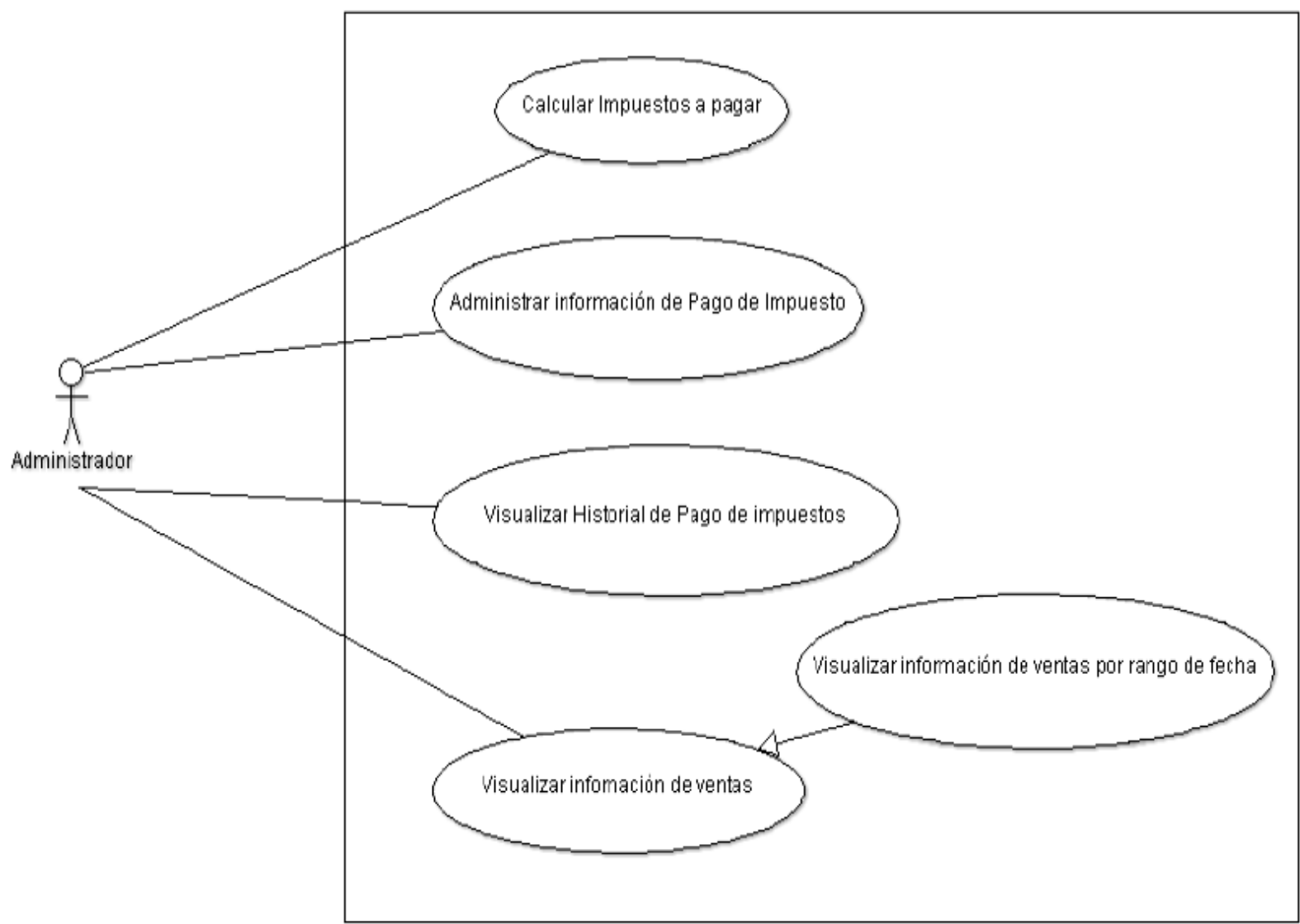


DIAGRAMA DE BASE DE DATOS (ENTIDAD-RELACIÓN)

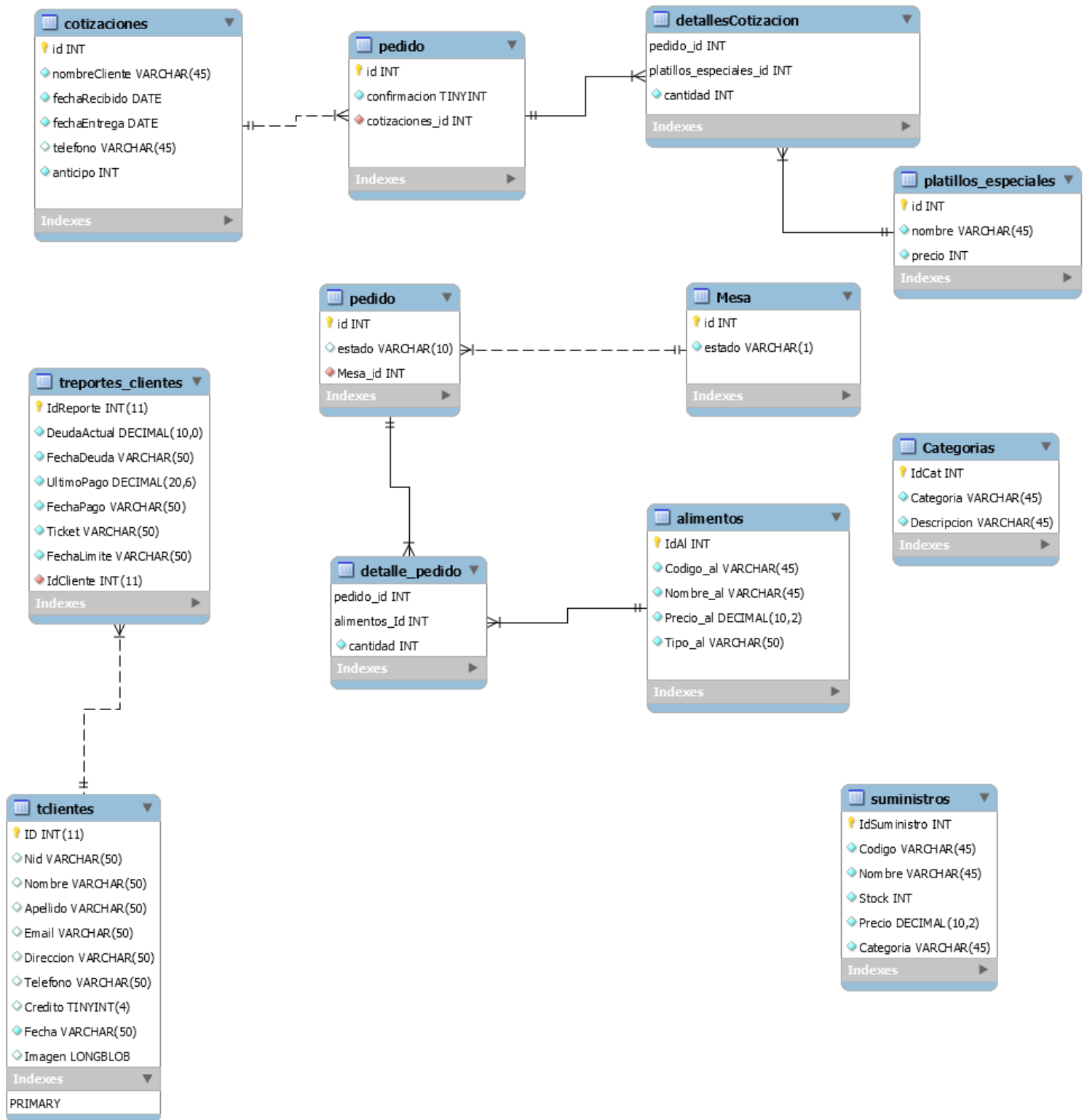


DIAGRAMA DE ACTIVIDAD

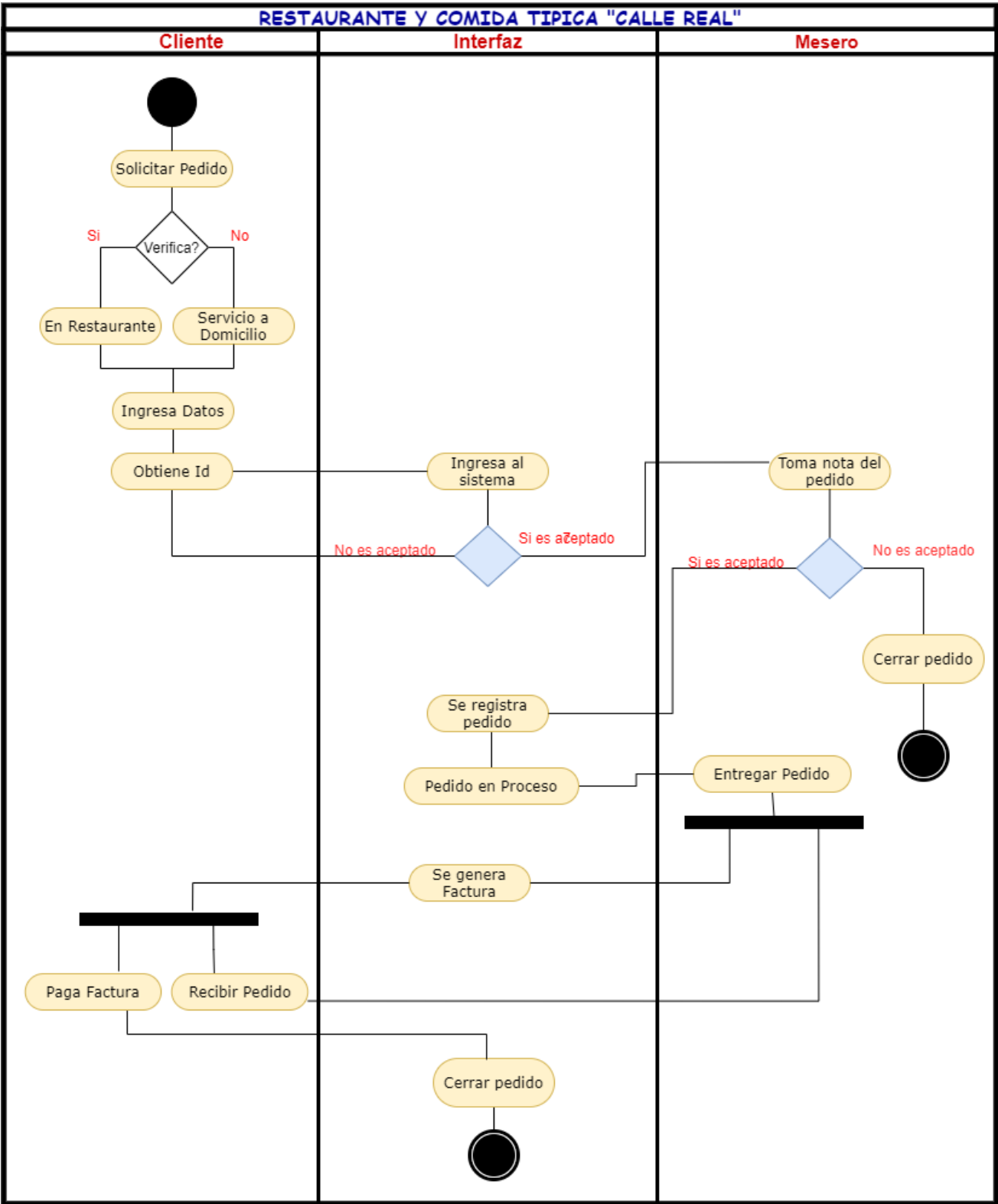


DIAGRAMA DE SECUENCIA

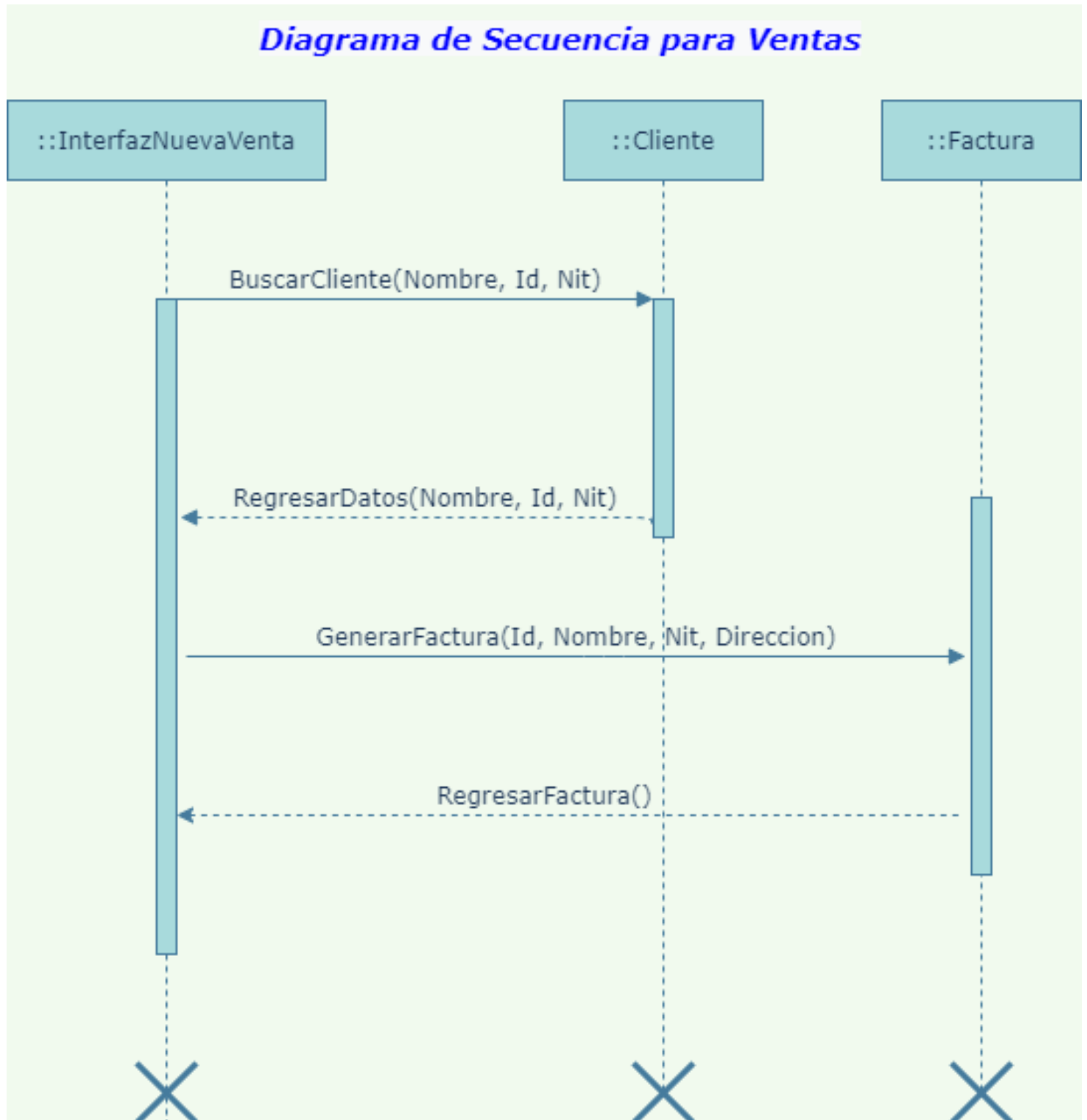


Diagrama de Secuencia para Inventario

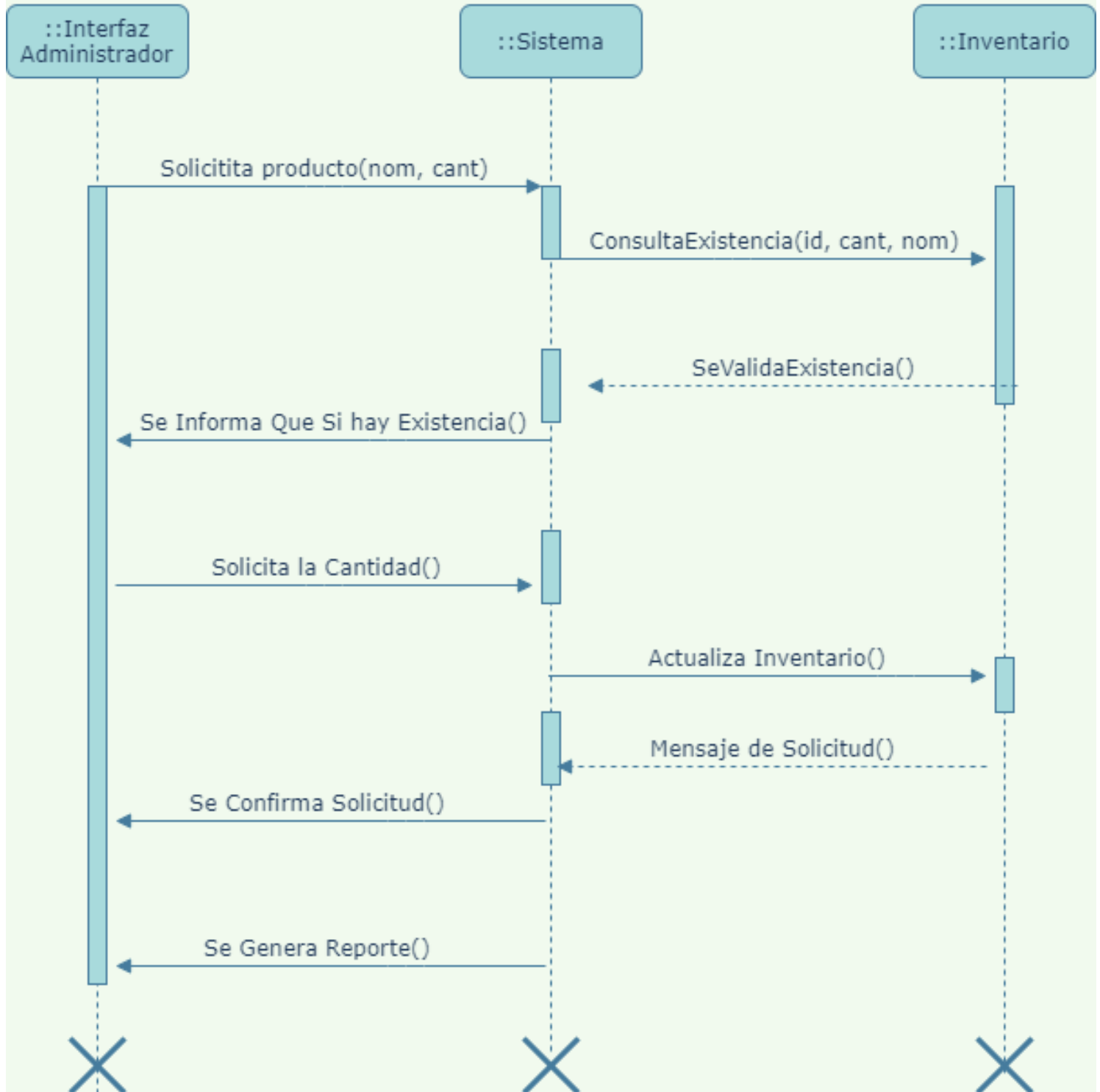
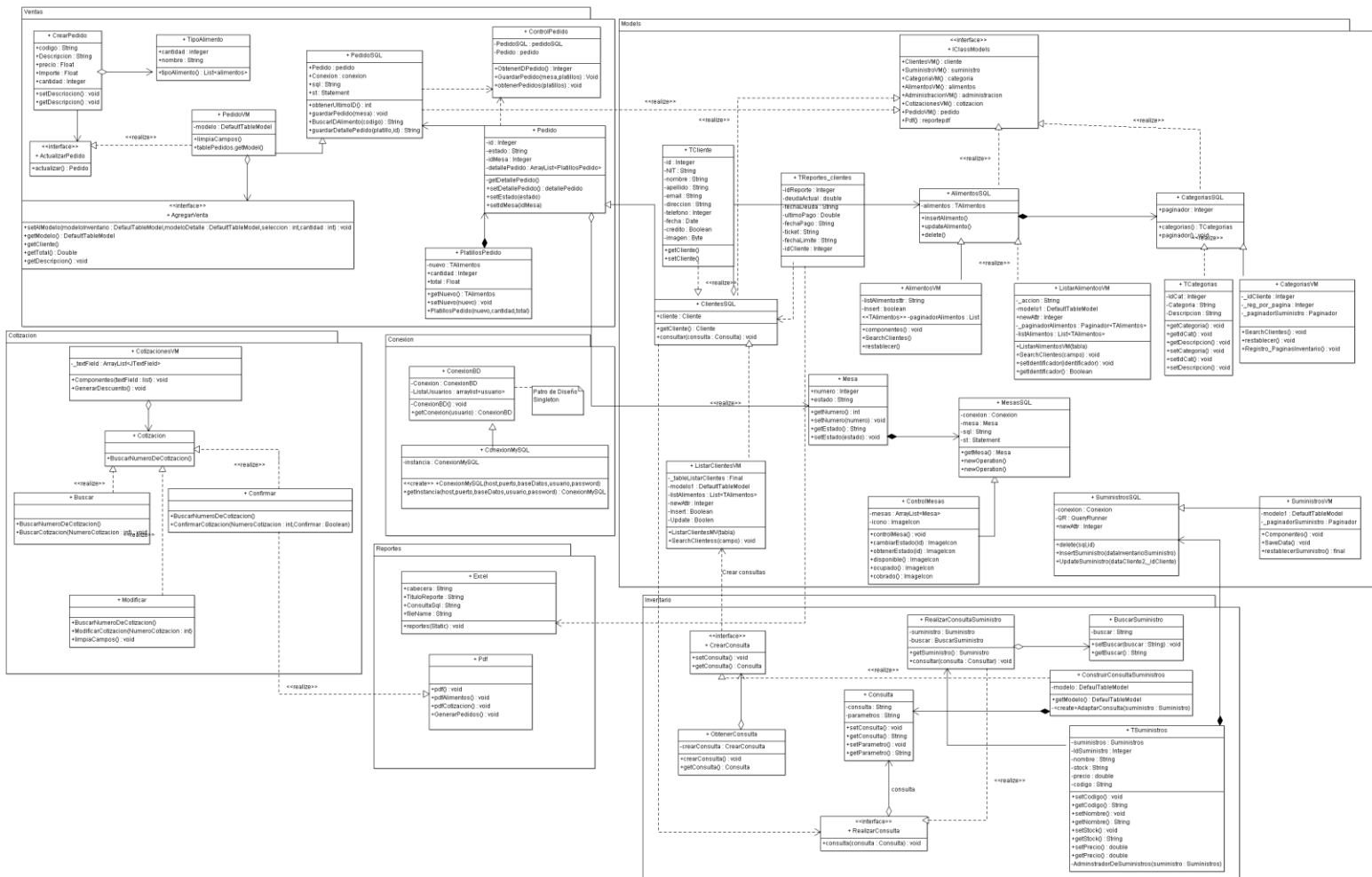
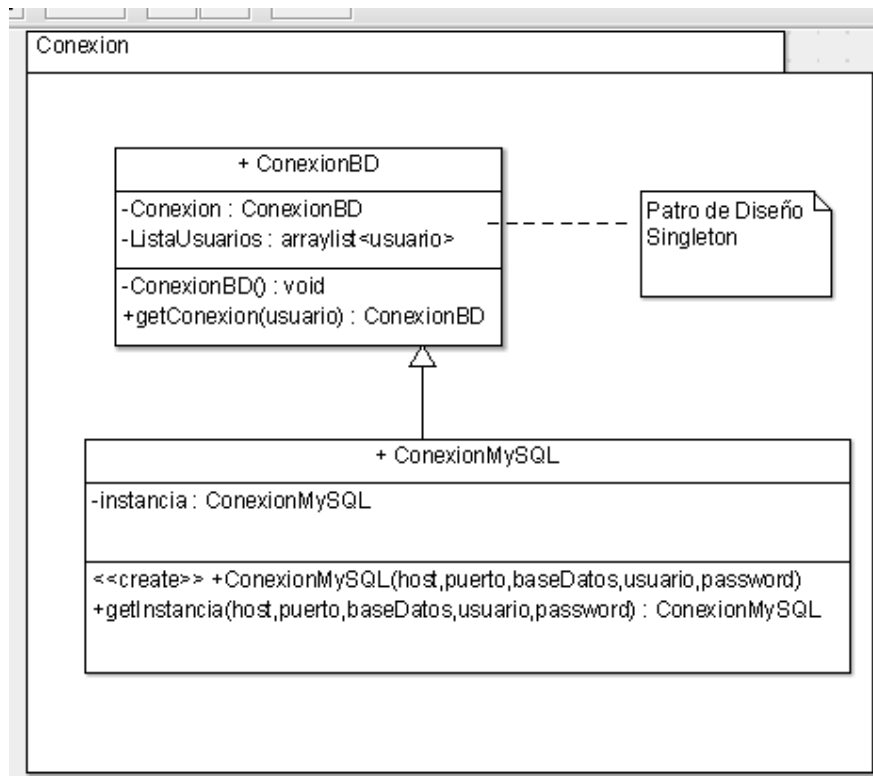


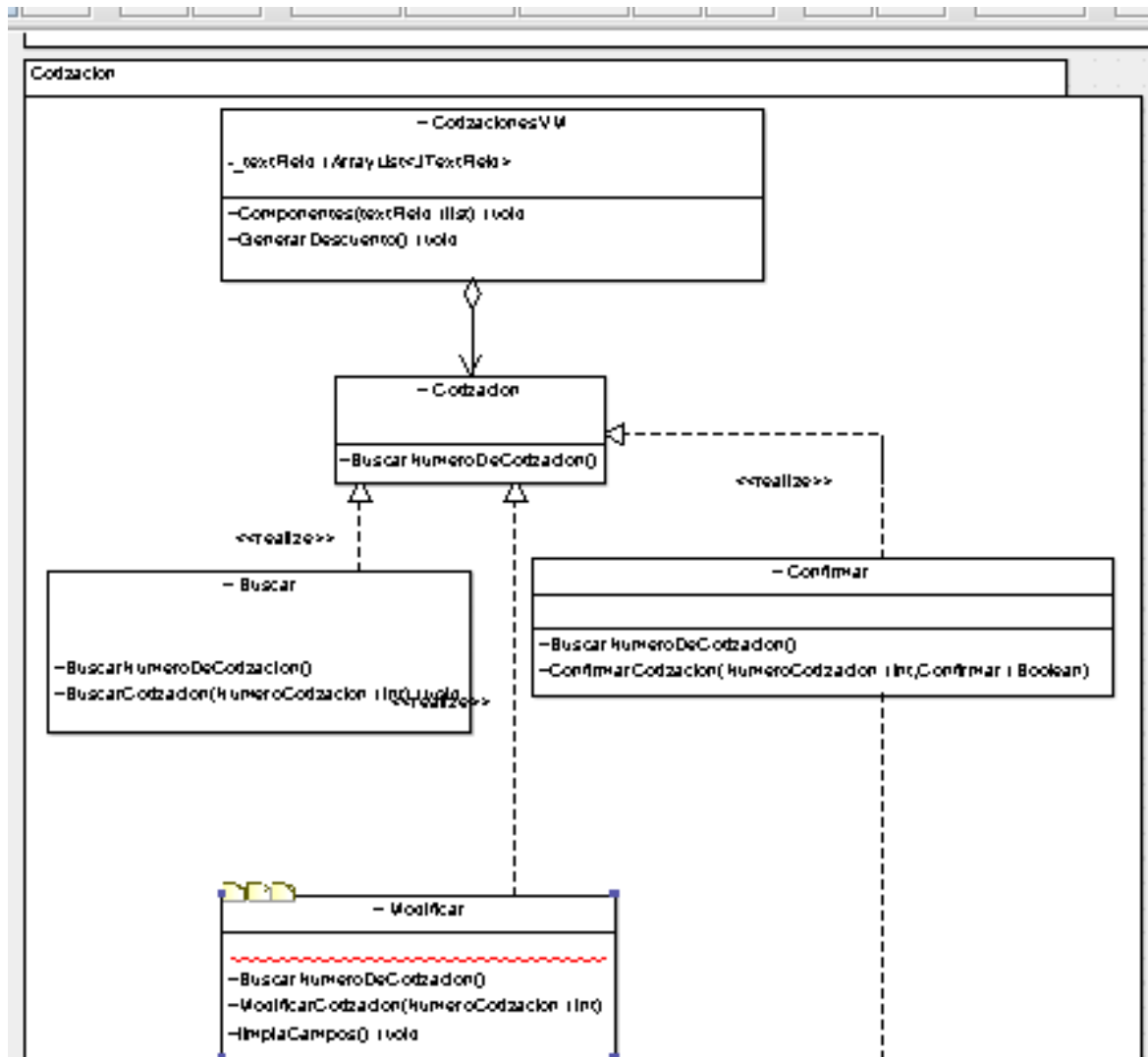
DIAGRAMA DE CLASES



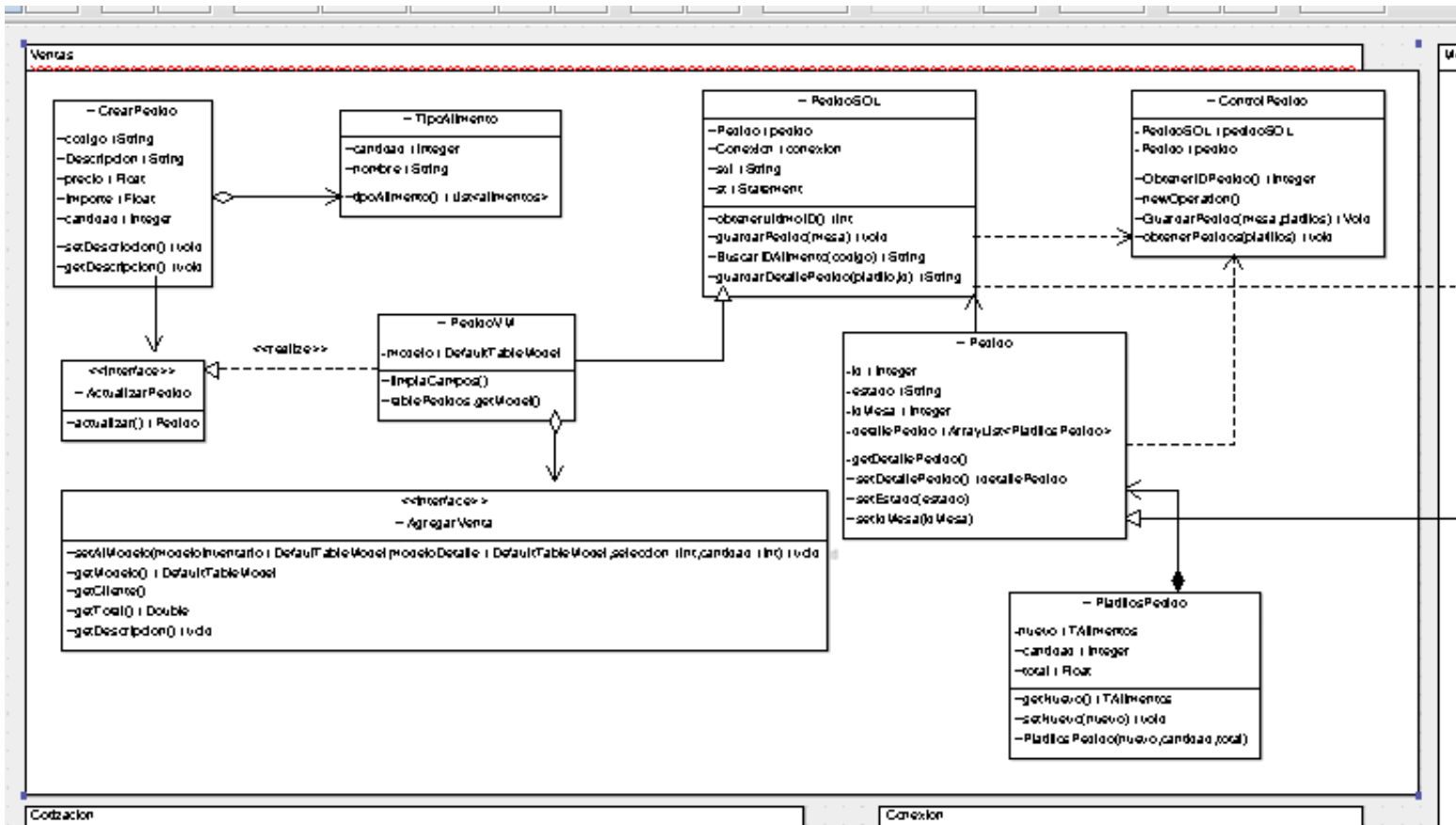
MODELOS Y PATRONES



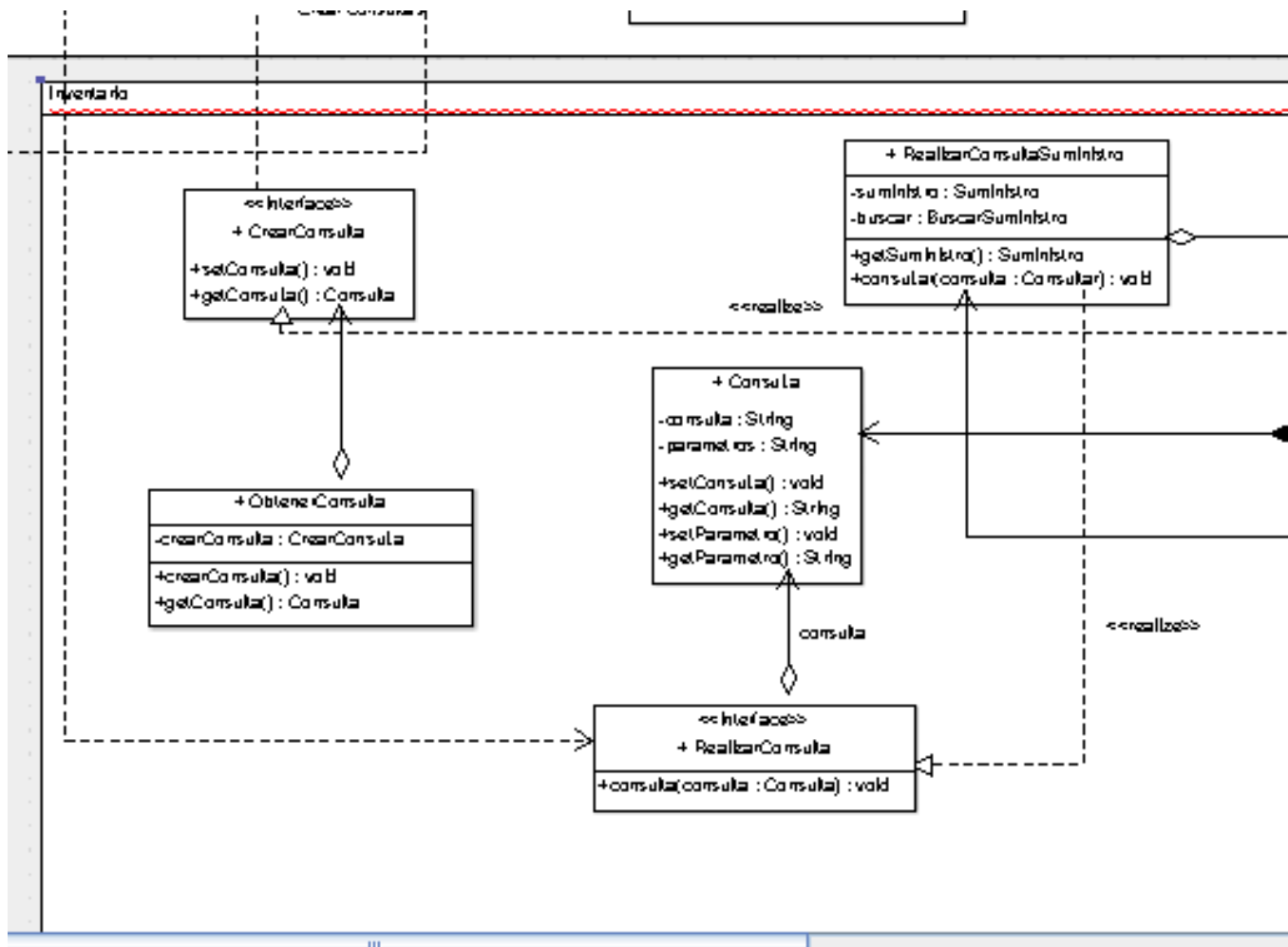
En el paquete de conexión se utilizó Patrón de diseño de la rama Creacional en este caso fue el Singleton el cual nos asegura que una clase solo puede ser instanciada una vez, y además de proveer un punto de acceso a esta; en este caso la única instancia se implementó en la clase conexion, esta está encargada únicamente de realizar la conexión cuando el usuario esté ya validado (usuario y contraseña según base de datos), sigue también el patrón de diseño Singleton ya que al dividir las tareas en partes más pequeñas esta clase se encarga únicamente de realizar la conexión.; así mismo se tomó en cuenta el Principio de SOLID implementando la responsabilidad única que establece que un componente o clase debe tener una responsabilidad única, sencilla y concreta. Esto simplifica el código al evitar que existan clases que cumplan con múltiples funciones, las cuales son difíciles de memorizar y muchas veces significan una pérdida de tiempo buscando qué parte del código hace qué función. También el principio de abierto/ cerrado y principio de sustitución de Liskov.



En cuanto al paquete de cotización en este diagrama nos enfocamos en los Patrones de diseño creacional específicamente en la Fábrica abstracta y el Método fábrica ya que se define una clase cotización para la creación de un objeto, pero deja que la subclase decida que clase instanciar en este caso las clases **Buscar** y **confirmar** separando responsabilidades y la fábrica abstracta en cuanto a la clase **CotizacionesVM** que provee una clase para la creación de familias de objetos sin especificar una clase en concreta. En este diagrama se utilizó los principios SOLID ya que está el principio de inversión de dependencia la cual establece que los módulos de alto nivel no deben depender de los de bajo nivel. En ambos casos deben depender de las abstracciones.; también el de open/closed.

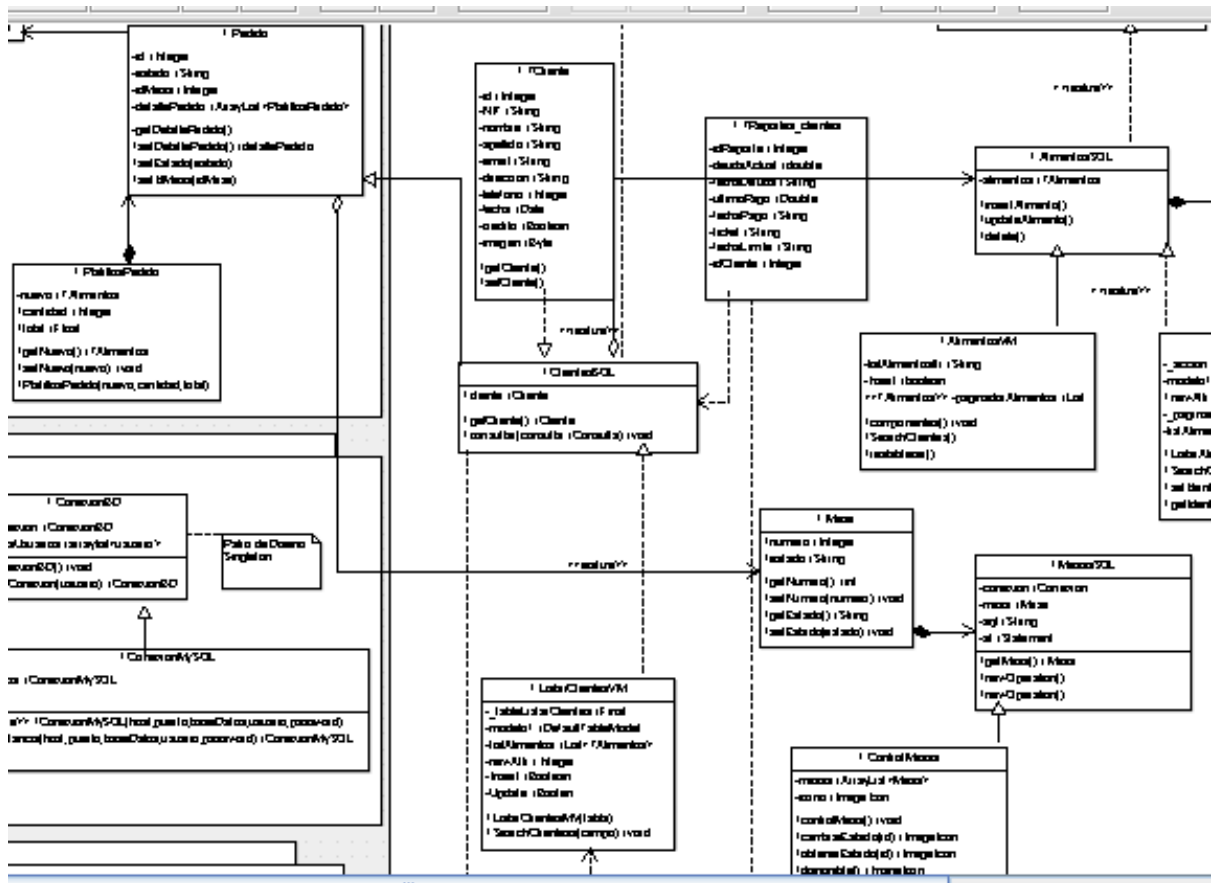


En cuanto al modulo de Ventas se utilizaron diferentes patrones de Diseño tanto estructurales como de comportamiento al igual que los cinco principios de SOLID como el de open/closed, el de principio de segregación de interfaces que establece que los clientes no deben ser forzados a depender de interfaces que no utilizan en este caso se utilizo el patrón Decorador que te permite añadir funcionalidades a objetos colocando estos objetos dentro de objetos encapsuladores especiales que contienen estas funcionalidades es este caso se implementó en la clase de PedidosSQL la cual implementa PedidosVM, pedidos y control de Pedido ya que cada uno de ellos deben ser parte fundamental para que se Cree un pedido, Al mismo tiempo se utilizo el patrón de comportamiento cadena de responsabilidad ya que evita acoplar el emisor de una petición a su receptor dando a más de un objeto la posibilidad de responder a una petición. Para ello, se encadenan los receptores y pasa la petición a través de la cadena hasta que es procesada por algún objeto al Agregar una venta y pasa todo el proceso.



En cuanto al paquete de Inventario en los diagramas para realizar las respectivas consultas y obtenerlas para el siguiente diagrama se utilizó el patrón de diseño estructural se utilizó para realizar consultas ya que explican cómo ensamblar objetos y clases en estructuras más grandes, a la vez que se mantiene la flexibilidad y eficiencia de estas estructuras en este caso se utilizó como Adapter la clase consulta ya que es capaz de trabajar tanto con la clase construir consultas como con la clase de ObtenerConsulta: implementa la interfaz Realizar consulta con el ClienteSQL, mientras envuelve el objeto de la clase obtener Consulta La clase adaptadora recibe llamadas del cliente a través de la interfaz RealizarConsulta y las traduce en llamadas al objeto envuelto de la clase de ClienteSQL pero en un formato que pueda comprenderse, fueron implementados los principios de SOLID

en cuenta el principio abierto/cerrado, Principio de sustitución de Liskov esta fue de gran ayuda ya que este principio establece que una subclase puede ser sustituida por su superclase. Es decir, podemos crear una subclase llamada AlimentosVM, la cual deriva de la superclase AlimentosSQL. Si al usar la superclase el programa falla, este principio no se cumple.



Se implemento es Patron de Estado (State) el cual pertenece al patrón de diseño de comportamiento que permite a un objeto altere su comportamiento cuando su estado interno cambia. de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes, en este caso se utilizó en MesasSQL, mesas y pedido ya que a la hora de realizar un pedido se le notifica que numero de mesa está siendo ocupada en cuanto cambia de estado estar en proceso el pedido por lo tanto las mesas cambian de color verde si está disponible, amarillo si están por salir y en rojo cuando están ocupadas entonces en estado de Mesas se observa en qué estado esta las mesas para no utilizarlas.

CONCLUSION

Gracias al análisis hacia la empresa Restaurante de comida típica “CALLE REAL” previo al desarrollo del Sistema de información y el haber contemplado sus componentes o pasos se pudieron identificar las necesidades y nos ayudó a traducir dichas necesidades en un modelo de Sistema que utiliza varios componentes, entre ellos el Software, hardware, personas, base de datos, documentación y procedimientos.

Hemos llegado a la conclusión que antes de comenzar con el desarrollo de cualquier proyecto, es necesario conducir un estudio de Sistemas para detectar todos los detalles de la situación actual de la empresa.

La información recopilada con el análisis y estudio sirve como base para crear varias estrategias de Diseño, ya que se entiende el funcionamiento o procesos que la empresa emplea, el flujo de información y así se puede saber cómo mejorar el rendimiento de la empresa, optimizar el uso de los recursos y satisfacer las necesidades, Dentro del diseño es donde se fomenta la calidad del Proyecto, el diseño es la única manera de materializar con precisión los requerimientos del cliente para tener ese boceto o plano en el que nos da una vista de los requerimientos.