

## Actividad | 2 | Paradigma Orientado a Objetos

### Análisis y Diseño de Sistemas

Ingeniería en Desarrollo de Software



TUTOR: Eduardo Israel Castillo García

ALUMNO: Adriana Esteban López

FECHA: 05 de julio de 2024

# INDICE

Introducción	03
--------------	----

.....

Descripción	05
-------------	----

.....

Justificación	06
---------------	----

.....

Desarrollo	07
------------	----

.....

Conclusión	12
------------	----

.....

## INTRODUCCIÓN

El Paradigma Orientado a Objetos (POO) es un modelo o un estilo de programación que está basado en el concepto de clases y objetos, para diseñar aplicaciones y programas; además de emplear varias propiedades como:

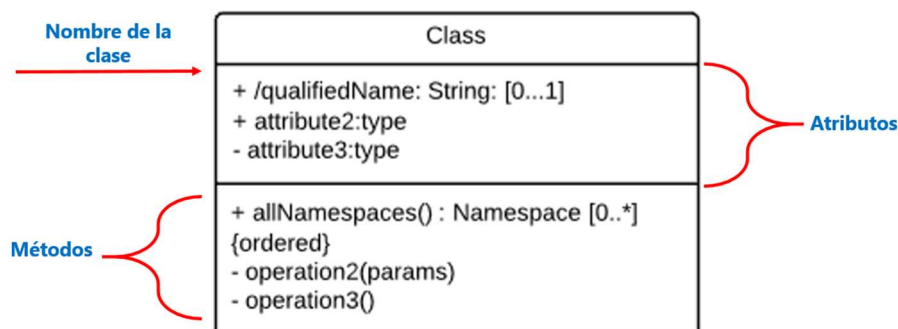
1. Abstracción de Datos
2. Herencia, esto nos permite reutilizar el código programado en una clase o extendiendo las características de un objeto
3. Encapsulamiento, esto nos ayuda a proteger la información o el estado de los atributos para que no se pueda ver o modificar la información de un determinado objeto.
4. Polimorfismo, es una colección de objetos con herencia.

El POO se centra en los objetos que son tipos de datos que se encapsulan con el mismo nombre, estructura de datos, operaciones o algoritmos que manipulan esos datos, esto es que le permite establecer controles para restringir la visibilidad y acceso de los miembros de una clase, de tal forma que puedan proteger y ocultar datos.

Dentro de este modelo tenemos dos herramientas, que serán las que estaremos viendo a lo largo del desarrollo de esta actividad:

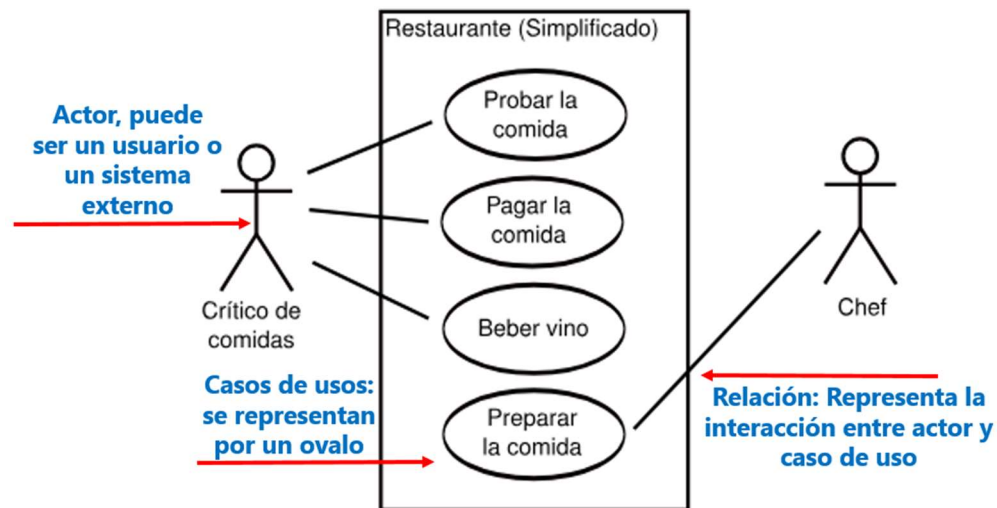
1. Diagrama de Clases: es una herramienta para comunicar el diseño de un programa que se creó para modelar relaciones entre diferentes entidades.

El diagrama de clases se representa por un rectángulo y consta de tres partes esenciales, Nombre, Atributos y Métodos:



2. Diagramas de casos de usos: es un tipo de diagrama de comportamiento y se usa frecuentemente para analizar varios sistemas, ya que permite visualizar los diferentes tipos de roles de un sistema y como estos roles interactúan con el sistema.

Los diagramas de casos de usos, tiene también tres componentes: Actor, Caso de Usos y Relaciones.



## DESCRIPCIÓN

Para el desarrollo de esta actividad, se tiene este contexto:

En una empresa de venta de ropa online, se requiere un sistema que conlleve los requerimientos para una gestión adecuada de los involucrados, así como del proceso de compra-venta de sus productos.

El contexto general es el siguiente:

- La empresa cuenta con: empleados, clientes, proveedores y sucursales.
- Sus procesos son: venta minorista y venta de mayoreo.

Cuando el cliente ingresa al sistema, este elige los productos de su preferencia. Posteriormente, al realizar la compra, tiene la posibilidad de adquirir en mayoreo a partir de 5 piezas. Todos los clientes deben tener un registro dentro del sistema. En este sentido, a cada cliente se le solicitan sus datos generales (nombre, apellido, dirección, correo electrónico, usuario y contraseña).

En la página del comercio el cliente podrá acceder para ver sus compras y podrá adquirir sus productos en línea. Esto mediante su nombre de usuario y contraseña previamente registrados. Identificar a los actores que se involucran en este proceso, así como cada una de sus actividades.

La actividad a realizar será diseñar los diagramas de clases y diagramas de casos de uso para la empresa de venta de ropa online.

## JUSTIFICACIÓN

El diagrama de clases tiene 2 propósitos:

1. Visualizar las clases de un sistema y sus propiedades.
2. Mostrar y analizar las relaciones entre clases.

Mientras que los diagramas de casos de usos se utilizan para reunir los requisitos de un sistema:

1. Identificar funciones y la forma en que los roles interactúan con ellas.
2. Para una visión de alto nivel del sistema.
3. Identificar los factores internos y externos.

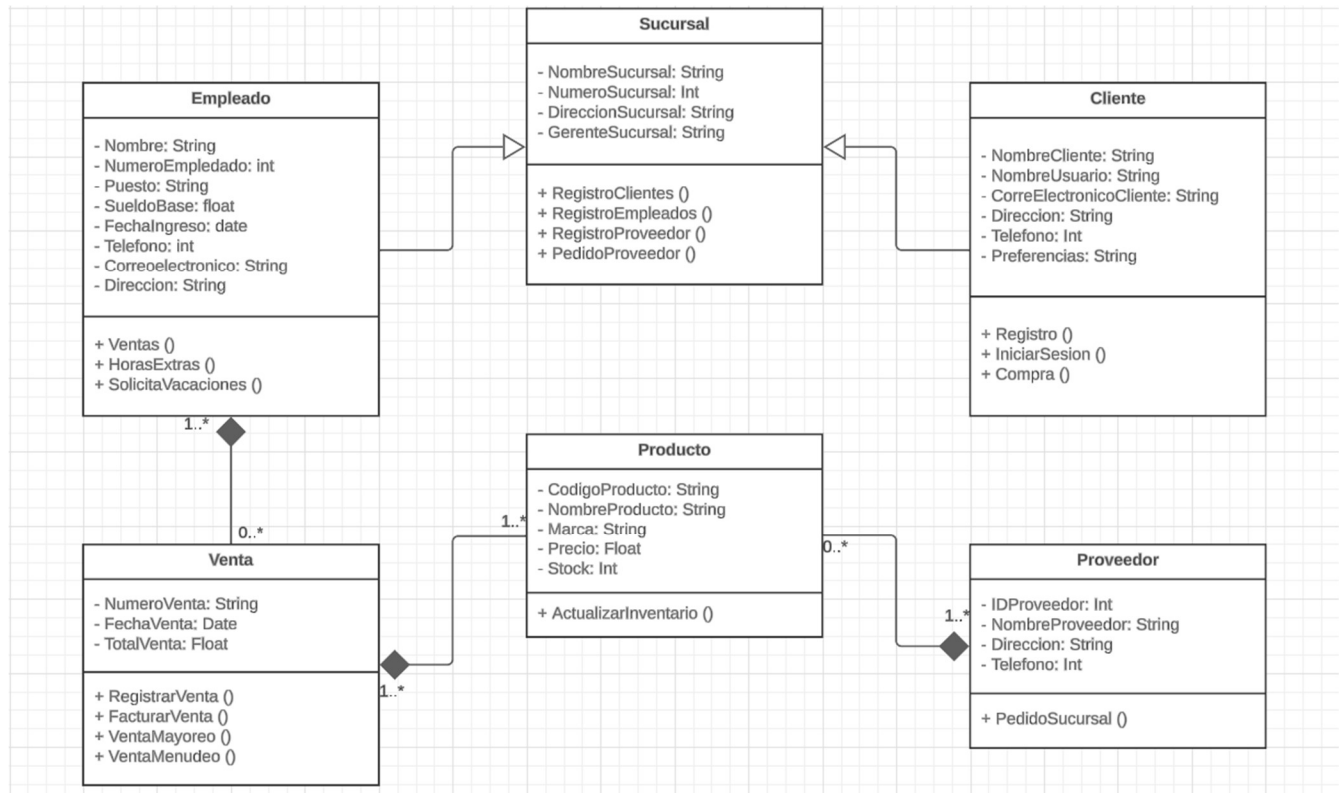
Al finalizar la actividad podremos identificar cada diagrama con sus respectivas características y definir de acuerdo a los requerimientos que tengamos que satisfacer que modelo se adecua más a nuestra actividad.

## DESARROLLO

Iniciamos con el desarrollo del Diagrama de Clases, para lo cual primero vamos a definir las clases de nuestro Sistema, tomando en cuenta que una clase refleja un objeto o nombre de una entidad.

Tomando en cuenta el contexto de la tienda de ropa online, podemos definir el siguiente diagrama de clases:

1. Primero en base a la información del contexto definimos las clases, recordando que una clase representa un objeto o un conjunto de objetos que comparten una estructura y comportamientos comunes y siempre se representa con un rectángulo, en el cual también se agregan sus atributos y los métodos u operaciones que se pueden realizar con esa clase.

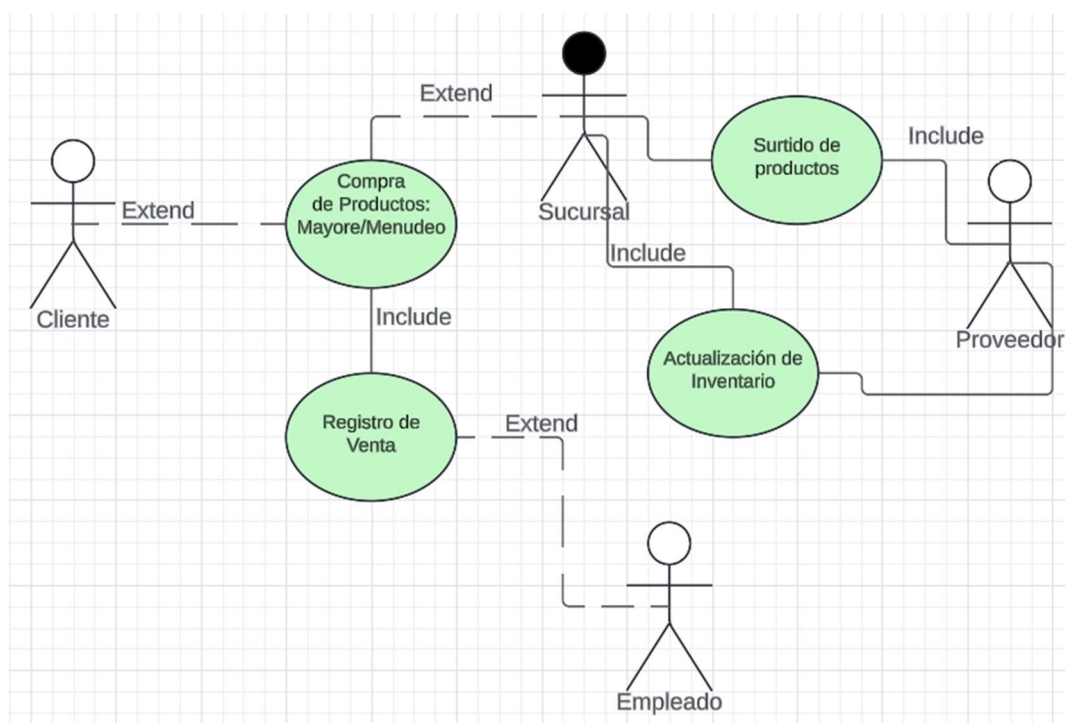


Dentro de este diagrama de clases, colocamos como **clase madre** a Sucursal, ya que es donde se inicia el proceso de venta, siendo las clases **Empleado** y **Cliente** consideradas como clases hijo ya que directamente dependen la clase Sucursal; por ejemplo, la clase **Empleado** cuenta con los siguientes atributos y métodos:

Atributos (Características de la Clase)	Métodos (Acciones que puede llevar la Clase)
<ul style="list-style-type: none"> <li>- Nombre: String</li> <li>- NumeroEmpleado: int</li> <li>- Puesto: String</li> <li>- SueldoBase: float</li> <li>- FechaIngreso: date</li> <li>- Telefono: int</li> <li>- Correoelectronico: String</li> <li>- Direccion: String</li> </ul>	<ul style="list-style-type: none"> <li>+ Ventas () <i>El empleado pueden realizar las ventas de la sucursal.</i></li> <li>+ HorasExtras () <i>El empleado solicitan pago de tiempo extra</i></li> <li>+ SolicitaVacaciones () <i>El Empleado solicitan tomar sus vacaciones de acuerdo a la antigüedad que tienen.</i></li> </ul>

Ahora realizaremos el mismo sistema para la tienda online de venta de ropa, pero utilizando el diagrama de clases de uso, recordando que son tres elementos que no pueden faltar: Actores (usuarios del sistema), Casos de Uso (se representa con óvalos y responde a la pregunta ¿Qué hace el sistema?) y finalmente las Relaciones (interacción entre el actor y caso de uso).

Los actores del sistema serían: Sucursal, Empleado, Cliente, Proveedor, quedando el diagrama de la siguiente manera:





Dentro del diagrama definimos el actor principal a **Sucursal** en la cual se genera el principal caso de uso **Compra de Productos Mayoreo/Menudeo**, en el cual interviene los actores de **Cliente y Empleado**; así mismo derivado esta interacción interviene el actor **Proveedor**, quien va a estar surtiendo los productos que vallan haciendo falta en el inventario.

## CONCLUSIÓN

Los diagramas de clases son herramientas prácticas de modelado para construir una arquitectura de software ya que facilita el proceso al señalar las clases, sus atributos y sus métodos y también visualizan como se relacionan las clases entre sí.

Así mismo los diagramas de clases respaldan cambios en el sistema al mostrar un esquema de una aplicación completa; aplicar cambios o adicionales a un diagrama de clases proporciona una vista previa de cualquier nueva clase y como el cambio impacta en las clases existentes.

Lo más importante de un diagrama de casos de uso es comunicar el comportamiento del sistema a los clientes y usuarios finales, por lo que en consecuencia debe de ser fácil de entender.

Los actores ayudan a delimitar el sistema y nos dan una clara visión de lo que se supone debe de realizar dicho sistema; es decir, son desarrolladores con base en las necesidades de los actores, asegurando así que el sistema satisfaga todos los requerimientos.

El uso de ambas herramientas va a depender de los requerimientos que se tengan para la elaboración de un sistema

Se agrega dicha actividad a la plataforma de GitHub a través del siguiente link:

<https://github.com/22HADRIA/An-lisis-y-Dise-o-de-Sistemas>