



## **Actividad | 2 | Diagramas de Clases y Objetos.**

Lenguaje Unificado de Modelado.

Ingeniería en Desarrollo de Software.

---



academi**ag**lobal

TUTOR: M.C. EDUARDO ISRAEL CASTILLO GARCIA.

---

ALUMNO: JONATHAN OSWALDO CARDENAS GARCIA.

---

FECHA: 07-enero-2024

---

## Tabla De Contenido

Tabla De Contenido .....	2
Introducción .....	3
Descripción .....	4
Justificación .....	5
Desarrollo.....	6
Diagrama de clases .....	6
Conclusión .....	7
Referencias.....	8

## **Introducción**

En este trabajo se sigue trabajando con la misma primicia del primero, se toman las base de lo echo anteriormente para ampliarlas en este y verlas bajo la óptica de otra representación gráfica, otro diagrama el cual es el diagrama de clases, ahora en este podemos ver de una manera más organizada y concreta las clases que interfieren en nuestro sistema, sus atributos, el tipo de atributo que son y las cosas que cada clase puede hacer o ejecutar, esto es muy útil porque al ver esta forma de interacción de las clases podemos ver su relación clara al ver los atributos que se pueden ir heredando entre las mismas y como las clases dependen como tal una de otra pero siempre viendo hasta qué punto esta relación se extiende a una dependencia existencial especifica o no y observando las capacidades de lo que cada clase puede llegar a hacer.

## **Descripción**

Una institución financiera con el nombre de Creciendo BMD quiere que se desarrolle un sistema para realizar un sorteo con los usuarios que realicen una transacción o compren un producto, claro este sorteo consta de distintas restricciones, las cuales se muestran a continuación:

- El cliente obtendrá un boleto participante por cada transacción de \$500 pesos o más.
- Si la transacción cumple con el requisito de \$500, el cliente recibe un boleto participante y un boleto ganador o perdedor, dependiendo del azar. Si no cumple, solo se le entrega el recibo de transacción.
- Los empleados no tienen permitido participar en esta dinámica.
- Cada 2000 boletos emitidos, hay un boleto ganador.
- En total, hay 1000 boletos ganadores disponibles.
- En caso de resultar ganador, el cliente deberá responder una pregunta de opción múltiple con tres posibles respuestas.
- Se permite intentar responder la pregunta incorrecta hasta seleccionar la respuesta correcta.
- Es crucial validar que el sorteo esté activo y dentro del rango de fechas especificado.

### **Justificación**

La implementación de diagramas UML en la definición de requisitos funcionales y no funcionales, casos de uso y fichas de especificaciones es esencial para lograr una comprensión profunda y una comunicación efectiva en el desarrollo de software. Estos diagramas proporcionan una representación visual que simplifica conceptos complejos, permitiendo a los desarrolladores visualizar la estructura del sistema y sus interacciones. En lugar de depender únicamente de descripciones textuales, los diagramas UML ofrecen una visión más completa y detallada, facilitando la identificación de actores, escenarios y relaciones críticas. Esta herramienta visual se convierte en un lenguaje compartido, mejorando la colaboración y reduciendo la ambigüedad en la interpretación de los requisitos. Además, agiliza el análisis, diseño y validación, contribuyendo a un proceso de desarrollo más eficiente y preciso. En resumen, la integración de diagramas UML potencia la comprensión y la eficacia en todas las etapas del desarrollo de software.

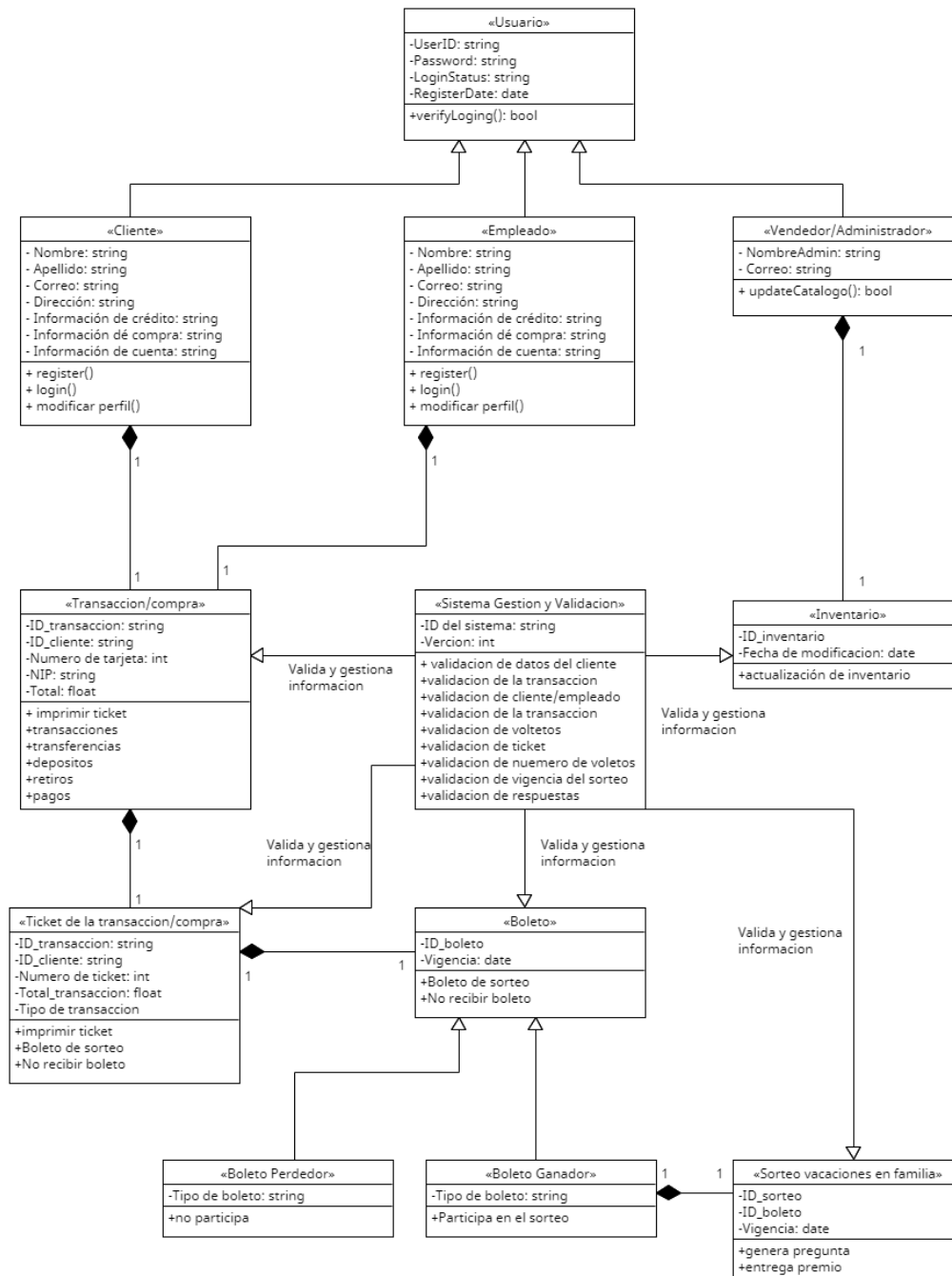
El diagrama de clases es muy importante en el desarrollo de software porque nos ayuda a ver todo como clases con atributos y actividades que se pueden realizar, a su vez podemos observar la interacción que tienen unas con otras para entender la relación que se pueden obtener y la importancia de las mismas.

## Desarrollo

### Diagrama de clases

**Figura 1**

*Diagrama de clases (clases, atributos, operaciones y relaciones)*



*Nota.* En este diagrama se ve la estructura del sistema, clases, atributos, relaciones y operaciones.

## **Conclusión**

Esta actividad fue un poco más complicada de pensar, no tanto por la complejidad de la misma sino más bien por una confusión que me salió al momento de realizarla y es en cuanto a las relaciones, pienso que la división de las clases está bien, es precisa y se entiende el cómo se separa y funciona el sistema, los atributos igual pienso que están bien explicados porque se ve claramente cómo se van pasando entre cada clase dependiendo de la relación, en la relación fue donde me confundí un poco más, no el que se une con que si no el cómo se une se me confundió un poco pero al final me gusto el hacer el diagrama, me resulto entretenido y se suma a las herramientas gráficas para entender el problema y verlo desde otra óptica tal vez un poco más estructurada que siempre me recuerda a un diagrama de una base de datos.

## Referencias

Enlace del repositorio de GitHub

<https://github.com/CardinalSG/Lenguaje-Unificado-de-Modelado.git>