

Actividad | #2| Diagramas de Clases y Objetos.

Ingeniería en Desarrollo de Software



TUTOR: Eduardo Israel Castillo García

ALUMNO: Edgar Enrique Cuamea Ochoa

FECHA: 04 de mayo del 2024

Contenido

Introducción.	3
Descripción.	4
Justificación.	5
Desarrollo.....	6
Conclusión	8
Referencias.....	9

Introducción.

Con base a la primera actividad presentada sobre el desarrollo de los casos de uso de un sorteo para una tienda departamental, será necesario realizar diagramas de clases identificando diferentes clases, atributos, operaciones y así poder identificar y describir los objetos del sistema además de sus características y sus relaciones entre ellas, identificaremos que tipo de relación tienen, ya sea una relación fuerte o débil y su multiplicidad para poder saber si alguna clase puede tener algún múltiplo de una clase diferente, como una empresa puede tener uno o muchos empleados, esto con la finalidad de poder identificar correctamente a los componentes de nuestro sistema y saber que tipo de relación tiene con alguna clase diferentes así como poder identificar los atributos de las clases e identificar la clase madre así como estas clases pueden heredar sus atributos a las demás clases hijas además de relacionarlas con alguna relación fuerte o débil.

Descripcion.

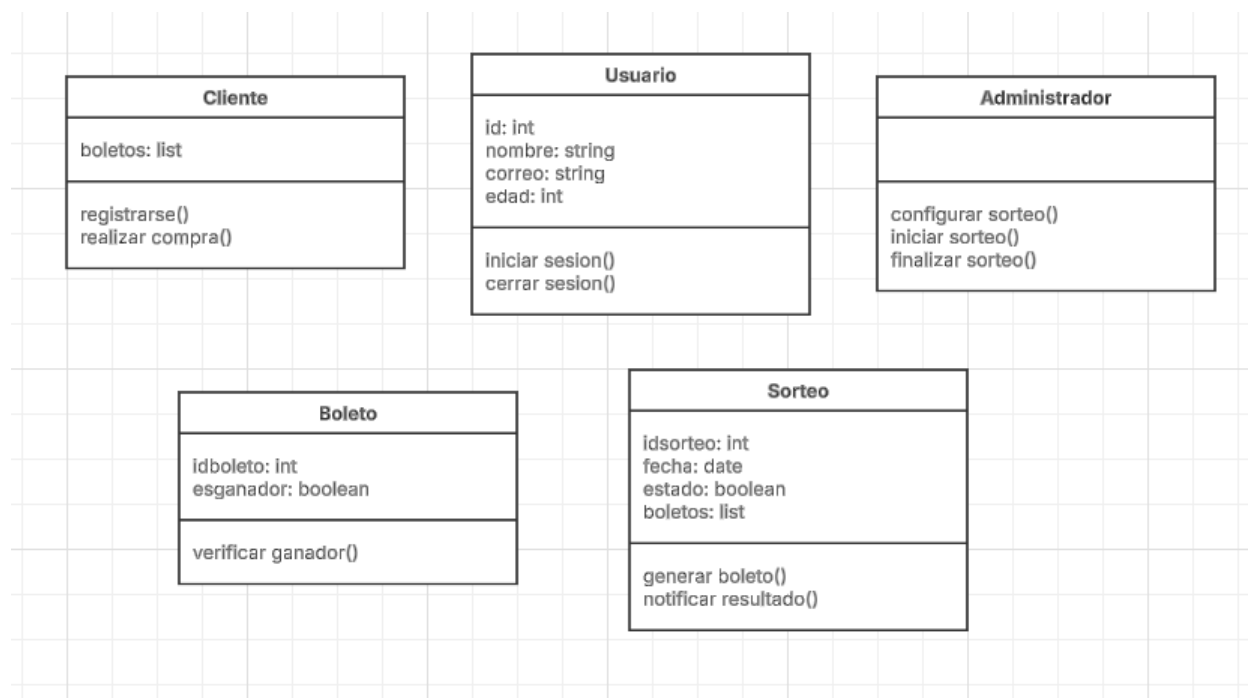
Usaremos un programa que nos permita realizar estos diagramas, así poder realizar las diferentes clases e identificar sus atributos, además de realizar estas clases tendremos que relacionarlas entre si en ese sistema para poder identificar si la clase se relaciona de una manera fuerte con alguna clase heredada o de manera débil con alguna otra clase con la que las podemos relacionar, además de esto mostraremos el nombre de la relación así como la multiplicidad que este tienen con alguna otra clase por lo que podremos identificar diferentes relaciones, clases, atributos y multiplicidades representadas de manera grafica utilizando como base la actividad anterior donde realizamos nuestro diagrama de casos de uso por lo que el diagrama podremos identificar a nuestro administrador y cliente como dos clases diferentes así como la herencia de una clase madre llamada usuario por lo que es importante identificar el tipo de relación que estas clases tienen entre si así como saber si un cliente puede tener uno o muchos boletos.

Justificación.

Realizaremos nuestro diagrama en la página web de Lucidchart ya que es una página que nos permite realizar estos diagramas de una manera fácil, así como de manera intuitiva ya que cuenta con plantillas que harán el proceso de realización de esta actividad un poco más fácil ya que podremos crear las clases, así como las relaciones que estos tienen, podremos realizar modificaciones en las relaciones de estas clases, así como indicar con diferentes relaciones si esta es una relación fuerte o débil, esta página cuenta con una gran variedad de herramientas con las que podemos realizar diferentes acciones así como agregar texto o formas por lo que hace que la comprensión de la creación de las clases así como la creación de las relaciones y su multiplicidad, sean más fáciles ya que podremos comprender como realizarlas basándonos en la creación de los casos de usos de la actividad anterior por lo que es necesario su comprensión para realizar el diagrama de clases

Desarrollo.

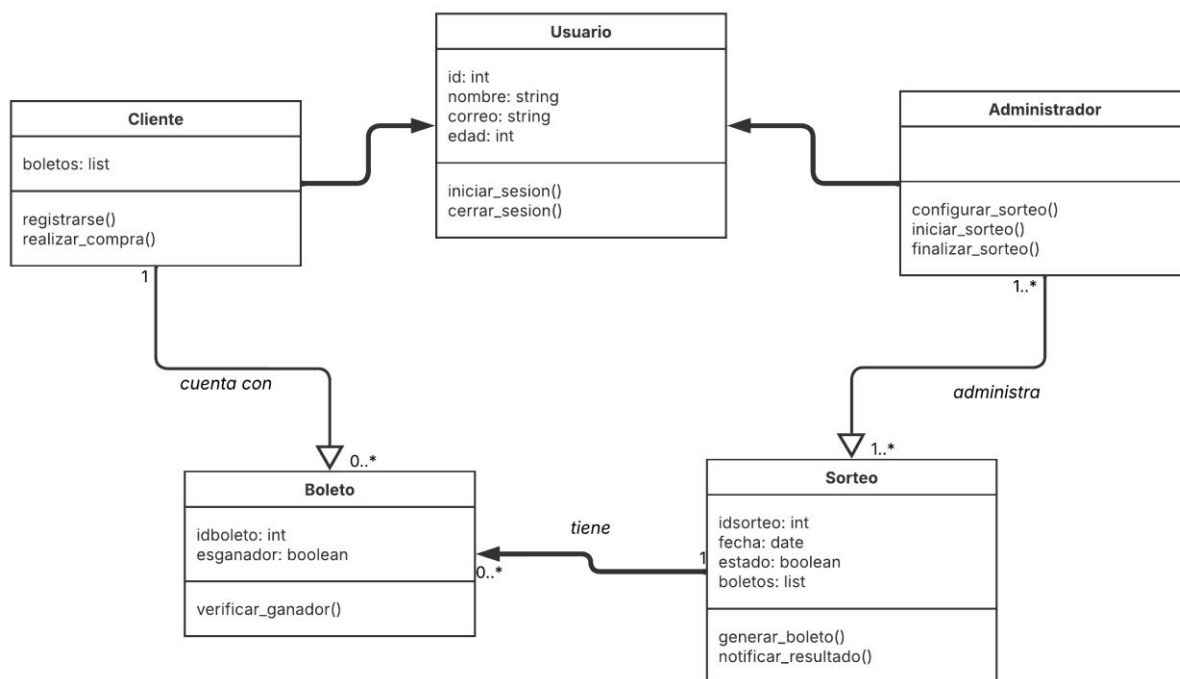
Realizaremos nuestras clases, en este caso identificamos 5 clases distintas, Cliente, administrador, sorteo y boleto, además de la clase madre usuario, estas 5 clases tienen sus propios atributos, así como sus métodos que los representaremos de la siguiente forma



Tenemos nuestra clase madre usuario donde este contiene los atributos de id, nombre, correo y edad, estas serán heredadas a las clases cliente y administrador, por lo tanto administrador no cuenta con atributos propios aunque se podría agregar opciones como niveles de acceso o áreas asignadas, este solo realizará las siguientes acciones, configurar sorteo, iniciar sorteo y finalizar sorteo, el cliente tiene su propio atributo boletos, ya que cada cliente distinto tendrá una lista con sus boletos generados donde el cliente puede registrarse y realizar compras. Todas estas clases tienen sus atributos así como el tipo de dato que se le aplicaría al momento de ingresarlo a una base de datos.

Además de los usuarios, tenemos 2 clases más, boletos y sorteo, la clase boleto tiene si id

de boleto y una verificación booleana en su atributo, ya que solo tiene 2 estados, si o no por lo que la acción que realiza es verificar si el boleto es ganador o no lo es, en nuestra clase sorteo tenemos como atributos el id del sorteo así como la fecha de este, un estado booleano para determinar si el sorteo está activo o no así como una lista de los boletos generados. Este puede realizar la generación de los boletos así como notificar el resultado de los boletos



Realizamos nuestras relaciones de las siguientes formas, la clase cliente y administrador son heredadas de nuestra clase madre usuario ya que esta clase le hereda los atributos a estas clases hijas, la clase administrador no cuenta con atributos propios ya que las hereda de la clase usuario, aunque podremos agregar algún atributo diferente como el departamento asignado o el nivel de el administrador, agregamos la multiplicidad por lo tanto uno o muchos administradores pueden administrar uno o muchos sorteos, esto lo relacionamos con una relación de agregación ya que el sorteo puede existir sin el administrador así como un sorteo lo puede administrar un administrador diferente, el sorteo lo relacionamos con la clase boleto con relación compuesta ya que la clase

boleto depende de el sorteo y no tiene sentido que existan boletos si no los genera el sorteo mismo, donde un solo sorteo puede tener 0 o muchos boletos generados, el boleto lo relacionamos de forma agregada a la clase cliente, ya que un solo cliente cuenta con 0 o muchos boletos y al ser una relacion de agregacion, aunque el cliente tenga boletos, esta clase puede existir independientemente ya que si eliminamos el cliente, estos aun pueden existir pues los genera el sorteo y este tiene una lista de boletos generados.

Conclusion

Identificamos 5 diferentes clases, así como una clase madre que hereda atributos a 2 clases distintas, identificamos los atributos de cada una de las clases así como el tipo de dato que utiliza cada atributo, ya sea (int) como numero entero o (string) como cadena de texto así como diferentes tipos de datos para todos los atributos, identificamos los métodos o acciones que pueden realizar las diferentes clases, identificamos las relaciones entre las clases, ya sean de tipo composición o de tipo agregación para diferenciar entre una relación fuerte o una relación débil además de la multiplicidad de la relaciones entre las clases así como una clase puede tener ninguna, una o más múltiplos de alguna otra clase además de poder relacionarlo con las creaciones de las bases de datos ya que también requerimos de clases para la realización de estos, todo esto utilizando como base el diagrama de casos de usos de la actividad anterior por lo que comprendemos las funciones que un sistema de un sorteo puede realizar así como las bases para la creación de bases de datos además de comprender mejor el diagrama de clases al momento de realizarla.

Referencias.

Hackemi. (s. f.). *Hackemi/Modelado_Uml*. GitHub.
https://github.com/Hackemi/Modelado_Uml

Clase UML | Lucidchart. (s. f.). https://lucid.app/lucidchart/a1e4bcab-025e-42f3-a320-70bff153cd71/edit?viewport_loc=-7%2C-113%2C2159%2C1093%2CHWEp-vi-RSFO&invitationId=inv_2eca283a-a3ab-4973-984f-2ade0b681d51