Taller de Ejercicios en UML

Diagrama de Clases desde Cero

Christian Ramirez
@christian_ramireezz

Ejercicio #1

En el ámbito deportivo, específicamente en el baloncesto, surge la necesidad de modelar un sistema que permita comprender las dinámicas y los elementos fundamentales que interactúan durante un partido. Este sistema será útil tanto para entrenadores como para analistas deportivos, ayudándoles a simular situaciones, evaluar tácticas y mejorar el rendimiento de los jugadores.

Para desarrollar este sistema, se requiere identificar y representar los componentes principales del juego de baloncesto, especificando sus atributos, comportamientos y relaciones entre ellos. A continuación, se detallan los elementos esenciales que deben incluirse:

Clase	Atributos	Métodos
Balón	Diámetro (tamaño del balón) Volumen (en cm^3)	driblar() tirar() pasar() avanzar()
Jugador	Nombre Estatura Peso	driblarBalón() pasarBalón() tirarBalón() rebotar() infraccionarOponente()
Defensa		
Equipo		
Cesto		

Clase	Atributos	Métodos
Delantero		
Centro		
Tiro		
Infracción		
LapsoDeTiro	- profesional: 24 segundos - colegial: 35 segundos - internacional: 30 segundos	
CronómetroDeJuego	- profesional: 4 cuartos de 12 minutos - colegial e internacional: 2 mitades de 20 minutos	

Clase	Atributos	Métodos
Duración	- profesional: 48 minutos - colegial e internacional: 40 minutos	
LíneaDeTresPuntos		
TiroLibre		
LíneaDeTiroLibre		
Cancha		

Ejercicio #2

Un investigador desea modelar un sistema para clasificar y estudiar diferentes tipos de animales según sus características principales. El modelo debe seguir una jerarquía que permita heredar propiedades y comportamientos comunes desde una clase general Animal, extendiéndolos a clases específicas como Mamífero, Anfibio y Reptil. Además, debe incluir subclases concretas, como Caballo, que heredan directamente de Mamífero

- Identificar las características comunes de todos los animales.
- Especializar comportamientos y atributos específicos en subclases según el tipo de animal
- Facilitar la extensión del modelo para incluir nuevos tipos de animales en el futuro

Clase	Descripción
Animal	Clase base que define las propiedades y métodos comunes para todos los animales
Mamífero	Subclase de Animal que representa animales que son mamíferos
Anfibio	Subclase de Animal que representa animales adaptados tanto a medios acuáticos como terrestres
Reptil	Subclase de Animal que representa animales que tienen escamas y suelen ser ectotérmicos
Caballo	Subclase de Mamífero que representa un animal específico con características concretas

Ejercicio #3

Un desarrollador de videojuegos está creando un simulador de deportes, y necesita implementar un sistema que modele las características de los jugadores según sus posiciones (Defensa, Delantero y Centro) y un reloj para controlar el tiempo de juego y el lapso para realizar tiros. Cada jugador debe tener atributos comunes, como su nombre, tamaño, peso, velocidad al correr y salto vertical. Además, según la posición del jugador, tendrá habilidades específicas

El sistema también debe controlar el tiempo total del juego y el tiempo limitado que cada jugador tiene para realizar un tiro.

Clase Jugador

Atributos	Tipo de Dato	Descripción
nombre	String	Nombre del jugador.
tamaño	float	Altura del jugador en metros.
peso	float	Peso del jugador en kilogramos.
velocidadAlCorrer	float	Velocidad del jugador al correr en metros por segundo.
saltoVertical	float	Altura máxima del salto vertical del jugador en metros.

Métodos	Descripción
driblar()	Simula la acción de driblar el balón
pasar()	Simula la acción de pasar el balón a otro jugador
rebotar()	Simula la acción de capturar un rebote
tirar()	Simula la acción de lanzar el balón al arco o canasta

Defensa hereda de Jugador

Métodos	Descripción
correrAlFrente()	Simula que el jugador corre rápidamente hacia el frente
quitarBalon()	Simula la acción de quitar el balón al oponente.

Clase Delantero Hereda de Jugador

(No hay métodos específicos definidos en el diagrama. Se podría considerar agregar métodos como anotarGol o tiroALargaDistancia, según el contexto del juego.)

Clase Centro hereda de Jugador

Métodos	Descripción
retacarBalon()	Simula la acción de encestar el balón con fuerza, como en un "mate" (slam dunk)

Clase Reloj

Métodos	Descripción
controlarTiempo()	Controla y actualiza el tiempo del juego o del tiro

Subclases de Reloj

Clase	Descripción
CronometroDelJuego	Hereda de Reloj y controla el tiempo total del juego
LapsoDeTiro	Hereda de Reloj y controla el tiempo máximo permitido para realizar un tiro

Ejercicio #4

Se necesita modelar la estructura de un equipo de cómputo desglosando sus componentes y subcomponentes físicos. Este modelo debe reflejar la relación jerárquica entre los elementos principales y sus dependencias (como teclado, monitor, gabinete, etc.), así como las conexiones adicionales entre partes internas del sistema (como unidades de almacenamiento y periféricos)

El objetivo es proporcionar una representación clara y estructurada para:

1. Visualizar la composición y conexiones del equipo
2. Organizar las dependencias entre los elementos
3. Facilitar el mantenimiento y análisis de los componentes en un contexto técnico

Clase	Descripción
EquipoComputo	Clase principal que representa al equipo completo, compuesto por varios subsistemas
Gabinete	Contiene los componentes internos del sistema como almacenamiento, tarjetas y memoria
Altavoz	Representa un dispositivo periférico para salida de audio
Monitor	Componente de salida visual conectado al equipo
Teclado	Periférico de entrada que interactúa con el usuario
Raton	Dispositivo de entrada compuesto por botones y una bola para desplazamiento
UnidadDisquete	Almacenamiento extraíble conectado al gabinete
UnidadDisco	Disco duro interno conectado al gabinete.
RAM	Memoria principal utilizada para procesamiento en tiempo real.

Clase	Descripción
CDROM	Unidad de lectura óptica conectada al gabinete
TarjetaVideo	Tarjeta gráfica instalada en el gabinete
TarjetaSonido	Tarjeta que gestiona la salida de audio del sistema
Boton	Parte del ratón utilizada para ejecutar acciones (clics)
Bola	Parte mecánica del ratón que permite el desplazamiento

Relaciones entre clases

Clase	Clase destino	Tipo de relación	Multiplicidad	Descripción
EquipoComputo	Gabinete	Agregación	1	El equipo está asociado a un gabinete, pero este puede existir por separado
EquipoComputo	Altavoz	Agregación	2	El equipo puede incluir dos altavoces, pero estos pueden existir aparte
EquipoComputo	Monitor	Agregación	1	El equipo incluye un monitor como periférico esencial
EquipoComputo	Teclado	Agregación	1	El equipo incluye un teclado como periférico, pero puede usarse en otro PC
EquipoComputo	Raton	Agregación	1	El equipo incluye un ratón como periférico de entrada

Clase	Clase destino	Tipo de relación	Multiplicidad	Descripción
Gabinete	UnidadDisquete	Agregación	1	El gabinete contiene exactamente una unidad de disquete
Gabinete	UnidadDisco	Agregación	1..*	El gabinete contiene una o más unidades de disco
Gabinete	RAM	Agregación	*	El gabinete puede incluir varias memorias RAM
Gabinete	CD-ROM	Agregación	1	El gabinete incluye exactamente una unidad de CD-ROM
Gabinete	TarjetaSonido	Agregación	1	El gabinete incluye exactamente una tarjeta de sonido
Gabinete	TarjetaVideo	Agregación	1	El gabinete incluye exactamente una tarjeta de video

Clase	Clase destino	Tipo de relación	Multiplicidad	Descripción
Ratón	Botón	Agregación	1..3	Un ratón puede estar conectado a entre 1 y 3 botones, que no necesariamente son exclusivos del ratón.
Ratón	Bola	Agregación	1	Un ratón puede estar conectado a una bola que puede existir como componente independiente.
Altavoz	TarjetaDeSonido	Agregación	2 a 1	Dos altavoces están conectados a una sola tarjeta de sonido que los controla

Ejercicio #5

Un restaurante desea modelar su sistema de gestión de menús utilizando UML. El sistema debe cumplir las siguientes reglas:

- **Jerarquía del Menú:** Cada menú (Comida) puede estar compuesto por una ensalada, un plato fuerte y un postre, pero debe incluir obligatoriamente el plato fuerte y el postre
- **Agregación:** El menú (Comida) tiene una relación de agregación con sus elementos (Ensalada, PlatoFuerte, Postre), lo que implica que los elementos pueden existir independientemente del menú
- **Restricción OR:** El menú puede incluir una ensalada o ningún componente opcional (indicado por {0})
- **Restricciones Obligatorias:** Cada menú debe incluir obligatoriamente un objeto de tipo PlatoFuerte y un objeto de tipo Postre
- **Exclusividad:** Los elementos asociados a un menú son exclusivos de ese menú y no pueden ser compartidos con otros

Clase	Descripción
Comida	Representa el menú completo del restaurante
Ensalada	Representa un componente opcional del menú (puede no estar presente)
PlatoFuerte	Representa el componente principal del menú (obligatorio)
Postre	Representa el componente dulce incluido en el menú (obligatorio)

Clase	Tipo	Multiplicidad	Descripción
Comida-Ensalada	Agregación	0..1	Un menú puede incluir opcionalmente una ensalada o ningún componente.
Comida-PlatoFuerte	Agregación	1:1	Cada menú debe tener exactamente un plato fuerte
Comida-Postre	Agregación	1:1	Cada menú debe tener exactamente un postre

Ejercicio #6

Una universidad desea implementar un sistema para gestionar su infraestructura tecnológica, incluyendo tanto el hardware como el software instalado en sus edificios y campus. Este sistema debe permitir administrar computadoras, software, sistemas operativos, y categorías de software. Además, debe registrar qué empleados son responsables del mantenimiento y soporte del software.

El sistema debe incluir las siguientes funcionalidades principales:

- Registrar los campus y edificios de la universidad, donde se encuentran los equipos informáticos
- Controlar las computadoras (hardware) disponibles en la universidad, con detalles como marca, modelo, especificaciones técnicas, y ubicación
- Gestionar el software instalado en las computadoras, permitiendo actualizar versiones, instalar y desinstalar software

Ejercicio #6

- Relacionar el software con sus categorías (como utilitarios, educativos, etc.) y sistemas operativos compatibles
- Registrar a los empleados que son responsables del mantenimiento del software, almacenando información de contacto y las áreas que supervisan
- Asociar cada computadora con un edificio específico
- Relacionar el software con las computadoras donde está instalado y con los expertos que supervisan su mantenimiento

Clase	Atributos	Métodos
EdificioCampus	- codigoCampus (String) - descripcionCampus (String)	- agregarEdificio()
Computadora	- numeroInventarioHardware (String) - tipoComputadora (String) - nombreMarca (String) - modelo (String) - numeroSerie (String) - fechaCompra (Date) - costoCompra (double) - costoReemplazo (double) - tamanoMemoria (int) - capacidadDiscoDuro (int) - capacidadSegundoDiscoDuro (int) - discoOptico (boolean) - garantia (int, en meses) - codigoCampus (String) - ubicacionSala (String) - numeroDistribuidor (String)	- agregarComputadora() - cambiarComputadora() - eliminarComputadora() - obtenerComputadora()

Clase	Atributos	Métodos
HardwareSoftware	- numeroInventarioHardware (String) - numeroInventarioSoftware (String)	- instalarSoftware() - actualizarVersion() - eliminarSoftware() - obtenerSoftware() - obtenerHardware()
Software	- numeroInventarioSoftware (String) - titulo (String) - nombreSistemaOperativo (String) - numeroVersion (String) - editor (String) - codigoCategoriaSoftware (String) - marcaComputadora (String) - memoriaRequerida (int) - licenciaSitio (String) - numeroDeCopias (int) - costoSoftware (double) - numeroEmpleado (String)	- agregarSoftware() - cambiarSoftware() - obtenerSoftware() - obtenerTituloSoftware() - eliminarSoftware()

Clase	Atributos	Métodos
SistemaOperativo	- codigoSistemaOperativo (String) - significadoSistemaOperativo (String)	- agregarSistemaOperativo() - cambiarSistemaOperativo() - eliminarSistemaOperativo() - obtenerSistemaOperativo()
CategoriaSoftware	- codigoCategoriaSoftware (String) - descripcionCategoriaSoftware (String)	- agregarNuevaCategoria() - cambiarCategoria() - eliminarCategoria() - obtenerCategoria()
ExpertoSoftware	- numeroEmpleado (String) - apellidoPaterno (String) - primerNombre (String) - telefonoOficina (String) - email (String) - codigoDepartamento (String) - ensenaCurso (boolean)	- agregarExperto() - cambiarExperto() - eliminarExperto() - obtenerExperto()

Relaciones entre Clases

Relación	Explicación
EdificioCampus → Computadora (1:N)	Cada edificio tiene varias computadoras asociadas, pero una computadora pertenece a un único edificio.
HardwareSoftware → Computadora	Computadora extiende las funcionalidades de HardwareSoftware para representar las características específicas del hardware físico.
HardwareSoftware → Software	Software extiende HardwareSoftware para representar las características del software instalado y las interacciones con el hardware.
SistemaOperativo → Computadora (1:N)	Un sistema operativo puede estar instalado en múltiples computadoras, pero cada computadora tiene un único sistema operativo.
CategoriaSoftware → Software (1:N)	Cada software pertenece a una categoría específica (por ejemplo, utilitarios, educativos, etc.), y una categoría puede tener varios softwares.
ExpertoSoftware → Software (1:N)	Cada software puede estar supervisado por un experto, pero un experto puede ser responsable de múltiples softwares.

Ejercicio #7

La universidad necesita un sistema para gestionar la información de los estudiantes, los cursos ofrecidos y las secciones en las que los estudiantes se inscriben. Actualmente, los registros y calificaciones se manejan de forma manual, lo que provoca errores en el seguimiento del rendimiento académico y la gestión de inscripciones. El sistema debe permitir registrar a los estudiantes con sus datos académicos y demográficos, gestionar los cursos y asignar secciones de manera automática, registrando también las calificaciones obtenidas por los estudiantes en cada curso. Además, los estudiantes deben poder consultar su progreso académico, y el sistema debe permitir agregar, modificar y buscar tanto estudiantes como cursos y secciones

Clase	Atributos	Métodos
Estudiante	- numeroEstudiante (int) - creditosCompletados (int) - calificacionPromedio (double) - departamento (String) - mayor (String) - menor (String)	+ cambiarEstudiante() + buscarEstudiante() + graduarEstudiante() + inicializar() + estudianteCompleto() + verEstudiante()
Curso	- numeroCurso (int) - descripcionCurso (String) - numeroDeCreditos (int) - numeroDepartamento (int)	+ agregarCurso() + cambiarCurso() + buscarCurso()
Sección	- numeroEstudiante (int) - numeroCurso (int) - anio (int) - semestre (String) - calificacion (double)	+ agregarSeccion() + cambiarCalificacion() + inscribirEstudiante() + registrarCalificacion() + retirarEstudiante()

Relación	Participantes	Multiplicidad	Descripción
Un Estudiante toma una Sección	Estudiante ↔ Sección	1..* ↔ 1..*	Cada estudiante puede estar inscrito en una o más secciones, y cada sección debe tener al menos un estudiante.
Una Sección pertenece a un Curso	Sección ↔ Curso	0..* ↔ 1	Una sección pertenece a un único curso, mientras que un curso puede tener ninguna o varias secciones asociadas.

Multiplicidad	Significado
1..*	Un participante debe estar asociado con al menos uno y puede estar asociado con múltiples instancias del otro participante.
..*	Un participante puede no estar asociado con ninguna instancia del otro participante, pero también puede estar asociado con múltiples instancias.

Elemento	Detalles
Estudiante ↔ Sección	Un estudiante debe tomar al menos una sección. Esto permite registrar múltiples asignaturas (o cursos) que toma el estudiante en un semestre
Sección ↔ Curso	Una sección corresponde únicamente a un curso, permitiendo organizar y registrar los estudiantes que pertenecen a dicha sección y su calificación en ese curso

Ejercicio #8

El sistema de gestión académica tiene como objetivo organizar y administrar los Departamentos y los Cursos ofrecidos por la universidad. Un Departamento puede tener varios Cursos, pero un Curso no necesariamente depende de un Departamento para su existencia, lo que implica una relación de agregación entre ambas clases. Los Cursos consisten en varias Asignaciones que están directamente relacionadas con Exámenes. Los Exámenes forman parte de las Asignaciones, por lo que existe una relación de composición entre ellos. Además, cada Curso tiene un conjunto de LibrosTexto que son necesarios para los estudiantes, estableciendo una relación de asociación entre el Curso y los LibrosTexto, pero no de composición, ya que los LibrosTexto pueden existir independientemente del Curso.

Clase	Atributos	Métodos
Departamento	nombreDepartamento: String, sillaDepartamento: String	+agregarDepartamento(), +verDepartamento()
Curso	numeroCurso: String, descripcionCurso: String, numeroCreditos: int, numeroDepartamento: String	+agregarCurso(), +cambiarCurso(), +buscarCurso()
LibroTexto	ISBN: String, autor: String, titulo: String, edicion: String, editorial: String, requerido: bool	+agregarTexto(), +cambiarTexto(), +buscarTexto(), +quitarTexto()
Asignacion	numeroAsignacion: String, descripcionAsignacion: String, puntosAsignacion: int, fechaEntregaAsignacion: Date	+agregarAsignacion(), +cambiarAsignacion(), +verAsignacion()
Examen	numeroExamen: String, nombreExamen: String, puntosExamen: int, versionExamen: int	+agregarExamen(), +cambiarExamen(), +buscarExamen()

Relación	Tipo de Relación	Explicación
Departamento - Curso	Agregación (1..*)	Un Departamento tiene varios Cursos, pero un Curso no depende completamente del Departamento. Si el Departamento es eliminado, los Cursos pueden seguir existiendo independientemente. La relación es de agregación, ya que los Cursos son parte de un Departamento, pero no dependen exclusivamente de él
Curso - LibroTexto	Asociación (1..*)	Un Curso puede tener varios LibrosTexto asociados, pero no hay dependencia directa entre ellos. Si el Curso es eliminado, los LibrosTexto no se eliminan necesariamente. Es una relación de asociación, no de composición.

Relación	Tipo de Relación	Explicación
Curso - Asignación	Composición (1..*)	Un Curso tiene varias Asignaciones, que son componentes fundamentales de dicho curso. Si el Curso es eliminado, las Asignaciones también lo son, ya que no existen sin el Curso.
Asignación - Examen	Composición (1..*)	Una Asignación tiene uno o más Exámenes asociados. Si la Asignación es eliminada, los Exámenes también desaparecen, ya que no pueden existir sin la Asignación. Esta es una relación de composición.

Nuestra comunidad

@latecnologiaavanza

latecnologiavanza

@latecnologiaavanza

La Tecnología Avanza (Comunidad)

"Empieza a aprender ahora porque la tecnología avanza y tú también"