TAREA #1 - ARQUITECTURA DE SOFTWARE

Juan Sebastian Jaramillo Silva - 30000056254

1. VIDEO JUEGO DE ROL

Se requiere rediseñar el video juego “Dragon ball z Kakarot” de tal forma que aumente la expectación de los jugadores en el mundo del “Gaming”, para ello los requisitos del sistema son:

\*El videojuego contará con personajes principales, secundarios y simulados a lo largo de la historia, contando con el tipo de rol del personaje, de tal forma aparecerán enemigos canónicos y simulados.

\*Existen objetos no funcionales el cual se componen de ambientación y vegetación en el mapa tales como: (Árboles, montañas, aldeas, edificaciones) disponiendo de unas coordenadas específicas para su posicionamiento.

\*Existen objetos funcionales los cuales únicamente los personajes principales podrán: recoger monedas, alimento, minerales, orbes Z, regalos, esferas del dragón, dichos objetos mejoran estadísticas/habilidades, vida, energía o construcción de máquinas. disponiendo de unas coordenadas específicas para su posicionamiento.

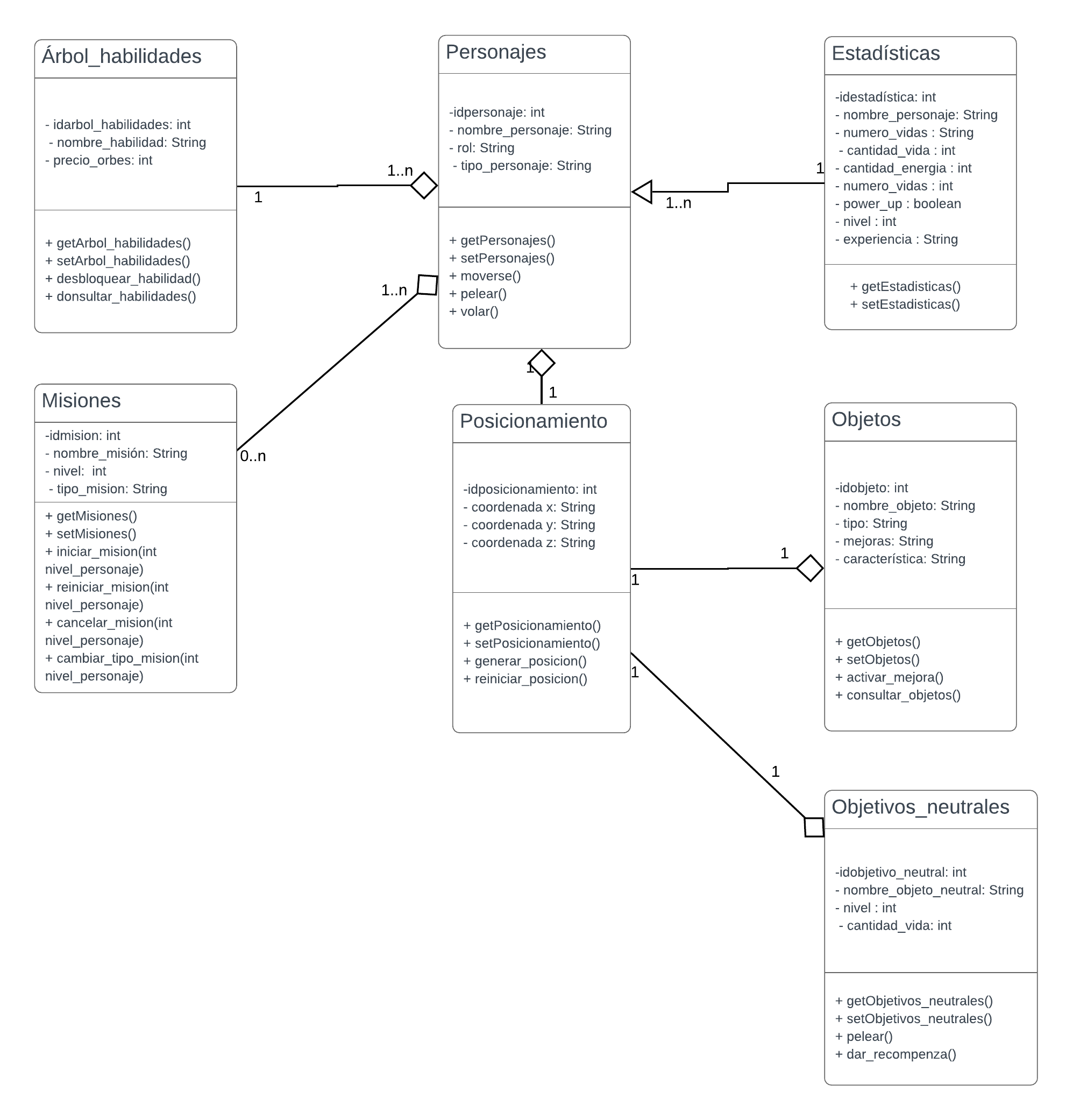
\*En mundo abierto hay objetos neutrales donde existen torres, naves espaciales o satélites los cuales albergan una recompensa sólo si son destruidos, dicha recompensa aumenta cantidad de experiencia y da objetos funcionales. Se dispone de unas coordenadas aleatorias para su posicionamiento

\*Todos los objetos y personajes contarán con coordenadas isométricas (x,y,z) para su posicionamiento aleatorio o específico.

\*De acuerdo con los personajes hay estadísticas específicas: Los personajes principales, secundarios y simulados, así como los enemigos canónicos y simulados tendrán: Número de vidas, cantidad de vida, cantidad de energía, Power Up, nivel, experiencia, los objetos neutrales contarán con nivel, y cantidad de vida.

\*Sólo los personajes principales y secundarios tendrán un árbol de habilidades el cual permite aprender y mejorar habilidades necesitando de orbes Z para desbloquearlos.

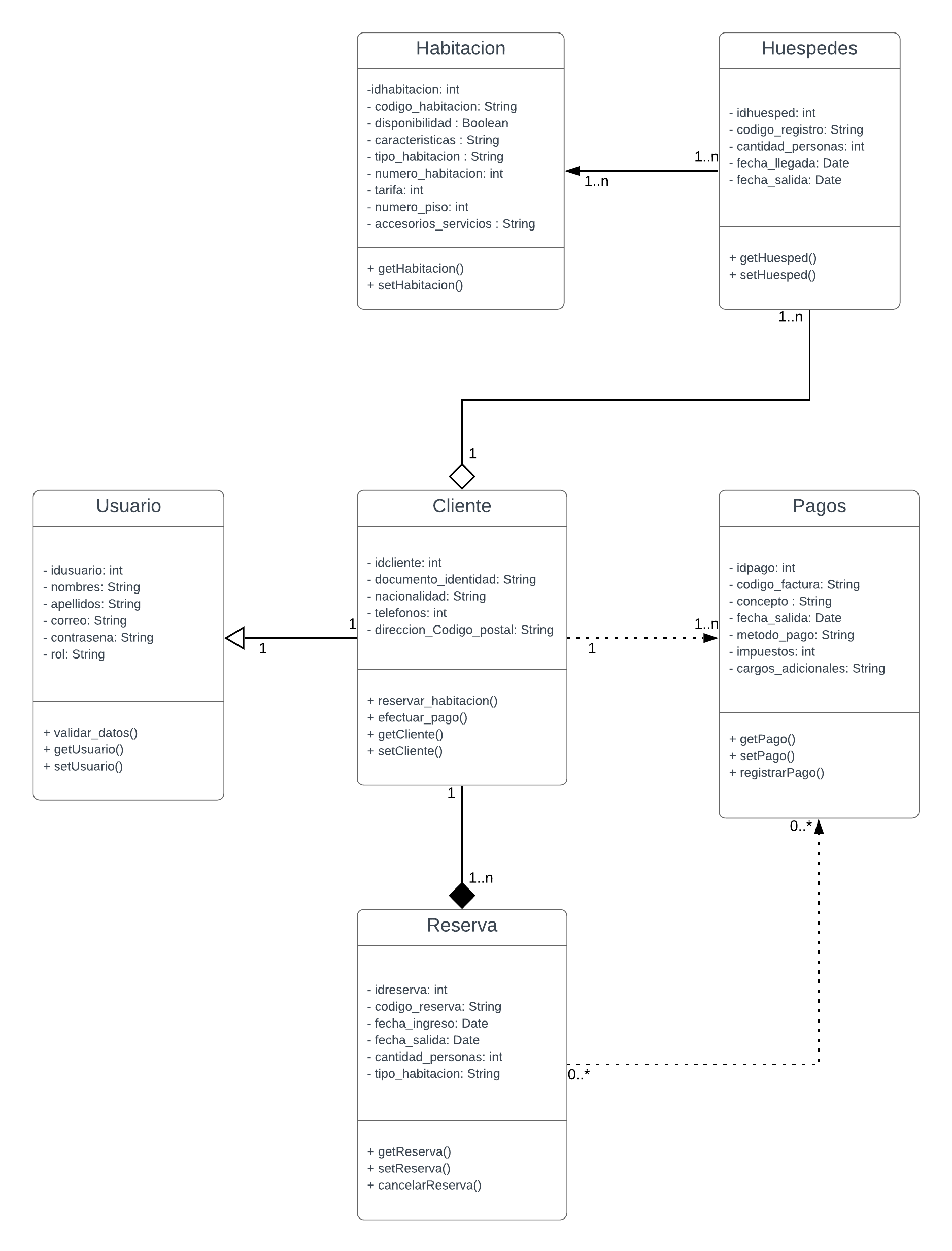
\* El juego tendrá misiones principales realizadas únicamente por personajes principales con enemigos principales, misiones secundarias realizadas únicamente por personajes principales y secundarios con enemigos principales y simulados, y mundo abierto el cual dependiendo del trayecto de la historia principal se juega con su respectivo(s) personaje(s) principal(es), el cual cada misión se compone de un nombre, nivel, tipo de misión.



1. RESERVACIÓN DE HABITACION HOTEL

Se requiere un software el cual permita hacer reservaciones de habitaciones en un hotel, para ello:

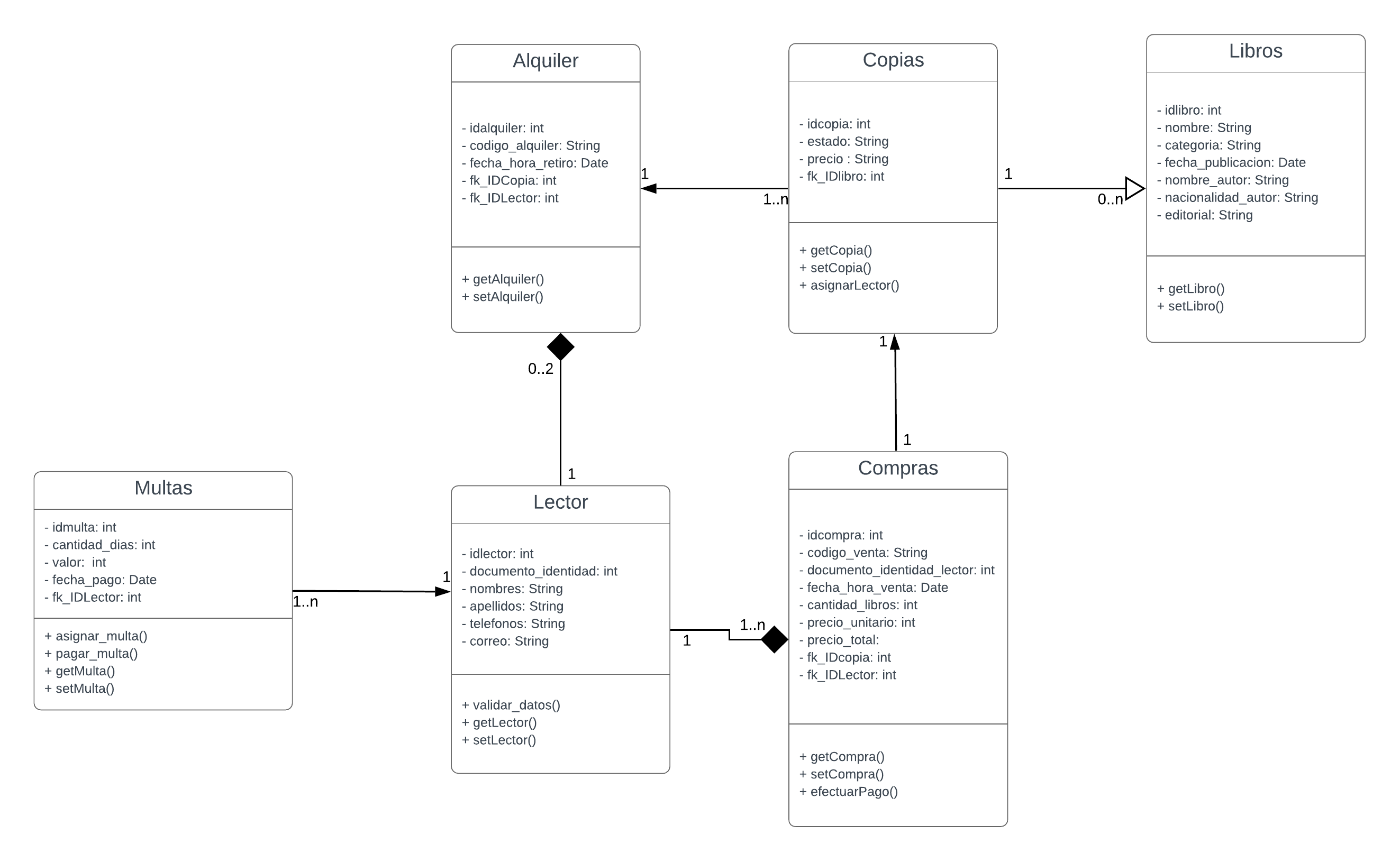
* El usuario deberá registrarse con sus datos personales tales como nombres, apellidos, correo, contraseña, el cual permitirá el ingreso y actualización de sus respectivos datos.
* El cliente, mediante el usuario registrado debe proporcionar datos adicionales como: Documento de identidad, nacionalidad, teléfonos, dirección o código postal.
* Las reservas se registran, verifican o cancelan, los atributos de dicha reserva son: Código, fecha ingreso y salida, cantidad de personas y tipo de habitación.
* Las habitaciones se podrán consultar o actualizar de acuerdo a su código y su disponibilidad, cada habitación contará con características, tipo, número de habitación, tarifa, piso y accesorios o servicios adicionales.
* La cantidad de huéspedes por cada reservación se registrarán al sistema mediante el código de registro, así mismo estará anidado al código de la reserva añadiendo la cantidad de personas y su fecha de llegada/salida.
* Los pagos se efectuarán una vez la reserva haya sido registrada sin haberla cancelado, se registrarán datos como Código de factura, concepto, fecha salida, Método de pago, impuestos y cargos adicionales (sólo si aplica).
* Para realizar modificaciones de habitaciones, un usuario con rol de administrador podrá efectuarlo.



COMPRA O ALQUILER DE LIBROS

1. Se realiza venta y préstamo de libros, para ello se necesita distribuir las copias de los mismos, dichos libros se componen de un nombre, categoría, año, autor, editorial, nacionalidad.

* El lector se compone de: Documento de identidad, nombres, apellidos, teléfonos, dirección y correo.
* Cada copia tiene un identificador, un estado sea prestado, disponible o vendido y un precio en el caso de ser vendido.
* Cada lector puede obtener no más de 2 copias prestadas y cualquier cantidad de libros comprados
* Se puede comprar cualquier cantidad de libros, para ello se genera una factura de venta el cual se compone del código de venta, código del libro, fecha y hora de venta, cantidad de libros, precio unitario y precio total.
* Si se alquila una copia el registro de alquiler debe tener el código del lector, el código del libro, fecha y hora de retiro.
* Si el lector no devuelve la copia en un máximo de 30 días, pagará una multa.
* En la multa se registrará los días excedidos de la devolución del libro, el valor de la multa y la fecha de pago, a su respectivo lector.



PUNTOS DE INVESTIGACIÓN:

2) ¿Qué es un patrón de Software

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse un millón de veces sin tener que hacer dos veces lo mismo.

Definición: Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular.

Elementos de un patrón de software

* Nombre: describe el problema de diseño.
* El problema: describe cuándo aplicar el patrón.
* La solución: describe los elementos que componen el diseño, sus relaciones, responsabilidades y colaboración.

Según su propósito se clasifican en:

* De creación: conciernen al proceso de creación de objetos.
* De estructura: tratan la composición de clases y/o objetos.
* De comportamiento: caracterizan las formas en las que interactúan y reparten responsabilidades las distintas clases u objetos.

3) ¿Qué es arquitectura de software?

Concepto: La Arquitectura de Software se refiere a “las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.

Definición: Es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

4) Características de un arquitecto de software

El Arquitecto de Software debe ser una persona con amplios conocimientos técnicos, gran experiencia en programación, liderazgo y que ejerza las siguientes funciones:

* Gestión de los requisitos no funcionales y definición de la Arquitectura de Software
* Selección de la Tecnología
* Mejora continua de la Arquitectura
* Facilitador
* Líder y Formador
* Aseguramiento de la Calidad

5) ¿Que es TDD?

DD o Test-Driven Development (desarrollo dirigido por tests) es una práctica de programación que consiste en escribir primero las pruebas (generalmente unitarias), después escribir el código fuente que pase la prueba satisfactoriamente y, por último, refactorizar el código escrito. Con esta práctica se consigue entre otras cosas: un código más robusto, más seguro, más mantenible y una mayor rapidez en el desarrollo.

El proceso de diseño de software, combinando TDD con metodologías ágiles, sería el siguiente:

1. El cliente escribe su historia de usuario.
2. Se escriben junto con el cliente los criterios de aceptación de esta historia, desglosándolos mucho para simplificarlos todo lo posible.
3. Se escoge el criterio de aceptación más simple y se traduce en una prueba unitaria.
4. Se comprueba que esta prueba falla.
5. Se escribe el código que hace pasar la prueba.
6. Se ejecutan todas las pruebas automatizadas.
7. Se refactoriza y se limpia el código.
8. Se vuelven a pasar todas las pruebas automatizadas para comprobar que todo sigue funcionando.
9. Volvemos al punto 3 con los criterios de aceptación que falten y repetimos el ciclo una y otra vez hasta completar nuestra aplicación.