

PARADIGMAS DE PROGRAMACION

Trabajo Práctico N° 4

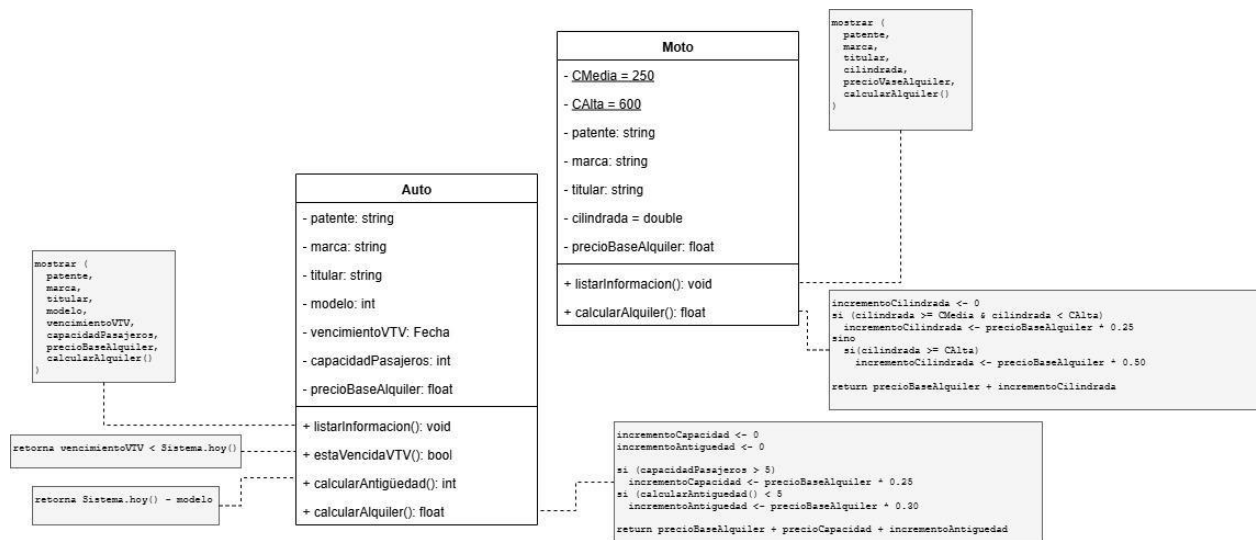
Fecha: 10/09/25

Tema: Diseño Orientado a Objetos. UML

1)

a. Diseñe en UML las clases *Auto* y *Moto* teniendo en cuenta la descripción en cada caso:

- Un *Auto* tiene una patente, la marca, el modelo (año que entró en circulación), fecha de vencimiento de la VTV, el nombre del titular del vehículo, la capacidad de personas que puede trasladar y el precio base de alquiler por día. Dado un *Auto* se debe poder listar su información por pantalla, obtener la patente y el nombre del titular, determinar si tiene la VTV vencida, calcular su antigüedad y el costo de alquiler, teniendo en cuenta que, si la capacidad de personas que puede trasladar el auto es superior a 5 personas, el precio base de alquiler se incrementa en un 25%, por otro lado, si el auto tiene menos de 5 años de antigüedad el costo de alquiler calculado se debe incrementar en un 30%.
- Una *Moto* posee una patente, la marca, el nombre del titular, la cilindrada y el precio base de alquiler por día. La cilindrada de una moto puede ser *baja* (hasta 250 cc), *media* (entre 250 cc y 600 cc) y *alta* (más de 600 cc). Dada una *Moto* se debe poder listar su información por pantalla, obtener la patente, el nombre del titular y la cilindrada de la moto y calcular el costo de alquiler que depende de la cilindrada: las motos de baja cilindrada pagan el precio base, pero si la moto es de media o alta cilindrada el precio base se incrementa un 25% y un 50% respectivamente.



b. Dadas las clases del punto anterior, identifique qué características tienen en común y cuáles son las diferencias. En base a esta información, diseñe en UML una relación de generalización adecuada.

PARADIGMAS DE PROGRAMACION

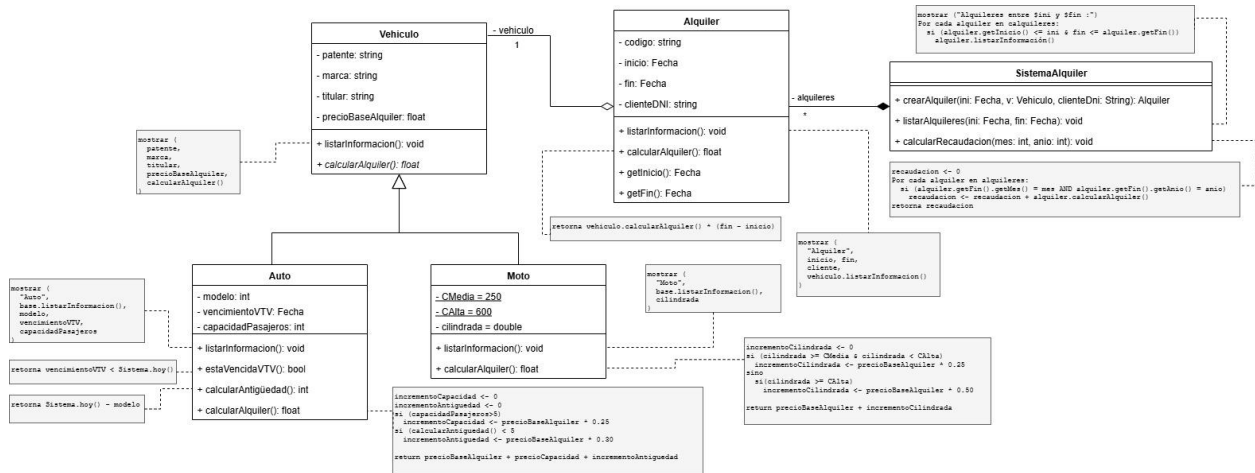
Trabajo Práctico N° 4

Fecha: 10/09/25

c. Complete el diagrama de clases anterior teniendo en cuenta la siguiente información:

La empresa de transporte *ParadigmasUNT* alquila autos y motos y desea implementar un sistema que le permita digitalizar la información de los alquileres y llevar un control de los mismos. Para ello, tenga en cuenta que:

- De cada alquiler se conoce el código, la fecha de inicio y fin, DNI del cliente y el vehículo que se alquiló. Dado un alquiler, se debe poder listar su información y calcular el costo del alquiler teniendo en cuenta la cantidad de días.
- El sistema de la empresa debe poder crear un alquiler, llevar un registro de los alquileres, listar la información de los alquileres realizados en una determinada fecha y calcular el monto total recaudado en un determinado mes.



2) Dada la siguiente descripción:

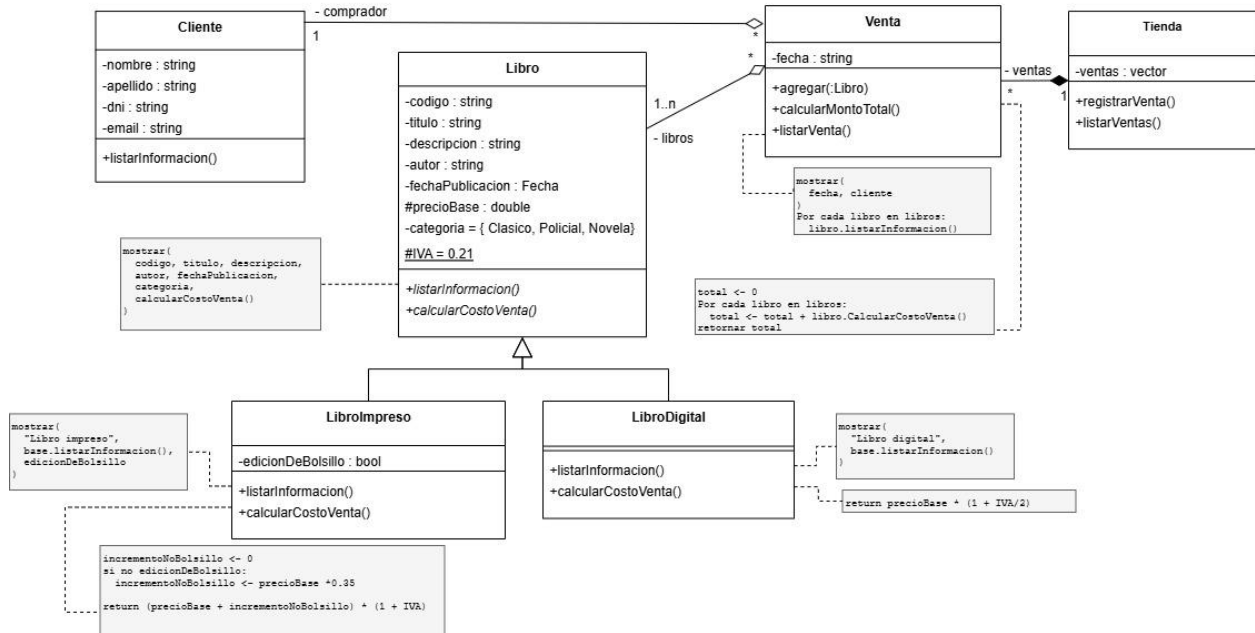
En una tienda de libros se desea modelar el sistema para la venta de libros online que pueden ser digitales (e-books) o impresos. De los **libros** se conoce el código, título, descripción, autor, la fecha de publicación y el precio base de venta. Además de los libros se conoce la categoría (clásico, policial, novela). De los libros impresos se conoce además si es una edición de bolsillo o no. Los objetos de la clase Libro deben ser capaces de listar su información y calcular el costo de venta que depende del tipo de libro (digital o impreso). El precio de los libros impresos se incrementa en un 35% si la edición no es de bolsillo. Además, al precio del libro se le debe agregar el incremento del IVA. Para beneficiar la ecología, los libros digitales solo pagan la mitad del valor del IVA. El sistema de la tienda debe mantener un registro de las ventas realizadas. De una venta se conoce la fecha en la que se produjo, información del cliente y los libros que se vendieron. Dada una venta, debe ser posible calcular el monto total de la venta.

- Identifique las clases intervinientes y la responsabilidad de cada una (atributos y métodos).
- Indique el tipo de relación que existe entre las clases, en caso de corresponder.
- En base a la información anterior, diseñe el diagrama de clases UML correspondiente.

PARADIGMAS DE PROGRAMACION

Trabajo Práctico N° 4

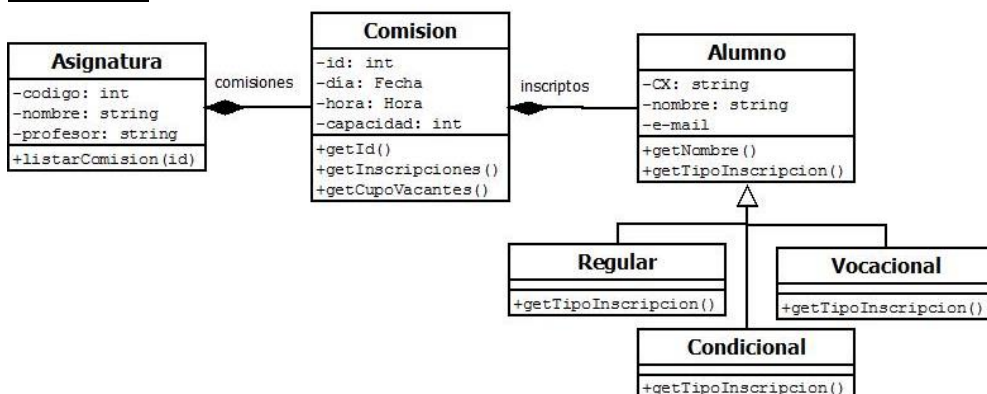
Fecha: 10/09/25



3) Analice los siguientes diagramas de clases simplificados e indique cuál de ellos representa una mejor solución al siguiente problema. Justifique su respuesta.

En la secretaría del Dpto. Cs. de la Computación se desea implementar la inscripción de los alumnos a las distintas comisiones de cursado de las asignaturas. De cada asignatura se conoce el código, nombre y profesor encargado, además de las distintas comisiones de cursado. Cada comisión se identifica con un número y se registra el día y horario, la capacidad máxima de alumnos que se pueden inscribir en la misma y las inscripciones de alumnos. Dada una comisión se puede obtener los alumnos inscriptos y el cupo de lugares vacantes. El alumno puede inscribirse en la comisión de una asignatura como alumno Condicional, Regular o Vocacional. El tipo de inscripción que realiza el alumno en una determinada asignatura depende de las condiciones de inscripción de dicha asignatura (materias correlativas, condiciones de cursado, etc.) y de la situación particular de cada alumno (tiene las correlativas solicitadas, cursa otra carrera, etc.).

Solución A

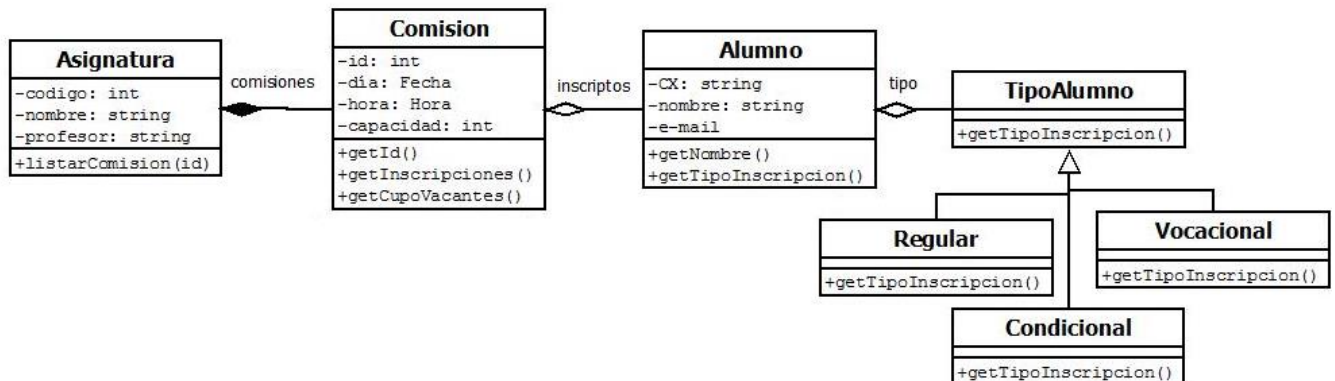


PARADIGMAS DE PROGRAMACION

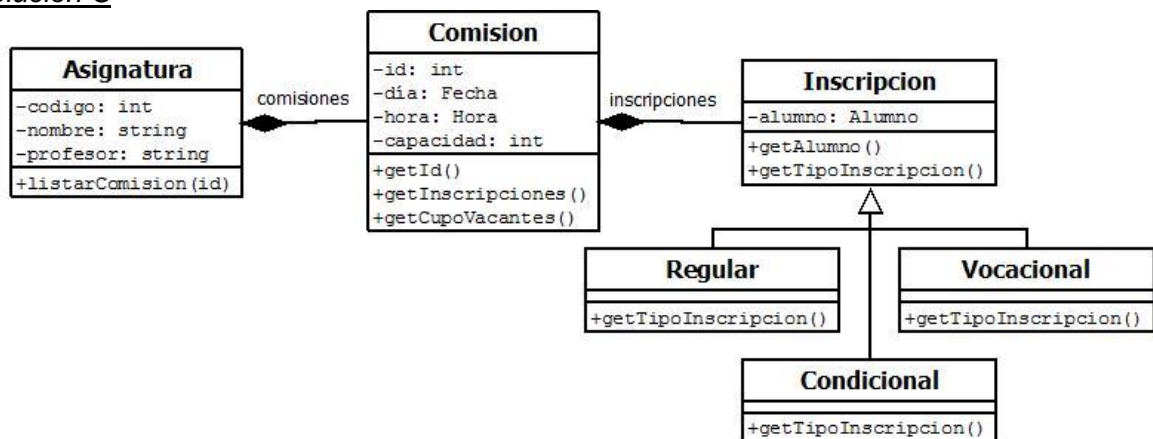
Trabajo Práctico N° 4

Fecha: 10/09/25

Solución B



Solución C



Solución A:

- Por la **composición** entre *Comision* y *Alumno*, un alumno no puede existir sin una comisión, lo cual es incorrecto, ya que un alumno existe independientemente de si está o no en una comisión. La vida de un objeto alumno no está ligada a la de un objeto comisión. Esta composición determina que un alumno es una parte integral y exclusiva de una comisión específica, lo cual es incorrecto. A lo sumo, la relación entre *Comision* y *Alumno* tiene que ser débil (esto es, agregación) para que modele de alguna manera el acto de inscripción.

- Modela el "tipo de inscripción" como una cualidad intrínseca y permanente del alumno (no hay una abstracción para inscripción), lo cual no es correcto. El tipo de inscripción es una característica de la inscripción, no del alumno en sí mismo. Además, está asumiendo incorrectamente que el tipo de alumno es una cualidad permanente, lo que no se ajusta a la realidad del problema.

- En resumen, el alumno solo puede tener un único tipo de inscripción en todo el sistema. No puede ser Regular en una materia y Condicional en otra porque solo puede estar inscripto en una sola.

PARADIGMAS DE PROGRAMACION

Trabajo Práctico N° 4

Fecha: 10/09/25

Solución B:

- Introduce `TipoAlumno` y sus especializaciones para modelar los distintos tipos de inscripciones.
- Por la **agregación** entre `Comision` y `Alumno`, un alumno puede existir independientemente de si está o no inscripto en una comisión y puede estar en más de una comisión, lo que mejora el diseño de representar el acto de inscripción, pero no del todo.
- El tipo de inscripción sigue siendo una propiedad del alumno, similar a la Solución A, lo que contradice el hecho de que el tipo de inscripción puede cambiar según la materia. Aunque el alumno debe poder cambiar su "tipo" de inscripción, este diagrama modela que un alumno solo tiene un tipo de inscripción para todas las comisiones en las que se inscriba, no captura la lógica de que el tipo de inscripción puede variar por asignatura.

Solución C:

- Esta solución es la más robusta y correcta. Introduce una clase intermedia, `Inscripcion` para modelar explícitamente el acto de inscribirse.
- La clase `Inscripcion` representa el hecho de que un `Alumno` puede inscribirse en una comisión. Esta clase intermedia es la que tiene la relación de **composición** con `Comision` (muchas inscripciones por comisión) y cada inscripción con el alumno (una inscripción por alumno).
- El diagrama no muestra la clase `Alumno` dibujada, pero sí el atributo de `Inscripcion` "alumno: `Alumno`" que denota una asociación de `Inscripcion` con `Alumno`. El diagrama no dice nada si esta asociación es de composición o agregación, queda en el lector esta interpretación.
- El tipo de inscripción (`Regular`, `Vocacional`, `Condicional`) se modela correctamente como una característica de la `Inscripcion`, no del `Alumno` como en los casos anteriores. Esto permite que un mismo alumno pueda tener diferentes tipos de inscripciones en distintas comisiones.

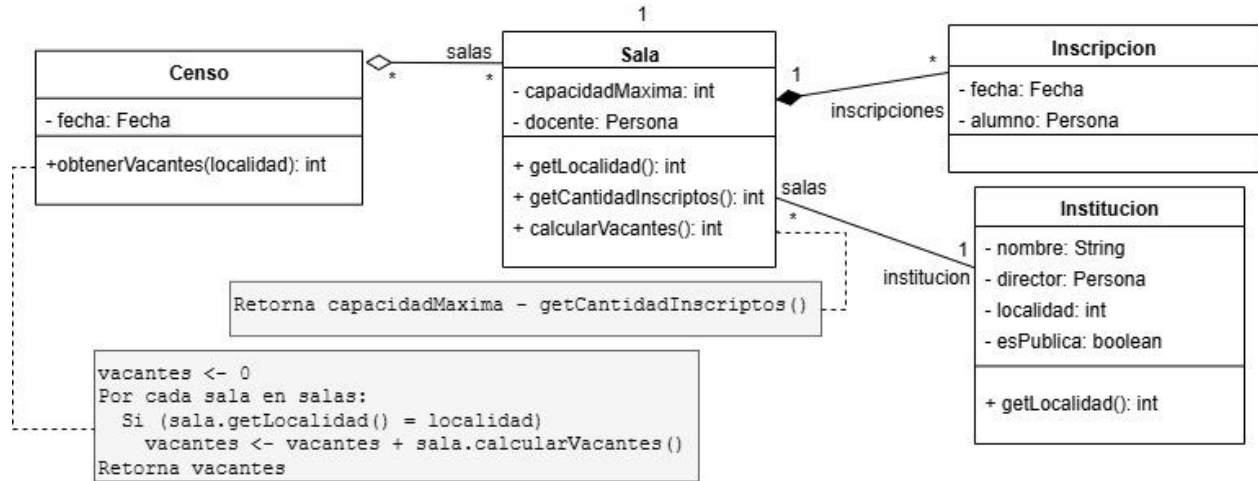
4) En los siguientes apartados identifique las clases que intervienen en el problema, las relaciones entre clases, las responsabilidades de cada clase y diseñe el diagrama de clases detallado del problema.

- a. El Gobierno de Tucumán necesita realizar un censo de todas las salas de Jardín de 5 años que funcionan en la provincia para adoptar medidas de crecimiento de acuerdo a las necesidades de la población. De cada sala se necesita saber la localidad a la que pertenece, su capacidad máxima, información del docente a cargo de la misma, alumnos inscriptos para el próximo año, información de la institución a la que pertenece la sala como ser el nombre de la institución, si es pública o privada, información sobre el director de dicha institución, etc. Con la información recabada, el gobierno necesita determinar cuántas vacantes se registran por localidad.

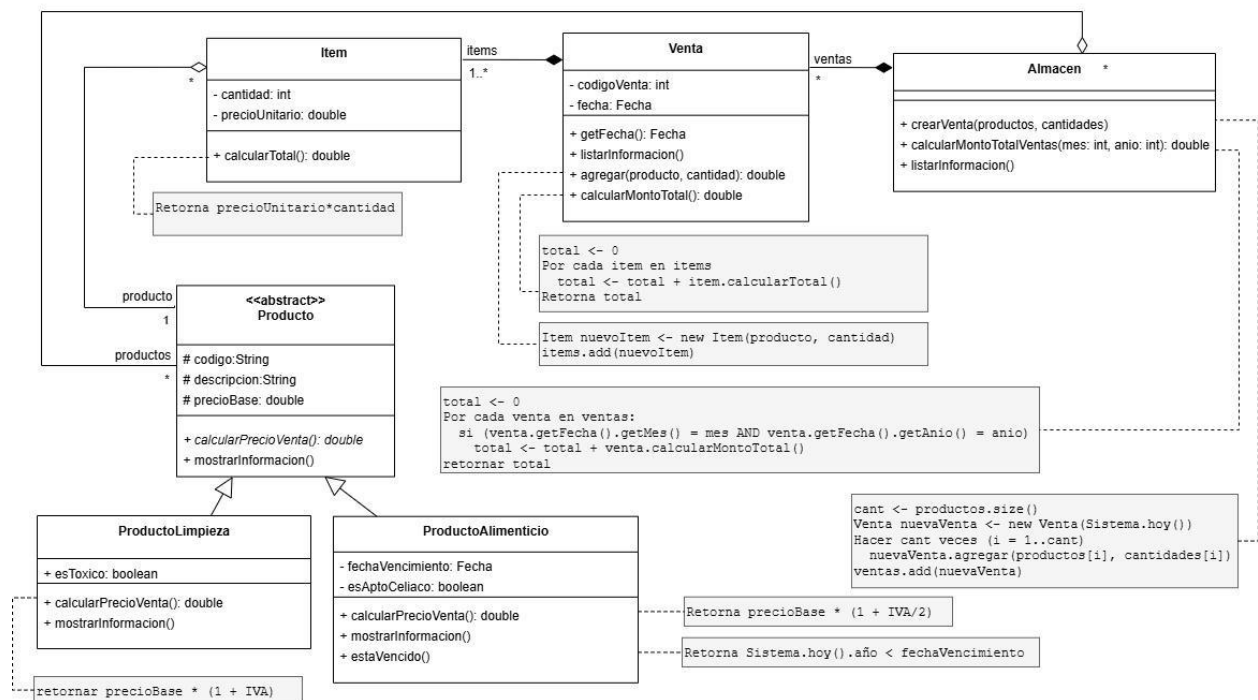
PARADIGMAS DE PROGRAMACION

Trabajo Práctico N° 4

Fecha: 10/09/25



- b. Considere un almacén que vende productos alimenticios y de limpieza. De los productos de limpieza se conoce el código, una descripción, una advertencia de si es tóxico y un precio base. Dado un producto de limpieza se puede escribir su información por pantalla, obtener el código, si es tóxico o no y cuánto cuesta teniendo en cuenta que, para la venta, al precio base se le debe incrementar el IVA (21%). Un producto alimenticio posee un código, el mes y año de vencimiento, una descripción, una leyenda de si es un alimento apto para celíacos y el precio. Un producto alimenticio permite escribir su información por pantalla, obtener su código de identificación y su descripción, si está vencido o no, si es apto para celíacos o no y su precio de venta. Dado que los productos alimenticios se encuentran subsidiados, el precio de venta corresponde al precio base con un incremento del 10,5% en concepto de IVA. El almacén desea llevar un registro de todas las ventas, de manera de poder calcular el monto total de ventas realizadas a fin de mes. De cada venta se conoce el código de venta, la fecha en que se realizó y los productos que se vendieron. Dada una venta, se debe poder listar la información y calcular el monto total de la misma.



PARADIGMAS DE PROGRAMACION

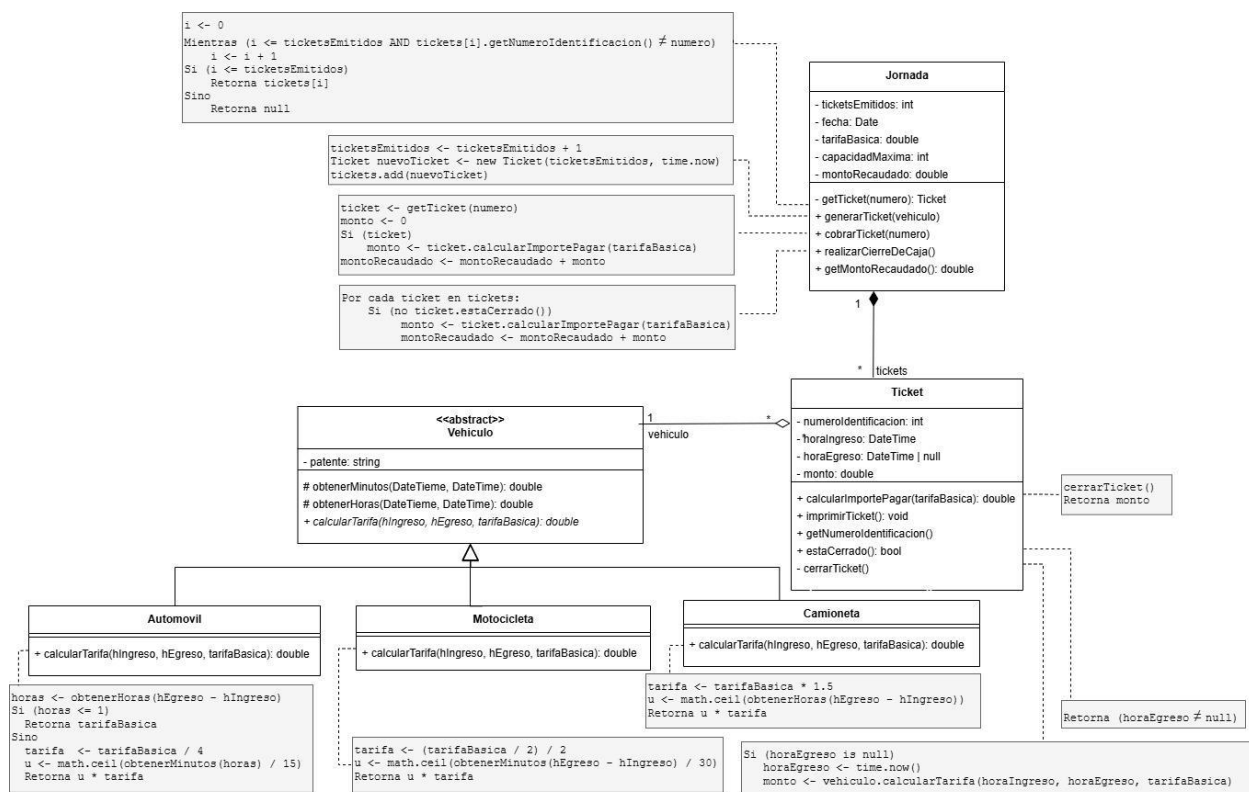
Trabajo Práctico N° 4

Fecha: 10/09/25

c. Un estacionamiento desea automatizar el registro diario de vehículos para simplificar la tarea del administrador. En una jornada de trabajo, cada vez que ingresa un vehículo al estacionamiento se emite un Ticket con un número único de identificación por día, información del vehículo y hora de ingreso. Cuando el vehículo se retira del estacionamiento, el sistema debe calcular el importe que debe pagar en función del tiempo de estadía en el lugar y el tipo de vehículo. Si el vehículo es:

- **Motocicleta:** se cobra por hora la mitad de la tarifa básica y el monto a pagar corresponde a la estadía en el lugar teniendo en cuenta que se fracciona el tiempo cada media hora.
- **Automóviles:** se cobra por hora la tarifa básica completa y se paga por la primera hora el importe completo (haya o no permanecido la hora) y a partir de ahí se cobra el excedente fraccionando el importe cada 15 minutos.
- **Camionetas:** se cobra por hora 1.5 de tarifa básica y se paga por hora, sin fraccionar el tiempo de estadía.

Cada día se inicia una nueva jornada de trabajo en el estacionamiento. De cada Jornada se conoce la fecha, los tickets emitidos durante la misma, la tarifa básica de estacionamiento por hora, la capacidad máxima de estacionamiento disponible ese día y el *monto recaudado** (monto recaudado hasta el momento que se realiza la consulta). Al finalizar la jornada, se debe realizar un cierre de caja. Cuando se solicita el cierre de caja, el estacionamiento debe estar vacío o se debe realizar la salida de todos los vehículos que queden en el lugar. Se debe poder obtener la información del monto recaudado por el estacionamiento en una determinada jornada de trabajo.



NOTA: El diagrama UML debe incluir la información relevante del problema (atributos, métodos y relaciones, niveles de visibilidad, notas con el pseudocódigo de los métodos mencionados y aclaraciones que crea conveniente).