

ASIGNATURA

Programación Orientada Objetos

Tutor: Miquel Piulats

Grados de Ingeniería

- **Informática**
- **Tecnologías de la Información**

Tutor: Miquel Piulats

Programación Orientada a Objetos

Sesión 1

Práctica

29 de marzo de 2025

9.30 a 11 TEAMS.

Programación Orientada a Objetos

Sesión 2

Práctica

10 de mayo de 2025

9.30 a 11 TEAMS.

Fechas Entrega

19 de mayo de 2025

20 de agosto de 2025

Normas de Entrega

La práctica se entregará a través de la tarea definida en el entorno virtual de la asignatura.

Cada alumno creará y comprimirá una carpeta nombrada con el DNI y primer apellido separados por guión:

8345385X-gonzalez

Normas Realización Práctica

- La realización de la práctica es obligatoria.
- Sólo se evaluará el examen si la práctica ha sido previamente aprobada.
- Ha de desarrollarse íntegramente empleando el entorno de desarrollo BlueJ.
- La práctica es un trabajo individual.

Normas Realización Práctica

- El tutor entrará en el espacio virtual de la asignatura dentro antes del 1 de junio, para meter las notas para sus alumnos.
- La única vía de entrega de la práctica es a través de la plataforma Agora.

Entrega de la Práctica

- La práctica se entrega a través de la plataforma Agora.
- Hay que subir un archivo comprimido.
- No se deben usar acentos en los nombres de los archivos ni las carpetas.

Entrega de la Práctica

- El archivo comprimido debe estar compuesto por una carpeta con el nombre del alumno debe contener dos cosas:
 - Memoria.
 - Una carpeta con el código.

La Memoria

- Portada con título “Práctica de Programación Orientada a Objetos – Curso 2024-2025”
- Los datos del alumno:
 - Nombre.
 - Apellidos.
 - Dirección de correo electrónico.
 - Teléfono de contacto.

La Memoria

- Análisis de la aplicación realizada.
- Mostrando el funcionamiento del programa.
Capturas de cada nivel de evaluación.
- Estrategias implementadas y decisiones de diseño establecidas.
- En general, toda aquella información que haga referencia a las diferentes decisiones tomadas a lo largo del desarrollo de la práctica.
- Justificación de dichas decisiones.

La Memoria

- Diagrama de clases, detallando claramente el tipo de relación entre ellas (uso, agregación, herencia, ...).
- Un texto en el que se describa cada clase/objeto, justificación de su existencia, métodos públicos que contiene y funcionalidad que realizan.

Carpeta con el Código

- Una carpeta con el código: incluyendo todos los ficheros *.java y *.class.
- Así como la Memoria en formato electrónico.
- Preferiblemente html o pdf.

Entrega de la Práctica

- Al hacer la entrega del trabajo se acepta que tanto el código fuente Java como la memoria de la práctica es original.

Entrega de la Práctica

- Si el archivo subido por parte del alumno no sigue estas indicaciones:
 - Está infectado con algún virus.
 - No se puede descomprimir.
- El equipo docente no aceptará la práctica y se calificará con una nota de 0.

Normas del Tutor

- Ayudar a los alumnos al principio del curso con el planteamiento de la práctica y las normas que tienen que seguir.
- El tutor puede dar fragmentos de código fuente a los alumnos.
- Si un alumno va a incluir un fragmento de código en su práctica, debe incluir un comentario al respecto

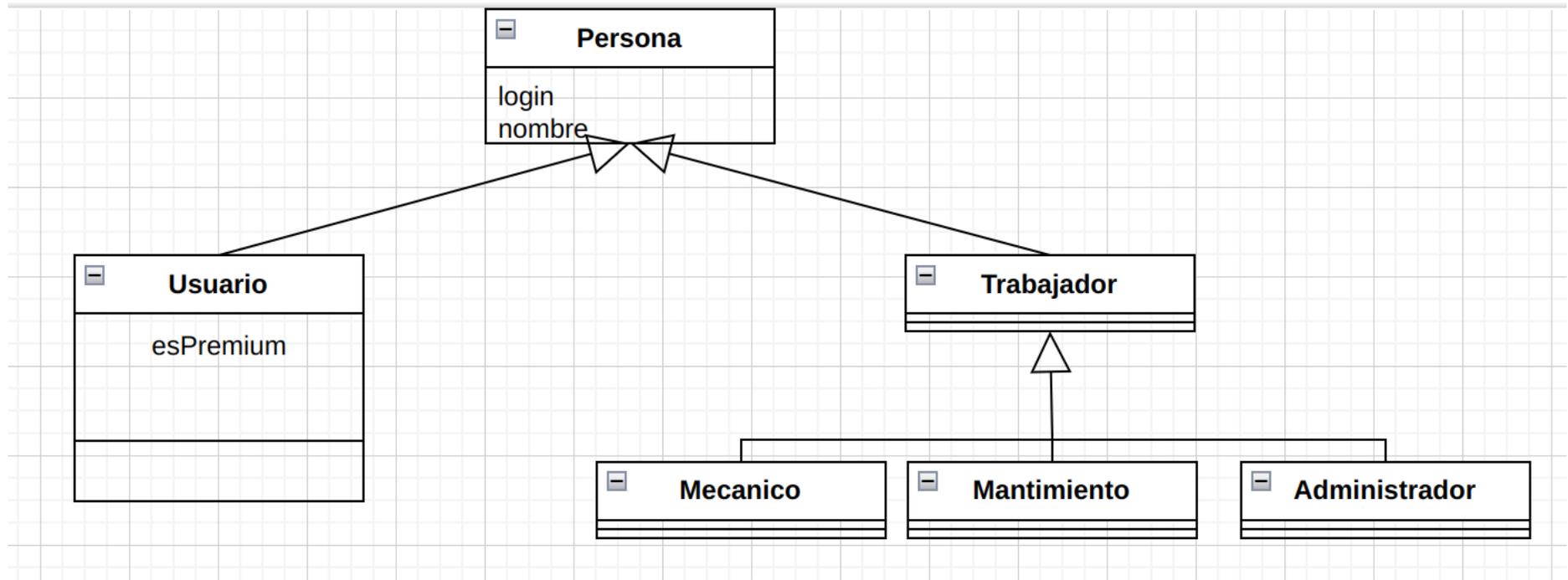
Normas del Tutor

- Una vez terminada y entregada la práctica, el tutor debe entrar en el espacio virtual de la asignatura.
- Antes del 1 de junio
- Para grabar las notas de sus estudiantes.
- Comunicar la calificación a sus alumnos.

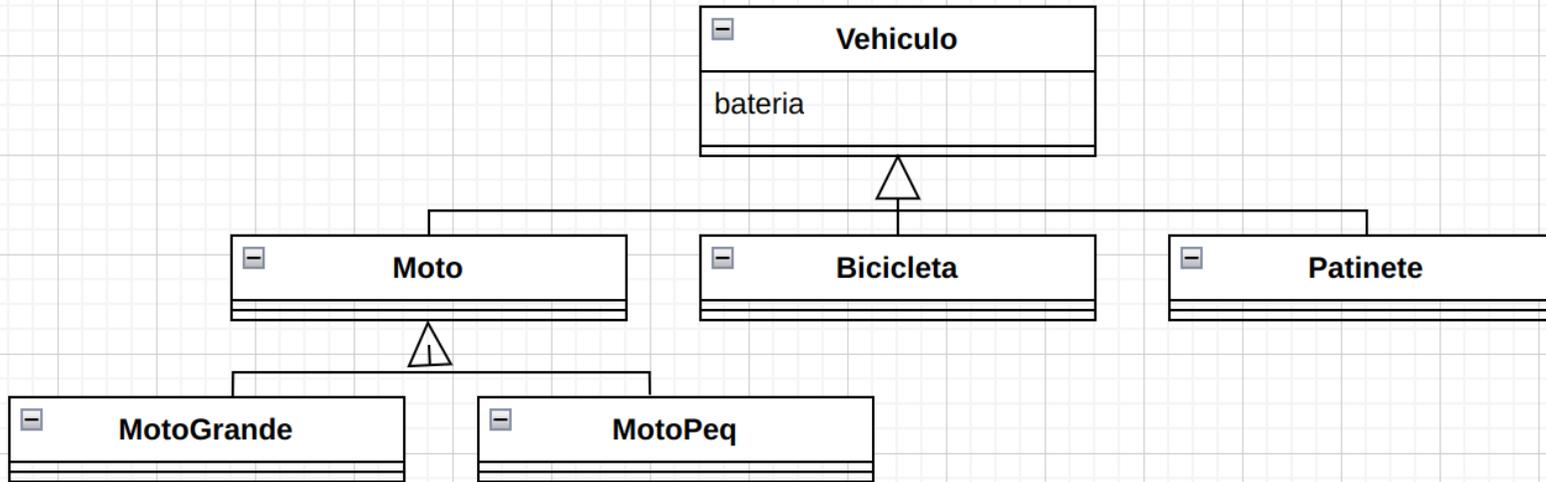
Descripción de la Práctica

Implementar un **sistema integrado de gestión del plan de movilidad sostenible** de una gran ciudad.

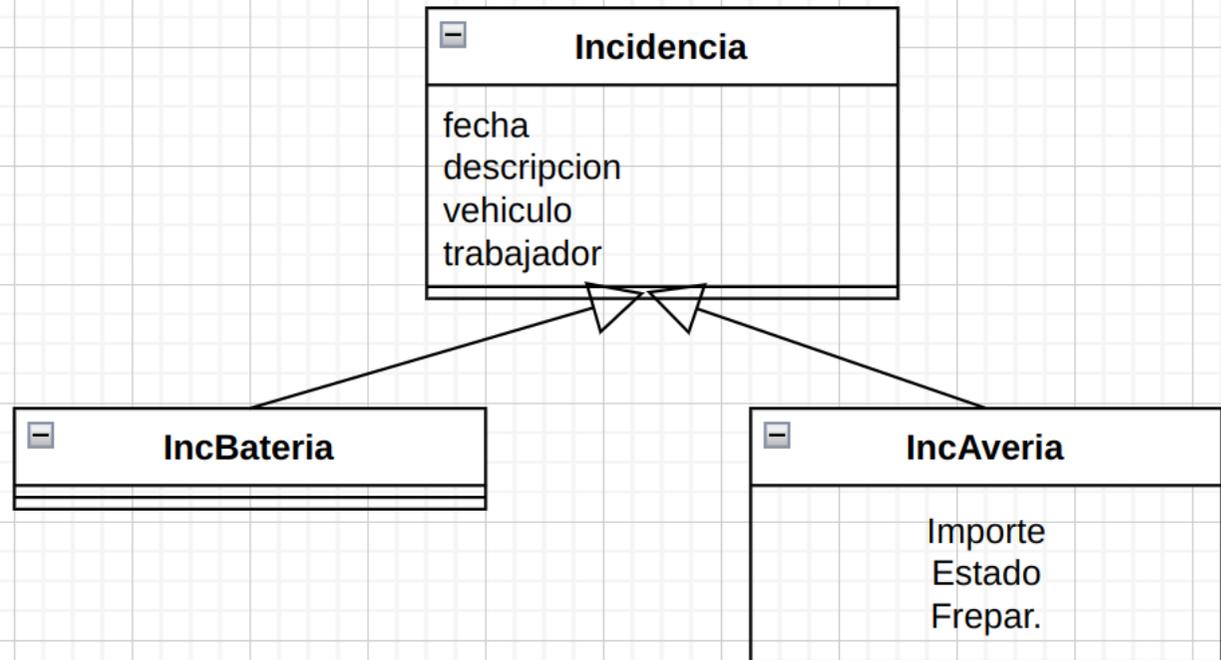
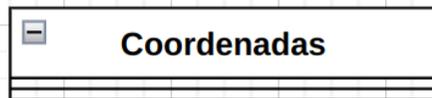
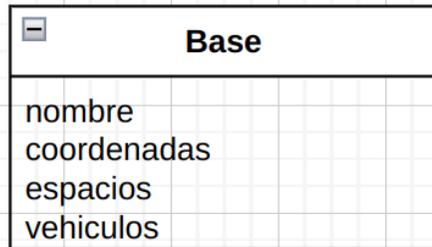
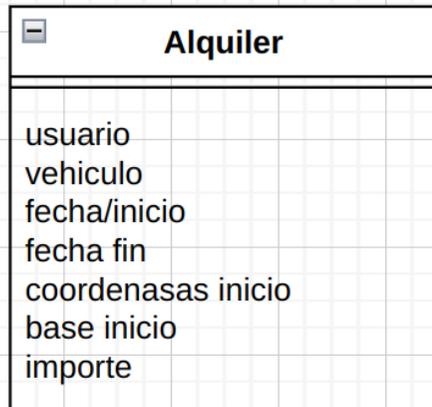
Personas



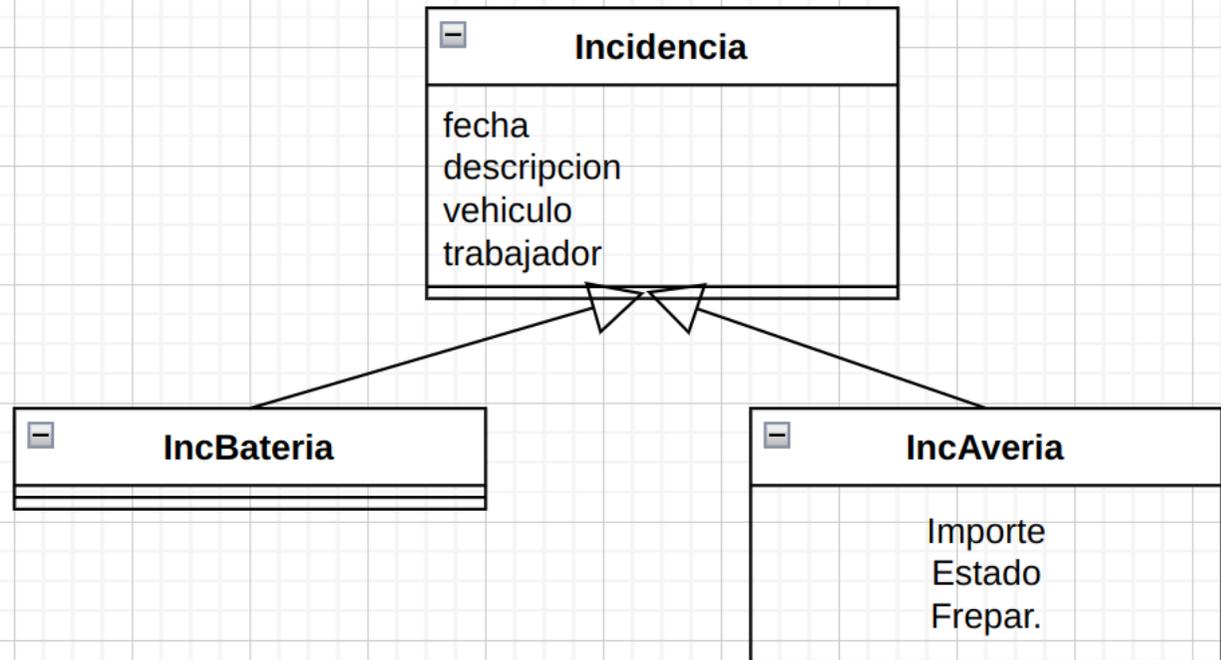
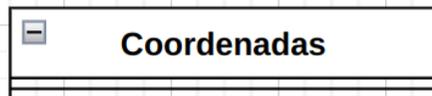
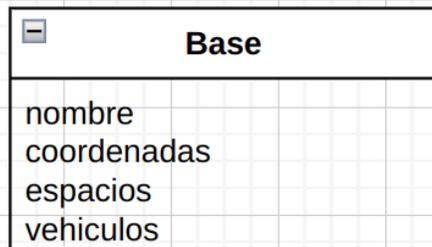
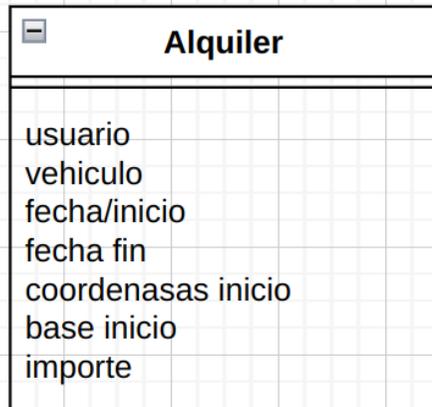
Vehículos



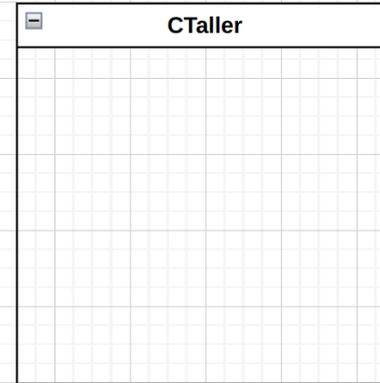
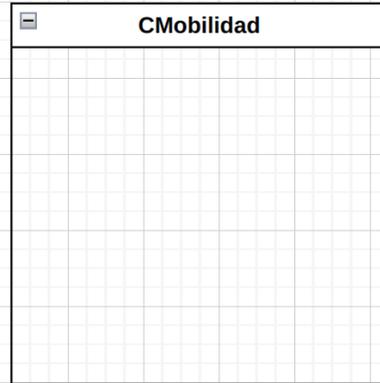
Objetos adicionales



Objetos adicionales



Controladores



Evaluación

NIVEL 1 (3,0 puntos).

Lo que se pretende que el alumno desarrolle en este nivel son las relaciones de clase, herencia y demás que van asociadas al desarrollo de la práctica. Así, se pide realizar las siguientes tareas:

- Planteamiento del problema: actores participantes, relaciones entre actores, funcionalidad a cumplir por la práctica a desarrollar.
- Establecimiento de diferentes clases a intervenir en la práctica, relaciones de dependencia entre clases, identificar diferentes jerarquías de clases, etc.

- Elaboración de un documento escrito (memoria de la práctica) que contenga el primer punto y los correspondientes ficheros para BlueJ que implementen el segundo.

Es necesario contemplar las jerarquías de herencia necesarias para la implementación de lo solicitado en la práctica (tipos de personas, vehículos, etc.).

NIVEL 2(hasta 7,0 puntos).

Los alumnos que implementen este nivel de finalización de la práctica recibirán una

puntuación máxima de 7 puntos. Sólo se podrá optar a este Nivel si se ha implementado

satisfactoriamente y en su totalidad los requerimientos especificados en el Nivel 1. Lo que

se pretende que el alumno desarrolle en este nivel es la parte de gestión de datos del

sistema usando una estructura de clases y métodos apropiados. De este modo, el sistema

deberá permitir lo siguiente:

- Añadir y actualizar los datos de la flota de vehículos del sistema, así como de las tarifas aplicadas y los descuentos para usuarios premium.
- Dar de alta a usuarios del sistema con sus datos personales.
- Realizar el alquiler de vehículos con todas las características asociadas (gestión de la batería, gestión del saldo, introducción avisos por fallos mecánicos, etc.).
- Gestionar toda la funcionalidad relativa a los usuarios premium: consulta de potenciales usuarios premium por parte del administrador,

asignación del estatus de premium a usuarios, gestión de los descuentos y de la posibilidad de realizar reservas.

Gestión de la posibilidad por parte de un usuario premium de alquilar vehículos con un nivel de batería entre un 10% y un 20%.

- Asignar vehículos con baja batería o con fallos mecánicos a encargados de mantenimiento para su retirada, y gestionar.
- Permitir que cada encargado de mantenimiento y mecánico vea los vehículos que tiene asignados y realice sus funciones sobre los mismos, dejando constancia, en el caso de los mecánicos, del importe de las reparaciones realizadas.
- Realizar búsquedas sencillas sobre los usuarios y empleados del sistema.
- Realizar consultas y actualizaciones del stock de vehículos del sistema.

Nivel 3(**Hasta 10,0 puntos**).

Los alumnos que implementen este nivel de finalización de la práctica recibirán una puntuación máxima de 10 puntos. Sólo se podrá optar a este Nivel si se han implementado satisfactoriamente y en su totalidad los requerimientos especificados en el Nivel 2. Lo que se pretende es que el alumno desarrolle en este nivel la interfaz textual del sistema para las funciones identificadas en el nivel 2 más la consulta de vehículos más cercanos y la generación de listados. De este modo, el sistema deberá permitir lo siguiente:

- Gestionar las funciones identificadas en el nivel 2.
- Permitir la búsqueda de los vehículos más cercanos: que el usuario pueda consultar cuáles son las bases de bicicletas o patinetes, o las motos individuales, más cercanas a su posición.

- Producir diferentes listados y estadísticas del funcionamiento del sistema:

- Listado de las bases de bicicletas y patinetes, ordenado según su demanda.

- Listado de los vehículos que han requerido reparaciones en un período de tiempo determinado, incluyendo el número de reparaciones (si hubiera más de una) y el importe de cada una de ellas, así como el importe total.

- Listado de los encargados de mantenimiento y mecánicos, ordenados de mayor a menor número de intervenciones realizadas (vehículos asignados).

- Listado de vehículos en función de su tipo, ordenado por su tiempo de uso, de mayor a menor.

- Listado de los usuarios del sistema, ordenados según sus gastos en alquileres de vehículos dado un período de tiempo determinado.

Es importante considerar que para optar a la calificación de un nivel superior han de cumplirse todas y cada una de las funcionalidades especificadas en el nivel inmediatamente anterior.

Caso de no ser así no se podrá obtener una calificación superior a la marcada por el nivel cuyas restricciones no se cumplen en su totalidad.

Los niveles han de implementarse en el orden que se indican.

No siendo posible implementar niveles no consecutivos para obtener calificaciones superiores.

Para cada uno de los niveles se van a indicar unos requisitos mínimos de cumplimiento.

Esto quiere decir que para cualquier otro detalle de diseño que no se encuentre descrito expresamente en lo indicado en este enunciado, el alumno tiene libertad para tomar cuantas decisiones considere oportunas.

Para obtener la nota mínima para aprobar hay que desarrollar los primeros **dos niveles** de la práctica.

Plan de Trabajo

Se seguirá el siguiente método de trabajo:

- En primer lugar se leerá detenidamente el enunciado de esta práctica.
- A continuación hay que diseñar, utilizando un paradigma orientado a objetos, los elementos necesarios para cada nivel de la aplicación explicada en el apartado anterior.
- Debe hacerse uso de los mecanismos de herencia siempre que sea posible.

- Se valorará un buen diseño que favorezca la reutilización de código y facilite su mantenimiento.
- El código estará debidamente comentado.
- La clase principal que abre la aplicación deberá llamarse “movilidad.class”.

Diagrama de Clases

