La elaboración de este ejercicio está dirigida a través de una secuencia de **10 preguntas**, agrupadas en **7 secciones**, en las que se solicita realizar determinadas operaciones y tareas de diseño, vistas en la asignatura. **La puntuación** es sobre un **total de 10** (más 1 punto de la sección BP). Pero **no todas** las cuestiones **puntúan igual**, porque las preguntas correspondientes a la elaboración de los Casos de Uso no requieren capacidades tan próximas a los objetivos de la asignatura como la realización de los Diagramas de Colaboración. Puede utilizar la cantidad de papel que necesite, pero conteste a las preguntas de cada sección en hojas diferentes. Por favor, lea TODO el ejercicio, hasta el final.

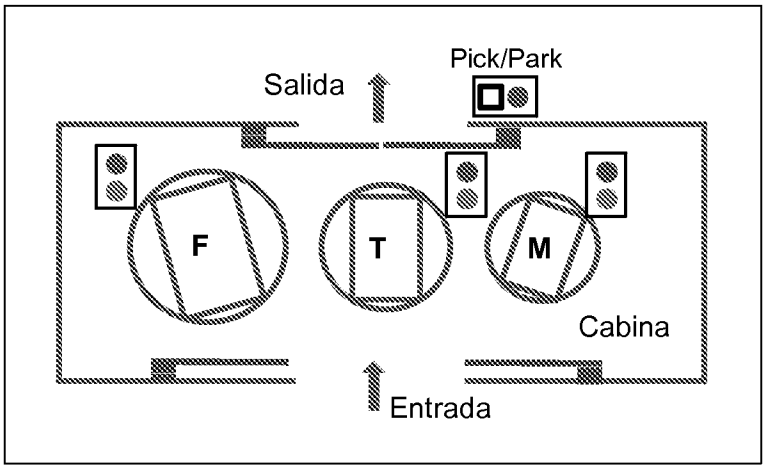
Se ruega al Tribunal que facilite, desde el inicio, 8 hojas adicionales para este ejercicio. Sólo se permite el uso del libro de texto de C. Larman.

Enunciado y planteamiento del caso de estudio.

El dominio del problema es el software que gestiona el aparcamiento de vehículos en un **aparcamiento totalmente robotizado** (AutoPark).

Este sistema tiene dos zonas diferenciadas: una de acceso (de '*embarque*'), a nivel de la vía pública, en la que el usuario deposita y recoge el vehículo; y otra de almacenamiento (subterráneo o elevado), por la que éste se transporta y en la que se guarda con seguridad y sin presencia humana (excepto para labores de mantenimiento).

La utilización del aparcamiento requiere que se registre el vehículo previamente. En el registro, el cliente debe declarar el tipo de vehículo (turismo, furgoneta o moto), la matrícula (o, según sea el modo de identificación, el identificador único del vehículo) y otros datos relacionados con la facturación (datos del cliente, método y forma de pago, etc.). El sistema no permite el acceso a vehículos no registrados. La capacidad de las plazas de cada tipo no puede ser menor que los vehículos registrados de ese tipo (todos los vehículos registrados tienen plaza de aparcamiento garantizada).



La zona de acceso tiene una cabina con una entrada y una salida para vehículos que normalmente están cerradas. En el interior de la cabina hay tres bahías o andenes para situarlos (antes de aparcar), porque se distingue la geometría (y precio) de tres tipos de vehículos: turismos, furgonetas y motos. Cada andén tiene un indicador de disponibilidad (semáforo rojo o verde). La superficie del andén en la que se sitúa el vehículo consiste en una plataforma (pallet), móvil, que es el acceso a la zona de almacenamiento y está controlado por el sistema robótico de aparcamiento (Parker). Los accesos a la cabina (de entrada y salida, con puertas o barreras) disponen de sensores que identifican unívocamente el vehículo (por ejemplo, con cámaras que reconocen su matrícula).

La zona de almacenamiento contiene las plazas en las que se depositan los vehículos mediante un sistema totalmente robótico de elevadores, transporte y posicionado de las bandejas o pallets sobre los que están situados. Esta zona tiene N_t, N_f y N_m plazas de aparcamiento, identificadas con un código y dedicadas a los vehículos de cada tipo. El sistema robótico de aparcamiento (Parker) es **externo** al caso de estudio y su misión consiste en recoger el vehículo desde el andén de embarque, obtener la plaza disponible que corresponda y que, además, optimice las trayectorias sin riesgo de colisión y minimizando el tiempo de maniobra (que debe mantenerse por debajo de 200 segundos). Y viceversa: recoge el vehículo de la plaza asignada y lo sitúa en el andén.

El funcionamiento es como sigue:

El usuario para el vehículo en la entrada (cerrada) de la cabina que, tras identificarle, se abre si la matrícula está registrada y si el andén de embarque para su tipo de vehículo está disponible. En el caso de que el sistema robótico de aparcamiento (Parker) no esté ocupado, el semáforo sólo indicará que hay disponibilidad (verde) en el andén correspondiente a ese tipo de vehículo. El usuario lo sitúa en el andén de embarque disponible (semáforo) según el tipo de vehículo. Tras esto, se cierra el acceso de entrada a la cabina (no se permite el acceso de otro vehículo), se abre el de salida, el usuario apaga el vehículo y lo cierra tras evacuar a todos los ocupantes. Todas las personas deben salir de la cabina y, entonces, se puede dar la orden de aparcar (por ejemplo, pulsando un botón). Antes de iniciar la maniobra de aparcamiento, se cierra el acceso de salida y el robot Parker selecciona una plaza disponible, devuelve el código de plaza en un ticket (código parking, plaza, matrícula, fecha, hora) y realiza el aparcamiento.

La recogida sería a la inversa: el usuario presenta el ticket en el acceso de salida de la cabina, Parker recoge el vehículo y lo sitúa en el andén, en posición de salida. Cuando todo esté en orden (por ejemplo, tras realizar el cobro o registrar la estancia), el sistema abre el acceso de salida para que el usuario retire el vehículo.

Detalles y simplificaciones admitidas:

- El aparcamiento tiene un único punto de acceso (cabina). El número y código de las plazas de aparcamiento es fijo durante el funcionamiento del sistema.
- Las operaciones de acceso, aparcamiento, recogida y salida no pueden interferirse entre sí.
- El funcionamiento del sistema debe garantizar la seguridad de las personas. Supóngase inicialmente que sólo está el conductor en el vehículo. Excepto la apertura y cierre de los accesos, el sistema no debe realizar ningún movimiento en la cabina mientras haya personas en su interior. Todo el sistema (incluido el robótico) debe poder ser bloqueado, instantáneamente, mediante un dispositivo accesible, tanto en el interior como en el exterior de la cabina.
- Aparte de un '*log*' en el que se registre la información de todas las operaciones realizadas y que permita hacer un seguimiento detallado, el sistema debe mantener dos registros permanentes: uno '*vivo*', con las plazas ocupadas, y otro '*histórico*' con la ocupación de un período temporal. En este ejercicio, sólo se consideran las operaciones relacionadas con el aparcamiento, no con el cobro ni la facturación de los servicios.

Otras especificaciones y comportamiento opcionales:

- Flexibilidad: el sistema debe ser fácilmente configurable (el número de plazas de cada tipo, la cancelación de alguna de ellas por obras o mantenimiento, el sistema de identificación del vehículo, etc.). ¿Qué ocurriría si el número de plazas inutilizadas por mantenimiento hiciera que el número de vehículos registrados fuera mayor que el de plazas disponibles? ¿Cómo afectaría al diseño (qué habría que modificar en él) si se abriera otro punto de acceso en el mismo aparcamiento?
  Y si el usuario sitúa el vehículo en el andén equivocado, ¿se ha previsto la recuperación de esta situación en el comportamiento del sistema? ¿Qué parte del software se encargaría de ello?

- Salvaguarda: el sistema Parker no está supervisado '*in situ*' por personas. Por ello, antes de que entre en acción y mueva la plataforma con el vehículo, se debe garantizar (razonablemente) que no quede nadie en la cabina y, entonces, se cierren los accesos. Si se admite que el vehículo puede estar ocupado por varias personas ¿qué cambios habría que hacer en el software para obtener el comportamiento descrito? ¿Y para comprobar, además, que el motor del vehículo está apagado? ¿Cómo se garantiza que no se pierde la información registrada tras un reinicio motivado por la activación del mecanismo de alarma y bloqueo?

NOTA: Estos requisitos no son imprescindibles para este ejercicio. Su consideración en el software desarrollado tendrá una repercusión positiva en la calificación siempre y cuando dicho desarrollo cumpla con el resto de los requisitos enunciados. En este sentido, tiene usted plena autorización para modificar aquellos aspectos del funcionamiento planteado en el enunciado que dificulten o impidan desempeñar la funcionalidad mínima y el comportamiento expuesto. En ese caso, debe indicarlo con claridad en sus respuestas.

Sección 1. Evaluación de los ***Casos de Uso***

1. (0'5 puntos) En relación al software de AutoPark considerado en el caso de estudio, identifique al menos 4 casos de uso primarios y sus actores correspondientes. Represente los resultados en un diagrama de casos de uso de UML.

2. (1 punto) Escriba el caso de uso *<<ProcesarEntradaPark>>* en un formato completo (se recomienda la variante '*en dos columnas*') y estilo esencial. Incluya tanto el escenario principal de éxito (flujo básico de acciones desde que un vehículo se detiene ante la entrada del acceso al aparcamiento, hasta que el usuario se va con el ticket de aparcamiento) como 2 extensiones o flujos alternativos que pudieran ser frecuentes. No escriba un encabezamiento demasiado elaborado del caso de uso (es decir, omita *propósito*, *resumen*...); en su lugar, afronte directamente el transcurso típico de los acontecimientos.

Sección 2. Evaluación del ***Modelado Conceptual***

3. (2 puntos) En relación al caso de uso anterior *<<ProcesarEntradaPark>>*, construya un Modelo de Dominio y represéntelo en notación UML. Represente los objetos conceptuales, las asociaciones y los atributos.

Sección 3. (Diseño) Evaluación de los ***Eventos del Caso de Uso***

4. (1'5 puntos) Circunscrito al caso de uso anterior *<<ProcesarEntradaPark>>*, construya un Diagrama de Secuencia (diagrama de interacción DS) en UML. Represente los actores y los eventos de los componentes del sistema para este caso de uso.

   NOTA: se pide un diagrama de secuencia en el que represente el paso de mensajes entre los actores y los objetos, **NO** del Sistema (DSS). Por tanto, represente las líneas de tiempo de los objetos identificados en el modelo en lugar de la del *sistema global*.

   A partir de este DS, especifique los contratos de **las** operaciones principales: '*Acceso*' y '*Park*' ('*Park*' comienza cuando el usuario da la orden de aparcar –por ejemplo, pulsa un botón—). Estas operaciones son consecutivas y cubren todo el caso de uso considerado; **no se calificará ninguna otra**.

Sección 4. Evaluación de la ***Asignación de Responsabilidades*** y ***Diseño de Colaboraciones***

5. (2 puntos) A partir del contrato de la operación *<<se omite la operación A>>* que haya indicado en el punto 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente.

   ATENCIÓN: lo que hay entre corchetes *<<se omite* >> es un ejemplo, usted lo debe sustituir por el nombre que le haya puesto a la operación principal '*Acceso*', cuyo contrato haya desarrollado en la pregunta 4.

6. (2 puntos) A partir del contrato de la operación *<<se omite la operación B>>* que haya indicado en el punto 4, complete el diagrama de colaboración en UML. Consigne cada mensaje con los patrones GRASP (Experto, Creador, etc.) o cualquier otro que lo justifique. Si añade responsabilidades no explicitadas en el contrato (porque crea que es importante señalarlas), explíquelas brevemente.

   ATENCIÓN: lo que hay entre corchetes *<<se omite* >> es un ejemplo, usted lo debe sustituir por el nombre que le haya puesto a la operación principal '*Park*', cuyo contrato haya desarrollado en la pregunta 4.

Sección 5. Evaluación de los ***Diagramas de Clases*** de diseño

7. (0'5 puntos) Elabore un diagrama de clases para el caso de uso que se está tratando *<<ProcesarEntradaPark>>* (DCD), centrado en la clase cuya responsabilidad es controlar que las acciones se realicen en la secuencia adecuada para que el comportamiento sea el correcto, según se ha descrito en el caso de uso. Represente los nombres de todos sus atributos, asociaciones (con la navegabilidad) y métodos.

Sección 6. Evaluación de la ***Transformación del Diseño en Código***

8. (0'5 puntos) A partir de los anteriores diagramas de clases y colaboraciones, elabore y defina la clase que haya definido, en el desarrollo anterior, como responsable de controlar la correcta secuencia de acciones en el caso de uso *<<ProcesarEntradaPark>>*. Incluya las definiciones de todas las variables que la componen (miembros), pero escriba solamente la definición completa del cuerpo para el método (o métodos) principal o más significativo: *<<se omite el método>>*. Ignore los pequeños detalles de sintaxis -el objetivo es evaluar la capacidad fundamental para transformar el diseño en código-. Utilice la sintaxis de Java.

   ATENCIÓN: lo que hay entre corchetes *<<se omite >>* es un ejemplo, usted lo debe sustituir por el nombre que le haya puesto al método principal que haya elegido.

RECOMENDACIÓN final: si tiene excesivas dificultades porque no ve cómo desarrollar las preguntas 4, 5, 6 y cerrar el Modelo de Dominio d e la pregunta 3; puede ser útil que escriba, en la pregunta 8, una función que secuencie las acciones de las operaciones '*Acceso*' y '*Park*'. Descomponga esas operaciones en sus acciones. A partir de ahí, piense en cómo modificar el código de esa función, qué métodos, variables y clases adicionales se necesitan para incorporar los requisitos y la funcionalidad mínima que le piden. Es probable que las conclusiones que obtenga, junto con la aplicación de los principios GRASP, le permitan afrontar con éxito dichas preguntas, concluir el Modelo de Dominio y elaborar un diagrama de clases (DCD) coherente.

Sección 7. Preguntas opcionales ***BP***. Motivación.

9. (0'5 puntos) Indique qué principios GRASP ha utilizado en el ejercicio y qué responsabilidades ha asignado guiándose por ellos.

10. (0'5 puntos) Indique qué patrones GoF ha utilizado en el ejercicio y para qué le han servido.