

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). **No todas** las cuestiones **puntúan igual**. 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, no se admite ningún otro material, texto impreso o manuscrito.

Enunciado y planteamiento del caso de estudio.

El dominio del problema es un **sistema de gestión para la venta de billetes de avión en una agencia de viajes** (e-Vuela).

El caso de estudio es similar al punto de venta (PdV) del libro de la asignatura y, como ahí, el trabajo se centrará, exclusivamente, en la lógica del negocio y en los servicios técnicos mínimos necesarios para implementar esa lógica en la aplicación. Es decir, aunque exista una capa de presentación, no debe tener en cuenta la interfaz de usuario, la presentación o cómo se capturan los eventos. Céntrese en la lógica de la interacción entre el usuario-agencia y su funcionamiento.

Se pretende que la aplicación tenga flexibilidad para ser utilizada tanto por un *operador* en un terminal de la propia agencia de viajes, por un *cliente* desde un dispositivo móvil o por un *administrador*, para realizar labores de supervisión y mantenimiento. Cada uno de estos perfiles determina las funcionalidades disponibles para un *usuario* concreto. Por tanto, la utilización de esta aplicación se restringe a usuarios registrados, con sus datos de acceso, personales y, en el caso de clientes, de facturación, almacenados en el sistema. En cualquier momento se puede registrar o dar de baja un nuevo *cliente* y cualquier usuario puede editar sus propios datos.

A parte de la interacción debida al mantenimiento de una cuenta (alta, baja, modificación de los datos del usuario, etc.), los servicios de la aplicación se originan cuando un usuario solicita o consulta un viaje en avión, con un número de plazas (pasajeros), entre un origen y un destino, para una fecha y condiciones de uso determinadas (necesidades especiales del viajero, facturación de equipaje, equipajes especiales, etc.). Con esos datos, la aplicación obtiene una *lista de itinerarios* mediante un sistema de planificación (externo) que, a su vez, consulta a las compañías aéreas que operen en cada punto de los itinerarios. Un itinerario consiste en una secuencia de uno o más vuelos entre el origen, las escalas intermedias (si las hay) y el lugar de destino. Cada itinerario contiene (referido a un único pasajero):

- Los datos de cada vuelo, con la siguiente información:
 1. Número de vuelo.
 2. Fecha y hora de salida (local del origen).
 3. Fecha y hora de llegada (local del destino).
 4. Aerolínea.
 5. Lugar de origen del vuelo.
 6. Lugar de destino del vuelo.
 7. Duración del vuelo.
- Duración total del viaje.
- Número de plazas (pasajeros).
- Coste total del viaje por pasajero. En el coste se incluyen los precios de cada vuelo así como las tasas (de aeropuerto, etc.) que correspondan.
- Condiciones de uso: facturación de equipaje incluido en el precio y otra información pertinente al respecto.

El resultado de la consulta (lista de itinerarios), se puede almacenar en su perfil (junto con los parámetros de la consulta), recuperar, actualizar y ordenar según la duración del viaje o su precio total. Con una lista de itinerarios, el usuario también puede seleccionar uno o más itinerarios y **reservarlos**. Al hacer la reserva se '*bloquean*', durante un tiempo (24 h), las plazas solicitadas para los vuelos de ese itinerario (¡Aún no se compran!!). Cada reserva queda reflejada en la lista de itinerarios y, transcurrida su vigencia, desaparece (se '*libera*') al recuperar o actualizar la lista de itinerarios.

Tras hacer una reserva, si está vigente, un usuario puede **comprar** los billetes del viaje de la siguiente manera:

1. Registrando al *cliente*, especialmente sus datos de facturación; o editándolos o confirmándolos si ya estaba registrado.
2. Introduciendo los datos, necesarios para los vuelos, de cada viajero.
3. El *cliente* realiza el pago del importe completo: de todos los vuelos, de todas las tasas, de todos los viajeros...
4. La aplicación emite los billetes de todos los vuelos del itinerario, nominales para cada viajero.

Los detalles y simplificaciones admitidas son:

- Un usuario sólo puede manejar su propia información. Un *operador*, además, puede compartir las listas de itinerarios con cualquier *cliente* o emitir billetes. Un *administrador*, además, puede realizar cualquier operación y manejar la información de cualquier usuario con los otros dos perfiles.
- Se obviará la relación entre las compañías aéreas y la agencia de viajes, que capacita a esta última para '*bloquear*' un determinado número de pasajes (reserva) en un vuelo. No obstante, dichas compañías aéreas son las únicas que manejan cualquier elemento relacionado con los vuelos. Es decir, si bien el planificador (externo) será el encargado de obtener, por parte de las compañías, la información de los vuelos, serán ellas las únicas que emitan los billetes de sus vuelos, previo pago y nominales para cada viajero, cuando lo reclame la agencia.
- Las fechas y horarios son siempre locales. En el cómputo temporal se ignorarán desfases por usos horarios, cambio de año, etc.
- En la reserva y compra, no se contempla la opción de seleccionar asiento.
- Los precios se presentan y manejan en la divisa local de la geolocalización en la que opera el usuario. Aunque se ignorarán los procedimientos de conversión de divisa, la aplicación se debe poder usar en cualquier país.
- En el coste de los vuelos, se considerará el precio por la facturación de única maleta estándar por viajero estándar. Es decir, no se considerarán los sobrecostes por exceso de equipaje o condiciones especiales que queden fuera de la tarifa establecida en el itinerario. Así mismo, la comisión por los servicios (según las políticas de precios y descuentos de la agencia de viajes) estará incluida en el precio final del viaje.
- En la **compra**, el pago se realiza a la agencia de viajes. En la emisión de los billetes, se ignorará el procedimiento por el que la agencia, a su vez, reparte el pago de los distintos vuelos y tasas entre las compañías aéreas incluidas en el itinerario.
- Si la compra se realiza a través de un *operador*, el pago (del cliente a la agencia) podría ser en cualquiera de las modalidades (efectivo, tarjeta, etc.). Sin embargo, si es el *cliente* el que opera a través de un dispositivo móvil, sólo

se considera el pago por medios electrónicos (tarjeta, portales de pago seguro, etc.)

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

1. (0'5 puntos) En relación al software de e-Vuela 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 <<ConsultarViaje>> en un formato completo (se recomienda la variante '*en dos columnas*') y estilo esencial. Incluya tanto el escenario principal de éxito como 2 extensiones o flujos alternativos que pudieran ser frecuentes. Suponga que el escenario de partida para este caso de uso es el de un **cliente** que ya está identificado en el sistema y decide consultar un viaje para un mínimo de 2 viajeros y reservar un itinerario. El flujo básico de acciones comienza con la solicitud de consulta y termina con su almacenamiento (**Consulta** = **lista de itinerarios** + **datos de la consulta**) en su perfil. En este flujo principal de éxito, considere la posibilidad de ordenar los itinerarios (por alguno de los criterios mencionados) previamente al almacenamiento de la consulta en el perfil. 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 <<ConsultarViaje>>, 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 <<ConsultarViaje>>, 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 **dos** de las siguientes operaciones: '*IntroducirDatosViaje*' (datos para la búsqueda de itinerarios), '*OrdenarItinerarios*' (según duración del viaje o coste total), '*ReservarItinerario*' (selecciona un itinerario y '*bloquea*' los vuelos durante 24h) o '*GuardarConsulta*' (almacena la consulta en el perfil, para que pueda ser recuperada más adelante).

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 primera operación principal** que haya elegido en la pregunta anterior 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 segunda operación principal** que haya elegido 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 <<ConsultarViaje>> (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 <<ConsultarViaje>>. 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/es o más significativo/s: <<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 los nombres que le haya puesto a los métodos que haya elegido como **principales** (por ejemplo, 'OrdenarItinerarios' y 'GuardarConsulta').

RECOMENDACIÓN final: si tiene excesivas dificultades porque no ve cómo desarrollar las preguntas 4, 5, 6 y cerrar el Modelo de Dominio de la pregunta 3; puede ser útil que escriba, en la pregunta 8, una función que secuencie las acciones de las operaciones 'IntroducirDatosViaje', 'OrdenarItinerarios', 'ReservarItinerario' y 'GuardarConsulta'. 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 qué mejoras ha conseguido, con su uso, en la elaboración o en el comportamiento del desarrollo de la aplicación.