# Requirements – Student #2

Please, fill in the following form, make sure that you have ticked the requirements that you consider fulfilled, save this document, **and attach it in its original format (.docx)** to every deliverable. Regarding your ID, please keep only four random digits and mask the others using an asterisk. **Please, note that this document must be edited with the desktop version of Word since the web version does not properly support forms.** Attaching this document entails that you are the authors of the work delivered, you have not cheated in any way, and you have read and understood the information delivered regarding the subject, with a special emphasis on the methodological guidelines and how your work is going to be graded. Make sure that your project works well with the latest version of the development framework.

|  |
| --- |
| **Group:** C2.006 |
| **Repository:** https://github.com/rubpergar/Acme-ANS-C2 |
| **Student #2**  **ID Number:** 8022\*\*\*\*\*  **UVUS:**  GCP5811  **Name:**  Pérez Franco, Laura  **Roles:**  Developer, Operator, Analyst |
| **Date:** Sevilla, Julio 4, 2025 |

# MANDATORY Deliverable D01: introduction

## Information requirements

Intentionally blank.

## Functional requirements

1. Modify the anonymous menu so that it shows an option that takes the browser to the home page of your favourite web site. The title must read as follows: “〈id-number〉: 〈surname〉, 〈name〉”, where “〈id-number〉” denotes your DNI, NIE, or passport number, “〈surname〉” denotes your surname/s, and “〈name〉” denotes your name/s.

X

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X

# MANDATORY Deliverable D02: data models

## Information requirements

1. **Customers** are the people who purchase flights. The system must store the following data about them: an **identifier** (unique, pattern "^[A-Z]{2-3}\d{6}$", where the first two or three letters correspond to their initials), a **phone number** (pattern "^\+?\d{6,15}$"), a **physical address** (up to 255 characters), plus a **city** and a **country** (both up to 50 characters). Optionally, customers may have some **earned points** (up to 500k points).

X

1. A **booking** is a reservation made by a **customer** to purchase a **flight**, guaranteeing some seats on a specific itinerary and associating some **passengers**' details with the trip. The system must manage the following information for each **booking**: a **locator code** (unique, pattern "^[A-Z0-9]{6,8}$"), a **purchase moment** (in the past), a **travel** **class** ("ECONOMY", BUSINESS"), and a **price**. Optionally, the system should record the **last nibble** of the credit card used for payment.

X OJO: atributo sin error personalizado.

A black and white image of a cat

AI-generated content may be incorrect.

LP: Se ha añadido el error personalizado con un *message* y el nombre del error definido en los ficheros i18n.

1. A **passenger** is an individual who takes a flight and he or she must be registered in the corresponding booking. The system must store the following data about passengers: a **full name** (shorter than 256 characters), an **email**, a **passport number** (pattern “^[A-Z0-9]{6,9}$”), a **date of birth**, and, optionally, his or her **special needs** (shorter than 51 characters).

X OJO: atributo sin valor personalizado

A black text with black text

AI-generated content may be incorrect.

LP: Se ha añadido una validación nueva para la entidad passenger que incluye la validación del patrón del número de pasaporte y sus errores personalizados, por lo que se valida únicamente con un @ValidString.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

1. Produce assorted sample data to test your application informally. The data must include two **customer** accounts with credentials “**customer1**/**customer1**” and “**customer2**/**customer2**”. Create an additional customer account with credentials “**customer3/ customer3”** that represents a customer with only profile data.

X

## Managerial requirements

1. Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X Mismo problema de antes.

LP: Comment – Student:

Usted indica que es el "mismo problema de antes", pero no me queda claro a qué problema se refiere específicamente, ya que el requisito 2 ha sido validado correctamente.

RC: por error no se indicaron los comentarios correspondientes al requisito D01-02. Al acceder al enlace que proporciona Ud. (<https://github.com/users/rubpergar/projects/5/views/1>), se obtiene lo siguiente:

A screenshot of a computer

AI-generated content may be incorrect.

No es difícil darse cuenta de que esta planificación está incompleta dado que faltan, por ejemplo, las actividades específicas de los estudiantes (asistencia a clase, estudio, reuniones, etc) o las tareas de control de calidad (absolutamente fundamentales en cualquier proyecto). Si revisa Ud. la metodología de trabajo podrá fácilmente identificar todas las tareas que no ha tenido en cuenta.

En la entrega D02 ha corregido Ud. este problema, por lo que el requisito se marcó incorrectamente como inválido haciendo referencia al mismo problema de la entrega anterior.

LP: Para que el dashboard siguiera la metodología establecida en la asignatura, he creado uno nuevo que incluye las tareas correspondientes a esta segunda convocatoria, tanto las relacionadas con la corrección de errores como las nuevas tareas añadidas.  
Además, se han incorporado tareas de testing realizadas por un compañero del grupo para verificar que todo funcionara correctamente, así como tareas de revisión en las que se corrigieron los errores detectados durante el proceso de pruebas.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

# MANDATORY Deliverable D03: implementing features

## Information requirements

Intentionally blank.

## Functional requirements

1. Operations by **customers** on **bookings**:

* List their bookings.
* Show the details of their bookings and the associated passengers, if any.
* Create or update their bookings. Bookings can be updated as long as they have not been published. A booking can be published only when the last credit card nibble has been stored.

X Se realizan las pruebas con customer3.

Dado que no es posible crear ningún Passenger, no es posible tampoco crear ningún booking. (Ver comentarios en el siguiente requisito).

Realizo pruebas con customer1. Registro los dos pasajeros siguientes: el primero NO está publicado y el segundo sí:

A screenshot of a computer

AI-generated content may be incorrect.

Añado el publicado a una reserva de prueba:

A screen shot of a computer

AI-generated content may be incorrect.

Me pregunto cómo podría eliminarlo dado que me he equivocado.

LP: Comment – Student:

Respecto a la primera imagen, he realizado varias pruebas utilizando los customer 1 y 2. En todos los casos, al publicar uno de los pasajeros, el listado se muestra correctamente, incluyendo el campo "isDraft" de manera adecuada. Adjunto una imagen como evidencia de este comportamiento:

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

RC: Comenta Ud. que ha realizado pruebas con el customer 1 y 2, pero tenemos que hacer pruebas con el customer 3. He vuelto a revisar el requisito y he hecho lo siguiente:

\* He entrado en el sistema como customer 3.

\* He accedido al formulario para crear pasajeros y lo he rellenado con los siguientes datos:

A white sheet with blue lines

AI-generated content may be incorrect.

Al pulsar en el botón “create”, el resultado es que vuelvo al mismo formulario y, como le comentaba en el informe original, al mirar el faldón de errores obtengo lo siguiente:

A blue screen with black text

AI-generated content may be incorrect.

Es decir, parece haber un bug en el servicio que se encarga de la creación de pasajeros que debe estar relacionado con algún problema que sólo acontece al trabajar con el customer 3. Es su responsabilidad depurar el problema para encontrar el bug y corregirlo.

Por lo tanto, tal y como se comentaba, no es posible probar este requisito en las mismas condiciones que el resto de estudiantes que no han tenido ningún problema al trabajar con el customer 3 y es la causa principal para no considerarlo satisfecho.

LP: En cuanto a la segunda imagen y a la pregunta planteada, es posible eliminar al pasajero de la reserva. Si pulsa en el pasajero o si accede directamente a la URL correspondiente al show de ese pasajero en la reserva, le aparecerá un botón que elimina al pasajero de la reserva. Adjunto imágenes que muestran el comportamiento:

Antes de eliminar el pasajero:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.





Tras eliminar el pasajero de la reserva:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

RC: Siguiendo la indicación del estudiante, se ha encontrado la opción para eliminar pasajeros de una reserva. Se ha realizado una evaluación normal del requisito encontrando el siguiente problema.

\* Se entra en el sistema como customer 1.

\* Se crea un pasajero con los siguientes datos y se publica:

A white and grey striped background

AI-generated content may be incorrect.

\* Se crea una reserva con los siguientes datos:

A screenshot of a computer

AI-generated content may be incorrect.

\* Se le añade el pasajero creado antes:

A screenshot of a computer

AI-generated content may be incorrect.

\* Al acceder al formulario, se hackea como se indica a continuación:

A red arrow pointing to a white bar

AI-generated content may be incorrect.

Y el resultado es el siguiente:

A screenshot of a computer

AI-generated content may be incorrect.

Es decir, una petición hackeada no ha sido desautorizada y al continuar ha desvelado un bug en el código. Se invita al estudiante que haga otras pruebas con pasajeros como el “100” o cualquier otro perteneciente a otro usuario publicado o no; seguramente encontrará otros problemas.

LP: Para corregir el error mencionado anteriormente, se ha añadido una validación en el método “authorise()”. Esta validación comprueba que la ID del pasajero no haya sido manipulada y coincida con la que realmente corresponde a la relación booking-passenger.  
En caso de que la ID del pasajero difiera, ya sea porque no existe o porque no pertenece a la relación con el booking, se lanza un error, ya que se considera un posible intento de hacking.

En la siguiente imagen se muestra el código relacionado a dicha corrección:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.



Como se ha explicado anteriormente, en caso de que no coincidan o el pasajero sea nulo porque no exista, salta error de autorización.

1. Operations by **customers** on **passengers**:

* List the passengers in their bookings.
* Show the details of their passengers.
* Create a passenger and record the information related to that passenger.
* Update a passenger as long as it has not been published.

X Relleno un formulario con los siguientes datos:

A white paper with black lines

AI-generated content may be incorrect.

Al pulsar en el botón para crear el pasajero, el sistema me devuelve exactamente el mismo formulario.

Reviso el faldón de debug y encuentro lo siguiente:

A blue rectangular object

AI-generated content may be incorrect.

Se realizan algunas pruebas adicionales con customer1.

Hay algo extraño en el siguiente listado de pasajeros.

A screenshot of a computer

AI-generated content may be incorrect.

LP: Comment – Student:

En la segunda imagen, como he mencionado anteriormente (en el Comment – Student del requisito número 8), el listado de los pasajeros tras haber hecho varias pruebas funciona correctamente mostrándose los publicados como “¿Es un borrador?” igual a false.

RC: Se repiten las pruebas, como se indicó en el requisito anterior y el comportamiento es el documento. El estudiante debe seguir el procedimiento documentado para comprobar que el problema existe. Por otra parte, se le aclara que lo “extraño” en el listado no es que ponga “true” o “false” en borrador; es que haya dos personas diferentes con el mismo pasaporte.

LP: Respecto al primer error mencionado, que también se indica en el requisito anterior, el problema estaba en una *query* de un repositorio que se usaba en el método “load()” del create de passenger.  
En esa consulta se intentaba obtener el customer a partir de un passenger para validar la relación con su ID. El error ocurría, por ejemplo, en el caso del customer 3, que no tenía ningún Passenger asociado, y por eso la *query* no encontraba nada y lanzaba un error.

Para solucionarlo, se ha corregido la *query* y ahora se busca directamente en la tabla de Customer en lugar de hacerlo desde Passenger.

La query comentada es la anterior y que daba error y la siguiente es la nueva corregida:

Texto

El contenido generado por IA puede ser incorrecto.

En cuanto al segundo error mencionado, se ha solucionado el problema de que dos pasajeros pudieran tener el mismo número de pasaporte. Para ello, se ha creado un validador en la entidad Passenger que impide que dos pasajeros del mismo customer tengan un número de pasaporte duplicado.  
Este control no se aplica entre pasajeros de distintos customer, ya que una misma persona podría estar registrada como pasajero desde cuentas diferentes.

La siguiente imagen muestra el método “isValid()” de validador de Passenger:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

El atributo sameAsCurrent se utiliza en el caso del update, ya que, al actualizar un pasajero, el número de pasaporte que ya tenía podría activar la validación por duplicado. Por eso, en caso de que el número no haya cambiado respecto al valor anterior, no se lanza el error.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X Mismo problema de antes.

LP: Como he mencionado anteriormente en el requisito 7, se ha creado un nuevo dashboard que incluye todas las tareas relacionadas con la segunda convocatoria, siguiendo la metodología enseñada en la asignatura.

# MANDATORY Deliverable D04: formal testing

## Information requirements

1. Create appropriate indices for your entities, if required.

X Índices redundantes.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A computer screen shot of a computer

AI-generated content may be incorrect.

LP: Para la corrección y verificación de posibles índices redundantes, se ha utilizado dBeaver en el apartado de índices para identificar cuáles ya existían y no eran necesarios.

Para **Customer** se ha eliminado el índice de “user\_account\_id” ya que se trataba de un atributo de navegación por lo que la base de datos lo crea automáticamente.



Para **Booking**, también se eliminaron los dos índices:

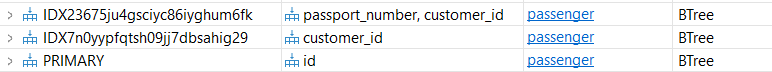
-El índice sobre locatorCode no es necesario porque la anotación @Column(unique = true) genera automáticamente un índice único.



-El índice sobre customer\_id fue eliminado porque, al ser un atributo de navegación, la base de datos lo crea automáticamente.



Por último, en **Passenger** no se eliminó el índice sobre customer\_id porque, al revisar los índices creados automáticamente, este no aparecía a pesar de ser un atributo de navegación. Por ello, resulta necesario mantenerlo para optimizar las consultas que lo utilicen. Además, debido a modificaciones en algunas consultas de los repositorios, fue necesario crear dos índices adicionales. En total, los índices definidos son:



Respecto a la consulta:

**@Query("select p from Passenger p where p.id = :id and p.customer.id = :customerId")**

no he considerado necesaria la creación de un índice adicional, ya que id es la clave primaria y para customer\_id ya existe un índice. Además, se comprobó mediante una consulta EXPLAIN en dBeaver que el rendimiento de la consulta sin el nuevo índice y con él es equivalente, siendo en ambos casos de tipo *const*. Por tanto, la creación de un índice adicional no mejora la eficiencia de esta consulta.

Los resultados para la consulta EXPLAIN sin el índice son los siguientes:



Y los resultados para la consulta EXPLAIN con la creación del índice

@Index(columnList = "id, customer\_id") son los siguientes:



Podemos observar que para ambas consultas el índice utilizado es el de la clave primaria por lo que la creación de este índice personalizado es innecesaria.

Además, respecto a la consulta:

**@Query("select p from Passenger p where p.customer.id = :customerId and p.isDraft = false")**

Tampoco he considerado necesaria la creación del índice ya que probando mediante la consulta EXPLAIN en dBeaver he obtenido los siguientes resultados:

Con el índice @Index(columnList = "customer\_id, isDraft") obtenemos los siguientes resultados:



A pesar de tener un índice personalizado para dicha consulta, no es utilizado.

Sin índice:



Se observa que la tabla es recorrida completamente (full table scan), sin aprovechar ningún índice.

Y, por último, añadiendo @Index(columnList = "isDraft"):



Tampoco se emplea el índice, y se sigue realizando un escaneo completo de la tabla.

Por tanto, se concluye que la creación de índices específicos para esta consulta no es necesaria, ya que no son utilizados durante su ejecución.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

1. Produce a test suite for Requirements #8 and #9.

X Un simple vistazo desvela que el estudiante no ha seguido la metodología recomendada y que las pruebas son incompletas:

A screenshot of a computer

AI-generated content may be incorrect.

LP: Para la nueva realización de las pruebas se han seguido minuciosamente las indicaciones de las diapositivas de la asignatura, creando numerosos casos de prueba que ponen el comportamiento de la aplicación en situaciones límite. A diferencia de las pruebas realizadas anteriormente, en esta ocasión se ha prestado especial atención a generar una mayor cantidad de casos de prueba en los campos de los formularios de *create* y *update*, aumentando así la variabilidad de las pruebas.

## Managerial requirements

1. Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X Mismo problema de antes.

LP: Como he mencionado en los requisitos 7 y 10, he creado un dashboard nuevo para la segunda convocatoria que engloba todas las tareas realizadas durante este período de tiempo.

1. Produce a testing report.

X El informe indica lo siguiente:

A screenshot of a computer

AI-generated content may be incorrect.

El valor crítico de la distribución no es su p-value. El estudiante no aplica correctamente el test dado que compara el valor crítico con el nivel de significación.

LP: Para la corrección del informe, se ha repetido el análisis de las nuevas pruebas utilizando dos ordenadores diferentes. Asimismo, se han recalculado tanto la estadística descriptiva como las pruebas Z. Sin embargo, en esta ocasión, en lugar de comparar el valor crítico bilateral con el nivel de significación, se ha comparado directamente el valor p (P(Z <=z) dos colas) con dicho nivel de significación.

Tabla, Excel

El contenido generado por IA puede ser incorrecto.

# SUPPLEMENTARY I Deliverable D01: introduction

## Information requirements

Intentionally blank.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

Intentionally blank.

# SUPPLEMENTARY I Deliverable D02: data models

## Information requirements

1. The system must handle **customers** **dashboards** with the following **indicators**:

* The last five destinations.
* The money spent in bookings during the last year.
* Their number of bookings grouped by travel class.
* Count, average, minimum, maximum, and standard deviation of the cost of their bookings in the last five years.
* Count, average, minimum, maximum, and standard deviation of the number of passengers in their bookings.

X

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce a UML domain model regarding the information requirements.

X

# SUPPLEMENTARY I Deliverable D03: implementing features

## Information requirements

Intentionally blank.

## Functional requirements

1. Operations by **anonymous principals** on user **accounts**:

* Sign up to the system and become a customer.

1. Operations by **customers** on user **accounts**:

* Update their profiles.

1. Operations by **administrators** on **bookings**:

* List the bookings in the system that are published.
* Show the details of the bookings (including the passengers).

1. Operations by **customer** on **dashboards**:

* Show their customer dashboards.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Provide a link to a video in which you informally test requirement #8 and #9. Videos should not exceed 10 minutes in length and must be stored at the USE's facilities.

# SUPPLEMENTARY I Deliverable D04: formal testing

## Information requirements

Intentionally blank.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Perform five mutations in your code and report on the results.

## Managerial requirements

1. Produce a lint report.

# SUPPLEMENTARY II Deliverable D01: introduction

## Information requirements

Intentionally blank.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce an analysis report.

X

1. Produce a planning and progress report.

X

# SUPPLEMENTARY II Deliverable D02: data models

## Information requirements

1. The system must include a board to recommend something in the city and/or country of a given airport. **Recommendations** can be about experiences, activities, restaurants, accommodation or any other thing that a person may find interesting at the destination. A web service must be used to populate this entity with information about recommendations. Thus, the exact data to store depends on the chosen service, and it is the students' responsibility to define them accordingly. It is also the students’ responsibility to find the appropriate service; no implicit or explicit liabilities shall be covered by the University of Seville or their individual affiliates if the students contract pay-per-use services!  The students are strongly advised to ensure that the service they choose is free of charge.

X

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce an analysis report.

X

1. Produce a planning and progress report.

X

# SUPPLEMENTARY II Deliverable D03: implementing features

## Information requirements

Intentionally blank.

## Functional requirements

1. Operations by **customers** on **recommendations**:

* List recommendations related to final destinations in their bookings.

1. Operations by **administrators** on **recommendations**:

* Populate the database with recommendations somehow related to locations (e.g., related to cities or countries).

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce an analysis report.

1. Produce a planning and progress report.

# SUPPLEMENTARY II Deliverable D04: formal testing

## Information requirements

Intentionally blank.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

1. Produce as a complete test suite as possible for Requirement #29 ensuring that the API is properly mocked.

## Managerial requirements

1. Produce an analysis report.

1. Produce a planning and progress report.