

**Analysis report**

C1.029

<https://github.com/users/pabmejbui/projects/2/views/1?filterQuery=assignee%3AFraperper5+label%3AIndividual>

| **Apellidos, Nombre** | **Email** |
| --- | --- |
| Pérez Lázaro, Francisco | fraperper5@alum.us.es |

**Tabla de contenidos**

[**Resumen ejecutivo 3**](#_heading=h.30j0zll)

[**Tabla de revisiones 4**](#_heading=h.1fob9te)

[**Introducción 5**](#_heading=h.3znysh7)

[**Contenidos 6**](#_heading=h.2et92p0)

[Requisitos de información 6](#_heading=h.tyjcwt)

[Requisitos funcionales 8](#_heading=h.4v41ihq09i7)

[Requisitos no funcionales 9](#_heading=h.6txsm3o9sb0a)

[Requisitos de prueba 9](#_heading=h.jk11hr8qa7jd)

[Requisitos de gestión 9](#_heading=h.7c6boexmukz8)

[**Conclusiones 12**](#_heading=h.3dy6vkm)

[**Bibliografía 13**](#_heading=h.1t3h5sf)

# **Resumen ejecutivo**

Con este análisis buscaremos imprecisiones, incongruencias e indeterminaciones en los requisitos individuales solicitados en la segunda entrega de la asignatura al Student #2. La finalidad será describir los mismos con la mayor precisión posible, buscando optimizar la satisfacción del cliente.

# **Tabla de revisiones**

| **Versión** | **Fecha** | **Descripción** |
| --- | --- | --- |
| v1.0 | 02/03/2025 | Documento creado. |
| v1.1 | 12/03/2025 | Documento completado. |
|  |  |  |

# **Introducción**

En este documento se va a representar el análisis de los requisitos individuales del Student #2 en la segunda entrega.

# **Contenidos**

## **Requisitos de información**

*3)* ***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).*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Creación de la Entidad Customer.  Para todos los atributos obligatorios usamos @Mandatory.  Creación de la validación para phone Number(@ValidPhone)  Añadir el drafMode de tipo **boolean**. | -Cumplir con lo solicitado.  -Respetar la estructura propuesta. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

*4) 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.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Creación de la Entidad Booking.  Para atributos obligatorios usar @Mandatory.  Para atributos opcionales usar @Optional.  Crear Enum para el tipo TravelClass.  Creación de la validación para phone Number(@ValidPhone).  Añadir el drafMode de tipo **boolean**.  Para evitar las relación ManyToMany que tiene Bookings con Passenger creamos una entidad BookingRecord, a su vez esta entidad se relaciona con Flight(ManyToOne). | -Cumplir con lo solicitado.  -Respetar la estructura propuesta. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

*5) 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).*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Creación de la Entidad Passenger.  Para atributos obligatorios usar @Mandatory.  Para atributos opcionales usar @Optional.  Crear un validador @ValidPassport | -Cumplir con lo solicitado.  -Rapidez y sencillez.  -Respetar la estructura propuesta. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Foro de EV de los profesores.

*15) 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.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Crear un AbstractForm para los Customers.  List<String> para guardar los ultimos 5 destinos.  Para los siguientes atributos tipo entero para contadores, y tipo double para costes y medias | -Cumplir con lo solicitado.  -Rapidez y sencillez.  -Respetar la estructura propuesta. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

*26) 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.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Creación de cuenta para OpenTripMap API  Crear una entidad Recommendation  Crear un RecommendationService para definir los metodos API  Crear un RecommendatioController para manejar la API dentro del sistema | -OpenTripMap API es gratuita con limitaciones de solicitudes. | -Se ha intentado implementar pero con fallos de ejecución y debido al tiempo de entrega no se ha completado y llevado a master. Se mantiene en la rama para su futuro desarrollo. |

## **Requisitos funcionales**

Intencionalmente en blanco.

## **Requisitos no funcionales**

Intencionalmente en blanco.

## **Requisitos de prueba**

*6) 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.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Crear un archivo customer.csv con datos de prueba para los atributos de la Entidad Customer, haciendo uso de la IA para proporcionar datos aleatoriamente. | Poder comprobar que los validadores funcionan correctamente y que todo está perfectamente funcionando. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

## **Requisitos de gestión**

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

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Poner en el documento un enlace que lleve al tablero de issues correspondiente al estudiante 2 y al D02. | - Mucha Rapidez para acceder a las tareas específicas del estudiante 2.  - Mejor organización con el uso de los filtros. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

*16) Produce a UML domain model regarding the information requirements.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Un Customer puede tener varios bookings  Varios bookings puede tener varios passengers, de ahí el BookingRecod, que puede tener varios vuelos | - Claridad sobre qué hacer en cada tarea. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

*24) Produce an analysis report.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Hacer un documento analizando todas las tareas por realizar del estudiante 2 en el D02. | - Claridad gracias a las estimaciones de tiempo.  - Velocidad con la organización del D02. | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

*25) Produce a planning and progress report.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Hacer un documento donde se vea la planificación y el progreso seguidos por el estudiante 2 durante el D02. | - Claridad gracias a las estimaciones de tiempo.  - Velocidad con la organización del D02 | -No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 1, ya que es la única alternativa que se ha encontrado.

**Fuente:** Ninguna.

# **Conclusiones**

En este entregable no será necesario hacer una consulta al cliente acerca de imprecisiones, errores, incongruencias o incompletitud en los requisitos individuales del Student #2. Se podrá proceder al desarrollo con normalidad. En todas las tareas se ha encontrado una alternativa clara y sin necesidad de buscar otras alternativas mejores, ya que estas han sido únicas y concisas.

# **Bibliografía**

Intencionalmente en blanco.