

**Analysis report**

C1.029

<https://github.com/pabmejbui/Acme-ANS.git>

| **Apellidos, Nombre** | **Email** |
| --- | --- |
| Mejías Buitrago, Pablo | pabmejbui@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 13](#_heading=h.4v41ihq09i7)

[Requisitos no funcionales 13](#_heading=h.6txsm3o9sb0a)

[Requisitos de prueba 13](#_heading=h.jk11hr8qa7jd)

[Requisitos de gestión 14](#_heading=h.7c6boexmukz8)

[**Conclusiones 16**](#_heading=h.3dy6vkm)

[**Bibliografía 17**](#_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 #1. 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 | 24/02/2025 | Documento creado. |
| v1.1 | 27/02/2025 | -Adición del análisis de las tareas. |
| v1.2 | 09/03/2025 | -Conclusión y bibliografía. |

# Introducción

En este documento se va a representar el análisis de los requisitos individuales del Student #1 en la segunda entrega. No se han encontrado errores potenciales dentro de la descripción de dichos requisitos.

# Contenidos

## Requisitos de información

*3) Airline managers are the people responsible for managing flights. The system must handle the following information about managers: an identifier number (unique, pattern "^[A-Z]{2-3}\d{6}$", where the first two or three letters correspond to their initials), years of experience in the airline, his or her date of birth and an optional link to a picture that must be stored somewhere else.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Hay que crear una entidad para el rol de manager. Para crear el rol se han tomado las siguientes decisiones sobre los atributos:  Clase Manager extiende a la clase abstracta AbstractRole.  Clase validada por el validador ValidManager.  Atributos:   * Identifier: String {Mandatory, Automapped, Column(unique=true)} * experience: Integer {Mandatory, Automapped, ValidNumber(min=0, max=120)} * birthDate: Date {Mandatory, ValidMoment(past=true), Temporal(TemporalType.TIMESTAMP), Automapped} * pictureLink: String {Optional, Automapped, ValidUrl}   Relación con airline, ManyToOne. | - Se siguen los criterios de la asignatura.  - El validador restringe el identifier del manager y devuelve un mensaje de error personalizado indicando que el patrón no es el correcto. | - La relación es redundante es innenecesaria.  - Varios atributos tienen más de 3 restricciones. |
| 2 | Hay que crear una entidad para el rol de manager. Para crear el rol se han tomado las siguientes decisiones sobre los atributos:  Clase Manager extiende a la clase abstracta AbstractRole.  Clase validada por el validador ValidManager.  Atributos:   * Identifier: String {Mandatory, Automapped, Column(unique=true)} * experience: Integer {Mandatory, Automapped, ValidNumber(min=0, max=120)} * birthDate: Date {Mandatory, ValidMoment(past=true), Temporal(TemporalType.TIMESTAMP), Automapped} * pictureLink: String {Optional, Automapped, ValidUrl}   Relación con airline, ManyToOne. | - Se elimina el problema y la redundancia de la relación.  - Las restricciones máximas por atributo es de 3.  - Se sigue las recomendaciones del profesor. | - No se ha encontrado ninguna. |

**Decisión tomada:** Se ha seguido la alternativa 2, ya que es la mejor alternativa que se ha encontrado.

**Fuente:** Diapositivas de teoría de la asignatura.

*4) A flight is a scheduled journey made by airlines to transport passengers between two locations. The system must store the following data about them: a tag that highlights some feature of the flight such as "the fastest", "the cheapest" (up to 50 characters), an indication on whether it requires self-transfer or not, a cost, an optional ŵdescription (up to 255 characters). It also stores information that comes from its legs, namely: a scheduled departure and a scheduled arrival that depends on the first scheduled departure moment of the first leg and the scheduled arrival moment of the last leg, the origin and destination cities that comes from the city of the airports to which first and last leg refers to, and finally, the number of layovers.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Hay que crear una entidad Flight. Para crearla se han tomado las siguientes decisiones sobre los atributos.  La entidad Flight extiende a la entidad abstracta AbstractEntity.  Atributos:   * tag: String {Mandatory, ValidString(min=1, max=50), Column(unique=true), Automapped} * indication: Indication {Mandatory, Valid, Automapped} * cost: Money {Mandatory, ValidMoney, Automapped} * description: String {Optional, ValidString, Automapped} * draftMode: Boolean {Mandatory, Valid, Automapped}   Atributos derivados:   * scheduledDeparture(): Date {Transient} * scheduledArrival(): Date {Transient} * originCity(): Date {Transient} * destinationCity(): Date {Transient} * numberOfLayovers(): int {Transient} * isAvailable(): boolean {Transient}   Relación con manager, ManyToOne. | - Se siguen los criterios de la asignatura. | - Varios atributos tienen más de 3 restricciones.  - draftMode no está definido con el primitivo boolean por lo que da opcion a que no sea solo true o false. |
| 2 | Hay que crear una entidad Flight. Para crearla se han tomado las siguientes decisiones sobre los atributos.  La entidad Flight extiende a la entidad abstracta AbstractEntity.  Atributos:   * tag: String {Mandatory, ValidString(min=1, max=50), Column(unique=true)} * indication: Indication {Mandatory, Valid, Automapped} * cost: Money {Mandatory, ValidMoney, Automapped} * description: String {Optional, ValidString, Automapped} * draftMode: boolean {Mandatory, Automapped}   Atributos derivados los mismos que la alternativa 1.  Relación con manager, ManyToOne. | - Las restricciones máximas por atributo es de 3.  - Se sigue las recomendaciones del profesor.  - Se corrige el problema con draftMode. | - No se ha encontrado ninguna. |

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

**Fuente:** Diapositivas de teoría de la asignatura.

*5) A flight aggregates several legs. A leg represents an individual segment of a flight, typically corresponding to layovers or connections. The system must store the following data for each leg: a unique flight number (composed of the airline's IATA code followed by four digits, unique), a scheduled departure and a scheduled arrival, a duration in hours, a status ("ON TIME", "DELAYED", "CANCELLED", "LANDED"). Additionally, each leg must track the departure and arrival airports, as well as the aircraft that will be deployed for the journey.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Hay que crear una entidad Leg. Para crearla se han tomado las siguientes decisiones sobre los atributos.  La entidad Leg extiende a la entidad abstracta AbstractEntity.  Dos validadores, ValidLeg, ValidFlightNumber  Atributos:  + flightNumber: String {Mandatory, Column(unique=true), Automapped}  + scheduledDeparture: Date {Mandatory, Temporal(TemporalType.TIMESTAMP), ValidMoment}  + scheduledArrival: Date {Mandatory, Temporal(TemporalType.TIMESTAMP), ValidMoment}  + status: LegStatus {Mandatory, Valid, Automapped}  + draftMode: boolean {Mandatory, Automapped}  Atributos derivados:  + getDuration(): Double {Transient}  Relaciones con airport x2, con aircraft y con airline. | - Se cumple con los requisitos de la asignatura.  - Se cumple con las recomendaciones realizadas por el profesor.  - Se corrige el problema con draftMode.  - Las restricciones máximas por atributo es de 3.  - Hay una relación con la entidad airline. | -No se ha encontrado ninguna. |
| 2 | Hay que crear una entidad Leg. Para crearla se han tomado las siguientes decisiones sobre los atributos.  La entidad Leg extiende a la entidad abstracta AbstractEntity.  Dos validadores, ValidLeg, ValidFlightNumber  Atributos:  + flightNumber: String {Mandatory, Column(unique=true), Automapped}  + scheduledDeparture: Date {Mandatory, Temporal(TemporalType.TIMESTAMP), ValidMoment, Automapped}  + scheduledArrival: Date {Mandatory, Temporal(TemporalType.TIMESTAMP), ValidMoment, Automapped}  + status: LegStatus {Mandatory, Valid, Automapped}  + draftMode: Boolean {Mandatory, Valid, Automapped}  Atributos derivados:  + getDuration(): Double {Transient}  Relaciones con airport x2, con aircraft. | - Se siguen los criterios de la asignatura. | - Varios atributos tienen más de 3 restricciones.  - draftMode no está definido con el primitivo boolean por lo que da opción a que no sea solo true o false.  - Faltaría una relación con airline. |

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

**Fuente:** Diapositivas de teoría de la asignatura.

*15) The system must handle* ***manager******dashboards*** *with the following* ***indicators****:*

* *The ranking the manager achieves based on their years of experience. The more years of experience, the higher the position in the ranking.*
* *The number of years to retire, assuming that they retire at 65.*
* *Ratio of on-time and delayed legs.*
* *The most popular and less popular airports within their flights. An airport is popular as long as it has been an origin or destination for many flights.*
* *The number of legs of their flights grouped by their status.*
* *The average, minimum, maximum, and standard deviation of the cost of their flights.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Se crea la entidad ManagerDashboard extendiendo a la clase Abstracta, AbstractForm, usada para crear forms.  + rankingBasedOnExperience; int  + yearsToRetire: int  + ratioOnTimeLegs: double  + ratioDelayedLegs: double  + mostPopularAirports: List<String>  + lessPopularAirports: List<String>  + numberOfLegsGroupedByStatus: Map<LegStatus,Integer>  + avgDeviationOfCost: Money  + minDeviationOfCost: Money  + maxDeviationOfCost: Money  + standardDesviationOfCost: Money | - Se siguen los criterios de la asignatura. | -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 track weather conditions. A web service must be used to populate this entity with information about weather conditions. 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 | Hay que hacer que el sistema maneje condiciones meteorológicas. | Intencionalmente en blanco. | Intencionalmente en blanco. |

**Decisión tomada:** No se ha realizado.

**Fuente:** Ninguna.

## 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 manager accounts with credentials “manager1/manager1” and “manager2/manager2”. Create an additional manager account with credentials “manager3/manager3” that represents a manager with no associated data, except for his or her profile.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Producir datos de ejemplos de las entidades para probar el funcionamiento de la aplicación informalmente. | -Aseguramiento del funcionamiento y de la calidad del código.  -Probar funcionamiento de los límites y restricciones de los atributos. | -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 1 y al D02. | - Mucha Rápidez para acceder a las tareas específicas del estudiante 1.  - 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.*

*and your schedule.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Hacer un modelo de dominio UML, en el que se representen todas las relaciones, entidades y roles pertenecientes a los requisitos del estudiante 1. | -Se tiene una clara idea de cómo son las relaciones entre entidades.  -Facilita el trabajo de código, al plantear ya las restricciones de los atributos. | -No se ha encontrado ninguna. |

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

**Fuente:** Se han tomado como base los ejemplos de las diapositivas de la asignatura.

*27) Produce an analysis report.*

| **Alternativa** | **Descripción** | **Ventajas** | **Desventajas** |
| --- | --- | --- | --- |
| 1 | Hacer un documento analizando todas las tareas por realizar del estudiante 1 en el D02. | - 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.

*28) 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 1 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 #1. 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

Diapositivas de teoría de la asignatura.