Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Analysis Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2023 – 2024

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 15/06/2024 | v1.3.0 |

|  |  |  |
| --- | --- | --- |
| **Grupo: C1.011** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Abouri, Mohamed - Y7156458E | Developer | Persona encargada de desarrollar el código. |
| Cabello Ranea, Rafael -32094496C | Operador | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa. |
| Calderón Rodríguez, Francisco Javier - 78233219F | Project Manager | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo |
| Delgado Pallares, David - 29519510E | Tester | Persona encargada de realizar pruebas sobre el código. |
| Ramírez Lara, Iván- 77852382G | Developer | Persona encargada de desarrollar el código. |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 08/03/2024 | v1.0.0 | Desarrollo de la primera versión. |
| 11/05/2024 | v1.1.0 | Añadido mayor detalles en las descripciones |
| 18/05/2024 | v1.2.0 | Añadido nueva decisión sobre totalTime |
| 07/06/2024 | v1.2.1 | Se ha detallado más la decisión sobre la relación trainingModule con Project tras ser invalidada. |
| 15/06/2024 | v1.3.0 | Cambiada la decisión sobre la relación trainingModule - Project |

**Índice de contenido**

Contenido

[1. Resumen ejecutivo 1](#_Toc160819141)

[2. Introducción 2](#_Toc160819142)

[3. Evaluación 3](#_Toc160819143)

[4. Conclusiones 7](#_Toc160819144)

# Resumen ejecutivo

En este documento se recogen los detalles de análisis de aquellos requisitos en los que ha sido necesario profundizar debido a sus posibles interpretaciones, para cumplir con el requisito 15 del proyecto, correspondiente a los requisitos suplementarios de la segunda entrega.

# Introducción

Tras la realización de todas las tareas de este segundo entregable, se realiza un análisis de los requisitos y decisiones tomadas.

La segunda entrega ha consistido en la creación del diagrama uml del modelo de datos, la implementación de las entidades descritas en los requisitos obligatorios, incluyendo un rol y un form. Algunas de estas tareas han supuesto el tomar decisiones acerca de cómo implementar ciertos elementos, como en el caso de algunos atributos de las entidades.

A continuación se desglosan los detalles en cada requisito en el que ha sido necesario.

# Contenido

“2) A **training module** consists of one or several short-term training activities aimed at extending or updating knowledge and skills related to the topic of a **project**. The system must store the following data about them: a **code** (pattern “[A-Z]{1,3}-[0-9]{3}”, not blank, unique), a **creation moment** (in the past), some **details** describing the training module (not blank, shorter than 101 characters), a **difficulty level** (“Basic”, “Intermediate”, or “Advanced”), an optional **update moment** (in the past, after the creation moment), an **optional link** with further information, and an estimated **total time**.”

Este requisito ha consistido en la implementación de la entidad training module. El requisito indica que cada training module tiene cierta relación con proyectos, que consisten en otra entidad implementada por otro componente del grupo. Tras analizar los requisitos de todo el documento, se han valorado varias alternativas para afrontar este problema:

* Alternativa 1: Crear una relación de trainingModule con project.
  + Ventajas:
    - Se establece la relación mencionada en el requisito.
  + Inconvenientes:
    - Complejidad: Al tener que añadir la relación aumenta la complejidad del desarrollo.
    - Solapamiento: La entidad project es desarrollada por el estudiante 1, por lo que no es positivo la creación de una relación con entidad fuera del dominio de desarrollo del estudiante 3.
* Alternativa 2: No crear la relación de trainingModule con project.
  + Ventajas:
    - Sencillez: Se reduce la complejidad del desarrollo al evitar crear esta relación.
    - Desacoplamiento: Se evitan solapamientos con el desarrollo de los requisitos correspondientes al Student 1, responsable de la entidad project.
    - Compatibilidad: Tras analizar todos los requisitos por implementar, se sabe que esta relación es prescindible. Esto es factible debido a que se conocen todos los requisitos de antemano y que no se van a añadir más, por lo que en este caso sí es factible. Si existiera la posibilidad de que se requieran nuevos requisitos, esta alternativa dejaría de ser factible.
    - Evitar futuros problemas: Al no relacionar la trainingModule con project, se evitan posibles problemas en el futuro sobre el desarrollo de los servicios de creación y actualización de la entidad, además de los posibles conflictos con el desarrollo de la entidad project.
  + Inconvenientes: Realmente no hay inconvenientes porque no se realizan operaciones en requisitos posteriores que impliquen esta relación.

Dado que es una decisión más compleja, se han evaluado las alternativas junto a otros compañeros. A parte de un concenso general sobre que ambas son válidas tras su correcta justificación, el estudiante 1 ve también positivo el evitar las relaciones y no tener que asumir mayor responsabilidad. Otro estudiante también está de acuerdo con adoptar la alternativa 2, pero ha optado por implementar la relación en sus entidades, aunque otros estudiantes sí que optaron por no crear la relación con project.

Finalmente se ha decidido sí implementar esta relación debido a que una vez entregado el proyecto final, el cliente ha especificado claramente que esta relación es necesaria. El cambio se encuentra también reflejado sobre el diagrama UML del Student #3 actualizado.

Además, analizando los requisitos se ha obtenido que son las cuentas “developer” las que realizan operaciones sobre esta entidad, por lo que se ha establecido una relación ManyToOne entre estos.

Se han encontrado también varias alternativas sobre el atributo estimated total time del requisito.

* Alternativa 1: Atributo fijo al que el usuario asigna su valor.
  + Ventajas: Al indicar en el requisito que es un tiempo estimado, el usuario decide cuál será esta estimación del tiempo de su módulo. Su implementación es sencilla, con únicamente un atributo numérico en horas.
  + Inconvenientes: Ninguno
* Alternativa 2: Atributo derivado calculado de la suma de las duraciones de las sesiones.
  + Ventajas: Se obtiene automáticamente el tiempo total del módulo.
  + Inconvenientes: Más complejo de implementar, y no se seguiría exactamente el requisito, que indica un tiempo estimado.

Se ha optado por la alternativa 1, ya que su implementación es más sencilla, cumple el requisito, se acerca más a lo que han entendido los analistas sobre el requisito, y se hace de forma similar en otros sistemas conocidos que ofrecen módulos de aprendizaje. Se le ha asignado un valor máximo a este tiempo de 2000 horas.

El resto de atributos no han requerido de una toma de decisiones compleja.

“4) Each **training module** is made up of **training sessions**.The system must store the following data about them: a **code** (pattern “TS-[A-Z]{1,3}-[0-9]{3}”, not blank, unique), a time **period** (at least one week ahead the training module creation moment, at least one week long), a **location** (not blank, shorter than 76 characters), an **instructor** (not blank, shorter than 76 characters), a mandatory **contact email**, and an **optional link** with further information.”

La complejidad de este requisito se ha encontrado en el atributo period, ya que se encuentran varias formas de implementarlo:

* Alternativa 1: Period de tipo Duration.
  + Ventajas: Se almacena la duración del período.
  + Inconvenientes: Realmente no es necesario almacenar la duración y complejidad en implementación.
* Alternativa 2: Fecha de inicio + número que represente la duración.
  + Ventajas: Se almacena la duración del período. Fácil de implementar
  + Inconvenientes: No es necesario almacenar la duración.
* Alternativa 3: Fecha de inicio + Fecha de fin.
  + Ventajas: Sencillo de implementar, y también sencillo de implementar validación de que debe durar una semana.
  + Inconvenientes: Ninguno

Al no presentar inconvenientes y ser la más sencilla, se ha optado por la alternativa 3.

También se ha establecido una relación de tipo composición entre TrainingModule y TrainingSession, ya que se indica explícitamente que los modulos están compuestos por sesiones.

“4) The system must handle **developer** dashboards with the following data: total number of **training modules** with an **update moment**; total number of **training sessions** with a **link**; average, deviation, minimum, and maximum **time** of the **training modules**”

Este requisito no ha requerido de un análisis en profundidad.

“5) Produce assorted sample data to test your application informally. The data must include two **developer** accounts with credentials “**developer1/developer1**” and “**developer2/developer2**”.

Este requisito no ha requerido de un análisis en profundidad.

# [Conclusiones](#_heading=h.1fob9te)

Los problemas encontrados durante este entregable no han sido de gran magnitud y se han resolvido de forma correcta, analizando las situaciones problemáticas y planteando alternativas para resolverlas.