

Informe de análisis individual Alejandro Gallardo Pelayo

Datos del grupo

Group: C1.03.06

Repository: <https://github.com/alejandromd/acme-L3-D01>

Student #1

ID: 32085666E

UVUS: alepridur
(alepridur@alum.us.es)

Name: Medina
Duran, Alejandro

Roles: manager,
desarrollador

Student #2

ID Number: 54183271D

UVUS: alegalpel
(alegalpel@alum.us.es)

Name: Gallardo Pelayo,
Alejandro

Roles: tester, desarrollador

Student #3

ID Number: 77924720F

UVUS: goncammej
(goncammej@alum.us.es)

Name: Campos
Mejías, Gonzalo

Roles: tester,
desarrollador

Student #4

ID Number: 77934292B

UVUS: alblopnov (
a.lopezbenjumea@gmail.com)

Name: Lopez-Benjumea
Novella, Alberto Miguel

Roles: operador, desarrollador

Student #5

ID Number: 29533138B

UVUS: fravarcas
(fravarcas@alum.us.es)

Name: Vargas Castro,
Francisco Jose

Roles: analista,
desarrollador

Date: Sevilla, a 20 de abril de 2023

Índice

1. Datos del grupo
2. Executive summary
3. Tabla de versiones
4. Introducción
5. Contenido
6. Conclusión

Executive summary

El objetivo de este documento es realizar un análisis de los requisitos que se han implementado durante este tercer entregable, en el que se incluye una copia exacta del propio requisito en sí, así como las decisiones y conclusiones tras haber completado la implementación de este requisito.

Tabla de versiones

| <u>Versión</u> | <u>Descripción</u> | <u>Fecha</u> |
|----------------|--|--------------|
| v1.0 | Creación y primera versión del documento | 20/04/23 |

Introducción

En este documento realizaremos un análisis de los requisitos que hemos desarrollado durante el entregable. En este entregable, se han implementado diferentes funcionalidades y operaciones sobre un rol, y sobre las dos entidades principales asociadas a dicho rol. También, se han implementado las operaciones sobre el formulario para ese mismo rol. Por tanto, se han implementado 4 requisitos funcionales principales.

Contenido

Vamos a realizar el análisis sobre dos de los requisitos funcionales que se han implementado. En concreto, sobre las operaciones realizadas sobre cursos por estudiantes y sobre las operaciones realizadas sobre enrolments por estudiantes.

El requisito de operaciones de estudiantes sobre cursos es el siguiente:

13) Operations by any students on courses: □ List the courses in the system that are published. □ Show the details of the courses in the system, including their lectures and lecturers

Problema:

El cliente nos dice que los estudiantes pueden listar los cursos publicados, pero no se especifica si se refiere a los cursos en los que un estudiante está ya matriculado o a todos los cursos que existen en el sistema. A partir de este problema, se proponen las siguientes alternativas.

Alternativa 1: Listar todos los cursos publicados

Ventajas:

- La implementación sería más rápida y sencilla

Desventajas:

- Para cualquier estudiante, esta funcionalidad proporcionaría la misma información.

Alternativa 2: Listar los cursos publicados en los que el estudiante tiene alguna inscripción

Ventajas:

- La información que proporcionaría esta funcionalidad dependerá de las inscripciones de cada estudiante, por lo que puede variar.

Desventajas:

- La implementación sería más difícil al tener que comprobar los cursos en los que el estudiante tiene una inscripción.

Una vez analizadas las dos alternativas y realizado el correspondiente análisis, he tomado la decisión de escoger la alternativa 1, ya que aparte de ser más fácil de implementar, es exactamente lo que nos dice el cliente en el requisito, que listemos todos los cursos publicados en el sistema. Aunque se puede pensar que listemos los cursos en los que un estudiante esta inscrito, lo que dice la alternativa 2, no tendría sentido ya que el cliente no especifica si debemos buscar los cursos en los que el estudiante tenga inscripciones.

En cuanto al requisito de operaciones de estudiantes sobre enrolments:

[MANDATORY] Operations by students on enrolments:

List their own enrolments.

Show the details of their enrolments.

Register an enrolment.

Update or delete an enrolment as long as it is not finalised.

Finalise an enrolment. This requires providing a valid credit card; the system will only store its holder and its lower nibble.

Problema:

Como gestionar e implementar el tema de proporcionar una tarjeta de crédito válida para poder finalizar un enrolment.

Alternativa 1: Crear atributos String para la tarjeta de crédito en la entidad Enrolment

Ventajas

Dado que un Enrolment está asociado a un estudiante ya, sólo hay que almacenar la información de la tarjeta de crédito. No se crea una tabla nueva, sólo se agrega un atributo.

Desventajas

Quizás al realizar el pago con otra tarjeta no asociada al estudiante pudiera dar problemas.

Alternativa 2: Crear una entidad para almacenar los datos de la tarjeta de crédito.

Ventajas:

Al quedar exclusivamente registrado en una tabla, es más visual de ver en la base de datos.

Desventajas:

Aumenta la complejidad al tener que realizar otra nueva entidad en el sistema.

Una vez analizadas ambas alternativas, he tomado la decisión de escoger la primera alternativa, creando dos atributos String que almacenen los datos necesarios dentro de la propia entidad de enrolment, ya que resulta mucho más sencillo e intuitivo que implementar una nueva entidad para almacenar los datos. En concreto, los datos que se almacenarán son el lower nibble y el holder name de la tarjeta de crédito y siendo obligatorio introducir estos campos para poder finalizar un enrolment.

Conclusión

En resumen, considero que a pesar de las dudas que me han surgido a la hora de interpretar los requisitos he escogido las soluciones que más se acercan a lo que realmente pide el cliente y he podido implementar todas las funcionalidades correctamente.

Bibliografía

intentionally blank