

INFORME DE ANÁLISIS

ESTUDIANTE #5
Óscar Menéndez Márquez
(oscmemar@alum.us.es)

16/02/2024

Grupo : C1.030
Repositorio:
<https://github.com/yesgarfue/Acme-SF-Do1.git>

ÍNDICE

<i>1. Historial de versiones</i>	<i>3</i>
<i>2. Contenidos.....</i>	<i>4</i>
	<i>D01-4</i>
	<i>D02-5</i>
	<i>D03-6-7</i>
	<i>D04-8</i>
<i>3. Conclusiones.....</i>	<i>9</i>
<i>4. Bibliografía</i>	<i>10</i>

HISTORIAL DE VERSIONES

Versión	Fecha	Descripción
1.0	16/02/2024	Entregable D01
1.1	05/03/2024	Entregable D02
1.2	10/04/2024	Entregable D03
1.3	24/04/2024	Entregable D04

CONTENIDOS

DO1 –

- ESTUDIANTE #X – 01:

“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.”

En esta tarea lo que se nos pide es modificar el menú anonimo de manera que al pinchar en nuestro nombre nos redireccione a nuestra pagina web favorita.

Este requisito es claro y directo por lo que no requiere de análisis.

- ESTUDIANTE #5 – 11:

“Produce an análisis report”

Se pide producir un informe de análisis siguiendo las pautas dadas en el documento Annexes.

Este requisito es claro y directo por lo que no requiere de análisis.

- ESTUDIANTE #5 – 12:

“Produce a planning and progress report”

Se pide producir un informe de planificación y progreso siguiendo las pautas dadas en el documento Annexes.

Este requisito es claro y directo por lo que no requiere de análisis.

D02:

Student #5 :

2) Code audits are essential pieces to ensure the quality 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), an execution date (in the past), a type (“Static”, “Dynamic”), a list of proposed corrective actions (not blank, shorter than 101 characters), a mark (computed as the mode of the marks in the corresponding auditing records; ties must be broken arbitrarily if necessary), and an optional link with further information.

-> En esta tarea lo que nos piden es crear la entidad de CodeAudits la cual consta de los atributos que vienen en el mismo enunciado, l aparte con mayor dificultad de esta tarea es encontrar las relaciones entre esta y las demás clases de este entregable. Mas allá de ello este requisito es directo y no requiere de un análisis mas profundo.

Student #5 :

3) The result of each code audit is based on the analysis of their audit records. The system must store the following data about them: a code (pattern “AU-[0-9]{4}-[0-9]{3}”, not blank, unique), the period during which the subject was audited (in the past, at least one hour long), a mark (“A+”, “A”, “B”, “C”, “F”, or “F-”), and an optional link with further information.

->En esta tarea lo que nos piden es la creación de otra entidad, en este caso se trata de AuditRecords, que es el resultado de cada CodeAudit, de nuevo la única complicación que surge en este punto es la relación entre clases, pero la resolvemos sin complicaciones. Mas allá de ello el requisito es directo y no requiere un análisis mas profundo.

Student #5 :

4) The system must handle auditor dashboards with the following data: total number of code audits for “Static” and “Dynamic” types; average, deviation, minimum, and maximum number of audit records in their audits; average, deviation, minimum, and maximum time of the period lengths in their audit records.

->En esta tarea lo que nos piden es crear la clase AuditorDashboard la cual contiene los atributos que nos indican en el enunciado. La tarea en cuestión es directa y no requiere de un análisis en mayor profundidad.

Student #5 :

5)Produce assorted sample data to test your application informally. The data must include two auditor accounts with credentials “auditor1/auditor1” and “auditor2/auditor2”.

->En esta tarea se nos pide que creemos dos usuarios nuevos. La tarea es directa y no requiere de un análisis mas profundo, la resolví sin problemas alguno.

- ESTUDIANTE #5 – 6:

6) Operations by **auditors** on **code audits**:

- List the **code audits** that they have created.
- Show the details of their **code audits**.
- Create, update, or delete their **code audits**. **Code audits** can be updated or deleted as long as they have not been published. For a **code audit** to be published, the mark must be, at least, "C".

Para realizar la tarea que se nos muestra a continuación, deberemos crear el controlador, los servicios, y el repositorio necesario en la entidad CodeAudits.

Posteriormente modificaremos y crearemos las nuevas vistas necesarias para que la aplicación nos lo muestre sin mas problemas

- ESTUDIANTE #5 – 7:

7) Operations by **auditors** on **audit records**:

- List the **audit records** in their **code audits**.
- Show the details of their **audit records**.
- Create and publish an **audit record**.
- Update or delete an **audit record** as long as it is not published.

Para realizar la siguiente tarea se realiza el mismo proceso en el backend, con la creación del controlador, el repositorio y los servicios correspondientes.

Seguimos el mismo proceso para la parte de frontend con la creación de las vistas necesarias

- ESTUDIANTE #5 – 8:

Operations by **auditors** on **auditor** dashboards:

- Show their **auditor** dashboards.

Esta tarea es similar a las dos anteriores lo único que esta vez se centra en todos los datos del rol del Student 5, accediendo a sus Audit Records y sus Code Audit.

- ESTUDIANTE #5 – 17:

Operations by anonymous principals on user accounts:

- Sign up to the system and become an **auditor**.

En esta tarea se nos pide que cualquier otro rol se pueda crear una cuenta de auditor. Se implementan los cambios necesarios y se termina la tarea sin ninguna complicación mas allá.

- **ESTUDIANTE #5 – 18:**

Operations by **auditors** on user accounts:

- Update their profiles.

En esta tarea se nos pide que los auditor puedan actualizar sus perfiles con la información que necesiten. Se realizan los cambios pertinentes y se acaba la tarea.

- **ESTUDIANTE #5 – 19:**

Operations by any principals on **code audits**:

- List the **code audits** in the system that are published.
- Show the details of the **code audits** that they can list (including their **audit records**).

En esta tarea se nos pide que cualquier persona pueda acceder a los Code Audit que estén publicados, además poder ver todos los Audit Records que pertenecen a cada uno de los Code Audit que tenemos la posibilidad de ver.

Para efectuar la realización de la tarea se crean los servicios, controladores y servicios correspondientes a las dos entidades.

- **ESTUDIANTE #5 – 20:**

Produce an analysis report.

Este requisito es claro y directo por lo que no requiere de análisis.

- **ESTUDIANTE #5 – 20:**

Produce a planning and progress report.

Este requisito es claro y directo por lo que no requiere de análisis.

DO4 –

- ESTUDIANTE #5 – 9:

Produce a test suite for Requirements #6 and #7.

Este requisito es claro y directo por lo que no requiere de análisis.

- ESTUDIANTE #5 – 10:

Produce a testing report.

Este requisito es claro y directo por lo que no requiere de análisis.

- ESTUDIANTE #5 – 23:

Produce an analysis report.

Este requisito es claro y directo por lo que no requiere de análisis.

- ESTUDIANTE #5 – 24:

Produce a planning and progress report.

Este requisito es claro y directo por lo que no requiere de análisis.

CONCLUSIONES

D01:

Tras realizar todas las tareas pertenecientes al entregable uno se concluye que todos los requisitos fueron claros y directos, por tanto, el tiempo de ejecución de cada tarea se vio minimizado y la comprensión fue fácil.

D02:

Tras realizar las tareas de este segundo entregable podemos observar como la dificultad de las tareas a realizar aumenta y surgen algunas dudas durante la realización de las mismas, no obstante con el apoyo del profesor en los Follow-Up, y la ayuda mutua de los compañeros podemos realizarlas sin asumir un retraso en cuanto al tiempo estimado de realización de las mismas

D03:

Tras realizar las tareas del tercer entregable observamos que sin duda es el “Sprint” que mas dificultad tiene, ya que las tareas a realizar son las que mas carga de programación tiene. Con la ayuda de nuestro profesor, en los follow-up, hemos conseguido realizarlas todas a tiempo y sin mas complicaciones.

D04:

Tras realizar las tareas de este sprint llegamos a la conclusión de que las pruebas de las funcionalidades que hemos ido implementando a lo largo de la asignatura son de vital importancia ya que gracias a ellas nos hemos llegado a dar cuenta de muchos de los fallos que teníamos en el código. Logrando así corregirlos.

BIBLIOGRAFÍA

No procede.