



# INFORME DE ANÁLISIS D03

Ignacio Blanquero Blanco

**Grupo:** C2.026

- Ignacio Blanquero Blanco ([ignblabla@alum.us.es](mailto:ignblabla@alum.us.es))
- Adrián Cabello Martín ([adrcabmar@alum.us.es](mailto:adrcabmar@alum.us.es))
- María de la Salud Carrera Talaverón ([marcartal1@alum.us.es](mailto:marcartal1@alum.us.es))
- Joaquín González Ganfornina ([joagongan@alum.us.es](mailto:joagongan@alum.us.es))
- Natalia Olmo Villegas ([natolmvil@alum.us.es](mailto:natolmvil@alum.us.es))

**Repositorio:** <https://github.com/maryycarrera/Acme-SF-D04>

**Fecha:** 2 de julio de 2024

## Tabla de contenido

Resumen ejecutivo .....	3
Historial de versiones.....	4
Introducción .....	5
Informe de análisis – Acme-SF-D03.....	6
Requisito obligatorio VI: .....	6
Análisis y decisiones .....	6
Solución .....	7
Problemas encontrados .....	7
Requisito obligatorio VII:.....	9
Análisis y decisiones .....	9
Solución .....	9
Problemas encontrados .....	10
Requisito obligatorio VIII:.....	11
Análisis y decisiones .....	11
Solución .....	11
Problemas encontrados .....	11
Requisito suplementario XVII: .....	13
Análisis y decisiones .....	13
Solución .....	13
Problemas encontrados .....	13
Requisito suplementario XVIII:.....	14
Análisis y decisiones .....	14
Solución .....	14
Problemas encontrados .....	14
Requisito suplementario XIX:.....	15
Análisis y decisiones .....	15
Solución .....	15
Problemas encontrados .....	15
Conclusiones .....	16
Bibliografía .....	17

## Resumen ejecutivo

El propósito de este informe es llevar a cabo un análisis exhaustivo de todos los requisitos obligatorios, así como de algunos opcionales, especificados en la tercera entrega del proyecto Acme-SF-D03 de la asignatura Diseño y Pruebas II. Este proyecto tiene como objetivo mejorar las competencias en el desarrollo web, enfocándose en la implementación de un sistema de información de tamaño moderado. El informe detalla las decisiones adoptadas para la implementación de estos requisitos.

## Historial de versiones

Versión	Fecha	Descripción
v1.0	04/04/2024	Estructura inicial Informe de Análisis D03.
v1.1	05/04/2024	Actualización Informe de Análisis D03.
V1.2	07/04/2024	Actualización Informe de Análisis D03.
V2.0	26/06/2024	Realización de mejoras en el análisis correspondiente a las tareas obligatorias VI, VII y VIII para la segunda convocatoria.
v2.1	27/06/2024	Realización de mejoras en el análisis correspondiente a las tareas suplementarias XVII, XVIII y XIX para la segunda convocatoria.
v2.2	01/07/2024	Redacción de conclusiones.
v2.3	02/07/2024	Finalización y revisión del documento para la segunda convocatoria.

## Introducción

El presente documento, titulado “Informe de Análisis D03”, constituye un análisis exhaustivo de los requisitos individuales asignados en el proyecto del grupo C2.026 al estudiante Ignacio Blanquero Blanco, en el marco del proyecto Acme-SF-D03. Este proyecto forma parte de la asignatura de Diseño y Pruebas II, de naturaleza educativa, y está orientado a que los estudiantes desarrollemos las competencias en el ámbito del desarrollo web.

El propósito principal de este informe es examinar y resolver los siguientes requisitos obligatorios y opcionales del proyecto. En primer lugar, se especificarán las operaciones que los desarrolladores pueden realizar sobre los módulos de entrenamiento, las cuales incluyen listar los módulos creados, mostrar sus detalles, así como crear, publicar, actualizar o eliminar módulos, siempre que no hayan sido publicados. En cuanto a las sesiones de entrenamiento, los desarrolladores tendrán la capacidad de listar las sesiones dentro de sus módulos, mostrar sus detalles, crear y publicar, así como actualizar o eliminar aquellas que no hayan sido publicadas. Adicionalmente, se describirán las operaciones relacionadas con los tableros de control de los desarrolladores, permitiéndoles visualizar sus propias estadísticas. Asimismo, se incluirán las acciones que los usuarios anónimos pueden realizar sobre las cuentas de usuario, como registrarse en el sistema y convertirse en desarrolladores, y las acciones que estos pueden efectuar sobre sus perfiles de usuario, como actualizarlos. Finalmente, se explicarán las operaciones que cualquier usuario puede llevar a cabo sobre los módulos de entrenamiento, las cuales incluyen listar los módulos ya publicados y mostrar sus detalles, excluyendo las sesiones de entrenamiento. Este informe detalla los pasos necesarios para identificar, implementar y validar estas modificaciones, los cuales se abordarán en las secciones siguientes.

El contenido del documento se estructura de la siguiente manera: **Resumen ejecutivo**, el cual proporciona una visión general del proyecto y sus objetivos; **Historial de versiones**, que incluye un registro detallado de los cambios y versiones del documento; **Introducción**, que ofrece una descripción general del contenido y la estructura del documento; **Informe de Análisis – Acme SF-D03**, que detalla el análisis realizado para cumplir con los requisitos obligatorios, así como sus soluciones y los problemas encontrados; **Conclusiones**, que presenta reflexiones finales sobre el entregable y los aprendizajes obtenidos; y **Bibliografía**, que enumera las referencias utilizadas para el desarrollo del proyecto.

## Informe de análisis – Acme-SF-D03

### Requisito obligatorio VI:

Operations by **developers** on **training modules**:

- List the **training modules** that they have created.
- Show the details of their **training modules**.
- Create, update, or delete their **training modules**. **Training modules** can be updated or deleted as long as they have not been published. For a **training module** to be published, it must have at least one training session.

### Análisis y decisiones

Para implementar todos los servicios de dicho requisito se ha seguido la metodología discutida en clase. En consecuencia, se ha decidido crear un servicio para cada funcionalidad solicitada: listar, mostrar, crear, editar, publicar y eliminar módulos de entrenamiento; siendo un requisito común para todas estas funcionalidades que los usuarios estén registrados con el rol de desarrollador.

Al **listar** los módulos de entrenamiento, se ha considerado que no es necesario mostrar todos los atributos por razones de diseño. Por lo tanto, se ha decidido mostrar únicamente los atributos “code”, “details”, “DifficultyLevel”, “estimatedTotalTime”, “draftMode” y “project”.

Al **mostrar** los módulos de entrenamiento, se exhibe la totalidad de la información contenida en dicha entidad, incluyendo los atributos “code”, “creationMoment”, “details”, “DifficultyLevel”, “updateMoment”, “link”, “estimatedTotalTime” y “project”.

En relación con el **servicio de creación** de módulos de entrenamiento, en esta clase, dentro del método validate(), se ha verificado que el atributo “code” no estuviera duplicado, con el objetivo de asegurar que cada código identificativo sea único para cada módulo. Además, se ha comprobado que la fecha de actualización sea posterior a la fecha de creación del módulo de entrenamiento. Con el fin de crear una interfaz de usuario más sencilla e intuitiva, se ha decidido implementar un menú desplegable para la selección del proyecto y del atributo “DifficultyLevel”, siendo este último de tipo enum, permitiendo así elegir entre las distintas opciones posibles. Los demás campos del formulario de creación deberán ser completados de manera manual (tecleando) por el usuario.

Para el **servicio de edición** del módulo de entrenamiento, se han aplicado las mismas restricciones en el método validate() que para el servicio de creación. No obstante, se ha añadido que, para que el módulo de entrenamiento pueda ser editado, el atributo “draftMode” debe estar establecido en true; de lo contrario, no podrá ser editado.

Para el **servicio de publicación** del módulo de entrenamiento, además de las restricciones establecidas en los servicios de creación y edición, se han implementado dos nuevas limitaciones. En primer lugar, un módulo de entrenamiento no puede ser publicado si no contiene al menos una sesión de entrenamiento creada. En segundo lugar, un módulo de entrenamiento que ya incluye sesiones de entrenamiento no puede ser publicado a menos que todas sus sesiones estén también publicadas.

En cuanto al **servicio de eliminación** de módulos de entrenamiento, se ha decidido implementar una restricción en el método validate(). Esta restricción establece que un módulo de

entrenamiento puede ser eliminado únicamente si no tiene sesiones de entrenamiento publicadas. Además, en el método `perform()`, se ha definido que, al eliminar un módulo de entrenamiento, se procede automáticamente a eliminar todas las sesiones de entrenamiento que estén asociadas a él.

Para implementar la **interfaz de usuario** del sistema conforme a este requisito, se ha decidido incorporar un menú desplegable titulado “Developer” o “Desarrollador” en el `AppNavBar`, dependiendo del idioma configurado en la aplicación. Al desplegar dicho menú, se muestra una opción denominada “List my training modules” o “Listar mis módulos de entrenamiento”, que redirige al listado de módulos de entrenamiento del usuario autenticado, donde, además de poder visualizar todos estos, también es posible crear nuevos módulos. Asimismo, se han establecido botones para eliminar, actualizar y editar en cada una de las páginas que muestran los detalles del módulo de entrenamiento, siempre y cuando el atributo “`draftMode`” esté establecido como verdadero. En caso contrario, solo será posible mostrar las sesiones de entrenamiento asociadas a dicho módulo, acción que puede realizarse independientemente del valor del atributo “`draftMode`”.

## Solución

Para cumplir con este requisito, se ha implementado toda la funcionalidad relacionada con la lógica y el procesamiento de datos (backend), incluyendo el repositorio, controlador y servicios, en el paquete denominado “`trainingModule`”, al cual se puede acceder mediante el siguiente enlace:

<https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/java/acme/features/developer/trainingModule>

Las modificaciones realizadas para implementar la interfaz de usuario están disponibles en los siguientes enlaces: los cambios realizados en el `AppNavBar` están disponibles en <https://github.com/maryycarrera/Acme-SF-D04/blob/master/src/main/webapp/WEB-INF/views/fragments/menu.jsp>, y el resto de cambios implementados se pueden revisar en <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/webapp/WEB-INF/views/developer/training-module>.

Una vez implementadas estas modificaciones, el requisito puede considerarse cumplido de acuerdo con las especificaciones detalladas en el enunciado.

## Problemas encontrados

Inicialmente, al abordar este requisito, no se consideraron varias restricciones descritas en el mismo, como, por ejemplo, la condición de que un módulo de entrenamiento no puede ser publicado si no contiene al menos una sesión de entrenamiento creada, y que un módulo que incluya sesiones de entrenamiento no puede ser publicado a menos que estas también lo estén. Para abordar esta situación, se incorporaron estas restricciones en el método `validate()` de la clase “`DeveloperTrainingModulePublishService`”.

```
if (object.getUpdateMoment() != null && object.getCreationMoment() != null && !super.getBuffer().getErrors().hasErrors("updateMoment"))
    super.state(MomentHelper.isAfter(object.getUpdateMoment(), object.getCreationMoment()), "updateMoment", "developer.training-module.form.error.update-date-not-valid");
{
    Collection<TrainingSession> trainingSessions;
    int totalTrainingSessions;

    trainingSessions = this.trainingModuleRepository.findAllTrainingSessionsByTrainingModuleId(object.getId());
    totalTrainingSessions = trainingSessions.size();
    super.state(totalTrainingSessions >= 1, "", "developer.training-module.form.error.not-enough-training-sessions");
    boolean published = true;
    for (TrainingSession c : trainingSessions)
        if (c.isDraftMode()) {
            published = false;
            break;
        }
    if (!published)
        super.state(published, "", "developer.training-module.form.error.all-training-sessions-must-be-published");
}
```

*Ilustración 1. Restricción añadida en la clase "DeveloperTrainingModulePublishService"*

Además, se omitió inicialmente la implementación de la restricción relacionada con la eliminación de un módulo de entrenamiento, que establece que esto solo puede ser eliminado si no tiene sesiones de entrenamiento publicadas. Este problema se resolvió al integrar una restricción en el método `validate()` de la clase “DeveloperTrainingModuleDeleteService”.

```
boolean canBeDeleted;

trainingSessions = this.trainingModuleRepository.findAllTrainingSessionsByTrainingModuleId(object.getId());
canBeDeleted = trainingSessions.stream().allMatch(trainingSession -> trainingSession.isDraftMode() == true);

if (!canBeDeleted && trainingSessions.size() > 0)
    super.state(canBeDeleted, "", "developer.training-module.form.error.can't-delete-published-training-session");
```

*Ilustración 2. Restricción añadida en la clase "DeveloperTrainingModuleDeleteService"*

Es importante señalar que tanto la restricción implementada en la clase “DeveloperTrainingModuleDeleteService” como la restricción que impide la publicación de un módulo de entrenamiento que contenga sesiones de entrenamiento no publicadas, en la clase “DeveloperTrainingModulePublishService”, fueron añadidas después de que el profesor de laboratorio identificara dichos errores, durante la corrección de estos para la segunda convocatoria.



## Requisito obligatorio VII:

Operations by **developers** on **training sessions**:

- List the **training sessions** in their **training modules**.
- Show the details of their **training sessions**.
- Create and publish a **training session**.
- Update or delete a **training session** as long as it is not published.

### Análisis y decisiones

Para implementar todos los servicios de dicho requisito se ha seguido la metodología discutida en clase. En consecuencia, se ha decidido crear un servicio para cada funcionalidad solicitada: listar, mostrar, crear, editar, publicar y eliminar sesiones de entrenamiento; siendo un requisito común para todas estas funcionalidades que los usuarios estén registrados con el rol de desarrollador.

Al **listar** las sesiones de entrenamiento, se ha considerado necesario mostrar todos los atributos por razones de diseño.

Al **mostrar** las sesiones de entrenamiento, se exhibe la totalidad de la información contenida en dicha entidad, incluyendo los atributos “code”, “startPeriodDate”, “finishPeriodDate”, “location”, “instructor”, “contactEmail” y “link”.

En relación con el **servicio de creación** de sesiones de entrenamiento, en esta clase, dentro del método validate(), se ha verificado que el atributo “code” no estuviera duplicado, con el objetivo de asegurar que cada código identificativo sea único para cada sesión. Además, se ha comprobado que la fecha de inicio del periodo sea al menos siete días posterior a la creación del módulo de entrenamiento, y que la fecha de finalización de la sesión sea, al menos, siete días después de su inicio.

Para los **servicios de edición y publicación** de sesiones de entrenamiento, se han aplicado las mismas restricciones en el método validate() que para el servicio de creación. Sin embargo, es importante destacar que una sesión de entrenamiento solo puede ser editada y publicada si su módulo de entrenamiento no está publicado.

En cuanto al **servicio de eliminación**, es importante mencionar que solo es posible eliminar una sesión de entrenamiento si esta no ha sido publicada y su módulo de entrenamiento tampoco ha sido publicado.

Para implementar la **interfaz de usuario** del sistema conforme a este requisito, se ha decidido que, al hacer clic en el botón para listar las sesiones de entrenamiento pertenecientes a un módulo específico, se redirija al listado correspondiente. En esta sección, se muestran todas las sesiones existentes y también se permite la creación de nuevas, siempre y cuando el módulo de entrenamiento no haya sido publicado. Además, se han integrado en cada página que muestra los detalles de las sesiones botones para eliminar, actualizar y editar, condicionados por el estado del atributo “draftMode” y la publicación del módulo de entrenamiento.

### Solución

Para cumplir con este requisito, se ha implementado toda la funcionalidad relacionada con la lógica y el procesamiento de datos (backend), incluyendo el repositorio, controlador y servicios, en el paquete denominado “trainingModule”, al cual se puede acceder mediante el siguiente

<https://github.com/maryycarrera/Acme-SF-D04>

enlace: <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/java/acme/features/developer/trainingSession>

Las modificaciones realizadas para implementar la interfaz de usuario están disponibles en el siguiente enlace: <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/webapp/WEB-INF/views/developer/training-session>

Una vez implementadas estas modificaciones, el requisito puede considerarse cumplido de acuerdo con las especificaciones detalladas en el enunciado.

## Problemas encontrados

Inicialmente, al abordar este requisito, no se consideraron ciertos detalles al implementar las restricciones relacionadas con la duración del periodo y la fecha de comienzo de este. En ambos casos, no se tuvo en cuenta la posibilidad de que las fechas fueran nulas, es decir, si se intentaba crear, publicar o actualizar una sesión de entrenamiento con una fecha de finalización específica, pero con un atributo de fecha de inicio nulo, la aplicación generaba un error 500. Esto ocurría porque, al intentar acceder o utilizar un atributo de fecha que es nulo, la aplicación intentaba realizar operaciones sobre dichos valores nulos, lo que podía resultar en un error de este tipo.

Este problema fue identificado por el profesor de laboratorio y corregido para la entrega de la segunda convocatoria, mediante la inserción del código que se muestra a continuación:

```
if (!super.getBuffer().getErrors().hasErrors("finishPeriodDate"))
    super.state(object.getStartPeriodDate() != null && object.getFinishPeriodDate() != null && MomentHelper.isAfter(object.getFinishPeriodDate(), object.getStartPeriodDate())
        && MomentHelper.isLongEnough(object.getStartPeriodDate(), object.getFinishPeriodDate(), 7, ChronoUnit.DAYS), "finishPeriodDate",
        "developer.training-session.form.error.too-close");

if (!super.getBuffer().getErrors().hasErrors("startPeriodDate"))
    super.state(object.getStartPeriodDate() != null && MomentHelper.isAfter(object.getStartPeriodDate(), object.getTrainingModule().getCreationMoment())
        && MomentHelper.isLongEnough(object.getTrainingModule().getCreationMoment(), object.getStartPeriodDate(), 7, ChronoUnit.DAYS), "startPeriodDate",
        "developer.training-session.form.error.start-too-early");
```

*Ilustración 3. Restricción añadida a los servicios de creación, actualización y publicación de sesiones de entrenamiento*

## Requisito obligatorio VIII:

Operations by **developers** on **developer** dashboards:

- Show their **developer** dashboards.

### Análisis y decisiones

Para cumplir con el requisito establecido, se ha desarrollado una arquitectura que incluye un controlador denominado "DeveloperDashboardController", un repositorio llamado "DeveloperDashboardRepository" para gestionar consultas a la base de datos, y un servicio único denominado "DeveloperDashboardShowService" dedicado a la presentación del tablero de control para desarrolladores. Este servicio se encarga de actualizar los atributos del tablero en función de la existencia de módulos y sesiones de entrenamiento asociadas al desarrollador. En caso de que existan módulos de entrenamiento, se actualizan estadísticas como el número total de módulos con momento de actualización, la desviación estándar, y los tiempos medio, mínimo y máximo de los módulos. Además, si hay sesiones de entrenamiento, se actualiza el número de estas que cuentan con enlace. Por último, en situaciones en las que los desarrolladores no cuentan con datos disponibles, se ha optado por reflejar esta condición indicando valores de 0.

Para implementar la **interfaz de usuario**, se ha decidido incorporar un menú desplegable titulado "Developer" o "Desarrollador" en el AppNavBar, dependiendo del idioma configurado en la aplicación. Al desplegar dicho menú, se presenta la opción "Dashboard" o "Panel de estadísticas", que redirige al tablero donde se pueden visualizar las estadísticas del desarrollador autenticado.

### Solución

Para cumplir con este requisito, se ha implementado toda la funcionalidad relacionada con la lógica y el procesamiento de datos (backend), incluyendo el repositorio, controlador y servicios, en el paquete denominado "dashboard", al cual se puede acceder mediante el siguiente enlace:

<https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/java/acme/features/developer/dashboard>

Las modificaciones realizadas para implementar la interfaz de usuario están disponibles en los siguientes enlaces: los cambios realizados en el AppNavBar están disponibles en <https://github.com/maryycarrera/Acme-SF-D04/blob/master/src/main/webapp/WEB-INF/views/fragments/menu.jsp>, y el resto de cambios implementados se pueden revisar en <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/webapp/WEB-INF/views/developer/developer-dashboard>

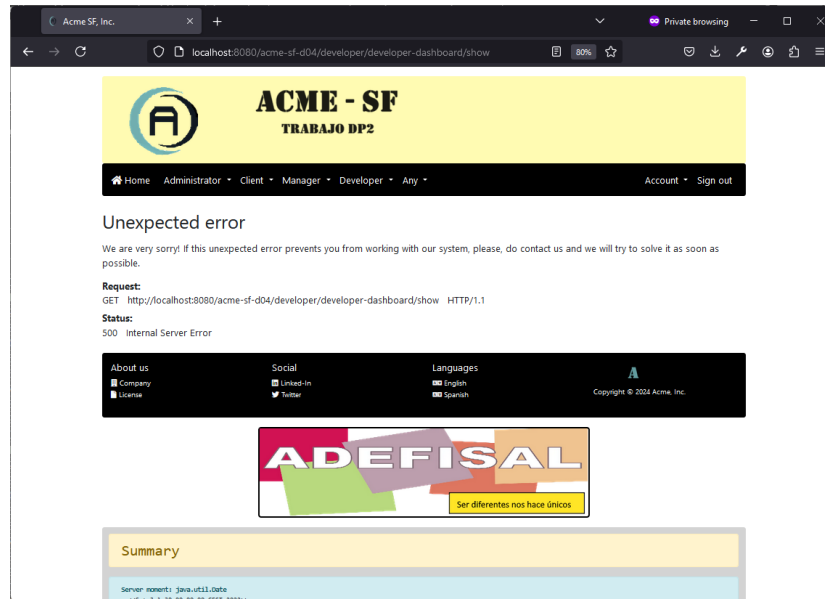
Con estas modificaciones implementadas, el requisito se considera como completado conforme a las especificaciones detalladas en el enunciado.

### Problemas encontrados

Inicialmente, los métodos del repositorio se ejecutaban sobre todos los desarrolladores registrados en la base de datos. Para abordar esta situación y asegurar que cada método del repositorio operara específicamente sobre el desarrollador autenticado, se decidió filtrar los métodos para que solo afectaran al desarrollador que había iniciado sesión.

<https://github.com/maryycarrera/Acme-SF-D04>

Por otro lado, el profesor de laboratorio identificó un error 500 a la hora de mostrar el tablero, como se puede ver a continuación:



*Ilustración 4. Error 500 en el tablero de desarrolladores*

Dicho error fue corregido para la entrega de la segunda convocatoria.

## Requisito suplementario XVII:

Operations by anonymous principals on user accounts:

- Sign up to the system and become a **developer**.

### Análisis y decisiones

Para cumplir con este requisito opcional, y dado que la opción de convertirse en desarrollador se debe mostrar siempre y cuando se haya iniciado sesión con un rol distinto a este, se ha determinado la creación de un paquete denominado “developer” dentro del ya existente “authenticated”, el cual contendrá un controlador (“AuthenticatedDeveloperController”), un repositorio (“AuthenticatedDeveloperRepository”), y un servicio (“AuthenticatedDeveloperCreateService”) diseñados para abordar los requerimientos específicos. Esta estructura se basa en el modelo “Consumer” presentado en el proyecto inicial “Hello World”.

Para implementar la **interfaz de usuario**, se ha decidido incorporar un menú desplegable en el AppNavBar titulado “Account” o “Cuenta”, dependiendo del idioma configurado en la aplicación en el momento de su uso. Al desplegar dicho menú, y siempre que el usuario no posea el rol de desarrollador, ha de aparecer una opción para convertirse en desarrollador. Al hacer clic, se redirige a un formulario en el que es necesario completar ciertos datos, permitiendo al usuario adquirir el rol de desarrollador.

### Solución

Para cumplir con este requisito, se ha implementado toda la funcionalidad relacionada con la lógica y el procesamiento de datos (backend), incluyendo el repositorio, controlador y servicios, en el paquete denominado “developer”, al cual se puede acceder mediante el siguiente enlace:

<https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/java/acme/features/authenticated/developer>

Las modificaciones realizadas para implementar la interfaz de usuario están disponibles en los siguientes enlaces: los cambios realizados en el AppNavBar están disponibles en <https://github.com/maryycarrera/Acme-SF-D04/blob/master/src/main/webapp/WEB-INF/views/fragments/menu.jsp>, y el resto de cambios implementados se pueden revisar en <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/webapp/WEB-INF/views/authenticated/developer>

Con estas modificaciones implementadas, el requisito se considera como completado conforme a las especificaciones detalladas en el enunciado.

### Problemas encontrados

Intencionalmente en blanco.

## Requisito suplementario XVIII:

Operations by **developers** on user accounts:

- Update their profiles.

### Análisis y decisiones

Para cumplir con este requisito opcional, hay que tener en cuenta que dicha opción solo debe ser mostrada por el sistema cuando se esté autenticado con el rol de desarrollador. Para implementarlo, tan sólo ha habido que añadir un nuevo servicio (“AuthenticatedDeveloperUpdateService”) al paquete mencionado en el requisito anterior (“developer”) y hacer algunos cambios en algunos ficheros relacionados con la interfaz de usuario como son menu.jsp y sus archivos de internalización, entre otros.

Para implementar la **interfaz de usuario**, se ha decidido incorporar un menú desplegable en el AppNavBar titulado “Account” o “Cuenta”, en función del idioma configurado en la aplicación en el momento de su uso. Al desplegar dicho menú, y siempre que el usuario posea el rol de desarrollador, ha de aparecer una opción llamada “Developer Data” o “Datos de desarrollador” para actualizar el perfil. Al hacer clic, el desarrollador puede editar la información deseada, actualizando así su perfil.

### Solución

Para cumplir con este requisito, se ha implementado toda la funcionalidad relacionada con la lógica y el procesamiento de datos (backend), incluyendo el repositorio, controlador y servicios, en el paquete denominado “developer”, al cual se puede acceder mediante el siguiente enlace: <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/java/acme/features/authenticated/developer>

Las modificaciones efectuadas para implementar la interfaz de usuario están accesibles en los siguientes enlaces: los cambios realizados en el AppNavBar pueden consultarse en <https://github.com/maryycarrera/Acme-SF-D04/blob/master/src/main/webapp/WEB-INF/views/fragments/menu.jsp>, mientras que el resto de las modificaciones implementadas están disponibles para su revisión en <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/webapp/WEB-INF/views/authenticated/developer>

Con estas modificaciones implementadas, el requisito se considera como completado conforme a las especificaciones detalladas en el enunciado.

### Problemas encontrados

Intencionalmente en blanco.

## Requisito suplementario XIX:

Operations by any principals on **training modules**:

- List the **training modules** in the system that are published.
- Show the details of the **training modules** that they can list (excluding their **training sessions**).

### Análisis y decisiones

Para implementar todos los servicios de dicho requisito se ha seguido la metodología discutida en clase. En consecuencia, se ha decidido crear un servicio para cada funcionalidad solicitada: listar y mostrar los módulos de entrenamiento que están publicados. Cabe destacar que para realizar estas acciones no es necesario estar logueado con el rol de desarrollador, ya que son operaciones que cualquier usuario del sistema puede ejecutar.

Al **listar** los módulos de entrenamiento, se ha considerado que no es necesario mostrar todos los atributos por razones de diseño. Por lo tanto, se ha decidido mostrar únicamente los atributos “code”, “details”, “DifficultyLevel”, “estimatedTotalTime” y “project”.

Al **mostrar** los módulos de entrenamiento, se muestra la totalidad de la información contenida en dicha entidad, incluyendo los atributos “code”, “creationMoment”, “details”, “DifficultyLevel”, “updateMoment”, “link”, “estimatedTotalTime” y “project”.

Para implementar la **interfaz de usuario** del sistema, se ha decidido incorporar un menú desplegable titulado “Any” o “Cualquier usuario” en el AppNavBar, según el idioma configurado en la aplicación. Al desplegar dicho menú, se muestra una opción denominada “List all published training modules” o “Listar todos los módulos de entrenamiento publicados”, que redirige al listado de módulos de entrenamiento que han sido publicados. Además, es posible visualizar los detalles de estos, excluyendo la información relacionada con sus sesiones de entrenamiento, al hacer clic en el módulo que se desee consultar.

### Solución

Para abordar este requisito, se ha implementado toda la funcionalidad relacionada con la lógica y el procesamiento de datos (backend), incluyendo el repositorio, controlador y servicios, en el paquete denominado “trainingModule”, al cual se puede acceder mediante el siguiente enlace:

<https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/java/acme/features/any/trainingModule>

Las modificaciones efectuadas para implementar la interfaz de usuario están accesibles en los siguientes enlaces: los cambios realizados en el AppNavBar pueden consultarse en <https://github.com/maryycarrera/Acme-SF-D04/blob/master/src/main/webapp/WEB-INF/views/fragments/menu.jsp>, mientras que el resto de las modificaciones implementadas están disponibles para su revisión en <https://github.com/maryycarrera/Acme-SF-D04/tree/master/src/main/webapp/WEB-INF/views/any/training-module>

Con estas modificaciones implementadas, el requisito se considera como completado conforme a las especificaciones detalladas en el enunciado.

### Problemas encontrados

Intencionalmente en blanco.

## Conclusiones

Este informe presenta un análisis exhaustivo de los requisitos obligatorios y suplementarios del proyecto Acme-SF-D03, desarrollado en el contexto de la asignatura Diseño y Pruebas II. Se han implementado diversas funcionalidades para los módulos y sesiones de entrenamiento, incluyendo la creación, visualización, edición, publicación, listado y eliminación, así como funcionalidades relacionadas con los perfiles de usuario, tales como la conversión a desarrollador y la actualización de datos personales. Todo el proceso de desarrollo se ha ejecutado siguiendo las metodologías discutidas en clase, lo que ha permitido cumplir con las especificaciones y garantizar la calidad del sistema.

Durante el desarrollo, se identificaron y corrigieron varios errores para asegurar el cumplimiento de los requisitos. Entre estos, se encontraron problemas relacionados con las restricciones en la duración del periodo y la fecha de inicio de las sesiones de entrenamiento, así como otras restricciones que impedían la publicación de un módulo de entrenamiento con sesiones no publicadas.

En conclusión, el informe demuestra un enfoque meticuloso y efectivo en la implementación de los requisitos, proporcionando una base sólida para futuros desarrollos y mejoras en el sistema. Este documento no solo detalla la implementación técnica de los requisitos, sino que también refleja el aprendizaje y la evolución en la gestión de proyectos de desarrollo web. La experiencia adquirida y las soluciones implementadas contribuyen significativamente al objetivo educativo de la asignatura.



## Bibliografía

- Documentación de la asignatura (08 Annexes).
- Diapositivas teóricas: S01 – Display features; S02 – Edition features; S03 – Advanced topics.
- Foro de la asignatura.