

DP II - 2022-2023

ANALYSIS REPORT



<https://github.com/pedlopruz/C1.04.12>

Miembros:

- Carolina Carrasco Díaz
- Pedro López Ruz
- Ángel Lorenzo Casas
- Manuel Navarro Sicre
- Manuel Ortiz Blanco

Tutor: RAFAEL CORCHUELO GIL

GRUPO - C1.04.12

1 ÍNDICE

2 Resumen Ejecutivo	2
3 Tabla de Versionado	2
4 Introducción	2
5 Contenido	3
6 Conclusión	13
7 Bibliografía	13

2 RESUMEN EJECUTIVO

A continuación la finalidad del presente documento será la de tener documentado por escrito cualquier cambio realizado en el proyecto en curso. Reflejándose en todo momento la aprobación o desestimación del cliente ante los cambios propuestos para realizar en dicho proyecto. Se proporcionará por tanto una breve descripción del problema u mejora, así como distintas soluciones a dicho cambio y la decisión final del cliente a dichas propuestas.

3 TABLA DE VERSIONADO

Versión	Fecha	Descripción
1.0	20/03/2023	Primera versión del documento
1.1	21/04/2023	Añadidos los contextos, problemas, alternativas y aprobaciones del cliente según cada observación o problemas encontrados.

4 INTRODUCCIÓN

La intención de este documento es la de recopilar y documentar el análisis de todos los problemas encontrados, los cuáles que van surgiendo a lo largo del desarrollo del proyecto) así como las propuestas de cambios proporcionadas al cliente en espera de su aprobación de cambio u correcciones necesarias, siempre y en todo momento de manera consensuada con el cliente.

5 CONTENIDO

A continuación se pone en contexto todas dudas y problemas encontrados a lo largo del siguiente entregable, con las propuestas sugeridas y la respuesta y/o aceptación del cliente en cada una de ella aprobando o desestimando dichos cambios propuestos.

#1 - Contexto: *requirements #24 and #25 of the Group Requirements document, in relation with the offers that should be shown to the different users that can access them in terms of start and end dates.*

The requirements state the following:

- *D03-G-24: Operations by authenticated principals on offers:*
 - *List the offers in the system and show their details.*
- *D03-G-25: Operations by administrators on offers:*
 - *List the offers in the system and show their details.*
 - *Post, update or delete an offer.*

Problema: Which offers should be shown to authenticated principals, and which to administrators?

Alternativas propuestas:

- **Alternativa 1:** We show every offer to both roles, without any regard for start and end dates of the offer
 - **Pros:**
 - The administrators will be able to access every offer, and be able to manage them.
 - Simpler to implement.
 - **Cons:**
 - Other authenticated principals will have access to offers that have already ended, and thus are of no interest to them
- **Alternativa 2:** We only show only active offers (that is, offers in which the current date is within the date range defined in the offer) to all users
 - **Pros:**
 - Will only show relevant offers
 - **Cons:**
 - The administrator will not be able to manage expired offers
 - Will require a more complex implementation

- Alternativa 3: Show all offers to administrators and only active offers to other authenticated principals.
 - **Pros:**
 - The administrators will be able to access every offer, and be able to manage them.
 - Other authenticated principals will only be shown relevant offers
 - **Cons:**
 - Will require a complex implementation

Despite the complexity of the **third option**, it provides the most adequate implementation, giving both the administrator and other authenticated principals the most relevant offers.

Respuesta Cliente:

"There's something incomplete regarding this requirement. Authenticated users in general should only see the offers that are active, that is: offers whose deadline has not elapsed (Recall that you must use class MomentHelper to implement time-aware operations); the administrators should be able to see all of the offers. The implementation is not more complex. It only requires to add a clause to a query to select offers whose availability period is not finished."

Conclusión: Cambio validado.

ENLACE FORO:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id= 63009 1 &nav=discussion board&conf id= 303964 1&forum id= 206215 1&message id= 362696

#2 - Contexto: requirement #28 of the Group Requirements document:

The requirement state the following:

- **D03-G-28: [Mandatory] Operations by administrators on banners:**
 - List the banners and show their details.
 - Create, update, or delete a banner.

[Mandatory] A banner allows administrators to advertise products, services, or organisations. The system must store the following data about them: an instantiation/update moment (in the past), a display period (must start at any moment after the instantiation/update moment and must last for at least one week), a link to a picture that must be stored somewhere else, a slogan (not blank, shorter than 76 characters), and a link to a target web document.

Problema: Siguiendo el enunciado, al realizar la implementación del requisito 28 grupal y observando los datos a almacenar sobre los banners llegamos a una cuestión que creemos es necesaria contactar con el cliente para poder resolverla.

¿Se podrían realizar operaciones sobre banners que están en el periodo de exhibición?

Alternativas propuestas:

- **Alternativa 1:** Se pueden realizar operaciones en los banners siempre que se quiera sin importar el periodo de exhibición.
 - **Pros:**
 - Siempre que el cliente quiera podrá eliminar y actualizar los banners lo que permite una mayor libertad para el cliente
 - **Cons:**
 - Los banners puede que no se muestren en su periodo de exhibición, por lo que no tiene sentido tener este periodo.
- **Alternativa 2:** Se pueden realizar operaciones en los banners solo si la fecha actual es anterior al comienzo del periodo de exhibición.
 - **Pros:**
 - Un banner se mostrará completamente dentro de su periodo de exhibición.
 - **Cons:**
 - Es posible que el cliente quiera realizar operaciones después de que finalice el periodo de exhibición.

- Alternativa 3: Se pueden realizar operaciones en los banners mientras la fecha actual este fuera del periodo de exhibición.

- **Pros:**

- Un banner se mostrará completamente dentro de su periodo de exhibición.
- Se permite realizar operaciones sin restricciones mayores como en la alternativa 2.

- **Cons:**

- Es posible que el cliente prefiera total libertad a la hora de realizar operaciones en los banners

Después de realizar una tarea de análisis, nos decantamos por **la opción número 3** dado que de esta forma el cliente podrá tener cierta libertad a la hora de realizar operaciones sobre banners, además de que los banners se mostraran a lo largo de todo el periodo de exhibición, pero como desconozco que es lo que prefiere el cliente por eso me pongo en contacto con usted por si podría resolverme esta duda.

Respuesta Cliente:

Su pregunta desvela, en efecto una duda razonable en los requisitos que podríamos calificar como "incompletitud". La pregunta tiene sentido y no tiene respuesta en los requisitos, por lo que éstos son incompletos. El cliente confirma que tal y como Ud. indica la opción correcta es la número 3: no debe ser posible modificar los banners mientras están en período de exhibición."

Conclusión: Cambio validado.

ENLACE FORO:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id= 63009_1&nav=discussion_board&conf_id= 303964_1&forum_id= 206215_1&message_id= 365806_1

#3 - Contexto: *requirement #29 of the Group Requirements document:*

The requirement state the following:

- *D03-G-29: [MANDATORY] Moments, money amounts, and Booleans must be internationalised when they are shown. Other kinds of data are not expected to be internationalised. This requirement must be fulfilled in this and every subsequent group or individual deliverable.*

Problema: Siguiendo el enunciado, ¿qué implementación considera usted más adecuada para implementar el requisito de entre las alternativas siguientes?

Alternativas propuestas:

- **Alternativa 1:** Usar los inputs correctos en los formularios y expresar booleanos mediante checkbox:
 - **Pros:**
 - El equipo no debe preocuparse por la internacionalización ya que van implementadas en las etiquetas input-money, input-moment...
 - **Cons:**
 - Todos los campos afectados deben tener el input correcto.
- **Alternativa 2:** Crear el código correspondiente para la implementación del requisito
 - **Pros:**
 - Se puede plantear de una forma diferente a las etiquetas propuestas.
 - **Cons:**
 - Supone un mayor esfuerzo y una pérdida de tiempo para el cliente.

Expuestas ambas alternativas, el equipo considera la primera opción como la más viable ya que el tiempo ahorrado puede dedicarse a otras tareas de desarrollo aportando mayor valor al cliente, sin embargo, nos gustaría saber cuál considera usted como la mejor implementación.

Respuesta Cliente:

“El requisito sobre el que pregunta es un tanto atípico dado que realmente no hay que hacer nada para cumplirlo. El framework Acme ya se encarga automáticamente de i18n los momentos y cantidades de dinero tanto en la entrada de datos como a la hora de mostrarlos. Los Booleanos () pueden gestionarse mediante etiquetas de tipo “acme:input-checkbox”; estas etiquetas muestran los booleanos como un cuadradito en el que Ud. puede hacer click para activarlo (true) o desactivarlo (false); al tratarse de una representación gráfica no requiere de i18n específica alguna dado que tanto en español como en inglés dicha representación tiene el mismo significado.*

Pese a tratarse de un requisito que se cumple trivialmente, requiere de un poco más de análisis. Los Booleanos no son más que un tipo enumerado predefinido; se pueden considerar al mismo nivel que un enumerado definido por el usuario como “ApplicationStatus” en el caso de Acme Jobs o el enumerado que representa el tipo

de curso, clase, actividad, etcétera en el caso de Acme L3 (THEORETICAL, HANDS ON o BALANCED). Por desgracia, el framework no proporciona un buen soporte para la i18n de los valores enumerados, por lo que en sus proyectos no influye para nada en la evaluación si i18n o no esos valores. Por ejemplo: si en un listado muestran Udes. un Booleano, sus valores saldrán siempre como "true" o "false" independientemente de que el locale sea "en" o "es". Ocurre lo mismo cuando muestran la naturaleza de un curso: siempre saldrá "THEORETICAL" o "HANDS-ON" o "BALANCED" con independencia al locale seleccionado.

El problema de la i18n de los valores enumerados NO es un problema específico de Acme Jobs o de Acme L3, por lo que requiere de una solución general a nivel de framework, pero, por desgracia, aún no se ha podido diseñar una solución genérica suficientemente buena. Tampoco se considera éste un issue prioritario de corregir. Por lo tanto, mientras no tengamos una buena solución no es necesario buscar ninguna solución ad hoc en sus proyectos.

Dicho de otra forma: tan sólo hay que marcar el requisito como satisfecho, no hay que hacer nada especial para cumplirlo."

Conclusión: No hay que realizar cambios.

ENLACE FORO:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id= 63009_1&nav=discussion_board&conf_id= 303964_1&forum_id= 206215_1&message_id= 365604_1

#4 - Contexto: requirement #31 of the Group Requirements document:

The requirement state the following:

- **D03-G-31:**

The system must prevent the principals from storing any data that can be considered spam. A piece of text is considered spam if the number of spam terms exceeds a predefined spam threshold. The default list of spam terms includes the following ones: "sex", "viagra", "cialis", "one million", "you've won", "nigeria", and their corresponding translations into the languages considered for internationalisation; the default spam threshold is 10%. Note that the previous default values can be changed at will by the administrators. Realise that a term must be considered spam irrespective of its case and the blanks in between its words; for instance, "one_million" is a spam term that matches "one_million", "ONE_MILLION", "OnE _ _ _ MiLLiOn", or "One _ ↵ _ Million"; it doesn't match "One_Millionaire", "One_or_two_millions", or "One_sex_million", though. The spam detector must be reusable across different projects; that is: it must be implemented as an independent project that must be packaged into a reusable dependency. Do not forget to deliver your spam detector project so that it can also be evaluated or, otherwise, this requirement shall not be considered fulfilled. This requirement must be fulfilled in this and every subsequent group or individual deliverable.

Problema: We have given some initial thought to the requirement and got some alternatives for the implementation but we are uncertain of whether those would work or even if they are correct according to how the work in acme should be done.

Note: alternatives 1 and 2 would have some common parts within the detector project, those are:

- An entity for keeping in the database the two variables needed.
- A group of controller, repository and services show and update for the administrator role.
- The initial data csv with the default values.

So, the question is:

Which of this alternatives would be valid, and from those, which could be the best option?

Alternativas propuestas:

- **Alternativa 1:** We implement it as a custom validation through the use of a javax constraint annotation.
 - **Pros:**
 - Easier to use once it has been developed.
 - Goes in keeping with the idea of not reutilising code, as we are gonna validate a lot of String fields with it.
 - **Cons:**
 - It most likely goes against the acme workflow, as we have been told to implement custom validations in each service.

- As it would require a repository for validating within the constraint validator (getting the configuration), it could create conflicts involving transactions when populating the samples, as it would be used at that point.
- Alternativa 2: We implement it as a helper class with a validate method which takes the configuration values as parameters, retrieved by that service's repository.
 - **Pros:**
 - It is closer to how we are currently working.
 - It would be easier to make the validation class.
 - **Cons:**
 - It would require a lot of work to apply the validator once finished on each attribute.
 - The same pattern of code is going to appear in each service/repository whose entity has at least one String attribute.
- Alternativa 3: Both the previous alternatives assume the extra project can be considered independent even if it requires the framework. This alternative is here just in case it is the other way around and it can't be considered independent. Implement the common parts mentioned before in the Acme project and leave on the independent detector project just the class that does the validation, taking as parameters both the value to validate and the configuration
 - **Pros:**
 - It is closer to how we are currently working.
 - It would be easier to make the validation class.
 - **Cons:**
 - It would require a lot of work to apply the validator once finished on each attribute.
 - The same pattern of code is going to appear in each service/repository whose entity has at least one String attribute.

(Same as Alternativa 2, as the only difference is we are relocating some code.)

Alternative 1 should likely be avoided, alternative 2 seems the better choice and if there was a need to make the side project fully independent, **we would then go for alternative 3.**

Respuesta Cliente:

“[...]

- *Alternative 1 should likely be avoided. It's very good, but I wouldn't endeavour to implement it now.*
- *Alternative 2 seems the better choice and if there was a need to make the side project fully independent. It's not bad. The only problem is that you don't avoid repeating the code to fetch the spam configuration from the database so that you can pass it into the method. But you may easily*

start with it and try to address that problem later. The real pro is that you can quickly implement the anti-spam algorithm and satisfy your customer's requirement and then work on something better from a technical point of view.

- *Then go for alternative 3. That is not totally clear to me. I think it's the same as Alternative 2.*
- [...]

Implement a helper class with a method that takes the text to check, the list of spam terms, and the threshold as parameters, implements the algorithm to detect spam (it's not very good, but no problem with that), and returns a boolean.

Use it to enhance one of your features with an anti-spam filter and test it informally so that you feel confident that there are not any evident mistakes. Extract that class into an individual project (let's call it Acme Anti-Spam, for instance), link it from Acme L3 and check that everything works well.

This solution shall be evaluated positively if it works well and the lecturer doesn't find any mistakes. "

Conclusión: Cambio validado.

ENLACE FORO:

https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id= 63009_1&nav=discussion_board&conf_id= 303964_1&forum_id= 206215_1&message_id= 366171_1

6 CONCLUSIÓN

Actualmente todos los cambios llevados a cabo durante esta entrega se han consensuado en todo momento con nuestro cliente, no habiendo ningún cambio realizado sin la aprobación del mismo. Sírvase además este documento de análisis como prueba de que todos los cambios realizados han sido aprobados y queda documentado en dicho reporte.

7 BIBLIOGRAFÍA

Intencionalmente en blanco.