Informe de análisis sobre tarea 114

Integrantes del grupo C1.026:

* 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))

Fecha: Sevilla, 26 de Marzo 2024

Tabla de contenido

[Resumen del Informe 3](#_Toc164527887)

[Historial de Versiones 4](#_Toc164527888)

[Introducción 5](#_Toc164527889)

[Informe de Análisis – Acme-SF-D02 6](#_Toc164527890)

[Registro de Análisis 1 6](#_Toc164527891)

[Requisito 6](#_Toc164527892)

[Análisis y decisiones 6](#_Toc164527893)

[Validación por el profesor 7](#_Toc164527894)

[Registro de Análisis 2 8](#_Toc164527895)

[Requisito 8](#_Toc164527896)

[Análisis y decisiones 8](#_Toc164527897)

[Validación por el profesor 9](#_Toc164527898)

[Registro de Análisis 3 10](#_Toc164527899)

[Requisito 10](#_Toc164527900)

[Análisis y decisiones 10](#_Toc164527901)

[Validación por el profesor 10](#_Toc164527902)

[Registro de Análisis 4 11](#_Toc164527903)

[Requisito 11](#_Toc164527904)

[Análisis y decisiones 11](#_Toc164527905)

[Validación por el profesor 11](#_Toc164527906)

[Registro de Análisis 5 12](#_Toc164527907)

[Requisito 12](#_Toc164527908)

[Análisis y decisiones 12](#_Toc164527909)

[Validación por el profesor 12](#_Toc164527910)

[Registro de Análisis 6 13](#_Toc164527911)

[Requisito 13](#_Toc164527912)

[Análisis y decisiones 13](#_Toc164527913)

[Validación por el profesor 13](#_Toc164527914)

[Conclusiones 14](#_Toc164527915)

[Bibliografía 15](#_Toc164527916)

# Resumen del Informe

El objetivo de este informe es proporcionar detalles sobre el análisis de todos los tipos de requisitos especificados en esta segunda entrega del proyecto, así como las decisiones tomadas para llevarlos a cabo y enlazarlo, si procediera, a las validaciones y observaciones planteadas por el profesor encargado de la asignatura.

# Historial de Versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Contenidos** | **Fecha** | **Contribuyente** |
| V0.1 | Estructura inicial Analysis Report D03 | 26/04/2024 | Adrian Cabello Martin |

# Introducción

En esta tercera entrega teníamos que implementar tres requisitos funcionales obligatorios y otros tres requisitos funcionales suplementarios. La carga de trabajo ha aumentado notablemente, por tanto, resulta de vital importancia una buena gestión de las tareas a cumplimentar y una buena gestión de estas en el tiempo. A continuación, se describirán con más detalle todos los requisitos implementados individualmente por Adrian Cabello Martin.

# Informe de Análisis – Acme-SF-D02

Antes de comenzar con el análisis cabe destacar que al comienzo de la implementación de todos estos requisitos por parte de Adrian Cabello Martin surgieron varios problemas, debido a que este integrante del grupo todavía no estaba familiarizado con el framework. No obstante, tras la implementación de todos ellos se ha logrado que Adrian Cabello Martin pueda trabajar de forma fluida sin grandes dificultades.

## Registro de Análisis 1

### Requisito

Operaciones de los patrocinadores en patrocinios:

* Listar los patrocinios que ellos han creado.
* Mostrar los detalles de sus patrocinios.
* Crear, actualizar o eliminar sus patrocinios. Los patrocinios pueden ser actualizados o eliminados siempre y cuando no hayan sido publicados. Para que un patrocinio sea publicado, la suma del dinero total de todas sus facturas debe ser igual al dinero del patrocinio.

### Análisis y decisiones

Por una parte, se ha decidido crear un servicio por cada funcionalidad que se pedía implementar tal y como se aconsejó en clase: mostrar, listar, crear, actualizar, borrar y publicar patrocinios.

A la hora de listar los contratos se ha supuesto que no sería necesario mostrar todos los atributos por cuestiones de diseño, por lo que se ha decidido que sólo se mostrarían los atributos "code", "moment", "amount" y “type”. A su vez, no se ha impuesta ninguna restricción adicional para poder acceder a este listado, sólo que esos contratos pertenecieran al patrocinador logueado.

Con respecto al servicio de crear patrocinios, es de gran importancia comentar que se ha añadido a la entidad Sponsorship un atributo “draftMode”, que nos permitirá conocer cuando un patrocinio está en modo borrador o no. Si se encuentra en modo borrador, podrá ser editado. En esta clase, dentro del método validate() se ha comprobado que el presupuesto no sea negativo y que las fechas tienen que ser después de otras con coherencia y que la distancia entre el inicio y el fin tiene que ser de 30 dias, ya que ese nos requería implementar esta regla de negocio en los requisitos de la primera entrega, pero no podíamos hacerlo hasta ahora debido a nuestra falta de conocimiento en ese momento.

A la hora de actualizar un patrocinio se decidió que esta acción se podría llevar a cabo siempre y cuando el contrato no sea nulo, se encuentre en modo borrador y pertenece al cliente logueado. Las validaciones y el resto del código es prácticamente igual que el de crear. Para eliminar un patrocinio lo más destacable sería que se ha especificado en el método perform() que una vez que se elimina un patrocinio también desaparecen sus facturas .

Por último, al publicar un patrocinio se valida la condición de que todos los precios de las facturas tienen que ser igual al precio del patrocinio y en el método perform() se deberá poner el draftMode a false para que no pueda ser editado.

En cuanto a decisiones de diseño del frontend se ha decidido que en los formularios a la hora de elegir el proyecto asociado a un patrocinio se abra un desplegable con todos los proyectos disponibles y que sean públicos. Con respecto a los botones, todos los botones aparecerán mientras el patrocinio esté en modo borrador, menos el de las facturas que siempre será visible.

### Validación por el profesor

El profesor no reviso los requisitos.

## Registro de Análisis 2

### Requisito

Operaciones de los patrocinadores en facturas:

* Listar las facturas de sus patrocinios.
* Mostrar los detalles de sus facturas.
* Crear y publicar facturas.
* Actualizar o borrar facturas siempre y cuando no se haya publicado.

### Análisis y decisiones

Por una parte, se ha decidido crear un servicio por cada funcionalidad que se pedía implementar tal y como se aconsejó en clase: mostrar, listar, crear, actualizar, borrar y publicar facturas. Este último no se pedía en los requisitos, pero se ha decidido añadir ya que más adelante hay un requisito que consiste en listar todos los informes de progreso de un patrocinio publicado. Este listado es accesible a todos los usuarios autenticados, así que estos sólo deberían acceder a informes de progreso que ya han sido publicados y sin posibilidad de modificación.

A la hora de listar los informes de progreso se ha supuesto que no sería necesario mostrar todos los atributos por cuestiones de diseño, por lo que se ha decidido que sólo se mostrarían los atributos "code", "registrationTime" y “dueDate”. A su vez, se ha decidido que se va a mostrar el botón de las facturas asociadas a ese patrocinio.

Con respecto al servicio de crear facturas, es de gran importancia comentar que se ha añadido a la entidad Invoice un atributo “draftMode”, que nos permitirá conocer cuando una factura está en modo borrador o no. Si se encuentra en modo borrador, podrá ser editado. En esta clase, dentro del método validate() se ha comprobado que el code sea único y se ha establecido el modo borrador a verdadero en el método load() y en el método perform()

A la hora de actualizar una factura se decidió que esta acción se podría llevar a cabo siempre y cuando el patrocinio al que está asociado no sea nulo y el mismo no se encuentre en modo borrador, pertenece al patrocinador logueado y que la factura esté en modo borrador. Las validaciones y el resto del código es prácticamente igual que el de crear.

Por último, al publicar una factura, tras hacer la misma comprobación que para actualizar y eliminar para ver si se tiene autorización para realizar esta acción a través del método authorise(), se realizan las validaciones oportunas, que son las mismas que las de los métodos anteriores, y por último en el método perform() se deberá poner el draftMode a false para que no pueda ser editado.

En cuanto a decisiones de diseño del frontend se ha seguido la misma estructura que para los patrocinios y no ha surgido ningún problema.

### Validación por el profesor

EL profesor no lo ha revisado.

## Registro de Análisis 3

### Requisito

Operaciones de los patrocinadores en la pantalla principal:

* Mostrar las pantallas principales de los patrocinadores.

### Análisis y decisiones

Se definieron los métodos correspondientes que se pedían en los requisitos de la entrega anterior en el repositorio y por consiguiente en su correspondiente servicio y controlador, sin encontrar ningún problema. Se decidió que en el repositorio al llamar a estos métodos habría que introducir cómo parámetro el id del patrocinador logueado para poder recoger los datos acerca de ese patrocinador en concreto y no tener en cuenta todos los patrocinadores de la base de datos.

### Validación por el profesor

El profesor no lo ha revisado.

## Registro de Análisis 4

### Requisito

Operaciones por usuarios anónimos en cuentas de usuario:

* Registrarse en el Sistema y convertirse en un patrocinador.

### Análisis y decisiones

Dado que para realizar las acciones que se piden el usuario debe estar autenticado, dentro del paquete authenticated se ha creado otro paquete sponsor y se ha creado un repositorio, un controlador y dos servicios, uno de ellos para crear el patrocinador y otro para actualizarlo, que se pide en el siguiente requisito. Para ello se ha seguido el modelo de consumer y provider, que venían proporcionados por el proyecto starter Hello World.

No ha surgido ningún problema grave durante el desarrollo.

Para que esta acción de convertirse en un patrocinador a partir de otro usuario registrado se llevara a cabo se tuvo que añadir un par de líneas de código a la clase menú.jsp para indicar que habría un nuevo desplegable de nombre “Become a sponsor”.

### Validación por el profesor

EL profesor no lo ha revisado.

## Registro de Análisis 5

### Requisito

Operaciones de los patrocinadores en sus cuentas de usuario:

* Actualizar sus perfiles.

### Análisis y decisiones

Teniendo en cuenta lo mencionado para el requisito anterior, este se ha implementado igual.

La única diferencia notable es que en el servicio se ha contemplado la posibilidad de que al actualizar se tratara del mismo código del patrocinador que se tenía antes y no debería de saltar error. Asimismo, a la hora de acceder a este formulario de actualización de perfil que hará a través del desplegable configurado en menú.jsp, pero esta vez, como se accede desde el cliente logueado, el texto se mostrará será “Sponsor data” en vez de “Become a sponsor”

### Validación por el profesor

El profesor no lo ha revisado.

# Conclusiones

En esta tercera entrega, nos hemos enfrentado a un aumento significativo de la carga de trabajo al implementar varios requisitos funcionales de la tarea individual. La gestión efectiva de estas tareas ha sido dirigida por Adrian Cabello Martin.

La implementación de todos los requisitos asignados para esta entrega ha generado varios problemas al comienzo de esta debido a la falta de familiarización con el framework. No obstante, se superaron los requisitos satisfactoriamente.

Se tomaron decisiones de diseño tanto en el backend como en el frontend para optimizar la experiencia del usuario, como la inclusión de desplegables para facilitar la selección del tipo de cliente, mejorando así la usabilidad del sistema y la aparición o no del botón que redirige a las facturas asociados a un patrocinio en función de si ese patrocinio está en modo borrador o no.

La falta de revisión ha jugado un papel fundamental pero gracias a que los otros miembros del grupo si han recibido feedback se ha podido avanzar al unisono.

En resumen, se ha cumplido con una exitosa implementación de todos los requisitos necesarios para esta entrega.

# Bibliografía

En blanco a propósito.