Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2023 – 2024

Reporte de análisis estudiante 4

Josué Rodríguez López

josrodlop19@alum.us.es

Repositorio: https://github.com/DP2-C1-013/Acme-SF-D01-24.1.0

Fecha	Versión
27/5/2024	v4.0

1. Resumen **2**
2. Control de versiones **3**
3. Introducción **3**
4. Contenidos **4**
4.1 Entregable 1 4
4.2. Entregable 2 5
4.2.1 Entidad Sponsorship 5
4.2.2 Entidad Invoice 7
4.2.3 SponsorDashboard 8
4.2.4 Rol Sponsor 9
4.3. Entregable 3 11
4.3.1 SponsorSponsorshipShow 11
4.3.2 SponsorSponsorshipCreate, SponsorSponsorshipUpdate y SponsorSponsorshipPublish 11
4.3.3 SponsorInvoiceList 12
4.3.4 SponsorInvoiceShow 12
4.3.5 SponsorInvoiceCreate y SponsorInvoicePublish 13
4.3.6 SponsorInvoiceUpdate y SponsorInvoiceDelete 14
4.3.7 SponsorSponsorDashboardShow 14
4.3. Entregable 4 15
4.4. Diagrama UML resultante del análisis 16
5. Conclusión **16**
6. Bibliografía **16**

1. Resumen

Este reporte tiene como objetivo realizar un listado del análisis de los distintos requisitos individuales asignados al estudiante #4 para el proyecto de la signatura. Para cada requisito se dará información sobre las conclusiones a las que se ha llegado tras el análisis y de las decisiones tomadas.

2. Control de versiones

Nº de revisión	Fecha	Descripción
1.0	14/02/2024	Desarrollo inicial del reporte de análisis
2.0	07/03/2024	Desarrollo del análisis de las entidades.
3.0	26/04/2024	Desarrollo del análisis de los servicios.
4.0	27/05/2024	Corrección de errores.

3. Introducción

El reporte de análisis está centrado en el análisis de los requisitos asignados al estudiante #4. En esta versión del documento corresponde al entregable 4 del proyecto, pero siguen apareciendo los análisis de entregables anteriores con nuevas correcciones. Se analizarán los requisitos más complejos, centrándose principalmente en aquellos elementos que hayan requerido la valoración de distintas opciones. Los requisitos triviales no serán tratados o se hará de forma muy superficial si hicieran falta para el desarrollo del análisis de otro más complejo. Todos los requisitos han sido revisados en las sesiones de Follow Up.

El contenido del reporte de análisis está estructurado en los siguientes puntos:

1. Resumen ejecutivo: resumen muy general del contenido principal del reporte.
2. Control de versiones del documento.
3. Introducción: a diferencia del resumen ejecutivo, se realiza un resumen más detallado del contenido principal al mismo y se explica la estructura del documento.
4. Contenidos: punto principal del reporte en el que se entra con todo detalle en el análisis de los distintos requisitos.
5. Conclusión.
6. Bibliografía.

4. Contenidos

4.1 Entregable 1

En el entregable 1 no ha sido necesario el análisis de ninguno de los requisitos debido a la trivialidad de los mismos.

4.2. Entregable 2

4.2.1 Entidad Sponsorship

The sponsorships are related to a project with the aim of achieving greater visibility in the market. The system must store the following data about them: a code (pattern "[A-Z]{1,3}-[0-9]{3}", not blank, unique), a moment (in the past), a duration (after the moment, at least one month long), an amount (positive), a type of sponsorship ("Financial", "In kind"), an optional contact email, and an optional link with further information.

Este requisito conlleva la creación de una entidad Sponsorship que guarda información sobre el patrocinio de los proyectos. Por ello, esta entidad debe estar relacionada con la entidad Project. Sin embargo, no queda claro el tipo de relación que se debe implementar. A continuación se presentan las opciones más viables barajadas:

1. **Relación simple Many to One con multiplicidad 0..*:** esta opción conlleva que la relación admite que un proyecto no tenga ninguna entidad Sponsorship asociada.
2. **Agregación con multiplicidad 0..*:** caso similar al 1 pero se entiende la entidad Sponsorship como una parte de Project.
3. **Relación simple Many to One con multiplicidad 1..*:** esta opción conlleva que la relación admite que un proyecto tenga por lo menos una entidad Sponsorship asociada.
4. **Agregación con multiplicidad 1..*:** caso similar al 3 pero se entiende la entidad Sponsorship como una parte obligatoria de Project.

Se ha optado por la opción 1. Los motivos son que se asume que un proyecto puede no tener asociado ninguna entidad Sponsorship si el proyecto no se patrocina, por eso la multiplicidad 0..* opcional. Además, se entiende el patrocinio como un elemento externo que se relaciona con el proyecto y no como una parte del mismo, por lo que se utiliza una relación simple.

A continuación se muestra el análisis de las restricciones de los atributos de la entidad. Como nota, se considerará que un atributo es obligatorio a menos que se indique explícitamente lo contrario:

1. **code:** código que identifica una instancia de la entidad Sponsorship, de tipo String. Este atributo no puede ser una cadena vacía (NotBlank), debe ser único (Unique) y debe seguir el formato especificado por la expresión regular *[A-Z]{1,3}-[0-9]{3}*. Un ejemplo de una cadena que cumple el patrón de la expresión es ABC-123.
2. **moment:** de tipo Date, este atributo representa la fecha de registro del patrocinio (en el pasado). En cuanto a qué datos guardar en la base de datos para manejar estos atributos se han barajado dos opciones:
    a. **Fecha y hora (Temporal(TIMESTAMP)**: permite una definición del momento más detallada. Ocupa más espacio en memoria y es ligeramente más complejo de manejar.
    b. **Fecha Temporal(Date))**: es ligeramente más simple de manejar y ocupa menos espacio en memoria. Sin embargo, limita más el nivel de definición que se puede alcanzar.

    La opción elegida ha sido la a, fecha y hora. El motivo es que se valora más la definición del intervalo de duración del patrocinio que la eficiencia en memoria o una insignificante mejora del manejo.

Estos dos momentos deben estar separados por al menos un mes, siendo moment el primero de los dos cronológicamente. Puesto que esta restricción relaciona dos atributos, será implementada sobre el servicio.

3. **start y end:** de tipo Date, este par de atributos representa el inicio (start) y final (end) del patrocinio. Ambos deben ser obligatorios (NotNull) y en el caso de start, este debe ser un momento pasado (Past). Al igual que con moment, se ha decidido guardar Fecha y hora (Temporal(TIMESTAMP).

   Estos dos momentos deben estar separados por al menos un mes, siendo start el primero de los dos cronológicamente. Puesto que esta restricción relaciona dos atributos, será implementada sobre el servicio. También es necesario que start sea posterior (o igual) a moment.

4. **type**: indica el tipo del patrocinio. Este puede tomar dos valores: *In_kind* y *Financial*. Debido a que hay una lista de valores posibles se implementa como un enumerado.
5. **amount**: representa el presupuesto asignado al patrocinio. Atributo obligatorio (NotNull). Se han barajado tres posibles tipos para este atributo:
   a. **Integer**: simple de implementar pero imposibilita poner valores decimales.
   b. **Double**: simple de implementar y permite valores decimales.
   c. **Money**: tipo de dato personalizado que incluye el framework que permite poner cantidades decimales y especificar el tipo de moneda usada.

   Se ha optado por la opción 3 puesto que en el contexto del proyecto es necesario poder diferenciar los distintos tipos de divisa. Además, también se cree necesario poder manejar cantidades decimales de dinero.

   También es necesario aplicar restricción para que la cantidad monetaria sea estrictamente positiva, por lo que debe ser mayor que 0. Sin embargo, se considera que es posible realizar un patrocinio de forma altruista. Este argumento es respaldado por el atributo anterior, type, en el que uno de sus posibles valores es *In_kind*, que indica que el patrocinio se realiza a coste 0. Por ello, el patrocinio solo puede ser gratuito si type toma el valor *In_kind.* Estas dos restricciones se aplicarán directamente sobre el servicio debido a los siguientes motivos:

   - Min(0) no se puede implementar con la notación @Min(0) directamente sobre la entidad Sponsorship puesto que el atributo amount es del tipo Money implementado por el framework.
   - La relación entre type y el valor de amount solo se puede realizar en el servicio pues relaciona dos atributos.
6. **email**: atributo opcional de tipo String que representa la dirección de correo electrónico del encargado del patrocinio. Tiene que cumplir la restricción que valida que el formato es correcto (Email). También tiene que cumplir que su longitud máxima no sea mayor que 255 caracteres debido a que es el tamaño máximo soportado por la base de datos para un objeto de tipo cadena.
7. **link**: atributo opcional de tipo String que representa una dirección web de información sobre el patrocinio. Debe cumplir la restricción que valida que el formato sea el correcto (URL). También tiene que cumplir que su longitud máxima no sea mayor que 255 caracteres debido a que es el tamaño máximo soportado por la base de datos para un objeto de tipo cadena.
8. **draftMode**: atributo de tipo Boolean que representa, como veremos en el siguiente requisito, si la composición que asocia las entidades SponsorShip e Invoice está completa.

4.2.2 Entidad Invoice

Each sponsorship is billed through the use of invoices. The system must store the following data about them: a code (pattern "IN-[0-9]{4}-[0-9]{4}", not blank, unique), a registration time (in the past), a due date (at least one month ahead the registration time), a quantity (positive not nought), the tax that it is applied (positive or nought), the total amount (calculated by adding together the quantity and the tax applied), and an optional link with further information.

Este requisito conlleva la creación de una entidad Invoice que guarda información sobre las facturas emitidas a partir de un patrocinio. Por ello, esta entidad debe estar relacionada con Sponsorship. No obstante, nuevamente es necesario analizar qué tipo de relación es la más adecuada:

9. **Relación simple Many to One con multiplicidad 0..*:** esta opción conlleva que la relación admite que una entidad Sponsorship no tenga ninguna entidad Invoice asociada.
10. **Agregación con multiplicidad 0..*:** caso similar al 1 pero se entiende la entidad Invoice como una parte de Sponsorship.
11. **Relación simple Many to One con multiplicidad 1..*:** esta opción conlleva que la relación admite que una entidad Sponsorship tenga al menos entidad Invoice asociada.
12. **Agregación con multiplicidad 1..*:** caso similar al 3 pero se entiende la entidad Invoice como una parte obligatoria de Sponsorship.

En este requisito se ha seleccionado la opción 4. El motivo es que todo patrocinio debe tener obligatoriamente una factura asociada al estar tratando con pagos. Además, una factura se considera una parte de un patrocinio, por lo que no tendría sentido tratar a Invoice y Sponsorship como entidades diferentes que se relacionan mediante una relación simple.

A continuación se muestra el análisis de las restricciones de los atributos de la entidad. Como nota, se considerará que un atributo es obligatorio a menos que se indique explícitamente lo contrario:

13. **code**: código que identifica una instancia de la entidad Sponsorship, de tipo String. Este atributo no puede ser una cadena vacía (NotBlank), debe ser único (Unique) y debe seguir el formato especificado por la expresión regular *IN-[0-9]{4}-[0-9]{4}*. Un ejemplo de una cadena que cumple el patrón de la expresión es IN-1234-1234.
14. **registrationTime** y **dueDate:** de tipo Date, este par de atributos representa la fecha de registro de la factura (registrationTime) y la fecha límite en la que se debe pagar (dueDate). Ambos deben ser obligatorios (NotNull) y en el caso de registrationTime, este debe ser un momento pasado (Past). En cuanto a qué datos guardar en la base de datos para manejar estos atributos se han barajado dos opciones:
    a. **Fecha y hora (Temporal(TIMESTAMP):** permite una definición del periodo más detallada, pudiendo especificar el momento exacto de inicio y fin. Ocupa más espacio en memoria y es ligeramente más complejo de manejar.
    b. **Fecha Temporal(Date)):** es ligeramente más simple de manejar y ocupa menos espacio en memoria. Sin embargo, limita más el nivel de definición que se puede alcanzar.

La opción elegida ha sido la a, fecha y hora. El motivo es que es necesario saber el momento exacto de emisión y la fecha exacta hasta la que se puede pagar.

Estos dos momentos deben estar separados por al menos un mes, siendo registrationTime el primero de los dos cronológicamente. Puesto que esta restricción relaciona dos atributos, será implementada sobre el servicio.

15. **quantity**: representa la cantidad a pagar en la factura sin contar impuestos. Atributo obligatorio (NotNull) Se han barajado tres posibles tipos para este atributo:
    a. **Integer**: simple de implementar pero imposibilita poner valores decimales.
    b. **Double**: simple de implementar y permite valores decimales.
    c. **Money**: tipo de dato personalizado que incluye el framework que permite poner cantidades decimales y especificar el tipo de moneda usada.

    Se ha optado por la opción 3 puesto que en el contexto del proyecto es necesario poder diferenciar los distintos tipos de divisa. Además, también se necesita poder manejar cantidades decimales de dinero.

    También es necesario aplicar restricción para que la cantidad monetaria sea estrictamente positiva, por lo que debe ser mayor que 0. No obstante, por el mismo motivo que con el atributo amount del requisito anterior, esta cantidad sí que deberá poder ser 0. Esta restricción se implementará directamente en el servicio al no poderse implementar con anotaciones sobre la entidad por ser un tipo de dato personalizado ofrecido por el framework.

16. **tax**: de tipo double, representa el impuesto que se aplica sobre quantity. Al ser un porcentaje, su valor debe estar entre 0 (Min(0)) y 1 (Max(1)).
17. **totalAmount**: atributo derivado (Transient) de tipo Money, se calcula como la suma del atributo quantity más el atributo tax por quantity. Se debe devolver la cantidad en la misma moneda que en el atributo quantity.
18. **link:** atributo opcional de tipo String que representa una dirección web de información sobre la factura. Debe cumplir la restricción que valida que el formato sea el correcto (URL). También tiene que cumplir que su longitud máxima no sea mayor que 255 caracteres debido a que es el tamaño máximo soportado por la base de datos para un objeto de tipo cadena.

4.2.3 SponsorDashboard

The system must handle sponsor dashboards with the following data: total number of invoices with a tax less than or equal to 21.00%; total number of sponsorships with a link; average, deviation, minimum, and maximum amount of the sponsorships; average, deviation, minimum, and maximum quantity of the invoices.

Este requisito expresa que el sistema debe manejar datos derivados de de las entidades Sponsorship y de invoice. Estos datos que se piden serán calculados mediante consultas relativamente simples a la base de datos, por lo que no será necesario persistir los datos calculados. Estas consultas se implementarán sobre el repositorio. Por ello, estos datos se implementarán como un Form con los siguientes atributos, que al ser derivados no es necesario validar puesto que se suponen correctos de primera instancia:

- **numInvoicesLessTax**: de tipo Integer, representa el número de facturas con impuestos menores o iguales a 21%.

- **numLinkedSponsorship**: de tipo Integer, representa el número de patrocinios que tienen asociados un enlace web.
- **averageAmountSponsorship**: de tipo Money, representa la media del atributo amount de las entidades Sponsorship.
- **deviationAmountSponsorship**: de tipo Money, representa la desviación típica del atributo amount de las entidades Sponsorship.
- **minAmountSponsorship**: de tipo Money, representa el valor mínimo del atributo amount de las entidades Sponsorship.
- **maxAmountSponsorship**: de tipo Money, representa el valor máximo del atributo amount de las entidades Sponsorship.
- **averageAmountInvoice**: de tipo Money, representa la media del atributo quantity de las entidades Invoice.
- **deviationAmountInvoice**: de tipo Money, representa la desviación típica del atributo quantity de las entidades Invoice.
- **minAmountInvoice**: de tipo Money, representa el valor mínimo del atributo quantity de las entidades Invoice.
- **maxAmountInvoice**: de tipo Money, representa el valor máximo del atributo quantity de las entidades Invoice.

4.2.4 Rol Sponsor

There is a new project-specific role called sponsor, which has the following profile data: name (not blank, shorter than 76 characters), a list of expected benefits (not blank, shorter than 101 characters), an optional web page with further information, and an optional email contact.

Este requisito conlleva la creación de un nuevo rol Sponsor. Este rol debe estar relacionado con Sponsorship puesto que a cada una de ellas se le tiene que asignar un creador y administrador.

que guarda información sobre las facturas emitidas a partir de un patrocinio. Por ello, esta entidad debe estar relacionada con Sponsorship. No obstante, nuevamente es necesario analizar qué tipo de relación es la más adecuada:

1. **Relación simple Many to One con multiplicidad 0..*:** esta opción conlleva que la relación admite que un patrocinador no tenga ningún patrocinio asociado.
2. **Agregación con multiplicidad 0..*:** caso similar al 1 pero se entiende la entidad Sponsorship como una parte de Sponsor.
3. **Relación simple Many to One con multiplicidad 1..*:** esta opción conlleva que la relación admite que un patrocinador tenga al menos un patrocinio asociado en todo momento (creación del sponsor incluida).
4. **Agregación con multiplicidad 1..*:** caso similar al 3 pero se entiende la entidad Sponsorship como una parte obligatoria de Sponsor.

En este requisito se ha seleccionado la opción 1. El motivo es que un sponsor no tiene por qué tener asociado un patrocinio en todo momento, especialmente después de la creación donde suele pasar un tiempo sin que tenga asociada una hasta que la cree (esto descarta las opciones 3 y 4). También en este contexto no tiene sentido concebir el sponsorship

como parte de sponsor puesto que son conceptos distintos por naturaleza esto descarta las opciones 2 y 4).

Sponsor se implementará usando una entidad que extienda a AbstractRole. El requisito pide que se guarde información que se ha modelado con los siguientes atributos (se considera obligatorio un atributo a menos que se indique explícitamente lo contrario):

- **name**: nombre de la persona asociada al rol, de tipo String. Debe ser menor de 76 caracteres (Length(max(75)) y no puede ser una cadena vacía (NotBlank).
- **benefits**: representa la lista de beneficios que obtiene el patrocinador. Como en los requisitos se indica que no puede ser una cadena vacía (NotBlank) y que tiene que tener una longitud menor que 76 caracteres (Length(max(75)) se asume que este atributo se modela usando el tipo String.
- **link**: de tipo String, representa un enlace a una página web que ofrece más información sobre el patrocinador. Es un atributo opcional. que debe cumplir una restricción que valida que el texto esté en el formato correcto (Email). También tiene que cumplir que su longitud máxima no sea mayor que 255 caracteres debido a que es el tamaño máximo soportado por la base de datos para un objeto de tipo cadena.
- **email**: de tipo String, representa el correo electrónico del patrocinador. Es una atributo opcional que debe cumplir una restricción que valida que el texto esté en el formato correcto (URL). También tiene que cumplir que su longitud máxima no sea mayor que 255 caracteres debido a que es el tamaño máximo soportado por la base de datos para un objeto de tipo cadena.

La información a guardar de un patrocinador queda clara, pero presenta un problema. En el sistema, un rol se modela extendiendo la entidad abstracta AbstractRole. Esta entidad abstracta tiene presenta una relación ManytoOne (obligatoria) con la entidad UserAccount, que a su vez, tiene un atributo de tipo DefaultUserEntity. Esta última obliga a guardar en sus atributos el nombre completo del usuario y su correo electrónico, por lo que no sería necesario modelar los atributos name y email al estar ya implementados por el framework.

4.3. Entregable 3

En todos los requisitos de este punto se asume que sólamente el rol especificado puede acceder a las funcionalidades especificadas. Aquellos requisitos que no se mencionan se consideran triviales.

4.3.1 SponsorSponsorshipShow

Operations by sponsors on sponsorships: Show the details of their sponsorships.

Sólo se debe permitir a un patrocinador acceder a la información de un patrocinio si este está creado por él.

4.3.2 SponsorSponsorshipCreate, SponsorSponsorshipUpdate y SponsorSponsorshipPublish

Operations by sponsors on sponsorships: Create, update, or delete their sponsorships. Sponsorships can be updated or deleted as long as they have not been published. For a sponsorship to be published, the sum of the total amount of all their invoices must be equal to the amount of the sponsorship.

Este requisito conlleva la implementación de las funcionalidades de crear, actualizar, publicar y eliminar un patrocinio. Sólamente un patrocinador que sea el creador de un patrocinio puede aplicar estas operaciones. Estas funcionalidades comparten algunas restricciones:

- La divisa (reconocida por el sistema) del campo amount puede ser distinta a la de las facturas asociadas, lo que dificulta el manejo de las cantidades. Por ello, se han barajado distintas alternativas:
    1. Implementar un conversor de monedas. El punto a favor de esta opción es que permite tener los valores reales en distintas divisas pero se visualizan con la divisa principal del sistema. El punto en contra es que es costoso de implementar.
    2. Obligar a que la divisa del patrocinio sea la misma que la de sus facturas. Esta opción es mucho más simple de implementar pero limita más el sistema.

  Se ha elegido la opción 2 al ser más económica en cuanto a tiempo y recursos. **Esta restricción no se aplica al borrado.**
- El proyecto asociado debe estar publicado. No tiene sentido crear, modificar o eliminar un patrocinio asociado a un proyecto que aún está en modo borrador.
- Si el patrocinio tiene asociado al menos una factura, no se podrá modificar el atributo type. Esto se debe a que en caso de no contar con esta restricción sería posible cambiar un patrocinio de tipo *In_kind* a *Financial* aún teniendo facturas con valor monetario 0.00 ya publicadas.
- Otro problema que aparece es cómo se interactúa con el atributo moment que representa la fecha de creación. Se plantean dos alternativas:
    1. Se genera de forma automática y no puede ser modificable una vez creado. Es consistente y evita problemas de errores humanos en las fechas.

2. El usuario se encarga de la gestión del atributo. Esta opción es más flexible pero menos exacta en el momento de la creación y puede desembocar en datos erróneos.

Se ha elegido la opción 1 puesto que es más consistente y cómoda para el usuario.

Otras restricciones son individuales de la función:

- (Publish) Publicar un patrocinio de tipo *In_kind* sin tener asociada ninguna factura. Esto no supone realmente un problema en el sistema, pero es conveniente tenerlo controlado. Por ello se plantean dos alternativas:
    1. Obligar a que haya al menos una factura con cantidad 0,00. La ventaja que plantea es tener más estandarizado el proceso de publicar un patrocinio.
    2. Permitir publicar sin facturas. Facilita el proceso.

  Se ha elegido la opción 1 tras observar como sistemas reales envían facturas con coste 0.
- (Publish y Update) Cuando se actualice la cantidad monetaria del patrocinio, esta no debe ser menor que la suma de la cantidad total (incluyendo impuestos) de sus facturas. Esto se debe a que en caso contrario los datos del sistema nos serían consistentes.

4.3.3 SponsorInvoiceList

Operations by sponsors on invoices: *List the invoices in their sponsorships.*

Este requisito conlleva el listado de las facturas. Sólo un patrocinador que haya creado el patrocinio asociado puede acceder a sus facturas. En cuanto a como se muestra el listado se han planteado dos alternativas:

1. Una ventana en la que estén todas las facturas creadas por el patrocinador autenticado en el sistema. La principal ventaja es que así se tiene toda la información en un solo punto por lo que es más fácil acceder a ella. La parte negativa es que es más complicado agruparlas por patrocinio.
2. Acceder sólo a las facturas de un patrocinio mediante un botón situado en la ventana que muestra la información del patrocinio. La ventaja es que así es más intuitivo acceder a las facturas de un patrocinio en concreto. La parte negativa es que es más incómodo mostrar todas las facturas al tener que ir de patrocinio en patrocinio.

Se ha elegido la alternativa dos puesto que se considera que es más natural y cómodo en la mayoría de los casos tener agrupadas las facturas dentro de su patrocinio.

4.3.4 SponsorInvoiceShow

Operations by sponsors on invoices: Show the details of their invoices.

Este requisito exige que el sistema permita a un patrocinador ver la información de sus facturas. Pero esto plantea un problema: si un patrocinio tiene facturas de otros patrocinadores este también debería ser capaz de verlas. Por ello se plantean las siguientes alternativas:

1. Un patrocinio solo puede tener asociado facturas creadas por el mismo patrocinador que creó el patrocinio. Esta forma es fácil de implementar y evita problemas de privacidad.
2. Un patrocinio puede tener asociado facturas creadas por varios patrocinadores pero un patrocinador sólo puede acceder a la información de las suyas. Esta forma le da flexibilidad al sistema conservando la privacidad de los usuarios. Sin embargo, impide una correcta gestión de las facturas de un patrocinio.
3. Un patrocinio puede tener asociado facturas creadas por varios patrocinadores y un patrocinador puede acceder a todas ellas. Esta forma le da flexibilidad al sistema conservando sin dificultar la gestión. No obstante, deja de lado la privacidad de los usuarios que podrían no desear que otras personas vean sus facturas.

La opción elegida es la 1 puesto que facilita la gestión de las facturas de los patrocinios al mismo tiempo que valora la privacidad a pesar de hacer el sistema más rígido. **Esta restricción se aplicará a las funciones de creación, modificación, publicación y borrado de facturas.**

4.3.5 SponsorInvoiceCreate y SponsorInvoicePublish

Operations by sponsors on invoices: Create and publish an invoice.

Este requisito conlleva implementar la creación y publicación de las facturas. Sólo un patrocinador que haya creado el patrocinio asociado puede hacer uso de estas funciones sobre una factura asociada a dicho patrocinio. Otra restricción a la hora de aplicarlas es que el patrocinio asociado esté en modo borrador (draft mode) puesto que se considera necesario que todas las facturas estén emitidas antes de poder ser publicado.

Al igual que con la creación, publicación y modificación de los patrocinios en estas funcionalidades no encontramos con el mismo problema sobre las divisas. La solución escogida ha sido la misma que en los patrocinios.

Otro problema que aparece es cómo se interactúa con el atributo registrationDate que representa la fecha de creación. Se plantean dos alternativas:

1. Se genera de forma automática y no puede ser modificable una vez creado. Es consistente y evita problemas de errores humanos en las fechas.
2. El usuario se encarga de la gestión del atributo. Esta opción es más flexible pero menos exacta en el momento de la creación y puede desembocar en datos erróneos.

Se ha elegido la opción 1 puesto que es más consistente y cómoda para el usuario.

El último problema encontrado es si dejar que un patrocinio se publique si aún tiene facturas en modo borrador. Se han planteado las siguientes alternativas:

1. Permitir la publicación. Esta opción es más cómodo para el usuario puesto que no tiene por qué gestionar todos los borradores en un patrocinio para poder publicarlo. Sin embargo, dejar información sin publicar, es decir, que no es realmente relevante, supone un gasto de almacenamiento. Además, puede generar problemas de inconsistencia si hubiese un bug que permitiera publicarlas.

2. No permitir la publicación mientras haya facturas en modo borrador. Aunque esta opción es más incómoda para el usuario, es más óptima en cuanto a recursos, a parte de ser más consistente.

Se ha elegido la opción 2 puesto que la integridad de los datos es lo que más se valora en este sistema.

4.3.6 SponsorInvoiceUpdate y SponsorInvoiceDelete

Operations by sponsors on invoices: Update or delete an invoice as long as it is not published.

Este requisito tiene exactamente las mismas restricciones que el anterior. Solo cambia que es obligatorio que la factura esté en modo borrador (draft mode) para poder realizar las operaciones.

4.3.7 SponsorSponsorDashboardShow

Operations by sponsors on invoices: Show their sponsor dashboards.

Este requisito exige mostrar el tablero del patrocinador. En este punto se han encontrado dos problemas:

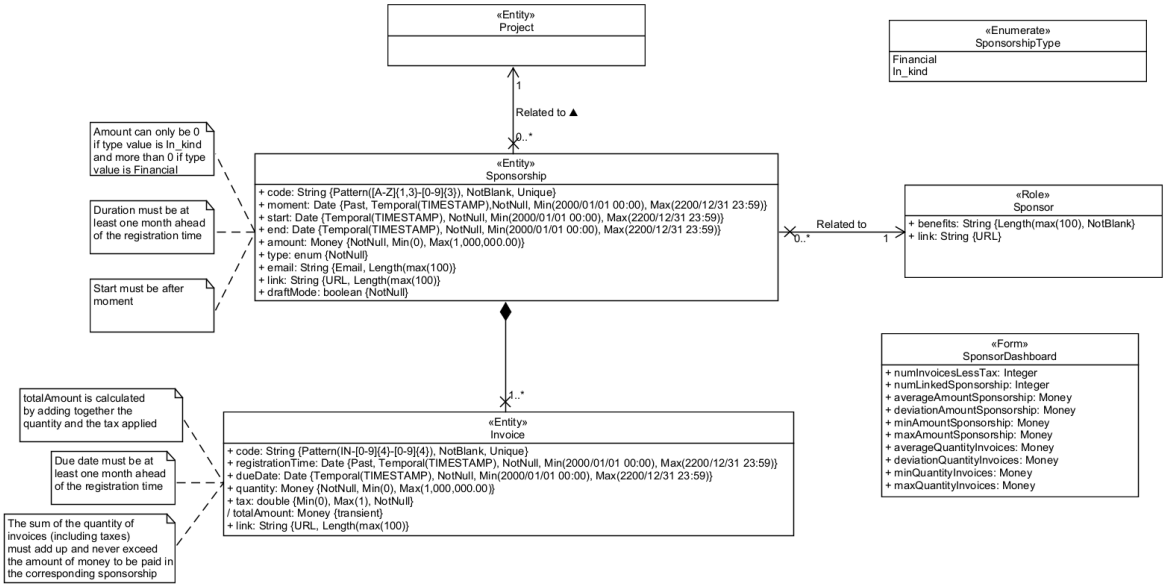
- Al haber valores en distintas divisas no se pueden hacer los cálculos de las estadísticas de forma coherente. Se proponen dos alternativas:
    1. Implementar un conversor de monedas. que permite tener estadísticas globales del sistema. El punto en contra es que es costoso de implementar.
    2. Separar las estadísticas por divisa. La implementación de esta alternativa es más económica en tiempo y recursos pero los resultados, al estar separados por divisas, no se pueden comparar y unificar fácilmente.

  Se ha elegido la opción 2 al ser más económica.
- Tener en cuenta todos los datos del sistema o solo aquellos que estén publicados. Se ha decidido utilizar solo los datos de entidades publicadas puesto que es común que los usuarios creen diversos borradores que al final se acaban descartando. Si se usaran todos los datos el tablero mostraría datos que no reflejan la realidad.

4.3. Entregable 4

En el entregable 4 no ha sido necesario el análisis de ninguno de los requisitos debido a la trivialidad de los mismos.

4.4. Diagrama UML resultante del análisis



5. Conclusión

Este documento está actualmente en su versión final, correspondiente al cuarto entregable. En esta última entrega se han detectado nuevos errores sobre los requisitos y se han corregido.

6. Bibliografía

Intencionadamente en blanco.