



LaSalaShop

Eduardo Balsa
Grado Ingeniería Informática
Ingeniería del Software

Oriol Martí
Santi Caballe

12/01/2017



Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-CompartirIgual
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>LaSalaShop</i>
Nombre del autor:	<i>Eduardo Balsa Sediles</i>
Nombre del consultor/a:	<i>Oriol Martí Girona</i>
Nombre del PRA:	<i>Santi Caballe Llobet</i>
Fecha de entrega (mm/aaaa):	01/2017
Titulación:	<i>Grado Ingeniería Informática</i>
Área del Trabajo Final:	<i>Ingeniería del software</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Java, Prototipos DCU, Ingeniería del software</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La naturaleza cambiante y ágil de determinados tipos de negocio hace que sea necesario el desarrollo de soluciones de software que se adapten a cada casuística particular.

El presente documento recoge el trabajo realizado para desarrollar una solución a medida para el negocio LaSalaShop, un espacio donde conviven las exposiciones punteras de artistas emergentes, con la venta de libros y *merchandising* en general. Este negocio presenta un problema de gestión, tanto de la mercancía, que se agrava por la naturaleza variada de esta, como de optimización de sus espacios de exposiciones.

Para acometer este proyecto se ha utilizado metodología basada el desarrollo ágil de software. Asimismo, se ha utilizado un sistema basado en el DCU, a través de prototipos diseñados específicamente para medir la adaptación, usabilidad y satisfacción de cada uno de los modelos, con la intención de priorizar el valor para el usuario. Desde el punto de vista de la arquitectura, se ha realizado un diseño en 3 capas para dirigido a tecnología Java, buscando cumplir con el principio open-closed.

Finalmente los resultados han sido los esperados puesto que se ha conseguido desarrollar aquello que se necesitaba, dentro del calendario previsto, y con una calidad aceptada por los usuarios. A modo de conclusión general se podría decir que la aplicación de las metodologías adquiridas a lo largo del Grado junto con una base sólida aportada por el PMBOK y la flexibilidad características de las metodologías ágiles constituyen una combinación que ofrece muchas garantías de éxito en este tipo de proyectos.

Abstract (in English, 250 words or less):

The agile and changing nature of some of today's business makes necessary developing software solutions able to address any concrete need.

The present document contains the work performed to design a customized solution for LaSalaShop, a space that mix the most vanguardist artistic events with the sale of art books and merchandising. This business has a critical issue in their management system or more precisely in the lack of any management system at all.

Most part of this project has been driven by an agile approach. In addition, a UCD system has been used, leading the design by UI prototypes, allowing an high UX satisfaction degree and a higher matching between needs and features.

From an architectural point of view, the system had been designed to be build upon a 3-layer Java technology software, in compliance with the open-closed design pattern.

Finally the expected results had been meet in terms of scope, time and quality. As a general conclusion we could say that the mix of a solid framework like the PMBOK, with the flexibility offered by the agile methodologies, in addition with the competencies acquired along the degree, conforms a very reliable combination to achieve project objectives.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	4
1.3 Enfoque y método seguido.....	5
1.4 Planificación del Trabajo.....	6
1.5 Breve resumen de productos obtenidos.....	10
1.6 Breve descripción de los otros capítulos de la memoria.....	11
2. Metodología.....	12
3. Análisis y registro de los interesados.....	15
4. Gestión de riesgos.....	16
5. Análisis funcional.....	18
5.1 Análisis y casos de uso del módulo de gestión de libros.....	19
5.2 Diagrama de clases del módulo de gestión de libros.....	26
5.3 Análisis y casos de uso del módulo de gestión de stock.....	27
5.4 Diagrama de clases del módulo de gestión de stock.....	30
5.5 Análisis y casos de uso del módulo de gestión de eventos.....	31
5.6 Diagrama de clases del módulo de gestión de eventos.....	33
5.7 Diagramas de secuencia.....	34
5.8 Módulo de administración.....	37
6. Diseño técnico.....	38
6.1 Arquitectura del sistema.....	38
6.2 Diagramas de arquitectura de la capa de presentación.....	39
6.3 Diagramas de arquitectura de la capa de negocio.....	42
6.4 Diagramas de arquitectura de la capa de persistencia.....	45
6.5 Modelo relacional.....	47
7. Prototipos: Evaluación de los modelos y decisiones adoptadas.....	51
8. Conclusiones.....	61
9. Glosario.....	64
10. Bibliografía.....	66
11. Anexo.....	67

Lista de figuras

Ilustración 1. EDT del proyecto	6
Ilustración 2. Tabla de hitos del proyecto	7
Ilustración 3. Diagrama de Gantt	9
Ilustración 4. Esquema de iteración	12
Ilustración 5. Registro de interesados	15
Ilustración 6. Registro de riesgos	16
Ilustración 7. Diagrama de clases módulo gestión libros	26
Ilustración 8. Diagrama de clases módulo gestión stock	30
Ilustración 9. Diagrama de clases módulo gestión eventos	33
Ilustración 10. Diagrama de secuencia CU_L1	34
Ilustración 11. Diagrama de secuencia CU_L7	35
Ilustración 12. Diagrama de secuencia CU_S7	35
Ilustración 13. Diagrama de secuencia CU_E5	36
Ilustración 14. Diagrama capa presentación módulo gestión libros	39
Ilustración 15. Diagrama capa presentación módulo gestión stock	40
Ilustración 16. Diagrama capa presentación módulo gestión eventos	41
Ilustración 17. Diagrama capa negocio módulo gestión libros	42
Ilustración 18. Diagrama capa negocio módulo gestión stock	43
Ilustración 19. Diagrama capa negocio módulo gestión eventos	44
Ilustración 20. Diagrama capa persistencia módulo gestión libros	45
Ilustración 21. Diagrama capa persistencia módulo gestión stock	46
Ilustración 22. Diagrama capa persistencia módulo gestión eventos	46

1. Introducción

1.1 Contexto y justificación del Trabajo

LaSalaShop, es un pequeño negocio dedicado a la exposición y promoción de artistas emergentes, principalmente nacionales. Tiene varias salas dedicadas a este propósito, en las que se muestran exposiciones temáticas de carácter temporal, de muy variadas temáticas, con un único elemento común: el arte contemporáneo.

Además, las instalaciones incluyen una tienda que ofrece una amplia gama de productos relacionados con el arte, tales como libros, ropa, regalos, etc. Esta es una parte muy importante del proyecto, puesto que constituye la principal fuente de ingresos, y es por tanto necesaria una cierta rentabilidad. En este sentido, la venta de libros, se ha ido convirtiendo paulatinamente en el área del negocio más rentable, por lo que se ha ido aumentando el número de libros en depósito, que las diferentes editoriales ceden a la sala.

La gestión de este negocio resulta algo compleja, debido a varios motivos:

- El elevado número de productos a ofrecer, así como la heterogeneidad en cuanto a tipos, proveedores etc., complican el seguimiento y control de ventas y stock.
- El aumento del volumen de negocio, y por tanto de mercancía en el área de libros, hacen muy costosa la gestión de devoluciones.
- El carácter temporal de las exposiciones, y la disponibilidad de obras y artistas hacen que sea difícil optimizar los metros disponibles para exponer.
- El propio modelo de negocio, obliga a hacer una gestión eficiente de los recursos para poder obtener la rentabilidad deseada.

Actualmente, el negocio carece de cualquier sistema de gestión informatizado, Toda la información se recoge de manera impresa, a través de albaranes, tickets de venta, cuadernos de notas, etc.

Esta carencia de cualquier tipo de sistema provoca que se pierda mucha cantidad de información, imposibilita la obtención de datos para toma de decisiones, y complica mucho cualquier operativa. La suma de todos estos factores provocan una pérdida considerable de eficiencia, y por tanto de rentabilidad económica.

Se trata, por tanto, de un proyecto de una gran relevancia, puesto que además de cumplir con las expectativas y especificaciones propias de un TFG universitario, deberá ser capaz de solucionar un problema real.

En este sentido se pretende diseñar un producto que sea capaz de cubrir los siguientes requisitos de carácter general, y desde el punto de vista del producto obtenido:

Requisito R-01	Se tiene que poder gestionar el sistema de depósito de libros
Descripción	Se podrá registrar y consultar cada uno de los libros, así como listarlos por editorial, preparar devoluciones etc.
Tipo	Funcionalidad.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

Requisito R-02	Se tiene que poder catalogar y controlar el stock existente.
Descripción	Se podrán crear categorías de artículos, que permitan su gestión y consulta, así como inventariar el stock existente.
Tipo	Funcionalidad.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

Requisito R-03	Se tiene que poder registrar información de las ventas
Descripción	Se podrá registrar información de cada venta, incluyendo tanto los productos que la forman, como el tipo de cliente que la ha realizado.
Tipo	Funcionalidad/Datos.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

Requisito R-04	Se tiene que poder generar informes y estadísticas de los procesos
Descripción	Se deberá poder usar la información registrada por el resto de procesos, a través de informes y estadísticas, que permitan la toma de decisiones estratégicas
Tipo	Funcionalidad/Datos.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

Requisito R-05	Se creará un registro centralizado con todos los datos de los artistas y exposiciones.
Descripción	Se deberá poder almacenar y consultar la información relativa a artistas y exposiciones de manera clara.
Tipo	Datos.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

Requisito R-06	Se creará una agenda que permita planificar y gestionar las salas y exposiciones
Descripción	Se deberá poder almacenar, consultar y actualizar una agenda que recoja la disponibilidad de las salas en unas fechas determinadas.
Tipo	Funcionalidad.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

1.2 Objetivos del Trabajo

1.2.1 Objetivo general

Crear un producto de software que permita solucionar los problemas y necesidades detectadas en el área de gestión del negocio. Concretamente el software resultante deberá ser capaz de facilitar e informatizar las tareas relacionadas con la gestión del stock, tanto de libros como de mercancía en general, y aportar un sistema de optimización de los espacios de exposiciones.

1.2.1 Objetivos específicos

- Poner en práctica y evaluar la efectividad de la metodología ágil y la experiencia de usuario, mediante prototipos, en un proyecto de duración media.
- Gestionar un proyecto basado tanto en los principios de las metodologías ágiles, como en un marco de gestión más tradicional.
- Acometer un proyecto real que permita evaluar las competencias adquiridas a lo largo de la titulación.
- Desarrollar un producto que permita gestionar un sistema de depósito de libros, así como la venta y catalogación.
- Implantar un sistema de gestión de stock, que permita catalogar y controlar toda la mercancía existente, y crear una trazabilidad mínima de los artículos.
- Obtener información de las ventas de artículos. En una primera fase se almacenarán estos datos, para posteriormente poder ser utilizados como base de datos estratégica.
- Crear un registro único y actualizado de todos los datos relacionados con los artistas, exposiciones y obras.

1.3 Enfoque y método seguido

Una vez definidos los requisitos y objetivos generales que deberá aportar la solución escogida, se procede al análisis de posibles alternativas, en lo que al tipo de producto se refiere. Principalmente, la disyuntiva en este punto consiste en escoger la adaptación de algún producto ya existente, o el desarrollo de uno completamente nuevo.

De cara a escoger una solución u otra, se procede a realizar un estudio de los diferentes productos ya existentes, basado principalmente en lo pueden aportar en el cumplimiento de los objetivos marcados.

Por un lado se ha analizado la posibilidad de instalación de algún software de gestión gratuito ya existente. Se evalúa la idoneidad de ciertos productos como *Alvendi* o *Prestashop*. El principal inconveniente de este tipo de productos, es que no encajan totalmente con las funcionalidades demandadas. Mientras que son demasiado complejos en alguna de las áreas, como la gestión de stock, no permiten integrar el resto de módulos requeridos, u otras necesidades que puedan surgir de la explotación del sistema informático.

Por otro lado, se pretende que la solución sea muy usable y adaptada, para reducir los tiempos y la complejidad, por lo que el uso de este tipo de soluciones genéricas no parecen una opción viable.

Se analiza del mismo modo, la posibilidad de implantar alguna solución de pago como *InterTPV* o *SQLPyme*, pero existe una problemática similar, a la que hay que añadir que los pagos de las licencias, comprometen la viabilidad de la inversión.

Teniendo en cuenta estos factores, se opta por el desarrollo de un sistema nuevo, a medida que se ajuste a las necesidades concretas de este negocio. Para ello, se desarrollará mediante la tecnología Java EE una aplicación dividida en módulos funcionales. Gracias a la potencia de esta tecnología, en combinación con EJB, JPA y JSF, podemos construir un proyecto muy flexible y robusto en poco tiempo. Además el carácter abierto, y la modularidad e interconexión de Java, deja totalmente abierta la posibilidad de ampliación del sistema con nuevas funcionalidades en el futuro, o la integración con nuevos tipos de software.

1.4 Planificación del Trabajo

De acuerdo a las recomendaciones recogidas en el PMBOK acerca de gestión de la planificación, se ha elaborado una EDT basada en teniendo en cuenta en primer lugar las fases, y a continuación los entregables.

El nivel de descomposición de las tareas, desciende sin descomponer los entregables en tareas de menos de 8 horas.

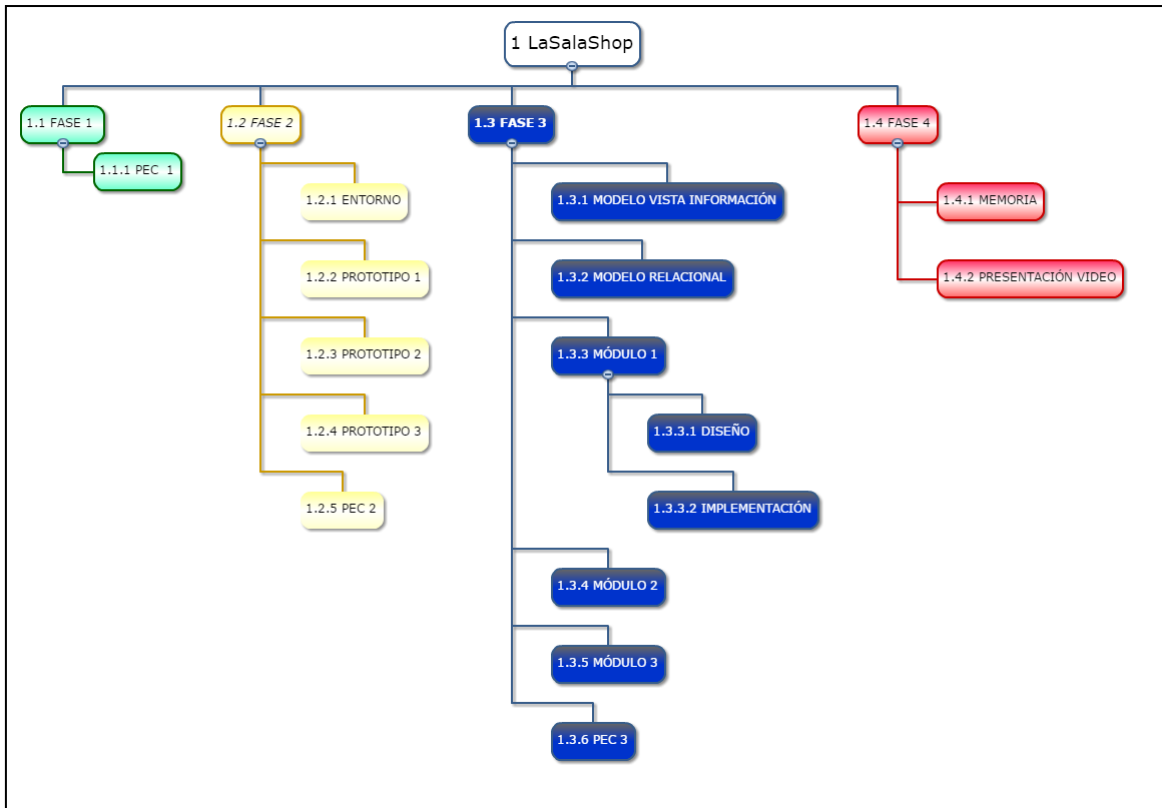


Ilustración 1. EDT del proyecto

El proyecto se divide en 4 fases, cada una de las cuales, culmina con la entrega de una de las prácticas, o entrega final.

Los hitos que marcan estas fases son los siguientes:

	INICIO	FIN
FASE 1	22/09/2016	05/10/2016
FASE 2	06/10/2016	09/11/2016
FASE 3	10/11/2016	23/12/2106
FASE 4	24/12/2016	12/01/2016

Ilustración 2. Tabla de hitos del proyecto

Además de esta tabla de hitos, para realizar la planificación temporal, se ha tenido en cuenta la descomposición de tareas (EDT), la matriz de riesgos que se lista en el siguiente apartado, la disponibilidad de los recursos, y en general, las recomendaciones y procesos del PMBOK.

A continuación, se desglosa con mayor detalle el propósito de cada una de las fases:

FASE 1: La primera fase del proyecto, incluye un primer análisis, tanto de contexto, como de requisitos, establecidos a través de una entrevista inicial con los usuarios. Se incluye además, la planificación general de todo el proyecto, el análisis de riesgos, de interesados, y se detalla el trabajo a realizar. Todo ello queda recogido en el entregable de esta fase, que es exclusivamente este documento.

FASE 2: A lo largo de la segunda fase, se va a desarrollar el análisis funcional y de usabilidad de 3 módulos distintos. Se seguirá una metodología ágil, por lo que se producirá una entrevista semana con los usuarios, con el propósito de matizar los requisitos de cada módulo, y valorar los prototipos. Además, a lo largo de esta fase, se completará la configuración de los entornos de desarrollo. El trabajo resultante de esta fase, quedará reflejado en la segunda práctica a entregar.

FASE 3: Una vez refinadas las necesidades funcionales de cada módulo, se procederá a completar el diseño técnico de toda la aplicación. Este diseño deberá incluir al menos los puntos de vista de la información y la tecnología, a través de diagramas UML, detallando de este modo la implementación de la solución.

A lo largo de esta fase, se acometerá además el desarrollo de una funcionalidad completa, mediante la implementación y consiguiente plan de pruebas, de un módulo funcional en su totalidad.

FASE 4: La última fase de este proyecto, incluye la elaboración tanto de la memoria como la presentación en video del mismo, así como las últimas revisiones del módulo implementado, resultantes del plan de pruebas de la fase 3.

CONTROL: De manera paralela al desarrollo por fases del proyecto, se seguirá un proceso control. De manera iterativa, cada dos semanas, o cada vez que se alcance un hito del proyecto, se evaluará el desarrollo actual respecto a la planificación, se revisarán los riesgos, la comunicación, y se abordará cualquier otra cuestión que pueda surgir.

Nota planificación: Se ha calculado el calendario de acuerdo a la disponibilidad de los recursos. Concretamente, se incluyen los sábados como laborales, y una jornada diaria de 4 horas. Se incluyen además las vacaciones planificadas durante este periodo.

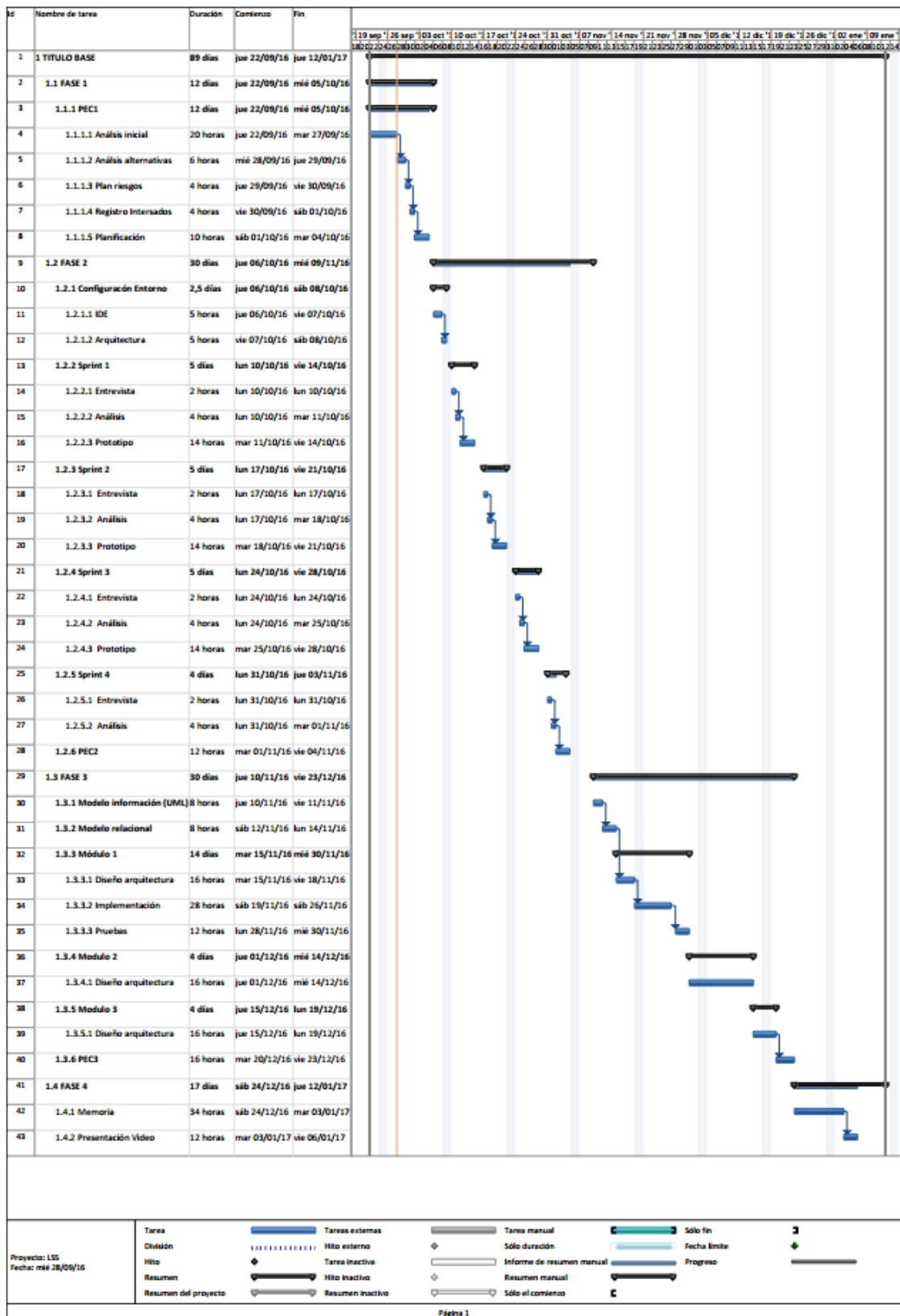


Ilustración 3. Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

El resultado final del proyecto está formado por los siguientes entregables:

- La presenta memoria final, en la que se detalla la metodología seguida, así como los resultados de los procesos de planificación, control y ejecución seguidos, con especial detalle en los apartados de análisis y diseño.
- El software desarrollado para ser entregado a los usuarios. Este es el resultante principal desde el punto de vista del producto, y contiene implementadas una serie de funcionalidades escogidas como las más valiosas entre las detectadas y analizadas.
- El video DemoLaSalaShop.mp4, que se incluye dentro de la entrega de la memoria, que debe servir a modo de demo de la aplicación resultante, y cuya finalidad es facilitar la revisión de la aplicación por el tribunal.
- La presentación PresentaciónLaSalaShop.mp4 en formato *PechaKucha*, destinada a servir de síntesis del desarrollo total del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

El primer capítulo del documento está dedicado íntegramente a hacer una introducción e ilustrar brevemente el contenido de la memoria. Asimismo, incluye toda la información técnica asociada al TFG.

A lo largo del segundo capítulo, se trata de ofrecer una imagen detallada de las metodologías escogidas para abordar el proyecto, las razones por las que se han escogido y los objetivos que se esperan de estas.

Tratándose de un proyecto con un impacto real, se ha considerado oportuno introducir un apartado especial dedicado al mostrar el análisis de los interesados, y el registro resultante de este. Asimismo, con la intención de tener respuestas ante los acontecimientos positivos o negativos que pudiese acontecer, se incluye un capítulo específico dedicado al análisis y gestión de los riesgos. Esta información se puede consultar en los capítulos tercero y cuarto respectivamente.

Por otro lado, siendo un proyecto relacionado con la especialidad de Ingeniería del software, se ha hecho especial hincapié en los capítulos quinto y sexto, que contienen toda la información relativa al análisis funcional y el diseño técnico de la solución. Estos, son además, por los motivos expuestos, los capítulos de una mayor extensión y un mayor nivel de detalle.

Los prototipos de interacción con el usuario han sido un elemento especialmente importante en el desarrollo del proyecto, por lo que se ha incluido un capítulo, el séptimo, con una pequeña muestra de los mismos, así como las decisiones tomadas a partir de las pruebas con usuarios.

El octavo capítulo está enteramente dedicado a recoger las conclusiones, lecciones aprendidas y activos de conocimiento en general adquiridos a lo largo del proyecto.

Los tres últimos capítulos incluyen información adicional, compuesta por el glosario de términos, la bibliografía, y un anexo con información acerca del código fuente y el resto de entregables.

2. Metodología

Se ha considerado interesante comenzar el apartado de la memoria dedicado al desarrollo del proyecto y el producto, con una explicación clara de la metodología y los fundamentos que se han aplicado. El motivo es que mediante la clarificación de la intención final y el modo de llegar a ella, puede resultar mucho más sencillo entender y analizar el resto de capítulos.

En este sentido, el tipo de negocio para el que se ha diseñado la solución, y la posibilidad de interactuar con los usuarios de manera directa, han condicionado en gran medida la decisión acerca de la hora de escoger las metodologías que debían guiar el proyecto. Estos hechos, junto con la necesidad de crear un producto a medida, que cumpliera exactamente con las necesidades de los usuarios, han hecho que este fuese un proyecto especialmente indicado para el uso de metodologías ágiles de desarrollo de software.

Esta metodología ha sido visible sobre todo a lo largo de la segunda fase del proyecto. A lo largo de esta fase (que culminó con la entrega de la PEC2), se realizaron sprints semanales, que comenzaron con una reunión con los usuarios del software. A lo largo de estos sprints, se concretaron y matizaron las funcionalidades de cada uno de los 3 módulos en los que se ha descompuesto el sistema.

Concretamente, se han seguido iteraciones semanales siguiendo el siguiente esquema:

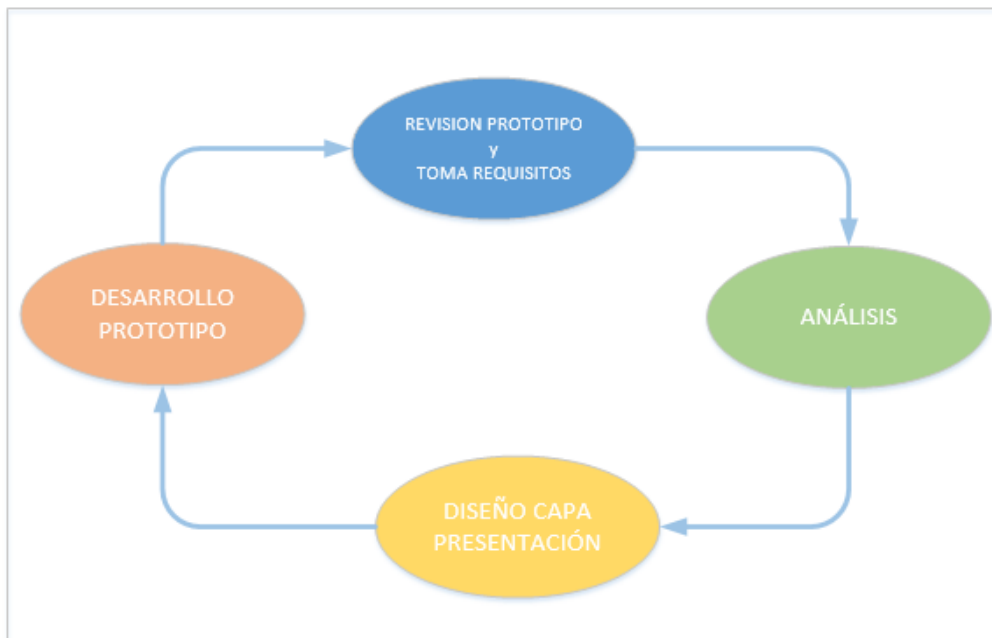


Ilustración 4. Esquema de iteración

Estos ciclos iterativos, no solo han permitido una gestión ágil próxima al *scrum*, sino que además han favorecido un desarrollo centrado en el usuario (DCU).

Tal y como se aprecia en la ilustración 4, el diseño ha sido guiado a través de prototipos de la capa de presentación, que han permitido ajustar la usabilidad y refinar el diseño, aumentando de este modo el valor final y la utilidad para los usuarios.

Una obra de referencia importante para haber escogido esta metodología y flujos de trabajo, es *Agile Estimating and Planning* de Mike Cohn [1]. Siguiendo las metodologías que propone Cohn, se ha llegado a la conclusión de que este modelo de trabajo está justificado en este caso, dado que los prototipos de la capa de presentación, aportan gran valor al usuario y además aportan un elevado nivel de conocimiento del producto, mediante la reducción de la incertidumbre:

“The early development of parts of the user interface allows for the system to be shown to real or likely users in an early form. Feedback from these users will result in new knowledge about the product, and this knowledge can be used to make sure the team is developing the most valuable product possible”.

Por otro lado, con el objetivo de realizar un proyecto exitoso, que cumpla las expectativas por parte de los interesados, asegurar el alcance, la calidad, y una correcta gestión, se ha utilizado como base la guía PMBOK, editada por el PMI.

Queda fuera del alcance y propósito de este proyecto el seguimiento completo de todos los procesos que conforman un sistema de gestión tan completo como el que propone el PMI. Por tanto, se debe considerar este, más bien, como una guía de referencia, que ha guiado a todas las fases del proyecto, a la hora de crear un marco de gestión de calidad.

En lo relativo a la valoración económica, se llega con los clientes a un acuerdo, en virtud del cual, el primer módulo del desarrollo será desarrollado gratuitamente, dado que su finalidad no solo está dedicada a solucionar un problema práctico, sino también a servir de base para un TFG.

Este primer módulo desarrollado debe servir, además, como base para establecer las estimaciones y valoraciones de los otros dos módulos. Es decir, en base a las horas de desarrollo reales invertidas, se podrá hacer una valoración más precisa del importe y volumen exacto de trabajo a realizar en los módulos siguientes.

Quedan por tanto fuera del ámbito de este TFG los importes de las cuotas que sean definidas y pactadas con los clientes a posteriori.

A modo de resumen del capítulo, se añaden dos nuevos requisitos generales, en este caso de proceso, que se han seguido a lo largo del proyecto:

Requisito	Se usará metodología ágil y DCU como paradigma de desarrollo
Descripción	Se asegurará la correcta usabilidad, y funcionalidad del sistema a través de sprints y revisiones de usabilidad con los usuarios.
Tipo	Proceso.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

Requisito	Se usará la guía PMBOK como metodología de gestión del proyecto
Descripción	Se asegurará la correcta gestión del proyecto a través de algunos de los procesos y grupos propuestos por el PMI en el PMBOK
Tipo	Proceso.
Criterios de Aceptación	Se efectuarán pruebas piloto con los usuarios para comprobar que la experiencia de usuario es correcta, y el producto ofrece las funcionalidades más acordes con las necesidades de negocio.

3. Análisis y registro de los interesados

Tal y como recomienda el PMBOK, una de las primeras tareas a acometer ha sido la identificación de los *stakeholders* relacionados con el proyecto. La guía del PMI, incluye este proceso como uno de los dos que se deben llevar a cabo dentro de la fase de inicio de cualquier proyecto.

Una vez realizado un análisis cualitativo y cuantitativo, se detectan los siguientes interesados relacionados con el proyecto:

Nombre	Rol	Espectativas	Impacto	Acción
Cris M	Dependiente-propietaria	Facilidad de uso	Alto	Consultar
Sam E	Propietario	Aumento rentabilidad	Alto	Consultar
Oriol Martí	Tutor UOC	Gestión de calidad. Cumplimiento de las dimensiones del proyecto	Alto	Consultar
Marisa B	Comercial editorial	Información clara de las devoluciones	Medio-bajo	Informar

Ilustración 5. Registro de interesados

Para determinar el tipo de acción y comunicación con cada uno de ellos, se ha tenido en cuenta impacto que pueden tener en el proyecto, basado en los intereses y la influencia que podrían tener sobre el mismo.

En un primer momento, se intentó que las distribuidoras entrasen con un rol colaborativo y de soporte en el proyecto. Se intentó contactar con sus departamentos de administración y tecnologías para tratar de integrar de algún modo el sistema de devolución con sus sistemas de información. Debido al nivel de facturación relativamente bajo del negocio, en comparación con otros tipos de clientes, no se ha conseguido su involucración.

4. Gestión de riesgos

Durante la fase de planificación del proyecto, se ha realizado un análisis con la intención de identificar los posibles riesgos existentes. Durante este primer análisis han sido identificados 6 riesgos, y se ha realizado una propuesta de acciones destinadas a proteger el proyecto de posibles perjuicios.

Adicionalmente, tal y como recomiendan algunas metodologías de gestión de proyectos, se ha tratado de identificar también los riesgos positivos, es decir, acontecimientos variables que podrían mejorar en alguna de sus dimensiones el proyecto, en caso de aparecer y ser explotados.

El resultado de este análisis puede verse en el registro de riesgos que se muestra a continuación. Cabe destacar que este registro ha sido consultado a lo largo de toda la duración del proyecto, con la intención de tratar de mantener el control de esta área, pero también intentando identificar nuevos riesgos que debiesen incluirse.

Estas actualizaciones del registro quedan patentes en su séptima entrada, que ha sido incluida en la tercera fase, con la intención de mantener la calidad de los entregables de las primeras fases.

RIESGO	RESPUESTA	CAUSA/GRUPO
Pérdida código	Evitar - Se usará un repositorio de código distribuido (GitHub)	Avería
Pérdida documentación	Evitar - Se usará un sistema de gestión en la nube (Google Drive)	Avería
Enfermedad o indisponibilidad temporal	Mitigar - Se modificará la planificación, eliminando alguno de los módulos	Calendario/Recursos
Horas reales por encima de estimadas	Mitigar - Se revisará la planificación para reajustar las horas según los nuevos parámetros de estimación	Calendario
Horas reales por debajo de estimadas	Explotar - Se utilizarán las horas sobrantes en mejorar aspectos como el diseño gráfico	Calendario
Problemas configuración arquitectura entornos	Evitar - Se adelantan las tareas de configuración en la planificación.	Arquitectura
Entregables PEC no cumplen con las expectativas	Mitigar – Se realizará un análisis de las características y el valor que aportan, con objeto de alinear mejor los esfuerzos con el cumplimiento de las expectativas.	Calidad

Ilustración 6. Registro de riesgos

En dos ocasiones ha sido necesario acudir al registro para gestionar incidencias ocurridas. Estos casos, se explican brevemente a continuación, con la intención de ejemplificar la gestión de riesgos llevada a cabo a lo largo del proyecto.

7/10/2016 – Problemas de integración de la arquitectura. El sistema no funciona correctamente. Los contenidos y pantallas son defectuosos. Visitando la matriz de riesgos, se observa que hay tiempo como para tomar decisiones, y se decide eliminar el componente problemático *bootstrap*, y optar por un diseño más sencillo a través de JSF.

14/10/2016 – Avería grave en el equipo principal de desarrollo. El riesgo de pérdida de código se había eliminado con el uso de un repositorio distribuido (GitHub).

5. Análisis funcional

Partiendo de los requisitos generales, y de los objetivos definidos para el proyecto, se pasa a definir con mayor detalle la parte funcional del sistema. Siguiendo la metodología ágil, basada en un contacto fluido con el cliente, se programan una serie de reuniones semanales que deben servir para definir las funcionalidades del sistema, a partir de un listado de casos de uso.

La primera decisión importante que se toma, se centra en el hecho de dividir en tres módulos separados las funcionalidades del sistema. Además de encajar con la idea de valoración económica que se ha explicado previamente, se aprecia que los problemas a abordar pertenecen a 3 ámbitos bastante separados.

En este sentido, se podría haber considerado que la gestión de libros se podía incluir como un artículo más dentro de la gestión de stock, pero el cliente explica que se necesita un tratamiento muy particular de este tipo de libros, y se necesita una gestión diferenciada. Una vez revisado este punto se acuerda hacerlo de este modo debido al sistema de depósito, y al hecho de que se quieren manejar unos datos muy concretos de los libros.

Por otro lado, se decide dejar fuera del diseño funcional el requisito R-04: "Se tiene que poder generar informes y estadísticas de los procesos". Tras analizarlo junto con el cliente se llega a la conclusión de que se necesitaría un módulo completo que aportase más funcionalidades relacionadas con la inteligencia de negocio y la toma de decisiones estratégicas. Por este motivo, se decide no abordar por el momento esta área, pero se debe condicionar todo el diseño, de manera que se recojan datos de todos los módulos funcionales, de manera que podrían ser utilizados en el futuro.

En último lugar, se decide que se debe hacer un mayor hincapié en el diseño del módulo de gestión de libros, puesto que es el que mayor valor aporta al usuario en primera instancia. La problemática de la gestión de las devoluciones y control de la mercancía disponible es un problema acuciante, por lo que se prioriza este módulo y se decide que será el que se implemente en primer lugar.

Tal y como se ha venido haciendo a lo largo del proyecto, para simplificar tanto el análisis como la presentación, se ha dividido la presente sección entre los 3 módulos que conforman el sistema.

Los siguientes casos de uso se presentan en dos niveles de granularidad y detalle diferentes. Dado que el módulo de gestión de libros ha sido implementado, se ha considerado oportuno hacer un análisis más formal de los casos de uso, con un mayor detalle de los pasos que conforman un escenario de éxito, y posibles escenarios alternativos.

Por otro lado, los casos de uso relacionados con los módulos restantes, se documentan con un estilo menos formal, con un patrón similar al de las historias de usuario, pero con algunos detalles que ayuden a comprender la ejecución. Este sistema ha sido especialmente útil para refinar el diseño, encontrar inconsistencias, casos de uso que necesitaban ser redefinidos etc.

5.1 Análisis y casos de uso del módulo de gestión de libros

Durante las reuniones con los usuarios, y los análisis posteriores, se acuerda:

- Relacionar los libros por distribuidora. Este hecho resulta bastante evidente, puesto que una de las funcionalidades requeridas principales consiste en preparar las devoluciones por distribuidora. Para facilitar este hecho, se crean una serie de casos de uso para gestionar las distribuidoras, y se decide asociarlas a los libros en el momento del alta de estos.
- Se acuerda incluir la venta de libros como un artículo más. Durante las pruebas de usabilidad se descubre que puede ser común que un cliente compre al mismo tiempo un libro y cualquier otro tipo de producto, y puesto que no existe ningún gran impedimento o restricción que lo impida, se decide incluir la venta de libros dentro del CU_S7.
- Para respetar la integridad del sistema, el campo “distribuidora” de los libros, no será editable. Por este motivo se intentará crear un diseño que minimice los errores en la introducción de datos.

CU_L1		Dar de alta un libro	
Versión	1.0 (15/11/2016)		
Dependencias	CU_T3 - Identificarse en el sistema.		
	CU_L4 - Dar de alta una distribuidora.		
Precondición	El usuario se ha identificado correctamente en el sistema. Existe alguna distribuidora dada de alta en el sistema		
Descripción	El sistema registrará un nuevo libro, asociado a una determinada distribuidora.		
Secuencia normal	Paso	Acción	
	1	El usuario selecciona el módulo de gestión de libros.	
	2	El sistema muestra las opciones existentes del módulo.	
	3	El usuario selecciona alta de libro.	
	4	El sistema carga y muestra las distribuidoras existentes.	
	5	El usuario selecciona la distribuidora del libro.	
	6	El usuario introduce los datos del libro	
	7	El usuario selecciona guardar libro.	
	8	El sistema guarda el nuevo libro.	
	9	El sistema informa de que el libro se ha guardado correctamente.	
Post-condición	El usuario puede poner a la venta el libro y el sistema ha registrado todos los datos.		
Excepciones	Paso	Acción	
	8	El libro ya existe.	
	8a	El sistema aumenta el número de unidades	
Comentarios	La distribuidora se selecciona a través de una pestaña que no admite valores nulos, por lo que es necesario que la distribuidora exista antes de realizar el alta de un libro.		

CU_L2		Buscar un libro	
Versión	1.0 (15/11/2016)		
Dependencias	CU_T3 - Identificarse en el sistema.		
	CU_L4 - Dar de alta una distribuidora.		
	CU_L1 - Dar de alta un libro.		
Precondición	El usuario se ha identificado correctamente en el sistema.		
Descripción	El sistema retornará un listado de libros que coincidan con el criterio de búsqueda.		
Secuencia normal	Paso	Acción	
	1	El usuario selecciona el módulo de gestión de libros.	
	2	El sistema muestra las opciones existentes del módulo.	
	3	El usuario selecciona buscar un libro.	
	4	El sistema carga y muestra las opciones de búsqueda existentes.	
	5	El usuario rellena la cadena a buscar dentro de uno de los filtros existentes.	
	6	El sistema busca los libros que coinciden con el patrón de búsqueda.	
	7	El sistema lista todos los libros que coinciden con el patrón introducido.	
Post-condición	El usuario obtiene un listado de libros, y puede acceder al detalle del libro que buscaba.		
Excepciones	Paso	Acción	
	7	No existe ningún libro que cumpla con el criterio de búsqueda.	
	7a	El sistema muestra un mensaje indicando que no existen libros con el criterio de búsqueda	
Comentarios	En la presente implementación, solo se permite la búsqueda por un solo filtro.		

CU_L3		Editar un libro	
Versión	1.0 (15/11/2016)		
Dependencias	CU_T3 - Identificarse en el sistema.		
	CU_L4 - Dar de alta una distribuidora.		
	CU_L1 - Dar de alta un libro.		
	CU_L2 - Buscar un libro.		
Precondición	El usuario se ha identificado correctamente en el sistema. Existe algún libro dado de alta en el sistema.		
Descripción	El sistema mostrará los detalles asociados a un libro, permitiendo su modificación.		
Secuencia normal	Paso	Acción	
	1	Partiendo del último paso del CU_L2, el sistema muestra un listado de libros.	
	2	El usuario selecciona un libro de la lista.	
	3	El sistema muestra todos los campos asociados al libro seleccionado.	
	4	El usuario modifica los campos deseados.	
	5	El usuario selecciona guarda.	
	6	El sistema modifica y actualiza los datos asociados al libro.	
	7	El sistema muestra un mensaje indicando que la operación se ha realizado exitosamente.	
Post-condición	El libro queda actualizable y será consultable a través de los nuevos datos con los que ha sido modificado.		
Excepciones	Paso	Acción	
	-	N/A	
Comentarios	En sistema precarga los datos asociados al libro, por lo que aunque no se modifiquen se produce actualización con los mismos datos.		

Versión	1.0 (15/11/2016)	
Dependencias	CU_T3 - Identificarse en el sistema.	
Precondición	El usuario se ha identificado correctamente en el sistema.	
Descripción	El sistema registrará una nueva distribuidora, a la cual se le podrán vincular los libros en depósito.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona el módulo de gestión de libros.
	2	El sistema muestra las opciones existentes del módulo.
	3	El usuario selecciona alta de distribuidora.
	4	El usuario introduce los datos de la distribuidora.
	5	El usuario selecciona guardar distribuidora
	6	El sistema guarda la nueva distribuidora.
	7	El sistema informa de que la distribuidora se ha guardado correctamente.
Post-condición	Se ha creado la nueva distribuidora. El usuario ya puede dar de alta libros mediante el CU_L1 asociados a la nueva distribuidora.	
Excepciones	Paso	Acción
	6	Ya existe una distribuidora con ese CIF.
	8a	El sistema informa de que no es posible guardar la nueva distribuidora.
Comentarios	La clave que diferenciará a las distribuidoras es el CIF.	

Versión	1.0 (15/11/2016)	
Dependencias	CU_T3 - Identificarse en el sistema.	
	CU_L4 - Dar de alta una distribuidora.	
	CU_L1 - Dar de alta un libro.	
Precondición	El usuario se ha identificado correctamente en el sistema. Existen libros dados de alta en el sistema con más de una unidad.	
Descripción	El sistema listará los posibles libros de devolución para una determinada distribuidora, y guardará un listado con aquellos que se marquen para devolver.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona el módulo de gestión de libros.
	2	El sistema muestra las opciones existentes del módulo.
	3	El usuario selecciona preparar una devolución.
	4	El usuario selecciona una distribuidora.
	5	El sistema muestra un listado de todos los libros relacionados con la distribuidora que tienen al menos una unidad,
	6	El usuario selecciona los libros que se van a devolver.
	8	El sistema guarda el listado y marca como retornados los libros indicados
	9	El sistema informa de que se ha actualizado el listado.
Post-condición	Los libros quedan marcados como retornados. El usuario puede proceder a la devolución física.	
Excepciones	Paso	Acción
	-	N/A
Comentarios	En el momento de guardar el listado, se considerarán retornados todos los libros seleccionados. Si se quiere añadir o eliminar alguno se puede hacer desde la pantalla de editar libro.	

5.2 Diagrama de clases del módulo de gestión de libros

El siguiente diagrama muestra una versión completa del modelo de información diseñado a partir del análisis de los casos de uso que conforman el presente módulo.

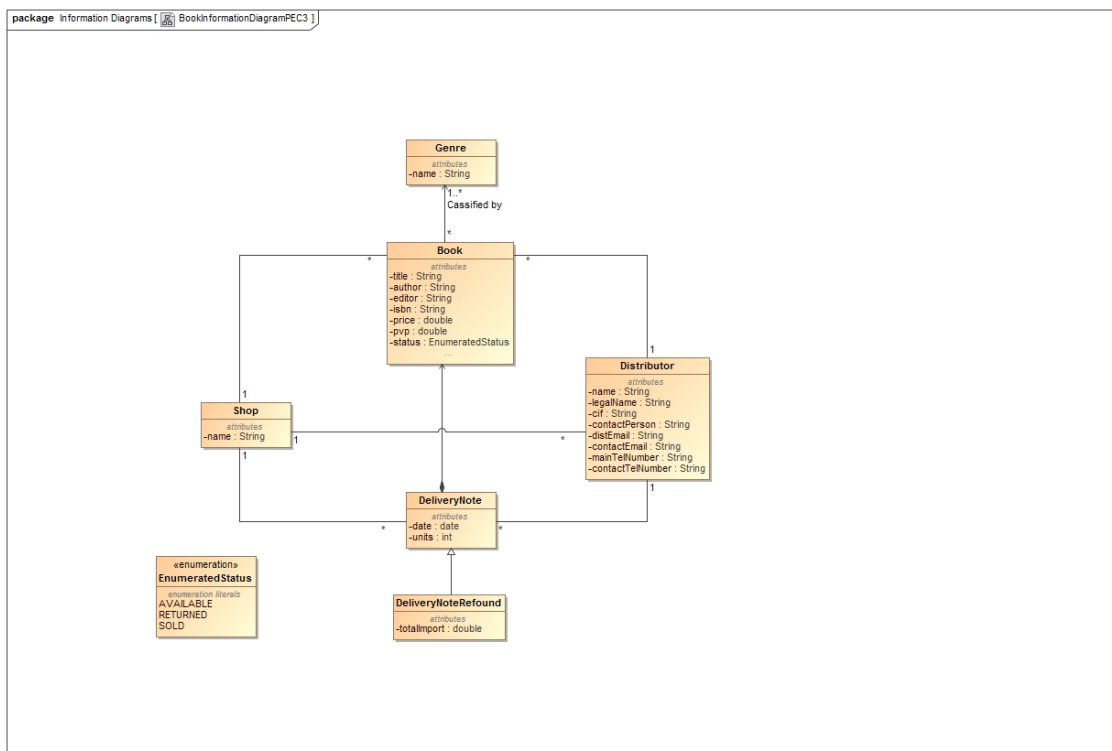


Ilustración 7. Diagrama de clases módulo gestión libros

- La clase enumerativa “*EnumeratedStatus*” debe servir para definir la trazabilidad de los libros en el sistema.
- Las altas de libros quedarán registradas en la clase “*DeliveryNote*”. Esta clase representará una entrega de libros, de una distribuidora, en una fecha concreta.
- La sub-clase “*DeliveryNoteRefound*” representa un albarán de devolución. Se generará automáticamente como resultado del caso de uso CU_L7. En esta clase se debe recoger, además de los datos de un albarán estándar, el total del importe a reembolsar, siendo este variable en función de lo negociado con cada distribuidora.

5.3 Análisis y casos de uso del módulo de gestión de stock

A lo largo de la evolución del proyecto, el módulo correspondiente a la gestión del stock ha sido modificado, sobre todo en la definición de los casos de uso. Las decisiones más importantes tomadas en esta área han sido:

- Todos los artículos se asociaran a una o varias familias. Este concepto se incluye deliberadamente como clase, puesto que las conversaciones con el cliente dejan entrever que es susceptible de ser modificado y ampliado con más datos, y por tanto es mejor plantearlo desde el principio como una clase independiente.
- Dada la necesidad de dar salida a ciertas familias o tipos de artículos, se crean una serie de casos de uso que permitan crear promociones.
- Se considera la venta como un movimiento más dentro de la gestión de stock. Por este motivo se incluye dentro de este módulo. Además al existir promociones que incluyen combinaciones de artículos, y por tanto, se deben calcular en el momento de la venta, se hará todo el tratamiento de promociones dentro del módulo de gestión del stock.
- Se ha modificado el caso de uso CU_S3: "Editar datos de una familia". Por un lado, se añade el caso de uso "Consultar datos de una familia", ya que este puede ser independiente. Por otro lado, se pueden dar dos opciones de edición muy distintas, por lo que se divide en dos casos de uso distintos, CU_S12: "Editar nombre de una familia" u CU_S13: "Eliminar artículos asociados a una familia".
- Se modifica el caso de uso CU_S11: "Reponer artículos" por CU_S11: "Entrada de artículos". Este caso de uso permitirá gestionar todas las compras a proveedores, almacenando información de las compras (fecha, artículos, proveedor, precio de cada artículo).
- Se mostrarán las promociones directamente por detalle, sin listado. No se prevé que haya muchas en al mismo tiempo y tampoco se desea conservar un histórico, puesto que quedará reflejado en las ventas. Por este motivo se modifica el nombre del caso de uso de CU_S9: "Listar promociones" a CU_S9: "Consultar promociones".
- Dada la intención de generar datos para un futuro sistema de toma de decisiones estratégicas, se decide incluir en el diseño información adicional de las ventas. Se deberá matizar esta información, por lo que provisionalmente se permite un campo de texto libre.

CU_S1: DAR DE ALTA UNA FAMILIA DE ARTICULOS:

El usuario quiere relacionar una serie de artículos mediante un tipo común, para poder organizar tanto los artículos, como los datos que estos generen. Para ello crea una nueva familia a la cual se podrán añadir los nuevos artículos que se vayan dando de alta.

CU_S2: LISTAR FAMILIAS DE ARTÍCULOS:

El usuario quiere poder consultar el listado de las familias existentes en el sistema para revisar la catalogación. El usuario elige la opción de listar familias, y obtiene un listado con todas las familias existentes en el sistema.

CU_S3: CONSULTAR DATOS DE UNA FAMILIA:

El usuario quiere consultar el nombre y artículos asociados a una familia para analizar la catalogación de artículos en el sistema. El usuario elige una familia del listado, y el sistema le muestra el nombre y los artículos asociados a ella.

CU_S12: EDITAR NOMBRE DE UNA FAMILIA:

El usuario quiere editar el nombre con el que está registrada en el sistema una familia para mejorar la categorización de artículos. Para conseguirlo, el usuario elige una familia del listado de familias, modifica su nombre en el detalle de la familia y guarda los datos en el sistema. La familia queda actualizada en el sistema y se mostrará con el nuevo nombre en todos los casos.

CU_S13: ELIMINAR ARTICULOS ASOCIADOS A UNA FAMILIA:

El usuario quiere eliminar uno o varios artículos del listado de artículos asociados a una determinada familia, para modificar la catalogación de ítems en el sistema. Para ello, el usuario, elige una familia del listado, consulta la lista e artículos que tiene asociados, y elimina aquellos que considera incorrectos. El sistema actualiza la relación entre familias y artículos.

CU_S4: DAR DE ALTA UN ARTÍCULO:

El usuario desea dar de alta en el sistema un nuevo artículo que se va a comercializar para que quede registrado en el sistema. Para lograrlo, el usuario introduce todos los datos asociados al artículo. El sistema almacena el nuevo artículo.

CU_S5: BUSCAR UN ARTÍCULO:

El usuario quiere consultar los datos asociados a un artículo, para analizarlos, modificarlos, o como parte del proceso de venta (CU_S7). Para ello, el usuario deberá escoger la opción buscar artículo del módulo de gestión de stock, y posteriormente introducir el nombre o el código de barras del artículo. El sistema retornará un listado con aquellos artículos que cumplan con los criterios de búsqueda. El usuario seleccionará de la lista el artículo deseado, y el sistema mostrará el detalle, o lo cargará en la lista de ventas si se trata de un proceso de venta.

CU_S6: EDITAR UN ARTÍCULO:

El usuario quiere cambiar alguno de los datos asociados a un determinado artículo. Para realizar la modificación, el usuario buscará un artículo (CU_S5), modificará los datos necesarios y aceptará los cambios. El sistema actualizará los datos asociados al artículo, y este aparecerá modificado para todos los casos.

CU_S7: REALIZAR UNA VENTA:

El cliente llega a la caja con uno o varios artículos que quiere comprar. El usuario buscar en el sistema los artículos, y los añade a la lista de artículos de la venta. El sistema calcula si hay que aplicar algún descuento promocional y el precio total. El usuario cobra el importe total y cierra la venta. El sistema actualiza las unidades de los artículos vendidos y almacena los datos de la venta.

CU_S8: CREAR UNA PROMOCIÓN:

El usuario quiere aplicar un descuento promocional en las ventas para ciertos artículos con el objeto de mejorar las ventas. Para ello, el usuario selecciona dentro del menú de promociones la familia, el artículo, combinación de un mismo artículo, o combinación de diferentes artículos, el porcentaje de descuento que desea aplicar al artículo o pack, y las fechas de comienzo y fin de la promoción. El sistema almacena estos datos, y cuando se realiza un venta, analiza si los artículos que conforman la misma coinciden con alguna de las promociones en vigor.

CU_S9: CONSULTAR PROMOCIONES:

El usuario quiere consultar las promociones que existen en el sistema para su consulta o modificación. Para ello selecciona la opción promociones del modulo de gestión de stock. El sistema va mostrando una a una las promociones, y el usuario va pasando de una a otra, hasta llegar a la deseada.

CU_S10: EDITAR UNA PROMOCIÓN:

El usuario quiere modificar alguno de los datos que forman una promoción para optimizar la oferta comercial. El usuario deberá consultar la promoción que quiera modificar (CU_S9), modificará la combinación de artículos, las fechas o el porcentaje de descuento y aceptará los cambios. El sistema actualizará la promoción, que será aplicada desde ese momento entre el rango de fechas escogido.

CU_S11: ENTRADA DE ARTÍCULOS:

El usuario quiere dar de alta en el sistema una serie de artículos que ha comprado a un proveedor, para poder gestionarlos y venderlos posteriormente. El usuario introduce los artículos de entrada, que previamente han sido dados de alta, el nombre del proveedor y las unidades. El sistema actualiza el número de unidades de los artículos, que podrán ser editados y vendidos desde ese momento.

5.4 Diagrama de clases del módulo de gestión de stock

El siguiente diagrama muestra una versión completa del modelo de información diseñado a partir del análisis de los casos de uso que conforman el presente módulo.

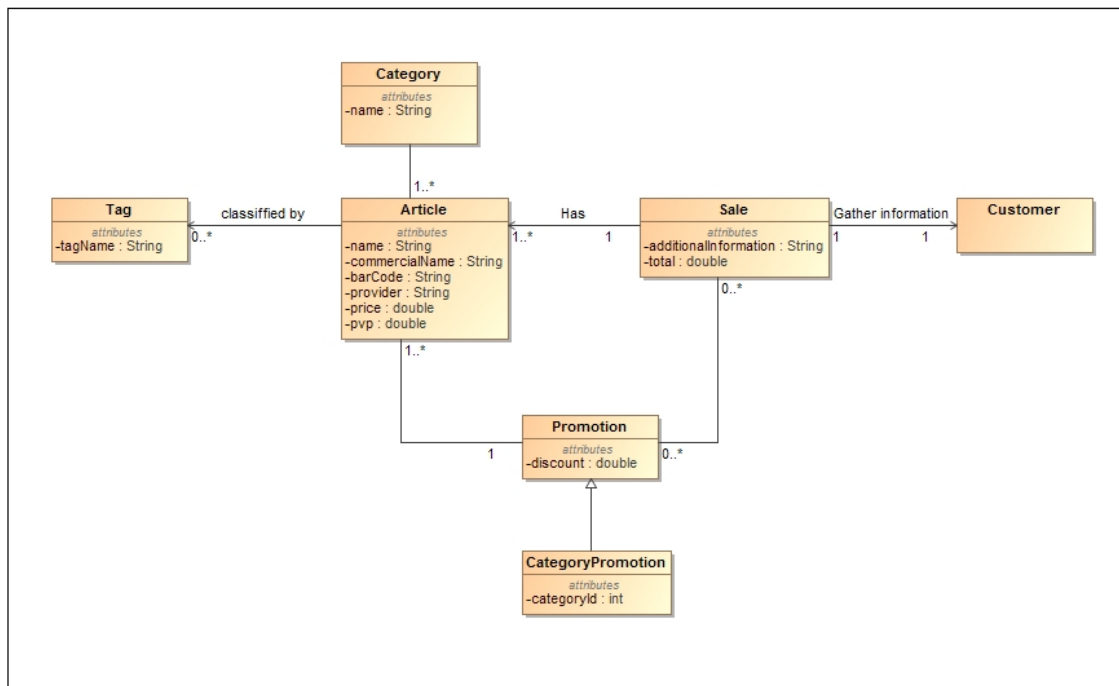


Ilustración 8. Diagrama de clases módulo gestión stock

Se ha excluido del diseño una clase que represente a los proveedores. Se ha consultado con los usuarios pero por el momento no se considera necesario porque son compras muy aleatorias y diferentes, por lo que se plantea como mejora para una posible implementación de análisis de negocio en una futura implementación. De momento se almacena simplemente el nombre como atributo, por si en el futuro fuese necesario recuperar el dato.

Se representan de manera diferenciada los dos tipos de promociones, por un lado, aquellas que incluyen una combinación de artículos, y por otro lado aquellas que se aplican a todos los artículos de una determinada familia. De este modo se consigue una representación y gestión más clara de ambos tipos dentro del sistema, y un mapeo relacional más sencillo.

5.5 Análisis y casos de uso del módulo de gestión de eventos

A lo largo de las fases del proyecto, el módulo correspondiente a la gestión de eventos también ha sido modificado en la definición de los casos de uso. Las decisiones más importantes tomadas en esta área han sido:

- Se ha considerado que los eventos tienen funcionalmente un significado y valor propios por lo que se ha creado una clase-entidad en el sistema para almacenar esta información. Este hecho queda reflejado en primer lugar en el CU_S5: “Reservar sala” que ahora pasa a ser CU_S5: “Crear evento”.
- En segundo lugar, se ha considerado necesario una serie de casos de uso que permitan buscar, consultar, editar y eliminar un evento existente. Por este motivo se han creado los casos de uso CU_E9, CU_E10, CU_E11 y CU_E12.
- Los casos de uso CU_E6 y CU_E7, se unifican en el CU_E7, puesto que ni desde el punto de vista práctico ni funcional tiene sentido consultar por separado la disponibilidad y la ocupación de una sala.
- Dada la relevancia que tienen las exposiciones audiovisuales para atraer visitantes, se condiciona el diseño de modo que incluya características que permitan facilitar esta gestión. De este modo, en primer lugar se crea una clase propia para definir a los artistas audiovisuales. De nuevo, en esta decisión pesa el hecho de que posiblemente se vayan ampliando los datos que diferencien a esta subclase. Además se crea el caso de uso CU_E8, que debería permitir una mejor gestión de los espacios audiovisuales.

CU_E1: REGISTRAR LOS DATOS DE UN ARTISTA:

El usuario quiere guardar la información, fechas de disponibilidad, y datos de contacto de un determinado artista en el sistema, para poder contactar posteriormente con él, y gestionar eventos. El usuario selecciona la opción de añadir artista dentro del módulo de gestión de eventos, rellena los campos con la información asociada al artista. El sistema guarda el nuevo artista, que podrá ser consultado/editado a partir de ese momento.

CU_E2: BUSCAR UN ARTISTA:

El usuario quiere buscar un artista determinado de los registrados en el sistema, para poder consultar o editar sus datos asociados. Para ello, selecciona la opción buscar artista, introduce total o parcialmente el apellido del artista, y el sistema le muestra a continuación el listado de artistas registrados que coinciden con el criterio de búsqueda.

CU_E3: CONSULTAR LOS DATOS DE UN ARTISTA:

El usuario quiere consultar los datos asociados a un artista. El usuario deberá buscar un artista (CU_E2) seleccionar alguno de los resultados retornados por el sistema, y el sistema le mostrará los detalles del artista.

CU_E4: EDITAR LOS DATOS DE UN ARTISTA:

El usuario quiere editar los datos asociados a un artista en concreto, para modificar su contacto, fechas de disponibilidad o corregir alguno de los datos personales. Para llevar a cabo esta acción, el usuario debe previamente consultar los datos de un artista (CU_E3), cambiar cualquiera de los datos que desee modificar, y aceptar los cambios. El sistema actualizará los datos y los mostrará modificados desde ese momento.

CU_E5: CREAR UN EVENTO:

El usuario desea crear un nuevo evento, que consistirá en una exposición fija o audiovisual de un determinado artista, y en que se expondrá en una de las salas del negocio entre unas fechas determinadas. Para llevar a cabo esta actividad, el usuario deberá seleccionar la opción nuevo evento, dentro del módulo de gestión de eventos. Una vez introducida toda la información relacionada con el evento intentará crearlo. El sistema validará que la sala elegida tenga disponibilidad en el rango de fechas seleccionado, y guardará el nuevo evento, actualizando la disponibilidad de salas, o avisará al usuario de que existe un conflicto de fechas.

CU_E6: CONSULTAR LA DISPONIBILIDAD DE UNA SALA:

Este caso de uso deja de tener sentido, por lo que se elimina de la implementación y se pasa a usar siempre el CU_E7.

CU_E7: CONSULTAR LA OCUPACION DE UNA SALA:

El usuario quiere saber la utilización que hay prevista para una sala en unas determinadas fechas, para poder optimizar el uso de los espacios. El usuario deberá seleccionar la opción de consulta de ocupación por sala, e indicar la sala que se desea consultar y el rango de fechas a consultar. El sistema mostrará un listado de los eventos asociados a dichas sala dentro del rango de fechas.

CU_E8: LISTAR ARTISTAS AUDIOVISUALES:

El usuario quiere consultar los artistas audiovisuales que tienen disponibilidad en una determinada fecha, con el objeto de conseguir la mayor explotación posible de la sala preparada para este tipo de eventos. Para ello, el usuario selecciona la opción listar artistas audiovisuales, y las fechas de referencia. El sistema retornará un listado de aquellos artistas audiovisuales para los que se ha registrado disponibilidad en las fechas solicitadas.

CU_E9: BUSCAR UN EVENTO:

El usuario desea buscar un evento creado en el sistema para poder consultarlo o editarlo. Para buscarlo, selecciona la opción buscar evento, e introduce alguno de los criterios de búsqueda (rango de fechas, sala, nombre del evento o apellido del artista). El sistema retorna un listado de aquellos eventos que cumplen con los criterios de búsqueda.

CU_E10: CONSULTAR UN EVENTO:

El usuario quiere consultar todos los datos asociados a un evento, para poder analizarlo o modificarlo. Para ello, buscará el evento (CU_E9), y seleccionará el deseado entre el listado de resultados retornado por el sistema. El sistema mostrará todos los datos asociados el evento seleccionado.

CU_E11: EDITAR UN EVENTO:

El usuario quiere modificar alguno de los datos asociados a un evento. El usuario deberá consultar el evento (CU_E10), modificar los datos que considere oportunos, y aceptar los cambios. El sistema actualizará los datos del evento, y los mostrará modificados desde ese momento.

CU_E12: ELIMINAR UN EVENTO:

El usuario necesita eliminar un evento del sistema, que previamente haya sido dado de alta. Para ello deberá consultar un evento (CU_E10) y seleccionar la opción de eliminación. El sistema eliminará el evento y ya podrá ser utilizado ni consultado desde ese momento.

5.6 Diagrama de clases del módulo de gestión de eventos

El siguiente diagrama muestra una versión completa del modelo de información diseñado a partir del análisis de los casos de uso que conforman el presente módulo.

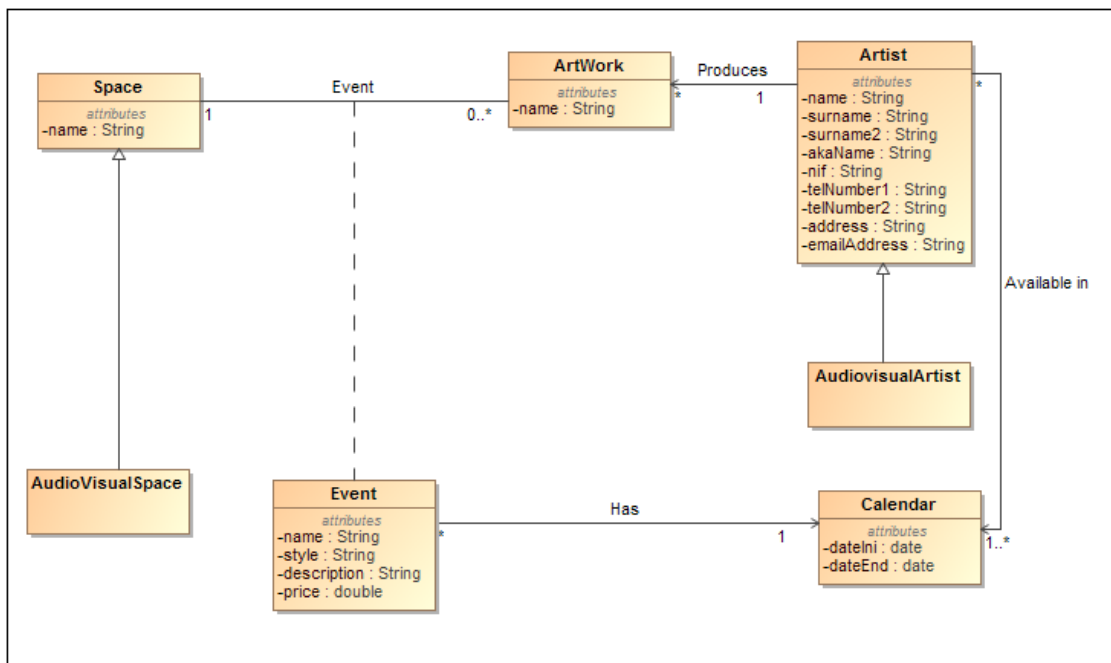


Ilustración 9. Diagrama de clases módulo gestión eventos

5.7 Diagramas de secuencia

Debido a la complejidad funcional de algunos de los casos de uso analizados, se incluyen una serie de diagramas de secuencia, que se han utilizado como puente entre el análisis y el diseño técnico. Estos diagramas han sido especialmente útiles para definir la mensajería entre capas, los métodos auxiliares necesarios, y las responsabilidades de cada una de las capas.

Se han escogido 2 casos de uso del módulo de gestión de libros para ser representados y matizados mediante un diagrama de secuencia. En primer lugar, el caso de uso CU_L1 “Dar de alta un libro”, no presenta una dificultad especial, pero sirve a modo de ejemplo de cómo se implementan este tipo de casos de uso, puesto que se incluyen varios casos de uso similares en el proyecto. Lo más destacable de estos casos de uso es el paso de datos entre capas encapsulado dentro de una clase que actúa como *bean*. Estas clases son las mismas que se usan en el modelo relacional, mediante JPA y representan las entidades en el sistema. Se considera más correcto el paso de datos entre capas dentro de estas clases, que el uso de funciones con numerosos parámetros a través de un desglose de los atributos de las entidades. Por tanto, una vez que se recogen los datos de las pantallas en el *Mangad Bean*, se instancia una entidad que será usada a través de la capa de negocio hasta la capa de persistencia.

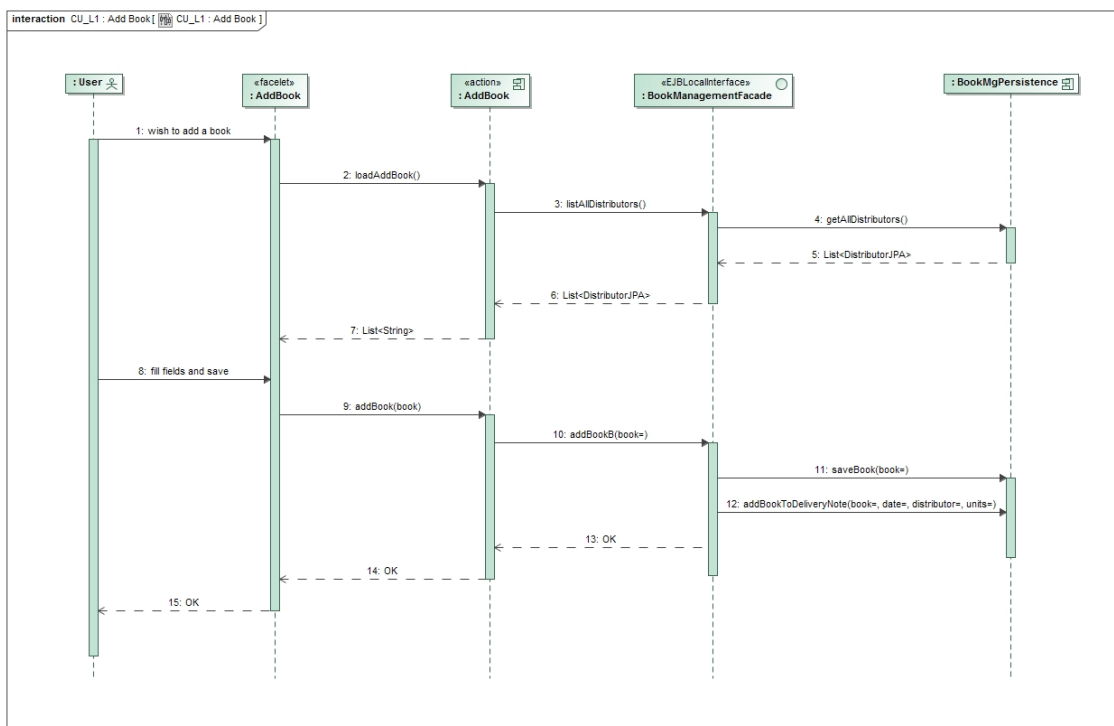


Ilustración 10. Diagrama de secuencia CU_L1

En segundo lugar, se ha seleccionado el caso de uso CU_L7 “Realizar una devolución”, puesto que se trata de una operativa algo compleja, y el diagrama ayuda a entender el modo en que ha sido implementada.

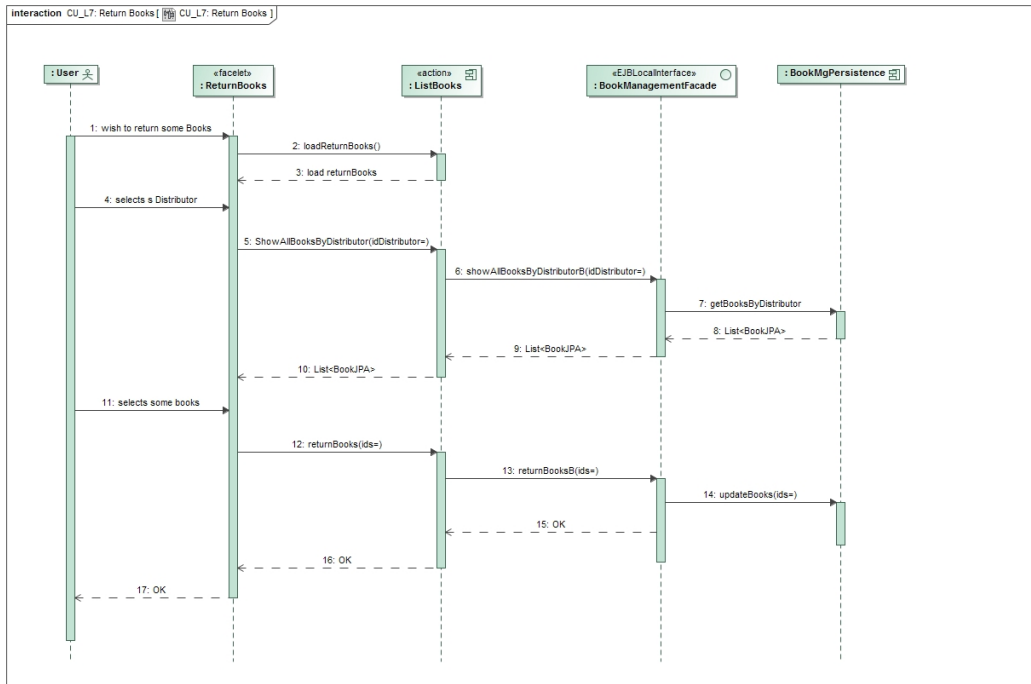


Ilustración 11. Diagrama de secuencia CU_L7

Referente al módulo de gestión del stock, se ha escogido el caso de uso CU_S7: “Realizar una venta” por su alta complejidad. Para implementar este caso de uso, es necesario realizar una búsqueda de los artículos en bucle, calcular las posibles promociones, actualizar el stock y por último grabar los datos de la venta.

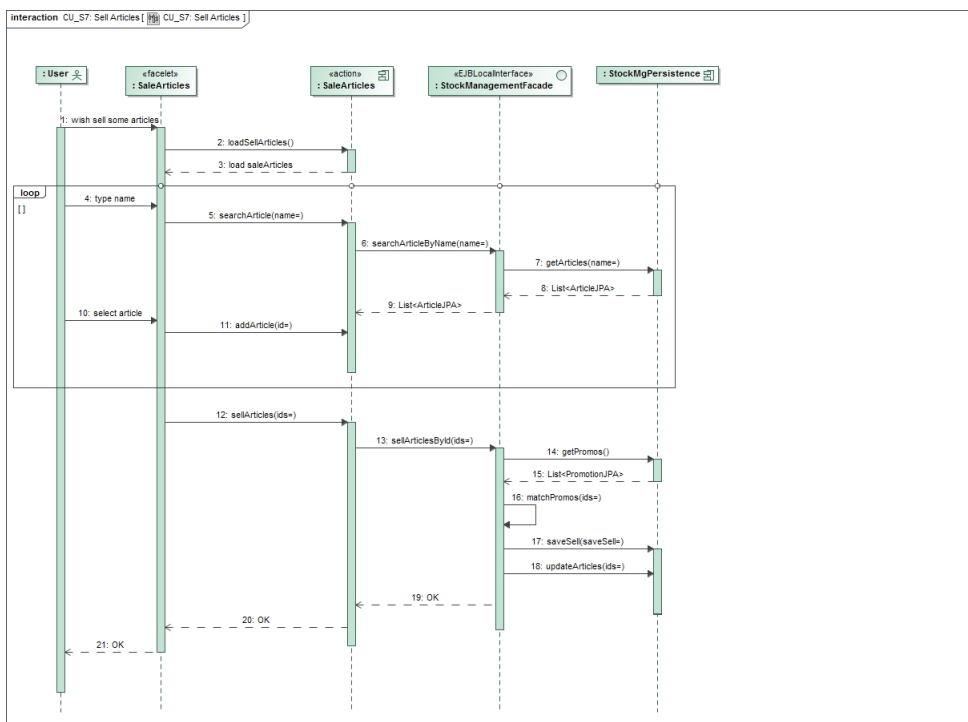


Ilustración 12. Diagrama de secuencia CU_S7

Por último, se ha escogido el caso de uso CU_E5: “Crear un evento”, por ser el único del módulo que presenta una dificultad más elevada. En este caso se

debe consultar la disponibilidad de las salas y ofrecer diferente respuesta en caso de se pueda crear el evento o no.

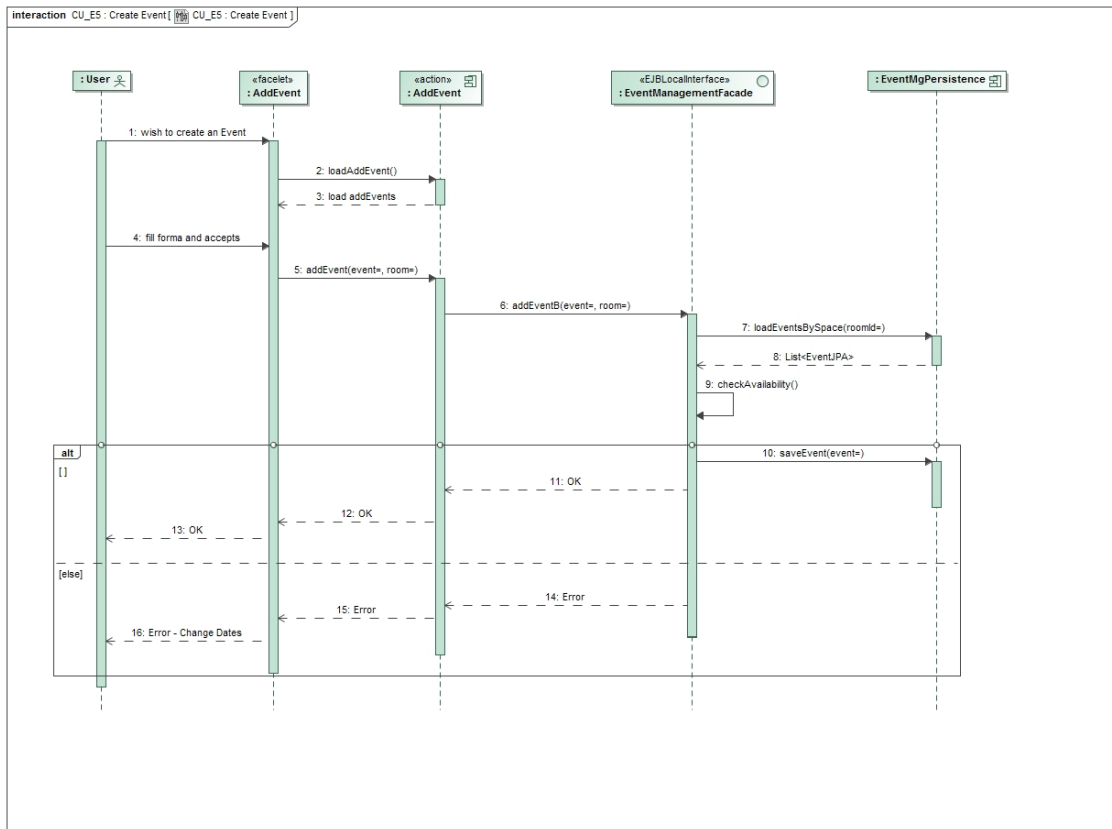


Ilustración 13. Diagrama de secuencia CU_E5

5.8 Módulo de administración

Adicionalmente, existen una serie de casos de uso transversales, que no se pueden ubicar en un módulo en concreto, pero son necesarios para el correcto funcionamiento del sistema. Aunque no se pretende implementarlos en esta fase del proyecto, se considera necesaria su descripción.

Se describen brevemente, a continuación, los casos de uso de este tipo que se han detectado.

CU_T1: DAR DE ALTA UN USUARIO:

El administrador quiere dar de alta un usuario en el sistema, para que tenga permisos para efectuar las acciones asignadas a estos. El administrador se identifica como tal en el sistema, accede al área restringida de administración, y añade un nuevo usuario y contraseña en el sistema. El usuario quedará registrado en el sistema con las credenciales de acceso introducidas, y podrá identificarse como tal desde ese momento.

CU_T2: MODIFICAR UN USUARIO:

El administrador quiere modificar los datos asociados a un usuario, para modificar sus credenciales o deshabilitarlo. Para ello, desde el menú de administración selecciona el usuario a modificar, realiza los cambios deseados y acepta los cambios. Desde ese momento el usuario queda modificado en el sistema.

CU_T3: IDENTIFICARSE EN EL SISTEMA:

El usuario quiere identificarse en sistema para efectuar cualquiera de las acciones descritas en cada uno de los módulos. Para ello, al iniciar el sistema deberá introducir las credenciales con las que ha sido registrado previamente.

CU_T4: CERRAR SESIÓN:

El usuario desea cerrar la sesión de trabajo, para evitar que puedan hacer uso inapropiado de ella. Para conseguirlo selecciona la opción de salir del sistema, y este cierra la sesión. Para poder efectuar cualquier opción, a partir de ese momento, será necesario volver a identificarse en el sistema (CU_T3).

CU_T5: CREAR RESPALDO DEL SISTEMA:

El administrador quiere tener una copia de respaldo distribuida en la nube con la intención de recuperar el sistema en caso de contingencia. Para ello, el sistema periódicamente envía una copia cifrada de los datos a un servidor en la nube, y un email con la confirmación al administrador.

6. Diseño técnico

6.1 Arquitectura del sistema

En la fase de definición del proyecto se decidió la utilización de Java EE, en combinación con los *frameworks* JPA y JSF. Se trata por tanto de una arquitectura claramente definida en 3 capas: Presentación con JSF, negocio con EJB y persistencia mediante JPA.

Además, adicionalmente, en un primer diseño se había decidido utilizar *bootstrap* para generar las plantillas de la capa de presentación. Tal y como se recoge en el apartado de riesgos, la integración entre JSF y *bootstrap* resultó desde el principio demasiado compleja, por lo que se modificó a JSF con CSS puro, y alguna librería de *javascript* para mejorar la funcionalidad.

Toda la aplicación ha sido definida para ejecutarse en un servidor de aplicaciones de tipo *Jboss 7*.

Con este diseño de arquitectura, se permite una implementación sin acoplamiento entre capas y una correcta integración. Además, se deja abierta la opción a la modificación y ampliación gracias a la flexibilidad de Java. Se ha tenido en cuenta especialmente la posibilidad de distribuir en el futuro la capa de presentación, mediante un cliente ligero, pudiendo ejecutarse en dispositivos móviles, para facilitar la ejecución de determinadas funcionalidades.

En este sentido, cabe destacar que todo el diseño de los componentes se ha realizado teniendo en cuenta los principios de diseño, y concretamente, dada la naturaleza del producto, el principio de abierto/cerrado, de cara a posibles extensiones de las funcionalidades.

6.2 Diagramas de arquitectura de la capa de presentación

La capa de presentación ha sido diseñada siguiendo un claro patrón de tipo MVC, utilizando los tipos de componentes que ofrece el *framework* JSF. Se

presenta en los diagramas, a modo de interfaz, las acciones a realizar por los usuarios, y tienen, por tanto, una relación 1 a 1 con los casos de uso definidos previamente.

El método elegido para diseñar la capa técnica ha sido pensado en todo momento desde el punto de vista de los usuarios, las necesidades que se debían cubrir, especificadas a modo de casos de uso, y las interacciones necesarias entre usuarios y sistema.

Por este motivo se ha comenzado el diseño pensando en las vistas, la información que debían contener y las acciones que debían permitir para cubrir estas necesidades. A continuación se han confeccionado las clases que debían actuar a modo de soporte y control de las vistas, para permitir la navegación y la integración con la capa de negocio.

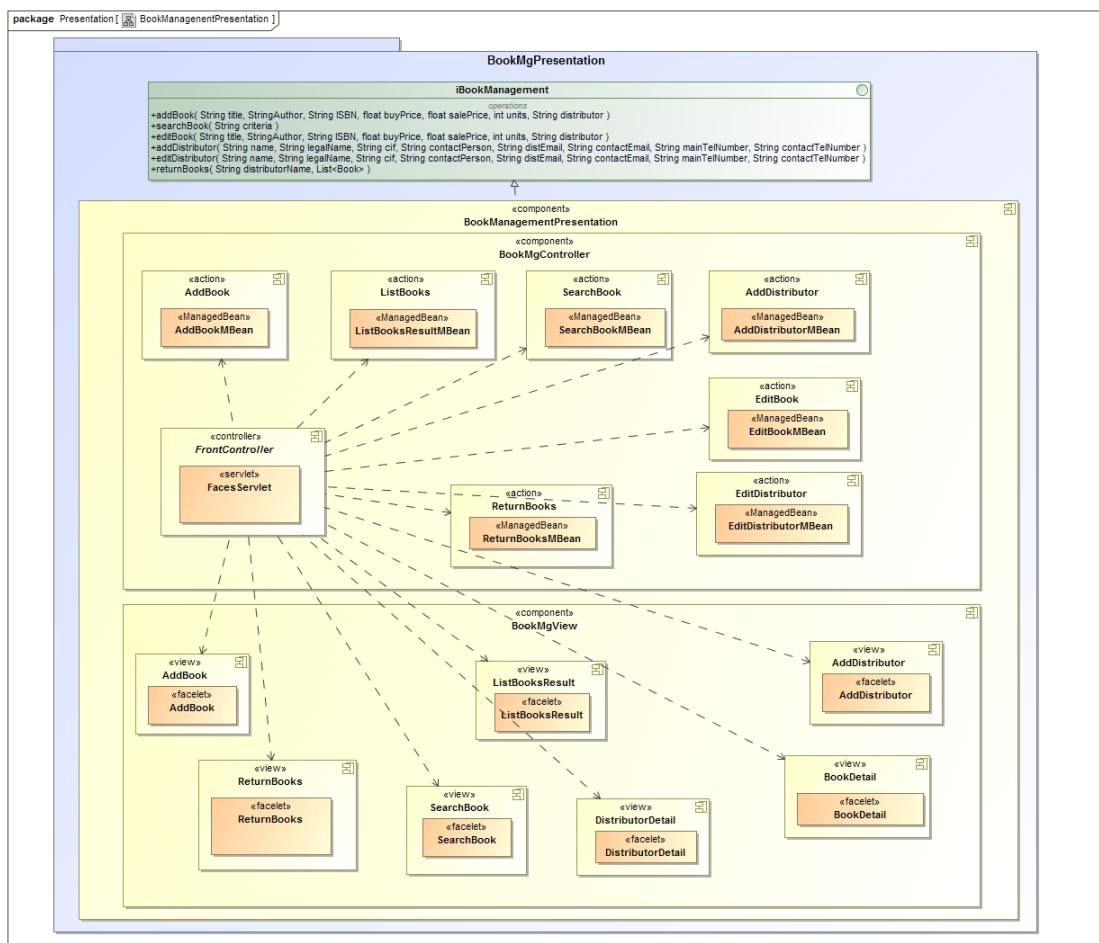


Ilustración 14. Diagrama capa presentación módulo gestión libros

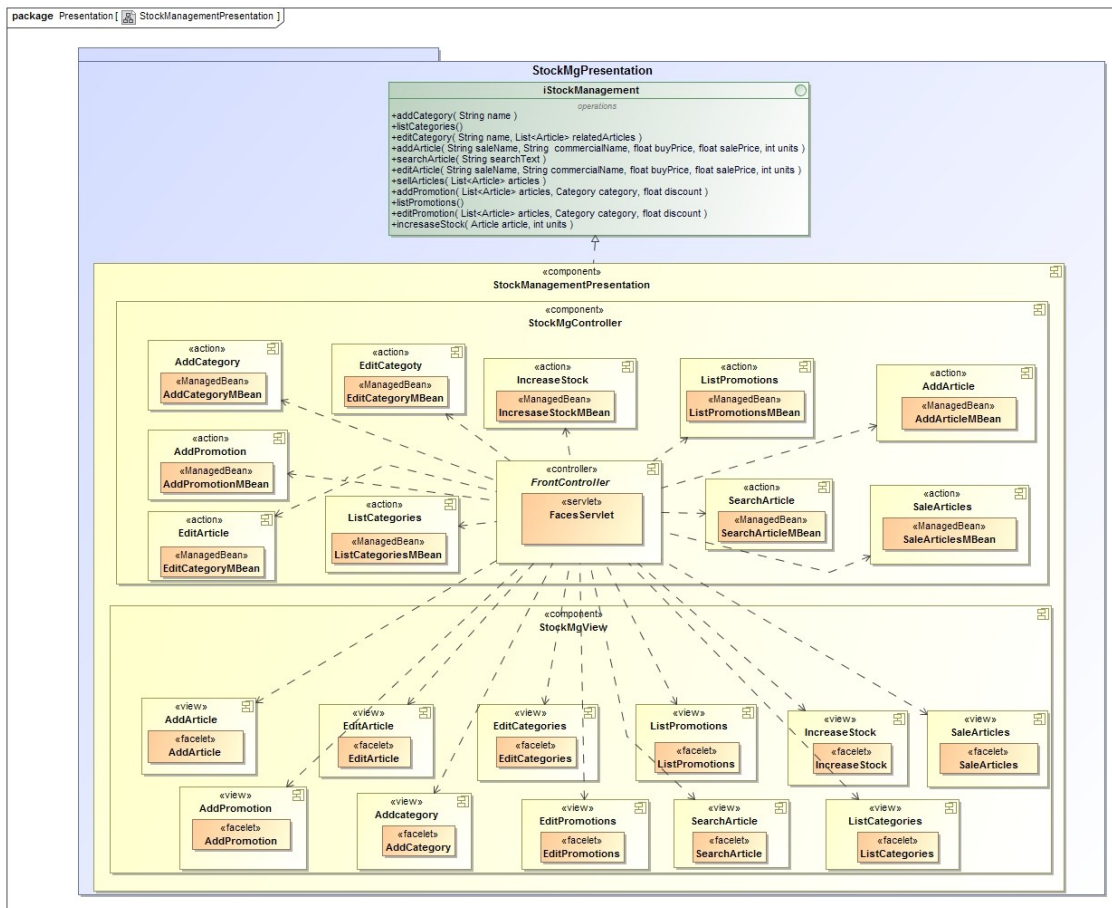


Ilustración 15. Diagrama capa presentación módulo gestión stock

El módulo de gestión del stock, ha sido el más complejo de diseñar, debido a la cantidad de casos de uso que presenta, y al hecho de que algunos de ellos necesitan varias vistas y clases de soporte.

Para reducir esta complejidad, en la implementación del diseño se ha intentado utilizar al máximo la modularidad, de manera que cada funcionalidad queda desligada del resto, reduciendo el acoplamiento entre clases.

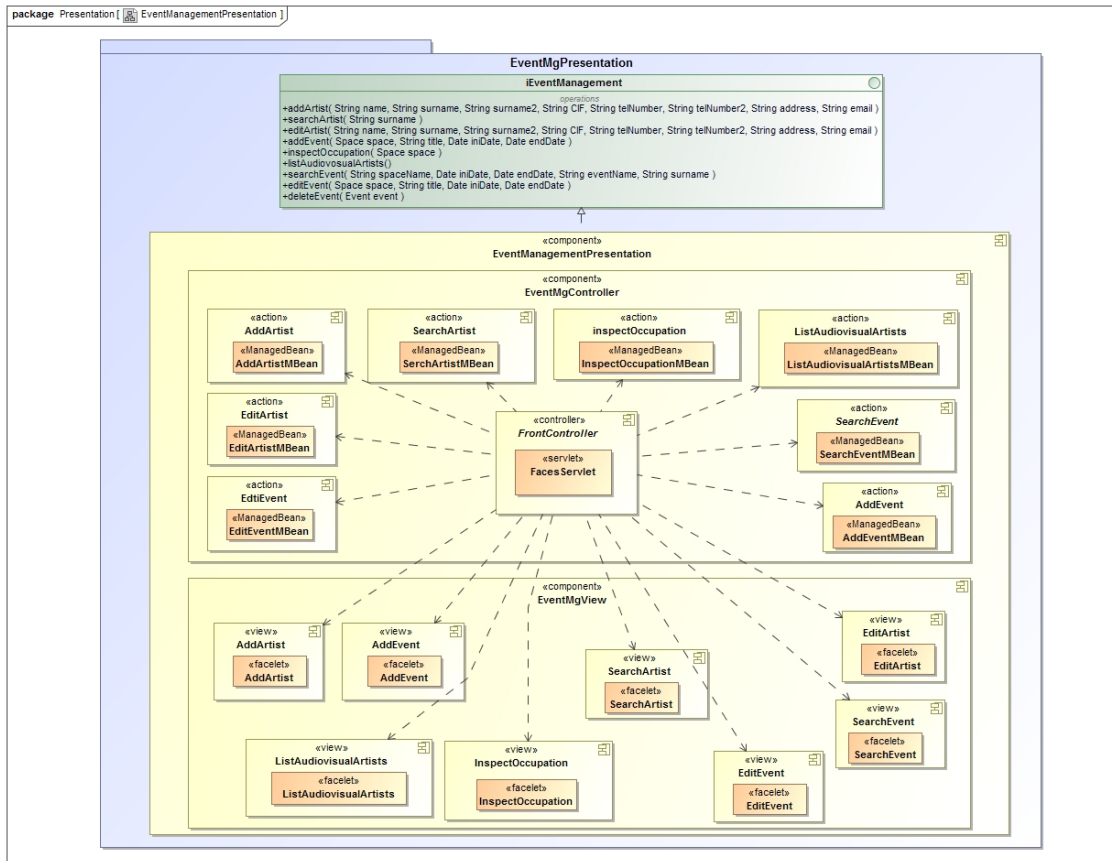


Ilustración 16. Diagrama capa presentación módulo gestión eventos

La capa de presentación del módulo de gestión de eventos no presenta una gran complejidad en cuanto al diseño, pero sí en las vistas. Algunas de las funcionalidades requeridas, como la documentada en el CU_E7: consultar la ocupación de una sala, requieren gráficos algo más complejos para facilitar la usabilidad. En estos casos se ha tratado de trabajar con datos sencillos, y trasladar la complejidad exclusivamente a la vista, donde se deberán de montar los gráficos complejos en base a los datos recibidos.

6.3 Diagramas de arquitectura de la capa de negocio

Los métodos que conforman la capa de negocio del módulo de gestión de libros, tienen una relación casi de 1 a 1 con los casos de uso definidos, con la excepción de algunos métodos auxiliares necesarios como “*showAllBooksByDistributorB()*”.

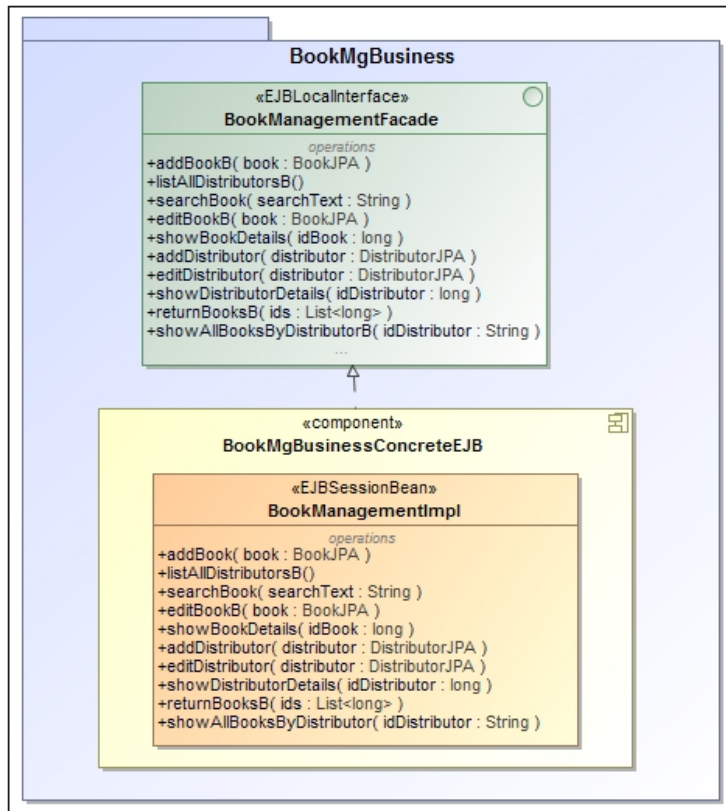


Ilustración 17. Diagrama capa negocio módulo gestión libros

Se ha seguido un patrón fachada por varios motivos. En primer lugar, desde el punto de vista del diseño, se ha considerado una buena práctica, puesto que en capsula la implementación interna de la capa de negocio, y solamente hace visible un interfaz externo. Además, desde el punto de vista tecnológico se prevé de esta manera la posibilidad de distribuir la capa de presentación de manera remota, por ejemplo a un dispositivo Android para preparar las devoluciones. Este es uno de los ejemplos de la aplicación del principio abierto/cerrado, anteriormente mencionado.

De nuevo, en el módulo de gestión del stock se ha empleado una política similar en cuanto a la definición de los métodos. Lo más destacable es la creación del método auxiliar “*matchPromos()*”. Este método auxiliar es el encargado de encapsular toda la lógica de aplicación de promociones en el momento de realizarse una venta. Recibe como parámetro una lista de los identificadores de los artículos que conforman la venta, y de manera interna aplica las reglas de negocio de promociones en cuanto a combinaciones, familias o precedencias en caso de coincidencia de promociones a aplicar. Se ha considerado que de este modo no se acopla la lógica de promociones al resto de la venta, al delegar en este método.

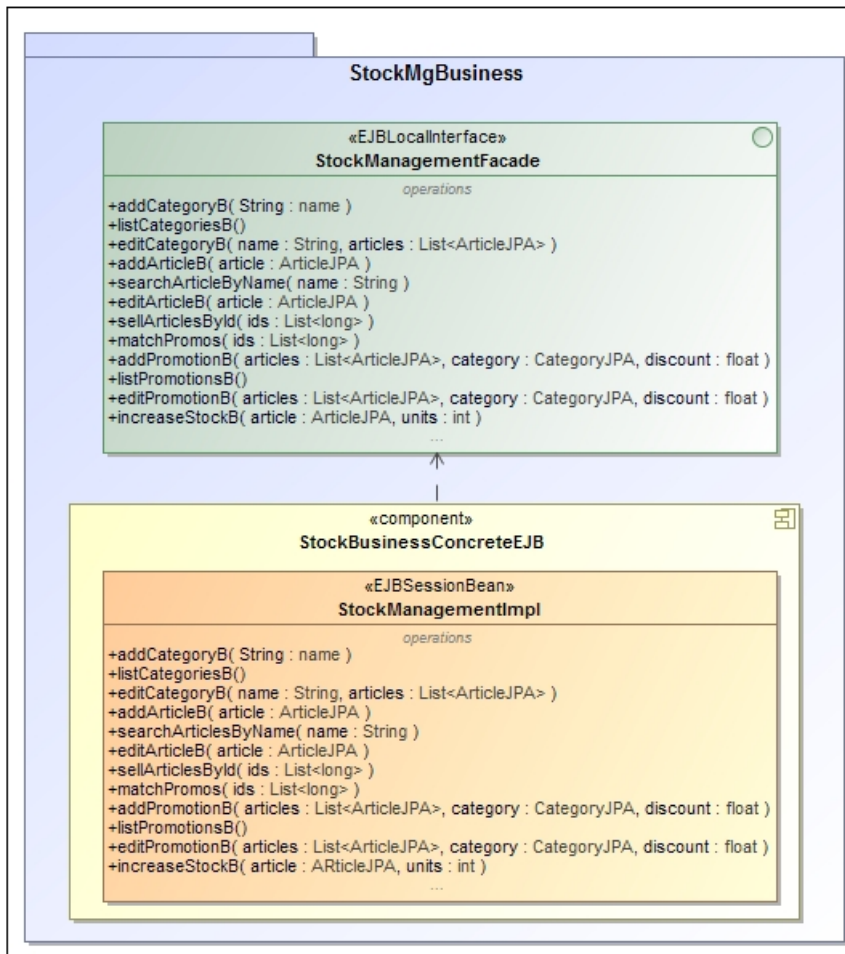


Ilustración 18. Diagrama capa negocio módulo gestión stock

En este caso, además de las ventajas desde el punto de vista del diseño del patrón fachada, permite la ampliación futura en caso de disponer de algún dispositivo de cobro específico.

La capa de negocio del módulo de gestión de eventos, se ha construido siguiendo el mismo modelo de los módulos anteriormente analizados. Lo más destacable dentro de este módulo es la similitud entre los métodos “*checkAvailability()*” y “*inspectOccupationB()*”. A pesar de que la firma de ambos métodos parece similar, tienen una funcionalidad y tipo de retorno distinto. El primero de ellos es un método auxiliar booleano que simplemente valida que las fechas escogidas para la creación de un evento son correctas desde el punto de vista de disponibilidad. El segundo de ellos, sin embargo, retorna un detalle completo de la disponibilidad y ocupación de una determinada sala, entre un determinado rango de fechas. En el momento de la implementación, posiblemente se cree un tercer método que sea invocado por estos, y concentre las operaciones comunes, siguiendo el principio de diseño “*DRY*”.

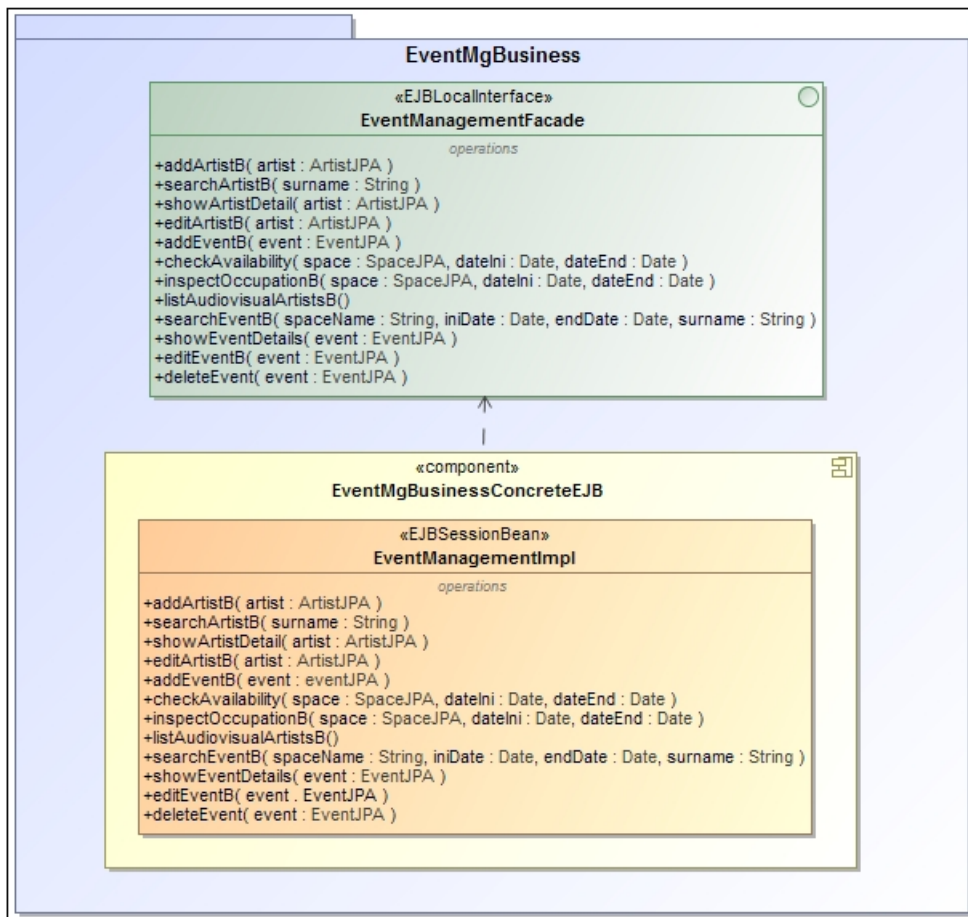


Ilustración 19. Diagrama capa negocio módulo gestión eventos

6.4 Diagramas de arquitectura de la capa de persistencia

En general, el mapeo desde las clases definidas en los diagramas de clases, hasta las clases de JPA ha sido directo en cuanto a entidades, atributos y tipos. De este modo se facilita la implementación, aprovechando la potencia tanto del paradigma de la Programación Orientada a Objetos, como de los *frameworks* de tipo ORM (JPA en este caso).

El modelo resultante, es el siguiente:

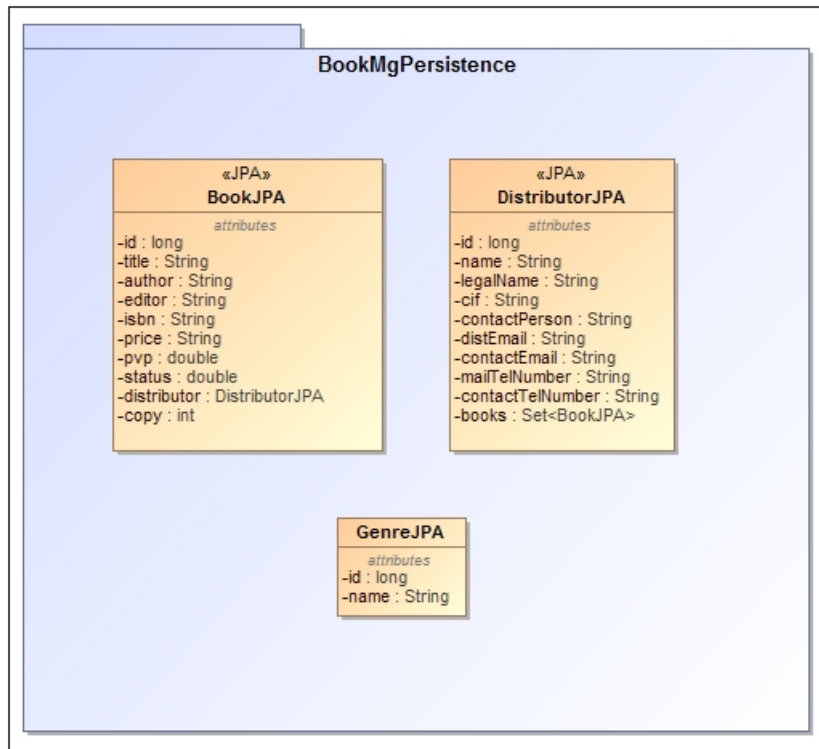


Ilustración 20. Diagrama capa persistencia módulo gestión libros

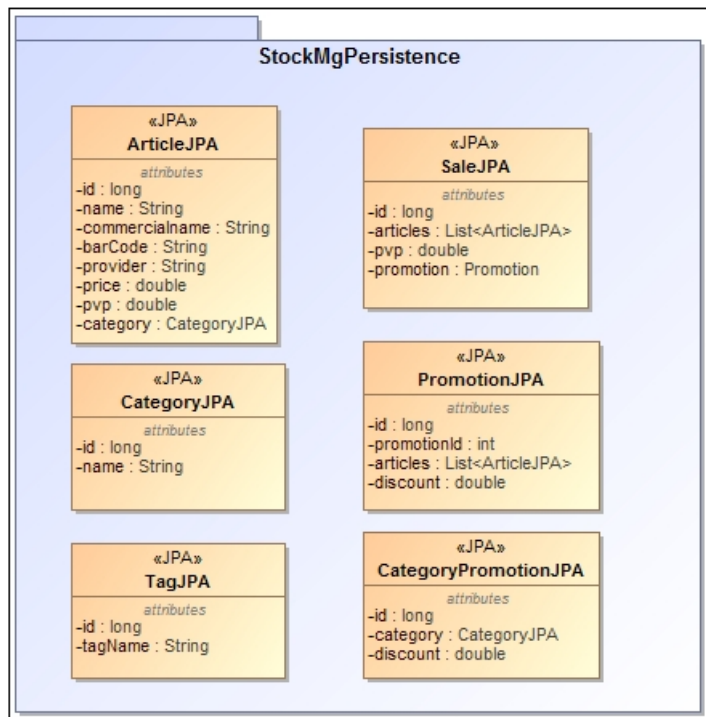


Ilustración 21. Diagrama capa persistencia módulo gestión stock

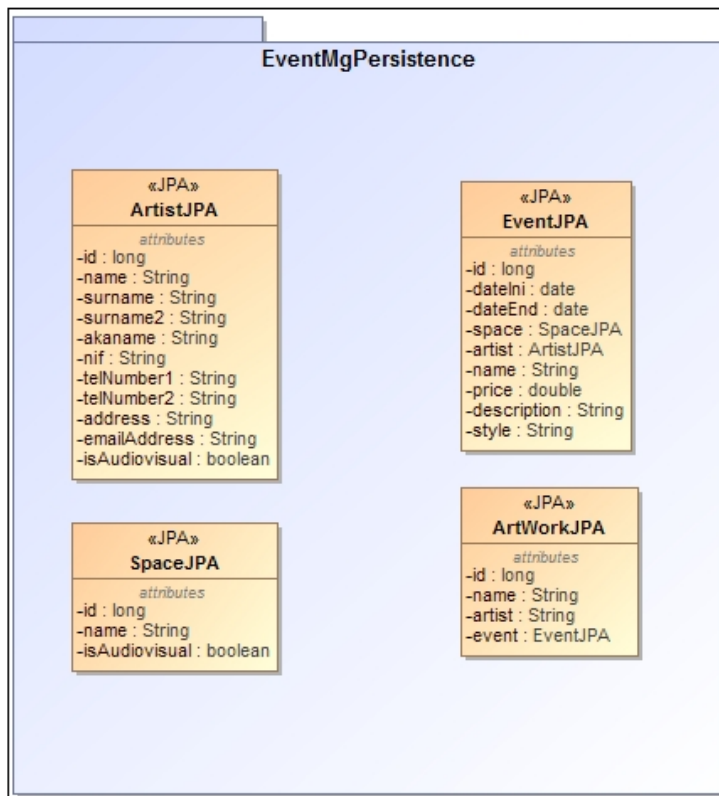


Ilustración 22. Diagrama capa persistencia módulo gestión eventos

6.5 Modelo relacional

A continuación, se muestra el modelo relacional necesario para dar soporte al modelo de información del sistema de los módulos previamente presentados. Se indican en **negrita** las claves de cada una de las tablas, y subrayadas las claves foráneas.

Se omite en el presente esquema la información relativa a los tipos de datos, puesto que esta ha ido indicada en el diagrama de clases, y además se delega en la tecnología de JPA para que mapee de manera automática los tipos de Java a los tipos de *PostgreSQL* en la creación de las tablas.

Gestión de libros:

- GENRES(**id**, genre_name)
- BOOKS(**id**, title, author, editor, isbn, price, pvp, status_id, copy, distributor_id)
- DISTRIBUTORS(**id**, name, legal_name, cif, contact_person, dist_email, contact_email, main_tel_number, contact_tel_number)
- DELIVERY_NOTES(**book_id**, **dist_id**, **date**, units)
- DELIVERY_NOTES_REFOUND(**book_id**, **dist_id**, **date**, units, total_impot)
- ENUMERATED_STATUS(**id**, status)

Cabe destacar que dado que las consultas a realizar son bastante sencillas y no requieren un extra de eficiencia, se ha optado mayoritariamente por un sistema de claves basado en identificadores numéricos autogenerados por JPA. De este modo se simplifica el diseño al optar por una política de delegación en el *framework* de persistencia.

Lo más destacable desde el punto de vista del análisis y diseño, es el tratamiento que se hace de las unidades. Es bastante habitual recibir varias unidades de un mismo libro, por lo que se planteaban dos posibilidades. En primer lugar se podía haber guardado un solo elemento en base de datos por cada libro, con un atributo numérico que indicase el número de copias disponibles. Este atributo se habría incrementado o disminuido al comprar, vender o devolver copias del ejemplar. Esta implementación presentaba un problema funcional, puesto que se quiere conservar el estado de los libros, para futuras implementaciones de análisis de datos. Por tanto es necesario tratar cada copia de un mismo libro como un ítem diferente en base de datos, de manera que cada una de las copias pueda tener un ciclo de vida independiente.

Para entender este hecho podemos pensar en 3 copias de un mismo libro, una disponible, otra vendida y otra retornada a la distribuidora. Esta información es mucho más rica y completa que el hecho de almacenar simplemente que se tiene 1 copia disponible. Por este motivo, se ha optado por esta segunda implementación, de manera que cada libro físico tiene una representación única en el sistema de persistencia, permitiendo una trazabilidad independiente del mismo.

En los casos de uso que necesiten un uso individual de cada copia, como por ejemplo una venta o retorno, se tratará, por tanto la entidad independiente del sistema. Sin embargo, en los casos en los que se requiera un tratamiento de todas las copias de un libro, como en el caso de uso de la edición, se agruparán mediante el atributo isbn.

En el caso de los albaranes de entrada (deliveryNote), se usará como clave la combinación de la clave del libro, más la clave de la distribuidora, mas la fecha, puesto que el sistema de unidades escogido para los libros garantiza que cada uno de ellos en el sistema sea único, pese a que se trata de copias de un mismo ejemplar.

Gestión de stock:

- TAGS(**id**, tag_name)
- CATEGORIES(**id**, name)
- ARTICLES(**id**, commercial_name, barcode, provider, price, pvp, category_id)
- PROMOTIONS(**id**, article_id, discount)
- CATEGORY_PROMOTIONS(**id**, category_id, discount)
- SALES(**id**, article_id, pvp, promotion_id)
- SALES_INFO(**id**, additional_info, total)

En este caso se ha optado totalmente por una política de claves basadas en ids numéricos autogenerados por los mismos motivos que los expuestos en el módulo de gestión de libros.

Desde el punto de vista del diseño, existen varias decisiones en la implementación que conviene destacar. En primer lugar, y a diferencia de la implementación usada en el módulo de gestión de libros, en este caso se ha optado por una representación de unidades como atributo de una única

entidad, puesto que en este caso no tiene sentido grabar una trazabilidad para cada una de las copias de un artículo.

Las promociones se almacenan en dos tablas distintas puesto que funcionalmente se han definido así a petición de los usuarios. Por un lado tenemos las promociones basadas en artículos, que pueden contener un solo artículo o una combinación de varios de estos, incluida la repetición de un mismo artículo n veces. Por otro lado, se prevén promociones a aplicar a una familia completa de artículos. En el momento de realizar una venta, tal como se refleja en el diagrama de secuencia, se buscará si los artículos pertenecen a alguna promoción y si esta es aplicable, es decir, si todos los artículos que la forman están presentes en la venta, o si pertenecen a alguna familia que se encuentra en promoción.

Por último, en lo referente a los precios de compra, este se fijará en el momento de la entrada del artículo. Si los proveedores ofrecen descuentos en ciertos artículos, se creará un nuevo artículo para indicar que es promocionado en la compra.

Gestión de eventos:

- SPACES(**id**, space_name, is_audiovisual)
- ARTWORKS(**id**, name, artist_id, event_id)
- ARTISTS(**id**, name, surname, surname2, akaname, nif, tel_number_1, tel_number_2, address, email_address, is_audiovisual)
- EVENTS(**id**, date_ini, date_end, space_id, artist_id, name, price, description, style)
- AVAILABILITIES(**artist_id**, **date_ini**, **date_end**)

En este caso se ha optado totalmente por una política de claves basadas en ids numéricos autogenerados por los mismos motivos que los expuestos en el módulo de gestión de libros.

Se ha estudiado la manera en la que se debía representar la herencia de las clases *AudioVisualSpace* y *AudiovisualArtist* y se llega a la conclusión de que es preferible implementar clases y subclases en una misma tabla. Solamente se necesita un dato de tipo *char* para distinguir si se trata de una u otra, por lo que actualmente no es un gasto significativo en espacio en base de datos. Por otro lado, el plantear dos tablas separadas hubiese requerido complicar las sentencias o incluso duplicarlas en alguna de las implementaciones.

El otro aspecto relevante de esta implementación es la manera en la que se representan los rangos de fechas de los artistas. Tratándose de una lista no finita en Java, se considera que la mejor implementación posible consistía en una tabla auxiliar que contuviese los identificadores del artista y tantas filas como rangos fuesen necesarios. De este modo, se optimiza el uso de espacio ya además se ofrece una versión normalizada de la base de datos.

7. Prototipos: Evaluación de los modelos y decisiones adoptadas

A lo largo de este capítulo, se pretende condensar los resultados del grueso del desarrollo dirigido por prototipos de interacción que se ha seguido en el proyecto. Concretamente, se muestran imágenes de los prototipos, así como un breve resumen de la evaluación de estos por parte de los usuarios, y las decisiones tomadas en base a estas pruebas.

Estos prototipos se han diseñado ajustando las horas a la disponibilidad planificada, con la intención de no perjudicar al resto de las tareas. Se han elegido dentro de cada módulo, aquellos casos de uso que se han considerado más críticos, o de un mayor valor funcional.

P1 – Alta libro:

LaSalaShop
Libros Gestión Eventos

Inserta un nuevo libro:

DISTRIB_1 ▼

Título:	<input type="text" value="título"/>
Autor:	<input type="text" value="autor"/>
ISBN:	<input type="text" value="isbn"/>
Precio Compra:	<input type="text" value="0.00"/>
PVP:	<input type="text" value="0.00"/>
Unidades:	<input type="text" value="1"/>

GUARDAR

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: La evaluación ha sido positiva por parte de los usuarios. Han encontrado la pantalla cómoda y correcta desde el punto de vista funcional. Valoran positivamente el uso de filtros en los campos para evitar errores.

Decisiones: Mantener el modelo.

P2 – Buscar libro:

LaSalaShop
Libros Gestión Eventos

Buscar un libro:

TITULO: libro

BUSCAR

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

LaSalaShop
Libros Gestión Eventos

Resultados búsqueda:

TITULO	PVP	
El Libro de la vida	19.99	<input type="checkbox"/>
El Libro de la muerte	11.99	<input type="checkbox"/>
El Libro con el titulo mas largo del mundo	9.0	<input type="checkbox"/>

SELECCIONAR

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: Los usuarios encuentran correcta la funcionalidad de búsqueda por título, pero es insuficiente. A lo largo de las pruebas se descubre que existen búsquedas por otros criterios que resultan igual de importantes.

Decisiones: Añadir otros criterios en la búsqueda. Concretamente se añadirá búsqueda por autor, ISBN y temática.

P3 – Preparar devolución:

LaSalaShop

Libros Gestión Eventos

Preparar devolución:

DISTRIB_1 ▼

TITULO	
El Libro de la vida	<input type="checkbox"/>
El Libro de la muerte	<input type="checkbox"/>
El Libro con el titulo mas largo del mundo	<input type="checkbox"/>
I	<input type="checkbox"/>
5añ_!	<input type="checkbox"/>

GUARDAR

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: Los usuarios quedan muy satisfechos con la funcionalidad, puesto que era uno de los temas que más les preocupaba. Se continúa haciendo pruebas para encontrar puntos de mejora, y se descubre que la ubicación del libro en la tienda es un dato que facilita aún más la devolución.

Decisiones: Incluir el atributo ubicación para la entidad libro, y mostrarlo en la pantalla de devoluciones.

P4 – Alta artículo:

LaSalaShop

Libros Gestión Eventos

Inserta un nuevo artículo:

REGALOS ▼

Nombre compra:	<input type="text" value="nombre compra"/>
Nombre comercial:	<input type="text" value="nombre comercial"/>
Precio Compra:	<input type="text" value="0.00"/>
PVP:	<input type="text" value="0.00"/>
Unidades:	<input type="text" value="1"/>

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: Los usuarios encuentran totalmente correcta la funcionalidad desarrollada.

Decisiones: Los usuarios realizarán más pruebas con el propósito de encontrar defectos funcionales o posibilidades de mejora lo antes posible.

P5 – Efectuar venta:

LaSalaShop

Libros Gestión Eventos

Busqueda artículos:

NOMBRE:	<input type="text" value="espiral"/>	<input type="button" value="BUSCAR"/>
---------	--------------------------------------	---------------------------------------

NOMBRE	PVP	
Camiseta espirales	25.59	<input type="button" value="AÑADIR"/>
Poster espirales Man Ray	99.59	<input type="button" value="AÑADIR"/>
Espirales taza	14.99	<input type="button" value="AÑADIR"/>

TOTAL VENTA:

ARTICULO	UNIDADES	PVP
Postales de La India	1	22.99
Set Manualities for dummies	0	21.59
Espirales taza	1	14.99

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: los usuarios encuentran correcto el prototipo, pero se muestran preocupados por la eficiencia del sistema de búsqueda, puesto que el prototipo no implementa esta funcionalidad, sino que la simula.

Decisiones: Preparar un prototipo que incluya la funcionalidad de búsqueda, antes de acometer completamente el desarrollo de esta funcionalidad.

P6 – Crear promoción:

The screenshot shows the 'LaSalaShop' interface for creating a promotion. At the top, the logo 'LaSalaShop' is displayed in blue, with the words 'Libros', 'Gestión', and 'Eventos' below it. The main heading is 'Crear promoción:'. Below this, there is a form with a label 'ARTICULO:' and a text input field containing 'Camiseta geométrica'. Underneath the input field, there are several radio button options: '2x1', '3x2', '5%', '10%', '15%', '100% - gratis', and 'other'. To the right of these options is a small text input field containing a '%' symbol. Below the radio buttons is a large button labeled 'EFECTUAR PROMOCION'. At the bottom of the form, there is a copyright notice: 'Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)'.

Evaluación: Como en el caso anterior, los usuarios dudan de la efectividad del sistema de búsqueda. El sistema diseñado no es del todo correcto, puesto que pueden aparecer otro tipo de promociones, en las que se combinen diferentes artículos.

Decisiones: Preparar una reunión monográfica con los usuarios para definir los sistemas de promociones. Posteriormente, se deberá rediseñar funcional y técnicamente este caso de uso.

P7 – Registrar nuevo artista:

The screenshot shows the LaSalaShop website header with the logo and navigation links 'Libros', 'Gestión', and 'Eventos'. Below the header, the text 'Registra un nuevo artista:' is displayed. The registration form consists of a table with eight rows, each containing a label and a text input field. The labels and their corresponding input field values are: 'Nombre:' with 'nombre', 'Apellido1:' with 'apellido1', 'Apellido2:' with 'apellido2', 'NIF-CIF:' with 'nif-cif', 'Telefono1:' with 'telefono1', 'Telefono2:' with 'telefono2', 'Dirección:' with 'dirección', and 'email:' with 'email'. Below the form is a 'GUARDAR DATOS' button. At the bottom of the page, there is a copyright notice: 'Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)'.

Nombre:	nombre
Apellido1:	apellido1
Apellido2:	apellido2
NIF-CIF:	nif-cif
Telefono1:	telefono1
Telefono2:	telefono2
Dirección:	dirección
email:	email

GUARDAR DATOS

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: Los usuarios validan positivamente la implementación de esta funcionalidad. Solicitan incluir un nuevo campo con el alias o nombre artístico.

Decisiones: Se mantiene el modelo, pero se añadirá el nuevo atributo 'alias'.

P8 – Reservar sala:

LaSalaShop
Libros Gestión Eventos

Inserta un nuevo evento:

Exposiciones LaSala ▼

Título Evento:	<input type="text" value="título evento"/>
Fecha inicio	<input type="text"/>
Fecha fin	<input type="text"/>

CREAR EVENTO

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: Los usuarios, en las primeras pruebas valoran positivamente el prototipo diseñado para esta funcionalidad, especialmente la inclusión del calendario javascript, que simplifica la introducción de las fechas. A partir de estas pruebas, se llega a la conclusión de los usuarios, de que sería interesante generar una vista que mostrase una línea temporal con las fechas de ocupación y vacantes de las salas.

Decisiones: Analizar y valorar el nuevo requisito de visualización de línea temporal de ocupación de las salas.

P9 – Lista artistas audiovisuales:

LaSalaShop

Libros Gestión Eventos

Artistas audiovisuales:

NOMBRE	APELLIDO	TELEFONO	
Albert	Gonzalez	98784563541	Consultar
John	Smith	83-123456789	Consultar
Tino	Gonzalez	9878-4563541	Consultar
Susana	Ninez	9878456-3541	Consultar
Maria	Garcia	98-7-84-56 35-41	Consultar
John	Gonzalez	98784563541	Consultar
Nombremuy largo	Apellido-muy-largo	98784563541	Consultar
Pepe	Perez	98745663541	Consultar
Albert	Lee	989-756-73541	Consultar
Mike	Doe	98784654541	Consultar

Copyright © TFG ebalsa@uoc.edu All Rights Reserved. | [LaSalaShop](#)

Evaluación: El listado es correcto, pero se necesita algún criterio de ordenación.

Decisiones: Tras analizar la funcionalidad con los usuarios, se considera que sería necesario mostrar la disponibilidad de los artistas, para aumentar el valor de la funcionalidad. Se analizará y valorará el cambio, consistente en introducir las fechas de disponibilidad de los artistas en su ficha, y poner fechas de inicio y fin a modo de filtro en el listado.

8. Conclusiones

A lo largo del presente capítulo se pretende hacer una descripción tanto de las conclusiones extraídas a lo largo del desarrollo del proceso, como las que se pueden obtener del producto final.

Además, se incluyen los activos de conocimiento que se han podido generar en el transcurso del proyecto, y una reflexión crítica acerca de los objetivos cumplidos frente a los inicialmente planteados.

Todo el contenido de este capítulo debería ser tenido en cuenta como elemento de información complementaria en la gestión de futuros proyectos, y especialmente en las fases de desarrollo pendientes de realizar de este producto.

En primer lugar, cabe destacar que uno de los objetivos específicos planteados estaba relacionado con medir la eficiencia real resultante de mezclar las bases de las metodologías ágiles, con algunos de los procesos del PMBOK. En este sentido, la experiencia ha sido sumamente satisfactoria. Mientras que la gestión más tradicional de proyectos ha aportado una seguridad y fiabilidad de base, los procesos iterativos de desarrollo han permitido aprovechar al máximo la interacción con los usuarios para ofrecer un producto que resultase más valioso.

En este sentido, han resultado especialmente útiles algunos de los procesos enmarcados dentro de la gestión de proyectos, como por ejemplo la gestión de riesgos. A través de estos procesos, se ha podido prever y salvar ciertos obstáculos que han ido surgiendo, y han ofrecido una serie de herramientas de eficacia probada para ir afrontando las diferentes fases del proyecto.

Relacionado con la gestión ágil, también se pretendía medir la efectividad de diseñar el producto basándose en prototipos que debían probar los usuarios. En general, se considera también una experiencia exitosa y satisfactoria, pero al mismo tiempo se han extraído algunas conclusiones que hay que señalar.

Por un lado, la interacción semanal mediante prototipos con los usuarios ha permitido ajustar las características del producto a las necesidades reales. Además, la detección temprana de defectos o necesidades de modificaciones, han reducido enormemente el coste de las acciones correctivas.

La parte negativa del desarrollo de prototipos está relacionada con el coste de los mismos. Se ha comprobado que incluso los prototipos más sencillos son costosos en cuanto a lo que horas de desarrollo se refiere, por lo que es importante asegurarse de prototipar aquello que se va a construir.

En relación a las conclusiones expuestas, podríamos decir que la implicación de los usuarios en el diseño es crucial, puesto que ayuda a desarrollar mucho

mejor un producto a medida. Pero a partir de esta idea, también hay una lección que se ha aprendido a lo largo del desarrollo del proceso, y es que hay que saber gestionar las expectativas de los usuarios. La implicación de los mismos es positiva, pero se debe tener especial cuidado con las nuevas peticiones, ideas contrapuestas o cambios que van surgiendo. Por tanto, es importante hacer un análisis reposado después de las reuniones con los usuarios, para filtrar, matizar y sobre todo valorar y priorizar aquellas peticiones que se propongan. La adaptabilidad de la metodología ágil a los cambios no debe servir en ningún caso como pretexto para tener que aceptar por sistema todos ellos.

En lo relativo al producto obtenido, las conclusiones son bastante positivas. La razón principal es que se ha conseguido desarrollar aquello que se necesitaba, dentro del calendario previsto, y con una calidad aceptada por los usuarios. Estos objetivos deberían ser la máxima principal de cualquier proyecto de Ingeniería del software, y por tanto el motivo de que se haga un balance positivo.

En cualquier caso, si se analiza con un espíritu crítico y de mejora el resultado obtenido, surgen puntos de mejora y sobre todo de aprendizaje. El más evidente es el relacionado con el aspecto visual del producto. Una lección valiosa que se ha aprendido es que el aspecto visual y estético siempre importa. Ya sea de una manera consciente o subconsciente, siempre se tiende a valorar de una manera más positiva aquello que luce mejor desde el punto de vista del diseño. Por tanto, habría que tener en cuenta este aspecto en todos los entregables, incluso en los prototipos, para no incidir negativamente en la percepción que los usuarios puedan tener del producto presentado.

Otra de las conclusiones que surgen de la autocrítica, es que en algunas fases del proyecto, se debería haber invertido mayor tiempo en analizar el balance entre coste de una tarea y valor real que aporta a los interesados. Esto ha sido especialmente patente en alguna de las entregas de prácticas parciales, puesto que ha habido partes muy costosas que no han aportado demasiado, y debido al tiempo empleado en su desarrollo se han omitido otras que eran mucho más necesarias. Por tanto, la conclusión es que los sistemas de valoración y priorización que se han empleado con el software, deberían utilizarse para todos los entregables de un proyecto.

Por último, tal y como se planteaba en los objetivos iniciales, se pretendía analizar la presencia de las competencias adquiridas a lo largo del Grado aplicadas a un proyecto real. En este sentido, las lecciones más importantes adquiridas han estado presentes en todo momento a la hora de tomar decisiones. En las fases de diseño y análisis han sido especialmente valiosas las competencias adquiridas en las asignaturas de *Ingeniería del software*, *Ingeniería de requisitos* y *Análisis y diseño con patrones*. Además, tratándose de un proyecto en el que se le ha dado una especial relevancia a la usabilidad, los conocimientos adquiridos en la asignatura *Interacción persona ordenador* han sido fundamentales.

Queda pendiente de analizar, por un lado, si el planteamiento abierto a la extensión que se ha hecho es realmente funcional cuando en el futuro se implementen nuevas funcionalidades, o se distribuya una parte de la capa de presentación.

Por otro lado, todo el sistema ha sido diseñado para almacenar los datos básicos de negocio de manera que en el futuro puedan ser explotados por un sistema de inteligencia de negocio. En este sentido será especialmente crítico averiguar si el sistema básico de minería de datos que se ha diseñado es suficientemente eficiente o no.

9. Glosario

A continuación se muestra, por orden alfabético, un listado de definiciones de los términos y acrónimos utilizados a lo largo de la presente memoria de proyecto.

Activos de conocimiento (Knowledge assets): Lecciones aprendidas a lo largo de la gestión de los proyectos, que se han consolidado como conocimiento, y pueden ser utilizadas como información de entrada en futuros proyectos.

Alvendi: Software gratuito de gestión de stock para comercios y PYMES. (<http://sistemaspaez.com/gestion-de-inventarios-con-alvendi/>).

DRY (Don't Repeat Yourself): Principio de diseño aplicado al software que recomienda no duplicar código, puesto que es una tendencia propensa a generar errores y difícil de mantener.

IntelTPV: Software de facturación y gestión de ventas para comercios. (<https://www.softpyme.net/software-tienda.php>)

Jboss: Servidor de aplicaciones para Java EE de código abierto, implementado en Java puro y por tanto, compatible con cualquier sistema operativo.

JPA (Java Persistence API): Es un framework de persistencia que permite el mapeo de objetos Java a entidades relacionales (ORM), de modo que puedan ser almacenadas y tratadas en una base de datos que siga el estándar SQL.

Metodologías ágiles: Metodologías de desarrollo de software que se basan en el desarrollo iterativo e incremental de funcionalidades completas, en la implicación de los usuarios, y en la mayor importancia de la comunicación fluida. Su base fundacional es el [manifiesto ágil](#).

MVC (Modelo-Vista-Controlador): Patrón de diseño que se aplica a la capa de presentación del software. Este patrón permite desligar los datos y la lógica de negocio de las vistas con las que interactúa el usuario, con lo que se reduce el acoplamiento y se favorece el encapsulamiento.

Open-Closed Principle (Principio Abierto-Cerrado): Principio de diseño aplicado al software que recomienda realizar implementaciones que permitan la extensión, pero sin necesidad de modificar el código interno.

Programación Orientada a Objetos (POO): Paradigma de diseño y programación cuyo origen está en Simula 67, u lenguaje creado por los Noruegos [Ole-Johan Dahl](#) y [Kristen Nygaard](#), del Centro de Cómputo Noruego en [Oslo](#).

Patrón fachada (facade): Patrón de diseño de software mediante el cual se reduce la complejidad de un sistema, al dividirlo en subsistemas, y ofrecer un punto único de entrada a modo de interfaz.

PMBOK (Project Management Body Of Knowledge): Guía publicada por el PMI (Project Management Institute), que recoge una serie de procesos, buenas prácticas y pautas para la gestión de proyectos.

PostgreSQL: Base de datos de tipo relacional de código abierto.

Prestashop: Software gratuito de e-commerce que permita crear una tienda on-line. (<https://www.prestashop.com/es/>)

Sprint: Cada una de las iteraciones que conforman un ciclo dentro de la metodología ágil de desarrollo de software.

SQLPyme: Software de tipo ERP para PYMES desarrollado por Distrtok. (<http://www.districtok.com/aplicaciones/programa-de-gestion/>)

UX (User eXperience): Percepción del usuario del uso de un determinado producto, compuesta por todos los elementos objetivos y subjetivos que se dan en dicha experiencia.

10. Bibliografía

[1] COHN, MIKE (2005); Agile Estimating and Planning. Stoughton, Massachusetts. Prentice Hall.

V.V.A.A. (2013); A guide to the project management body of knowledge (PMBOK guide), fifth edition. Pennsylvania. Project Management Institute, inc.

11. Anexo

El código fuente de la aplicación desarrollada puede consultarse en el repositorio GitHub, mediante la url:

- <https://github.com/ebalsaUOC/lasalashop>

