

Préstamo de material deportivo

Alberto Fernández Galdo

Ingeniería informática - Computación
Desarrollo web

Profesor Consultor: Gregorio Robles Martínez

Profesor responsable: Santi Caballe Llobet

12/06/2020



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Préstamo de material deportivo</i>
Nombre del autor:	<i>Alberto Fernández Galdo</i>
Nombre del consultor/a:	<i>Gregorio Robles Martínez</i>
Nombre del PRA:	<i>Santi Caballe Llobet</i>
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	Grado Ingeniería informática (Computación)
Área del Trabajo Final:	<i>Desarrollo web</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Liferay, Desarrollo Web, Java</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El objetivo del proyecto es facilitar al departamento encargado, la gestión del préstamo de material que la Diputación de Barcelona pone a disposición de los entes locales, los cuales realizan reservas de estos materiales para actividades en sus municipios.</p> <p>Actualmente, el departamento recibe consultas y realiza reservas del material disponible de manera telefónica. La gestión de las reservas que realizan las administraciones locales se hace mediante hojas de cálculo. Este procedimiento conlleva que el departamento emplee un tiempo en contabilizar el material que han prestado para saber si disponen de stock disponible y así cubrir las solicitudes de reserva. También deben de tener en cuenta que tras cada préstamo el material pasa un periodo de revisión para verificar su estado.</p> <p>Tras el desarrollo del proyecto, el departamento podrá llevar un control del material, saber el stock total y disponible de este, consultar rápidamente las reservas y su estado, además del material y la administración que lo ha solicitado. También un control de los materiales que están siendo revisados tras los préstamos.</p> <p>Por otra parte, las administraciones locales podrán realizar consultas y reservas de material a través de la página web de la Diputación de Barcelona.</p> <p>En conclusión, con el nuevo sistema, el procedimiento de gestión de préstamos se realizará de una manera más rápida, clara y automatizada. Tanto los entes locales como el departamento encargado realizarán sus correspondientes trámites de una manera más rápida y eficaz.</p>	

Abstract (in English, 250 words or less):

The goal of the project is help to improve the managing of borrowed material that Barcelona provincial government has as a service for other local municipalities.

The service consists to lend material to this municipalities with previous booking, but now the managing is been done by phone and saving the bookings in a spreadsheet. This procedure is hard, tedious and slow for the department staff.

This department must have noted the quantity of each material and modify it when some kind of material is borrowed or returned.

Also, they must to be controlled the material which is sended for a revision after the loan is returned.

When the new management system had been developed, the staff will make bookings fastly, to know the total and available stock for each material and manage also the maintenance.

Another interesting point is the local administrations who are interested to know the availability of the material, can do it through the web site, they can make bookings for this material as well.

This new system will help to the department to manage the loans better, it will be faster, easiest and more organize for them.

Índice

1. INTRODUCCIÓN.....	1
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	1
1.2 OBJETIVOS DEL TRABAJO	2
1.3 ENFOQUE Y MÉTODO SEGUIDO	3
1.4 PLANIFICACIÓN DEL TRABAJO	5
1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS.....	7
1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA.....	8
2. EVALUACIÓN DE RIESGOS	9
3. DESCRIPCIÓN DE LAS TAREAS	10
3.1 REQUISITOS DEL PROYECTO	10
3.2 PREPARACIÓN DEL ENTORNO.....	10
3.3 DISEÑO DEL PROYECTO	11
3.3.1 <i>Modelo conceptual</i>	11
3.3.2 <i>Back Office</i>	12
3.3.3 <i>Web</i>	12
3.4 DESARROLLO DEL PROYECTO	13
3.4.1 <i>Service-Builder</i>	13
3.4.1.1 <i>Creación Service-Builder</i>	13
3.4.1.2 <i>service.xml</i>	14
3.4.1.3 <i>Generar entidades</i>	22
3.4.2 <i>Back Office o Admin (panel-app)</i>	23
3.4.2.1 <i>Creación módulo Admin</i>	23
3.4.2.2 <i>Menús</i>	24
3.4.2.3 <i>portlet-model-hints.xml</i>	35
3.4.2.4 <i>Métodos adicionales</i>	36
3.4.2.5 <i>Custom Query</i>	37
3.4.2.6 <i>Scheduler</i>	39
3.4.2.7 <i>Language properties</i>	41
3.4.3 <i>Web (mvc-portlet)</i>	42
3.4.3.1 <i>Creación módulo Web</i>	42
3.4.3.2 <i>Búsqueda por fechas</i>	43
3.4.3.3 <i>Búsqueda por materiales</i>	44
3.4.3.4 <i>Finalizar reserva</i>	45
3.5.1 <i>Pruebas Back Office</i>	47
3.5.1.1 <i>Altas/Modificaciones/Bajas</i>	47
3.5.1.2 <i>Reservas/Préstamos</i>	47
3.5.1.3 <i>Mantenimientos</i>	49
3.5.1.4 <i>Stock</i>	49
3.5.2 <i>Pruebas Web</i>	50
3.5.2.1 <i>Reservas mediante búsqueda por fechas</i>	50
3.5.2.3 <i>Confirmación de reserva</i>	50
4. CONCLUSIONES	51
4.1 GENERAL	51
4.2 MÉTODO ACTUAL VS MÉTODO DESARROLLADO	51
4.3 OBJETIVOS LOGRADOS	52
4.4 AMPLIACIÓN	52
4.5 ANÁLISIS FINAL	53
5. ANEXOS.....	54
5.1 INSTALACIÓN ENTORNO DE DESARROLLO.....	54
5.2 PROBLEMAS ENCONTRADOS	55
6. BIBLIOGRAFÍA	59
7. GLOSARIO	60

Lista de figuras

Ilustración 1: Hoja de cálculo para la gestión de reservas y stock	1
Ilustración 2: Diagrama de tecnologías	4
Ilustración 3: Diagrama de Gantt	5
Ilustración 4: Prioridad de tareas	6
Ilustración 5: Modelo conceptual	11
Ilustración 6: Creación Service Builder (Paso1)	13
Ilustración 7: Creación Service Builder (Paso2)	13
Ilustración 8: Prefijo de nuestras tablas en BBDD	14
Ilustración 9: Entidad Maintenance en service.xml	15
Ilustración 10: Entidad Material en service.xml	16
Ilustración 11: Entidad Activity en service.xml	17
Ilustración 12: Entidad ActivityDependsOn en service.xml	18
Ilustración 13: Entidad LoanActivity en service.xml	19
Ilustración 14: Entidad Loan en service.xml	20
Ilustración 15: Entidad LocalEntity en service.xml	21
Ilustración 16: Construcción Service-Builder mediante service.xml	22
Ilustración 17: Estructura generada tras Service-Builder	22
Ilustración 18: Despliegue módulos Api i Service	22
Ilustración 19: Tablas generadas por Service-Builder	23
Ilustración 20: Creación módulo Admin con panel-app (Paso1)	23
Ilustración 21: Estructura generada tras la creación del módulo Admin	23
Ilustración 22: Creación módulo Admin con panel-app (Paso2)	23
Ilustración 23: Estructura de la carpeta Resources de service	24
Ilustración 24: Pantalla Mantenimiento Principal	25
Ilustración 25: Pantalla Mantenimiento Alta	25
Ilustración 26: Pantalla Mantenimiento Editar	26
Ilustración 27: Pantalla Mantenimiento Acciones	26
Ilustración 28: Pantalla Préstamo Principal	27
Ilustración 29: Pantalla Préstamo Alta 1/2	28
Ilustración 30: Pantalla Préstamo Alta 2/2	28
Ilustración 31: Pantalla Préstamo Editar	29
Ilustración 32: Pantalla Material Principal	30
Ilustración 33: Pantalla Material Alta	30
Ilustración 34: Pantalla Material Editar	31
Ilustración 35: Pantalla Actividades Principal	32
Ilustración 36: Pantalla Actividades Alta	33
Ilustración 37: Pantalla Actividad Editar	33
Ilustración 38: Pantalla Entidades locales Principal	34
Ilustración 39: Pantalla Entidades locales Alta	34
Ilustración 40: Pantalla Entidades locales Editar	34
Ilustración 41: Definir max-length en fichero portlet-hints.xml	35
Ilustración 42: Ocultar hora para fechas en fichero portlet-hints.xml	35
Ilustración 43: Definir métodos adicionales en las clases xxxImpl.java	36
Ilustración 44: Creación Custom Query (Paso1)	37
Ilustración 45: Creación Custom Query (Paso2)	37
Ilustración 46: Creación Custom Query (Paso3)	37

Ilustración 47: Creación Custom Query (Paso4)	38
Ilustración 48: Creación Custom Query (Paso5)	38
Ilustración 49: Creación de la clase Scheduler (Paso1)	39
Ilustración 50: Creación de la clase Scheduler (Paso2)	40
Ilustración 51: Creación de la clase Scheduler (Paso3)	40
Ilustración 52: Creación de la clase Scheduler (Paso4)	40
Ilustración 53: Ficheros Language.properties	41
Ilustración 54: Configuración fichero language.properties	41
Ilustración 55: Creación módulo web mediante mvc-portlet (Paso1)	42
Ilustración 56: Creación módulo web mediante mvc-portlet (Paso2)	42
Ilustración 57: Creación módulo web mediante mvc-portlet (Paso3)	42
Ilustración 58: Buscador por fechas de módulo web	43
Ilustración 59: Selección de material resultante	43
Ilustración 60: Buscador por materiales del módulo web	44
Ilustración 61: Confirmación de reserva en módulo web	45
Ilustración 62: Comprobante de reserva realizada	46
Ilustración 63: Tabla de compatibilidades Liferay 7.1	55
Ilustración 64: Consola GoGo shell	55
Ilustración 65: Fichero bundle de módulo api	56
Ilustración 66: Fichero bundle del módulo-service	56
Ilustración 67: Fichero bundle del módulo admin	56
Ilustración 68: Clase BundleActivator	58

1. Introducción

1.1 Contexto y justificación del Trabajo

La Diputación de Barcelona pone a disposición de los entes locales una variedad de artículos y materiales deportivos, los cuales pueden disfrutar de manera gratuita durante un período de tiempo.

Para realizar esta petición, los interesados pueden consultar desde la [página web de Préstecs de material esportiu](#)[8] todo el material del cual dispone la Diputación de Barcelona, para realizar la petición sobre un material, el interesado debe contactar telefónicamente y consultar la disponibilidad del mismo. Una vez se ha confirmado que el material está disponible, se anota la reserva en un documento Excel que se utiliza a modo de reservas y stock.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG		
1																																			
2																																			
3																																			
4																																			
5	DEMANDES	MATERIAL		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
393		Rugbi Inicicació																																	
394		Rugbi Inicicació																																	
395		Rugbi Inicicació																																	
396		Rugbi Inicicació																																	
397		Rugbi Inicicació																																	
398		Rugbi Inicicació																																	
399																																			
400		Tàndem T.																																	
401		Tàndem T.																																	
402		Tàndem T.																																	
403		Tàndem T.																																	
404																																			
405		Tanques Diba 3 m (5)																																	
406		Tanques Diba 3 m (5)																																	
407		Tanques Diba 3 m (5)																																	
408		Tanques Diba 3 m (5)																																	
409		Tanques Diba 3 m (5)																																	
410		Tanques Diba 3 m (5)																																	
411		Tanques Diba 3 m (5)																																	
412		Tanques Diba 3 m (5)																																	
413		Tanques Diba 3 m (5)																																	
414																																			
415		Tanques Diba 2.5 m																																	
416		Tanques Diba 2.5 m																																	
417																																			

Ilustración 1: Hoja de cálculo para la gestión de reservas y stock

Tras esta primera gestión, el interesado debe confirmar la reserva rellenando un [formulario](#)[9] que descargará de la [página web de solicitud](#)[10], indicando el material, fechas, lugar del evento, entre otros.

1.2 Objetivos del Trabajo

El objetivo del proyecto es informatizar y facilitar la gestión de préstamos comentada en el punto anterior.

El departamento encargado de la gestión del material podrá llevar un control del stock, préstamos y mantenimiento de material (tras su uso). En cuanto a los entes locales, podrán tener acceso vía web de una manera más fácil y rápida al material disponible de manera actualizada para realizar la reserva.

Para ello, deberemos realizar las siguientes tareas:

BBDD

- ✓ Integrar con la BBDD¹ existente las tablas necesarias para el control de préstamos.
- ✓ Creación de entidades
 - ✓ Actividad -> Conjunto de material que vamos a prestar a las entidades
 - ✓ Material -> El material contenido en la actividad
 - ✓ Entidad Local -> Interesados en reservar material
 - ✓ Mantenimiento -> Revisión y puesta a punto del material devuelto
 - ✓ Reserva / Préstamo -> Reserva y confirmación de material

BackOffice²

- ❖ Alta, baja y gestión del material.
- ❖ Alta, baja y gestión de las actividades.
- ❖ Alta, baja y gestión de las entidades locales.
- ❖ Mantenimiento de los materiales -> Material en revisión, fechas previstas...
- ❖ Confirmación de reserva, con lo que pasará a ser un préstamo.

Web

- ❖ Agenda: Visualización de la disponibilidad del material mediante un calendario.
- ❖ Categorización del material disponible.

BackOffice y Web

- ❖ Gestión del stock: Modificar stock en función de reservas, mantenimientos...
- ❖ Gestión de disponibilidad: Mostrar la disponibilidad del material en función de los diferentes factores.
- ❖ Reservas: Crear una relación del material, para las fechas indicadas con la entidad interesada. Se aplicará la modificación de stock pertinente.

A tener en cuenta:

Aclarar que el material que vamos a prestar le llamamos Actividad, ya que es el conjunto de materiales que lo componen. A veces nos podemos referir a él como Material, ya que es como se hace referencia a él hoy en día en la página web.

La diferencia entre reserva y préstamo es que la reserva no está confirmada y el préstamo sí, una reserva puede expirar si al cabo de un determinado tiempo no se confirma. Para confirmar una reserva, el departamento introduce una referencia llamada PMT³ en la reserva. Es por esto que una reserva puede tener los estados: *No confirmado, Confirmado, Reserva expirada, Prestado, En mantenimiento, Devuelto.*

1.3 Enfoque y método seguido

Para lograr los objetivos mencionados en el punto anterior vamos a dividir en varias fases el proyecto y así poder organizar el trabajo de una manera más eficiente. Nuestras fases serán:

❖ Fase I: Modelaje

En esta primera fase plasmamos sobre documento nuestra idea, realizamos un modelo lógico para crear nuestra BBDD, donde definimos nuestras entidades y atributos.

En esta fase también montamos nuestro entorno de trabajo. Instalación de software necesario y configuración del mismo. Entre el software que vamos a utilizar, necesitamos instalar el IDE⁴ Liferay Developer[4] con el que realizaremos el desarrollo de nuestro proyecto en la plataforma Liferay. Esta plataforma está desarrollada sobre lenguaje Java[3], que será el que utilizaremos para casi todo el desarrollo y trabajará contra una BBDD Oracle[11].

❖ Fase II: Implementación y desarrollo

En la segunda fase comenzaremos el desarrollo de nuestro sistema. Como pilar principal de proyecto está Service-Builder[12], un sistema de gestión de capas propio de Liferay[6] que se encarga de comunicas las capas modelo, persistencia y servicio con lo que facilita muchas tareas al programador.

Service-Builder además se encarga de la creación de las tablas necesarias para nuestro proyecto, así como el mapeo de estas con nuestras entidades, para ello se apoya en Hibernate[5], una herramienta de mapeo objeto-relacional para la plataforma Java.

Tras generar Service-Builder podremos centrarnos en el desarrollo de las tareas definidas anteriormente: Back Office y Web.

❖ Fase II: Pruebas

En la fase de pruebas corregiremos los errores o problemas que podamos ir encontrando a medida que testeamos nuestra aplicación. Es conveniente no dejar esta fase muy cercana a la fecha de entrega para poder evitar complicaciones con posibles imprevistos.

❖ Fase IV: Documentación

A pesar de ser la última fase, la documentación se realiza paralelamente con el desarrollo, de esta manera podemos tener un documento más fielmente redactado, además de que evitaremos tener una gran carga de trabajo al final si lo realizamos de esta manera.

En el siguiente esquema tenemos en el centro Service-Builder que nos proporciona Liferay como herramienta, además se apoya en Java e Hibernate para comunicar con nuestra BBDD Oracle.

Para la parte web utilizaremos JSP[13], jQuery[2] para manipulación de algunos eventos en los ficheros jsp y Bootstrap[1] para la estructura de páginas.

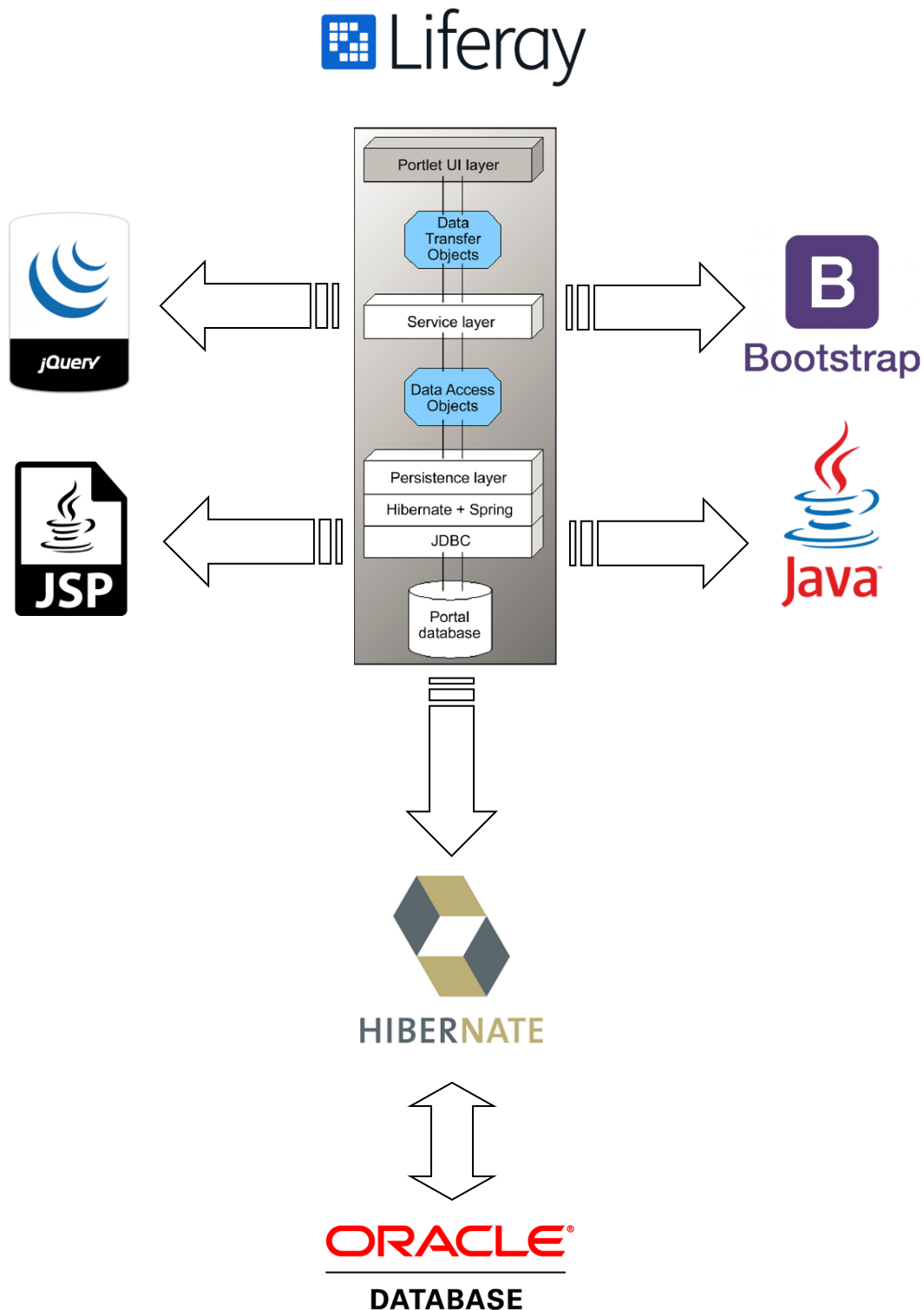


Ilustración 2: Diagrama de tecnologías

1.4 Planificación del Trabajo

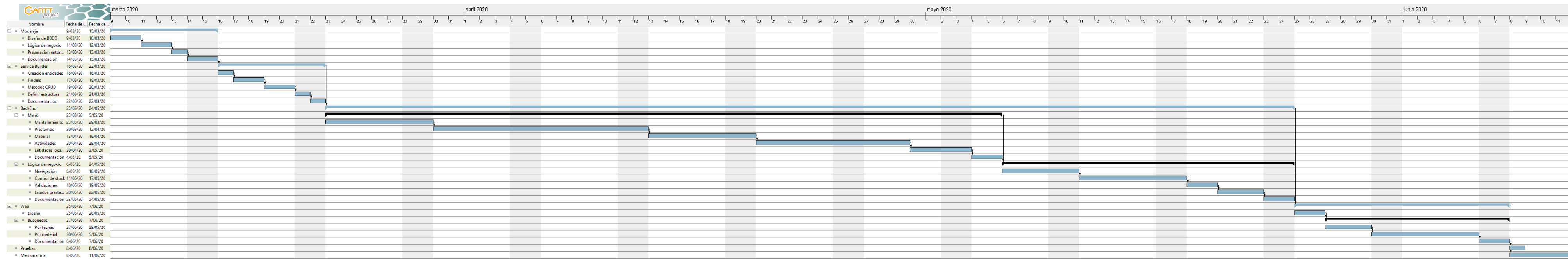


Ilustración 3: Diagrama de Gantt

En la siguiente tabla podemos ver las tareas definidas anteriormente, además del periodo de fechas que comprende cada una de ellas también podemos ver la prioridad que hemos definido.

Por defecto hemos establecido una prioridad “Alta” ya que prácticamente todas las tareas son funcionalidades básicas para alcanzar la meta del proyecto.

Las tareas prescindibles las hemos marcado con prioridad “Baja”, estas son:

Estados préstamos: No es necesaria para el funcionamiento del sistema, aunque es de gran ayuda para la gestión de este módulo. Tiene un valor de 2 días.

Búsquedas por material: Como tenemos otra búsqueda más prioritaria, podemos asegurar que la aplicación funcionará sin esta búsqueda, aunque ayudará al usuario a buscar el material en cuestión de una manera más personalizada. Tiene un valor de 6 días.

Validaciones: Tiene una prioridad normal y pese a ser bastante importante, el sistema llegaría a funcionar de todos modos.

Nombre	Fecha de inicio	Fecha de fin	Prioridad
☐ • Modelaje	9/03/20	15/03/20	Alta
• Diseño de BBDD	9/03/20	10/03/20	Alta
• Lógica de negocio	11/03/20	12/03/20	Alta
• Preparación entorno	13/03/20	13/03/20	Alta
• Documentación	14/03/20	15/03/20	Alta
☐ • Service Builder	16/03/20	22/03/20	Alta
• Creación entidades	16/03/20	16/03/20	Alta
• Finders	17/03/20	18/03/20	Alta
• Métodos CRUD	19/03/20	20/03/20	Alta
• Definir estructura	21/03/20	21/03/20	Alta
• Documentación	22/03/20	22/03/20	Alta
☐ • BackEnd	23/03/20	24/05/20	Alta
☐ • Menú	23/03/20	5/05/20	Alta
• Mantenimiento	23/03/20	29/03/20	Alta
• Préstamos	30/03/20	12/04/20	Alta
• Material	13/04/20	19/04/20	Alta
• Actividades	20/04/20	29/04/20	Alta
• Entidades locales	30/04/20	3/05/20	Alta
• Documentación	4/05/20	5/05/20	Alta
☐ • Lógica de negocio	6/05/20	24/05/20	Alta
• Navegación	6/05/20	10/05/20	Alta
• Control de stock	11/05/20	17/05/20	Alta
• Validaciones	18/05/20	19/05/20	Media
• Estados préstamos	20/05/20	22/05/20	Baja
• Documentación	23/05/20	24/05/20	Alta
☐ • Web	25/05/20	7/06/20	Alta
• Diseño	25/05/20	26/05/20	Normal
☐ • Búsquedas	27/05/20	7/06/20	Alta
• Por fechas	27/05/20	29/05/20	Alta
• Por material	30/05/20	5/06/20	Baja
• Documentación	6/06/20	7/06/20	Alta
• Pruebas	8/06/20	8/06/20	Alta
• Memoria final	8/06/20	11/06/20	Alta

Ilustración 4: Prioridad de tareas

1.5 Breve resumen de productos obtenidos

❖ Entorno

Pueden variar en función de la versión de Liferay que queramos instalar, para nuestro caso en concreto, que cuenta con una versión Liferay 7.1, los requerimientos son los siguientes.

- ✓ Portal Liferay versión “Liferay DXP 7.1.0 GA1”
- ✓ Apache Tomcat 9[4] (Incluido en la versión del portal)
- ✓ Oracle Database 11g / MySQL 5.7
- ✓ Java SDK 1.8

❖ Préstamo de material deportivo

Para el sistema que vamos a desarrollar vamos a obtener una serie de módulos desde los cuales vamos a poder realizar las gestiones necesarias. En primer lugar, el departamento encargado de gestionar las reservas y préstamos podrá acceder al módulo de administración o Back Office.

Desde este módulo podrá crear, modificar y eliminar las instancias para entidades locales, materiales, actividades, préstamos y mantenimientos.

- ✓ Módulo Back Office para la administración del departamento
 - ✓ Pantalla principal, alta, modificación para:
 - ✓ Entidades locales
 - ✓ Materiales
 - ✓ Actividades
 - ✓ Préstamos
 - ✓ Mantenimiento

Por otro lado, desde el módulo web, accesible desde el portal de la Diputación de Barcelona, las entidades locales podrán consultar el material disponible mediante uno o dos tipos de búsqueda y realizar reservas de este.

- ✓ Módulo web desde el cual las administraciones reservan el material
 - ✓ Búsqueda por fechas
 - ✓ Búsqueda por material

1.6 Breve descripción de los otros capítulos de la memoria

En primer lugar, estudiaremos la mejor manera de abordar el proyecto, realizaremos el modelo conceptual, del cual partirá el pilar del proyecto Service-Builder y en el cual crearemos las capas persistence⁵, model⁶ y service⁷, además, deberemos definir los finders⁸, que utilizaremos para acceder a las instancias de los objetos creados mediante sus atributos, es decir, podemos definir métodos get a través de finders, pero únicamente para entidades individuales. Si quisiéramos referenciar varias entidades deberíamos realizar Custom Querys⁹, que también veremos.

Además, crearemos una clase Scheduler¹⁰, donde programaremos una tarea diaria para verificar el estado de los préstamos y stock, de esta manera actualizará lo que sea necesario en nuestro sistema y mantendrá la congruencia de datos.

Tras generar nuestro Service-Builder, comenzaremos el desarrollo de los diferentes módulos que hemos visto en el punto anterior, módulo admin (Back Office) que deberemos generar mediante un proyecto tipo panel-app.

Para el otro módulo, el web, deberemos generar un proyecto tipo mvc-portlet¹¹, que también explicaremos.

De esta manera nuestro proyecto tendrá la siguiente estructura:

- ❖ PrestecMaterialEsportiu
 - ❖ Service Service-Builder
 - ❖ Api¹² Service-Builder
 - ❖ Admin panel-app
 - ❖ Web mvc-portlet

En los siguientes capítulos, realizaremos una descripción de las tareas realizadas a medida que el desarrollo de nuestro sistema avanza, además, se argumentará el motivo de cada toma de decisiones.

2. Evaluación de riesgos

Algunos de los problemas técnicos que nos podemos encontrar debido a la complejidad que presentan algunos puntos a desarrollar en el proyecto pueden ser:

- ❖ Mostrar la disponibilidad de material según las fechas seleccionadas, al igual que el stock disponible, ya que dependen de varios factores como son:
 - Los préstamos
 - El mantenimiento de material
 - Periodo previo a una reserva sobre un tipo de material en concreto y que define el departamento
 - Periodo posterior a una reserva sobre un tipo de material en concreto y que define el departamento

- ❖ La visualización de la disponibilidad del material mediante el calendario
 - Los préstamos
 - El mantenimiento de material

A los puntos comentados anteriormente debemos añadir la inexperiencia en Liferay, sumado al poco conocimiento en desarrollo web, el cual es posible que sea requerido en el transcurso del proyecto.

Por todo ello, es posible que no se alcancen los plazos de tiempo estipulados en la “Planificación temporal”, con lo que se debería alargar el plazo de entrega o en su defecto, reducir el alcance del proyecto y los objetivos establecidos.

3. Descripción de las tareas

3.1 Requisitos del proyecto

Pueden variar en función de la versión de Liferay que queramos instalar, para nuestro caso en concreto, que cuenta con una versión Liferay 7.1, los requerimientos son los siguientes.

Requerimientos software

- ❖ Java SDK 1.8
- ❖ Liferay Developer Studio 3.7.1.201910160309-ga2
- ❖ Oracle / MySQL 5.7[7]
- ❖ Liferay CE Portal 7.1.0 GA1 / Liferay DXP 7.1.0 GA1
- ❖ Apache Tomcat 9 (Ya incluido en el portal Liferay)

3.2 Preparación del entorno

Para el desarrollo del proyecto no necesitaremos instalar Liferay ya que nuestro proyecto consiste en la integración de la funcionalidad previamente explicada en el portal de la Diputación de Barcelona. Aun así, para poder situarnos mejor, pasaré a explicar los pasos para poder construir el entorno necesario y en el cual se desarrollará este proyecto.

Para conocer los pasos de la instalación, podemos dirigirnos al anexo en la página 54 de este documento, donde se detalla en profundidad.

3.3 Diseño del proyecto

3.3.1 Modelo conceptual

En el siguiente modelo conceptual podemos ver el proyecto que se quiere llevar a cabo. A partir de este modelo podremos realizar el diseño de BBDD, las entidades que necesitamos y también los atributos utilizados.

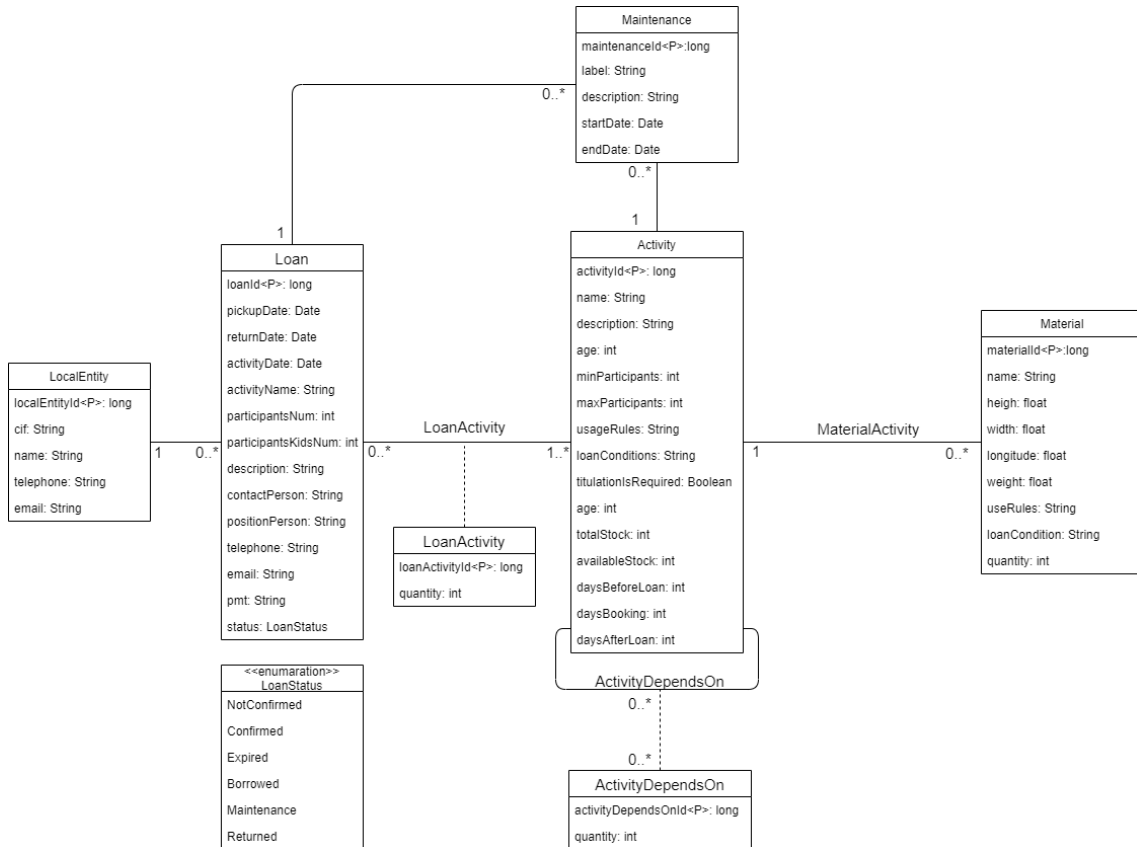


Ilustración 5: Modelo conceptual

3.3.2 Back Office

Desde el Back Office el personal de la Diputación de Barcelona podrá realizar el alta, baja o modificación de entidades, actividades, material y mantenimiento de estos, así como la confirmación y visualización de las reservas que realicen las entidades.

3.3.3 Web

Desde la zona web, las entidades podrán visualizar el material disponible para poder reservar.

La consulta de disponibilidad pasa por dos opciones:

- ❖ Consulta de todo el material disponible para las fechas seleccionadas previamente.
- ❖ Consulta las fechas disponibles para el material seleccionado previamente.
 - Muestra un calendario con los días marcados para ayudar visualmente al usuario.
 - Si el material no está disponible, se ofrece una sugerencia de material similar.

Una vez seleccionado el material deseado, el usuario procederá a rellenar el formulario de reserva, estos datos se almacenarán en BBDD y la reserva deberá ser confirmada por el personal responsable (desde Back Office).

Si la reserva no pasa a un estado confirmado al cabo de n días (se debe definir el valor de n previamente), se anulará automáticamente y el material pasará a estar disponible de nuevo.

3.4 Desarrollo del proyecto

3.4.1 Service-Builder

Como he comentado anteriormente, Service-Builder genera la capa persistence, service y model. De manera que gestiona el manejo de tablas y creación de entidades realizando el mapeo¹³ entre estas de manera automática. Del mismo modo, genera los métodos necesarios con los que trabajaremos para realizar el proyecto.

Para realizar esto, únicamente deberemos configurar el fichero service.xml¹⁴ donde indicaremos las entidades y atributos que queremos crear.

3.4.1.1 Creación Service-Builder

Para crear un nuevo Service-Builder seguimos los pasos que se muestran en las imágenes.

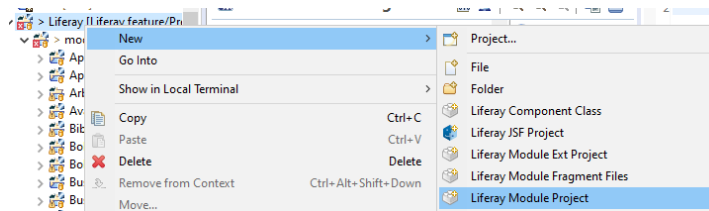


Ilustración 6: Creación Service Builder (Paso1)

Indicaremos el nombre, además de la versión de Liferay a utilizar, en nuestro caso será la 7.1 y, por último, el Project Template que será service-builder.
Si pulsamos en siguiente, podremos seleccionar el package a utilizar, en este caso hemos indicado el siguiente: cat.diba.materialesportiu

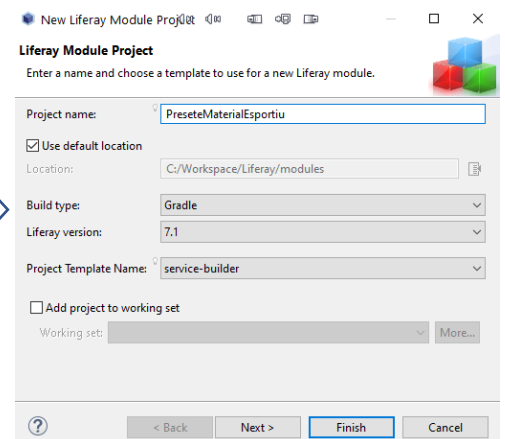


Ilustración 7: Creación Service Builder (Paso2)

3.4.1.2 service.xml

Tras crear nuestro módulo, podremos encontrar el fichero services.xml dentro de:

- ❖ PrestecMaterialEsportiu
 - ❖ PrestecMaterialEsportiu-service
service.xml

Aquí crearemos nuestras tablas y para ello podemos hacerlo de dos maneras, editando el fichero desde Source o desde Overview, este último de manera gráfica y más fácil.

El código generado en Source para nuestro service.xml es el siguiente:

En primer lugar, indicaremos el package en el que trabajaremos y namespace¹⁵, que será el prefijo que se añadirá a las tablas generadas e indicadas en este mismo fichero.

```
<service-builder package-path="cat.diba.materialesportiu">  
  <namespace>PME</namespace>
```

Ilustración 8: Prefijo de nuestras tablas en BBDD

Ahora procedemos a definir todas las entidades necesarias que serán las siguientes:

Maintenance, Material, MaterialActivity, Activity, ActivityDependsOn, LoanMaterial, Loan, LocalActivity

Para cada una de ella deberemos definir las siguientes propiedades:

- local-service: entidad accesible desde la propia aplicación.
- name: nombre de la entidad.
- remote-service: la entidad será accesible desde el exterior.
- uuid: identificador único.
- column: atributos para cada entidad
- order: ordenación por defecto
- finder: búsquedas que usaremos para esta entidad

Los siguientes atributos estarán en todas las tablas para facilitar la búsqueda:

- companyId: id del site en el que estamos
- userId: id del usuario
- userName: nombre del usuario
- createDate: fecha de creación
- modifiedDate: fecha de modificación

3.4.1.2.1 Maintenance

Esta es la entidad para el mantenimiento o revisiones de material tras ser devueltos.

- `maintenanceId`: Identificador para esta entidad
- `activityId`: Clave foránea de Activity
- `label`: Etiqueta identificativa para el material
- `description`: Descripción para el mantenimiento
- `startDate`: Fecha de inicio para el mantenimiento
- `endDate`: Fecha de fin para el mantenimiento
- `isDone`: Si el mantenimiento ya está hecho =1, si no = 0
- `loanId`: Clave foránea del préstamo, puede ser null

```
<entity local-service="true" name="Maintenance" remote-service="false" uuid="false">
  <!-- PK fields -->
  <column name="maintenanceId" primary="true" type="Long" />

  <!-- Group instance -->
  <column name="activityId" type="Long" primary="false" filter-primary="false" />
  <column name="label" type="String" />
  <column name="description" type="String" />
  <column name="startDate" type="Date" />
  <column name="endDate" type="Date" />
  <column name="isDone" type="Boolean"></column>
  <column name="loanId" type="Long"></column>

  <column name="groupId" type="Long" />

  <!-- Audit fields -->
  <column name="companyId" type="Long" />
  <column name="userId" type="Long" />
  <column name="userName" type="String" />
  <column name="createDate" type="Date" />
  <column name="modifiedDate" type="Date" />

  <!-- Other fields -->

  <!-- Order -->
  <order by="desc">
    <order-column name="startDate" />
  </order>

  <!-- Finder methods -->
  <finder name="ActivityId" return-type="Collection">
    <finder-column name="activityId" />
  </finder>
  <finder name="ActivityId_Label" return-type="Maintenance">
    <finder-column name="activityId"></finder-column>
    <finder-column name="label" />
  </finder>

  <!-- References -->
  <finder name="CompanyGroup" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
  </finder>

  <reference entity="AssetEntry"
    package-path="com.liferay.portlet.asset" />
  <reference entity="AssetTag"
    package-path="com.liferay.portlet.asset" />
</entity>
```

Ilustración 9: Entidad Maintenance en service.xml

3.4.1.2.2 Material

- materialId: Identificador para esta entidad
- heigh, width, longitude: Dimensiones del material
- weight: Peso del material
- activityId: Clave foránea de Activity
- quantity: Unidades de este material para Activity

Los atributos de dimensiones y peso se utilizarán para calcular el volumen y peso total de una reserva, con lo que el interesado podrá deducir el tipo de vehículo necesario para transportar el material.

```
<entity local-service="true" name="Material" remote-service="false" >
  <column name="materialId" primary="true" type="long"></column>
  <column name="name" type="String" localized="true"></column>
  <column name="heigh" type="float"></column>
  <column name="width" type="float"></column>
  <column name="Longitude" type="float"></column>
  <column name="weight" type="float"></column>
  <column name="activityId" type="long"></column>
  <column name="quantity" type="int"></column>
  <column name="groupId" type="Long"></column>
  <column name="companyId" type="Long"></column>
  <column name="userId" type="Long"></column>
  <column name="userName" type="String"></column>
  <column name="createDate" type="Date"></column>
  <column name="modifiedDate" type="Date"></column>
  <order by="asc">
    <order-column name="name" order-by="asc"></order-column>
  </order>
  <finder name="MaterialId" return-type="Material">
    <finder-column name="materialId"></finder-column>
  </finder>
  <finder name="ActivityId" return-type="Collection">
    <finder-column name="activityId"></finder-column>
  </finder>
  <finder name="CompanyGroup" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
  </finder>
  <finder name="CompanyGroupActivityId" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
    <finder-column name="activityId"></finder-column>
  </finder>
</entity>
```

Ilustración 10: Entidad Material en service.xml

3.4.1.2.3 Activity

- activityId: Identificador para esta entidad
- titulationIsRequired: Si la actividad requiere titulación para su uso
- totalStock: El stock total para esta actividad
- availableStock: El stock disponible en este momento.
- daysBeforeLoan: Días previos a la reserva en los que el interesado debe recoger el material
- daysBooking: Días que esta actividad se bloquea en estado reservada antes de que expire
- daysAfterLoan: Días que la actividad permanece en mantenimiento tras su devolución

```
<entity name="Activity" local-service="true" remote-service="false">
  <column name="activityId" type="long" primary="true"></column>
  <column name="name" type="String" localized="true"></column>
  <column name="description" type="String" localized="true"></column>
  <column name="age" type="int"></column>
  <column name="minParticipants" type="int"></column>
  <column name="maxParticipants" type="int"></column>
  <column name="usageRules" type="String" localized="true"></column>
  <column name="LoanConditions" type="String" localized="true"></column>
  <column name="titulationIsRequired" type="boolean"></column>
  <column name="totalStock" type="int"></column>
  <column name="availableStock" type="int"></column>
  <column name="daysBeforeLoan" type="int"></column>
  <column name="daysBooking" type="int"></column>
  <column name="daysAfterLoan" type="int"></column>

  <!-- PK fields -->

  <!-- Group instance -->
  <column name="groupId" type="long"></column>

  <!-- Audit fields -->

  <column name="companyId" type="long"></column>
  <column name="userId" type="long"></column>
  <column name="userName" type="String"></column>
  <column name="createDate" type="Date"></column>
  <column name="modifiedDate" type="Date"></column>
  <order by="asc">
    <order-column name="name" order-by="asc"></order-column>
  </order>
  <finder name="ActivityId" return-type="Activity">
    <finder-column name="activityId"></finder-column>
  </finder>
  <finder name="AvailableStock" return-type="Collection">
    <finder-column name="availableStock" comparator=">"></finder-column>
  </finder>
  <finder name="CompanyGroup" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
  </finder>
</entity>
```

Ilustración 11: Entidad Activity en service.xml

3.4.1.2.4 ActivityDependsOn

Esta entidad es la encargada de gestionar actividades que dependen de otras, es decir, una actividad puede estar contenida en otra. Si la actividad padre es reservada, las actividades hijos también se reserva. Esto influye en el recuento total de stock en todos los sentidos.

- activityDependsOnId: Identificador para esta entidad
- activityId: Activity hijo
- parentActivity: Actividad padre
- quantity: Unidades de Activity hijo en Activity padre

```
<entity name="ActivityDependsOn" local-service="true" remote-service="false">
  <column name="activityDependsOnId" primary="true" type="Long"></column>
  <column filter-primary="false" name="activityId" type="Long"></column>
  <column name="parentActivityId" type="Long"></column>

  <!-- PK fields -->

  <!-- Group instance -->
  <column name="quantity" type="int"></column>

  <column name="groupId" type="Long"></column>

  <!-- Audit fields -->

  <column name="companyId" type="Long"></column>
  <column name="userId" type="Long"></column>
  <column name="userName" type="String"></column>
  <column name="createDate" type="Date"></column>
  <column name="modifiedDate" type="Date"></column>

  <!-- PK fields -->

  <!-- Group instance -->

  <!-- Audit fields -->
  <finder name="CompanyGroup" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
  </finder>
  <finder name="ActivityId" return-type="Collection">
    <finder-column name="activityId"></finder-column>
  </finder>
  <finder name="ActivityIdParentActivityId" return-type="ActivityDependsOn">
    <finder-column name="activityId"></finder-column>
    <finder-column name="parentActivityId"></finder-column>
  </finder>
  <finder name="ParentActivityId" return-type="Collection">
    <finder-column name="parentActivityId"></finder-column>
  </finder>
</entity>
```

Ilustración 12: Entidad ActivityDependsOn en service.xml

3.4.1.2.5 LoanActivity

Esta entidad es la encargada de almacenar las actividades de una reserva/préstamo, así como el número de unidades de cada.

- loanActivityId: Identificador para esta entidad
- loanId: Clave foránea para Loan
- activityId: Clave foránea para Activity
- quantity: Unidades de Activity para Loan

```
<entity name="LoanActivity" local-service="true" remote-service="false">
  <column name="LoanActivityId" type="Long" primary="true"></column>
  <column name="LoanId" type="Long"></column>
  <column name="activityId" type="Long"></column>
  <column name="quantity" type="int"></column>

  <!-- PK fields -->

  <!-- Group instance -->

  <column name="groupId" type="Long"></column>

  <!-- Audit fields -->

  <column name="companyId" type="Long"></column>
  <column name="userId" type="Long"></column>
  <column name="userName" type="String"></column>
  <column name="createDate" type="Date"></column>
  <column name="modifiedDate" type="Date"></column>
  <finder name="CompanyGroup" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
  </finder>
  <finder name="LoanId" return-type="Collection">
    <finder-column name="LoanId"></finder-column>
  </finder>
  <finder name="ActivityId" return-type="Collection">
    <finder-column name="activityId"></finder-column>
  </finder>
  <finder name="LoanIdActivityId" return-type="LoanMaterial">
    <finder-column name="LoanId"></finder-column>
    <finder-column name="activityId"></finder-column>
  </finder>
</entity>
```

Ilustración 13: Entidad LoanActivity en service.xml

3.4.1.2.6 Loan

Esta entidad pertenece a préstamos y guardamos los datos de la reserva.

- loanId: Identificador para esta entidad
- localEntityId: Clave foránea para la entidad local
- pickupDate: Fecha de recogida del préstamo
- activityDate: Fecha del acontecimiento que celebra la entidad
- returnDate: Fecha de devolución del préstamo
- pmt: Código que introduce el departamento para confirmar la reserva y pasa a ser un préstamo
- status: Estados posibles de Loan
 - NOTCONFIRMED: La reserva se ha hecho y falta confirmar
 - EXPIRED: Ha excedido “daysBooking” antes de confirmar
 - CONFIRMED: Ha sido confirmada en el plazo
 - BORROWED: El material ha sido recogido
 - MAINTENANCE: Material está en revisión tras su devolución
 - RETURNED: Material revisado y devuelto

```
<entity name="Loan" local-service="true" remote-service="false" >
  <column name="loanId" type="Long" primary="true"></column>
  <column name="localEntityId" type="Long"></column>
  <column name="pickupDate" type="Date"></column>
  <column name="returnDate" type="Date"></column>
  <column name="activityDate" type="Date"></column>
  <column name="activityName" type="String"></column>
  <column name="participantsNum" type="int"></column>
  <column name="participantsKidsNum" type="int"></column>
  <column name="description" type="String"></column>
  <column name="contactPerson" type="String"></column>
  <column name="positionPerson" type="String"></column>
  <column name="telephone" type="String"></column>
  <column name="email" type="String"></column>
  <column name="pmt" type="String"></column>
  <column name="status" type="String"></column>

  <!-- PK fields -->

  <!-- Group instance -->

  <column name="groupId" type="Long"></column>

  <!-- Audit fields -->

  <column name="companyId" type="Long"></column>
  <column name="userId" type="Long"></column>
  <column name="userName" type="String"></column>
  <column name="createDate" type="Date"></column>
  <column name="modifiedDate" type="Date"></column>
  <order by="desc">
    <order-column name="pickupDate"></order-column>
  </order>
  <finder name="CompanyGroup" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
  </finder>
</entity>
```

Ilustración 14: Entidad Loan en service.xml

3.4.1.2.7 LocalEntity

En esta entidad guardamos los entes locales que realizarán las reservas y préstamos.

- localEntityId: Identificador para esta entidad
- cif: La entidad deberá introducirlo al reservar por web

```
<entity name="LocalEntity" local-service="true" remote-service="false">
  <column name="LocalEntityId" type="Long" primary="true"></column>
  <column name="cif" type="String"></column>
  <column name="name" type="String"></column>
  <column name="telephone" type="String"></column>
  <column name="email" type="String"></column>

  <!-- PK fields -->

  <!-- Group instance -->

  <column name="groupId" type="Long"></column>

  <!-- Audit fields -->

  <column name="companyId" type="Long"></column>
  <column name="userId" type="Long"></column>
  <column name="userName" type="String"></column>
  <column name="createDate" type="Date"></column>
  <column name="modifiedDate" type="Date"></column>
  <order by="asc">
    <order-column name="name" order-by="asc"></order-column>
  </order>
  <finder name="Name" return-type="Collection">
    <finder-column name="name"></finder-column>
  </finder>
  <finder name="CompanyGroup" return-type="Collection">
    <finder-column name="companyId"></finder-column>
    <finder-column name="groupId"></finder-column>
  </finder>
</entity>
```

Ilustración 15: Entidad LocalEntity en service.xml

3.4.1.3 Generar entidades

Una vez tenemos creado nuestro service.xml y queremos generar nuestras entidades debemos construir el servicio (build-service), como muestra la imagen.

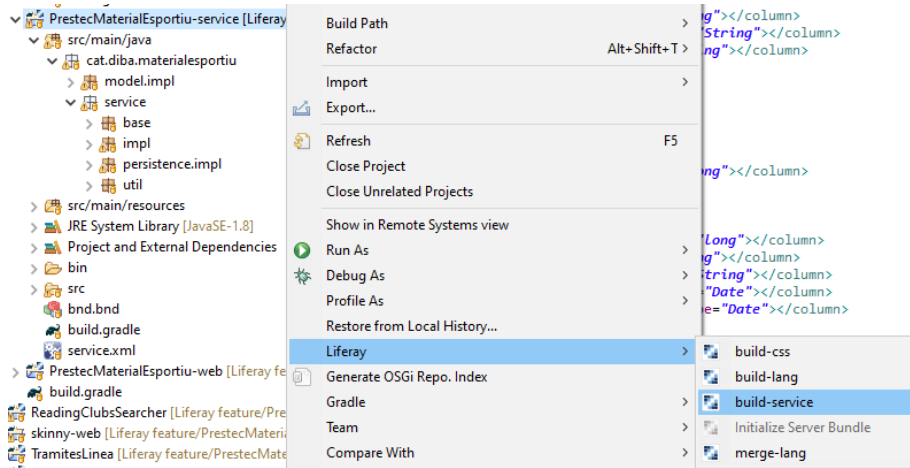


Ilustración 16: Construcción Service-Builder mediante service.xml

Esto genera la siguiente estructura:

- ❖ PrestecMaterialEsportiu
 - ❖ PrestecMaterialEsportiu-api
 - ❖ cat.diba.materialesportiu
 - ❖ constants
 - ❖ exception
 - ❖ model
 - ❖ service
 - ❖ PrestecMaterialEsportiu-service
 - ❖ cat.diba.materialesportiu
 - ❖ model.impl
 - ❖ service
 - ❖ base
 - ❖ impl
 - ❖ persistence.impl
 - ❖ util

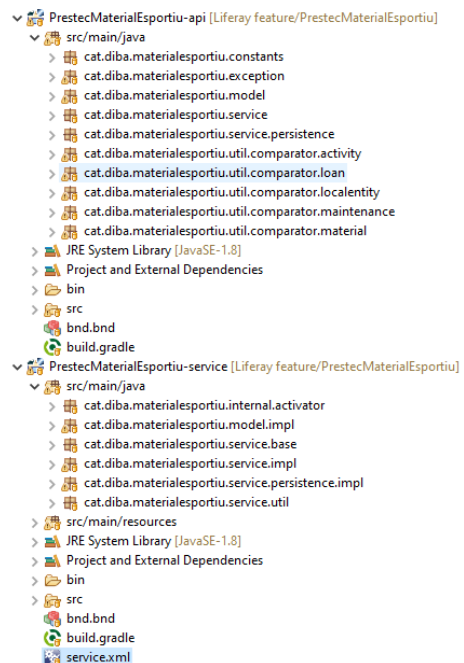


Ilustración 17: Estructura generada tras Service-Builder

El siguiente paso será hacer un despliegue de los dos módulos generados (api-service) en nuestro servidor. Podemos hacerlo como en la siguiente imagen o arrastrándolos a nuestro servidor.

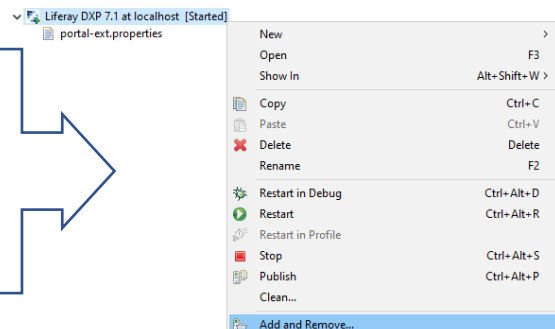
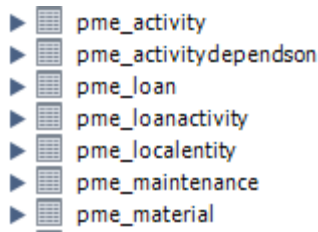


Ilustración 18: Despliegue módulos Api i Service



Una vez se haya desplegado, podremos comprobar como en nuestra BBDD se han generado las tablas que hemos definido en el fichero services.xml

Ilustración 19: Tablas generadas por Service-Builder

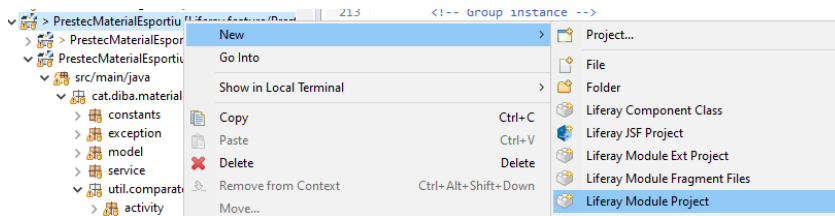
3.4.2 Back Office o Admin (panel-app¹⁶)

Como he explicado en puntos anteriores, desde el Back Office o panel de administración, el departamento de préstamos podrá llevar a cabo la gestión de entidades.

3.4.2.1 Creación módulo Admin

Para generar el módulo de administración generamos un nuevo proyecto de la siguiente manera, dentro de nuestro proyecto existente “PrestecMaterialEsportiu”.

Ilustración 20: Creación módulo Admin con panel-app (Paso1)



Indicaremos el nombre que queramos, con nuestra versión de Liferay y esta vez, seleccionamos “panel-app” en Project Template. Una vez tengamos generado nuestro proyecto, el nuevo módulo se habrá agregado a nuestra estructura, que quedará de la siguiente manera.

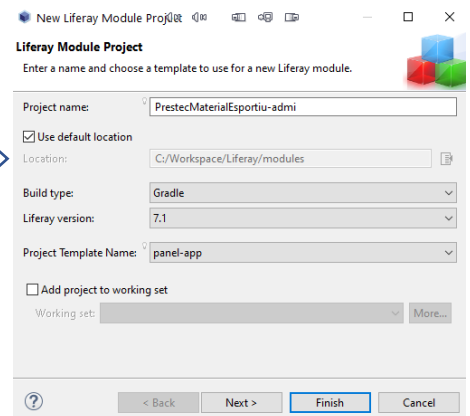


Ilustración 22: Creación módulo Admin con panel-app (Paso2)



Ilustración 21: Estructura generada tras la creación del módulo Admin

- ❖ PrestecMaterialEsportiu
 - ❖ PrestecMaterialEsportiu-api
 - ❖ PrestecMaterialEsportiu-service
 - ❖ PrestecMaterialEsportiu-admin
 - ❖ cat.diba.materialesportiu.admin
 - ❖ application.list
 - ❖ constants
 - ❖ portlet
 - ❖ display.context

3.4.2.2 Menús

Dentro de nuestra estructura, encontraremos la carpeta resources, es aquí donde crearemos nuestras pantallas de menú mediante ficheros jsp¹⁷ y estilos scss¹⁸.

En este caso, he separado cada uno de los menús por carpetas para una mejor organización, dentro de cada carpeta podemos encontrar las pantallas a las que tendremos acceso.

También hemos añadido varios ficheros Language para los diferentes idiomas del sitio web.

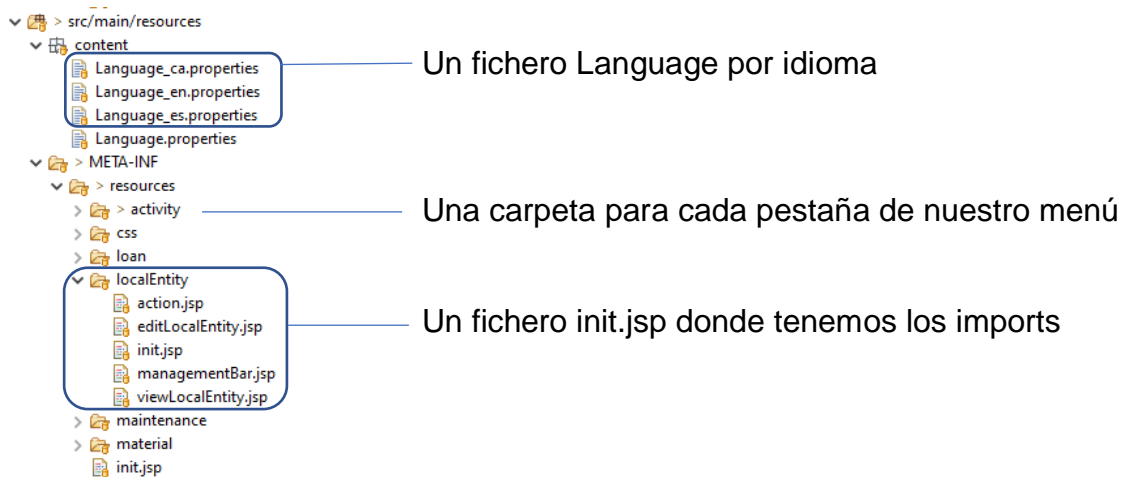


Ilustración 23: Estructura de la carpeta Resources de service

3.4.2.2.1 Mantenimiento

Desde esta sección podemos visualizar el material que ha estado, está o estará en mantenimiento. Después de cada préstamo el material pasará a mantenimiento para realizar las correspondientes verificaciones.

También podemos realizar alta para un material en concreto sin necesidad que haya sido prestado con lo que podemos o no vincularlo a una entidad.

Además, tenemos un estado donde podemos ver si el mantenimiento se está realizando o ya se ha realizado.

Ilustración 24: Pantalla Mantenimiento Principal

Activitat	Etiqueta	Descripció	Data inici	Data fi	Estat	Préstec	Creat	Modificat
Bastons d'invidents	Bastons d'invidents	trencat	17/05/2020	17/05/2020			17/05/2020	17/05/2020

Si queremos dar de alta un mantenimiento accederemos a la siguiente pantalla.

Ilustración 25: Pantalla Mantenimiento Alta

Activitat *

Etiqueta *

Descripció *

Data d'inici
19/05/2020

Data de finalització
19/05/2020

Vinculat *

Desa Cancel·la

Si queremos modificar un registro accederemos a la siguiente pantalla.

Ilustración 26: Pantalla Mantenimiento Editar

Id 81801

Activitat *
Bastons d'invidents

Etiqueta
Bastons d'invidents

Descripció *
trenca

Data d'inici
17/05/2020

Data de finalització
17/05/2020

Vinculat

Desa Cancel·la

Desde la ventana principal podemos acceder al menú de acciones para cada uno de los registros dados de alta, desde ahí podemos marcar un mantenimiento como hecho, con lo que pasará a estar disponible antes de la fecha indicada.

Desde este menú también podemos editar y eliminar registros.

Ilustración 27: Pantalla Mantenimiento Acciones

Activitat	Etiqueta	Descripció	Data inici	Data fi	Estat	Préstec	Creat	Modificat
Bastons d'invidents	Bastons d'invidents	trenca	17/05/2020	17/05/2020				<ul style="list-style-type: none">✓ Marca-ho com a fet Edita Esborra

3.4.2.2.2 Prestamos

Desde esta sección gestionaremos las reservas realizadas ya sean hechas desde web o desde este mismo módulo. Podemos ver los préstamos y sus diferentes estados en función de si han sido confirmados y las fechas. Los diferentes estados para las reservas son:

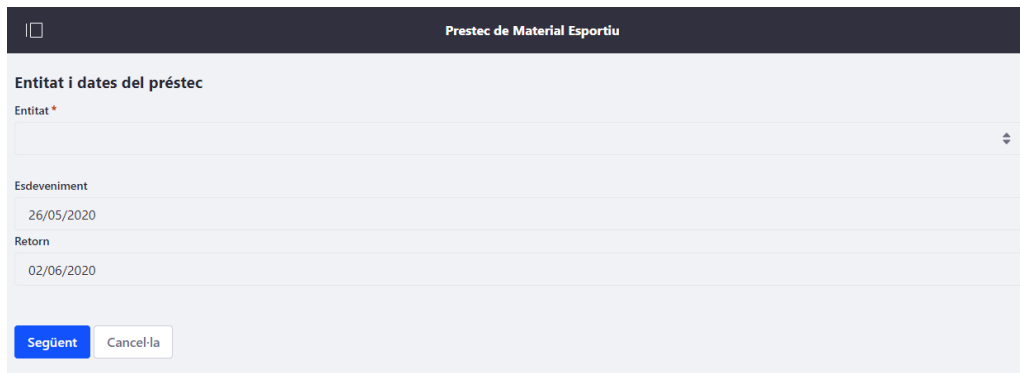
- No confirmada: Ha sido reservado, no ha excedido el número de días de bloqueo de reserva y el día de recogida es posterior a la fecha actual.
- Confirmada: Ha sido confirmada, el día de recogida es posterior a la fecha actual.
- Expirada: No se ha confirmado y se ha superado el número de días de bloqueo de reserva o el día de recogida ha pasado.
- Recogida hoy: El material está confirmado y se recoge hoy.
- Prestado: El material está confirmado y la fecha actual comprende la fecha de recogida y devolución.
- Devolución hoy: El material está prestado y se devuelve hoy.
- Devuelto: El material ha sido devuelto.

Ilustración 28: Pantalla Préstamo Principal

Entitat	Estat	Recollida	Esdeveniment	Retorn	Reservat per	Telèfon	Correu electrònic	PMT	Creat	Modificat
Barcelona		12/05/2020	15/05/2020	16/05/2020	Alberto	686880837	albertofgaldo@gmail.com		17/05/2020	17/05/2020
Barcelona		12/05/2020	17/05/2020	17/05/2020	Alberto	686880837	albertofgaldo@gmail.com		17/05/2020	17/05/2020
Barcelona		12/05/2020	17/05/2020	19/05/2020	Alberto	686880837	albertofgaldo@gmail.com		17/05/2020	17/05/2020
Barcelona		27/05/2020	01/06/2020	05/06/2020	asdf	686880837	albertofgaldo@gmail.com		17/05/2020	17/05/2020
Barcelona		17/05/2020	19/05/2020	21/05/2020	Alberto	686880837	albertofgaldo@gmail.com		17/05/2020	17/05/2020
Barcelona		20/05/2020	25/05/2020	29/05/2020	Alberto	686880837	albertofgaldo@gmail.com		17/05/2020	17/05/2020

El departamento también puede realizar reservas desde este módulo, para ello deberá indicar las fechas de la actividad y devolución, además de la entidad que va a realizar la reserva.

Ilustración 29: Pantalla Préstamo Alta 1/2

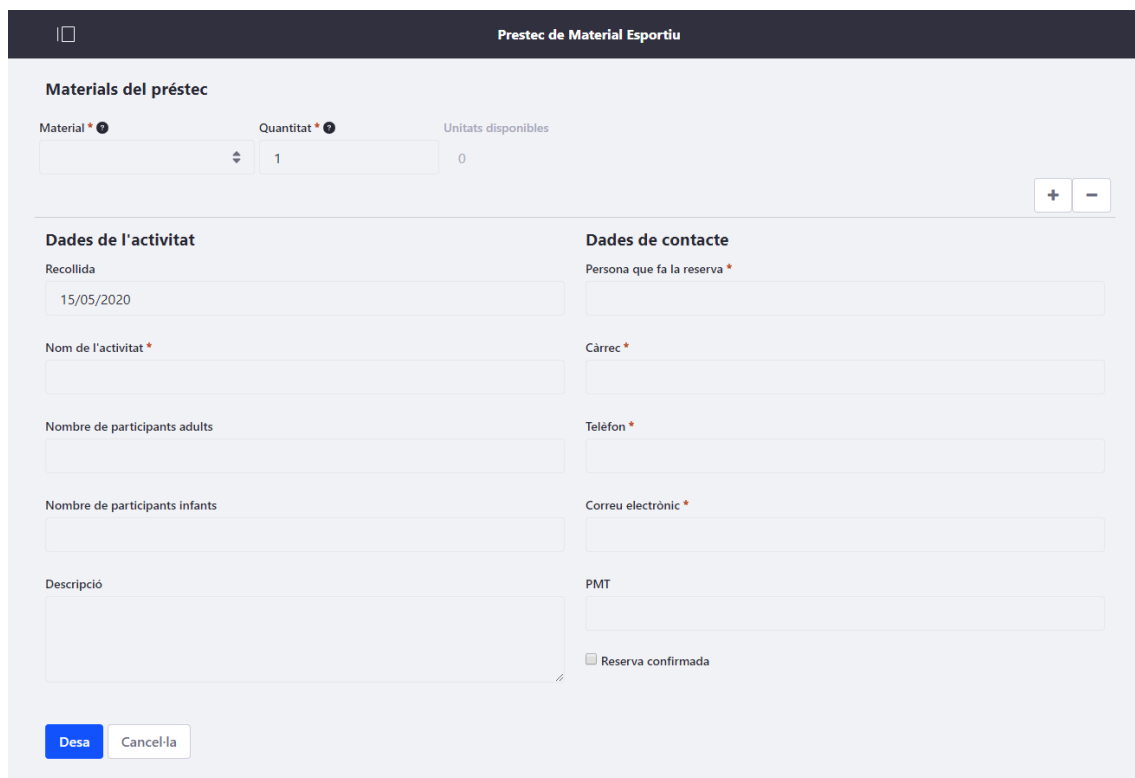


Esto nos devolverá los materiales disponibles para esas fechas, el sistema tendrá en cuenta las reservas y mantenimientos de esas mismas fechas para calcular el stock.

También deberemos introducir los datos restantes, el sistema calculará la fecha de recogida en función al campo daysBeforeLoan (entidad Activity) con mayor valor de nuestros materiales seleccionados.

Al pulsar en guardar el sistema crea una reserva.

Ilustración 30: Pantalla Préstamo Alta 2/2



Podremos modificar los datos de las reservas creadas con anterioridad desde la siguiente ventana.

Ilustración 31: Pantalla Préstamo Editar

Prestec de Material Esportiu

Prestec 81813

Materials del préstec

Activitat * ● Quantitat * ● Unitats disponibles

Aualè - Joc Intercultural (16) 3 16

+ -

Dades de l'activitat

Recollida
12/05/2020

Nom de l'activitat *
asdf

Nombre de participants adults
0

Nombre de participants infants
0

Descripció
08020

Dades de contacte

Persona que fa la reserva *
Alberto

Càrrec *
tècnic

Telèfon *
686880837

Correu electrònic *
albertofgaldo@gmail.com

PMT

Reserva confirmada

Desa Cancel·la

3.4.2.2.3 Material

Este módulo hace referencia a los materiales que tiene cada actividad y la cantidad correspondiente. Esta entidad la utilizaremos para realizar una estimación del volumen y peso que conforma una reserva ya que, cada material dispone de medidas y peso.

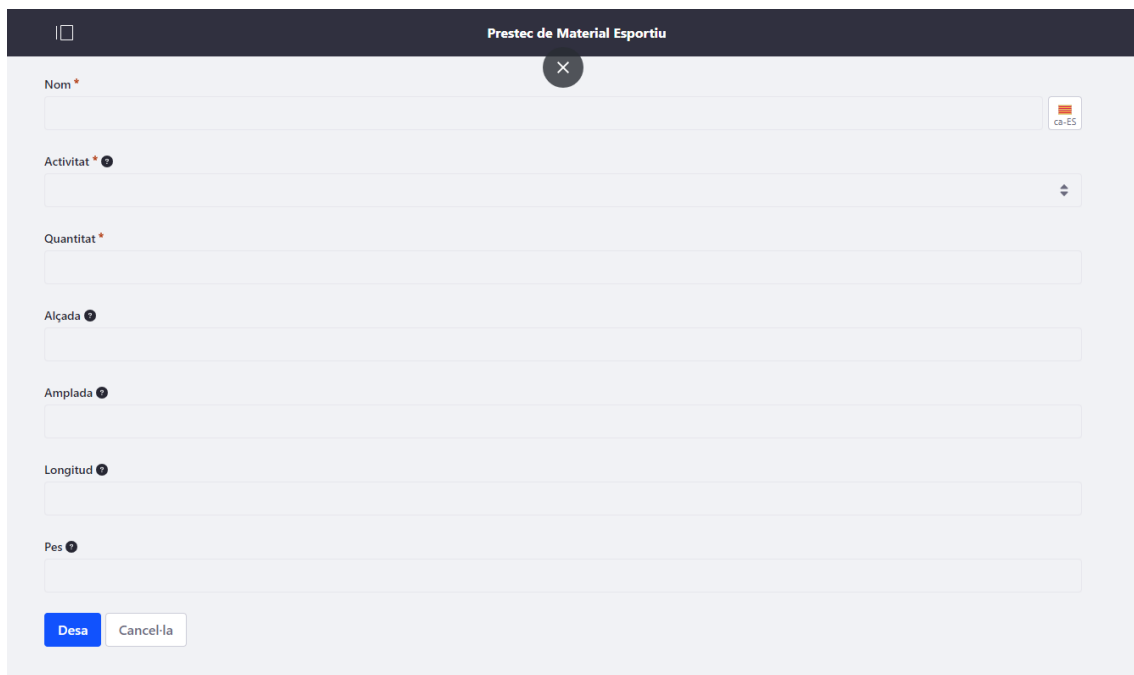
Il·lustració 32: Pantalla Material Principal



Nom	Volum	Alçada	Amplada	Longitud	Pes	Creat	Modificat
Tauler d'auale (amb sis forats i dos guarides)	0.0m ³	0.0cm	0.0cm	0.0cm	0.0kg	11/05/2020	11/05/2020
Llavors	0.0m ³	0.0cm	0.0cm	0.0cm	0.0kg	11/05/2020	11/05/2020
Caçaboles	0.0m ³	0.0cm	0.0cm	0.0cm	0.0kg	11/05/2020	11/05/2020

Vemos los datos necesarios para crear un material, que corresponderá a una actividad determinada. En caso de no indicar medidas, estas obtendrán un valor por defecto. La cantidad de unidades es obligatoria.

Il·lustració 33: Pantalla Material Alta



Nom *

Activitat *

Quantitat *

Alçada *

Amplada *

Longitud *

Pes *

Desa Cancel·la

Como sucede con los otros módulos, desde este también podemos modificar los datos que hemos introducido previamente.

Il·lustració 34: Pantalla Material Editar

The screenshot shows a web form titled "Prestec de Material Esportiu" with a dark header bar. The form contains the following fields and controls:

- Id 41091**: A label for the item ID.
- Nom ***: A text input field containing "Tauler d'aualé (amb sis forats i dos guarides)". To the right is a small icon with "ca-ES".
- Activitat ***: A dropdown menu showing "Aualé - Joc Intercultural".
- Quantitat ***: A text input field containing "1".
- Alçada ***: A text input field containing "0,0".
- Amplada ***: A text input field containing "0,0".
- Longitud ***: A text input field containing "0,0".
- Pes ***: A text input field containing "0,0".
- At the bottom left, there are two buttons: "Desa" (Save) in blue and "Cancel·la" (Cancel) in white.

3.4.2.2.4 Actividades

En este módulo creamos las actividades, desde la página web o a la hora de realizar las reservas nos referimos a estas actividades como Material y es lo que prestaremos a las entidades interesadas, además, tiene un stock necesario para saber si hay material disponible para realizar el préstamo en las fechas solicitadas.

Desde esta ventana principal, además del stock total y el stock disponible a día de hoy, podemos ver otros datos como los días previos necesarios para la recogida, días que se bloqueará la reserva hasta la confirmación o días de mantenimiento tras el préstamo.

Cuando una reserva tenga más de un material, el valor de días previos de recogida que se atribuirá a esa reserva será el más alto de entre los materiales reservados.

Algunas actividades, como vemos en la imagen, requieren de titulación.

Ilustración 35: Pantalla Actividades Principal



Nom	Total	Dispo.	Recollida	Reserva	Mante.	Edat	Min. parti.	Mox. parti.	Titulació	Creat	Modificat
Aualè - Joc Intercultural	21	21	1 dies	5 dies	3 dies	8	2	2		11/05/2020	17/05/2020
Bastons d'invidents	10	9	1 dies	5 dies	3 dies	6	10	20		11/05/2020	20/05/2020
Bagul "Jocs del Món"	2	2	1 dies	4 dies	3 dies	7	1	30		10/05/2020	14/05/2020

Desde la siguiente pantalla vemos como se crea una actividad, además de los atributos comentados antes, vemos también que una actividad puede depender de otra (Actividad padre).

Il·lustració 36: Pantalla Activitats Alta

The screenshot shows a web form titled "Prestec de Material Esportiu" for creating a new activity. The form includes several text input fields: "Nom" (Name), "Descripció" (Description), "Regles d'ús" (Usage rules), and "Condicions de préstec" (Loan conditions). Below these are several numeric input fields: "Edat" (Age), "Mínim de participants" (Minimum participants), "Màxim de participants" (Maximum participants), and "Estoc total" (Total stock). There is also a checkbox for "Titulació necessària" (Necessary qualification). Further down, there are three more numeric input fields: "Dies previs al préstec per a recollir el material" (Days before loan to collect material), "Dies en que el material romandrà en estat 'reservat'" (Days when material will be in 'reserved' status), and "Dies necessaris pel manteniment després de la devolució" (Days necessary for maintenance after return). A dropdown menu for "Activitats pare" (Parent activities) is visible, with options: "Aualè - Joc Intercultural", "Bagul 'Jocs del Món'", and "Bastons d'invidents". At the bottom, there are two buttons: "Desa" (Save) and "Cancel·la" (Cancel).

Pantalla de edició, desde dónde podemos modificar actividades que hemos creado con anterioridad.

Il·lustració 37: Pantalla Actividad Editar

The screenshot shows the same "Prestec de Material Esportiu" form, but in edit mode. The activity ID "Activitat 39589" is displayed at the top. The "Nom" field contains "Bastons d'invidents". The "Descripció" field contains the text: "Material de sensibilització vers la limitació que pateixen les persones amb ceguesa, amb un objectiu clar: conscienciar la població que el fet de patir una discapacitat no té perquè impedir fer una vida normal. Amb aquest material es pot convertir qualsevol equipament educatiu o espai públic del municipi en un escenari lúdic per treballar la sensibilització i la". The "Regles d'ús" field contains "10 persones (o 20, si es converteix la modalitat en un grup de 10 persones fent de guia dels altres 10 participants)". The "Condicions de préstec" field contains two bullet points: "• Habilitar un espai per poder fer l'activitat." and "• Disposar de dos educadors per dinamitzar l'activitat.". The numeric fields are populated: "Edat" is 6, "Mínim de participants" is 10, "Màxim de participants" is 20, and "Estoc total" is 10. The "Titulació necessària" checkbox is checked. The other numeric fields are: "Dies previs al préstec per a recollir el material" is 1, "Dies en que el material romandrà en estat 'reservat'" is 5, and "Dies necessaris pel manteniment després de la devolució" is 3. The "Activitats pare" dropdown menu is open, showing "Aualè - Joc Intercultural" and "Bagul 'Jocs del Món'". At the bottom, there are two buttons: "Desa" (Save) and "Cancel·la" (Cancel).

3.4.2.2.5 Entidades locales

Las entidades locales las necesitaremos para validar que quién realiza la reserva a través de la web es una entidad local, para ello, necesitaremos dar de alta todas ellas en nuestra BBDD.

Estas entidades se validarán a través de su CIF para poder realizar reservas.

Ilustración 38: Pantalla Entidades locales Principal



Desde esta pantalla realizaremos las altas de las entidades locales.

Ilustración 39: Pantalla Entidades locales Alta

Menú

CIF *

Nom *

Telèfon *

Correu electrònic *

Desa Cancel·la

Como en todos los módulos, también podemos realizar modificaciones.

Ilustración 40: Pantalla Entidades locales Editar

Entitat 34519

CIF *

000000A

Nom *

Barcelona

Telèfon *

98765432

Correu electrònic *

barcelona@bcn.cat

Desa Cancel·la

3.4.2.3 portlet-model-hints.xml

En este fichero podemos definir el formato con el que trabajaremos en los campos de cada entidad.

Por ejemplo, para los campos multilinguaje que utilizan la propiedad “locale”, debemos aumentar el tamaño, ya que guardará la información duplicándola tantas veces como el número de idiomas con el que trabajemos. En nuestro caso serán 3 idiomas (ES, CA, EN) y añadiremos la siguiente propiedad a los campos deseados.

```
<field localized="true" name="description" type="String">  
  <hint name="max-length">4000</hint>  
</field>
```

Ilustración 41: Definir max-length en fichero portlet-hints.xml

Otro factor a tener en cuenta son las fechas, en nuestro caso omitiremos la hora, con lo que podemos indicarles a los campos Date que no la necesitamos.

```
<field name="pickupDate" type="Date">  
  <hint name="show-time">false</hint>  
</field>  
<field name="returnDate" type="Date">  
  <hint name="show-time">false</hint>  
</field>  
<field name="activityDate" type="Date">  
  <hint name="show-time">false</hint>  
</field>
```

Ilustración 42: Ocultar hora para fechas en fichero portlet-hints.xml

3.4.2.4 Métodos adicionales

Para poder gestionar las entidades que hemos creado necesitamos métodos de creación, lectura, actualización y borrado, estos son los llamados CRUD¹⁹. Estos métodos los encontramos ya creados en la clase xxServiceUtil.java, pero si queremos añadir otro podemos hacerlo desde la clase LocalServiceImpl.java

Por ejemplo, para la entidad LocalEntity queremos añadir métodos de búsqueda y modificar el método addLocalEntity, lo haremos entonces desde LocalEntityLocalServiceImpl.java.

- ❖ PrestecMaterialEsportiu
 - ❖ cat.diba.prestecmaterialesportiu
 - ❖ service
 - ❖ impl
 - ❖ LocalEntityLocalServiceImpl.java
 - ❖

Añadimos el método addLocalEntity, el cual recibirá por parámetros el objeto LocalEntity que previamente hemos generado desde la vista.

En este método en concreto, Liferay nos proporciona una herramienta para el contador de nuestras instancias. Desde CounterLocalService podemos generar un ID y asignarlo a nuestro nuevo objeto para crearlo en BBDD.

```
public class LocalEntityLocalServiceImpl extends LocalEntityLocalServiceBaseImpl {  
    /*  
    * NOTE FOR DEVELOPERS:  
    *  
    * Never reference this class directly. Always use {@link cat.diba.materialesportiu.service.LocalEntityLocalServiceUtil}  
    */  
    public LocalEntity addLocalEntity(LocalEntity localEntity) throws SystemException{  
        CounterLocalService counterLocalService = getCounterLocalService();  
        localEntity.setLocalEntityId(counterLocalService.increment());  
        return super.addLocalEntity(localEntity);  
    }  
  
    public List<LocalEntity> getLocalEntityByName(String name, int start, int end) throws SystemException{  
        return getLocalEntityPersistence().findByName(name, start, end);  
    }  
  
    public List<LocalEntity> getLocalEntityByCompanyGroup(long companyId, long groupId, int start, int end, OrderByComparator  
        return getLocalEntityPersistence().findByCompanyGroup(companyId, groupId, start, end, obc);  
    }  
  
    public int getCountByCompanyGroup(long companyId, long groupId) throws SystemException{  
        return getLocalEntityPersistence().countByCompanyGroup(companyId, groupId);  
    }  
}
```

Ilustración 43: Definir métodos adicionales en las clases xxxImpl.java

El resto de métodos que vemos en esta clase hacen referencia a finders que previamente creamos en nuestro fichero services.xml. Podemos ver como dentro del método llamamos a nuestro finder a través de getLocalEntityPersistence().nuestroFinder()

3.4.2.5 Custom Query

En algunos casos, los Finders que nos proporciona Liferay se quedarán cortos en cuanto a funcionalidad. Para solventar este problema podemos recurrir a las Custom Query, consultas de sql que definiremos en un fichero y se ejecutarán a través de un finder que crearemos manualmente.

Los pasos a seguir son:

Definir la Finder en el package `.service.persistence.impl`, en nuestro caso sería la clase `ActivityFinderImpl.java` en el package `cat.diba.materialesportiu.service.persistence.impl`

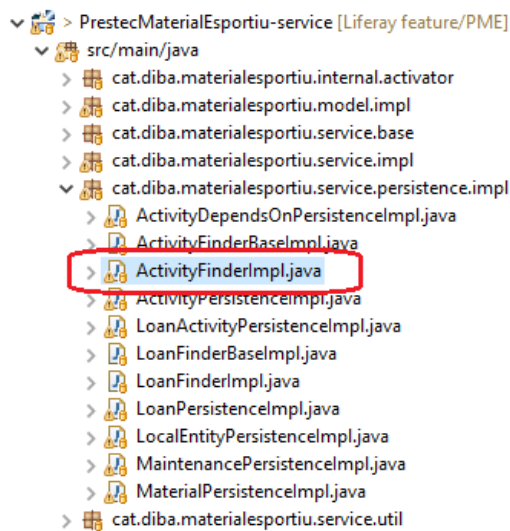
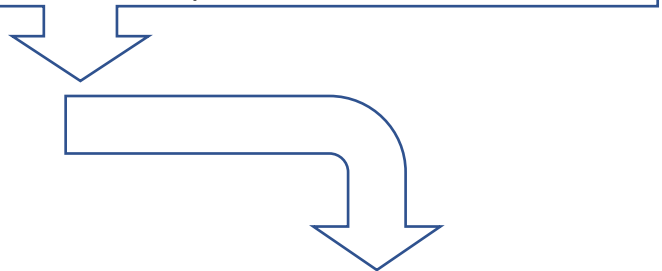


Ilustración 44: Creación Custom Query (Paso1)



Dentro de esta nueva clase, extendemos de la clase `ActivityFinderBaseImpl` y lanzamos `Service-Builder`.

```
package cat.diba.materialesportiu.service.persistence.impl;
import com.liferay.portal.dao.orm.custom.sql.CustomSQL;

public class ActivityFinderImpl extends ActivityFinderBaseImpl
```

Ilustración 45: Creación Custom Query (Paso2)

Esto hará que Liferay cree las clases restantes necesarias para la Custom Query.

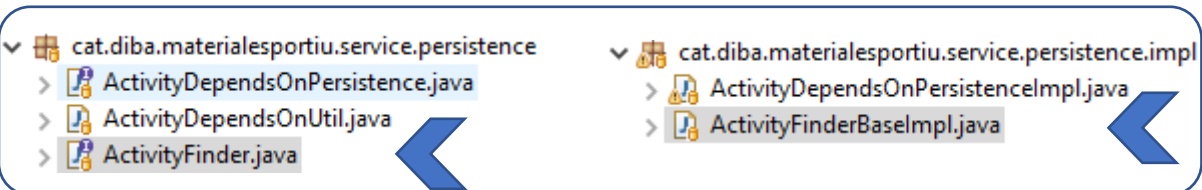


Ilustración 46: Creación Custom Query (Paso3)

Ahora crearemos un nuevo fichero en la carpeta “resourcer/custom-sql/default.xml”, en este fichero construiremos nuestras custom queries con SQL e indicaremos los parámetros que vamos a pasar (?), también el nombre del finder por el cual llamaremos a nuestra Custom Query.

```
<?xml version="1.0" encoding="UTF-8"?>
<custom-sql>
  <sql id="cat.diba.materialesportiu.service.persistence.ActivityFinder.findByActivitiesWithStockBetweenDates">
<![CDATA[
  select * from
  lportal.pme_activity a
  where
  (a.totalStock - (
  select SUM(la.quantity) from lportal.pme_activity a
  inner join lportal.pme_loanactivity la on a.activityId=la.activityId
  inner join lportal.pme_loan l on la.loanId=l.loanId
  where
  (? between l.pickupDate and adddate(l.returnDate, a.daysAfterLoan) or
  ? between l.pickupDate and adddate(l.returnDate, a.daysAfterLoan))))
  > 0
]]>
  </sql>
</custom-sql>
```

Ilustración 47: Creación Custom Query (Paso4)

Por último, podemos terminar de implementar la clase generada al inicio. Vincularemos también la consulta generada en el punto anterior mediante el nombre del finder creado, además, le pasamos por parámetros las variables que necesitamos para realizar la consulta (?).

```
package cat.diba.materialesportiu.service.persistence.impl;

import com.liferay.portal.dao.orm.custom.sql.CustomSQL;

public class ActivityFinderImpl extends ActivityFinderBaseImpl implements cat.diba.materialesportiu.service.persistence.ActivityFinder {

    private static final String FIND_BY_ACTIVITIES_WITH_STOCK_BETWEEN_DATES =
        ActivityFinder.class.getName() + ".findByActivitiesWithStockBetweenDates";

    @ServiceReference(type=CustomSQL.class)
    private CustomSQL _customSQL;

    @Override
    public List<Activity> findByActivitiesWithStockBetweenDates(Date startDate, Date endDate) {
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
        String startDate1 = format.format(startDate);
        String endDate2 = format.format(endDate);

        Session session = null;
        try {
            session = openSession();
            String sql = _customSQL.get(getClass(), FIND_BY_ACTIVITIES_WITH_STOCK_BETWEEN_DATES);
            SQLQuery q = session.createSQLQuery(sql);
            q.setCacheable(false);
            q.addEntity("Activity", ActivityImpl.class);

            QueryPos qPos = QueryPos.getInstance(q);
            qPos.add(startDate1);
            qPos.add(endDate2);
            return (List<Activity>)q.list();

        } catch (Exception e) {
            return new ArrayList<Activity>();
        } finally {
            closeSession(session);
        }
    }
}
```

Ilustración 48: Creación Custom Query (Paso5)

3.4.2.6 Scheduler

Para gestionar los estados de los préstamos y el stock de los materiales debemos actualizar las tablas de manera periódica. De esta manera mantendremos una congruencia entre actividades, préstamos y stock disponible. Realizaremos las siguientes modificaciones:

- ❖ Cambiar el estado de las reservas a “Expired” si estas han expirado.
- ❖ Cambiar el estado de las reservas a “Borrowed” si estas han sido recogidas.
- ❖ Cambiar el estado de los préstamos a “Maintenance” (para su revisión) si han sido devueltas.
- ❖ Cambiar el estado de los préstamos a “Returned” tras finalizar su revisión.
- ❖ Además, también deberán actualizar el stock cuando los materiales entren en estado “Borrowed” y “Returned” según sea su caso.

Para llevar a cabo esto deberemos crear una clase Scheduler que se encargará de actualizar nuestras tablas diariamente.

```
@Component(immediate = true, property = { "cron.expression=0 0 0 * * ?" }, // Scheduler run daily at 0AM
           service = SampleScheduler.class)
public class SampleScheduler extends BaseMessageListener {

    private static Log log = LogFactoryUtil.getLog(SampleScheduler.class);

    @Override
    protected void doReceive(Message message) throws Exception {

        LoanLocalServiceUtil.updateLoanStatus();

        // - to clear all the Database caches
        //CacheRegistryUtil.clear();

        // - clearing cache across JVM clusters
        MultiVMPoolUtil.clear();

    }

    @Activate
    @Modified
    protected void activate(Map<String, Object> properties) throws SchedulerException {

        try {
            String cronExpression = GetterUtil.getString(properties.get("cron.expression"), "cronExpression");
            log.info(" cronExpression: " + cronExpression);

            String listenerClass = getClass().getName();
            Trigger jobTrigger = TriggerFactoryUtil.createTrigger(listenerClass, listenerClass, new Date(), null,
                cronExpression);

            SchedulerEntryImpl schedulerEntryImpl = new SchedulerEntryImpl(listenerClass, jobTrigger);
            SchedulerEngineHelperUtil.register(this, schedulerEntryImpl, DestinationNames.SCHEDULER_DISPATCH);

        } catch (Exception e) {
            log.error(e);
        }

    }

    @Deactivate
    protected void deactivate() {
        SchedulerEngineHelperUtil.unregister(this);
    }

}
```

Método para desactivar el proceso.
Método para activar el proceso, crea el disparador y carga la configuración.
Método dónde se indican las acciones a realizar. Update y limpieza de caché.
"cron.expression=0 0 0 * * ?" -> El proceso se ejecutará diariamente a las 00:00h.

Ilustración 49: Creación de la clase Scheduler (Paso1)

La función que llamamos dentro del método doReceive contiene varias llamadas a unas Custom Query que hemos creado previamente y que realizan las actualizaciones que hemos comentado al principio de este apartado.

```
public void updateLoanStatus() {  
    loanFinder.updateLoanStatusToExpired();  
    loanFinder.updateLoanStatusToBorrowed();  
    loanFinder.updateLoanStatusToReturned();  
    loanFinder.updateMaintenanceDone();  
}
```

Ilustración 50: Creación de la clase Scheduler (Paso2)

Aquí uno de los métodos Custom Query que llama la función anterior

```
@Override  
public void updateLoanStatusToExpired() {  
    Session session = null;  
    try {  
        session = openSession();  
        String sql = _customSQL.get(getClass(), UPDATE_LOAN_STATUS_TO_EXPIRED);  
        SQLQuery q = session.createSQLQuery(sql);  
        q.setCacheable(false);  
        q.executeUpdate();  
    } catch (Exception e) {  
    }  
  
    } finally {  
        closeSession(session);  
    }  
}
```

Ilustración 51: Creación de la clase Scheduler (Paso3)

Y esta llama a la query que hemos creado en el fichero default.xml y que ya hemos explicado en el punto Custom Query anterior

```
<sql id="cat.diba.materialesportiu.service.persistence.LoanFinder.updateLoanStatusToExpired">  
<![CDATA[  
    update lportal.pme_loan l  
    inner join (  
        select la.loanId, MIN(a.daysBooking)days from lportal.pme_activity a  
        inner join lportal.pme_loanactivity la on a.activityId=la.activityId  
        group by la.loanId  
    ) ala on l.loanId=ala.loanId  
    set l.status='EXPIRED'  
    where l.status like 'NOTCONFIRMED' and  
    (l.pickupDate < localtime()  
    or  
    adddate(  
        l.createDate,  
        ala.days  
    )  
    )  
]]>  
</sql>
```

Ilustración 52: Creación de la clase Scheduler (Paso4)

3.4.2.7 Language properties

Nuestra aplicación puede trabajar con varios idiomas, esto es útil cuando el usuario quiere cambiar el idioma del portal web, automáticamente el sistema carga el fichero del idioma que el usuario a seleccionado. Para que esto funcione necesitamos crear un fichero por cada idioma que pondremos a disposición del usuario y, por supuesto, definir todos los términos en cada uno para su idioma correspondiente.

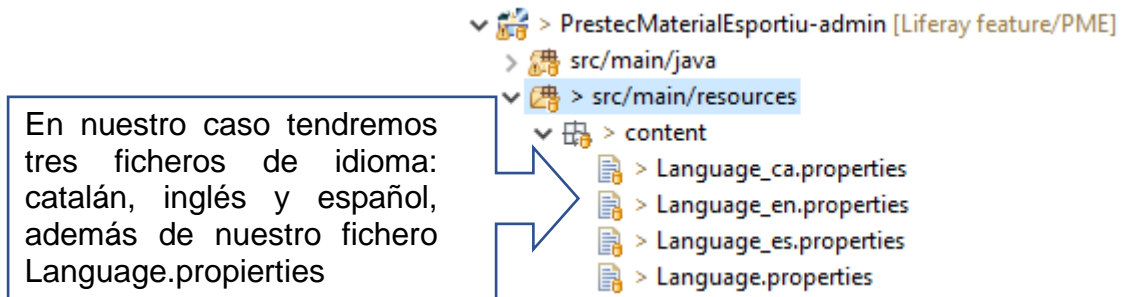


Ilustración 53: Ficheros Language.properties

Este es el aspecto de una parte de nuestro fichero Language.properties:

```
#category.custom.label=Category Custom Label
javax.portlet.display-name.prestecmaterialesportiuadmin=PrestecMaterialEsportiuAdmin
javax.portlet.keywords.prestecmaterialesportiuadmin=PrestecMaterialEsportiuAdmin
javax.portlet.short-title.prestecmaterialesportiuadmin=PrestecMaterialEsportiuAdmin
javax.portlet.title.prestecmaterialesportiuadmin=PrestecMaterialEsportiuAdmin
prestecmaterialesportiuadmin.caption>Hello from PrestecMaterialEsportiuAdmin!
#Titles
title-local-entity=Ens locals
title-loan=Préstecs
title-activity=Activitats
title-material=Materials
title-maintenance=Manteniment
#=====
#LOCAL ENTITY=====
#=====
#View/New/Edit-----
localEntityId=Entitat
cif=CIF
name=Nom
telephone=Telèfon
email=Correu electrònic
createDate=Creat
modifiedDate=Modificat
#Barra búsqueda-----
add-localEntity=Afegeix entitat
#Messages-----
no-results-localEntity=No s'han trobat ens locals
localEntity-breadcrumb-edit-localEntity=Edit Local Entity {0}
#=====
#LOAN=====
#=====
#View-----
loanId=Préstec
local-entity-id=Entitat
isConfirmed=Estat
pickupDate=Recollida
activityDate=Esdeveniment
returnDate=Retorn
contactPerson=Reservat per
pmt=PMT
#New/Edit-----
loan-data=Entitat i dates del préstec
loan-materials=Materials del préstec
loan-material=Material
max-quantity=Unitats disponibles
loan-info=Dades del préstec
loan-activity=Dades de l'activitat
loan-contact=Dades de contacte
participants-num=Nombre de participants adults
participants-kids-num=Nombre de participants infants
activity-date=Data de l'activitat
activity-name=Nom de l'activitat
contact-person=Persona que fa la reserva
position-person=Càrrec
is-confirmed=Reserva confirmada
#Barra búsqueda-----
add-loan=Afegeix préstec
#Messages-----
no-results-loan=No s'han trobat préstecs
entity-combo-text=Selecciona una entitat de la llista
#Help-----
help-loan-activity=Activitats disponibles (estoc disponible)
help-loan-quantity=Unitats per aquest material
#Error-----
error-loan-min-quantity=La quantitat d'unitats ha de ser com a mínim 1
error-loan-max-quantity=La quantitat d'unitats no pot ser superior a l'estoc dispo
#Images-----
alt-loan-confirmed=Confirmat
alt-loan-not-confirmed=No confirmat
alt-loan-expired=La reserva ha expirat
alt-loan-returnDay=El material es retorna avui
alt-loan-retornat=Retornat
alt-loan-borrowing=Prestat
alt-loan-pickupToday=El material es recull avui
#Action.jsp-----
action-return-loan=Retornat
action-loan-activity=Materials del préstec
#=====
#ACTIVITY=====
#=====
#View-----
activityId=Activitat
totalStock=Total
availableStock=Dispo.
daysBeforeLoan=Recollida
daysBooking=Reserva
daysAfterLoan=Mante.
age=Edat
minParticipants=Min. parti.
maxParticipants=Màx. parti.
titulationIsRequired=Titulació
#Barra búsqueda-----
add-activity=Afegeix activitat
#Messages-----
no-results-activity=No s'han trobat activitats
activity-breadcrumb-edit-activity=Edit Activity {0}
#New/Edit-----
description=Descripció
min-participants=Mínim de participants
max-participants=Màxim de participants
usage-rules=Regles d'ús
loan-conditions=Condicions de préstec
titulation-is-required=Titulació necessària
total-stock=Estoc total
days-before-loan=Dies previs al préstec per a recollir el material
days-booking=Dies en que el material romandrà en estat "reservat"
days-after-loan=Dies necessaris pel manteniment després de la devolució
activities-activityDependsOn=Activitats pare
currentParentActivities=Selecciona les activitats pare per aquesta activitat
#Help-----
default-value=Si el camp es buit, el valor per defecte serà:
material-activity-list-help=Selecciona i desselecciona amb la tecla Ctrl pulsada
#Error-----
error-quantity-total-stock=El nombre d'unitats es superior a l'estoc del material
#Images-----
required-titulation=Si es necessària
not-required-titulation=No es necessària
#Action-----
materials=Materials de l'activitatmaterials
quantity-activity=Nombre d'unitats per aquest material
quantity-activity-activity=Unitats per activitat
```

Ilustración 54: Configuración fichero language.properties

3.4.3 Web (mvc-portlet)

3.4.3.1 Creación módulo Web

Para generar el módulo web, que será accesible desde la página web de la Diputación de Barcelona para que la gente interesada pueda realizar los préstamos, realizamos los siguientes pasos.

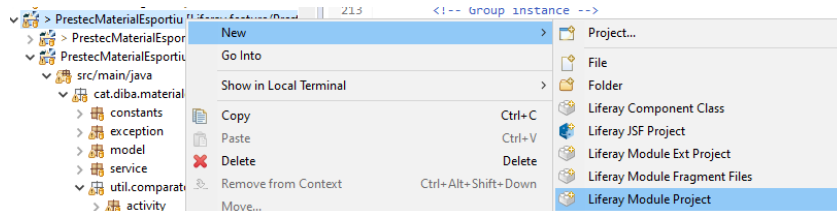


Ilustración 55: Creación módulo web mediante mvc-portlet (Paso1)

Indicaremos el nombre que queremos, con nuestra versión de Liferay y esta vez, seleccionamos “mvc-portlet” en Project Template.

Una vez tengamos generado nuestro proyecto, el nuevo módulo se habrá agregado a nuestra estructura, que quedará de la siguiente manera.

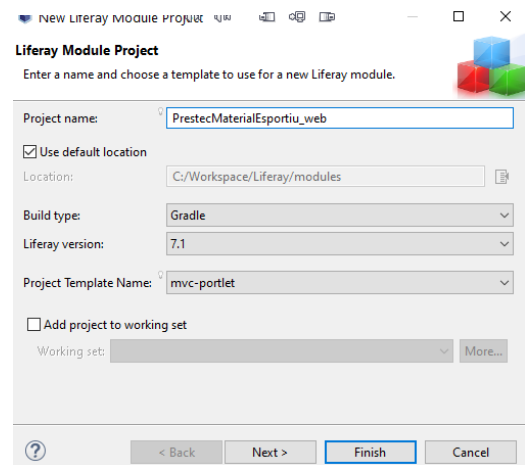


Ilustración 56: Creación módulo web mediante mvc-portlet (Paso2)




Ilustración 57: Creación módulo web mediante mvc-portlet (Paso3)

- ❖ PrestecMaterialEsportiu
 - ❖ PrestecMaterialEsportiu-api
 - ❖ PrestecMaterialEsportiu-service
 - ❖ PrestecMaterialEsportiu-admin
 - ❖ PrestecMaterialEsportiu-web
 - ❖ cat.diba.materialesportiu.web
 - ❖ constants
 - ❖ portlet

Este módulo web se incorporará en la página de la diputación, concretamente en la zona de préstamo de material, desde dónde las entidades interesadas podrán realizar dos tipos de búsquedas para conocer la disponibilidad del material. Estas búsquedas se explican a continuación.

3.4.3.2 Búsqueda por fechas

Esta primera búsqueda permite la posibilidad de conocer el material disponible para unas determinadas fechas que el interesado debe introducir previamente.



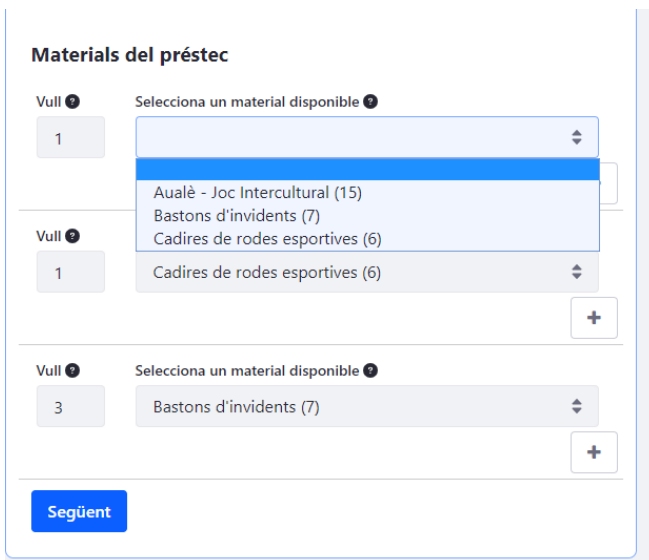
Il·lustración 58: Buscador por fechas de módulo web

Una vez introducimos estas fechas, el sistema mostrará el material y la cantidad de este que estará disponible en un desplegable.

Desde este desplegable podremos indicar el material que queremos y la cantidad correspondiente, claro está, sin superar el material disponible. Si superamos esta cantidad el sistema nos informará de ello y no nos permitirá avanzar.

En caso de que queramos añadir otro material, deberemos pulsar sobre el botón de agregar nuevo material.

Cuando tengamos seleccionado el material clicaremos en el botón siguiente, para finalizar la reserva.



Il·lustración 59: Selección de material resultante

3.4.3.3 Búsqueda por materiales

En caso de que no conozcamos las fechas o queramos saber la disponibilidad para uno o varios materiales utilizaremos esta búsqueda.

Material	Stock
<input type="checkbox"/> Aualè - Joc Intercultural	21
<input type="checkbox"/> Bagul "Jocs del Món"	2
<input type="checkbox"/> Bastons d'invidents	10
<input type="checkbox"/> Cadires de rodes esportives	12
<input type="checkbox"/> Diana de golf	7

Cerca

Il·lustració 60: Buscador por materiales del módulo web

Una vez indiquemos los materiales, el sistema nos mostrará un calendario con las fechas en común para esos materiales, además, nos mostrará la cantidad de material disponible para cada día.

3.4.3.4 Finalizar reserva

Después de seleccionar las fechas y el material utilizando cualquiera de las dos búsquedas, el siguiente paso será rellenar los datos restantes de nuestra reserva. Además de los datos obligatorios, el sistema comprobará que el CIF introducido corresponda con una Entidad Local dada de alta previamente en el sistema, si no es así, no podremos realizar la reserva.

Además de ver el material seleccionado, podemos ver el volumen y peso total del material, con lo que podemos hacernos una idea del transporte necesario para hacer la recogida.

Otro dato que se calcula en esta pantalla es la fecha de recogida, esta fecha corresponde a la fecha del evento menos el valor del campo daysBeforeLoan del material (entidad Activity) reservado y que tiene mayor valor.

PRESTEC DE MATERIAL ESPORTIU

Dates de la reserva

Data de l'activitat	Data de retorn	Data de recollida
05/06/2020	12/06/2020	04/06/2020

Materials del préstec

1	Cadires de rodes esportives	0.0m ³	0.0kg
3	Bastons d'invidents	0.0m ³	0.0kg
Volum i pes total de tots els materials		0.0m ³	0.0kg

Dades de l'activitat

Nom de l'activitat *
Activitat de prova

Nombre de participants adults
60

Nombre de participants infants
10

Descripció
Activitat per les festes

Dades de contacte

CIF de l'entitat local *
B4257869

Persona que fa la reserva *
Alberto

Càrrec *
Responsable d'events

Telèfon *
686880837

Correu electrònic *
albertofgaldo@gmail.com

Reserva Cancel·la


Il·lustració 61: Confirmació de reserva en mòdul web

Por último y tras insertar los datos anteriores, podemos ver un resumen de nuestra reserva, además de un código de referencia que se corresponde con el identificador de la tabla loan.

El usuario también tiene la opción de imprimir el comprobante de reserva para el día de la recogida.

PRESTEC DE MATERIAL ESPORTIU

Reservat

Ref: 116306
 [Imprimeix](#)

Dates de la reserva

Data de recollida	Data de l'activitat	Data de retorn
04/06/2020	05/06/2020	12/06/2020

Materials del préstec

1	Cadires de rodes esportives	0.0m ³	0.0kg
3	Bastons d'invidents	0.0m ³	0.0kg
Volum i pes total de tots els materials		0.0m ³	0.0kg

Dades de l'activitat	Dades de contacte
Nom de l'activitat: Activitat de prova	Entitat organitzadora: Sant Cugat
Nombre de participants adults: 60	Persona que fa la reserva: Alberto
Nombre de participants infants: 10	Càrrec: Responsable d'events
Descripció: Activitat per les festes	Telèfon: 686880837
	Correu electrònic: albertofgaldo@gmail.com

Il·lustración 62: Comprobante de reserva realizada

3.5 Pruebas del proyecto

Todas las pruebas realizadas se están llevando a cabo en un entorno en local, tras estas pruebas se moverá a un entorno de integración, para finalmente migrar a producción.

Una característica interesante desde el IDE de Liferay es la posibilidad de visualizar los cambios de manera inmediata a medida que vamos guardando nuestro trabajo. Esto es gracias a que automáticamente se encarga de desplegar los cambios en el servidor. Esto agiliza la corrección de posibles errores a medida que vamos realizando las pruebas.

3.5.1 Pruebas Back Office

Las pruebas descritas en este apartado se centran en la parte Back Office y con la que trabajará el departamento encargado de gestionar los préstamos y el resto de entidades relacionadas.

- ✓ Todas las pruebas anteriores han sido satisfactorias.

3.5.1.1 Altas/Modificaciones/Bajas

Se realizan pruebas de altas en todos los módulos, también modificaciones y bajas.

Se verifican los cambios en BBDD.

- ✓ Todas las pruebas anteriores han sido satisfactorias.

3.5.1.2 Reservas/Préstamos

Materiales y stock disponible

Al realizar una reserva lo primero que hacemos es indicar las fechas en las que necesitamos los materiales, debemos asegurarnos que muestra la disponibilidad real para los materiales en las fechas indicadas.

Estas pruebas se llevan a cabo realizando varias reservas, con diferentes o mismos materiales y realizando un conteo manual del stock que debería quedar. Además, debemos tener en cuenta dos factores en este recuento, los materiales que tenemos en mantenimiento para esas fechas y los materiales que dependen de otros materiales ya que, cuando dejemos un material padre, este incluirá todos los hijos, con lo que el stock de estos últimos también variará para futuras consultas.

- ✓ Todas las pruebas anteriores han sido satisfactorias.

Transición entre estados

Desde que creamos una reserva hasta que el material pasa por mantenimiento, esta recorre una serie de estados que deben activarse según corresponda, para comprobar cada uno de estos estados realizamos las siguientes pruebas.

No PMT: La reserva no se ha confirmado.









Sí PMT: La reserva se ha confirmado.

Método manual: Se necesita acción de usuario.

Método Scheduler: La clase Scheduler lanza el método de comprobación.

Método IF: La comprobación se hace con un condicional en el fichero jsp

{1,2..} Los diferentes casos en los que funciona.

	Estado	PMT	Prueba	Método
	No confirmada	No	Crear reserva.	Manual
	Expirada	No	{1} "Fecha de recogida" >= "Fecha actual". {2} Diferencia entre "Fecha creación" y "Fecha actual" > "daysBooking" con menor valor de los materiales reservados.	Scheduler
	Confirmada	Sí	Introducir PMT.	Manual
	Recogida hoy	Sí	No es estado, sólo muestra imagen. Cuando "Fecha de recogida" = "Fecha actual".	IF
	Prestado	Sí	"Fecha recogida" > "Fecha actual" y el estado actual de la reserva es "Confirmada".	Scheduler ó Manual
	Devolución hoy	Sí	No es estado, sólo muestra imagen. Cuando "Fecha devolución" = "Fecha actual".	IF
	Devuelto y revisando	Sí	"Fecha devolución" > "Fecha actual" y el estado actual de la reserva es "Prestado".	Scheduler ó Manual
	Devuelto y revisado	Sí	Cuando todos los materiales pertenecientes a esta reserva están revisados.	Scheduler

✓ Todas las pruebas anteriores han sido satisfactorias.



3.5.1.3 Mantenimientos

Además de las pruebas anteriores, para este módulo necesitamos verificar que decrementa el stock al añadir un mantenimiento y se incrementa al finalizar el mismo.

Método manual: Se necesita acción de usuario.

Método Scheduler: La clase Scheduler lanza el método de comprobación.

{1,2..} Los diferentes casos en los que funciona.

	Estado	Prueba	Método
	En revisión	{1} Alta desde mantenimiento. {2} Alta desde préstamos con estado "Prestado". {3} Alta automática tras fecha de devolución.	Scheduler ó Manual
	Hecho	{1} Marcamos el material como revisado. {2} Cuando "Fecha fin" > "Fecha actual"	Scheduler ó Manual

✓ Todas las pruebas anteriores han sido satisfactorias.

3.5.1.4 Stock

Cuando el material pasa a estado prestado, se decrementa el stock del material.

Cuando el material en mantenimiento pasa a estado "Hecho" se incrementa el stock del material.

También se han hecho pruebas para materiales que dependen de otros materiales, es decir, cuando se presta un material padre, el sistema debe decrementar el stock para los materiales hijos ya que también se prestan esos materiales.

En el caso del mantenimiento, el stock del material padre únicamente se incrementa cuando todos los hijos han finalizado la revisión de mantenimiento.

✓ Todas las pruebas anteriores han sido satisfactorias.

3.5.2 Pruebas Web

Las pruebas realizadas desde web, simulan el proceso que deberá hacer una entidad local para reservar uno o varios materiales.

3.5.2.1 Reservas mediante búsqueda por fechas

Esta reserva es muy similar a la que puede realizar el departamento desde el módulo admin, difiere básicamente en la interfaz gráfica y que, en este caso, la entidad debe indicar su CIF para así validar que existe en la BBDD. En caso de no existir, el sistema no dejará realizar la reserva.

Se realiza la búsqueda para varias fechas teniendo en cuenta los materiales y stock disponible como se ha hecho en las pruebas Back Office.

- ✓ Todas las pruebas anteriores han sido satisfactorias.

3.5.2.3 Confirmación de reserva

Se introducen los datos y un CIF inexistente en BBDD, el sistema no crea la reserva.

Se introduce un CIF correcto, el sistema genera la reserva y devuelve el resumen de esta.

La fecha de recogida se genera correctamente, se calcula mediante el valor máximo "daysBeforeLoan" de los materiales de la reserva y se resta a la fecha del evento.

El volumen y peso de los materiales de la reserva se calcula de manera correcta.

- ✓ Todas las pruebas anteriores han sido satisfactorias.

4. Conclusiones

4.1 General

Podemos destacar la importancia de realizar un trabajo previo al desarrollo, tal como el análisis, diseño, modelaje y tareas a desarrollar, así como definir unos tiempos y prioridades para cada una de ellas.

Esto nos ayuda a tener una idea más exacta de las dimensiones del proyecto, anticiparnos a los posibles problemas que puedan surgir, de manera que podamos analizarlos antes y tomar así decisiones que puedan ahorrarnos mucho tiempo.

En este caso en particular, algunas tareas de baja prioridad, como la búsqueda por materiales del módulo web, no se ha podido finalizar. Podemos asumirlo ya que las funcionalidades básicas del aplicativo se mantienen, es decir, se pueden realizar reservas mediante el módulo web a través de la búsqueda por fechas.

4.2 Método actual vs método desarrollado

Comparando la metodología de trabajo actual del departamento podemos destacar los siguientes beneficios:

- ✓ Mayor control del stock en todas sus vertientes (total, actual, entre fechas).
- ✓ La recepción de llamadas da paso a la reserva mediante el portal online, esto genera una menor carga de trabajo.
- ✓ Consulta de manera rápida el estado de las reservas y préstamos.

4.3 Objetivos logrados

BBDD

- ✓ Integrar con la BBDD existente las tablas necesarias para el control de préstamos.
- ✓ Creación de entidades
 - ✓ Material -> Material disponible para el préstamo
 - ✓ Actividades -> A las que pertenece el material
 - ✓ Entidades Locales -> Interesados en reservar material
 - ✓ Mantenimientos -> Revisión y puesta a punto del material devuelto
 - ✓ Préstamos -> Reservar y confirmación de material por parte de los entes

BackOffice

- ✓ Alta, baja y gestión del material.
- ✓ Alta, baja y gestión de las actividades.
- ✓ Alta, baja y gestión de las entidades locales.
- ✓ Mantenimiento de los materiales -> Material en revisión, fechas previstas...
- ✓ Confirmación de reserva, con lo que pasará a ser un préstamo.

Web (Actualmente en desarrollo)

- ✓ Búsqueda de material disponible mediante fechas.
- ❖ Agenda: Visualización de la disponibilidad del material mediante un calendario.
- ❖ Categorización del material disponible.

BackOffice y Web

- ✓ Gestión del stock: Modificar stock en función de reservas, mantenimientos...
- ✓ Gestión de disponibilidad: Mostrar la disponibilidad del material en función de los diferentes factores.
- ✓ Reservas: Crear una relación del material, para las fechas indicadas con la entidad interesada. Se aplicará la modificación de stock pertinente.

4.4 Ampliación

Mejoras que se pueden realizar de manera progresiva:

- ❖ Avisos vía mail de material disponible a entidades interesadas.
- ❖ Sugerencias de material similar cuando el principal no está disponible.
- ❖ Filtros y búsquedas más precisas en el módulo web mediante los atributos de Activity.
- ❖ Diseño por medio de hojas de estilo para el módulo web y back office.

4.5 Análisis final

A lo largo del proyecto se ha intentado seguir los plazos marcados en el diagrama de Gantt. Aunque el plan no se haya seguido minuciosamente, podemos decir que prácticamente todos los objetivos se han alcanzado.

El motivo por el que no se han podido cumplir algunos tiempos asignados a cada tarea es porque se ha invertido mucho de ese tiempo en investigación, ya que la tecnología con la que se ha desarrollado era desconocida para mí y, como en cualquier inicio, se requiere un tiempo de adaptación y aprendizaje.

Podríamos asegurar que cuanto más conocimiento y experiencia se tiene sobre el proyecto a desarrollar y las tecnologías que lo rodean, con mayor precisión podemos establecer los tiempos que requieren cada una de las tareas.

Cambios

Por otro lado, durante el desarrollo se ha tenido que readaptar la BBDD y con ello parte del código por algunos factores que no habíamos tenido en cuenta en el análisis previo.

Modificación atributo “isConfirmed” por “status”

De inicio se partió con un campo booleano “isConfirmed” para confirmar reservas en la tabla Loan. Este campo fue sustituido por un enum “status”, y gracias al cual podemos controlar el ciclo de vida de una reserva.

Reemplazo de Custom Query por métodos finder

Para que Liferay se encargue de la gestión íntegra de estados del aplicativo, se ha optado por crear finders para la búsqueda, comprobación y actualización, si procede, de los estados y stock del aplicativo. De esta manera no alteramos directamente la BBDD con el uso de update mediante Custom Query.

Sin implementar

En cuanto a algún objetivo que no se ha cumplido, como la segunda búsqueda desde el módulo web (búsqueda por material), podemos asumirlo ya que era una de las tareas de baja prioridad, puesto que tenemos la búsqueda principal y las reservas se pueden realizar.

5. Anexos

5.1 Instalación entorno de desarrollo

En primer lugar, deberemos instalar en nuestra máquina el software necesario que se especifica en el punto anterior. Los tres primeros puntos no vamos a explicarlos, para la instalación de Liferay Portal y Tomcat, comentar que este último ya está incluido en la descarga del portal Liferay, con lo que no será necesario instalarlo.

El portal de Liferay lo ubicaremos en una ruta que indicaremos a la hora de configurar nuestro servidor desde el IDE de Liferay, como muestra la siguiente imagen.

Una vez tengamos arrancado nuestro servidor, podemos acceder a localhost:8080 desde el navegador para configurar el acceso a BBDD.

Antes de introducir los datos, debemos crear la BBDD para nuestro portal. El nombre por defecto será "lportal", user: root, pass: root.

Desde localhost:8080 ya podemos introducir los datos de conexión a la BBDD y también el usuario para el portal Liferay.

Con esto, tendremos nuestro entorno listo para comenzar a desarrollar.

5.2 Problemas encontrados

Compatibilidades con Liferay 7.1

	Certified	
Application Server	<ul style="list-style-type: none"> • JBoss EAP 7.1 • JBoss EAP 7.2 • tcServer 4.0 • Tomcat 9.0 • Weblogic 12c Release 2 	<ul style="list-style-type: none"> • Websphere 8.5.5 Fix Pack 11+ • Websphere 9.0 • Wildfly 10.0 • Wildfly 11.0 • Wildfly 16.0
Database	<ul style="list-style-type: none"> • Amazon Aurora • DB2 10.5 • DB2 11.1 • Maria DB 10.2 • MySQL 5.7 • Oracle Database 12c Release 2 	<ul style="list-style-type: none"> • Oracle Database 19c • PostgreSQL 9.6 • PostgreSQL 10 • PostgreSQL 11 • SQL Server 2016 • Sybase ASE 16
Operating System	<ul style="list-style-type: none"> • Alpine Linux 3.8 • Amazon Linux 2 • CentOS 7 • Debian 9 • IBM AIX 7.1 • Oracle Linux 6 	<ul style="list-style-type: none"> • Oracle Linux 7 • Oracle Solaris 11 • RedHat Enterprise Linux 7 • SUSE Enterprise Linux 12 • Ubuntu LTS 16.04 • Windows Server 2016
Desktop Browser*	<ul style="list-style-type: none"> • Chrome (Latest) • Edge (Latest) • Firefox ESR 52 	<ul style="list-style-type: none"> • Internet Explorer 11 • Safari 11
Mobile Browser**	<ul style="list-style-type: none"> • Chrome (Latest) • Safari (Latest) 	
JDK***	<ul style="list-style-type: none"> • IBM J9 JDK 8 • Oracle JDK 8 • Oracle JDK 11 • All Java Technical Compatibility Kit (TCK) compliant builds of Java 11 and Java 8 	

Ilustración 63: Tabla de compatibilidades Liferay 7.1

Despliegue de los módulos Api y Service

Desde GoGo Shell, ejecutamos diag y vemos los siguientes errores

```

2237|Installed | 1|PrestecMaterialEsportiu-api (1.0.0)|1.0.0
2238|Installed | 1|PrestecMaterialEsportiu-service (1.0.0)|1.0.0
3! diag
:at.diba.materialesportiu.api [2237]
  Unresolved requirement: Import-Package: com.liferay.expando.kernel.model; version="[1.1.0,2.0.0)"

:at.diba.materialesportiu.service [2238]
  Unresolved requirement: Import-Package: cat.diba.materialesportiu.exception; version="[1.0.0,2.0.0)"
)
  -> Export-Package: cat.diba.materialesportiu.exception; bundle-symbolic-name="cat.diba.materiale
sportiu.api"; bundle-version="1.0.0"; version="1.0.0"; uses:="com.liferay.portal.kernel.exception"
  cat.diba.materialesportiu.api [2237]
    Unresolved requirement: Import-Package: com.liferay.expando.kernel.model; version="[1.1.0,2
.0.0)"
    
```

Ilustración 64: Consola GoGo shell

Solución

Desde el fichero `bnd.bnd` debemos añadir en `source` las siguientes líneas
`PrestecMaterialEsportiu-api/bnd.bnd`

```
Bundle-Name: PrestecMaterialEsportiu-api
Bundle-SymbolicName: cat.diba.materialesportiu.api
Bundle-Version: 1.0.0
Export-Package: \
    cat.diba.materialesportiu.exception,\
    cat.diba.materialesportiu.service,\
    cat.diba.materialesportiu.service.persistence,\
    cat.diba.materialesportiu.model
-includeresource: META-INF/service.xml=../PrestecMaterialEsportiu-service/service.xml
Import-Package:\
    com.liferay.expando.kernel.model;resolution:=optional
```

Ilustración 65: Fichero bundle de módulo api

`PrestecMaterialEsportiu-service/bnd.bnd`

```
Bundle-Name: PrestecMaterialEsportiu-service
Bundle-SymbolicName: cat.diba.materialesportiu.service
Bundle-Version: 1.0.0
Liferay-Require-SchemaVersion: 0.0.1
Liferay-Service: true
Import-Package:\
    cat.diba.materialesportiu.exception;resolution:=optional
Export-Package: \
    cat.diba.materialesportiu.exception
Require-Capability: liferay.extender; filter:="(liferay.extender=spring.extender)"
```

Ilustración 66: Fichero bundle del módulo-service

`PrestecMaterialEsportiu-admin/bnd.bnd`

```
Bundle-Name: PrestecMaterialEsportiu-admin
Bundle-SymbolicName: cat.diba.materialesportiu.admin
Bundle-Version: 1.0.0
Export-Package: cat.diba.materialesportiu.admin.constants
Import-Package:\
    com.liferay.application.list;resolution:=optional
```

Ilustración 67: Fichero bundle del módulo admin

Tamaño en algunos campos de la BBDD (para oracle, en mySql no es necesario) por tipo de dato localized

Error: value too large for column

Solución: Ampliar el tamaño de los campos para los localized

```
alter table pme_material  
modify name varchar(4000);
```

```
alter table pme_material  
modify usageRules varchar(4000);
```

```
alter table pme_material  
modify loanConditions varchar(4000);
```

```
alter table pme_activity  
modify name varchar(4000);
```

```
alter table pme_activity  
modify description varchar(4000);
```

```
alter table pme_activity  
modify usageRules varchar(4000);
```

```
alter table pme_activity  
modify loanConditions varchar(4000);
```

Tras instalar la BBDD realizando pruebas, no se puede acceder a view.jsp, mensaje de error:

```
2020-04-13 17:23:42.132 ERROR [http-nio-8080-exec-4][IncludeTag:128]  
Current URL  
/group/guest/~-/control_panel/manage?p_p_id=cat_diba_materialesportiu_admin_PrestecMaterialEsportiuAdminPortlet&p_p_lifecycle=0&p_p_state=maximized&p_p_mode=view&_cat_diba_materialesportiu_admin_PrestecMaterialEsportiuAdminPortlet_mvcPath=%2Factivity%2FviewActivity.jsp&p_p_auth=0w522uT  
m generates exception: java.lang.NullPointerException  
java.lang.NullPointerException
```


Solución:

El servicio “cat.diba.materialesportiu.service” debe aparecer en la tabla release_, además, el campo schemaVersion debe coincidir con el del bundle (bnd.bnd) del service Liferay-Require-SchemaVersion: 0.0.1

Para esto, hemos generado la clase Activator en nuestro service, además de borrar el contenido del fichero indexes.sql.

```
package cat.diba.materialesportiu.internal.activator;

import com.liferay.portal.kernel.module.framework.ModulesServiceLifecycle;

public class PrestecMaterialEsportiuServiceBundleActivator implements BundleActivator {

    private ServiceTracker<Object, Object> _serviceTracker;

    @Override
    public void start(BundleContext bundleContext) throws Exception {
        Filter filter = bundleContext.createFilter(StringBundler.concat("&(objectClass=",
            ModulesServiceLifecycle.class.getName(), ")", ModulesServiceLifecycle.DATABASE_INITIALIZED, "));

        _serviceTracker = new ServiceTracker<Object, Object>(bundleContext, filter, null) {

            @Override
            public Object addingService(ServiceReference<Object> serviceReference) {

                try {
                    BaseUpgradeServiceModuleRelease upgradeServiceModuleRelease = new BaseUpgradeServiceModuleRelease() {

                        @Override
                        protected String getNamespace() {
                            return "pem";
                        }

                        @Override
                        protected String getNewBundleSymbolicName() {
                            return "cat.diba.materialesportiu.service";
                        }

                        @Override
                        protected String getOldBundleSymbolicName() {
                            return "prestecmaterialesportiu-portlet";
                        }
                    };

                };

                upgradeServiceModuleRelease.upgrade();

                return null;
            } catch (UpgradeException ue) {
                throw new RuntimeException(ue);
            }
        };

        _serviceTracker.open();
    }

    @Override
    public void stop(BundleContext context) throws Exception {
        _serviceTracker.close();
    }
}
```

Ilustración 68: Clase BundleActivator

6. Bibliografía

- [1] Mark Otto and Jacob Thornton. 2011. Bootstrap. Retrieved from <https://getbootstrap.com/>
- [2] John Resig. 2006. jQuery. Retrieved from <https://jquery.com/>
- [3] 1996. Java. Retrieved from <https://www.java.com/es/>
- [4] 1999. Apache Tomcat 9. Retrieved from <http://tomcat.apache.org/>
- [5] 2001. Hibernate. Retrieved from <https://hibernate.org/>
- [6] 2004. Liferay. Retrieved from <https://www.liferay.com/es/>
- [7] 2008. MySQL. Retrieved from <https://dev.mysql.com/downloads/mysql/5.7.html>
- [8] Préstec de material esportiu. Retrieved from <https://www.diba.cat/es/web/esports/prestec-de-material>
- [9] Formulario de préstamo. Retrieved from <https://www.diba.cat/documents/205293/276469185/C1-119-20.pdf>
- [10] Solicitud de préstamo. Retrieved from <https://www.diba.cat/es/web/esports/esdev1linies2prest3solic>
- [11] Oracle Database. Retrieved from <https://www.oracle.com/es/index.html>
- [12] Bootstrap. Retrieved from <https://help.liferay.com/hc/en-us/articles/360017899492-What-is-Service-Builder->
- [13] JSP. Retrieved from <https://www.diba.cat/es/web/esports/esdev1linies2prest3solic>

7. Glosario

- ¹ Base de datos.
- ² Módulo de gestión para el departamento de préstamos.
- ³ Código para confirmar las reservas utilizado por el departamento de préstamos.
- ⁴ Integrated Development Environment (Entorno de desarrollo integrado).
- ⁵ Es la capa que interactúa directamente con base de datos, realiza operaciones CRUD.
- ⁶ Representación de la información, capa donde se definen nuestras entidades.
- ⁷ Donde crearemos nuestra lógica y métodos para que el aplicativo realice sus funciones.
- ⁸ Métodos propios de Liferay que actúan como métodos get y devuelven un objeto o colección de objetos con los atributos solicitados.
- ⁹ Consulta definida por el usuario mediante lenguaje SQL y que normalmente se utiliza cuando queremos realizar consultas que los métodos finder no permiten.
- ¹⁰ Clase que utiliza un timer para ejecutar tareas de manera periódica.
- ¹¹ Tipo de plantilla Liferay para la creación del módulo web.
- ¹² Interfaz de programación de aplicaciones, donde tendremos un conjunto de funciones y procedimientos generados por Service-Builder.
- ¹³ Procedimiento por el cual se enlaza cada una de las entidades y atributos generados con las tablas y atributos creados en base de datos.
- ¹⁴ Fichero en el que definimos nuestras entidades y que Service-Builder utiliza para generar nuestras tablas, además de los módulos service y api.
- ¹⁵ Propiedad que define el prefijo de nuestras tablas en BBDD
- ¹⁶ Tipo de plantilla Liferay para la creación del módulo de administración.
- ¹⁷ Ficheros Java Server Pages donde se define el código de nuestras páginas web de servidor.
- ¹⁸ Hojas de estilo en cascada, donde definimos el formato y diseño de nuestras páginas web.
- ¹⁹ Métodos básicos para manipular nuestros objetos (Create, Read, Update, Delete).