



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
[INGENIERÍA DE SOFTWARE]

Desarrollo de la aplicación web - Museo E.T.S.I. Informática

Parte III

**Web Application Development - E.T.S.I. Informática Museum
Part III**

Realizado por
Jorge Luis Palomares Ruiz

Tutorizado por
Eduardo Guzmán De los Riscos

Departamento
Lenguaje y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2019

Fecha defensa:

Fdo. El/la Secretario/a del Tribunal



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

Resumen

Este proyecto se basa en la utilización de las tecnologías más recientes para desarrollar una aplicación novedosa para el museo de la E.T.S.I Informática, ante la ausencia de una actual que cumpla con las necesidades actuales. Se recogerán las características necesarias para cumplir este objetivo en un Documento General de Requisitos. Este trabajo se realizara en grupo, mediante la metodología ágil de *Scrum*, dividiendo la carga de trabajo en *sprints*, donde se irán abordando los distintos requisitos. Todos los integrantes participaran en todas las etapas de desarrollo de manera transversal. Se desarrollaran la base de datos, el servidor y el cliente web. El resultado final será una aplicación web que permita visualizar todas las obras de museo de manera interactiva, como también la administración de todas ellas, por otro lado se adquirirán los conocimientos necesarios de un desarrollo *full stack*, en tecnologías punteras por parte del alumnado.

Palabras clave:

Aplicación web para Museo de Informática, Angular, TypeScript, SpringBoot, MongoDB.

Abstract

This Project is based on the use of the most recent technologies to develop a modern web application for the E.T.S.I Informática museum, because of the absence of a current one that meets the current needs. The necessary characteristics will be described to fulfill this objective in a Product Requirements Document. This work will be done in a group, using the agile Scrum methodology, dividing the workload into sprints, where the different requirements will be addressed. All the members will participate in all stages of development in a transversal manner. The database, the server and the web client will be developed. The final result will be a web application that allows users to visualize all the museum works interactively, as well as the administration of all of them, on the other hand the students will acquire the necessary knowledge of a *full stack* development, in cutting edge technologies.

Keywords:

Web Application for Computer Science Museum, Angular, TypeScript, SpringBoot, MongoDB.

Índice

Resumen	7
Abstract	9
Índice	11
Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Metodología	3
1.4 Tecnologías y herramientas utilizadas	5
1.4.1 Aplicación cliente.....	5
1.4.2 Servidor	6
1.4.3 Base de datos.....	7
1.4.4 Herramienta para pruebas de servicios.....	7
1.4.5 Herramienta de gestión de tareas.....	8
1.4.6 Control de versiones.....	9
1.4.7 Herramienta de planificación.....	9
1.4.8 Herramientas de comunicación.....	10
1.4.9 Herramienta de almacenamiento	10
1.4.10 Herramienta de diseño.....	11
1.5 Estudio de otros trabajos similares	11
1.6 Estructura de la memoria	15
Fases del desarrollo del proyecto	17
2.1 Iteración 0	17
2.2 Iteración 1	23
2.3 Iteración 2	30
2.4 Iteración 3	36
Conclusiones y líneas futuras	1
3.1 Conclusiones	1
3.2 Líneas futuras	2
Referencias	5
Manual de Instalación	9
Requerimientos:	9
Manual de Usuario	11
1. Navegación por la web	11

1.1. Inicio.....	11
1.2. Documentación.....	11
1.3. Colección.....	11
1.4 Donaciones.....	12
1.5 Mapa del Museo.....	12
1.6 Acerca de.....	12
1.7 Accesos Directos.....	12
2. Zona Administración.....	12
2.1 Documentación.....	13
2.2 Colección.....	14
2.3 Marca.....	16
2.4 Tipo.....	17
2.5 Sala.....	18
3.Manual de usuario.....	19
Documento General de Requisitos.....	21
Diseños Técnicos - Iteración 1.....	31
Vista Colección.....	31
Vista Piezas.....	33
Pruebas de sistema - Iteración 1.....	37
Contenidos.....	37
Diseños Técnicos - Iteración 2.....	47
Vista CRUD Pieza.....	48
Vista Crear Pieza.....	49
Pruebas de sistema - Iteración 2.....	53
Contenidos.....	53
Diseños Técnicos - Iteración 3.....	73
Vista Acerca De.....	73
Vista Mapa Del Museo.....	75

1

Introducción

1.1 Motivación

La motivación para la realización de este Trabajo de fin de grado surge de la idea de nuestro tutor de modernizar y ampliar un trabajo anterior de desarrollar una página web para el museo de la Escuela Técnica Superior de Ingeniería Informática, con el uso de los códigos QR. Actualmente el museo de la escuela no cuenta con una aplicación web donde los visitantes puedan disfrutar de la amplia gama de obras con la que cuenta el museo.

Teniendo en cuenta nuestro futuro profesional y el uso por parte de las empresas de las nuevas tecnologías, vimos la oportunidad de ampliar nuestros conocimientos al usar estas nuevas tecnologías, a la vez que profundizar en los conocimientos generales aplicables a todo tipo de proyectos que se han aprendido en la carrera. Además se dio la oportunidad de realizar este proyecto en grupo, con lo que se incrementaría la aptitud de trabajo en grupo. Con este fin se decidió crear una web moderna y completa para el museo de nuestra escuela con las últimas tecnologías del mercado.

1.2 Objetivos

Mencionado anteriormente, el objetivo del proyecto es el desarrollo completo de una aplicación web, desde la base de datos, hasta aplicación cliente, pasando por el servidor. Todo esto para cumplir con el comportamiento acorde a la especificación de requisitos planteadas en el documento general de requisitos acordados inicialmente y las variaciones de estos requisitos con el avance del proyecto. Los puntos acordados en dicho documento son:

- Creación, modificación, eliminación y visualización de las piezas, documentos, artículos, marcas, tipos y salas de las que consta el museo.
- Una vista usuario y otra vista administración que requiera un inicio de sesión para acceder a la zona restringida de la web.
- Elaboración de un cronograma o línea temporal donde se visualiza la evolución cronológica de los elementos del museo.
- Implementación de la seguridad en los servicios del servidor.
- Alojamiento de la aplicación servidor en remoto, al que se accede mediante peticiones REST.
- Tener un dominio remoto accesible a cualquier usuario.
- Construcción de un manual de usuario.

Todo este desarrollo orquestado por un desarrollo ágil y partiendo desde el desconocimiento de la mayoría de tecnologías usadas, pero con las técnicas aprendidas como ingenieros de software que permiten desarrollar un producto de calidad mediante pruebas.

Con este enfoque se planea aprender y experimentar la elaboración de una página web real, en un equipo de desarrollo desde los cimientos y lo que involucra, como inversión para el futuro. Es muy común que los profesionales se especialicen en un determinado área dentro del desarrollo web, sin embargo muchos en el comienzo no queremos decantarnos por una opción sin conocer el resto de las áreas. Al crear la página en grupo, pero siendo participes en todas las etapas del desarrollo se formará una visión que ayude a cada uno a elegir su camino personal y elaborar una perspectiva crítica respecto de su trabajo con el de los compañeros.

1.3 Metodología

Después de algunas dudas, para conseguir que el proyecto fuera un éxito se optó por utilizar una metodología ágil, en lugar de una metodología basada en cascada. El motivo fue que era la primera vez que realizábamos un proyecto de estas características y no se sabían cuales iban a ser los tiempos necesarios para realizar cada una de las tareas. Esto suponía un problema porque no sabíamos que carga de trabajo íbamos a ser capaces de realizar antes de que se acabara el tiempo de desarrollo. En las metodologías ágiles se proporciona una gran flexibilidad al contrario que la metodología en cascada en la que se necesitaría ordenar las etapas del proceso rigurosamente. Por este motivo se eligieron las metodologías ágiles y dentro de estas la de *Scrum*.

Siguiendo esta metodología se eligió a un *Scrum Master*, y se redactaron las tareas necesarias para el inicio del proyecto. Todas estas tareas y las siguiente correspondientes a las iteraciones se colgaron como notas en un tablón usando la aplicación de *Trello*, donde estas tareas tendrían que estar en una lista que indicara el estado en el que se encontraba y la parte del proyecto a la que pertenecían.

Los estados posibles de las tareas son *To Do* (tareas que hay que hacer, pero en las que todavía no se está trabajando), *In Progress* (tareas que se están realizando) y *Done* (tareas terminadas). Dependiendo de en qué iteración nos encontráramos las notas estarían en listas distintas, formándose el nombre de la lista a partir de la iteración donde se realizara y el estado de las tarjetas que contuviese, por ejemplo "To Do Iteración 1". A cada nota , también llamada tarjeta en la aplicación , se le puede incluir uno o varios miembros, que son las personas que se encargaran de llevarla a cabo. Al haber sido un trabajo en grupo algunas tareas que se han realizado en común, tienen varios miembros mientras que otras las ha realizado un solo miembro. Siempre que algún integrante del grupo ha tenido que ayudar a otro se ha intentado reflejar este hecho añadiéndolo a la tarjeta en la que hubiese participado.

En las iteraciones se ha utilizado el siguiente procedimiento para realizar las pestañas de la web. Primero se realizaba una reunión entre los desarrolladores y se establecían las pestañas que se realizarían durante la iteración o *sprint* (en *Scrum* a las iteraciones se les llama *sprint*), el tiempo de duración del *sprint* y los responsables de cada tarea. Por cada pestaña, se establecían las siguientes tareas para asegurar que se cumplían los requisitos establecidos y las medidas de calidad: un diseño técnico de la página que incluye una maqueta/*mockup*; implementación ; y pruebas. En caso de que no pasara las pruebas alguna característica o que se

encontraran desperfectos, se crearía una nueva tarea indicando el error, esta tarea empezaría en el *To Do* de iteración en la que fue encontrada, en caso de no poderse llevar a cabo en esa iteración pasaría a la lista de *To Do* de la siguiente iteración para que se empezara a trabajar en ella y posteriormente finalizada. Cabe destacar, que algunas tareas de carácter especial no cumplen este procedimiento.

Como es un trabajo en grupo, y necesitábamos una base sólida a partir de la cual poder trabajar, se decidió crear una *Iteración 0*, donde se trabajó estrechamente para conseguir un entorno de trabajo lo mejor preparado para el trabajo que se iba a desarrollar y los requisitos que nuestra aplicación tuviese que cumplir, además de preparar la base de datos junto con algunos de los servicios básicos y el portal principal. Este trabajo en grupo queda constatado principalmente en el Documento General de Requisitos (DGR) documento al que hemos acudido siempre que hemos tenido dudas sobre el aspectos del proyecto.

En este documento se recogen las necesidades del sistema. Es el medio mediante el cual podemos transmitir los requisitos que tenía cumplir el sistema a los interesados en el proyecto. Mediante este documento también se puede comprobar, cuando el desarrollo ha acabado si el sistema cumple con las necesidades que se acordaron. Además dentro del documento se encuentran el modelo de la base de datos, el modelo del dominio, las estimaciones de tiempos y los modelos de navegación (se hará hincapié en ellos más adelante).

Después de esta primera *Iteración 0*, se comenzaron las iteraciones en las que se desarrollan pestañas para la ampliación . Su funcionamiento era dividir las pestañas que se creían convenientes para cada iteración y realizar el procedimiento de creación de pestañas explicado anteriormente. Cada pestaña era desarrollada completamente por el estudiante, desde la base de datos hasta el código del cliente, es decir un desarrollo transversal por funcionalidades. Aun así y aun participando todos en todas las capas, cada estudiante ha dedicado más parte de esfuerzo a una de las capas (en este trabajo se realizara más hincapié a la capa *back-end*) y se intentara profundizar más en el trabajo personal desarrollado. En la descripción de cada iteración, en el capítulo de las **2.fases de desarrollo del proyecto**, se explicará detalladamente cuál ha sido la contribución de este TFG a las diferentes partes de la aplicación.

En la iteración 1 se priorizo diseñar las pestañas que mostraran el museo al usuario sobre las piezas y los documentos, y la página de recepción. La asignación de estas tareas fue al azar y las tareas que evolucionaron a partir de estas fueron encomendadas a la misma persona.

En la segunda iteración el objetivo era permitir a la persona encargada del mantenimiento de la web introducir nuevas piezas, editarlas y borrarlas. Con esto en mente se creó un rol de administrador el cual tendría acceso a una nueva pestaña desde la cual podría realizar todas estas acciones (a parte del rol de administrador se crearon todas las pestañas correspondientes a la creación, edición y borrado de los objetos de la base de datos).

En la última iteración (*Iteración 3*), se añadieron pestañas que ofrecían funcionalidades, que mejoraran la experiencia del visitante de la web y que modernizaran la imagen de la página usando elementos de *APIs* externas, por otro lado se añadió la seguridad a los servicios. Además se realizó un cambio en todo el estilo de la aplicación para que todas las pestañas siguieran un mismo estilo y la estética de la web estuviera más acorde con la web de la universidad.

1.4 Tecnologías y herramientas utilizadas

Fueron muchas las tecnologías y herramientas que se barajaron para desarrollar el proyecto y otras necesidades aparte del desarrollo como la planificación, diseño etc. Las tecnologías de desarrollo se defienden en tres capas: las dos primeras correspondientes a la base de datos y al servidor forman parte del *back-end* y la tercera el cliente forma parte del *front-end*. A continuación se explicarán las tecnologías usadas para cada capa y las aplicaciones que han ayudado en otros aspectos al desarrollo de la aplicación.

1.4.1 Aplicación cliente

Para el cliente se propusieron 2 *frameworks*, *React* y *Ionic*. Se evaluaron los pros y las contras de cada uno:

- *React*: Mantenido por Facebook, es una biblioteca de *JavaScript* de código abierto que facilita el desarrollo de aplicaciones de una sola página, además está muy extendida y la comunidad de usuarios es muy grande.
- *Ionic*: Es un estándar para el desarrollo de aplicaciones híbridas y permite con una sola base de código que la aplicación funcione en la mayoría de plataformas. *Ionic* funciona con angular, tanto con *JavaScript* como con *TypeScript*.

A la hora de elegir se optó por *Ionic*, el motivo fue que queríamos utilizar *JavaScript* o *TypeScript*, esto lo hubiéramos conseguido con cualquiera de las dos, pero nos parecía muy interesante trabajar con Angular y también pensamos que

era interesante la facilidad con la que podríamos hacer nuestra página accesible desde todos los dispositivos electrónicos (tables, móviles u ordenadores).

Comenzando el desarrollo se encontró que los módulos que ofrecía *Ionic* estaban tan enfocados a aplicaciones móviles, que no encajaban con lo que pretendíamos conseguir. Por este motivo se decidió usar solamente *Angular*, con *TypeScript*, al ser un lenguaje tipado facilitaba la gestión de errores.

Angular es un *framework* de código abierto, creado por *Google* y utiliza el Modelo Vista Controlador(MVC) al que ya estamos acostumbrados. Tiene una gran variedad de componentes los cuales nos permiten personalizar el diseño de nuestra página a los requisitos iniciales, además las propiedades de estos componentes se usan para el *binding* de los datos, lo que facilita el trabajo del desarrollador.

Para realizar el cliente en angular, se decidió usar todos el mismo editor (en todas las capas se ha intentado que utilicemos las mismas herramientas) para facilitar el intercambio de información. Se eligió *Visual Studio Code*, por tres factores principales: primero ofrecía muchas extensiones para desarrollar en Angular; segundo, es un editor ligero; y finalmente ofrece una consola interna la cual es muy útil al trabajar con proyectos de angular.

1.4.2 Servidor

En la parte del servidor se ha usado *SpringBoot*. El manejo de las dependencias en los proyectos es una labor muy tediosa a la par que importante, cuando hay muchas dependencias, manejarlas a mano es complicado. *SpringBoot* está diseñado para solucionar este problema, creando un *pom* (o *gradle*, soporta tanto *maven* como *gradle*) con todas las dependencias que tenga un proyecto. De esta forma se reduce el tiempo que le dedicamos al mantenimiento de las dependencias para dedicárselo al desarrollo. Además con Spring Initializr se puede crear ya directamente el proyecto base con las dependencias que se le indiquen. Para el servidor se ha usado el entorno de desarrollo IntelliJ IDEA por la buena integración que tiene de proyectos de SpringBoot y la cantidad de documentación que hay sobre proyectos de este tipo con este entorno.

Por otro lado se lanza el servidor en una maquina remota, la cual mantiene el servidor estable para recibir y enviar información de manera constante, se profundizará sobre este aspecto en la sección **Iteración 0** del capítulo 2.

1.4.3 Base de datos

Siempre habíamos utilizado base de datos relacionales a lo largo de la carrera, pero en los últimos años debido al gran volumen de datos con los que trabajan las empresas se están usando cada vez más las bases de datos no relaciones. En este proyecto no era una condición necesaria la utilización de una base no relacional, pero como se mencionó al principio nuestro interés era el de aprender nuevas formas de hacer las cosas. Así se decidió usar *MongoDB* que es uno de los sistemas de base de datos NoSQL orientado a documentos con más implantación.

En todo los procedimientos hemos intentado externalizar al máximo los procesos locales y realizarlos en la nube. *MongoDB Atlas* es una servicio perfecto para esta función, es un servicio automatizado en la nube de la plataforma MongoDB, la base de datos en remoto permite a todos los miembros del grupo tener esta actualizada constantemente y trabajar simultáneamente, al igual que *Spring Boot*, nos permite reducir el tiempo que dedicamos a la administración de la base de datos para otras labores.

1.4.4 Herramienta para pruebas de servicios

Para probar las respuestas de los servicios del servidor se utilizó *ARC* una aplicación que permite realizar peticiones a un *endpoint* y ver los resultados. Con esta aplicación se puede realizar distintos tipos de peticiones, como la *get*, *post* o *delete*.

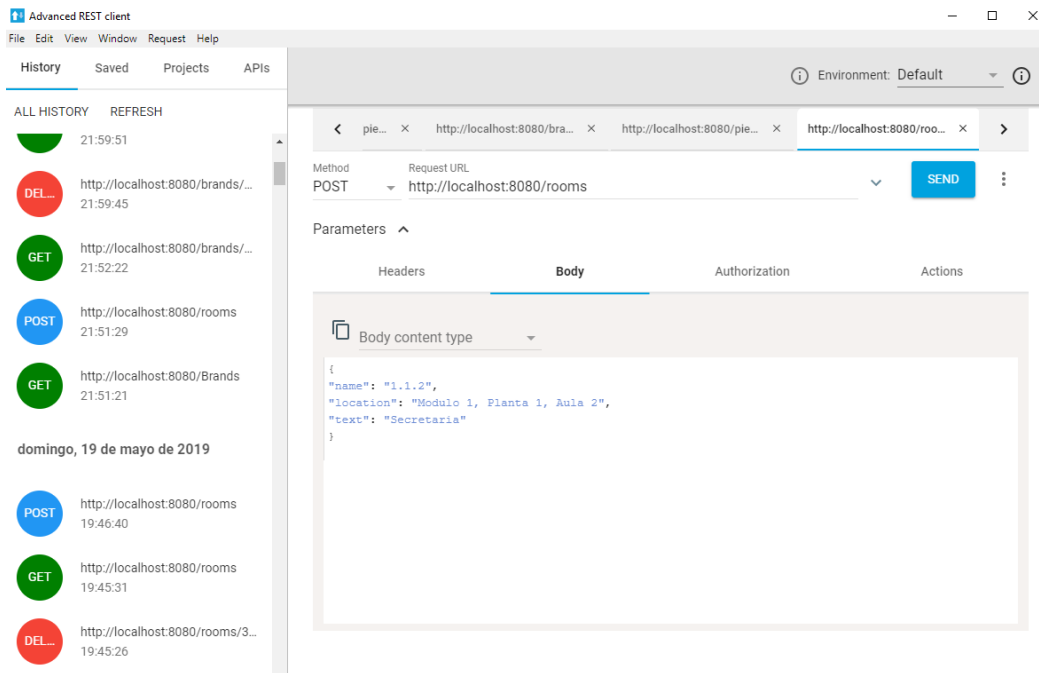


Figura 1.1 Petición *POST* para crear una sala mediante ARC.

1.4.5 Herramienta de gestión de tareas

En este apartado, intentamos usar *Jira* o *Redmine* que son herramientas profesionales de gestión de tareas, lamentablemente no estaban a nuestro alcance y se decidió usar Trello que nos daba las funcionalidades que necesitábamos de forma gratuita. Como ya se explicado anteriormente el tablero lo dividimos en listas que indicaban la iteración y el estado de la tarea. Las tareas son notas que se van moviendo entre las listas dependiendo de su estado y estas tarjetas tendrán asignado un o unos responsables de ejecutarla. Este tablero es una forma visual de saber en qué están trabajando tus compañeros a la vez que de saber que tareas te quedan por hacer y planificar las siguientes. Además se crearon listas especiales para los errores de diseño llamadas *To Do Fixes CSS*.

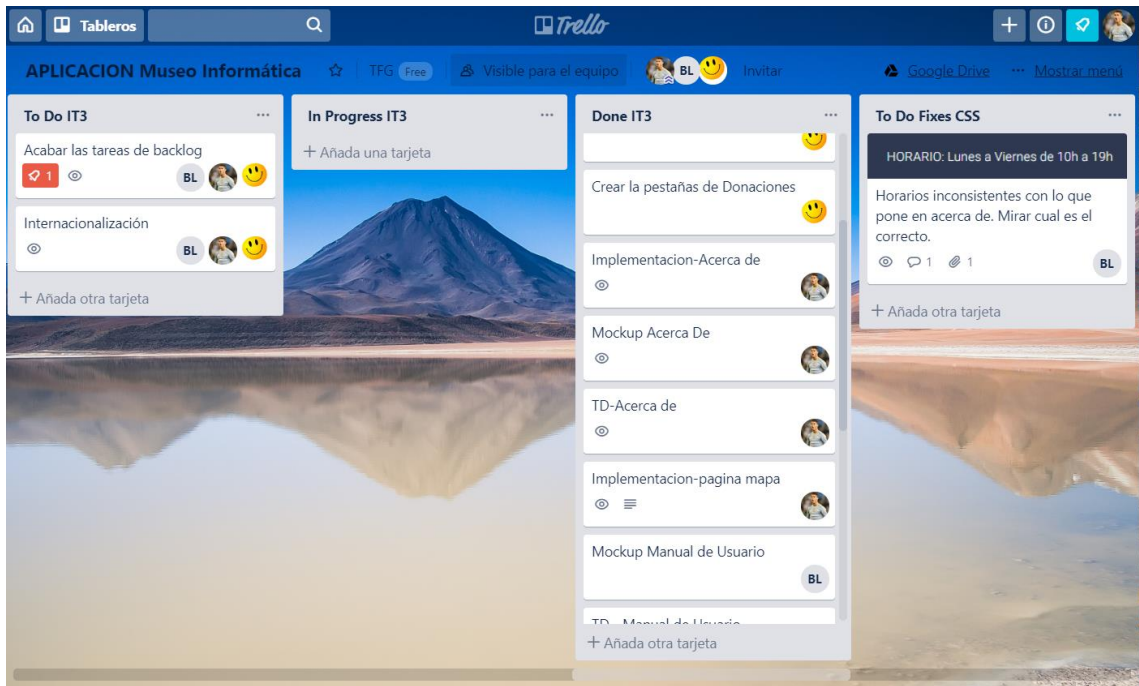


Figura 2.1 Tablero de Trello Iteración 3

1.4.6 Control de versiones

En este apartado no tuvimos duda la herramienta predilecta para el mantenimiento y control de versiones es *GitHub*. Al trabajar en un proyecto que involucra a varias personas y de un tamaño considerable, es necesario crear distintas ramas donde cada desarrollador pueda trabajar su funcionalidad sin tener que estar preocupándose de lo que los demás estén haciendo y cuando esté finalizada fusionarla con el código de los demás compañeros, pudiendo retroceder a una versión estable deshaciendo los cambios, si fuera necesario. En nuestro caso somos 3 desarrolladores y tenemos una rama para cada uno y una rama principal *master* donde se unen. Es gratificante ver todo el movimiento que ha habido en los repositorios en estos meses.



Figura 3.1 Gráfico 1 de las ramas de Github

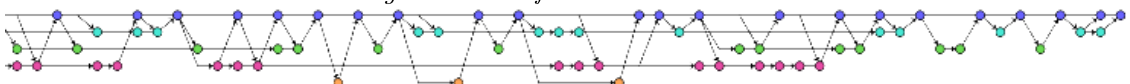


Figura 3.2 Continuación del gráfico

1.4.7 Herramienta de planificación

Para la planificación se utilizó *Google Calendar*, que nos permitía en un calendario marcar franjas de tiempo de un determinado color de manera visual, además por cada franja se permite hacer introducir una descripción. De esta forma basándose en otros proyectos software y unas pruebas iniciales que hicimos decidimos una planificación acorde al proyecto. En la planificación la fase de implementación es única y hubiese sido interesante haberla dividido por iteraciones.

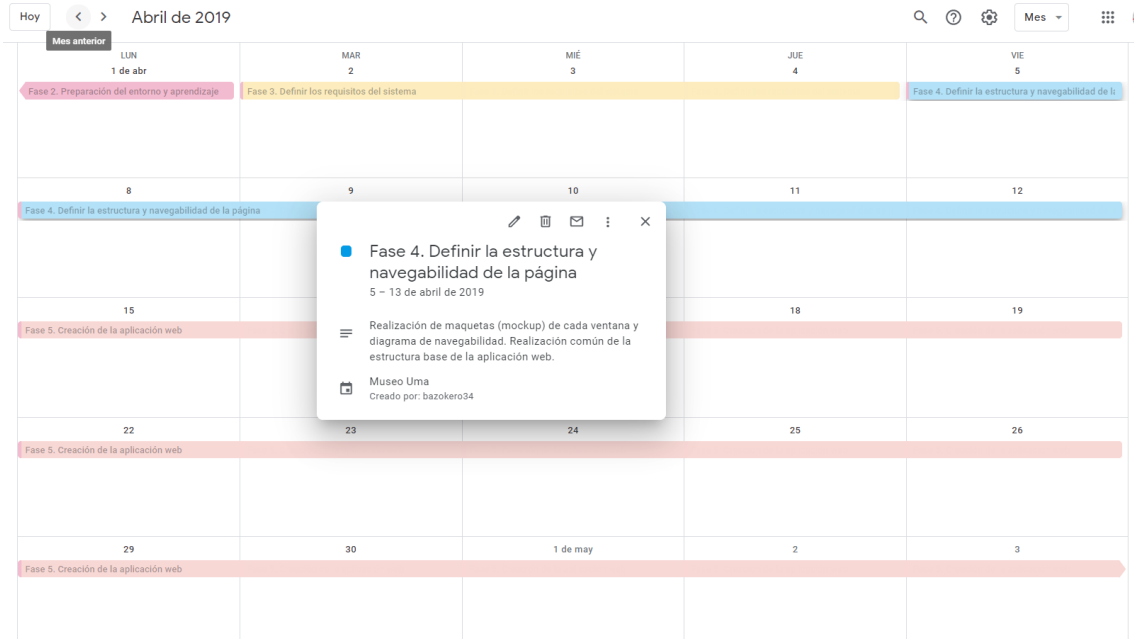


Figura 4.1 Planificación del mes de Abril

1.4.8 Herramientas de comunicación

Para comunicarnos con nuestro tutor hemos utilizado la herramienta *Slack*, tanto para preguntar dudas, intercambio de información o concertar citas.

A nivel interno de los desarrolladores hemos utilizado *Discord* para realizar llamadas grupales o compartir pantalla cuando ha sido necesario y para temas o cuestiones rápidas se creó un grupo de *Whatsapp*.

1.4.9 Herramienta de almacenamiento

Para almacenar toda la documentación relacionada con el proyecto de forma que fuera de fácil acceso y común, se creó una carpeta compartida en Google Drive que es un servicio de alojamiento en la nube.

1.4.10 Herramienta de diseño

Para diseñar las maquetas de la aplicación web se usó la página web de *Cacoo* que permite diseñar una página de la web con los componentes que ofrece, o añadir personalizados para ajustar la maqueta al diseño buscado.

1.5 Estudio de otros trabajos similares

Siempre al comenzar un proyecto es importante informarse sobre otros anteriores y más todavía si no se es experto en la materia. Para ello se investigaron varios grandes museos de varios países intentando encontrar el estándar de página que siguen los museos punteros. Esto es de gran ayuda a la hora de diseñar nuestra propia web del museo y gran parte de la página está basada en partes de otras.

La página en la que más nos hemos basado ha sido del museo PICASSO (enlace en las referencias) la cual hemos imitado en nuestra página principal.

Figura 5.1 Cabecera de inicio del Museo Picasso



La cabecera de la web que hemos realizado se basa en esta, disponiendo de un título o logo en la izquierda y mostrando información en la parte derecha junto con el conjunto de las redes sociales de la web.

En la parte inferior se encuentra un menú que permite navegar entre las distintas partes de la página, una manera simple y visual.



Figura 5.2 Contenido de inicio del Museo Picasso

De la parte principal del cuerpo cabe destacar en la parte central un carrusel que va cambiando entre distintas exposiciones, lo que aporta variedad y movimiento. Al intentar modernizar no queríamos tener solo elementos estáticos. También cuenta con 3 tarjetas inferiores las cuales muestran imágenes llamativas las cuales sirven como enlaces a distintas pestañas de la página.

HORARIO Todos los días de 10.00h a 19.00h Más información	VENTA ANTICIPADA Acceda sin esperas con sus entradas impresas Más información	CONECTA CON MPM Facebook Twitter Youtube Pinterest Instagram Spotify	MUSEO PICASSO MÁLAGA Palacio de Buenavista C/ San Agustín, 8 29015 Málaga, España
Junta de Andalucía FABA		Información legal Contacto Política de privacidad	

Figura 5.3. Pie de página del Museo Picasso

Del pie de página además hemos tomado algunas referencias, el museo PICASSO de Málaga muestra su horario e información adicional como su localización o los enlaces de sus redes sociales.

También en la aplicación se toman referencias de otros museos para crear ideas nuevas para nuestra página como el mapa del Museo Metropolitano del Arte en Estados Unidos.

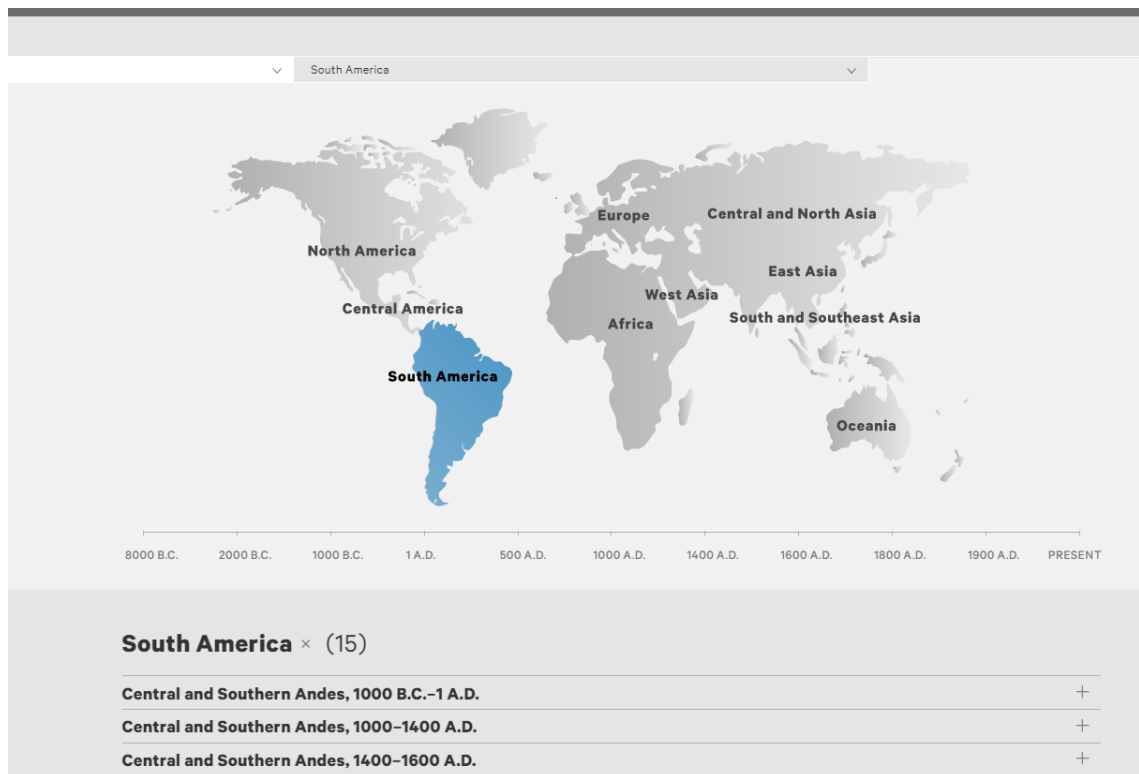


Figura 5.4 *Metropolitan Museum of Art*. Línea Temporal de la historia del arte

En la figura 5.4, se puede apreciar un mapa del mundo donde pulsando en una región, se pueden visualizar las obras que hay disponibles de esta región y la línea temporal. Esto nos parecía curioso y que podría aportar una mejora considerable a la búsqueda de piezas del museo de cara a los visitantes.

Para el buscador principal de la página, se buscaron simples y visuales, sin tener que dar mucho trabajo al usuario. Por este motivo se priorizaron los menús y casillas de selección.

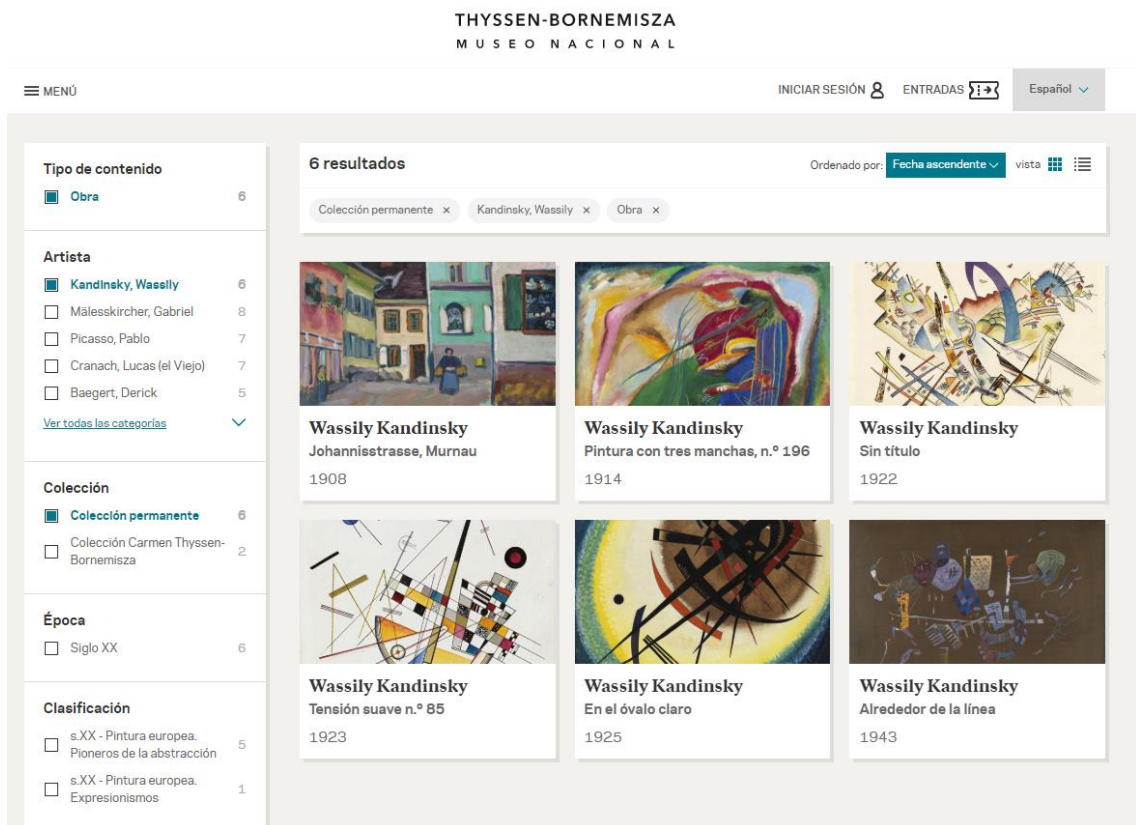


Figura 6.1 Buscador de contenido del Museo THYSSEN-BORNEMISZA

A parte se buscaba nuevamente que la información buscada se mostrara de manera clara y estética con esto en mente.

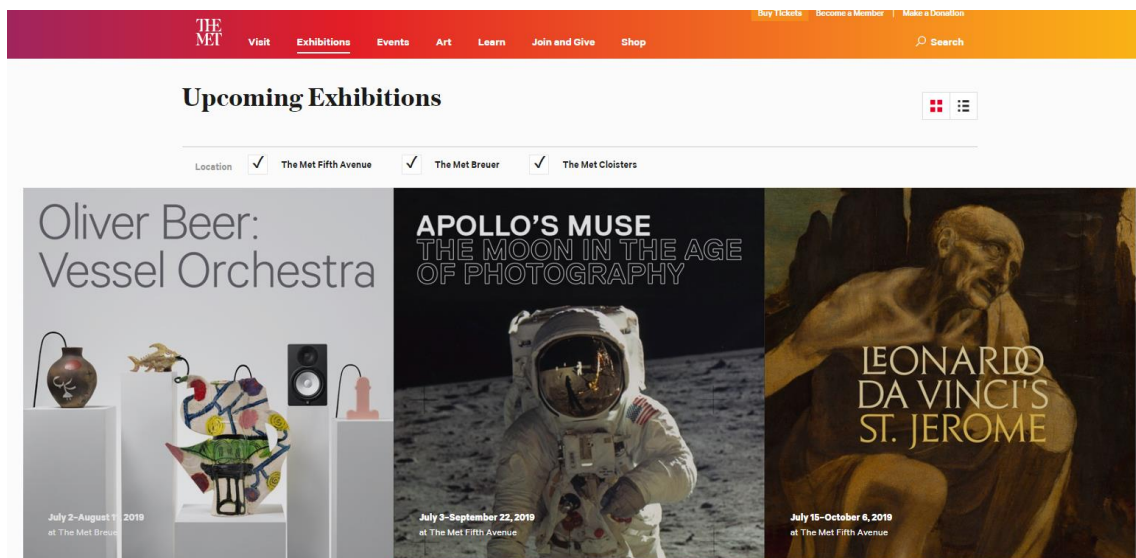


Figura 7.1 Buscador de exhibiciones del Metropolitan Museum of Art

En todas estas se muestra la información como tarjetas, cuyo fondo es una imagen perteneciente a la obra, y la mínima información para no sobrecargar. Además en los buscadores no es necesario pararse a recordar y escribir los títulos.

1.6 Estructura de la memoria

La memoria está estructurada en tres capítulos con sus respectivos subcapítulos. Cada capítulo intenta relatar un aspecto troncal de este proyecto desde un determinado punto de vista distintito de cara al desarrollo.

En primer lugar, tenemos la introducción donde en sus subcapítulos exponemos la razón del propósito y todo lo relacionado con este. Con este objetivo se describirán todos los procesos realizados como preparativos para el inicio de la implementación de la web. En estos apartados explicamos nuestros argumentos, para decantarnos al elegir la metodología de desarrollo seguida para el proyecto y las distintas tecnologías usadas. Finalmente el estado del arte en el que nos encontramos.

La segunda parte son las fases del desarrollo del proyecto. Este proyecto utiliza una metodología ágil como se explicara próximamente, por este motivo el trabajo se divide en iteraciones. En este apartado explicamos el trabajo realizado en cada iteración, los problemas encontrados junto con las medidas tomadas y como se ha llevado a cabo toda la implementación.

En tercer lugar, llegamos a la conclusión y a las líneas futuras, donde se hace una crítica constructiva al proyecto, concluyendo que partes de podrían haber cambiado o hecho mejor, a su vez las partes que se consideren un éxito. Por otro lado en las líneas futuras se darán nociones de posibles cambios que se consideren beneficiosos para la página y características nuevas que pudieran suponer una mejora para la web.

Después de estos capítulos, se mostrara las referencias en las que nos hemos basado para realizar los encontrar todas las referencias disfrutadas y consultadas para lograr las características del proyecto. Posteriores a las referencias se encontraran la guía de instalación junto con el manual se usuario y los anexos. Los anexos contienen toda la documentación que el proyecto ha generado y que perfectamente podrían estar incluidas dentro del segundo capítulo, pero que por extensión se ha decidido dejar fuera por si el lector quiere profundizar.

2

Fases del desarrollo del proyecto

2.1 Iteración 0

La Iteración 0 nace de la necesidad de establecer una base para el desarrollo igual para todos los participantes con el fin de desarrollar de la misma forma y poder poner en común más fácilmente los conocimientos y ayudar eficientemente a resolver problemas que surjan. Así lo primero en mente fue, ¿Qué vamos a hacer? y posteriormente ¿Cómo lo vamos a llevar a cabo?. Para responder a la primera pregunta se realizó un Documento General de Requisitos(DGR). Conociendo los objetivos que teníamos que cumplir, se estableció la base de datos y el servidor básico, realizando así, la base necesaria para responder a la segunda pregunta.

El DGR completo se puede encontrar en los apéndices(Apéndice C) donde se puede encontrar toda la información recogida, analizada que define las características y las necesidades de alto nivel del sistema. A continuación se comentarán pinceladas de este documento, que no hayan sido explicadas anteriormente.

Nuestra aplicación está enfocada para los usuarios visitantes de la web, pero también se definió un rol distinto preparado para la administración del lugar. Cada Participante posee unos permisos distintos que les permiten realizar distintas funcionales dentro de la página para cumplir con sus necesidades:

- **Usuario:** Un usuario podrá visualizar los contenidos de la aplicación web.
- **Administrador:** Un administrador tiene los mismos permisos que un usuario y además, puede acceder a una zona restringida de administración, proporcionando los credenciales necesarios. En esta zona podrá crear, actualizar o borrar información de la web que se almacena en una base de datos.

Como se mencionó en la sección **1.2 Objetivos** este proyecto se basó en unos requisitos que se definieron inicialmente con posibilidad de variaciones. Estas características quedan formalmente recogidas en el apartado Requisitos Funcionales del Documento General de Requisitos. A continuación, el elemento crucial en la fase de análisis de requisitos, los propios requisitos.

Requisito Funcional	Prioridad
RF 1 - Crear pieza: La pieza debe tener los atributos nombre, texto, año, fecha de creación, lista de imágenes, código QR, nombre de la sala, lista de tipos y nombre de la marca a la que pertenece.	ALTA (IT2)
RF 1.1- Editar pieza: El sistema permite al administrador la edición de piezas que ya habían sido creadas anteriormente.	ALTA (IT2)
RF 1.2 - Eliminar pieza: El sistema debe permitir al administrado eliminar una pieza que ya había sido creada anteriormente.	ALTA (IT2)
RF 1.3- Visualizar pieza: El sistema debe permitir a un usuario visualizar el contenido visible de una pieza .	ALTA (IT1)

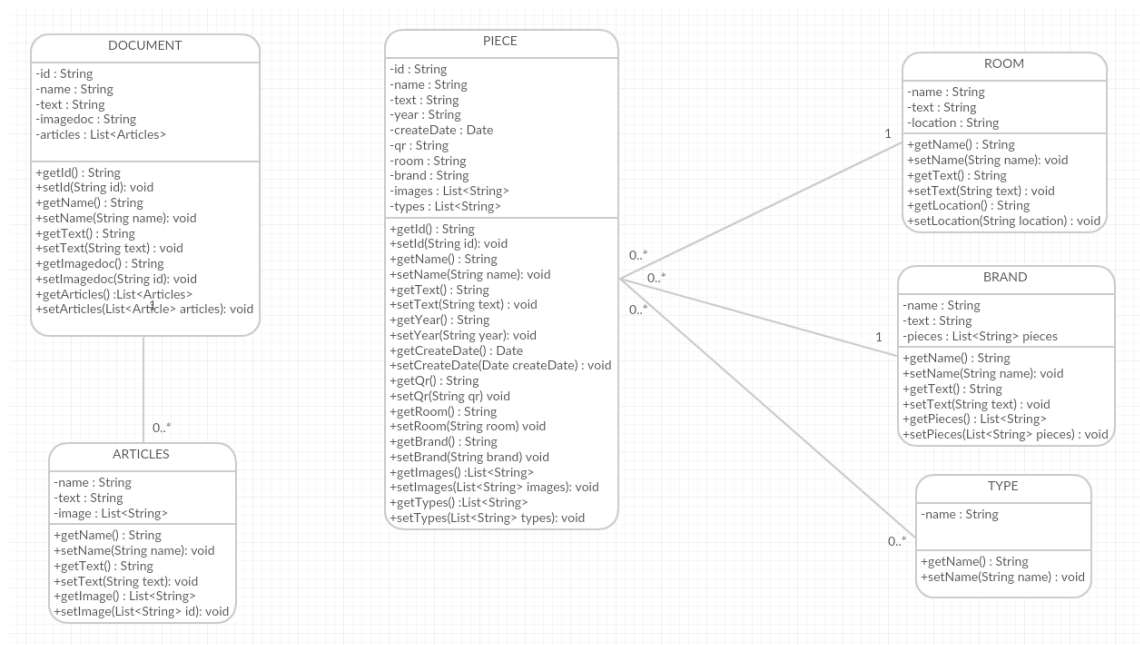
Figura 8.1 Requisitos funcionales sobre las piezas

En la figura 8 se pueden visualizar los requisitos enfocados a la creación, visualización, edición y eliminación de las piezas. Igual que este se pueden encontrar los requisitos derivados para los documentos salas etc. A cada requisito

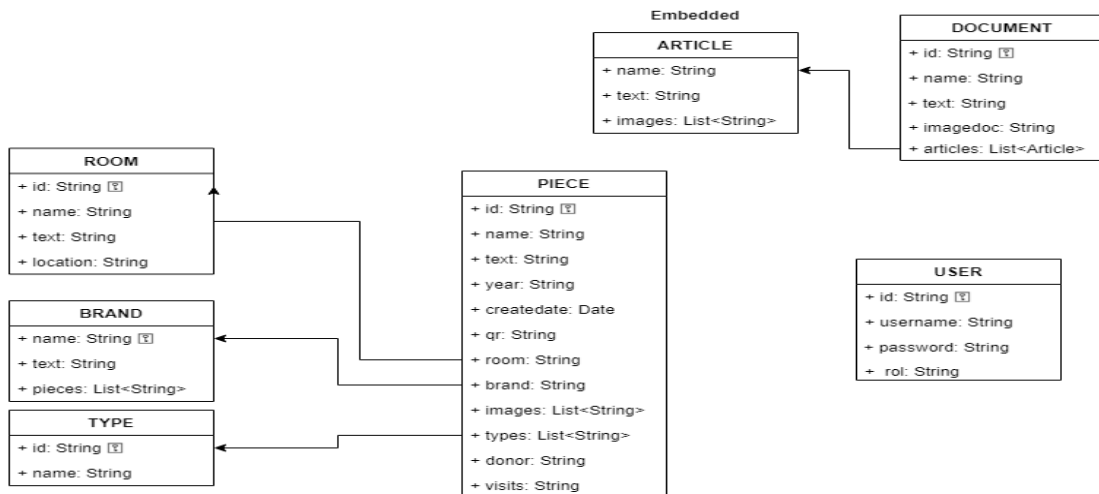
se le asigna una prioridad que indica la importancia de este para el sistema, acompañada de la iteración en la que se provee su desarrollo.

También se establecieron ciertos requisitos sobre para la generación de documentación, para asegurar la calidad del el correcto uso de la aplicación. A esta parte se le llamo Requisitos de Documentación. Este requisito género en la necesidad de una guía de usuario la cual explicara cada una de las funcionalidades y procesos de la página.

En este documento(DGR) también se establecen los objetos a crear, sus atributos, y las relaciones entre ellos. Se realizaron diagramas del modelo del dominio y conceptuales. También se crearon modelos de navegación, para definir el flujo entre pantallas.



9.1 Modelo del dominio



10.1 Modelo conceptual de la base de datos

Con la aceptación del DGR por todas las partes involucradas en el proyecto de comenzo ha desarrollar la base para el desarrollo.

Se creó la base de datos remota en un *cluster* de la aplicación *mongoDBAtlas*, donde se crean las colecciones, que son las clases en *mongoDB*, desde la misma aplicación se pueden hacer consultas a la base de *datos*.

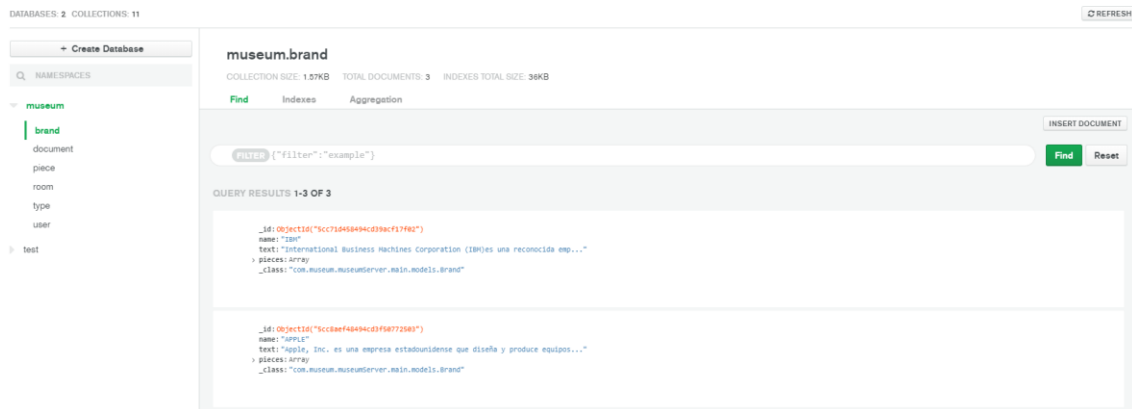


Figura 11.1 Instancias de "Marca" en la base de datos *museum* en *MongoDB Atlas*

Diseñar una base de datos NoSQL no es tarea fácil, primero se realizo la base de datos relacional con todos los problemas y dudas que surgen a la hora de diseñar un sistema. Esta se transformó en una no relacional manualmente mediante recomendaciones que seguimos de distintas documentaciones. Una de las decisiones más complicadas era la eleccion de crear “documentos” con informacion embebida, un documento embebido es el caso de los artículos, que

están dentro de los documentos. Esto supone que teniendo el documento, obtenemos todos los artículos que contiene. Esto es útil, pero en el caso de que solo quisieramos ver un artículo determinado, estaríamos trayendo más información de la necesaria. Por ese motivo solo se puede usar cuando el subdocumento embebido no se va a llamar por separado.



11.2 Documento con información embebida, imagen de la documentación de mongo.

Una vez se tenía la base de datos se comenzó con el desarrollo del servidor, mencionado anteriormente SpringBoot, nos permitía desarrollar el servidor de manera rápida y con las dependencias necesarias con su aplicación *Spring Initializr*. Con *Spring Initializr* creamos un proyecto servidor *gradle* con las dependencias para *MongoDB* y sus repositorios.

Para poder crear un servicio que pueda tratar los objetos de nuestra base de datos en Spring se necesitan crear dos elementos (se usará como ejemplo servicios de la clase pieza).

- Primero se crea su modelo, donde se crean variables para cada uno de sus atributos y su identificador. Para cada uno de estos atributos es necesario crear además su *get* y su *set*, para que puedan ser modificados.


```

public class Piece implements Serializable {

    @Id
    @NotNull
    private String id;

    private String name;
    private String text;
    private String year;
    private Date createdate;
    private String qr;
    private String room;
    private String brand;
    private String visits;
    private String donor;
    private List<String> images;
    private List<String> types;

    public String getId() { return id; }

    public void setId(String id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }
}

```

Figura 12.1 Parte del código de Piece.java

- En segundo lugar, necesitamos crear una interfaz para su repositorio donde estarán todos los servicios.

```

@Repository
public interface PieceRepository extends MongoRepository<Piece, Serializable> {

    List<Piece> findByName(@Param("name")String name, Pageable pageable);

    List<Piece> findByTypesContains(@Param("types")String[] types);

    List<Piece> findByRoom(@Param("room")String room);

    List<Piece> findByNameLikeIgnoreCaseOrTextLikeIgnoreCase(@Param("word")String word, @Param("word1")String word1);

    List<Piece> findByYearBetween(@Param("minyear")String minyear, @Param("maxyear")String maxyear);
}

```

Figura 13.1 Parte del código de PieceRepository.java

Como se puede apreciar, crear consultas de esta forma es muy sencillo. Si se quiere obtener una lista de piezas por el nombre de la sala, solo hay que poner lo que se espera que devuelva la consulta (una lista de Piezas) y escribir el nombre de la función siguiendo las referencias de la documentación de *Spring* para nombres de métodos, finalmente poner los parámetros que sean necesarios. Spring

agilizo mucho la creación de los servicios al crear código limpio y fácil de entender. También al extender los repositorios de *MongoRepository*, ya incluyen el CRUD básico.

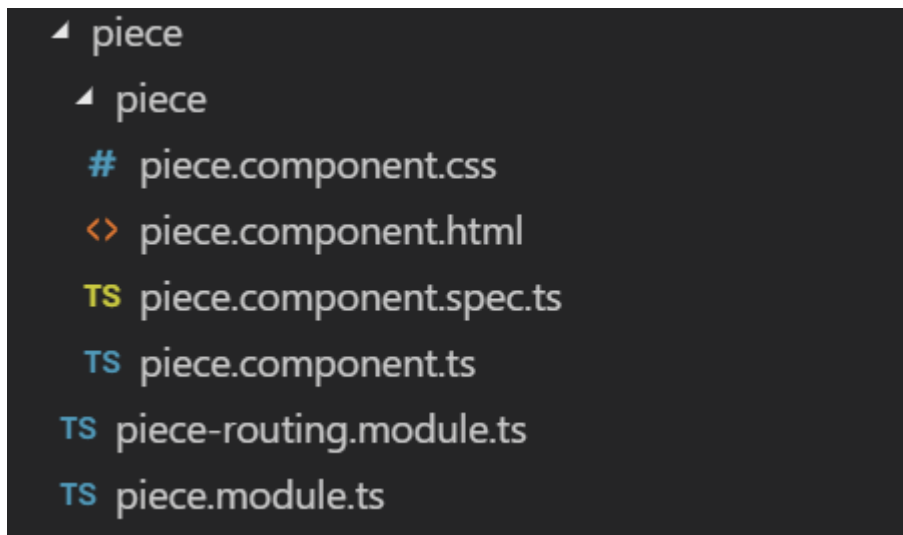
Posteriormente y en una tercera iteración se implementaría la seguridad en los servicios, explicada en profundidad en la **Parte I** del Desarrollo de la aplicación web - Museo E.T.S.I. Informática

En esta iteración dio lugar la creación de la estructura base del servidor. Así pues, como ya se ha comentado anteriormente, el proyecto inicial fue creado con Spring Initializr incluyendo dependencias a MongoDB y a SpringDataRest en un fichero *gradle*.

Para finalizar esta iteración, se alquiló una máquina virtual de *Microsoft Azure*. Las máquinas de *Azure* funcionan muy bien para desplegar el servidor remotamente y son mediante el modelo de pago por uso. Nuestra máquina fue una "*Tomcat on CentOS 7.3*". EL procedimiento una vez tenemos la maquina preparada es pasar el instalador del servidor, transferirlo a la máquina virtual, se usó *winscp*. A continuación, se usó *kitty* como cliente *SSH/telnet* para conectarse a la máquina de *Azure* y desplegar el servidor. La máquina viene solamente con las librerías básicas de Linux, por lo que fue necesario instalar algunas otras para poder desplegar el servidor. Una vez se demostró su correcto funcionamiento y utilidad se suspendió el servicio, para un futuro.

2.2 Iteración 1

A partir del comienzo de la Iteración 1 se inició el proceso de desarrollo de las páginas de la aplicación y sus servicios. Las vistas se desarrollaron con **Angular.ts**, la estructura de estos proyectos produce para cada página una serie de ficheros.



14.1 Estructura de la página piece en Angular

Los ficheros que caben destacar son:

- `piece.component.css`: donde está la hoja de estilos que usa la página.
- `piece.component.html`: en este fichero estará el código html que usa la página.
- `piece.component.ts`: es la parte lógica de la parte, en esta parte se llaman a los servicios y se añade parte del comportamiento de la página.
- en los demás ficheros se establece el *routing*, las importaciones y los *providers* (son las clases creadas para tratar los métodos del servidor).

Para ayudar con el desarrollo de esta sección y las siguientes a esta, se usará la documentación generada por en esta iteración para explicar exhaustivamente los progresos hechos en la aplicación, así como de las vistas generadas, entre otros los diseños técnicos y los casos de prueba. Finalmente se comprarán los diseños realizados como maqueta con el diseño final.

Vista Colección

Primero se hablará de la **Vista Colección**. En esta ventana se muestran las piezas que hay en la base de datos que coinciden con los criterios de búsqueda seleccionados por el usuario. La ventana consta de un buscador en el que el usuario puede filtrar las piezas por la marca, por un rango de antigüedad y finalmente

seleccionar los tipos de piezas que desea buscar. En la parte derecha de la ventana el usuario observará las piezas que cumplan con esos criterios, mostrando el número total de resultados, los resultados se cargaran en lotes. Pulsando en cualquiera de las piezas, el usuario accede a la pestaña de la pieza en cuestión.

Requisitos alcanzados:

Esta ventana alcanzaba el requisito:

- **RF 1.3- Visualizar pieza:** El sistema debe permitir a un usuario visualizar el contenido visible de una pieza.

Modelo y repositorio:

- Piece.java – PieceRepository.java
- Brand.java– BrandRepository.java
- Type.java– TypeRepository.java

Servicios:

- Al cargar la página se llamarán a las operaciones:
 1. getNPieces(int n) – Obtiene las primeras n piezas.
 2. getBrands() - Obtiene las marcas de piezas disponibles.
 3. getTypes() - Obtiene los tipos de piezas disponibles.
- Al buscar las piezas:
findByYearBetweenAndTypesContainsAndBrandLike(String minyear,String maxyear, String[] types, String brandname)Busca coincidencias dentro de la base de datos, respecto a los criterios de entrada.
- Otros servicios:
 1. findByTypesContains(String[] types) - Obtiene las piezas que contienen los tipos pasados por parámetro.
 2. FindByYearBetween(String minyear, String maxyear) - Obtiene las piezas que están entre el rango de años pasados por parámetros.
 3. FindByBrand(String brandname)- Obtiene las piezas cuya marca sea la que se pasa por parámetro.

Boceto

El boceto se usaba como plantilla para elaborar la página y poner en consenso con los otros desarrolladores los contenidos que inicialmente tendría la vista. En esta vista se buscó que en buscador y los resultados estuvieran en una misma vista y que no se tuviera que escribir por teclado. Como también se quiso hacer una barra deslizable para hacer más visual y dinámica el formulario.

Además, se tuvo en cuenta al usuario dejando un texto al final del formulario recalcando los datos introducidos.



15.1 Boceto de la vista Colección

Esta vista en su aspecto final no contiene muchas diferencias respecto a la versión del boceto. En su desarrollo se han usado tarjetas de *Bootstrap* que ofrecen una visión en un tamaño reducido de la información correspondiente a la pieza y la cual se puede pulsar como enlace. El problema de la página es el tamaño variable de las tarjetas las cuales, al tener distintas dimensiones sus imágenes, descripciones o tipos, hace variable el tamaño de las tarjetas. Se podrían haber diseñado de otra forma manteniendo en un tamaño fijo los parámetros antes mencionados, pero las tarjetas no aprovecharían tan bien el espacio y el resultado actual con tarjetas de tamaño variable no nos disgustó.

E.T.S. INGENIERÍA INFORMÁTICA UNIVERSIDAD DE MÁLAGA

Visita Teclas.org

f y t i

Administración

Inicio Documentación **Colección** Donaciones Mapa del Museo Acerca de

Colección de Piezas

¿Buscas una pieza? Coincidencias: 5

Tipos de Piezas

- Bolsillo
- Consola de Juegos
- Eléctrico
- Ordenador
- Programable
- Calculadora

Año de la pieza

Año mínimo:

Año máximo:

1900 2020

Marca de la pieza


Valores Seleccionados

Tipos:
No ha elegido ningún tipo

Rango de Tiempo:
[1900,2019]

Marca:
Ninguna


BUSCAR



Portable PC (5155)

Primitivo ordenador portátil, el primero de IBM (si no contamos el 5100, de 1975 y que pesaba 25kg.)...


Eléctrico,Ordenador



II Europlus

Con unidad de disco, dos manuales, tres tarjetas de expansión y dos joysticks.


Ordenador



Macintosh Classic

Modelo M1420. El mítico Apple Macintosh se resiste a morir. Este modelo sigue con la CPU Motorola 6...

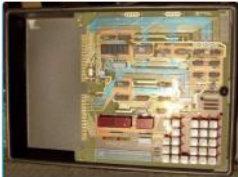
Eléctrico,Programable,Ordenador



HP-12C

Calculadora de bolsillo con funciones financieras y programable. Probablemente una de las máquinas má...

Calculadora,Bolsillo,Programable,Eléctrico



HP 5036A Microprocessor LAB

Aquí tenemos un entrenador lógico basado en el procesador Intel 8085. Se vendía dentro de un maletí...

Eléctrico

15.2 Aspecto final de la vista Colección

Tabla de componentes:

Esta tabla detalla todos los elementos que aparecen en el boceto. En la Figura 15.1 se muestra una pequeña parte de esta tabla. En las siguientes vistas de las que se hable en las iteraciones se omitirá la tabla de componentes. Para ver la información detallada de cada vista se pueden consultar los Apéndices correspondientes al diseño técnico de la vista en concreto.

Nombre (id)	Componente	Obligatori o	Validació n	Valor por defect o	Notas
brandsFiel d	selectOneListb ox	No	No		Servicio: getBrands()
yearsField	slider-container	No	No	1970- 2020	

15.2 Parte de la tabla de componentes de la vista Colección

Vista Piezas

En esta ventana se muestra la pieza que el usuario eligió en alguna de las ventanas de la web. El id de la pieza se pasará como parámetro en la *URL* de la página. Se visualizará el título o nombre, la descripción, el año, la fecha de adquisición, la marca, el nombre de la sala donde está expuesta, las visitas, los tipos en los que está incluida y las imágenes de esta. En la parte izquierda, se sitúa una galería de imágenes donde se podrá seleccionar cualquiera de ellas para verlas a mayor tamaño. A la derecha disponemos de toda la información referente mencionada anteriormente, menos la descripción que se encuentra en la parte inferior separada.

Requisitos alcanzados:

- RF 1.3- Visualizar pieza: El sistema debe permitir a un usuario visualizar el contenido visible de una pieza.

Modelo y repositorio:

- Piece.java– PieceRepository.java
- Brand.java– BrandRepository.java
- Type.java– TypeRepository.java

Servicios:

Al cargar la vista:

- getPieceById(String id) - Busca la pieza por su id (id pasado por parámetro en la url).

- `increaseVisits()`- Incrementa el número de visitas en la unidad

Boceto:

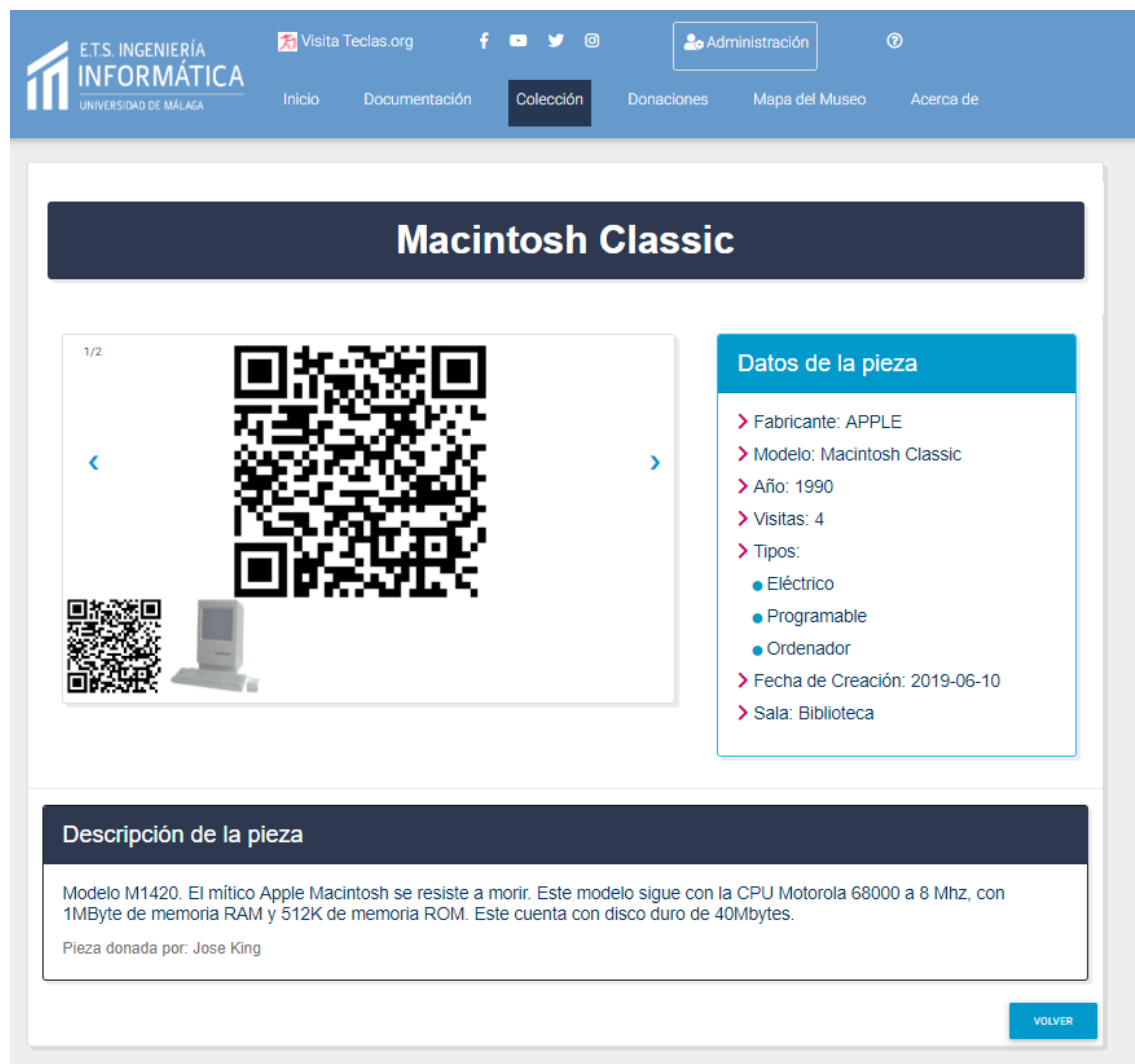
En la vista *Piezas* se quería mostrar toda la información almacenada de la pieza en la base de datos, de una forma ordenada y sencilla, por eso se resaltan los datos importantes en mayor tamaño y con negrita. Por otro lado, cada pieza tenía un numero variable de imágenes , por ese motivo se creó una galería que permitiese ir cambiando entre las distintas imágenes de la pieza.



16.1 Boceto de la vista Piezas

En esta vista respecto al boceto, se añadió más información adicional sobre la pieza como la fecha de creación, además se introdujo un botón en la esquina inferior derecha que permite volver a la vista Colección en cualquier momento. Por otro lado para solucionar el problema que surgía al mostrar la página cuando una pieza no tenía ninguna imagen y además mostrar el QR, se añadió la imagen

del QR a la galería como la primera imagen disponible, este código QR se genera a través de la API *QR Code Generator* (<http://goqr.me/api/>) en la vista *Crear Pieza*.



16.2 Aspecto final de la vista Piezas

2.3 Iteración 2

En la segunda iteración se optó por completar el *CRUD* de los objetos de la base de datos, creando las vistas necesarias para crear, editar o eliminar. Para ello se crearon las vistas de *CRUD Pieza* y la vista *Crear Pieza*. Así como los servicios necesarios para cumplir con las funcionalidades y se corrigieron defectos de la Iteración 1. Por otro lado, se añadieron otras funcionalidades como la generación a través de api externa de códigos QR para las piezas.

Los **requisitos alcanzados** con estas pantallas fueron:

- **RF 1-Crear pieza:** La pieza debe tener los atributos nombre, texto, año, fecha de creación, lista de imágenes, código QR, nombre de la sala, lista de tipos y nombre de la marca a la que pertenece.
- **RF 1.1- Editar pieza:** El sistema permite al administrador la edición de piezas que ya habían sido creadas anteriormente.
- **RF 1.2 - Eliminar pieza:** El sistema debe permitir al administrado eliminar una pieza que ya había sido creada anteriormente.
- **RF 4 - Crear marca:** Si se intenta añadir una pieza con una marca que no está creada, el sistema deberá permitir crear una nueva.
- **RF 4.1 - Eliminar marca:** El sistema debe permitir al administrador eliminar una marca que ya había sido creada anteriormente.
- **RF 5 - Crear tipo:** Si se intenta añadir una pieza con un tipo que no está creado, el sistema deberá permitir crear un nuevo tipo aportando su nombre.
- **RF 5.1 - Eliminar tipo:** El sistema debe permitir al administrador eliminar un tipo que ya había sido creado anteriormente.
- **RF 6 - Crear sala:** La sala debe tener los atributos nombre, localización y texto.
- **RF 6.1- Editar sala:** El sistema permite al administrador la edición de salas que ya habían sido creadas anteriormente.

Vista CRUD Pieza

En esta ventana se muestra una lista con todas las piezas registradas en la base de datos, mostrando su nombre. Desde la lista podemos borrar o editar las piezas pulsando en los botones dispuestos, la ventana también presenta un botón en la esquina superior izquierda para poder crear una nueva pieza(Vista Crear Pieza). Esta vista está dentro de un “panel tab” donde se encuentras los demás vistas CRUD de Documentos, Tipos, Marcas y Salas.

Modelo y repositorio:

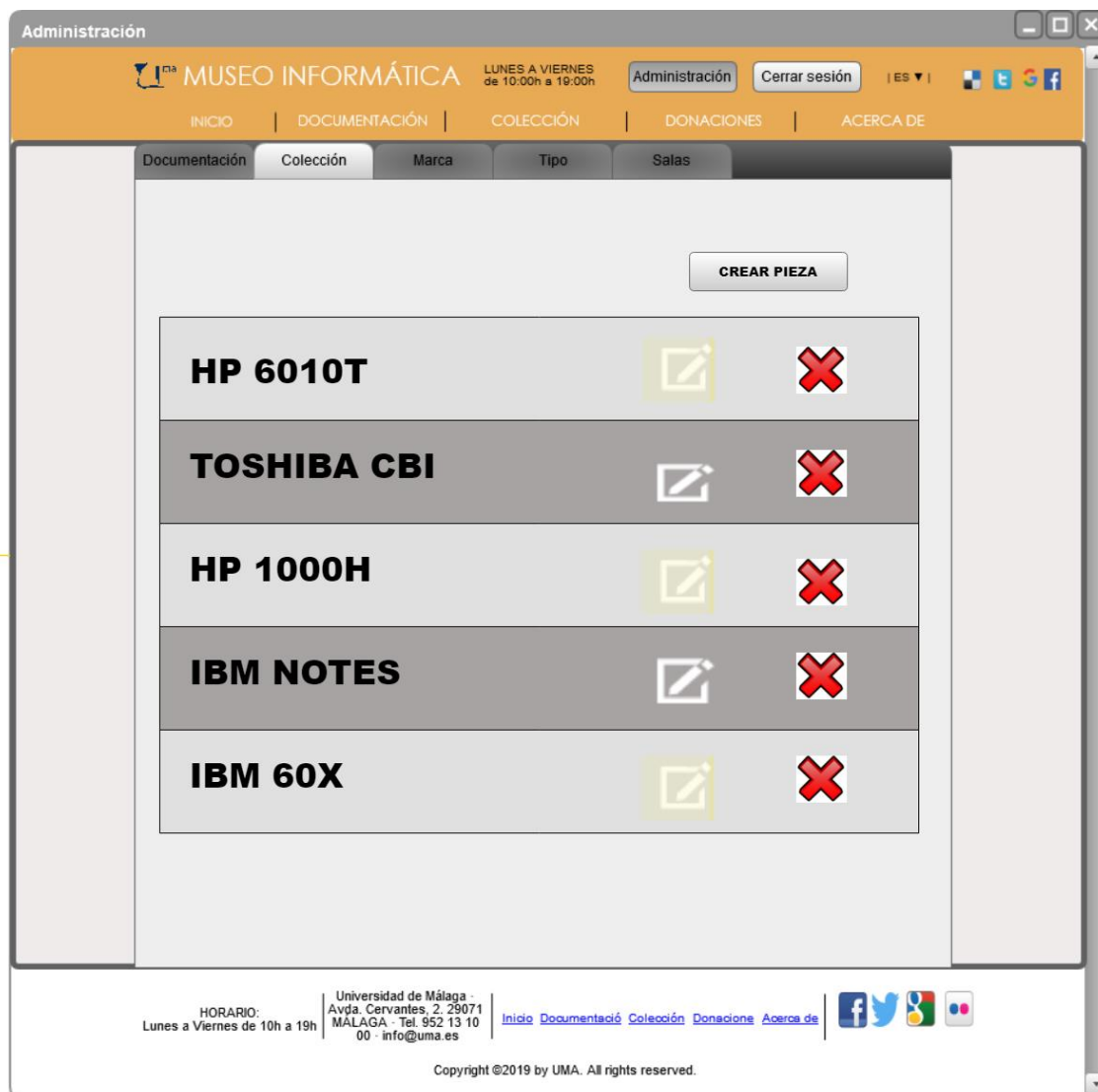
Piece.java– PieceRepository.java

Servicios:

findAll() - Devuelve una lista con todas las piezas de la base de datos.

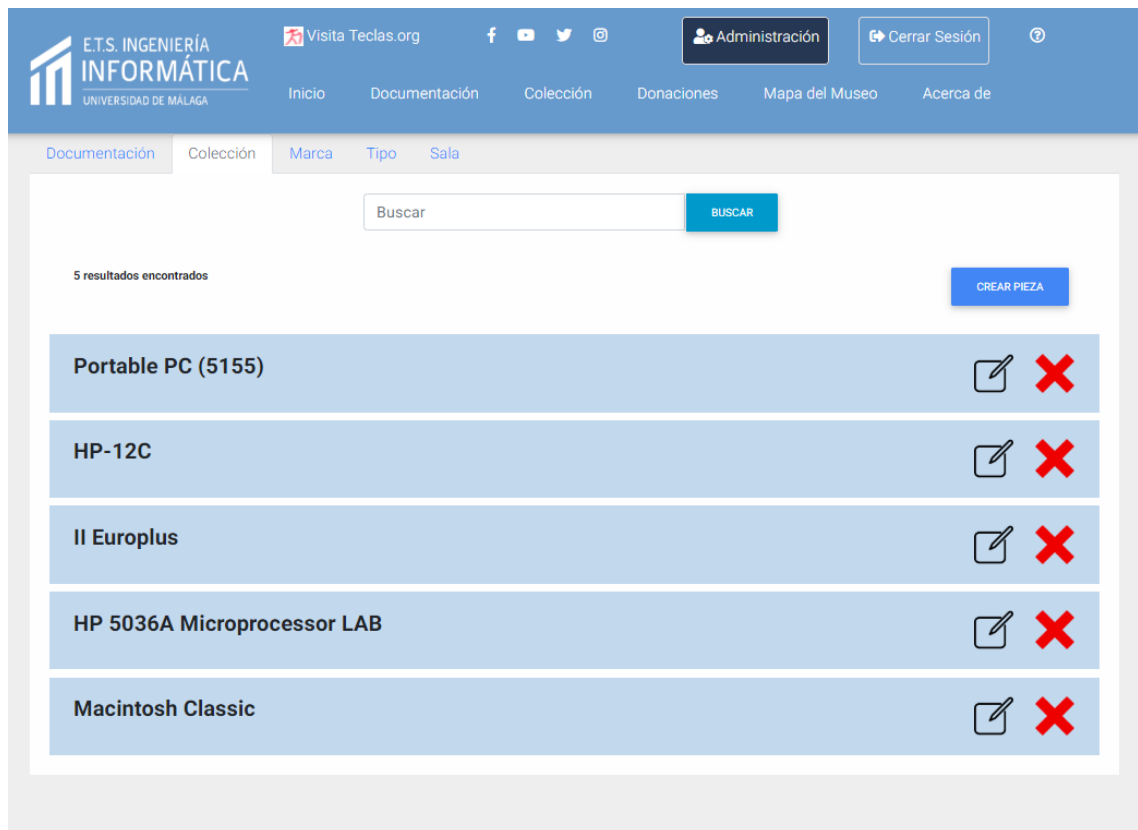
Boceto

Se creo un estilo común para las vistas del CRUD para agilizar el trabajo de desarrollo y darle cohesión a la aplicación. Se dispone de unos botones los cuales permiten cumplir los requisitos mencionados anteriormente de crear editar o eliminar una pieza.



17.1 Vista CRUD Pieza

Siguiendo con el estilo común creado, se pasó esta vista a código, donde en su aspecto final se añadió el número de resultados encontrados y un buscador.



17.2 Aspecto final de la vista CRUD Pieza

Vista Crear Pieza

En esta ventana se muestra un formulario con los atributos necesarios para añadir una nueva pieza en nuestra base de datos. La ventana consta de cuadros de texto y seleccionables para introducir los datos pertinentes en una columna izquierda. En la parte derecha de la ventana el usuario observará los tipos de piezas y las imágenes introducidas, pudiendo eliminarlas de la pieza a crear de ser necesario. Además, se cuenta con unas pestañas para crear nuevos fabricantes, tipos y salas por si el campo que quiere introducir no existía anteriormente . Pulsando en cualquiera de los botones situados en las pestañas de "Crear Marca", "Crear Sala" o "Crear Tipo" se creará una nueva instancia de esa clase, finalmente pulsando en "Crear Pieza" se creará una nueva pieza .

Modelo y repositorio:

- Piece.java– PieceRepository.java
- Brand.java– BrandRepository.java
- Type.java– TypeRepository.java

- Room.java-RoomRepository.java

Servicios:

- Al cargar la página se llamarán a las operaciones:
 - getRooms() – Obtiene las salas disponibles.
 - getBrands() - Obtiene las marcas de piezas disponibles.
 - getTypes() - Obtiene los tipos de piezas disponibles.
- Al crear alguna marca, sala o tipo:
 - createBrand(Brand brand) - Crea la marca con los parámetros marcados.
 - createType(Type type) - Crea el tipo con el parámetro nombre.
 - createRoom(Room room) - Crea la sala con los parámetros marcados.
- Al crear la pieza :
 - createPiece(String name, String text, String year, Date createdate, String qr, String room, String brand, String visits, String donor, String[] images, String[] types) Crea la pieza con los parámetros marcados en el formulario.

Boceto

El objetivo en esta ventana fue poder crear la pieza con todos sus datos y en caso de querer introducir una marca, tipo o sala que no estuviera ya almacenada en la base de datos poder crear ese elemento directamente desde esta página sin tener que pasar a otra. Por ese motivo se diseñó un menú de navegación a la derecha que permitía crear cualquiera de los mencionados anteriormente. Además, el caso de introducir cualquier tipo o imagen y querer corregir un error se contempla y se soluciona con unos menús donde se muestran estos tipos e imágenes y se permiten eliminarlos.

MUSEO INFORMÁTICA LUNES A VIERNES de 10:00h a 19:00h Administración | ES |

INICIO | DOCUMENTACIÓN | **COLECCIÓN** | DONACIONES | ACERCA DE

Crear Pieza

Inserte el nombre de la pieza
Nombre

Inserte el descripción de la pieza
Descripción

Inserte el año de creación de la pieza
Año de creación

Inserte la sale de la pieza
 Sala 3.1.1

Seleccione la marca de la pieza
 HP

Inserte el donador de la pieza
Donador

Inserte las imagen/es de la pieza
Imágenes

Inserte los tipos de la pieza
Tipos

Crear Tipo

Crear Tipo

Tipos introducidas

Tipo 1	✖
Tipo 2	✖
Tipo 3	✖

Imágenes introducidas

	✖		✖		✖
--	---	--	---	--	---

Crear Pieza

HORARIO: Lunes a Viernes de 10h a 19h | Universidad de Málaga - Avda. Cervantes, 2. 29071 MÁLAGA - Tel. 952 13 10 00 - info@uma.es | [Inicio](#) [Documentación](#) [Colección](#) [Donaciones](#) [Acercas de](#)

Copyright ©2019 by UMA. All rights reserved.

18.1 Vista Crear Pieza

Posiblemente una de las paginas a la que más tiempo se le ha dedicado junto con la de Colección, debido a la gran cantidad de servicios requeridos, y a la complicación de combinarlos todos en una sola pantalla. Por un lado, se decidió dejar de lado la idea del panel de navegación dentro de la página que permitiera la creación de marcas, tipos y salas dentro de esta. El motivo fue la complejidad de la implementación, y se optó por hacer paginas distintas para su implementación. Por otro lado, se creó un espacio vertical que permite al usuario introducir la *URL* de una imagen y añadirla a la pieza; muy parecido es también el contenedor donde se almacenan los tipos seleccionados, cualquiera de estos se puede eliminar en caso de error. Finalmente hay que explicar que esta página se desarrolló como un dos por uno, la misma página sirve tanto para crear una pieza desde cero, o cargar los datos de la pieza creada anteriormente y editarlos.

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Visita Teclas.org | f | v | t | i | Administración | Cerrar Sesión

Inicio | Documentación | Colección | Donaciones | Mapa del Museo | Acerca de

Editar Pieza

Nombre
II Europlus

Descripción
Con unidad de disco, dos manuales, tres tarjetas de expansión y dos joysticks.







Año
1979

Sala
Biblioteca

Marca
APPLE

Pieza donada por:
Bea Corredera

Imágenes
Imágenes

-  ❌
-  ❌
-  ❌
-  ❌
-  ❌
-  ❌

Tipos

- Ordenador ❌

18.2 Aspecto final de la vista Crear Pieza

2.4 Iteración 3

En la última iteración se dejó la aplicación lo mejor preparada posible para su presentación final. Se pulieron defectos que de todas las iteraciones y se cambió el estilo de la página completamente, para que fuera más actual e igual para todas las páginas. Además, se desarrollaron nuevas páginas que mejoraban la experiencia de usuario añadiéndole más información para visualizar y de una forma más amena, como las vistas del *Mapa Del Museo* o *Donaciones*. Por esta misma razón los requisitos alcanzados en esta iteración, ya se consiguieron implementar anteriormente en otras vistas y solo ofrecen una alternativa o nueva información que no estaba contemplada en los requisitos, pero se implementaron finalmente.

Requisitos alcanzados:

- RF 1.3 - **Visualizar pieza:** El sistema debe permitir a un usuario visualizar el contenido visible de una pieza.
- RF 6.3 - **Visualizar sala:** El sistema debe permitir a un usuario visualizar un mapa del edificio que permita ver todas las piezas pertenecientes a una sala.

Las vistas desarrolladas en este trabajo fueron:

Vista Acerca De

En esta ventana se facilita información sobre el museo a los visitantes de la página web. Se podrá visualizar en un mapa la ubicación del museo junto con sus horarios, conocer quienes realizaron la página web y con qué motivaciones y finalmente las personas que han ayudado en la realización de este proyecto

Modelo y Repositorio

- NA. No se usa ningún modelo o repositorio.

Servicios

- NA. No se llama a ningún servicio.

Boceto

Esta vista pretende mostrar al visitante toda la información relacionada con esta aplicación y con el museo. Se proporciona un mapa para poder llegar fácilmente al museo y un horario para planificar la visita. Finalmente se incluyen pinceladas sobre los desarrolladores y terceros.

Administración

MUSEO INFORMÁTICA

LUNES A VIERNES de 10:00h a 19:00h
Administración
Cerrar sesión
| ES ▼ |

INICIO
DOCUMENTACIÓN
COLECCIÓN
DONACIONES
ACERCA DE

Acerca del Museo de la Informática

¿Donde nos encontramos?



¿Quienes somos?

Somos unos estudiantes de Ingeniería del Software, Con nuestro TFG en mente, las ideas de nuestro tutor y basándonos en un TFG donde se desarrollaba una plataforma web para el museo de la ETSI Informática, vimos la posibilidad de usar nuevas tecnologías y aportar nuevas funcionalidades para darle un lavado de cara y que sea funcional para los estudiantes y las personas que visiten el museo.

Todo esto ha sido posible gracias a:

Agradecer a Daniel Sancho dueño del museo por cedernos toda la información sobre su colección y por reunir todas las piezas durante estos años para el disfrute y documentación de todos los visitantes. Además de permitirnos hacer su página como nuestros Trabajo de Fin de Grado, su página [teclas](http://www.teclas.org) la cual os animamos a visitar.

También hacer un especial agradecimiento a nuestro profesor de Tecnologías Web y tutor de nuestro TFG, Eduardo Guzmán de los Riscos, Rafael Pernil por guiarnos con su trabajo anterior con IONIC y Angular y a Álvaro García-Faure Torres, quien ya hizo un trabajo sobre la misma idea y la cual hemos querido ampliar.

HORARIO:
Lunes a Viernes de 10h a 19h

Universidad de Málaga -
Avda. Cervantes, 2. 29071
MALAGA - Tel. 952 13 10
00 - info@uma.es

[Inicio](#) [Documentación](#) [Colección](#) [Donaciones](#) [Acerca de](#)

Copyright ©2019 by UMA. All rights reserved.

19.1 Boceto de la vista Acerca De

Esta página no tenía ninguna complejidad desde el punto de vista de los servicios, al no llamar a ninguno, se hizo el trabajo necesario para exponer de la forma más visual la información sobre el museo y sobre nosotros y los terceros que colaboraron en este proyecto. Las únicas variaciones respecto al boceto es la disposición de las tarjetas y el contenido de ellas.



Acerca del Museo de la Informática

¿Dónde puedo visitar el museo?



El museo de la informática ubicado en el interior de la Escuela Técnica Superior de Ingeniería Informática. Creada en 1987, dispone de un amplio muestrario de piezas divididas entre las distintas salas y pasillos de la Universidad.

¿Cuándo visitarnos?: Concerta una cita con la escuela para ver todas las piezas.

Nuestra Motivación



Buenas, nos presentamos, somos Beatriz Correa Lopera, Jorge Luis Palomares Ruiz y Tomás Sayago Gutiérrez, tres estudiantes de Ingeniería del Software en la Universidad de Málaga (Promoción 2015/2019) y este es nuestro Trabajo de Fin de Grado.

En este museo podrás encontrar una amplia variedad de piezas antiguas que cubren los sectores de las máquinas de cálculo y la informática, de entre muchas marcas. Pero no dispone de una página moderna en la que poder acceder a todas ellas de una manera cómoda y visual.

Con nuestro TFG en mente, las ideas de nuestro tutor y basándonos en un TFG donde se desarrollaba una plataforma web para el museo de la ETSI Informática, vimos la posibilidad de usar nuevas tecnologías y aportar nuevas funcionalidades para darle un lavado de cara y que sea funcional para los estudiantes y las personas que visiten el museo. La idea era realizar una página web adaptativa para distintos dispositivos, ya sean de escritorio como móviles, la cual permitiese obtener información sobre las diferentes piezas del museo de la informática ubicado en la ETSI Informática.

19.2 Aspecto final de la vista Acerca De

Vista Mapa Del Museo

En esta ventana se facilita un mapa sobre el museo a los visitantes de la página web. Se podrá visualizar en el mapa la ubicación en plantas y salas del museo junto con la descripción de salas más concretas. Pulsando en las distintas salas se mostrará información a la derecha sobre la sala en cuestión y las piezas ubicadas dentro de ella.

Modelo y Repositorio

- Piece.java – PieceRepository.java
- Room.java – RoomRepository.java

Servicios

- findAll() - Devuelve una lista con todas las piezas de la base de datos.
- findByRoomNameLike() - Devuelve una lista con todas las piezas de una sala.

Boceto

En esta vista se adaptó la idea del mapa del Museo Metropolitano de Arte(**capítulo 1.6**), pero adaptada al museo. En lugar de un mapa del mundo se usó un mapa proporcionado nuestro tutor. En este mapa se muestran las distintas salas donde están expuestas las piezas: en un panel de la izquierda se muestran las piezas encontradas, de una manera más simple (con menos datos) que en la vista colección. Por otro lado, se puede encontrar información de cada sala.



20.1 Boceto vista Mapa Del Museo

Para realizar esta página se utilizaron las etiquetas *map* de *html* que permiten pulsar una determina parte de una imagen, además se usó el evento de *onmouseover* and *onmouseout* para que cambiara el color de la zona seleccionada y que fuera más visual. Al pulsar se llama a un servicio que busca la sala seleccionada y otro que busca las piezas de esa sala. Finalmente se muestra la información en el cuadro derecho.



Biblioteca

Localizada en la: **Planta 3**
 Descripción: **Biblioteca de la ETSI**
 Piezas en esta sala: **4**

HP-12C
 Calculadora de bolsillo con funciones financieras y programable. Probablemente una de las máquinas más...

II Europlus
 Con unidad de disco, dos manuales, tres tarjetas de expansión y dos joysticks.

HP 5036A Microprocessor LAB
 Aquí tenemos un entrenador lógico basado en el procesador Intel 8085. Se vendía dentro de un...

20.2 Aspecto final de la vista Colección

3

Conclusiones y líneas futuras

3.1 Conclusiones

Este proyecto ha intentado dar respuesta a la necesidad de disponer de un portal web dinámico para poder exponer, de forma virtual, las piezas del museo de nuestra escuela, a la par que experimentábamos un desarrollo real y total de todas las capas que compone una aplicación web. Si analizamos nuestro trabajo se puede decir que ambos puntos han sido un éxito. Por un lado, se ha sabido dar respuesta a las características demandadas y por el otro de manera más fácil o difícil todos los miembros hemos sido capaces de desarrollar para cada una de las capas. Siendo críticos, hay algunos aspectos de la página que se pueden mejorar, sobre todo los visuales, para los cuales no considero que hallamos sabido encontrar un diseño llamativo que nos complaciera a todos.

El trabajo en grupo ha tenido sus ventajas (que han sido mayores) e inconvenientes, aunque por un lado se tenía la seguridad de poder recurrir a la ayuda de otra persona que tuviera el mismo objetivo, como también muchas veces poder reutilizar componentes creados por unos compañeros. Las decisiones en grupo no siempre han gustado por igual e intentar repartir el trabajo equitativamente, aunque con éxito fue una tarea laboriosa de planificación y organización.

Personalmente me llevo una experiencia inolvidable después de estos meses de trabajo, de solucionar problemas que surgían constantemente, de reuniones para planificar las iteraciones y de discusiones por elegir el color de fondo de la página que de seguro darán sus frutos en muy poco tiempo. Respecto a la aplicación web , si bien casi la totalidad de los requisitos se han cumplido y cumple unas pruebas de calidad, no creemos que tenga un gran impacto debido al volumen de visitas del museo y el interés de los estudiantes en él mismo. Por nuestra parte esperamos que la aplicación consiga cambiar esto y despertar el interés de los estudiantes y visitantes por la historia que hay en cada obra del museo de nuestra facultad.

3.2 Líneas futuras

Se han trabajado y desarrollado las partes más fundamentales que requería esta aplicación, pero durante la etapa de desarrollo han ido surgiendo ideas y propuestas que nos gustaría presentar. Estas propuestas mejorarían más aun la aplicación y podrían ser llevadas a cabo en un futuro por nosotros mismo o terceras personas con interés en ellas. Estas son:

1. **Editor de texto para introducir piezas u otros elementos:** aunque las páginas sobre las piezas están bien construidas para ofrecer un mismo formato para todas las piezas no da libertad al administrador para establecer un formato de texto determinado, elegir la disposición de este o introducir imágenes en ciertas partes. La solución sería la implementación de un editor de texto como el de *Moodle*. Que permitiría al usuario

personalizar cada una de las páginas de piezas.

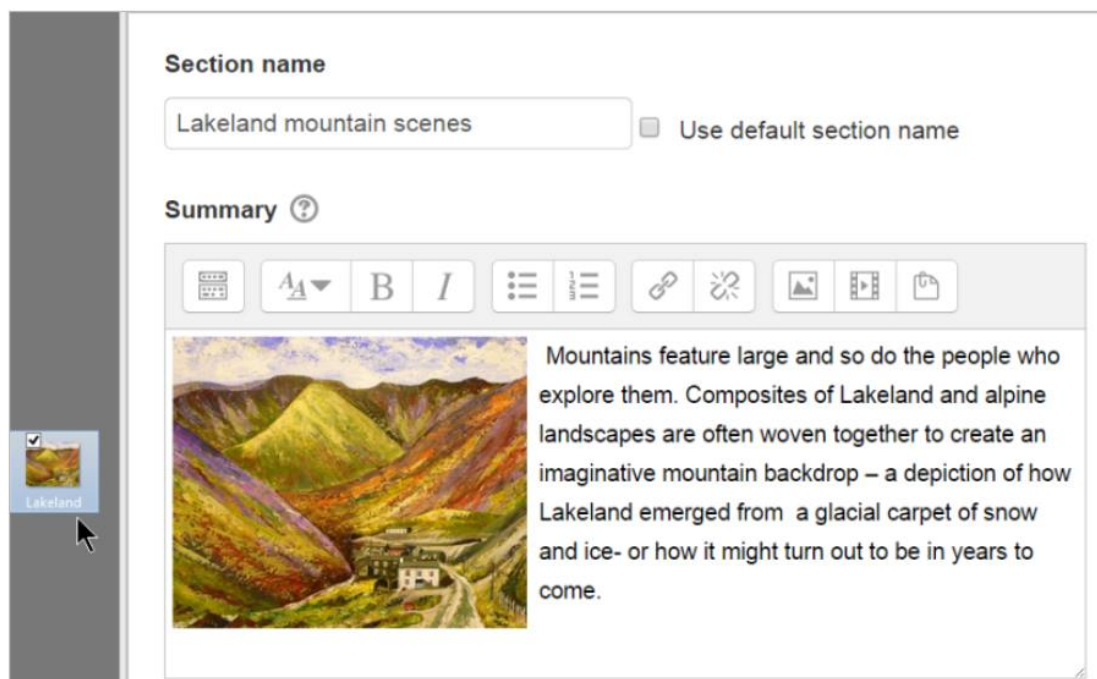


Figura 21.1 Editor de texto de Moodle.

2. **Internalización:** En nuestro objetivo inicial estaba la internalización de la página para español e inglés. Era un trabajo que ya se había realizado anteriormente y no supuso mucho esfuerzo anteriormente, pero no se tuvo en cuenta que habría que tener una base de datos en español y otra en inglés. Esto es así porque no solo habría que crear un fichero de internalización con las traducciones de los elementos del código de la página, sino también los contenidos que llegan de la base de datos.
3. **Calendario de eventos:** Actualmente el museo no tiene eventos, pero sería interesante crear un calendario donde el museo pudiera establecer fechas para determinados eventos y los visitantes pudiesen verlos y apuntarse.
4. **Funcionalidades para usuarios:** Nos parece buena idea incluir usuarios con permisos de visitante en la aplicación, que pudiesen escribir comentarios en las piezas, darle me gusta a su pieza favorita. Registrarse en eventos del calendario mencionado anteriormente y las demás

funcionalidades que suelen tener las redes sociales que se pudieran ajustar al modelo del museo.

- 5. Concertar visitas:** Actualmente para visitar el museo hay que pedir en secretaria cita previa, sin embargo, esta tarea se podría realizar a través de la aplicación.
- 6. Procesamiento por lotes:** Con el volumen de datos que hemos trabajado no ha sido necesario un procesamiento por lotes, pero sería interesante si en algún momento el número de piezas fuera más elevado.

Referencias

Ionic Framework (2019, January 21). API Index.

<https://ionicframework.com/docs/api/>

Material Angular IO (s.f.). Paginator.

<https://material.angular.io/components/paginator/overview>

Material Angular IO (s.f.). Card.

<https://material.angular.io/components/card/overview>

Museo Picasso Málaga (s.f.). Inicio

<https://www.museopicassomalaga.org/>

Metropolitan Museum of Art (s.f.). Inicio

<https://www.metmuseum.org/>

Brillout (2017, November 3). Github. Awesome Angular Components.

<https://github.com/brillout/awesome-angular-components>

Componentes angular. Tabla con filtro:

Material Angular IO (s.f.). Table.

<https://material.angular.io/components/table/overview>

EpokK (2013, August 7). Stackoverflow. Limit the length of a string with angularjs .

<https://stackoverflow.com/questions/18095727/limit-the-length-of-a-string-with-angularjs>

Piotrdz (2019, April 7). Angular Slider Github. Ng5 Slider.

<https://angular-slider.github.io/ng5-slider/demos#styled-slider>

ccwasden (2016, October 21). Stackoverflow. Angular 2 get values of multiple checked checkboxes.

<https://stackoverflow.com/questions/34997128/angular-2-get-values-of-multiple-checked-checkboxes>

The Telerik Team (2017, March 28). Telerik. From ng-model to ngModel.
<https://www.telerik.com/blogs/ng-model-ngmodel>

Oliver Drotbohm (2011, November 18). Stackoverflow. How do i query for dates in spring data mongoDB repository.
<https://stackoverflow.com/questions/8179519/how-do-i-query-for-dates-in-spring-data-mongodb-repository>

Dan Newton (2017, May 29). LANKY DAN DEV BLOG. Embedded documents with Spring Data and MongoDB.
<https://lankydanblog.com/2017/05/29/embedded-documents-with-spring-data-and-mongodb/>

Marcela Sena (2017, Mar 28). Medium. Como pasar de SQL a NoSQL sin sufrir.
<https://medium.com/techwomenc/como-pasar-de-sql-a-nosql-sin-sufrir-e34dd22349e5>

Baeldung (2018, November 6). Baeldung. A Guide to Queries in Spring Data MongoDB.
<https://www.baeldung.com/queries-in-spring-data-mongodb>

Spring Reference Documentation (s.f.). MongoDB repositories.
<https://docs.spring.io/spring-data/mongodb/docs/1.2.0.RELEASE/reference/html/mongo.repositories.html>

Oliver Drotbohm (2015, October 11) Stackoverflow. Spring Data MongoDB Repositories Query multiple fields.
<https://stackoverflow.com/questions/33056062/spring-data-mongodb-repositories-query-multiple-fields>

Jimmym715 (2012, September 7) Stackoverflow. Knockout.js two-way Binding: Number Formatted as String.
<https://stackoverflow.com/questions/12320622/knockout-js-two-way-binding-number-formatted-as-string>

Pivotal Software (2019). Spring Official Web. Building a RESTful Web Service.
<https://spring.io/guides/gs/rest-service/>

Pivotal Software (2019). Spring Official Web. Accessing MongoDB Data with REST. <https://spring.io/guides/gs/accessing-mongodb-data-rest/>

Byte Code (2018, July 3). Cómo crear un API RESTful con MongoDB y Spring boot - Conectando Java a NoSQL.
<https://www.youtube.com/watch?v=aVdrV3847xU>

Nic Raboy (2018, September 28). The Polyglot Developer. Developing A RESTful API With Node.js And MongoDB Atlas.
<https://www.thepolyglotdeveloper.com/2018/09/developing-restful-api-nodejs-mongodb-atlas/>

Innova Engineers (2018, January 13). Introduction to Spring Data MongoDB Cloud.
<https://www.youtube.com/watch?v=FbIAs8278h4>

Innova Engineers (2018, April 15). Spring Boot + Spring Data MongoDB How to perform CRUD Operations.
<https://www.youtube.com/watch?v=A5NWMpp4H7s>

Pivotal Software (2019). Spring API. Interface MongoRepository.
<https://docs.spring.io/spring-data/mongodb/docs/current/api/org/springframework/data/mongodb/repository/MongoRepository.html>
<https://www.baeldung.com/spring-boot-war-tomcat-deploy>

MuleSoft (2019). Advanced Rest Client.
<https://install.advancedrestclient.com/install>

JetBrains s.r.o. (2019, February 1). Creating a Remote Server Configuration.
<https://www.jetbrains.com/help/idea/creating-a-remote-server-configuration.html>

JetBrainsTV (2013, June 13). Deployment and Remote Hosts in PhpStorm - PhpStorm Video Tutorial.
<https://www.youtube.com/watch?v=AHK20LWEWXQ>

Umesh (2018, September 4). JavadevJournal. Change the default port in Spring Boot.
<https://www.javadevjournal.com/spring-boot/change-the-default-port-in-spring-boot/>

Didin J (2018, November 9). Djamware. Ionic 4, Angular 7 and Cordova Tutorial: Build CRUD Mobile Apps.

<https://www.djamware.com/post/5be52ce280aca72b942e31bc/ionic-4-angular-7-and-cordova-tutorial-build-crud-mobile-apps>

Angular Official Web (2010-2019). Angular - Getting Started.
<https://angular.io/guide/quickstart>

Daniel Sancho(s.f.) Teclas. Principal.
<http://www.teclas.org/index.php>

Apéndice A

Manual de Instalación

Requerimientos:

Servidor

Para el servidor, necesitamos el zip del proyecto "museumServer". Este zip se descomprime y se siguen los siguientes pasos:

1. Abrir la terminal dentro de la carpeta museumServer.
2. Escribir el comando "gradlew build" en la terminal.
3. Dentro del proyecto entrar en build/libs donde se encontrara un archivo con extensión ".jar".
4. Finalmente abrir la consola en ese directorio y usar el comando para desplegarlo "java -jar <nombre del fichero.jar>".

Cliente

Por parte del cliente necesitaremos además del zip del cliente "museumClient", instalar todas las dependencias, para esto primero necesitaremos instalar *Node.js* y *Angular CLI* siguiendo los siguientes pasos:

1. Instalar *Node.js* que normalmente viene acompañado con npm(instalar npm aparte si no se instaló con Node.js).
2. Instalar *Angular CLI* que permite inicializar, desarrollar y mantener las aplicaciones *Angular*, con el comando "npm install -g @angular/cli".
3. Entramos a la carpeta museumClient y abrimos una terminal en ese directorio.
4. Finalmente usamos el comando "ng serve -open" para compilar y desplegar el cliente.

Apéndice B

Manual de Usuario

1. Navegación por la web

Al acceder a la web, comenzaremos en la página Inicio. En la parte superior siempre encontraremos una cabecera con enlaces a las diferentes páginas que podemos acceder y que serán explicadas a lo largo de este documento. La página en la que nos encontramos en cada momento se verá señalada en la cabecera.

1.1. Inicio

En esta página encontraremos un carrusel donde irán apareciendo piezas y sobre las que, si hacemos clic, nos llevará a la página de información de dicha pieza.

En la parte inferior encontraremos tres tarjetas con información de tres marcas y sobre las que, si hacemos clic, nos llevará a la ventana de colección con dicha marca seleccionada.

1.2. Documentación

Al acceder a dicha página, encontraremos un listado con todos los documentos que podemos encontrar, así como realizar una búsqueda. Al hacer clic sobre uno de los documentos, podremos ver los diferentes artículos pertenecientes a dicho documento.

1.3. Colección

Aquí podremos realizar una búsqueda de las piezas pudiendo filtrar por diferentes criterios. Al hacer clic sobre un resultado, accederemos a la ventana con la información de dicha pieza detallada.

1.4 Donaciones

En esta ventana encontramos un listado de las personas que han realizado una o varias donaciones a este museo. Además, veremos información sobre cómo donar. Así como un Top de los tres mayores donadores hasta el momento.

1.5 Mapa del Museo

En esta ventana, se puede visualizar el mapa del museo. Al clicar sobre una de las salas, aparecerá en la parte derecha información de la sala seleccionada, su localización, así como un listado de las piezas que podemos encontrar en dicha sala.

1.6 Acerca de

En la ventana acerca de, aparece información de dónde visitar este museo, así como información sobre la motivación de realizar la misma y agradecimientos.

1.7 Accesos Directos

Conjunto de accesos directos a diferentes redes sociales de la Universidad de Málaga, Escuela Técnica Superior de Ingeniería Informática, así como del museo.

2. Zona Administración

Se accederá al hacer clic sobre el botón:

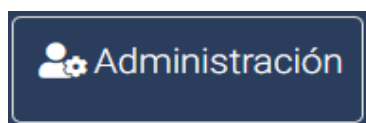


Figura 1 Apéndice B. Botón Administración

A esta zona sólo podrán acceder los usuarios que tengan permisos de administrador. Para ello deben iniciar sesión con sus cuentas válidas.

Los iconos de edición y eliminación a los que nos referiremos a continuación son los siguientes:



<i>Editar</i>	<i>Eliminar</i>
	

Figura 2 Apéndice C. Iconos de edición y eliminación

2.1 Documentación

Ventana para la creación, edición y eliminación de documentos. Al entrar nos muestra el listado de todos los documentos que hay en la base de datos. También se puede realizar una búsqueda para encontrar el documento buscado más rápidamente.

2.1.1 Crear

Al pulsar sobre el botón de creación, nos lleva a otra ventana en la que veremos todos los posibles campos a rellenar para la creación. Los datos por rellenar con la información son: nombre (obligatorio), descripción, imágenes y artículos.

Las imágenes se añadirán indicando la URL donde se encuentra dicha imagen y se podrán añadir todas las que se desee. Para añadir artículos, se deberá crear uno o varios nuevos (los que se desee añadir a dicho documento). Para ello debemos pulsar sobre el botón “Crear Artículo”. Aparecerá un pop-up en el que podremos rellenar los datos de dicho artículo: nombre (obligatorio), descripción e imágenes.

Las imágenes para este artículo se añadirán de la misma forma mencionada anteriormente. Tras rellenar todos los campos se pulsará “Guardar” para añadir el artículo al documento o “Cerrar” si no deseamos guardar la información. Si hemos guardado, nos aparecerá el nuevo artículo en el listado. Cuando finalicemos y deseemos crear el documento con toda la información añadida, pulsaremos el botón “Crear Documento”. Si no deseamos guardar los datos y abandonar el proceso de creación, pulsaremos el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de documentos y si hemos creado uno nuevo, dicho documento aparecerá en dicha lista. Además, si se ha realizado una creación, aparecerá un mensaje indicando que se ha realizado correctamente.

2.1.2 Editar

Accederemos haciendo clic sobre el icono de “Editar” sobre el documento deseado. La edición será muy parecida a la creación, con la diferencia de que al acceder nos aparecerán los campos rellenos con la información contenida en la base de datos.

Podremos modificar o mantener alguno o varios datos (Pero siempre deberá estar el campo Nombre relleno). Podremos añadir más imágenes o eliminar alguna de las ya existentes pulsando sobre el icono de “Eliminar”. Además, se podrán añadir nuevos artículos, modificar uno ya existente o eliminarlo. Estos cambios se realizan de forma similar a la mencionada anteriormente.

Cuando finalicemos y deseemos editar el documento con toda la información modificada, pulsaremos el botón “Editar Documento”. Si no deseamos guardar los datos y abandonar el proceso de edición, pulsaremos el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de documentos y si hemos modificado uno, dicho documento se verá modificado. Además, si se ha realizado una edición, aparecerá un mensaje indicando que se ha realizado correctamente.

2.1.3 Eliminar

Para la eliminación de un documento, tendremos que hacer clic sobre el icono de “Eliminar” del documento deseado. Al hacer clic nos aparecerá un pop-up preguntando si deseamos eliminar dicho documento. Si es así, pulsaremos el botón “Borrar”, en caso contrario, el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de documentos y si hemos eliminado uno, dicho documento no aparecerá en el listado.

2.2 Colección

Ventana para la creación, edición y eliminación de piezas. Al entrar nos muestra el listado de todas las piezas que hay en la base de datos. También se puede realizar una búsqueda para encontrar la pieza buscada más rápidamente.

2.2.1 Crear

Al pulsar sobre el botón de creación, nos lleva a otra ventana en la que veremos todos los posibles campos a rellenar para la creación. Los datos por rellenar con la información son: nombre (obligatorio), descripción, año, sala, marca, donante de la pieza, imágenes y tipos. Las imágenes se añadirán indicando la URL donde se encuentra dicha imagen y se podrán añadir todas las que se desee.

Tanto las salas como las marcas y los tipos son desplegables de los que se podrá seleccionar la sala, la marca y uno o varios tipos según se desee. Para añadir tipos, se selecciona un elemento del desplegable y se pulsa el botón añadir. Dicho tipo aparecerá en una lista inferior con un símbolo de eliminación al lado por si ocurre una equivocación a la hora de seleccionarlo.

Cuando finalicemos y deseemos crear la pieza con toda la información añadida, pulsaremos el botón “Crear Pieza”. Si no deseamos guardar los datos y abandonar el proceso de creación, pulsaremos el botón “Volver”. En ambos casos volveremos a la página anterior con el listado de piezas y si hemos creado una nueva, dicha pieza aparecerá en dicha lista. Además, si se ha realizado una creación, aparecerá un mensaje indicando que se ha realizado correctamente.

2.2.2 Editar

Accederemos haciendo clic sobre el icono de “Editar” sobre la pieza deseada. La edición será muy parecida a la creación, con la diferencia de que al acceder nos aparecerán los campos rellenos con la información contenida en la base de datos.

Podremos modificar o mantener alguno o varios datos (Pero siempre deberá estar el campo Nombre relleno). Podremos añadir más imágenes o eliminar alguna de las ya existentes pulsando sobre el icono de “Eliminar”, al igual que con los tipos.

Cuando finalicemos y deseemos editar la pieza con toda la información modificada, pulsaremos el botón “Editar Pieza”. Si no deseamos guardar los datos y abandonar el proceso de edición, pulsaremos el botón “Volver”. En ambos casos volveremos a la página anterior con el listado de piezas y si hemos modificado alguna, dicha pieza se verá modificada. Además, si se ha realizado una edición, aparecerá un mensaje indicando que se ha realizado correctamente.

2.2.3 Eliminar

Para la eliminación de una pieza, tendremos que hacer clic sobre el icono de “Eliminar” de la pieza deseada. Al hacer clic nos aparecerá un pop-up preguntando si deseamos eliminar dicha pieza. Si es así, pulsaremos el botón “Borrar”, en caso contrario, el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de piezas y si hemos eliminado una, dicha pieza no aparecerá en el listado.

2.3 Marca

Ventana para la creación, edición y eliminación de marcas. Al entrar nos muestra el listado de todas las marcas que hay en la base de datos. También se puede realizar una búsqueda para encontrar la marca buscada más rápidamente.

2.3.1 Crear

Al pulsar sobre el botón de creación, nos lleva a otra ventana en la que veremos todos los posibles campos a rellenar para la creación. Los datos por rellenar con la información son: nombre (obligatorio) y descripción.

Cuando finalicemos y deseemos crear la marca con toda la información añadida, pulsaremos el botón “Crear Marca”. Si no deseamos guardar los datos y abandonar el proceso de creación, pulsaremos el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de marcas y si hemos creado una nueva, dicha marca aparecerá en dicha lista. Además, si se ha realizado una creación, aparecerá un mensaje indicando que se ha realizado correctamente.

2.3.2 Editar

Accederemos haciendo clic sobre el icono de “Editar” sobre la marca deseada. La edición será muy parecida a la creación, con la diferencia de que al acceder nos aparecerán los campos rellenos con la información contenida en la base de datos. Podremos modificar o mantener alguno o varios datos (Pero siempre deberá estar el campo Nombre relleno).

Cuando finalicemos y deseemos editar la marca con toda la información modificada, pulsaremos el botón “Editar Marca”. Si no deseamos guardar los datos y abandonar el proceso de edición, pulsaremos el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de marcas y si hemos modificado alguna, dicha marca se verá modificada. Además, si se ha realizado una edición, aparecerá un mensaje indicando que se ha realizado correctamente.

2.3.3 Eliminar

Para la eliminación de una marca, tendremos que hacer clic sobre el icono de “Eliminar” de la marca deseada. Al hacer clic nos aparecerá un pop-up preguntando si deseamos eliminar dicha marca. Si es así, pulsaremos el botón “Borrar”, en caso contrario, el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de marcas y si hemos eliminado una, dicha marca no aparecerá en el listado.

2.4 Tipo

Ventana para la creación, edición y eliminación de tipos. Al entrar nos muestra el listado de todos los tipos que hay en la base de datos. También se puede realizar una búsqueda para encontrar el tipo buscado más rápidamente.

2.4.1 Crear

Al pulsar sobre el botón de creación, nos lleva a otra ventana en la que veremos todos los posibles campos a rellenar para la creación. El único por rellenar con la información es el nombre (obligatorio).

Cuando finalicemos y deseemos crear el tipo con toda la información añadida, pulsaremos el botón “Crear Tipo”. Si no deseamos guardar los datos y abandonar el proceso de creación, pulsaremos el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de tipos y si hemos creado uno nuevo, dicho tipo aparecerá en dicha lista. Además, si se ha realizado una creación, aparecerá un mensaje indicando que se ha realizado correctamente.

2.4.2 Editar

Accederemos haciendo clic sobre el icono de “Editar” sobre el tipo deseado. La edición será muy parecida a la creación, con la diferencia de que al acceder nos aparecerá el campo relleno con la información contenida en la base de datos. Podremos modificar el campo “Nombre”, pero nunca dejarlo vacío.

Cuando finalicemos y deseemos editar el tipo con toda la información

modificada, pulsaremos el botón “Editar Tipo”. Si no deseamos guardar los datos y abandonar el proceso de edición, pulsaremos el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de tipos y si hemos modificado alguno, dicho tipo se verá modificado. Además, si se ha realizado una edición, aparecerá un mensaje indicando que se ha realizado correctamente.

2.4.3 Eliminar

Para la eliminación de un tipo, tendremos que hacer clic sobre el icono de “Eliminar” del tipo deseado.

Al hacer clic nos aparecerá un pop-up preguntando si deseamos eliminar dicho tipo. Si es así, pulsaremos el botón “Borrar”, en caso contrario, el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de tipos y si hemos eliminado uno, dicho tipo no aparecerá en el listado.

2.5 Sala

Ventana para la creación, edición y eliminación de salas. Al entrar nos muestra el listado de todas las salas que hay en la base de datos. También se puede realizar una búsqueda para encontrar la sala buscada más rápidamente.

2.5.1 Crear

Al pulsar sobre el botón de creación, nos lleva a otra ventana en la que veremos todos los posibles campos a rellenar para la creación. Los datos por rellenar con la información son: nombre (obligatorio), descripción, localización.

Cuando finalicemos y deseemos crear la sala con toda la información añadida, pulsaremos el botón “Crear Sala”. Si no deseamos guardar los datos y abandonar el proceso de creación, pulsaremos el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de salas y si hemos creado una nueva, dicha sala aparecerá en dicha lista. Además, si se ha realizado una creación, aparecerá un mensaje indicando que se ha realizado correctamente.

2.5.2 Editar

Accederemos haciendo clic sobre el icono de “Editar” sobre la sala deseada. La edición será muy parecida a la creación, con la diferencia de que al acceder nos aparecerán los campos rellenos con la información contenida en la base de datos. Podremos modificar o mantener alguno o varios datos (Pero siempre deberá estar el campo Nombre relleno).

Cuando finalicemos y deseemos editar la sala con toda la información modificada, pulsaremos el botón “Editar Sala”. Si no deseamos guardar los datos y abandonar el proceso de edición, pulsaremos el botón “Volver”. En ambos casos volveremos a la página anterior con el listado de salas y si hemos modificado alguna, dicha sala se verá modificada. Además, si se ha realizado una edición, aparecerá un mensaje indicando que se ha realizado correctamente.

2.5.3 Eliminar

Para la eliminación de una sala, tendremos que hacer clic sobre el icono de “Eliminar” de la sala deseada. Al hacer clic nos aparecerá un pop-up preguntando si deseamos eliminar dicha sala. Si es así, pulsaremos el botón “Borrar”, en caso contrario, el botón “Volver”.

En ambos casos volveremos a la página anterior con el listado de salas y si hemos eliminado una, dicha sala no aparecerá en el listado.

Para salir de la zona de administración, sólo habrá que clicar en el botón Cerrar Sesión.

3.Manual de usuario

En este último enlace, encontramos toda la información detallada a lo largo de este documento.

Apéndice C

Documento General
de Requisitos

DOCUMENTO GENERAL DE REQUISITOS

Clave: **TFG-MUSEO-INFORMATICA-DGR**

N.º **09**
Páginas:

Museo de la Informática
Título: **DOCUMENTO GENERAL DE REQUISITOS**

"Empresa registrada conforme a las normas UNE-EN-ISO-9001"
Laboratorio de ensayo acreditado por **ENAC** con acreditación nº **91/LE208**

1. INTRODUCCIÓN

1.1. OBJETIVOS

El objetivo de este documento es recoger, analizar y definir las características y necesidades de alto nivel el sistema <<*Museo de la Informática*>>. Se centrará en describir las expectativas de cada una de las partes del proyecto y de los usuarios finales. Los detalles de cómo <<*Museo de la Informática*>> cumple estas necesidades se detallan en los requisitos del sistema y en las especificaciones adicionales.

1.2. METODOLOGÍA

Para la realización del sistema <<*Museo de la Informática*>> se utilizará una metodología iterativa dónde se irán completando los requisitos por iteraciones. En un principio se implementarán los requisitos básicos que dará a la aplicación web funcionalidades reducidas, aumentando estas a lo largo del desarrollo. El cumplimiento de todos los requisitos estará limitado por el tiempo y las complicaciones encontradas durante la vida del proyecto.

- IT1: Iteración 1
- IT2: Iteración 2
- IT3: Iteración 3

1.3. RESUMEN DE LOS PARTICIPANTES

Nombre	Rol
Usuario	Un usuario podrá visualizar los contenidos de la aplicación web.
Administrador	Un administrador tiene los mismos permisos que un usuario y, además, puede acceder a una zona restringida de administración, proporcionando los credenciales necesarios. En

	esta zona podrá crear, actualizar o borrar información de la web que se almacena en una base de datos.
--	--

1.4. REQUISITOS FUNCIONALES


Requisito Funcional	Prioridad
RF 1 - Crear pieza: La pieza debe tener los atributos nombre, texto, año, fecha de creación, lista de imágenes, código QR, nombre de la sala, lista de tipos y nombre de la marca a la que pertenece.	ALTA (IT2)
RF 1.1- Editar pieza: El sistema permite al administrador la edición de piezas que ya habían sido creadas anteriormente.	ALTA (IT2)
RF 1.2 - Eliminar pieza: El sistema debe permitir al administrado eliminar una pieza que ya había sido creada anteriormente.	ALTA (IT2)
RF 1.3- Visualizar pieza: El sistema debe permitir a un usuario visualizar el contenido visible de una pieza .	ALTA (IT1)
RF 2 - Crear documento: Un documento debe tener los atributos nombre, imagen y lista de artículos que contiene.	ALTA (IT2)
RF 2.1 - Editar documento: El sistema permite al administrador la edición de documentos que ya habían sido creadas anteriormente.	ALTA (IT2)
RF 2.2 - Eliminar documento: El sistema debe permitir al administrado eliminar un documento que ya había sido creado anteriormente.	ALTA (IT2)
RF 2.3 - Visualizar documento: El sistema debe permitir a un usuario visualizar el contenido visible de un documento .	ALTA (IT1)
RF 3 - Crear artículo: Un artículo debe tener los atributos nombre, texto y lista de imágenes asociadas.	ALTA (IT2)

RF 3.1 - Editar artículo: El sistema permite al administrador la edición de artículos que ya habían sido creadas anteriormente.	ALTA (IT2)
RF 3.2 - Eliminar artículo: El sistema debe permitir al administrador eliminar un artículo que ya había sido creado anteriormente.	ALTA (IT2)
RF 3.3 - Visualizar artículo: El sistema debe permitir a un usuario visualizar el contenido visible de un artículo .	ALTA (IT1)
RF 4 - Crear marca: Si se intenta añadir una pieza con una marca que no está creada, el sistema deberá permitir crear una nueva marca aportando su nombre y su texto.	ALTA (IT2)
RF 4.1 - Editar marca: El sistema debe permitir al administrador la edición de una marca que ya había sido creada anteriormente.	ALTA (IT2)
RF 4.2 - Eliminar marca: El sistema debe permitir al administrador eliminar una marca que ya había sido creada anteriormente.	ALTA (IT2)
RF 5 - Crear tipo: Si se intenta añadir una pieza con un tipo que no está creado, el sistema deberá permitir crear un nuevo tipo aportando su nombre.	ALTA (IT2)
RF 5.1 - Editar tipo: El sistema debe permitir al administrador la edición de un tipo que ya había sido creado anteriormente.	ALTA (IT2)
RF 5.2 - Eliminar tipo: El sistema debe permitir al administrador eliminar un tipo que ya había sido creado anteriormente.	ALTA (IT2)
RF 6 - Crear sala: La sala debe tener los atributos nombre, localización y texto.	MEDIA (IT2)
RF 6.1- Editar sala: El sistema permite al administrador la edición de salas que ya habían sido creadas anteriormente.	MEDIA (IT2)
RF 6.2 - Eliminar sala: El sistema debe permitir al administrado eliminar una sala que ya había sido creada anteriormente.	MEDIA (IT2)

RF 6.3 - Visualizar sala: El sistema debe permitir a un usuario visualizar un mapa del edificio que permita ver todas las piezas pertenecientes a una sala .	MEDIA (IT2)
RF 7 - Administración : El sistema debe tener una pantalla de inicio de sesión como administrador que acceda a la zona restringida.	ALTA (IT2)
RF 8 - Cronograma o línea temporal: El sistema tendrá un cronograma o línea temporal donde se mostrará al usuario la evolución de las piezas por orden cronológico.	BAJA (IT3)
RF 9 - Seguridad servidor: El sistema deberá contar con un método de cifrado que impida el acceso a los datos por usuarios no autorizados.	MEDIA (IT3)
RF 10 - Servidor remoto : El sistema deberá constar de un servidor remoto al cual se pueda acceder mediante servicios REST.	MEDIA (IT3)
RF 11 - Dominio remoto: El sistema deberá estar disponible y accesible a cualquier usuario bajo un dominio predeterminado.	MEDIA (IT3)
RF 12 - Manual de usuario : El sistema contará con un botón que accede al manual de usuario, donde se podrá encontrar información sobre el uso de la zona restringida para administradores.	ALTA (IT3)

2. REQUISITOS DE DOCUMENTACIÓN

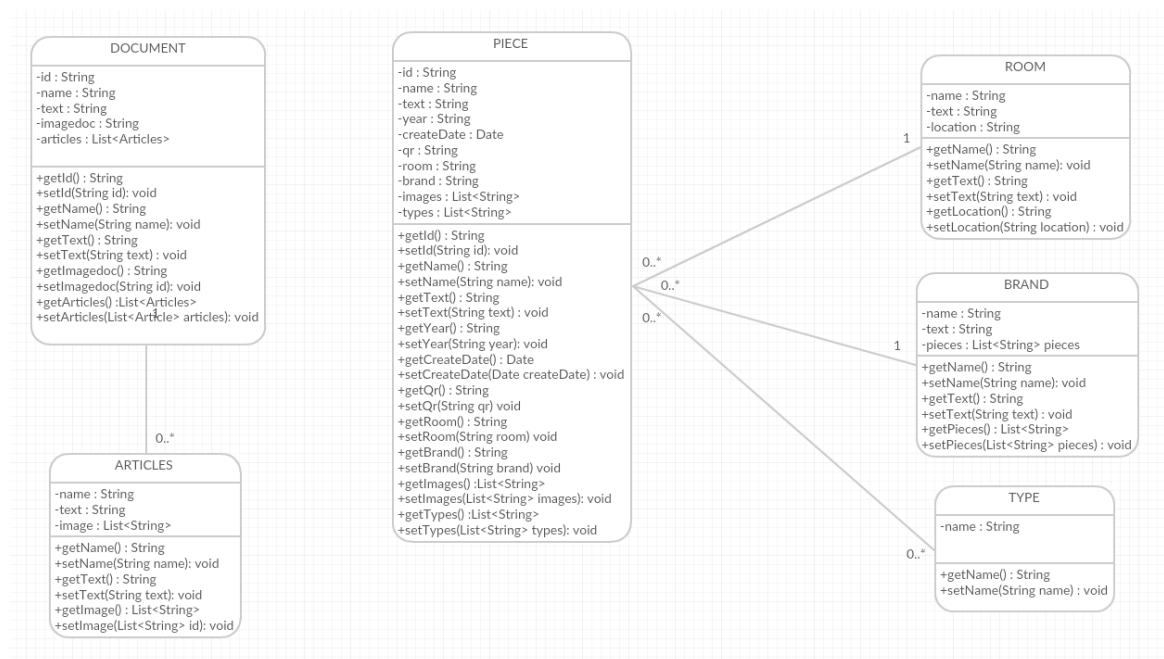
2.1. MANUAL DE USUARIO

El sistema contará con un botón (icono de ayuda ) que accede al manual de usuario, donde se podrá encontrar información sobre el uso de la zona restringida para administradores.

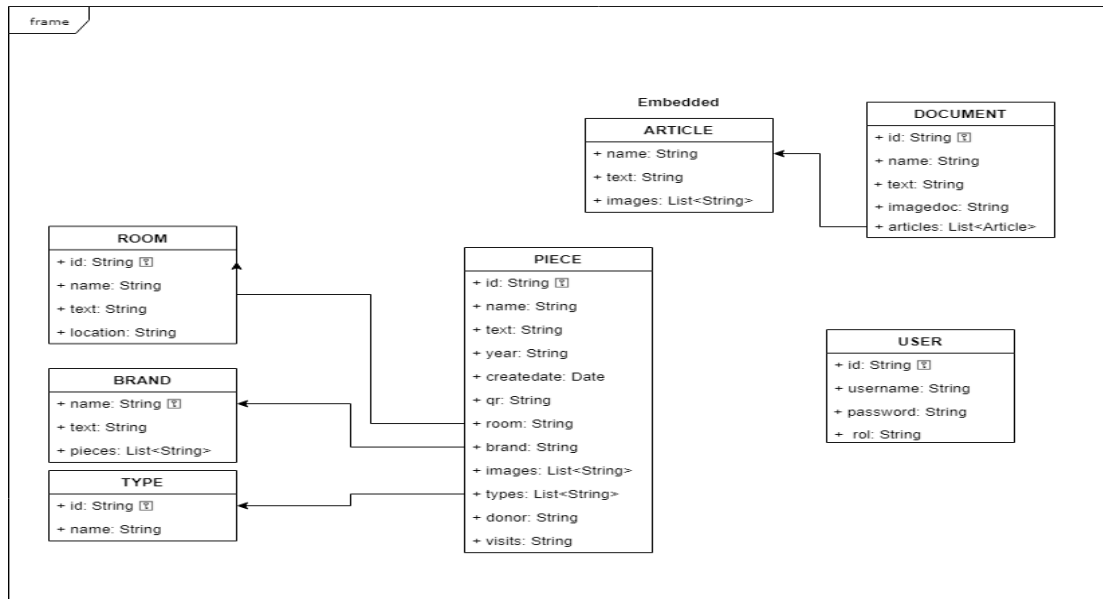
Más específicamente, se podrá encontrar información para gestionar la información de la base de datos a través de una herramienta con interfaz gráfica como es la misma aplicación web.

Esta sección de ayuda te guiará en la creación, edición o eliminación de Documentos, Artículos, Piezas, Marcas, Tipos y Salas.

2.2. MODELO DEL DOMINIO



2.3. MODELO CONCEPTUAL DE LA BASE DE DATOS

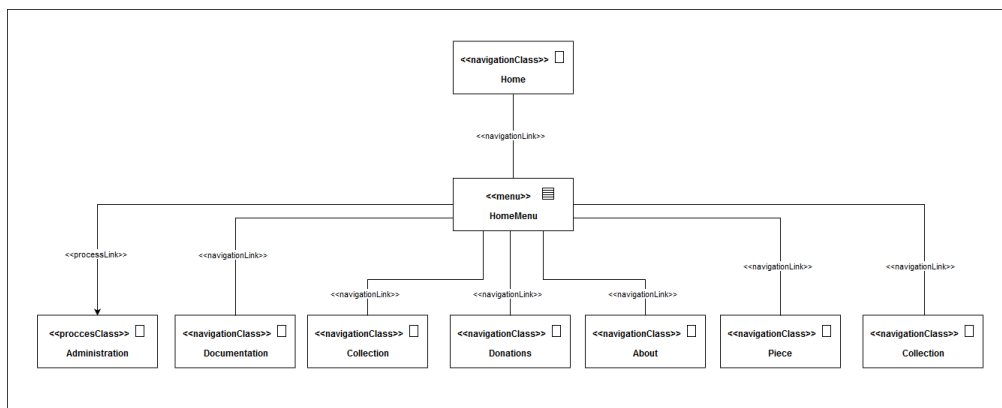


2.4. MODELOS DE NAVEGACIÓN

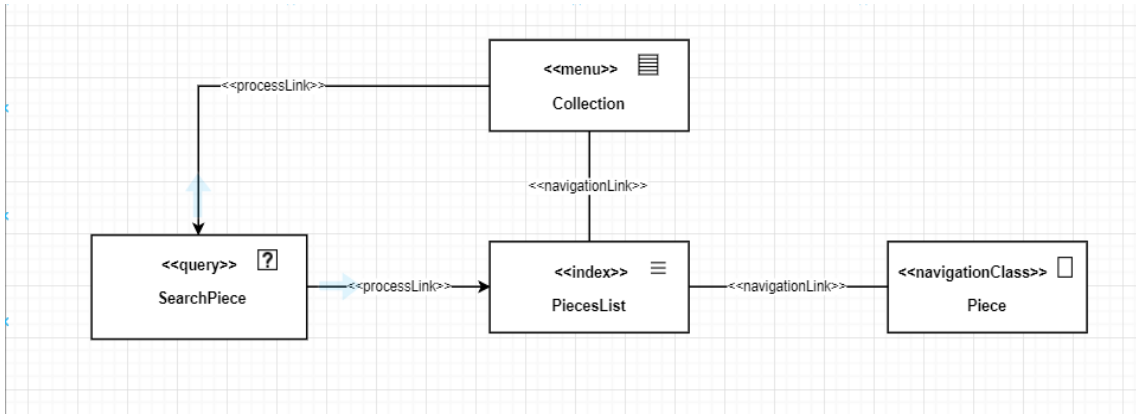
- HOME:

Este modelo es el único donde se pueden ver las navegaciones mediante los hipervínculos situados en la cabecera y en el pie de la aplicación web.

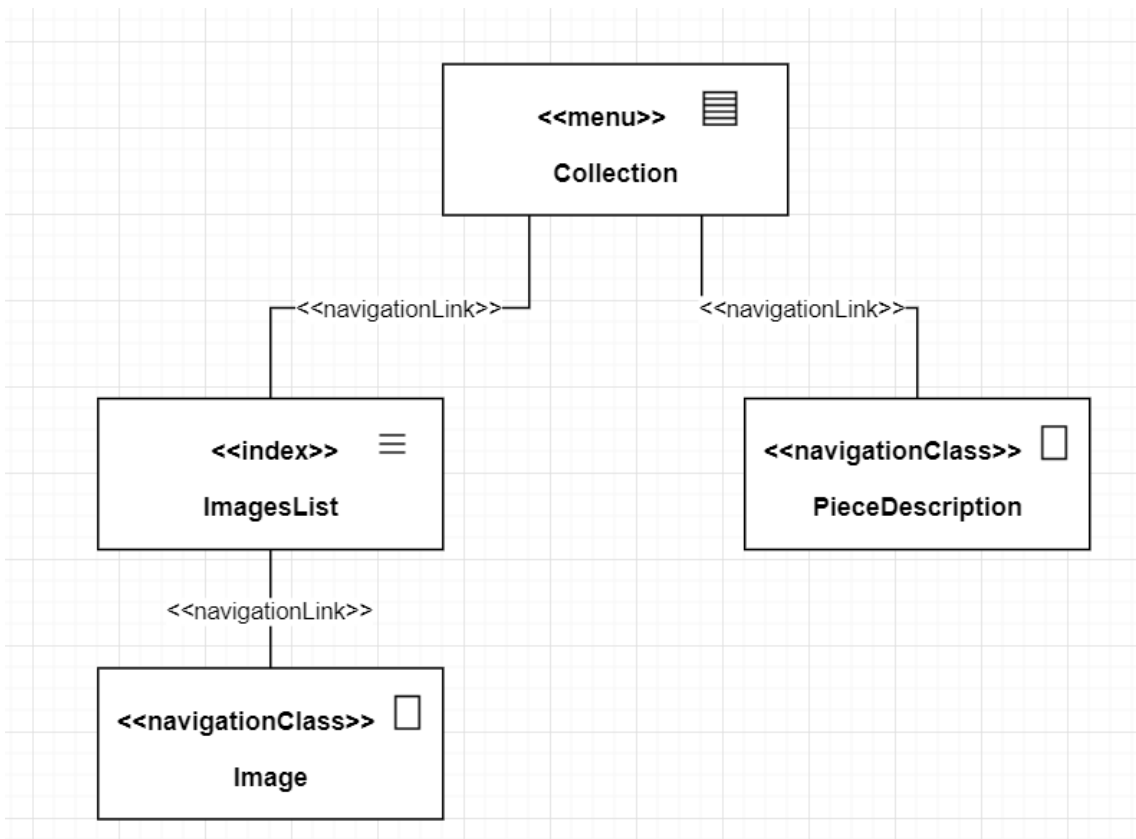
Navegación Home



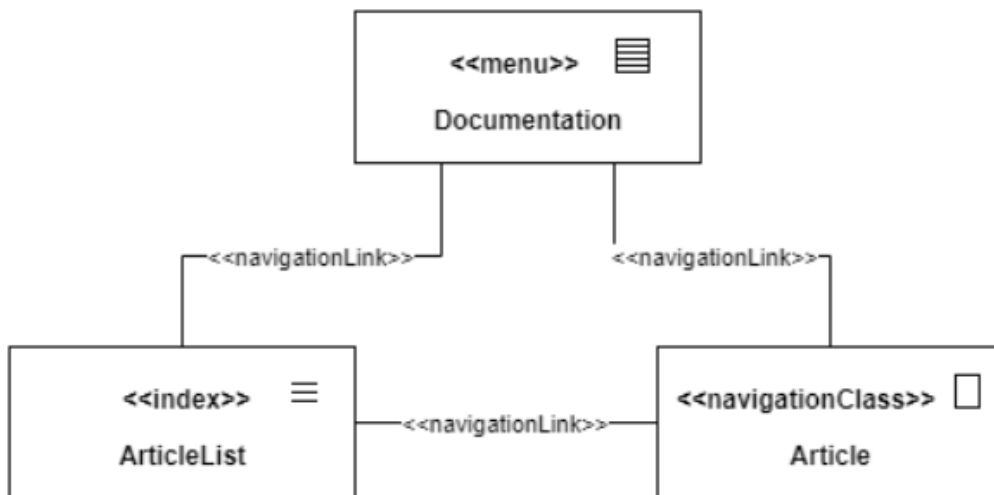
- COLECCIÓN:



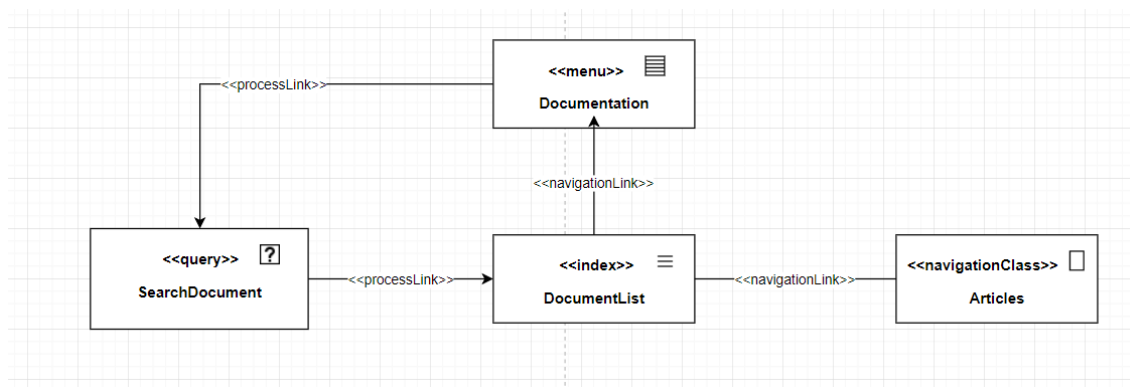
- **PIEZA:**



- **ARTÍCULO:**



- **DOCUMENTACIÓN:**



6. CUESTIONES ABIERTAS

Debido al reducido tiempo de vida de este proyecto y a las complicaciones que pueden surgir durante el desarrollo que provocaría incumplir los tiempos estimado, pueden quedar requisitos de baja prioridad y de la última iteración a alcanzar. Esta acción no tendría consecuencias sobre la aplicación web ya que estos requisitos no tienen dependencias ni son fundamentales para el correcto funcionamiento de la web, limitándose a añadir funcionalidad extra de cara al usuario que visita virtualmente el Museo de la Informática.

Apéndice D

Diseños Técnicos -

Iteración 1

Vista Colección

En esta ventana se muestran las piezas que hay en la base de datos que coinciden con los criterios de búsqueda seleccionados por el usuario. La ventana consta de un buscador en el que el usuario puede filtrar las piezas por la marca, por un rango de antigüedad y finalmente seleccionar los tipos de piezas que desea buscar. En la parte derecha de la ventana el usuario observará las piezas que cumplan con esos criterios, mostrando el número total de resultados, los resultados se cargaran en lotes. Pulsando en cualquiera de las piezas, el usuario accede a la pestaña de la pieza en cuestión.

Requisitos alcanzados:

Esta ventana alcanzaba el requisito:

- **RF 1.3- Visualizar pieza:** El sistema debe permitir a un usuario visualizar el contenido visible de una pieza.

Modelo y repositorio:

- Piece.java – PieceRepository.java
- Brand.java– BrandRepository.java
- Type.java– TypeRepository.java

Servicios:

- Al cargar la página se llamarán a las operaciones:
 4. getNPieces(int n) – Obtiene las primeras n piezas.

5. getBrands() - Obtiene las marcas de piezas disponibles.
 6. getTypes() - Obtiene los tipos de piezas disponibles.
- Al buscar las piezas:
 - findByYearBetweenAndTypesContainsAndBrandLike(String minyear,String maxyear, String[] types, String brandname) Busca coincidencias dentro de la base de datos, respecto a los criterios de entrada.
 - Otros servicios:
 4. findByTypesContains(String[] types) - Obtiene las piezas que contienen los tipos pasados por parámetro.
 5. FindByYearBetween(String minyear, String maxyear) - Obtiene las piezas que están entre el rango de años pasados por parámetros.
 6. FindByBrand(String brandname)- Obtiene las piezas cuya marca sea la que se pasa por parámetro.

Boceto

Colección

MUSEO INFORMÁTICA LUNES A VIERNES de 10:00h a 19:00h Administración | ES | [Social Media Icons]

INICIO | DOCUMENTACIÓN | **COLECCIÓN** | DONACIONES | ACERCA DE

Coincidencias: 48 Resultados

Criterio de búsqueda:

Seleccione la marca:

Canon | IBM | MSI | Canon | HP | Casio

Seleccione la antigüedad de la pieza:

1970 1980 1990 2000 2010 2020

Seleccione los tipos:

Tipos

- Calculadora
- Ordenador
- Regla
- Eléctrica
- Bolsillo
- Programable
- Consola Juegos

Los valores actualmente seleccionados: Calculadora,Regla,Bolsillo

BUSCAR

Canola 1614P
Esta es una calculadora programable, un modelo posterior a la 167P y totalmente compatible. también presente en este museo. Al igual que la 167P, dispone de un lector de tarjetas perforadas para la lectura v almacenamiento de datos....

DM-562
Agenda electrónica de bolsillo.
Tiene calculadora básica, reloj, calendario, listín telefónico, agenda para planificar tareas y alarmas programas.

LS-56H
Calculadora de bolsillo con pantalla de cuarzo líquido con dígitos enormes.
Tiene un teclado generoso que la hace muy cómoda de utilizar programas.

Palmtronic LC-8M
Calculadora de bolsillo con pantalla de cuarzo líquido.
Tiene, además de las cuatro operaciones básicas, %, raíz cuadrada y memoria.
Chip Toshiba T3709.

HORARIO: Lunes a Viernes de 10h a 19h | Universidad de Málaga - Avda. Cervantes, 2. 29071 MÁLAGA - Tel. 952 13 10 00 - info@uma.es | Inicio Documentación Colección Donaciones Acerca de | [Social Media Icons]

Copyright ©2019 by UMA. All rights reserved.

Boceto de la vista Colección

Nombre (id)	Componente	Obligatorio	Validación	Valor por defecto	Notas
brandsField	selectOneListBox	No	No		Servicio: getBrands()
yearsField	slider-container	No	No	1970-2020	
typesField	selectManyCheckbox	No	No	Todas deseleccionadas	Servicio: getTypes()
results	label	No	No	Número de piezas en la base de datos	
coincidencias	label	No	No		
searchCriteria	label	No	No		
table	table	No	No	Si no hay piezas: "No hay piezas que coincidan con los criterios"	Tabla con https://material.angular.io/components/table/overview
buttonSearch	button	No	No		Servicio: getPiecesByBrandYearType

Vista Piezas

En esta ventana se muestra la pieza que el usuario eligió en alguna de las ventanas de la web. El id de la pieza se pasará como parámetro en la *URL* de la página. Se visualizará el título o nombre, la descripción, el año, la fecha de adquisición, la marca, el nombre de la sala donde está expuesta, las visitas, los tipos en los que está incluida y las imágenes de esta. En la parte izquierda, se sitúa una galería de imágenes donde se podrá seleccionar cualquiera de ellas para verlas a mayor tamaño. A la derecha disponemos de toda la información referente mencionada anteriormente, menos la descripción que se encuentra en la parte inferior separada.

Requisitos alcanzados:

- RF 1.3- Visualizar pieza: El sistema debe permitir a un usuario visualizar el contenido visible de una pieza.

Modelo y repositorio:

- Piece.java– PieceRepository.java
- Brand.java– BrandRepository.java
- Type.java– TypeRepository.java

Servicios:

Al cargar la vista:

- getPieceById(String id) - Busca la pieza por su id (id pasado por parámetro en la url).
- increaseVisits()- Incrementa el número de visitas en la unidad

Boceto:

Colección > Piezas

MUSEO INFORMÁTICA LUNES A VIERNES de 10:00h a 19:00h Administración | ES ▼ |

INICIO | DOCUMENTACIÓN | **COLECCIÓN** | DONACIONES | ACERCA DE

Canola 1614P



Fabricante: Canon

Modelo: Canola 1614P

Año: 1973

Visitas: 21

Tipos: Calculadora, Eléctrica. Portátil.

2011/34/1 21/12/2011

Esta es una calculadora programable, un modelo posterior a la 167P y totalmente compatible, también presente en este museo. Al igual que la 167P, dispone de un lector de tarjetas perforadas para la lectura y almacenamiento de datos y programas. La mayor diferencia entre ambos modelos, es su mayor capacidad de almacenamiento y la existencia de un puerto de expansión en la parte posterior.

Es una máquina que dispone de una pantalla de tipo "nixie" de 16 dígitos. Dispone de 14 registros de memoria, y puede almacenar programas, creo de hasta 240 pasos. Vista posterior de la calculadora. Se aprecia la ranura del lector de tarjetas perforadas y el puerto de expansión.

Para más información consulta la Canola 167P.

Interior: Parte posterior, en la que se puede ver el lector de tarjetas y la fuente de alimentación.

HORARIO: Lunes a Viernes de 10h a 19h | Universidad de Málaga - Avda. Cervantes, 2. 29071 MÁLAGA - Tel. 952 13 10 00 - info@uma.es | [Inicio](#) [Documentación](#) [Colección](#) [Donaciones](#) [Acerca de](#) |

Copyright ©2019 by UMA. All rights reserved.

Boceto de la vista Piezas

Nombre (id)	Componente	Obligatorio	Validación	Valor por defecto	Notas
pieceName	label	No	No	piece.name	
pieceModel	label	No	No	piece.name	
pieceBrand	label	No	No	piece.brand	
pieceYear	label	No	No	piece.year	
pieceTypes	label	No	No	piece.types	
pieceDescription	textarea	No	readonly	piece.description	
visits	label	No	No	0	
imagesList	table	No	No	piece.images	Al clicar una imagen se podrá visualizar a mayor tamaño

Apéndice E

Pruebas de sistema

- Iteración 1

Contenidos

1. *Contenidos*
2. *Control de versión*
 - 2.1. Control de cambios
3. *Pruebas de sistema*
 - 3.1. TC001
 - 3.1.1. Descripción
 - 3.1.2. Resultado
 - 3.1.3. Evidencias
 - 3.1.4. Base de datos
 - 3.1.5. *Ficheros*
 - 3.2. TC002
 - 3.2.1. *Descripción*
 - 3.2.2. *Resultado*
 - 3.2.3. *Evidencias*
 - 3.2.4. *Base de datos*
 - 3.2.5. *Ficheros*
 - 3.3. TC003
 - 3.3.1. *Descripción*
 - 3.3.2. *Resultado*
 - 3.3.3. *Evidencias*
 - 3.3.4. *Base de datos*
 - 3.3.5. *Ficheros*

2. Control de versión

2.1. Control de cambios

Fecha	Autor	Versión	Comentario
04/05/2019	Jorge Luis Palomares Ruiz	1.0	Versión inicial, estructura de cada escenario

3. Pruebas de sistema

Estas son las pruebas del sistema para las ventanas Colección y Pieza que hacen referencia con el diseño técnico Colección - IT1.

Para este test, no hay ningún usuario con sesión iniciada, por lo que se accederá a la página como un usuario visitante.

Se comprobará todos los posibles escenarios, tanto válidos como escenarios de error, mostrando una breve descripción del caso de prueba, su resultado ('Passed'/'Failed'), evidencias de los pasos seguidos con sus respectivas capturas de pantalla, capturas de la base de datos en caso de que la prueba de sistema involucre alguna operación de modificación en la base de datos y ficheros generados o usados si se da el caso.

3.1. TC001

3.1.1. Descripción

Pruebas de sistema para la ventana Colección.

Navegar hasta la ventana Colección, haciendo clicando en el menú de navegación en "Colección" donde se cargaran todas las piezas al inicio.

3.1.2. Resultado

Passed.

3.1.3. Evidencias

Entramos en la página principal.



HP

HP Inc., es conocida como HP; es una empresa estadounidense con sede en Palo Alto, California, surgida de la separación de Hewlett-Packard en dos empresas en noviembre de 2015. Es la empresa líder en venta de impresoras en el mundo y la segunda a nivel mundial en venta de computadores personales y portátiles, después que en julio de 2013 fuese superada por Lenovo.



Apple

Apple, Inc. es una empresa estadounidense que diseña y produce equipos electrónicos, software y servicios en línea. Tiene su sede central en el Apple Park, en Cupertino (California, Estados Unidos) y la sede europea en la ciudad de Cork (Irlanda).



IBM

International Business Machines Corporation (IBM) es una reconocida empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York. IBM fabrica y comercializa hardware y software para computadoras, y ofrece servicios de infraestructura, alojamiento de Internet, y consultoría en una amplia gama de áreas relacionadas con la informática, desde computadoras centrales hasta nanotecnología.

Seleccionamos la pestaña Colección en la barra de navegación..

Colección de Piezas

¿Buscas una pieza?
Tipos de Piezas

- Juegos
- Consola de Juegos
- Electrónico
- Ordenador
- Programador
- Calculadora

Año de la pieza

Año mínimo: [1950] Año máximo: [1990]

1960 1980 2019

Marca de la pieza

Valores Seleccionados:

Tipos: No ha elegido ningún tipo

Rango de Tiempo: [1950, 1990]

Marca: No ha elegido ninguna marca

Buscar

Coincidencias: 4

18C Business Consultant
Calculadora de bolsillo con funciones financieras. Tiene pantalla de cristal líquido monocolor. Tiene:
Programación/Calculadora de bolsillo

Portable PC (5155)
Primera ordenadora portátil, el primer de IBM (lo usó costando el \$100, de 1975 y que pesaba 25kg.).
Electrónico/Ordenador

Europus
Con unidad de disco, dos manuales, tres logros de expansión y dos papeleta.
Ordenador

HP-12C
Calculadora de bolsillo con funciones financieras y programada. Probablemente una de las más populares más.
Calculadora/Calculadora/Programador

3.1.4. Base de datos

N/A

3.1.5. Ficheros

N/A

3.2. TC002

3.2.1. Descripción

Pruebas de sistema para la ventana Colección.

Realizar una búsqueda usando los filtros proporcionados de tipos, año y marca. Se podrán ver las piezas que cumplan esos criterios, junto con el número de piezas encontradas para estos.

3.2.2. Resultado

Passed.

3.2.3. Evidencias

1. Seleccionamos la pestaña Colección en la barra de navegación.

The screenshot shows the 'Colección de Piezas' website interface. At the top, there is a navigation bar with links for 'Inicio', 'Documentación', 'Colección', 'Donaciones', and 'Ayuda de'. Below the navigation bar, the page title 'Colección de Piezas' is displayed. The main content area is divided into two sections. On the left, there is a search filter panel titled '¿Buscas una pieza?' with the following options: 'Tipos de Piezas' (with radio buttons for 'Calculadora', 'Consola de Juegos', 'Disquete', 'Ordenador', 'Programable', and 'Calculadora'), 'Año de la pieza' (with input fields for 'Año mínimo' and 'Año máximo'), and 'Marca de la pieza' (with a dropdown menu). Below these filters, there is a 'Buscar' button. On the right, there is a search results section titled 'Coincidencias: 4'. It displays four items in a grid: 1. '18C Business Consultant' (Calculadora de bolsillo con funciones financieras), 2. 'Portable PC (5136)' (Primera computadora portátil, empresa de IBM), 3. 'Europlus' (Con unidad de disco, dos manuales, tres tarjetas de expansión y diez periféricos), and 4. 'HP-12C' (Calculadora de bolsillo con funciones financieras y programable). Each item includes a small image and a brief description.

2. Seleccionamos tipos, rango de años y marcas cualesquiera.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f o t @ Administración

Coleccion de Piezas

¿Buscas una pieza?

Tipos de Piezas

- Boletillo
- Consola de Juegos
- Eléctrico
- Ordenador
- Programable
- Calculadora

Año de la pieza

Año mínimo: [1925]

Año máximo: [1995]

1900 1925 1995 2019

Marca de la pieza

[HP]


Valores Seleccionados:

Tipos: Boletillo

Rango de Tiempo: [1925,1995]

Marca: HP


Coincidencias: 4



18C Business Consultant

Calculadora de bolsillo con funciones financieras. Tiene pantalla de cuarzo líquido monocolor. Tiene...


Programable;Calculadora;Bolsillo



Portable PC (5155)

Primeros ordenador portátil, el primero de IBM (si no contamos el 5100, de 1975 y que pesaba 22kg)...


Eléctrico;Ordenador



II Europlus

Con unidad de disco, dos manuales, tres tarjetas de expansión y dos joystick.

Ordenador



HP-12C

Calculadora de bolsillo con funciones financieras y programable. Probablemente una de las máquinas más...

Calculadora;Bolsillo;Programable

Inicio Documentación Colección Donaciones Acerca de

HORARIO: Lunes a Viernes de 10h a 19h
 Universidad de Málaga - Avda. Cervantes, 2. 29071 MÁLAGA - Tel. 952 13 10 00 - info@uma.es
 Copyright ©2019 by UMA. All rights reserved.

3. Seleccionamos el botón Buscar.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f o t @ Administración

Coleccion de Piezas

¿Buscas una pieza?

Tipos de Piezas

- Boletillo
- Consola de Juegos
- Eléctrico
- Ordenador
- Programable
- Calculadora

Año de la pieza

Año mínimo: [1925]

Año máximo: [1995]

1900 1925 1995 2019

Marca de la pieza

[HP]


Valores Seleccionados:

Tipos: Boletillo

Rango de Tiempo: [1925,1995]

Marca: HP


Coincidencias: 2



18C Business Consultant

Calculadora de bolsillo con funciones financieras. Tiene pantalla de cuarzo líquido monocolor. Tiene...

Programable;Calculadora;Bolsillo



HP-12C

Calculadora de bolsillo con funciones financieras y programable. Probablemente una de las máquinas más...

Calculadora;Bolsillo;Programable

Inicio Documentación Colección Donaciones Acerca de

HORARIO: Lunes a Viernes de 10h a 19h
 Universidad de Málaga - Avda. Cervantes, 2. 29071 MÁLAGA - Tel. 952 13 10 00 - info@uma.es
 Copyright ©2019 by UMA. All rights reserved.

3.2.4. Base de datos

N/A

3.2.5. Ficheros

N/A

3.3. TC003

3.3.1. Descripción

Pruebas de sistema para la ventana Colección.

Realizar una búsqueda con el fin de no encontrar datos, apreciar cómo la información mostrada en pantalla se actualiza al igual que el número de resultados encontrados.

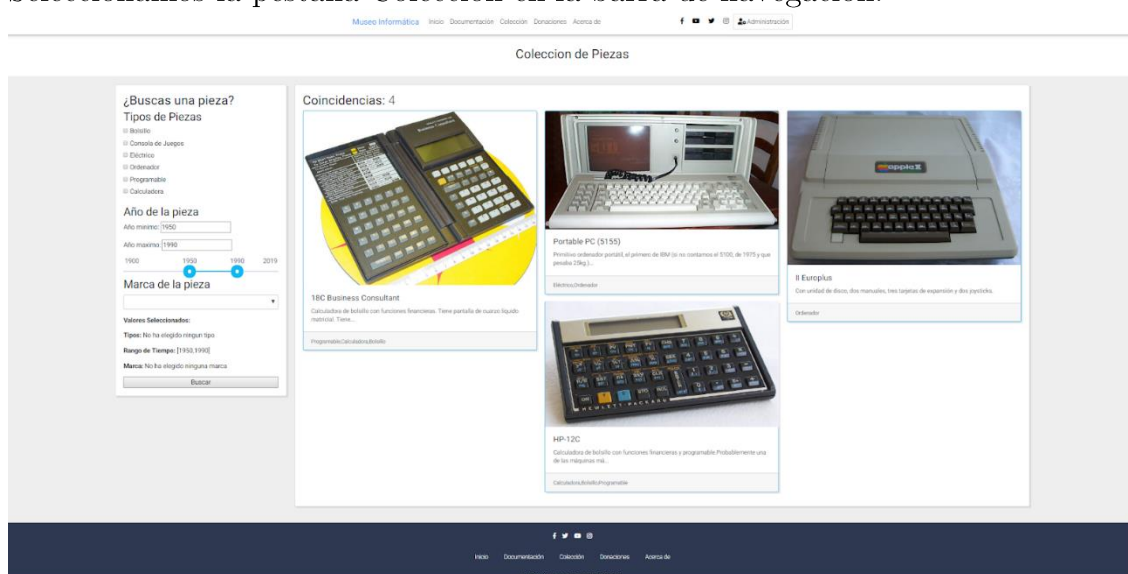
Un mensaje informando de que no hay piezas aparece.

3.3.2. Resultado

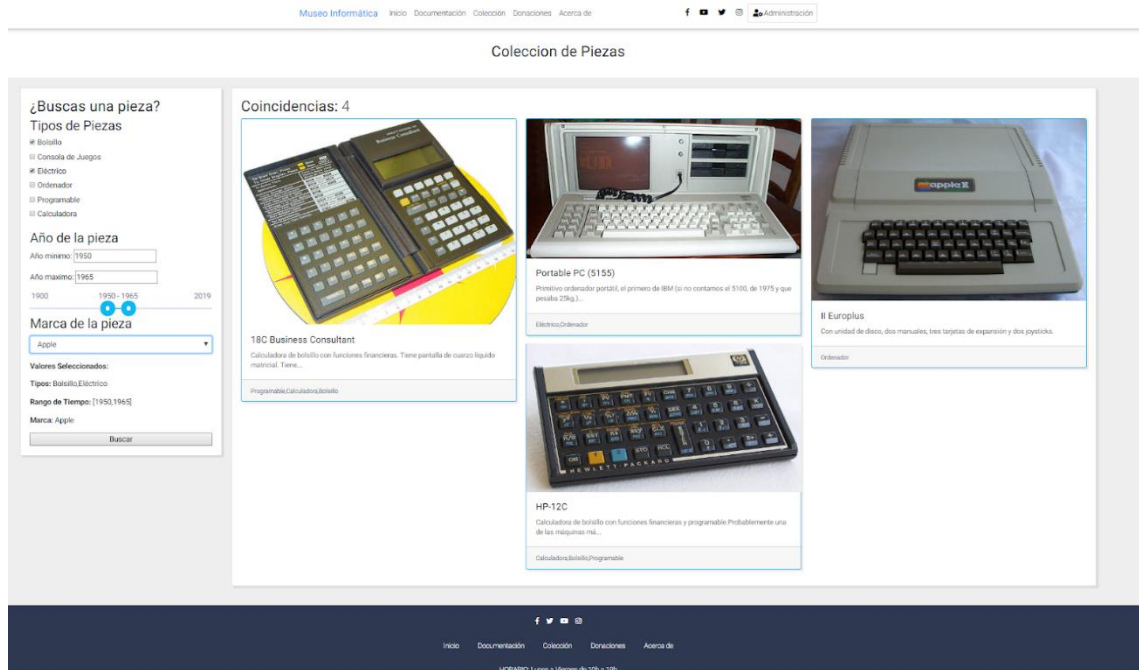
Passed.

3.3.3. Evidencias

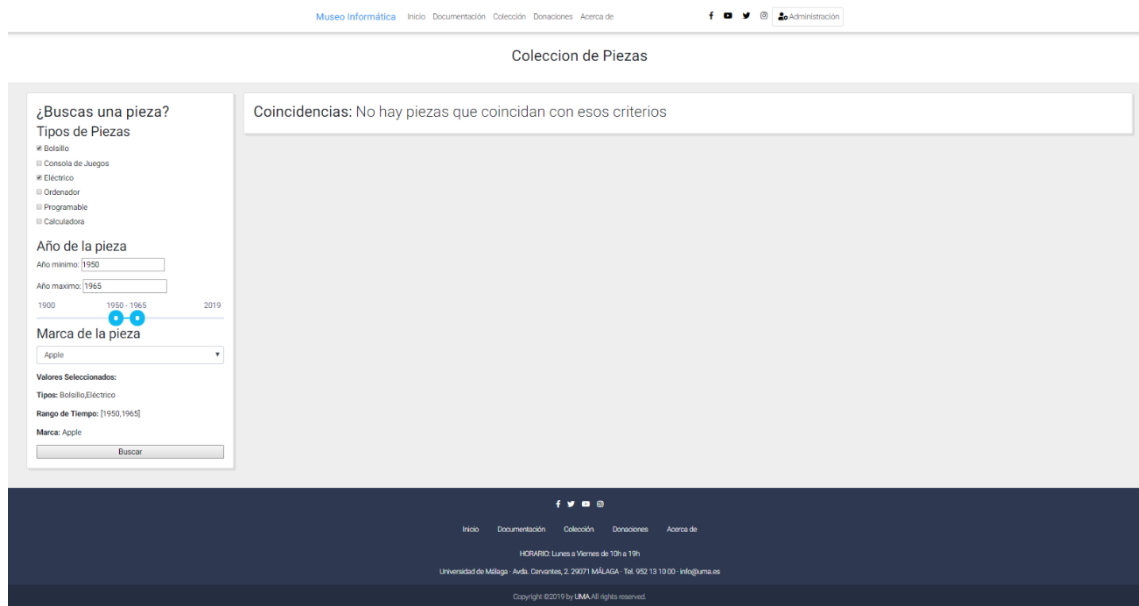
Seleccionamos la pestaña Colección en la barra de navegación.



2. Seleccionamos tipos, rango de años y marcas cualesquiera.



3. Seleccionamos el botón Buscar.



3.3.4. Base de datos

N/A

3.3.5. Ficheros

N/A

3.4. TC004

3.4.1. Descripción

Pruebas de sistema para la ventana Colección y Pieza.

Seleccionar un pieza y acceder a la la página de descripción de la pieza.

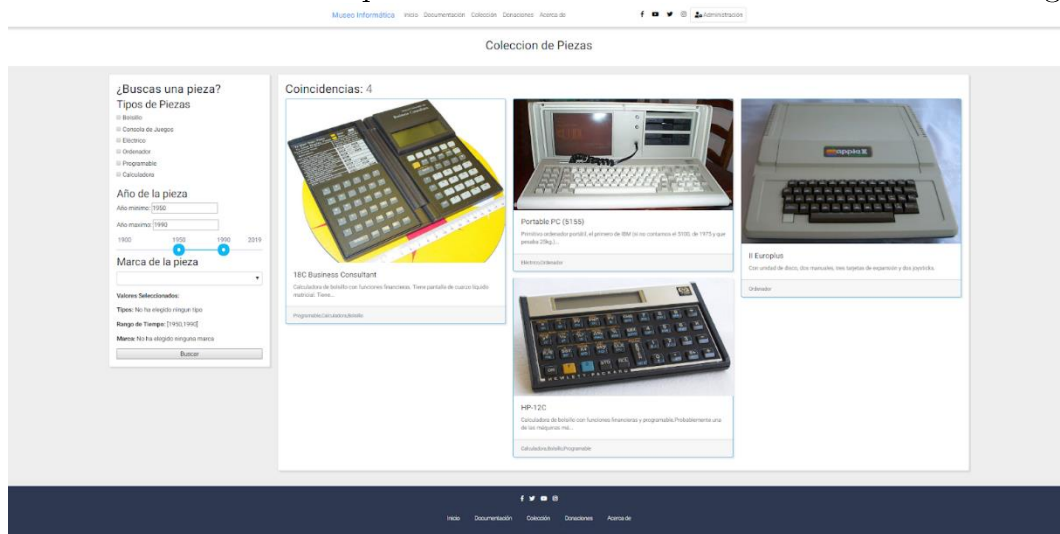
Seleccionar una foto de la galería y navegar entre ellas.

3.4.2. Resultado

Passed.

3.4.3. Evidencias

1. Seleccionamos la pestaña Colección en la barra de navegación.



2. Seleccionamos una pieza.





3. Navegamos clicando en las flechas en la imagen “>”.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f yt ig Administración

18C Business Consultant

Galeria

UNIVERSIDAD DE MÁLAGA

Datos de la Pieza:
Fabricante: HP
Modelo: 18C Business Consultant
Año: 1986
Visitas: 0
Tipos: Programable,Calculadora,Bolsillo
Fecha de Creacion: 2019-04-25
Sala: 3.0.1

Descripción de la pieza:
 Calculadora de bolsillo con funciones financieras. Tiene pantalla de cuarzo líquido matricial. Tiene un diseño peculiar, como un libro que se abre. Se trata de la primera calculadora de bolsillo con puerto infrarrojo para transmitir la información a la impresora. Esta máquina no utiliza la notación polaca inversa (RPN) que era característico en máquinas de la marca, utiliza el sistema algebraico. No se puede decir estrictamente que sea programable, pero si permite introducir fórmulas complejas y asignarlas a las seis teclas que se encuentran debajo de la pantalla.
 Pieza donada por: Samuel Del Pozo

4. Podemos navegar también clicando directamente en las imágenes inferiores

Museo Informática Inicio Documentación Colección Donaciones Acerca de f yt ig Administración

18C Business Consultant

Galeria




UNIVERSIDAD DE MÁLAGA

Datos de la Pieza:
Fabricante: HP
Modelo: 18C Business Consultant
Año: 1986
Visitas: 0
Tipos: Programable,Calculadora,Bolsillo
Fecha de Creacion: 2019-04-25
Sala: 3.0.1

Descripción de la pieza:
 Calculadora de bolsillo con funciones financieras. Tiene pantalla de cuarzo líquido matricial. Tiene un diseño peculiar, como un libro que se abre. Se trata de la primera calculadora de bolsillo con puerto infrarrojo para transmitir la información a la impresora. Esta máquina no utiliza la notación polaca inversa (RPN) que era característico en máquinas de la marca, utiliza el sistema algebraico. No se puede decir estrictamente que sea programable, pero si permite introducir fórmulas complejas y asignarlas a las seis teclas que se encuentran debajo de la pantalla.
 Pieza donada por: Samuel Del Pozo

5. Volvemos a la pestaña documentación clicando en el hipervínculo “Colección”.

Colección de Piezas

¿Buscas una pieza?

Tipos de Piezas

- Básico
- Computo de Jueg
- Electro
- Ordenador
- Programable
- Calculadora

Año de la pieza

Año mínimo: 1950

Año máximo: 1999

1950 1999 2019

Marca de la pieza

Valores Seleccionados:

Tipos: No ha elegido ningún tipo

Rango de Tiempo: [1950-1999]

Marca: No ha elegido ninguna marca

Buscar

Coincidencias: 4



18C Business Consultant

Calculadora de bolsillo con funciones financieras. Tiene pantalla de cristal líquido

horizontal. Tiene...

Programable/Calculadora/Básico



Portable PC (5155)

Personal de tamaño portátil, el primer de IBM (si no contamos al 5100, de 1975 y que

pesaba 15kg).

Electro/Ordenador



II Europius

Con unidad de disco, dos manuales, tres tarjetas de expansión y dos periféricos.

Ordenador



HP-12C

Calculadora de bolsillo con funciones financieras y programable. Probablemente una

de las siguientes...

Calculadora/Básico/Programable

3.4.4. Base de datos

N/A

3.4.5. Ficheros

N/A

Apéndice F

Diseños Técnicos - Iteración 2

Los **requisitos alcanzados** con estas pantallas fueron:

- **RF 1-Crear pieza:** La pieza debe tener los atributos nombre, texto, año, fecha de creación, lista de imágenes, código QR, nombre de la sala, lista de tipos y nombre de la marca a la que pertenece.
- **RF 1.1- Editar pieza:** El sistema permite al administrador la edición de piezas que ya habían sido creadas anteriormente.
- **RF 1.2 - Eliminar pieza:** El sistema debe permitir al administrado eliminar una pieza que ya había sido creada anteriormente.
- **RF 4 - Crear marca:** Si se intenta añadir una pieza con una marca que no está creada, el sistema deberá permitir crear una nueva.
- **RF 4.1 - Eliminar marca:** El sistema debe permitir al administrador eliminar una marca que ya había sido creada anteriormente.
- **RF 5 - Crear tipo:** Si se intenta añadir una pieza con un tipo que no está creado, el sistema deberá permitir crear un nuevo tipo aportando su nombre.
- **RF 5.1 - Eliminar tipo:** El sistema debe permitir al administrador eliminar un tipo que ya había sido creado anteriormente.
- **RF 6 - Crear sala:** La sala debe tener los atributos nombre, localización y texto.
- **RF 6.1- Editar sala:** El sistema permite al administrador la edición de salas que ya habían sido creadas anteriormente.

Vista CRUD Pieza

En esta ventana se muestra una lista con todas las piezas registradas en la base de datos, mostrando su nombre. Desde la lista podemos borrar o editar las piezas pulsando en los botones dispuestos, la ventana también presenta un botón en la esquina superior izquierda para poder crear una nueva pieza(Vista Crear Pieza). Esta vista está dentro de un "panel tab" donde se encuentran los demás vistas CRUD de Documentos, Tipos, Marcas y Salas.

Modelo y repositorio:

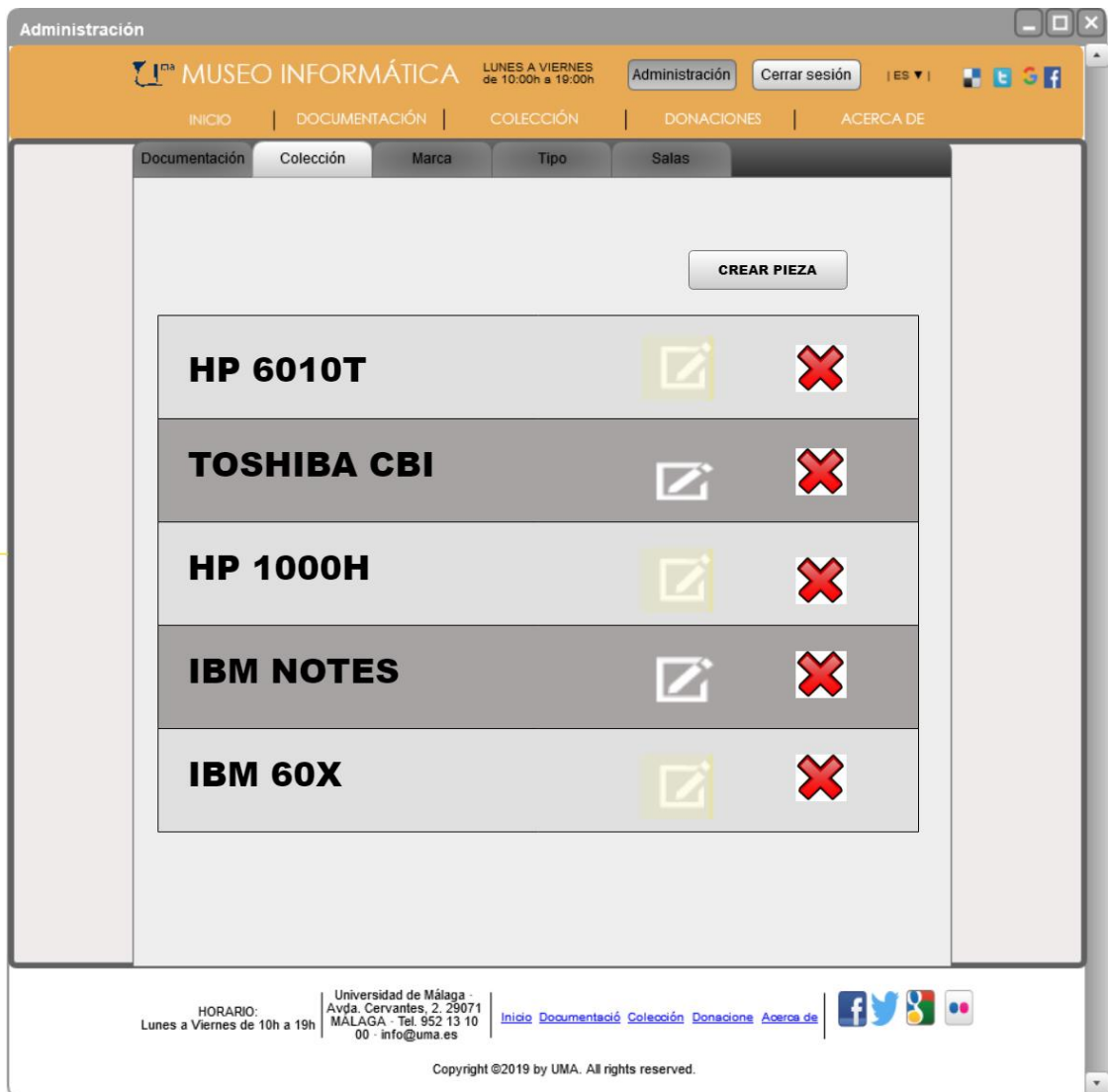
Piece.java– PieceRepository.java

Servicios:

findAll() - Devuelve una lista con todas las piezas de la base de datos.

Boceto

Se creó un estilo común para las vistas del CRUD para agilizar el trabajo de desarrollo y darle cohesión a la aplicación. Se dispone de unos botones los cuales permiten cumplir los requisitos mencionados anteriormente de crear editar o eliminar una pieza.



Vista CRUD Pieza

Nombre (id)	Componente	Obligatorio	Validación	Valor por defecto	Notas
piecesTable	table	No	No	pieces	Consta de un botón editar y otro eliminar en cada una de sus filas
createButton	button	No	Si	“Crear pieza”	Redirige a la siguiente vista

Vista Crear Pieza

En esta ventana se muestra un formulario con los atributos necesarios para añadir una nueva pieza en nuestra base de datos. La ventana consta de cuadros de texto y seleccionables para introducir los datos pertinentes en una columna izquierda.

En la parte derecha de la ventana el usuario observará los tipos de piezas y las imágenes introducidas, pudiendo eliminarlas de la pieza a crear de ser necesario. Además, se cuenta con unas pestañas para crear nuevas marcas, tipos y salas por si el campo que quiere introducir no existía anteriormente . Pulsando en cualquiera de los botones situados en las pestañas de “Crear Marca”, “Crear Sala” o “Crear Tipo” se creará una nueva instancia de esa clase, finalmente pulsando en “Crear Pieza” se creará una nueva pieza .

Modelo y repositorio:

- Piece.java– PieceRepository.java
- Brand.java– BrandRepository.java
- Type.java– TypeRepository.java
- Room.java– RoomRepository.java

Servicios:

- Al cargar la página se llamarán a las operaciones:
 - getRooms() – Obtiene las salas disponibles.
 - getBrands() - Obtiene las marcas de piezas disponibles.
 - getTypes() - Obtiene los tipos de piezas disponibles.
- Al crear alguna marca, sala o tipo:
 - createBrand(Brand brand) - Crea la marca con los parámetros marcados.
 - createType(Type type) - Crea el tipo con el parámetro nombre.
 - createRoom(Room room) - Crea la sala con los parámetros marcados.
- Al crear la pieza :
 - createPiece(String name, String text, String year, Date createdate, String qr, String room, String brand, String visits, String donor, String[] images, String[] types) Crea la pieza con los parámetros marcados en el formulario.

Boceto

El objetivo en esta ventana fue poder crear la pieza con todos sus datos y en caso de querer introducir una marca, tipo o sala que no estuviera ya almacenada en la base de datos poder crear ese elemento directamente desde esta página sin tener que pasar a otra. Por ese motivo se diseñó un menú de navegación a la derecha que permitía crear cualquiera de los mencionados anteriormente. Además, el caso de introducir cualquier tipo o imagen y querer corregir un error

se contempla y se soluciona con unos menús donde se muestran estos tipos e imágenes y se permiten eliminarlos.

Nombre
Descripción
Año de creación
 Sala 3.1.1
 HP
Donador
Imágenes
Tipos

Crear Marca | **Crear Tipo** | Crear Sala

Tipos introducidas

Tipo 1	✗
Tipo 2	✗
Tipo 3	✗

Imágenes introducidas

Crear Pieza

HORARIO: Lunes a Viernes de 10h a 19h | Universidad de Málaga - Avda. Cervantes, 2. 29071 MALAGA - Tel. 952 13 10 00 - info@uma.es | Inicio Documentación Colección Donaciones Acerca de

Copyright ©2019 by UMA. All rights reserved.

Vista Crear Pieza

Nombre (id)	Componente	Obligatori o	Validació n	Valor por defect o	Notas
brandsField	selectOneListbo x	No	No		Servicio: getBrands()
roomsField	selectOneListbo x	No	No		Servicio: getRooms()
typesField	inputText	No	No		
nameField	inputText	No	No		
descriptionField	inputText	No	No		
yearofCreationField	inputText	No	No		
donorField	inputText	No	No		

imagesField	InputText	No	No		
insertName	label	No	No		
insertDescription	label	No	No		
insertYearofCreatio n	label	No	No		
insertRoom	label	No	No		
insertBrand	label	No	No		
insertDonor	label	No	No		
insertImages	label	No	No		
insertTypes	label	No	No		
introducedTypes	label	No	No		
introducesImages	table	No	No		
buttonCreatePiece	button	No	No		Servicio: createPiece(Piece piece))
buttonCreateType	button	No	No		Servicio: createType(Strin g name)
createTabVar	tabVar	No	No		
listIntroducedTypes	list-ul	No	No		
listIntroducedImage s	list-ul	No	No		

Apéndice G

Pruebas de sistema - Iteración 2

Contenidos

Contenidos

2. Control de versión

2.1. Control de cambios

3. Pruebas de sistema

3.1. TC001

3.1.1. Descripción

3.1.2. Resultado

3.1.3. Evidencias

3.1.4. Base de datos

3.1.5. Ficheros

3.2. TC002

3.2.1. Descripción

3.2.2. Resultado

3.2.3. Evidencias

3.2.4. Base de datos

3.2.5. Ficheros

3.3. TC003

3.3.1. Descripción

3.3.2. Resultado

3.3.3. Evidencias

3.3.4. Base de datos

3.3.5. Ficheros

3.4. TC004

3.4.1. Descripción

3.4.2. Resultado

3.4.3. Evidencias

3.4.4. Base de datos

3.4.5. Ficheros

2. Control de versión

2.1. Control de cambios

<i>Fecha</i>	<i>Autor</i>	<i>Versión</i>	<i>Comentario</i>
<i>25/05/2019</i>	<i>Jorge Luis Palomares Ruiz</i>	<i>1.0</i>	<i>Versión inicial, estructura de cada escenario</i>

3. Pruebas de sistema

Estas son las pruebas del sistema para la ventana PiezaCRUD (zona de administración de piezas) que hacen referencia con el diseño técnico de PiezaCRUD - IT2.

Para este test, iniciaremos sesión con los credenciales de administrador para acceder a la zona de administración.

Se comprobará todos los posibles escenarios, tanto válidos como escenarios de error, mostrando una breve descripción del caso de prueba, su resultado ('Passed'/'Failed'), evidencias de los pasos seguidos con sus respectivas capturas de pantalla, capturas de la base de datos en caso de que la prueba de sistema involucre alguna operación de modificación en la base de datos y ficheros generados o usados si se da el caso.

3.1. TC001

3.1.1. Descripción

Pruebas de sistema para la ventana piezaCRUD.

Con una sesión de administrador iniciada, buscamos una pieza usando el buscador.

Posteriormente crearemos una pieza.

3.1.2. Resultado

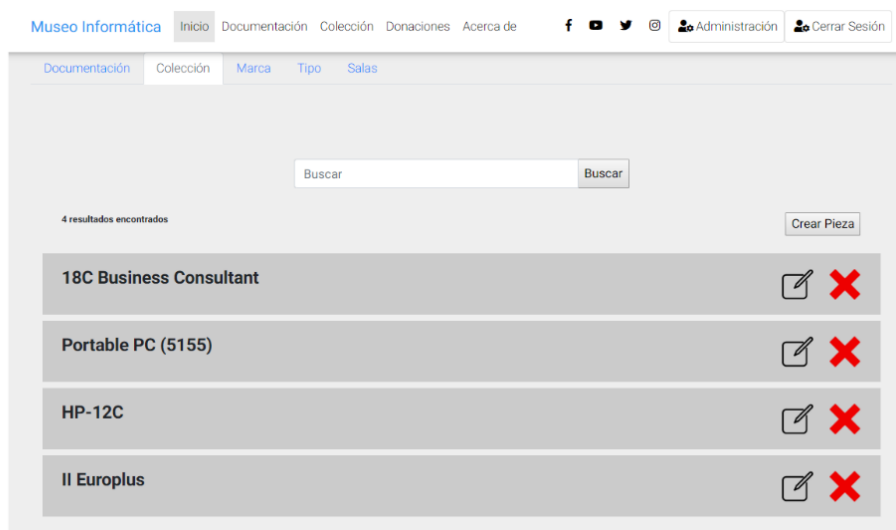
Passed.

3.1.3. Evidencias

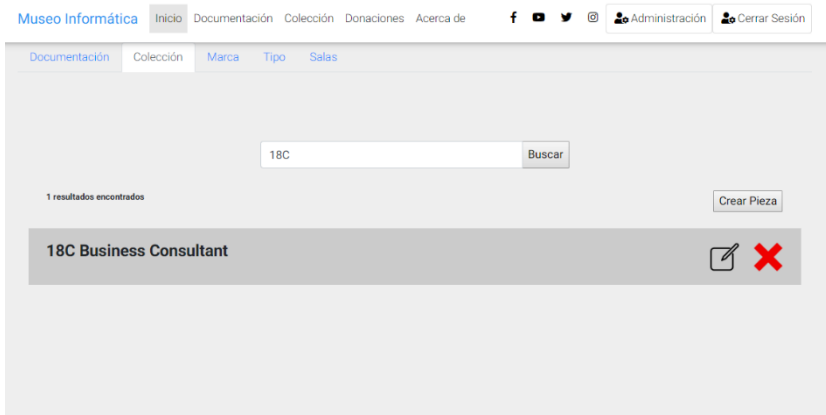
1. *Pulsamos el botón Administración.*



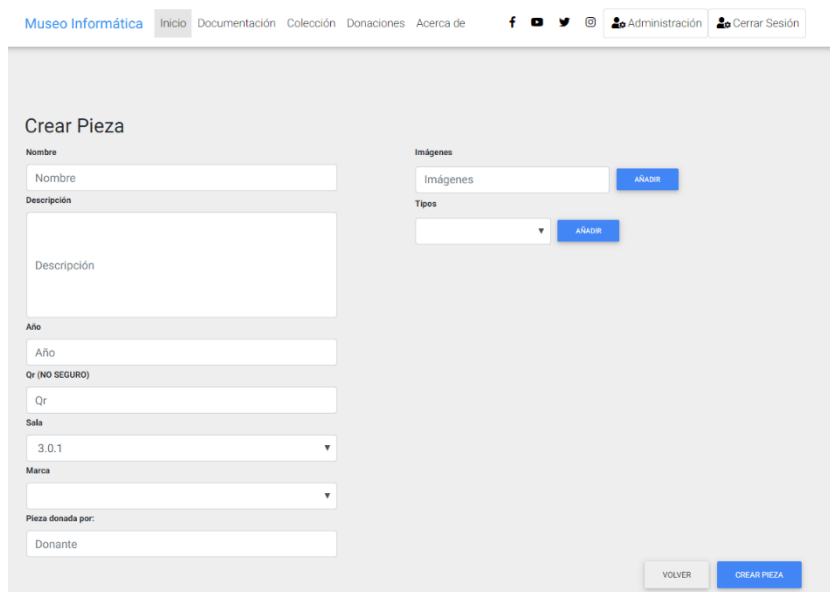
2. *Pulsamos Colección*



3. *Realizamos una búsqueda de una pieza.*



4. Pulsamos el botón *Crear Pieza*.



5. Introducimos todos los datos.



6. Pulsamos el botón Crear Pieza.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t i @ Administración Cerrar Sesión

Crear Pieza

Nombre
hp omnigo 700lx

Descripción
Ordenador de "bolsillo" compatible PC que funciona bajo el

Año
1996


Sala
3.0.1

Marca
HP

Pieza donada por:
HP

Imágenes

Imágenes



Tipos

Calculadora

- Bolsillo
- Eléctrico
- Calculadora

3.1.4. Base de datos

Antes de la creación:

```
QUERY RESULTS 1-4 OF 4

  _id: ObjectId("5cc73fd58494c4d37427d117")
  name: "IBC Business Consultant"
  text: "Calculadora de bolsillo con funciones financieras. Tiene pantalla de c..."
  year: "1986"
  createdate: 2019-04-25T00:00:00.000+00:00
  qr: "https://androidayuda.com/app/uploads/2017/10/cXC3W3digo-qr-android-ay..."
  room: "3.0.1"
  brand: "HP"
  visits: "0"
  donor: "Samuel Del Pozo"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Piece"

  _id: ObjectId("5cc740b78494c4d37427d118")
  name: "Portable PC (5155)"
  text: "Primitivo ordenador portátil, el primero de IBM (si no contamos el 510..."
  year: "1984"
  createdate: 2019-04-25T00:00:00.000+00:00
  qr: "https://androidayuda.com/app/uploads/2017/10/cXC3W3digo-qr-android-ay..."
  room: "3.0.1"
  brand: "IBM"
  visits: "0"
  donor: "Jorge Palomeras"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Piece"

  _id: ObjectId("5cc741388494c4d37427d119")
  name: "HP-12C"
  text: "Calculadora de bolsillo con funciones financieras y programable.Probab..."
  year: "1981"
  createdate: 2019-04-25T00:00:00.000+00:00
  qr: "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."
  room: "3.0.1"
  brand: "HP"
  visits: "0"
  donor: "messagu"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Piece"

  _id: ObjectId("5ccb0d3ba62f0424cbad673")
  name: "II Europlus"
  text: "Con unidad de disco, dos manuales, tres tarjetas de expansión y dos jo..."
  year: "1979"
  createdate: 2019-05-02T00:00:00.000+00:00
  qr: "https://androidayuda.com/app/uploads/2017/10/cXC3W3digo-qr-android-ay..."
  room: "3.0.1"
  brand: "APPLE"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Piece"
```

Después de la creación:

```
QUERY RESULTS 1-5 OF 5

  _id: "hp omnigo 7881x"
  name: "hp omnigo 7881x"
  text: "Ordenador de "Bolsillo" compatible PC que funciona bajo el sistema ope..."
  year: "1995"
  createdate: 2019-05-25T18:57:00.157+00:00
  qr: "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."
  room: ""
  brand: "HP"
  visits: "0"
  done: "HP"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Pieza"

  _id: ObjectId("5cc73f45849cd437427d117")
  name: "IBC Business Consultant"
  text: "Calculadora de bolsillo con funciones financieras. Tiene pantalla de c..."
  year: "1985"
  createdate: 2019-04-25T08:00:00.000+00:00
  qr: "https://androidayuda.com/app/uploads/2017/10/CLK35R3digo-qr-android-ay..."
  room: "3.8.1"
  brand: "HP"
  visits: "0"
  done: "Samuel Del Pozo"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Pieza"

  _id: ObjectId("5cc740b7640cd437427d118")
  name: "portable PC (5155)"
  text: "Pequeño ordenador portátil, el primero de IBM (si no contamos el 510..."
  year: "1984"
  createdate: 2019-04-25T08:00:00.000+00:00
  qr: "https://androidayuda.com/app/uploads/2017/10/CLK35R3digo-qr-android-ay..."
  room: "3.8.1"
  brand: "IBM"
  visits: "0"
  done: "Jorge Palomeras"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Pieza"

  _id: ObjectId("5cc74138849cd437427d119")
  name: "HP-12C"
  text: "Calculadora de bolsillo con funciones financieras y programable. Probab..."
  year: "1981"
  createdate: 2019-04-25T08:00:00.000+00:00
  qr: "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."
  room: "3.8.1"
  brand: "HP"
  visits: "0"
  done: "Tonsagu"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Pieza"

  _id: ObjectId("5ccb0d30ba2f8424cbad673")
  name: "II Europlus"
  text: "Con unidad de disco, dos manuales, tres tarjetas de expansión y dos jo..."
  year: "1979"
  createdate: 2019-05-02T00:00:00.000+00:00
  qr: "https://androidayuda.com/app/uploads/2017/10/CLK35R3digo-qr-android-ay..."
  room: "3.8.1"
  brand: "APPLE"
  > images: Array
  > types: Array
  _class: "com.museum.museumServer.main.models.Pieza"
```

3.1.5. Ficheros

N/A

3.2. TC002

3.2.1. Descripción

Pruebas de sistema para la ventana piezaCRUD.

Con una sesión de administrador iniciada, intentaremos crear una pieza sin introducir datos.

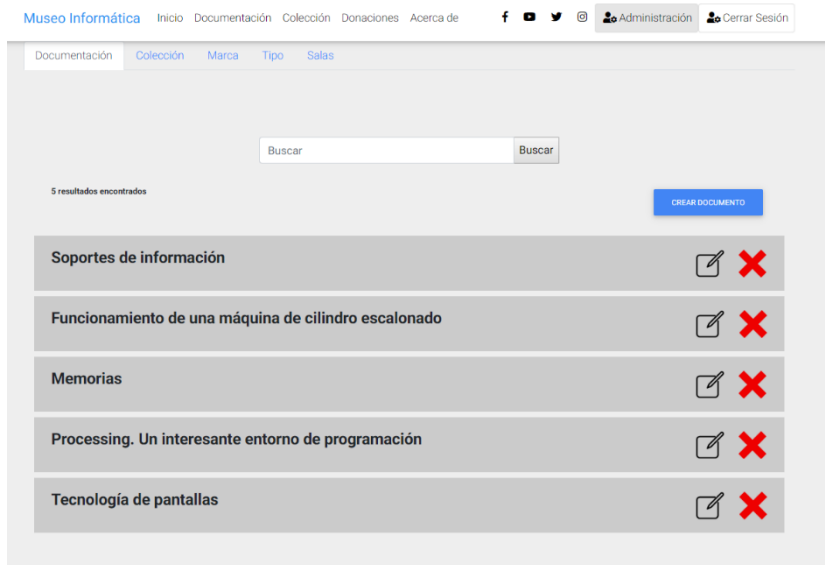
Saltarán mensajes de error y validaciones.

3.2.2. Resultado

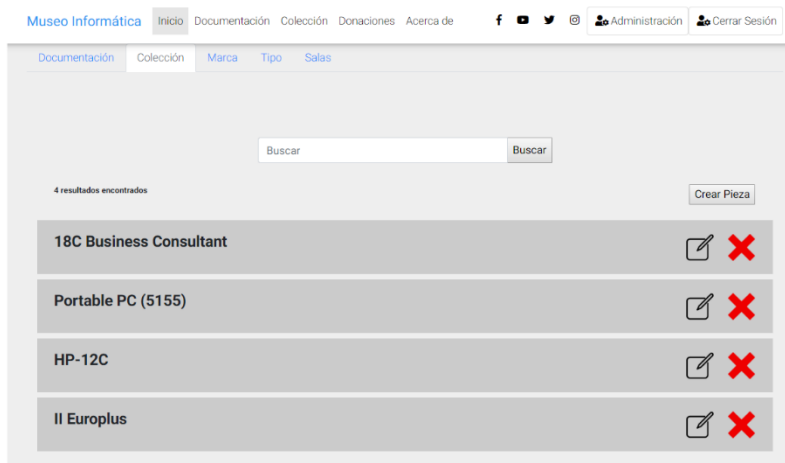
Passed.

3.2.3. Evidencias

1. Pulsamos el botón Administración.



2. Pulsamos Colección.



3. Pulsamos el botón Crear Pieza.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t i @ Administración Cerrar Sesión

Crear Pieza

Nombre

Descripción

Año

Qr (NO SEGURO)

Sala

Marca

Pieza donada por:

Imágenes
 AÑADIR

Tipos
 AÑADIR

VOLVER **CREAR PIEZA**

4. Pulsamos el botón Crear Pieza.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t i @ Administración Cerrar Sesión Introduce un nombre.

Crear Pieza

Nombre

Descripción

Año

Sala

Marca

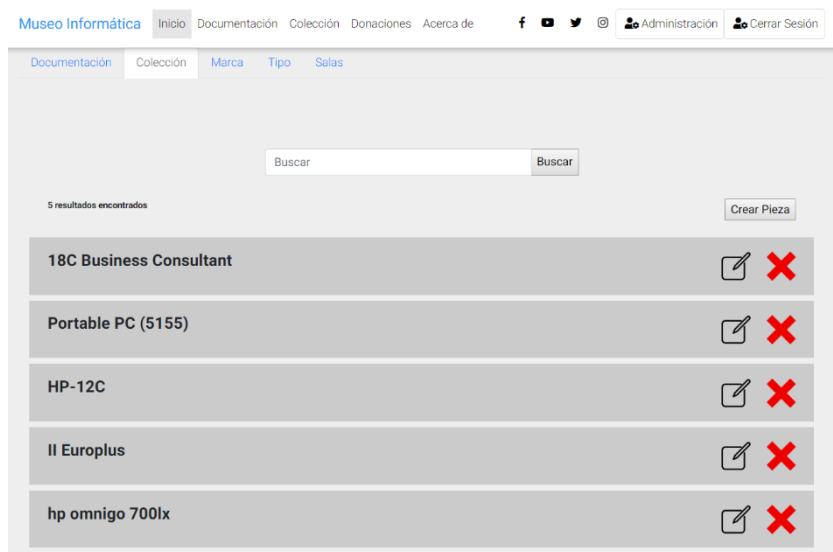
Pieza donada por:

Imágenes
 AÑADIR

Tipos
 AÑADIR

VOLVER **CREAR PIEZA**

5. Pulsamos el botón Volver.



3.2.4. Base de datos

N/A

3.2.5. Ficheros

N/A

3.3. TC003

3.3.1. Descripción

Pruebas de sistema para la ventana piezaCRUD.

Con una sesión de administrador iniciada, editaremos la pieza anteriormente creado.

3.3.2. Resultado

Passed.

3.3. Evidencias











1. Pulsamos el botón Administración.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t i o Administración Cerrar Sesión

Documentación Colección Marca Tipo Salas

Buscar Buscar

5 resultados encontrados [CREAR DOCUMENTO](#)

Soportes de información	 
Funcionamiento de una máquina de cilindro escalonado	 
Memorias	 
Processing. Un interesante entorno de programación	 
Tecnología de pantallas	 









2. Pulsamos en Colección.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t i o Administración Cerrar Sesión

Documentación Colección Marca Tipo Salas

Buscar Buscar

4 resultados encontrados [Crear Pieza](#)

18C Business Consultant	 
Portable PC (5155)	 
HP-12C	 
II Europlus	 

3. Pulsamos el icono de editar de la pieza creada anteriormente.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t w i @ Administración Cerrar Sesión

Editar Pieza

Nombre: hp omnigo 700lx

Descripción: Ordenador de "bolsillo" compatible PC que funciona bajo el


Año: 1996

Sala: 3.0.1

Marca:

Pieza donada por: HP

Imágenes: Añadir



Tipos: Añadir

- Bolsillo
- Eléctrico
- Calculadora

VOLVER EDITAR PIEZA

4. Modificamos los datos de la pieza.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t w i @ Administración Cerrar Sesión

Editar Pieza

Nombre: hp omnigo 700lx

Descripción: Ordenador de "bolsillo"


Año: 1995

Sala: 3.0.1

Marca: HP

Pieza donada por: Prueba

Imágenes: Añadir

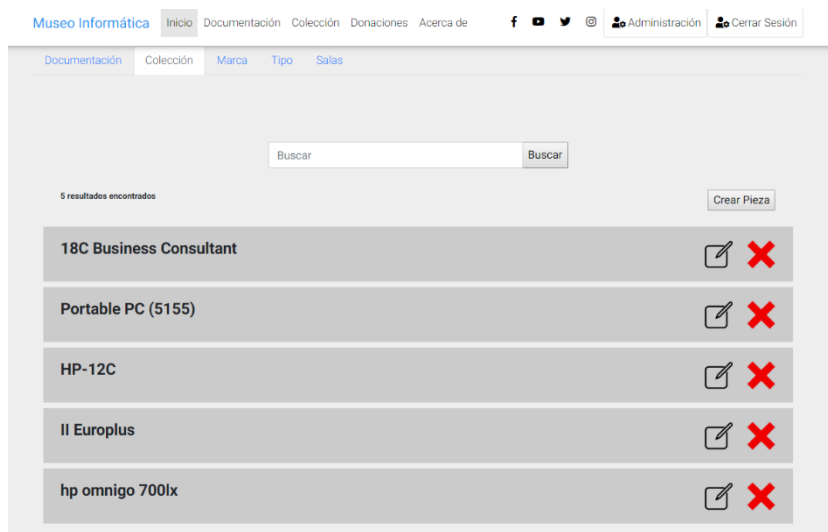


Tipos: Añadir

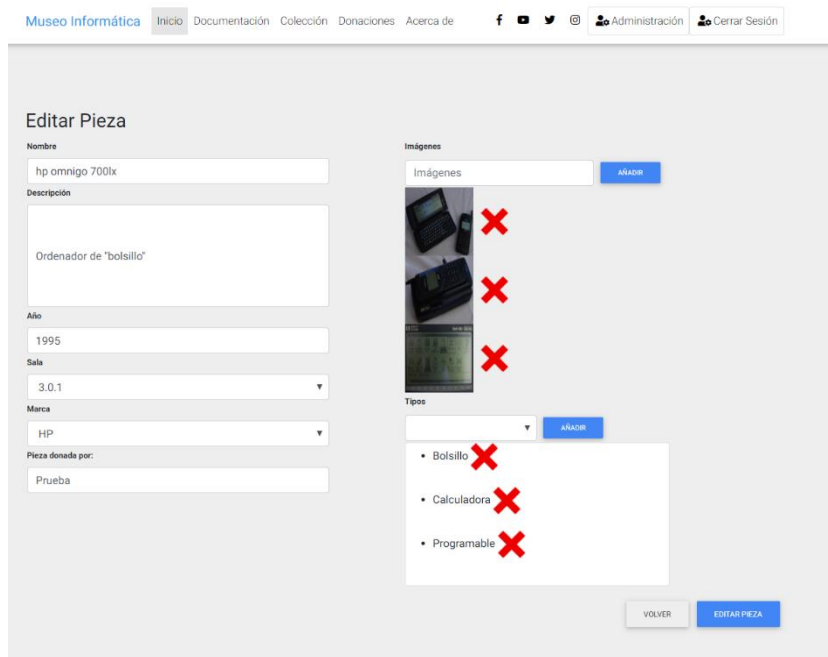
- Bolsillo
- Calculadora
- Programable

VOLVER EDITAR PIEZA

5. Pulsamos el botón editar Pieza.



6. Pulsamos editar información de la pieza para ver la información modificada.



7. Pulsamos volver.

The screenshot shows the website's search interface. At the top, there is a navigation bar with 'Museo Informática' and links for 'Inicio', 'Documentación', 'Colección', 'Donaciones', and 'Acercas de'. Social media icons for Facebook, YouTube, Twitter, and Instagram are also present, along with 'Administración' and 'Cerrar Sesión' buttons. Below the navigation bar, there are tabs for 'Documentación', 'Colección', 'Marca', 'Tipo', and 'Salas'. A search bar with the text 'Buscar' and a 'Buscar' button is centered. Below the search bar, it indicates '5 resultados encontrados' and a 'CREAR DOCUMENTO' button. The search results are displayed as a list of five items, each with a pencil icon and a red 'X' icon:

- Soportes de información
- Funcionamiento de una máquina de cilindro escalonado
- Memorias
- Processing. Un interesante entorno de programación
- Tecnología de pantallas

3.3.4. Base de datos

Antes de la modificación:

```
> {
  "_id": "hp omnigo 7001x"
  "name": "hp omnigo 7001x"
  "text": "Ordenador de "bolsillo" compatible PC que funciona bajo el sistema ope..."
  "year": "1996"
  "createdate": "2019-05-25T18:57:09.157+00:00"
  "qr": "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."
  "room": ""
  "brand": "HP"
  "visits": "0"
  "donor": "HP"
  > images: Array
  > types: Array
  "_class": "com.museum.museumServer.main.models.Piece"
}
```

Después de la modificación:

```
_id: "hp omnigo 7001x"  
name: "hp omnigo 7001x"  
text: "Ordenador de "bolsillo""  
year: "1995"  
createdate: 2019-05-25T18:57:09.157+00:00  
qr: "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."  
room: ""  
brand: "HP"  
visits: "0"  
donor: "Prueba"  
> images: Array  
> types: Array  
_class: "com.museum.museumServer.main.models.Piece"
```

3.3.5. Ficheros

N/A

3.4. TC004

3.4.1. Descripción

Pruebas de sistema para la ventana piezaCRUD.

Con una sesión de administrador iniciada, borraremos la pieza creada para estas pruebas.

3.4.2. Resultado

Passed.

3.4.3. Evidencias

1. Pulsamos el botón Administración.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t i @ Administración Cerrar Sesión

Documentación Colección Marca Tipo Salas

Buscar Buscar

5 resultados encontrados CLEAR DOCUMENTO

Soportes de información		
Funcionamiento de una máquina de cilindro escalonado		
Memorias		
Processing. Un interesante entorno de programación		
Tecnología de pantallas		

2. Pulsamos en Colección.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y t i @ Administración Cerrar Sesión

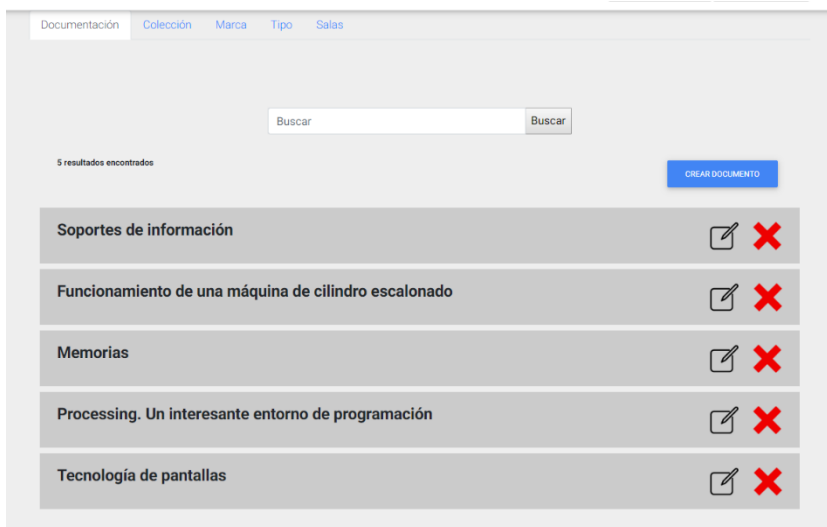
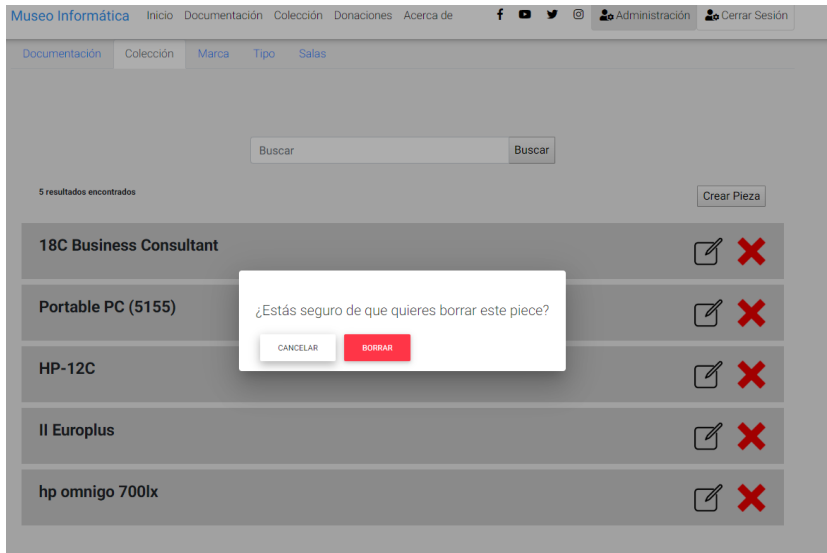
Documentación Colección Marca Tipo Salas

Buscar Buscar

4 resultados encontrados Crear Pieza

18C Business Consultant		
Portable PC (5155)		
HP-12C		
II Europlus		

3. Pulsamos el icono de borrar una pieza.



3.4.4. Base de datos

Antes del borrado:

QUERY RESULTS 1-5 OF 5

```
._id: ObjectId("5cc734d58694cd437427d117")
name: "IBC Business Consultant"
text: "Calculadora de bolsillo con funciones financieras. Tiene pantalla de c..."
year: "1980"
createdate: 2019-04-25T00:00:00.000+00:00
qr: "https://androidayuda.com/app/uploads/2017/10/cNC383dign-qr-android-ay..."
room: "3.0.1"
brand: "HP1"
visits: "0"
donor: "Samuel Del Pozo"
> images: Array
> types: Array
._class: "com.museum.museumServer.main.models.Piece"
```

```
._id: ObjectId("5cc740b78494cd437427d118")
name: "Portable PC (5155)"
text: "Primitivo ordenador portátil, el primero de IBM (si no contamos el 510..."
year: "1984"
createdate: 2019-04-25T00:00:00.000+00:00
qr: "https://androidayuda.com/app/uploads/2017/10/cNC383dign-qr-android-ay..."
room: "3.0.1"
brand: "IBM"
visits: "0"
donor: "Jorge Palomeras"
> images: Array
> types: Array
._class: "com.museum.museumServer.main.models.Piece"
```

```
._id: ObjectId("5cc741388694cd437427d119")
name: "HP-12C"
text: "Calculadora de bolsillo con funciones financieras y programable.Probab..."
year: "1981"
createdate: 2019-04-25T00:00:00.000+00:00
qr: "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."
room: "3.0.1"
brand: "HP1"
visits: "0"
donor: "Tomásagu"
> images: Array
> types: Array
._class: "com.museum.museumServer.main.models.Piece"
```

```
._id: ObjectId("5ccb0d3ba6240426cbad673")
name: "II Europlus"
text: "Con unidad de disco, dos manuales, tres tarjetas de expansión y dos jo..."
year: "1979"
createdate: 2019-05-02T00:00:00.000+00:00
qr: "https://androidayuda.com/app/uploads/2017/10/cNC383dign-qr-android-ay..."
room: "3.0.1"
brand: "APPLE"
> images: Array
> types: Array
._class: "com.museum.museumServer.main.models.Piece"
```

```
._id: "hp omnigo 700ix"
name: "hp omnigo 700ix"
text: "Ordenador de "bolsillo""
year: "1995"
createdate: 2019-05-25T18:57:09.157+00:00
qr: "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."
room: "1.1.2"
brand: "HP"
visits: "0"
donor: "Prueba"
> images: Array
> types: Array
._class: "com.museum.museumServer.main.models.Piece"
```

Después del borrado:

```
QUERY RESULTS 1-4 OF 4

>
  _id:ObjectId("5cc73fd58494cd437427d117")
  name:"IBC Business Consultant"
  text:"Calculadora de bolsillo con funciones financieras. Tiene pantalla de c..."
  year:"1986"
  createdate:2019-04-25T00:00:00.000+00:00
  qr:"https://androidayuda.com/app/uploads/2017/10/cK3K3digo-qr-android-ay..."
  room:"3.0.1"
  brand:"HPi"
  visits:"0"
  donor:"Samuel Del Pozo"
  > images:Array
  > types:Array
  _class:"com.museun.museunServer_main.models.Piece"

  _id:ObjectId("5cc740b78494cd437427d118")
  name:"Portable PC (5155)"
  text:"Primitivo ordenador portátil, el primero de IBM (si no contamos el 510..."
  year:"1984"
  createdate:2019-04-25T00:00:00.000+00:00
  qr:"https://androidayuda.com/app/uploads/2017/10/cK3K3digo-qr-android-ay..."
  room:"3.0.1"
  brand:"IBM"
  visits:"0"
  donor:"Jorge Palomeras"
  > images:Array
  > types:Array
  _class:"com.museun.museunServer_main.models.Piece"

  _id:ObjectId("5cc741388494cd437427d119")
  name:"HP-12C"
  text:"Calculadora de bolsillo con funciones financieras y programable.Probab..."
  year:"1981"
  createdate:2019-04-25T00:00:00.000+00:00
  qr:"https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=http://1..."
  room:"3.0.1"
  brand:"HPi"
  visits:"0"
  donor:"Tomasagu"
  > images:Array
  > types:Array
  _class:"com.museun.museunServer_main.models.Piece"

  _id:ObjectId("5ccb8d3bba6290426cbad073")
  name:"II Europius"
  text:"Con unidad de disco, dos manuales, tres tarjetas de expansión y dos jo..."
  year:"1979"
  createdate:2019-05-02T00:00:00.000+00:00
  qr:"https://androidayuda.com/app/uploads/2017/10/cK3K3digo-qr-android-ay..."
  room:"3.0.1"
  brand:"APPLE"
  > images:Array
  > types:Array
  _class:"com.museun.museunServer_main.models.Piece"
```

3.4.5. Ficheros

N/A

Apéndice H

Diseños Técnicos - Iteración 3

Requisitos alcanzados:

- RF 1.3 - **Visualizar pieza:** El sistema debe permitir a un usuario visualizar el contenido visible de una pieza.
- RF 6.3 - **Visualizar sala:** El sistema debe permitir a un usuario visualizar un mapa del edificio que permita ver todas las piezas pertenecientes a una sala.

Las vistas desarrolladas en este trabajo fueron:

Vista Acerca De

En esta ventana se facilita información sobre el museo a los visitantes de la página web. Se podrá visualizar en un mapa la ubicación del museo junto con sus horarios, conocer quienes realizaron la página web y con qué motivaciones y finalmente las personas que han ayudado en la realización de este proyecto

Modelo y Repositorio

- NA. No se usa ningún modelo o repositorio.

Servicios

- NA. No se llama a ningún servicio.

Boceto

Esta vista pretende mostrar al visitante toda la información relacionada con esta aplicación y con el museo. Se proporciona un mapa para poder llegar fácilmente al museo y un horario para planificar la visita. Finalmente se incluyen pinceladas sobre los desarrolladores y terceros.



Boceto de la vista Acerca De

Nombre (id)	Componente	Obligatorio	Validación	Valor defecto	por	Notas
aboutus_lb	label	No	No			
where_lb	label	No	No			
who_lb	label	No	No			
thanks_lb	label	No	No			
map	iframe	No	No			
who_text	text	No	No			
thanks_text	text	No	No			

Vista Mapa Del Museo

En esta ventana se facilita un mapa sobre el museo a los visitantes de la página web. Se podrá visualizar en el mapa la ubicación en plantas y salas del museo junto con la descripción de salas más concretas. Pulsando en las distintas salas se mostrará información a la derecha sobre la sala en cuestión y las piezas ubicadas dentro de ella.

Modelo y Repositorio

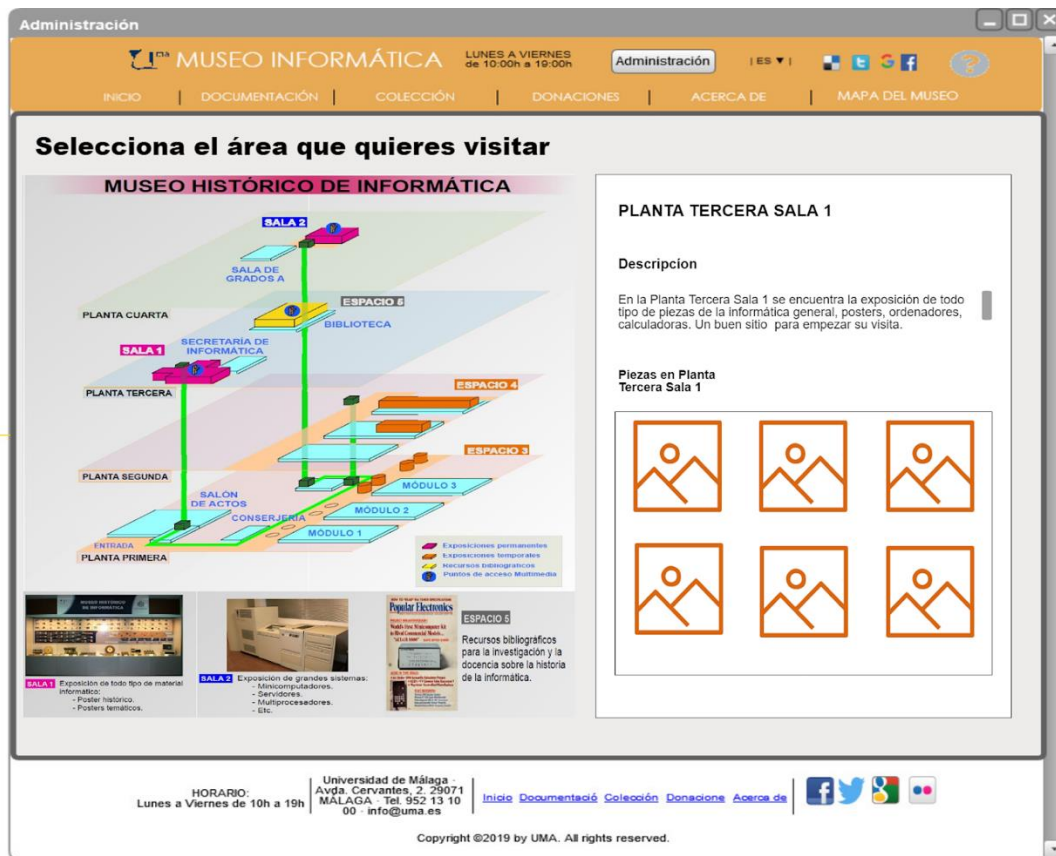
- Piece.java – PieceRepository.java
- Room.java – RoomRepository.java

Servicios

- findAll() - Devuelve una lista con todas las piezas de la base de datos.
- findByRoomNameLike() - Devuelve una lista con todas las piezas de una sala.

Boceto

En esta vista se adaptó la idea del mapa del Museo Metropolitano de Arte(**capítulo 1.6**), pero adaptada al museo. En lugar de un mapa del mundo se usó un mapa proporcionado nuestro tutor. Este mapa muestra las distintas salas donde están expuestas las piezas, en un panel de la izquierda se muestran las piezas encontradas, de una manera más simple(con menos datos) que en la vista colección. Por otro lado, se puede encontrar información de cada sala.



20.1 Boceto vista Mapa Del Museo

Nombre (id)	Componente	Obligatorio	Validación	Valor por defecto	Notas
title_lb	label	No	No		
map_img	img	No	No		
roomName_lb	label	No	No		
description_lb	label	No	No		
roomDescription-text	text	No	No		
roomLocation_text	label	No	No		
piecesTable	table	No	No	pieces	Se muestran todas las piezas de la sala y se puede acceder a la página de la pieza pulsando en ella.