



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DEL SOFTWARE

**DESARROLLO DE LA APLICACIÓN WEB – MUSEO
E.T.S.I. INFORMÁTICA**

PARTE I

**WEB APPLICATION DEVELOPMENT – E.T.S.I.
INFORMÁTICA MUSEUM**

PART I

Realizado por
Tomás Sayago Gutiérrez

Tutorizado por
Eduardo Guzmán de los Riscos

Departamento
Lenguaje y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, 26 DE JUNIO DE 2019

Fecha defensa: de julio de 2019

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 Trabajo de Fin de Grado está basado en el desarrollo de una aplicación web, más concretamente en la creación de un sitio web para el museo de la ETSI Informática de la Universidad de Málaga.

En este proyecto realizado entre tres alumnos, se efectúa un desarrollo de extremo a extremo. Esto quiere decir que tanto la base de datos, el servidor como la aplicación cliente está implementada por nosotros. De esta manera, se personaliza con totalidad esta intención, adaptándola perfectamente a nuestra idea principal.

Se ha utilizado Angular para la creación del cliente, SpringBoot para el servidor y MongoDB para la base de datos. Las tecnologías han sido estudiadas previo el proceso de implementación, con el objetivo de hacer una aplicación moderna y perdurable.

Palabras clave:

Aplicación para el Museo de Informática, *Angular*, *Spring Boot*, *MongoDB*.

Abstract

This Bachelor thesis is based on the development of a web application, in particular, the creation of a website for the ETSI Informatica museum of the University of Malaga.

This project has been carried out between three students, and an end-to-end development has been effectuated. That means, both the database, the server and the client application are implemented by us. In this way, the plan is completely customized, adapting it to our main idea.

Angular has been used for the creation of the client, SpringBoot for the server and MongoDB for the database. The technologies have been studied prior to the implementation process, with the aim of making a modern and lasting application.

Keywords:

Informatica Museum's web application, Angular, SpringBoot, MongoDB.

Índice

<i>Resumen</i>	1
<i>Abstract</i>	1
<i>Índice</i>	1
<i>Introducción</i>	7
1.1 <i>Motivación</i>	7
1.2 <i>Objetivos</i>	7
1.3 <i>Metodología</i>	8
1.4 <i>Tecnologías y herramientas utilizadas</i>	10
1.4.1 <i>Aplicación cliente</i>	10
1.4.2 <i>Servidor</i>	11
1.4.3 <i>Base de datos</i>	11
1.4.4 <i>Herramienta de gestión de tareas</i>	12
1.4.5 <i>Control de versiones</i>	12
1.4.6 <i>Otras herramientas</i>	13
1.5 <i>Estudio de otros trabajos similares</i>	13
1.6 <i>Estructura de la memoria</i>	16
<i>Fases del desarrollo del proyecto</i>	17
2.1 <i>Iteración 0</i>	17
2.2 <i>Iteración 1</i>	22
2.3 <i>Iteración 2</i>	29
1. <i>Vista Log In</i>	29
2. <i>Documentación CRUD</i>	32
2.1 <i>Vista Lista de documentos - Zona administrador</i>	32
2.2 <i>Vista Crear/Editar Documento - Zona administrador</i>	35
2.3 <i>Vista Crear/Editar Artículo - Zona administrador</i>	37
2.4 <i>Iteración 3</i>	43
2.4.1 <i>Vista donaciones</i>	43
2.4.2 <i>Seguridad del servidor</i>	45
<i>Conclusiones y líneas futuras</i>	47
3.1 <i>Conclusiones</i>	47
3.2 <i>Líneas futuras</i>	48
<i>Referencias</i>	49
<i>Manual de Instalación</i>	53
1. <i>Requerimientos</i>	53
1.1 <i>Compilación y despliegue del servidor</i>	53

1.2 Cliente.....	55
<i>Manual de Usuario</i>	57
<i>Requerimientos:</i>	57
1. Navegación por la web.....	57
1.1 Inicio	57
1.2 Documentación.....	57
1.3 Colección.....	57
1.4 Donaciones	58
1.5 Mapa del Museo	58
1.6 Acerca de	58
1.7 Accesos Directos	58
2. Zona Administración.....	58
2.1 Documentación.....	59
2.1.1 Crear	59
2.1.2 Editar.....	60
2.1.3 Eliminar	60
2.2 Colección.....	60
2.2.1 Crear	60
2.2.2 Editar.....	61
2.2.3 Eliminar	62
2.3 Marca	62
2.3.1 Crear	62
2.3.2 Editar.....	62
2.3.3 Eliminar	63
2.4 Tipo.....	63
2.4.1 Crear	63
2.4.2 Editar.....	64
2.4.3 Eliminar	64
2.5 Sala.....	64
2.5.1 Crear	64
2.5.2 Editar.....	65
2.5.3 Eliminar	65
3. Manual de Usuario	66
<i>Documento General de Requisitos</i>	67
1. Introducción.....	69
1.1. Objetivos.....	69
1.2. Metodología.....	69
1.3. Resumen de los participantes	69
1.4. Requisitos funcionales	70
2. Requisitos de documentación.....	72
2.1. Manual de usuario	72
2.2. Modelo de dominio.....	73
2.3. Modelo conceptual de la base de datos.....	73
2.4. Modelos de navegación	73
3. Cuestiones abiertas	76
<i>Pruebas de Sistema - Iteración 1 - Inicio</i>	77

1. Contenido	77
2. Control de versión	77
2.1. Control de cambios	77
3. Pruebas de sistema	77
3.1. TC001.....	78
3.1.1. Descripción	78
3.1.2. Resultado	78
3.1.3. Evidencias	78
3.1.4. Base de datos.....	79
3.1.5. Ficheros	80
3.2. TC002.....	80
3.2.1. Descripción	80
3.2.2. Resultado	80
3.2.3. Evidencias	80
3.2.4. Base de datos.....	83
3.2.5. Ficheros	83
3.3. TC003.....	83
3.3.1. Descripción	83
3.3.2. Resultado	84
3.3.3. Evidencias	84
3.3.4. Base de datos.....	90
3.3.5. Ficheros	90
<i>Pruebas de Sistema - Iteración 2 - Administración.....</i>	<i>91</i>
1. Contenidos.....	91
2. Control de versión	91
2.1. Control de cambios	91
3. Pruebas de sistema	91
3.1. TC001.....	92
3.1.1. Descripción	92
3.1.2. Resultado	92
3.1.3. Evidencias	92
3.1.4. Base de datos.....	94
3.1.5. Ficheros	94
3.2. TC002.....	95
3.2.1. Descripción	95
3.2.2. Resultado	95
3.2.3. Evidencias	95
3.2.4. Base de datos.....	97
3.2.5. Ficheros	97
3.3. TC003.....	97
3.3.1. Descripción	97
3.3.2. Resultado	97
3.3.3. Evidencias	97
3.3.4. Base de datos.....	102
3.3.5. Ficheros	103
3.4. TC004.....	103
3.4.1. Descripción	103

3.4.2. Resultado	104
3.4.3. Evidencias	104
3.4.4. Base de datos	108
3.4.5. Ficheros	108
3.5. TC005.....	108
3.5.1. Descripción	108
3.5.2. Resultado	108
3.5.3. Evidencias	108
3.5.4. Base de datos	116
3.5.5. Ficheros	118
3.6. TC006.....	118
3.6.1. Descripción	118
3.6.2. Resultado	118
3.6.3. Evidencias	118
3.6.4. Base de datos	119
3.6.5. Ficheros	121

1

Introducción

1.1 Motivación

Todo nace a partir de una propuesta del tutor del TFG. Nuestra idea principal era desarrollar una aplicación web completa usando tecnologías y lenguajes de programación distintos a los ya aprendidos durante la carrera. Así surgió la idea de crear esta aplicación aplicando todos los métodos ingenieriles aprendidos, desde la obtención de requisitos y maquetación del proyecto, hasta la realización de documentos de casos de pruebas pasando por el desarrollo por capas de la aplicación.

El hecho de que la página web del museo de nuestra escuela sea creada por nosotros también nos enorgullece. Se pretende que esta plataforma sea una herramienta de utilidad en un futuro para la Escuela de Informática, proveyendo de una recopilación de todas las piezas, documentos y artículos que actualmente posee. Esto hará que resulte más fácil acceder a esta información.

1.2 Objetivos

Se ha impuesto como objetivo entregar un producto acorde con los requisitos planteados en las fases iniciales del proyecto. Éste debe cumplir, al menos, con las funcionalidades mínimas exigidas por el cliente, y que hace de esta aplicación una herramienta práctica de cara a los usuarios que la visiten y de los administradores que la dirijan. Sobre ella, se han aplicado todos los conocimientos

adquiridos durante el grado, desde buenas prácticas de programación, hasta criterios de usabilidad y leyes y derechos de Internet.

Por otro lado, mi objetivo personal siempre ha sido aprender. Ser capaz de involucrarme en un proyecto independiente, en el que todo se construye a partir de mis propias ideas. Además, ser participe en la toma de decisiones importantes a la hora de diseñar un producto, requiere un grado de profesionalidad y perfección que nunca antes me había enfrentado a él.

Asimismo, el hecho de tener los conocimientos suficientes para crear una aplicación web funcional totalmente personalizada y creada al milímetro siempre me ha llamado la atención, por lo que vi una posibilidad única de aprovechar este trabajo de fin de grado para llevarla a cabo. La realización completa de una web también implica nuevos conocimientos sobre manejo de bases de datos y creación de servidores, además de un gran sinfín de ideas, algoritmos y, en general, formas de hacer las cosas que desconocía y se serán de gran ayuda en un futuro de cara a afrontar nuevos problemas.

Finalmente, quería desarrollar mi sentido de la autosuficiencia, aprender a resolver problemas por mí solo y familiarizarme con la búsqueda de información específica en el inmenso mundo de Internet.

1.3 Metodología

Tras ciertos debates sobre qué técnica era la más eficaz para este proyecto nos decantamos por seguir una metodología ágil. Agile proporciona un elevado grado de rapidez y flexibilidad, dos aptitudes esenciales observando el corto plazo que teníamos para realizar el trabajo.

Así, uno de los miembros del grupo acaparó el papel de *Scrum Master*, elaborando una lista de tareas en la herramienta de gestión *Trello* que cumplieran los estados *To Do* (tareas pendientes para hacer), *In Progress* (tareas en la que algún miembro ya está trabajando) y *Done* (tareas terminadas y probadas).

El contenido se repartió entre los integrantes del grupo, en el que cada uno de ellos se encargaban de una parte de la web distinta. El funcionamiento es muy sencillo, para desarrollar una funcionalidad se comenzaba por la recopilación de información, tales como diseños técnicos, diagramas de navegación, bocetos de las pantallas. Posteriormente, se empieza el desarrollo probando su funcionalidad sobre la marcha y midiendo si su actividad es correcta, para acabar ofreciendo una solución final. Siguiendo Agile, si se detecta un cambio, éste se podrá incluir aunque la tarea se encuentre en el estado *In Progress* o *Done*. Para dar por

finalizada una labor, es necesario que la misma haya pasado todas las pruebas unitarias y de sistema correspondientes.

Al ser un trabajo en grupo, ha sido necesario realizar una parte inicial en conjunto. Para ello se ha dividido el ciclo de vida del proyecto en diferentes fases.

A la fase principal donde se recoge toda la labor colectiva la hemos denominado *Iteración 0*. En ella, todos los componentes del equipo trabajamos codo con codo para definir los requisitos iniciales cruciales antes de comenzar el desarrollo individual. Cada miembro comenzó a recopilar información sobre un tema específico, poniéndola en común y documentándola mediante diagramas, modelos y referencias.

Como resultado surgieron ciertos documentos:

1. Documento General de Requisitos(DGR): Recoge, analiza y define las características y necesidades de alto nivel del sistema. Se centrará en describir las expectativas de cada una de las partes del proyectos y usuarios finales.

2. Modelo conceptual de la base de datos: Representa la realidad a un alto nivel de abstracción. Se define una estructura con todos los objetos, sus atributos y la relación entre ellos que almacenará nuestra aplicación.

3. Modelo del dominio: Proporciona una visión estructural del dominio. Se presenta como un diagrama de clases en UML que contiene conceptos propios de un sistema software y, también, de la propia realidad física.

4. Estimaciones de tiempos: Se realizaron en conjunto una evaluación del esfuerzo necesario para acabar cada iteración. Así, se pudo organizar el ciclo de vida del proyecto considerando el tiempo del que disponíamos. Tras las aproximaciones, se fijaron unas fechas para crear un entregable y presentárselo al cliente.

Antes de comenzar con las fases de desarrollo me gustaría aclarar que la división del trabajo ha sido de extremo a extremo. Con esto quiero decir que cada miembro ha trabajado tanto en las capas más cercanas al cliente como es la interfaz de usuario, hasta en las más internas como son el servidor y la base de datos. De esta manera, se ha logrado que mis compañeros y yo aprendamos al completo el desarrollo web.

La segunda iteración (*Iteración 1*) está centrada en la vista predeterminada de los usuarios. Es decir, el acceso a visualizar la información del museo.

La *Iteración 2* amplía funcionalidad añadiendo el rol de administrador. Se crea una zona restringida accesible a ella a través de una pantalla de inicio de sesión. Esta nueva área dispone de las operaciones necesarias para crear, editar o borrar cualquier elemento guardado. En este punto, ya se cuenta con una página funcional que cumple los requisitos mínimos pero que, aún, le quedan muchas cosas por perfeccionar.

Por último, la *Iteración 3* está dedicada a añadir nuevas funciones, optimizar el producto y arreglar aquellos fallos detectados y apuntados para su resolución futura en un estado *Backlog*. En mi caso, se implementará la seguridad del servidor. Este aspecto es muy importante, ya que impide que cualquier usuario pueda realizar cualquier petición a la base de datos, como por ejemplo, borrado de información.

Estas fases serán ampliamente detalladas paso a paso en el capítulo **2. Fases del desarrollo del proyecto.**

1.4 Tecnologías y herramientas utilizadas

Como ya se ha citado antes, el desarrollo de la aplicación está dividida por tres capas principales: aplicación cliente (*front-end*), servidor (*back-end*) y base de datos (*back-end*). Por otro lado, también se comentarán las herramientas utilizadas para la organización de las tareas o control de versiones, así como cualquier material auxiliar que ha sido útil como apoyo a ellas.

1.4.1 Aplicación cliente

En un primer lugar se pretendió hacer esta parte usando *Ionic Framework*, que usa Angular para crear aplicaciones móviles híbridas. Pero este mismo fue la causa primordial para decantarnos por usar simplemente Angular. *Ionic* está más orientado a realizar webs para móviles mientras que nuestro objetivo era crear un sitio web para navegadores.

De este modo, comenzamos a usar Angular junto con el editor de código fuente desarrollado por Microsoft, *Visual Studio Code*. Angular provee el comando *ng* que sirve para crear proyectos, componentes o módulos, muy útiles para el desarrollo web. Además, dota de componentes propios y compatibles que aportan gran variedad de elección a la hora de diseñar el sitio. Como complemento, se ha

usado Node.js para gestionar los paquetes y elementos auxiliares añadidos al proyecto.

1.4.2 Servidor

Las tecnologías que han sido usadas para crear el servidor se eligieron en un primer momento y no se ha requerido de cambios.

Su construcción se ha basado en *SpringBoot Framework* que usa *Java* como lenguaje nativo de programación. SpringBoot facilita la generación de un proyecto y el despliegue del mismo ya que consta con una herramienta complementaria llamada *Spring Initializr*. Initializr crea una estructura base del servidor donde el usuario puede elegir qué dependencias requiere y qué sistema de automatización de construcción, a escoger entre gradle o maven. Se usa el estilo de arquitectura *REST* a la hora de realizar una comunicación entre el cliente y el servidor. Para ver con más detalle el proceso de creación del servidor, diríjase hacia el apartado **2.1 Iteración 0**.

Para lanzar el servidor de forma remota y así poder acceder desde cualquier máquina mediante una url específica (DNS), hemos usado una máquina virtual remota (VM) que nos ha proporcionado *Microsoft Azure*. Mediante *kitty* nos hemos conectado a esta máquina virtual remota mediante una conexión ftp y hemos realizado el despliegue en ella (<http://localhost:8080>). Para poder acceder desde fuera de la VM, hemos enrutado los puertos, para poder acceder al puerto 8080 de forma remota. Así, con la DNS que nos proporciona Azure y el puerto, podemos obtener respuesta de nuestro servidor y usar estos servicios REST para rescatar la información almacenada en nuestra base de datos.

Por otro lado, el entorno de desarrollo utilizado es *IntelliJ IDEA*. El IDE desarrollado por *JetBrains* proporciona gran soporte para nuestro framework Spring. Además, usa el servidor de aplicaciones *Apache Tomcat* que nos permite realizar el despliegue del servidor.

1.4.3 Base de datos

Como gestor de almacenamiento se ha usado *MongoDB*. Ésta es una de las base de datos NoSQL más populares. El porqué de su elección es sencillo: A diferencia de los modelos relacionales, los datos no son almacenados en tablas, sino que se utilizan archivos JSON. Asimismo, al no utilizar un modelo entidad-relación, es posible hacer cambios de los esquemas sin tener que parar la base de datos, lo que implica de un grado de adaptación mucho mayor.

Para añadir, se ha utilizado *MongoDB Atlas*, un servicio automatizado en la nube. Atlas nos permite tener la base de datos remota y de forma común a todos los miembros del grupo. Esto hace que reduzca el tiempo para su administración y que haya consistencia en los datos usados de cara a hacer pruebas en la aplicación cliente.

1.4.4 Herramienta de gestión de tareas

Como se ha comentado en anteriores subcapítulos se ha empleado *Trello* como herramienta gestora de tareas. Trello proporciona lista y tarjetas donde se puede añadir todas las tareas que se requieren hacer y en qué estado se encuentra. Además, permite asignar uno o más miembros del equipo a cada tarea para saber quién se ha encargado de esa labor, añadir comentarios, descripción, enlaces y fotos, plazos de vencimiento, etc. A continuación, se puede ver una imagen del estado de las tareas de la iteración 1:

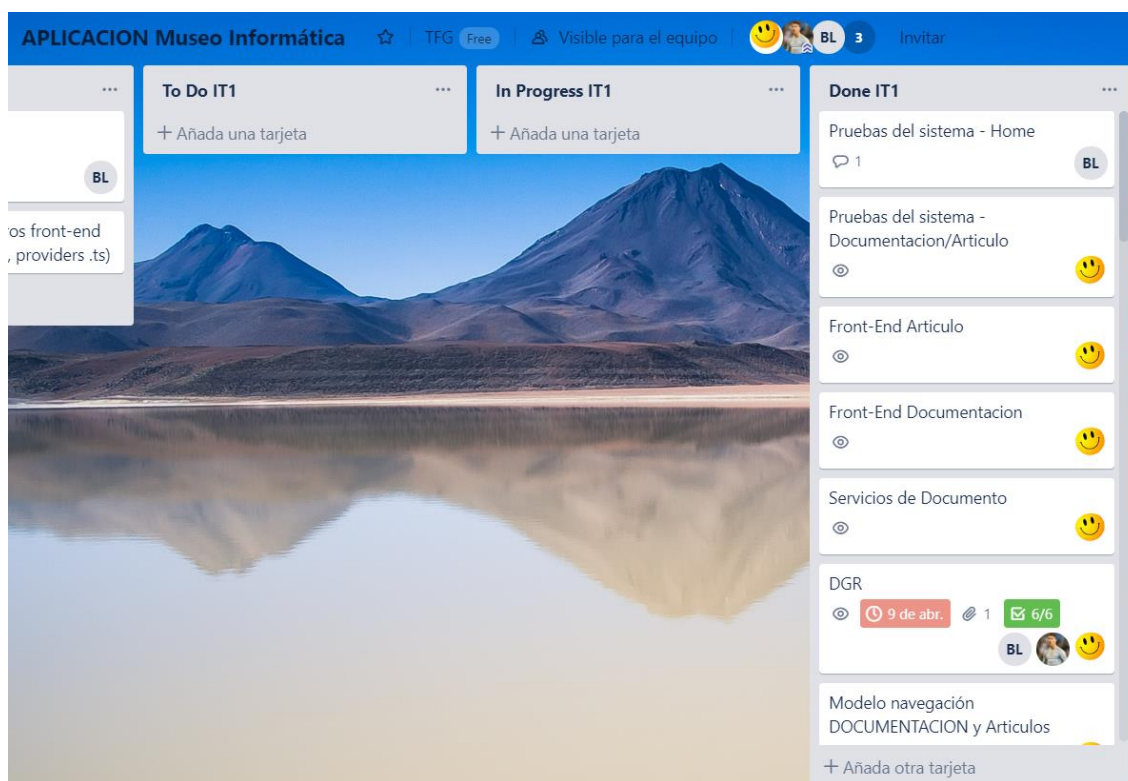


Figura 1. Tablero de gestión de tareas en Trello

1.4.5 Control de versiones

Pensando en la eficiencia y confiabilidad del mantenimiento de versiones se ha usado el famoso software Git. Git proporciona unos repositorios donde se puede subir un elevado número de código fuente. Igualmente, otorga la posibilidad

de crear ramas donde los desarrolladores podrán realizar sus cambios sin apuros a estropear la implementación que ya funciona o tener conflictos con el desarrollo de otra persona. Por otro lado, existirá una rama principal (rama máster) donde se unirán todos los avances de los compañeros.

Adicionalmente, se han etiquetado versiones del proyecto al finalizar las iteraciones. De esta manera, se puede recuperar fácilmente el estado del proyecto con las funcionalidades de cada fase, un estado que pasa todas las pruebas del sistema en un momento determinado y es estable. Esto es muy útil en caso de que se produzca algún fallo y se quede la aplicación inservible. De esta manera se podrá recuperar la versión etiquetada en cualquier momento.

Los enlaces a los repositorios creados para la aplicación cliente y el servidor se pueden encontrar en las *referencias* de esta memoria.

1.4.6 Otras herramientas

Como aplicaciones auxiliares cabe mencionar:

1. **Google Calendar:** Agenda y calendario electrónico útil para la planificación y determinación de tiempos estimados de la totalidad del ciclo de vida del proyecto.

2. **Slack:** Herramienta de comunicación en equipo. Ha sido utilizada para tener una relación más directa con el tutor y compañeros del TFG. Así, el equipo siempre ha estado informado de todos los cambios y novedades del trabajo.

3. **Google Drive:** Servicio de alojamiento de archivos donde se ha almacenado todos los documentos, imágenes y datos varios en la nube. De esta forma, se ha dispuesto de una copia de seguridad de los informes generados en un sitio seguro y común para todos los integrantes.

4. **Cacoo:** Software web para diseñar y dibujar diagramas. Ha sido utilizado para la realización de los bocetos o mockups previo a la implementación.

1.5 Estudio de otros trabajos similares

Antes de comenzar con el desarrollo, se realizó un estudio exhaustivo con el fin de encontrar un estándar a seguir para que nuestro diseño sea consistente con otras páginas de museos ya existentes. De igual manera, se pretendió buscar modelos y formas de presentar o buscar la información que se adaptase lo máximo posible a la idea que teníamos en mente.

Respecto a la estructura principal de nuestra página nos hemos basado en el encabezado y pie de la web oficial del *Museo Picasso de Málaga*.



Figura 2. Página principal Museo Picasso Málaga

El esbozo de la Figura 2 nos ha servido para recoger las siguientes ideas:

1. Logo o nombre del museo. Sobre este elemento se puede hacer clic y se redirige a la página principal.

2. Horario de apertura. Información sobre los días y horas a la que se puede visitar el museo de forma presencial.

3. Selección de idioma. Desplegable en el que se puede elegir el lenguaje con el que desea visualizar la página.

4. Redes sociales. Iconos que se pueden clicar para ver la red social acorde con su logo corporativo.

5. Carrusel. Muestra algunas piezas del museo aleatoriamente junto con breves datos de la misma.

6. Márgenes laterales. La apariencia del sitio web será tal que el contenido se mostrara en una columna central, dejando unos márgenes laterales a ambos lados.

Por otro lado, haciendo referencia a la forma de explorar o presentar los diferentes elementos del museo, nos hemos basado en la típica búsqueda de filtro y tarjetas con información. Más concretamente en la web oficial del *Museo Thyssen*.



Figura 3. Página de búsqueda Museo Thyssen

Como se puede ver en la Figura 3, hemos escogido los diferentes componentes para incluirnos en nuestra aplicación con carácter similar:

1. Buscador por palabras. Se podrá buscar un elemento específico examinando la palabra introducida entre los títulos o descripciones de los componentes.

2. Filtro de búsqueda. Es útil introducir una búsqueda más personalizada para depurar la información que se requiere encontrar.

3. Número de coincidencias encontradas. Es importante que el usuario conozca la cantidad de resultados que se han obtenido.

4. Tarjetas de visualización. La información se presentará en forma de postal donde se mostrará una imagen del elemento acompañada de reducidos datos sobre el mismo.

Todos los enlaces a los sitios utilizados en este apartado están en las referencias de la memoria.

1.6 Estructura de la memoria

Esta memoria está dividida en tres capítulos principales. Cada uno de ellos, está a su vez dividido en subcapítulos donde se abarcarán temas importantes acerca del proyecto. Así, se podrá apreciar con gran detalle la evolución del mismo, partiendo de una idea principal, y llegando hasta el resultado final.

Para ello es necesario redactar todo lo relacionado con el propósito. Lluvia de ideas, investigación de las herramientas, estudio de sitios webs similares, debates y conclusiones alcanzadas en las reuniones de equipo. Igualmente es muy importante definir la metodología utilizada, pues es crucial para entender la transformación del boceto principal. Todo esto se manifiesta en el primer capítulo de este documento.

A continuación, es sustancial desglosar el proceso de desarrollo, puesto que se trata de la parte de más esfuerzo. En ella se detallarán todos los pasos de construcción del producto. Está dividido en diferentes iteraciones como se señala en el apartado *1.3 Metodología*.

En tercer lugar, se encuentra una de las lecciones en la cual se deja de lado el detalle técnico y se alcanza una redacción subjetiva. Una reflexión que tiene lugar en el desenlace de la intención y que narra todas las críticas y posibles mejoras que, desde la más humilde y experta opinión, un creador puede dar a su obra.

Para terminar, se podrán encontrar todas las referencias disfrutadas y consultadas para nutrirme de saber y llevar a cabo las ideas anotadas, como varios anexos aportando información técnica sobre algunos puntos específicos y noticiados durante el escrito.

2

Fases del desarrollo del proyecto

2.1 Iteración 0

Como ya he mencionado anteriormente, ésta es la fase común del proyecto. Se puede decir que son los cimientos, donde nacen las ideas y se define una estructura a seguir durante todo el ciclo de vida.

El primer paso fue hablar con el tutor del TFG, que actuó como cliente ficticio. A él se le preguntó todos los detalles que se nos ocurrían de cara a tener una especificación de qué es lo que el cliente quiere con la máxima definición.

Una vez que la idea quedó clara, comenzamos a reflejar el planteamiento sobre el *Documento General de Requisitos* (véase Apéndice C). En él podemos encontrar plenitud de conceptos sobre el proyecto.

En primer lugar, es fundamental definir quién va a visitar nuestra página y definir qué grado de libertad tendrá. Es decir, se tiene que diferenciar entre un usuario normal que puede visitar la página libremente, sin necesidad de aportar unos credenciales de autenticación, y visualizar todo el contenido permitido, de un usuario administrador que introduciendo los datos de identificación correctos puede acceder a una zona restringida de la web y realizar operaciones críticas sobre el contenido de la misma.

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.

Figura 4. Resumen de los participantes

A continuación, el elemento crucial en la fase de análisis de requisitos, los propios requisitos. Es el paso más importante y el que requiere mayor perfección. Sobre ellos se construirá la aplicación y el objetivo es cumplir el mayor número de requisitos funcionales. Es nuestro caso, hemos catalogado la lista por nivel de prioridad (alta, media o baja) y hemos adjudicado cada requisito a una iteración específica. De esta forma, al finalizar cada etapa podremos asegurarnos y comprobar que hemos alcanzado aquellos que se querían cubrir.

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)

Figura 5. Algunos de los requisitos funcionales

Por otro lado, teníamos que diseñar varios modelos donde se pudiese ver claramente la totalidad de la información que se almacenará con el máximo detalle.

Para ello, en modelos de las figuras 6 y 7 se describe cada objeto con sus atributos, tipos, claves primarias y relaciones entre ellos. De esta manera, el cliente podrá confirmar si los datos que se guardarán y, por consiguiente, mostrarán en la página son los deseados.

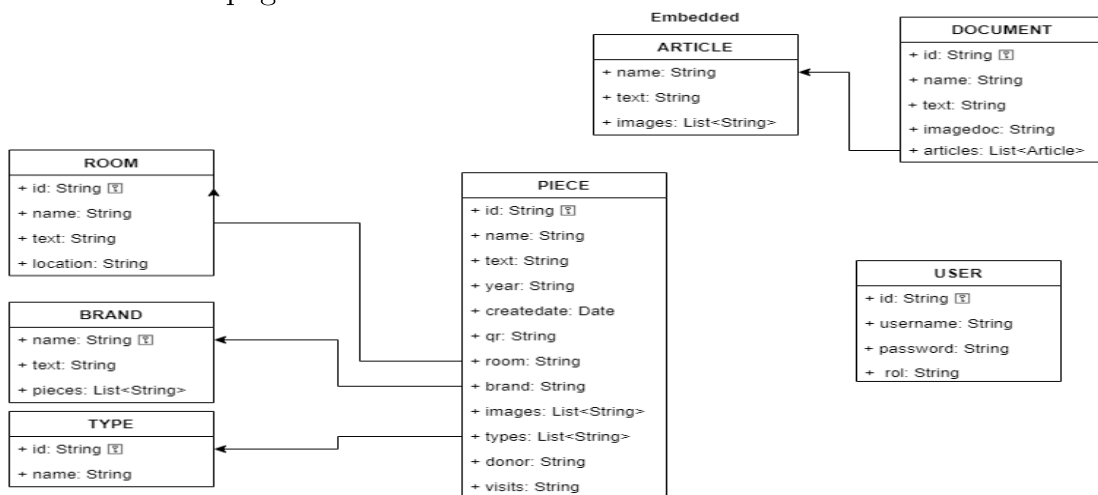


Figura 6. Modelo conceptual de la base de datos

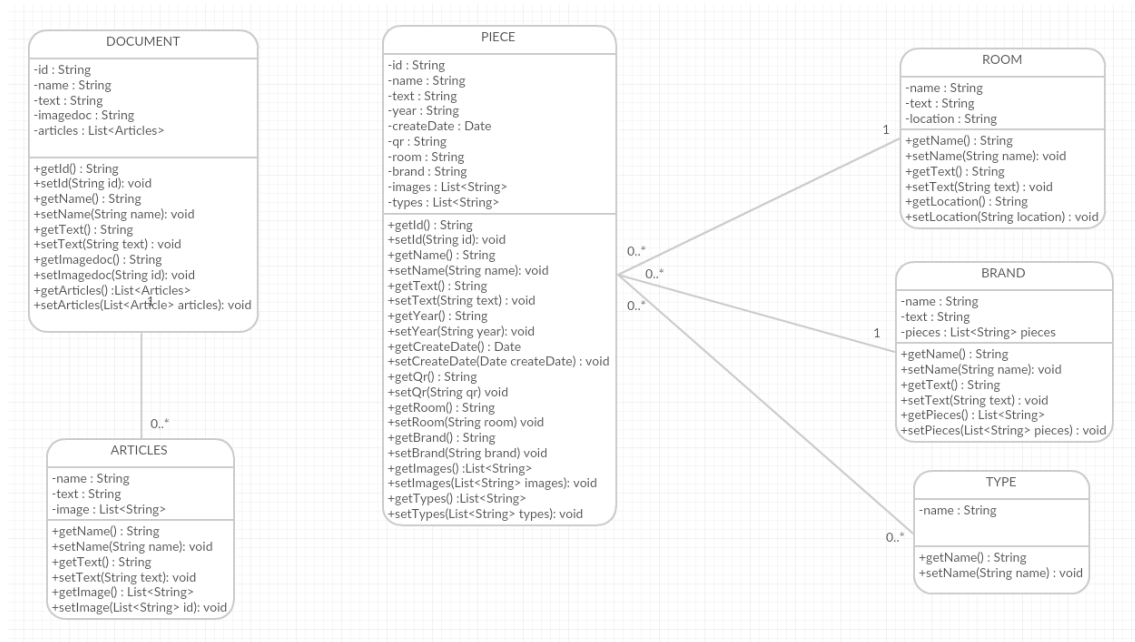


Figura 7. Modelo del dominio

Para añadir a este DGR, decidimos realizar un diagrama de navegación como el de la figura 8. Así, se describe todas las posibles conexiones de una vista con las demás, haciendo uso de los hipervínculos.

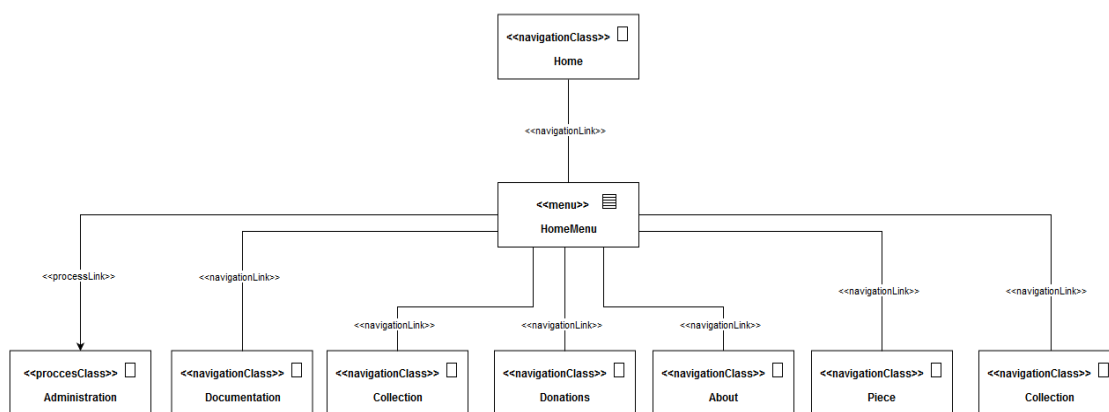


Figura 8. Diagrama de navegación. Pantalla principal.

Este informe fue presentado al cliente, exponiendo los requisitos principales y modelos respectivos. Tras ciertos cambios y puntualizaciones, el cliente aprobó esta especificación. Este fue el pistoletazo de salida hacia la implementación.

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*.

Posteriormente se crearon los modelos respectivos a las tablas de la base de datos y las interfaces repositorios donde se definen todas las operaciones REST. Estos repositorios extienden de *MongoRepository<Modelo,Serializable>* por lo que se implementan el CRUD básico. Además, se pueden definir métodos más personalizados usando patrones que la misma clase padre posee.

```
package com.museum.museumServer.main.repository;

import ...

@Repository
public interface DocumentRepository extends MongoRepository<Document, Serializable>{
    List<Document> findByNameLikeIgnoreCaseOrTextLikeIgnoreCase (@Param("word")String word, @Param("word1")String word1);
}
```

Figura 9. Interfaz *DocumentRepository*.

Para completar y efectuar un trabajo profesional y organizado, se ejecutó un estudio de los tiempos estimados que iban a requerir cada fase. De este modo, los desarrolladores pudimos establecer unas fechas límites para terminar cada iteración, presentar un entregable y efectuar una demostración al cliente.

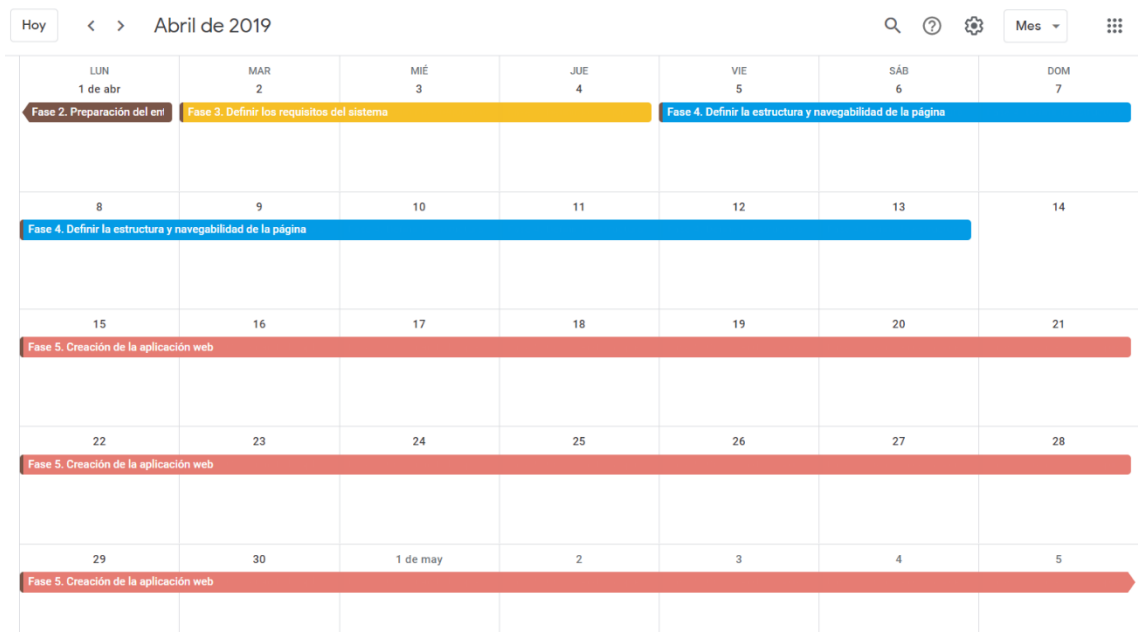


Figura 10. Planificación. Realizado con Google Calendar.

2.2 Iteración 1

Como se detalló en la iteración inicial, el proceso de desarrollo constaría de una primera etapa de generación de documentos, creación de bocetos y diseños técnicos donde se especificaran todas las características de la vista a implementar. A continuación, una etapa intermedia donde se comienza la programación tanto de la parte del cliente como de la implementación de los servicios necesarios. En último lugar, el software desarrollado será sometidos a unas pruebas del sistema.

1. Vista Documentación

En esta ventana se muestran todos los documentos que hay en la base de datos. Consta de un buscador que filtra por nombre de Documento y Artículo. Para reducir la carga de los servicios, se utiliza un botón “Mostrar más” que carga los datos de cinco en cinco.

Requisitos alcanzados

RF 2.3 - Visualizar documento: El sistema debe permitir a un usuario visualizar el contenido visible de un documento.

Modelo y Repositorio

Document.java – DocumentRepository.java

Servicios

Al buscar documento: `findByNameLikeOrTextLike(String word)` – Busca coincidencias entre la palabra proporcionada por el usuario y los nombres de los documentos y sus descripciones.

Ficheros Front-End

Documentation.component.css - documento donde se especifican todas las clases CSS que aportan estilos.

Documentation.component.html - vista en lenguaje HTML de marcado que construye la página web.

Documentation.component.spec.ts - documento para realizar pruebas unitarias. No has sido usado en este proyecto.

Documentation.component.ts - controlador en lenguaje Typescript.

Documentation-routing.module.ts - asistente para el enrutado

Documentation.module.ts - módulo donde se definen todos los ficheros importados, exportados y providers.

Providers

Los providers son clases en Typescript en el proyecto cliente donde se definen las llamadas a los diferentes servicios REST mediante una dirección(endpoint) específico.

DocumentProvider.ts

Boceto



Figura 11. Boceto funcional. Vista Documentación.

Nombre (id)	Componente	Obligatorio	Validación	Valor por defecto
searchField	inputText	No	No	Placeholder: “Buscar Documento”
results	label	No	No	Tamaño de la lista de documentos
table	table	No	No	Si no hay documentos: “No hay documentos para mostrar” ? documents

Figura 12. Tabla de componentes. Vista Documentación.

2. Vista Artículos

En esta ventana se muestran los artículos del documento seleccionado en la anterior ventana. El id del documento se pasará como parámetro en la URL. Se visionará el primer artículo del documento mostrando su título o nombre, la descripción y las imágenes del mismo. En la parte derecha, se sitúa un panel de navegación donde se muestran todos los artículos del documento determinado, pudiendo clicar su nombre y ver su información. La página consta de un menú de paginación donde se podrá navegar al anterior y siguiente al artículo. Además, las imágenes se podrán clicar para verlas a mayor tamaño.

Requisitos alcanzados

RF 3.3 - Visualizar artículo: El sistema debe permitir a un usuario visualizar el contenido visible de un artículo.

Modelo y Repositorio

Document.java – DocumentRepository.java
Article.java – DocumentRepository.java

Servicios

Al cargar la página: getDocumentById(String id) - Busca el documento por su id (id pasado por parámetro en la URL).

Ficheros Front-End

Article.component.css
Article.component.html
Article.component.spec.ts
Article.component.ts
Article-routing.module.ts
Article.module.ts

Providers

DocumentProvider.ts

Boceto



Figura 13. Boceto funcional. Vista Artículo.

Nombre (id)	Componente	Obligatorio	Validación	Valor por defecto
document Label	label	No	No	document.name getDocumentById()
article MainName	label	No	No	article.name
article Description	textarea	No	readonly	article.description
navigation Panel	sidemenu	No	No	documents.articles.name - getArticlesByDocument()
paginator	mat-paginator	No	No	Página 1
galery Label	label	No	No	String text
images List	table	No	No	article.images

Figura 14. Tabla de componentes. Vista Artículo.

Para terminar se pueden apreciar cambios a lo largo del desarrollo de esta iteración. Estas modificaciones se producen debido a complicaciones y limitaciones producidas por el lenguaje utilizado, así se busca una alternativa. Se puede considerar el caso del botón *mostrar más* de Documentación que se decidió hacer una tabla scrollable o la galería de la Artículo que se añadió funcionalidad mostrando un menú para navegar las imágenes.

Otro cambio significativo fue permitir la búsqueda por palabra ignorando las mayúsculas y minúsculas. Para ello tan solo fue modificar los servicios determinados.

Se cambió *findByNameLikeOrTextLike(String word)* por *findByNameIgnoreCaseLikeOrTextIgnoreCaseLike(String word)*.

El documento de pruebas de sistema se puede encontrar en el [Apéndice D](#). En él se da una pequeña descripción del caso de prueba, se muestran los pasos seguidos con sus respectivas capturas de pantalla y los cambios reflejados en la base de datos si corresponde.

El resultado final de estas vistas ha sufrido ciertos cambios de la interfaz de usuario, pretendiendo dar un toque más consistente y moderno. El logo de la aplicación ha sido obtenido del repositorio del apartado para la ETSI Informática en la web oficial de la Universidad de Málaga. Enlace en la referencia.

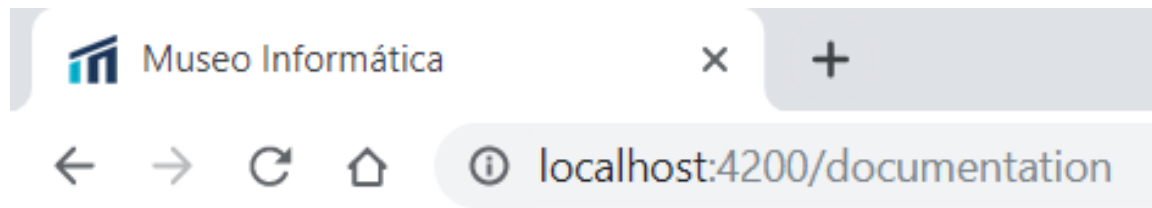


Figura 14. Resultado final. Pestaña navegador.

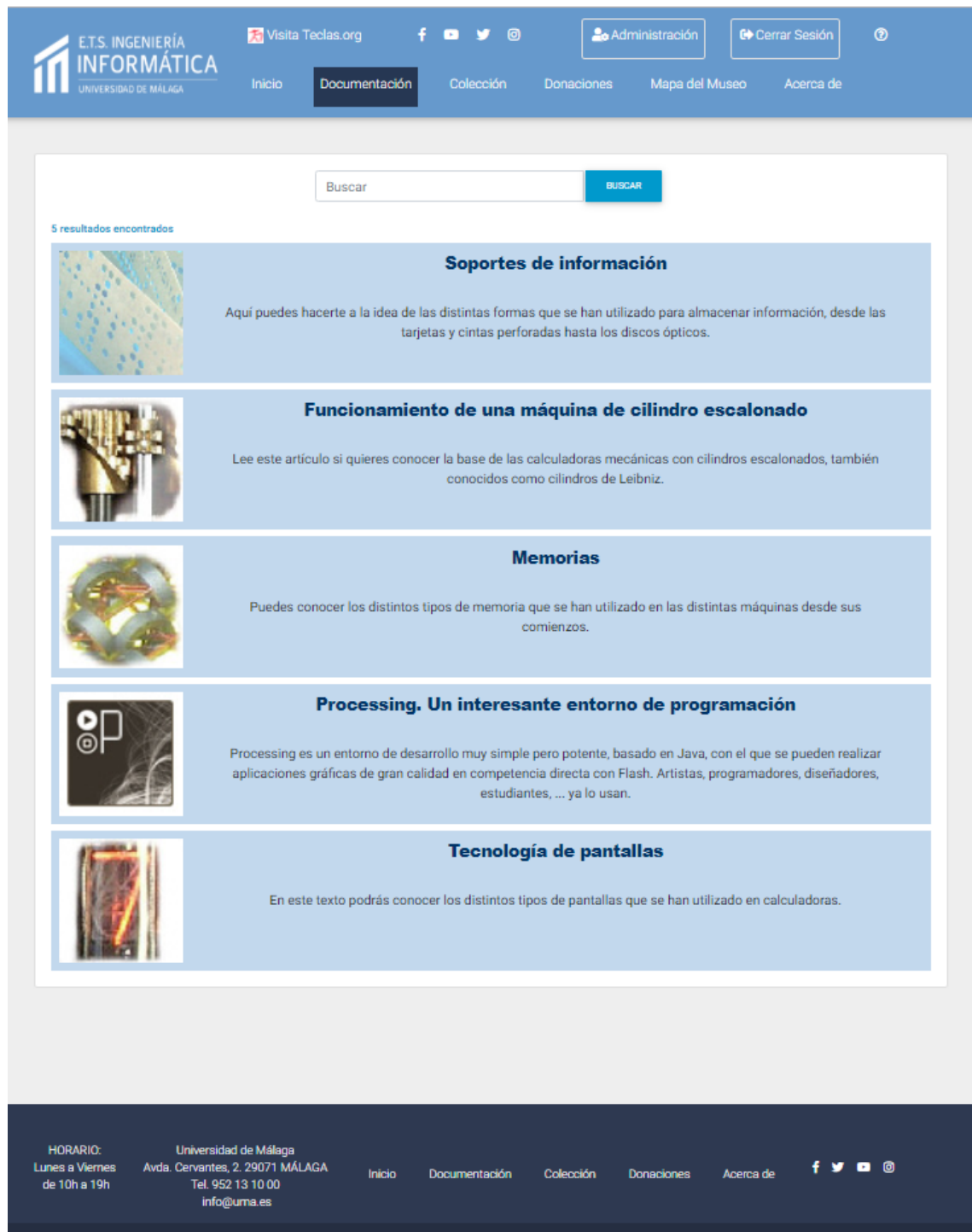


Figura 15. Resultado final. Vista Documentación



[Visita Teclas.org](#)
f
v
t
@

Administración
Cerrar Sesión
?

Inicio
Documentación
Colección
Donaciones
Mapa del Museo
Acerca de

Documento: Memorias

Memorias en calculadoras mecánicas

Las calculadoras mecánicas también tienen registros en los que se almacenan números y resultados. No tienen, claro está, memorias del estilo de los ordenadores, en las que se almacenan tanto datos como programas. Las memorias (o registros) de las calculadoras mecánicas son muy variados.

[Previous](#) [Next](#)

Artículos

- > [Memorias en calculadoras mecánicas](#)
- > [Memoria de Tambor](#)
- > [Memorias de Línea de retardo](#)

Galería de imágenes




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

f
t
v
@

Copyright ©2019 by UMA. All rights reserved.

Figura 16. Resultado final. Vista Artículo.

2.3 Iteración 2

En esta fase intermedia, se procede a desarrollar la zona restringida pensada para que los administradores del sistema tengan una interfaz donde les sea fácil y cómodo acceder y modificar la información de la base de datos.

Para ello, es necesario implementar una pantalla de inicio de sesión, donde las personas que quieran acceder deberán introducir unos credenciales predeterminados (usuario: *admin*, contraseña: *admin*) y proporcionados por los desarrolladores de la aplicación. Estas vistas estarán protegidas, por lo que será imposible acceder a ellas incluso a menos que se introduzcan los datos válidos.

1. Vista Log In

Esta ventana será mostrada al pulsar el botón Administración situado en la barra superior de navegación, en el caso de que no haya ningún usuario ya registrado en sesión. En ella será necesario aportar unos credenciales específicos para entrar en la zona restringida de Administración. En caso que el inicio de sesión sea exitoso, se accederá a un menú de administración. En caso contrario, se mostrará un mensaje de usuario o contraseña incorrecta.

Si hay un usuario registrado, aparecerá otro botón de Cerrar Sesión junto al de Administración. Al pulsar el botón de Cerrar Sesión, el usuario almacenado en sesión será borrado y el usuario será redirigido a la página principal. Si se pulsa el botón de Administración mientras haya un usuario registrado, el usuario será redirigido al menú de administración.

Requisitos alcanzados

RF 7 - Administración: El sistema debe tener una pantalla de inicio de sesión como administrador que acceda a la zona restringida.

Modelo y Repositorio

- **User.java –UserRepository.java**

Guards

Restringe el acceso a determinadas direcciones de la aplicación web.

AdminGuard.java - Implementa CanActivate de @angular/router. True para permitir acceder al path(URL específica de la vista).

- findById(sessionStorage.getItem("user_id")) - El id del usuario con el que se ha accedido será guardado en sesión. Este método comprueba que el rol de este usuario sea de administrador.

Servicios

- findByUsernameAndPassword(String username, String password) - Busca en la colección User si existen los credenciales introducidos.

Ficheros Front-End

Login.component.css
Login.component.html
Login.component.spec.ts
Login.component.ts
Login-routing.module.ts
Login.module.ts

Providers

UserProvider.ts

Boceto

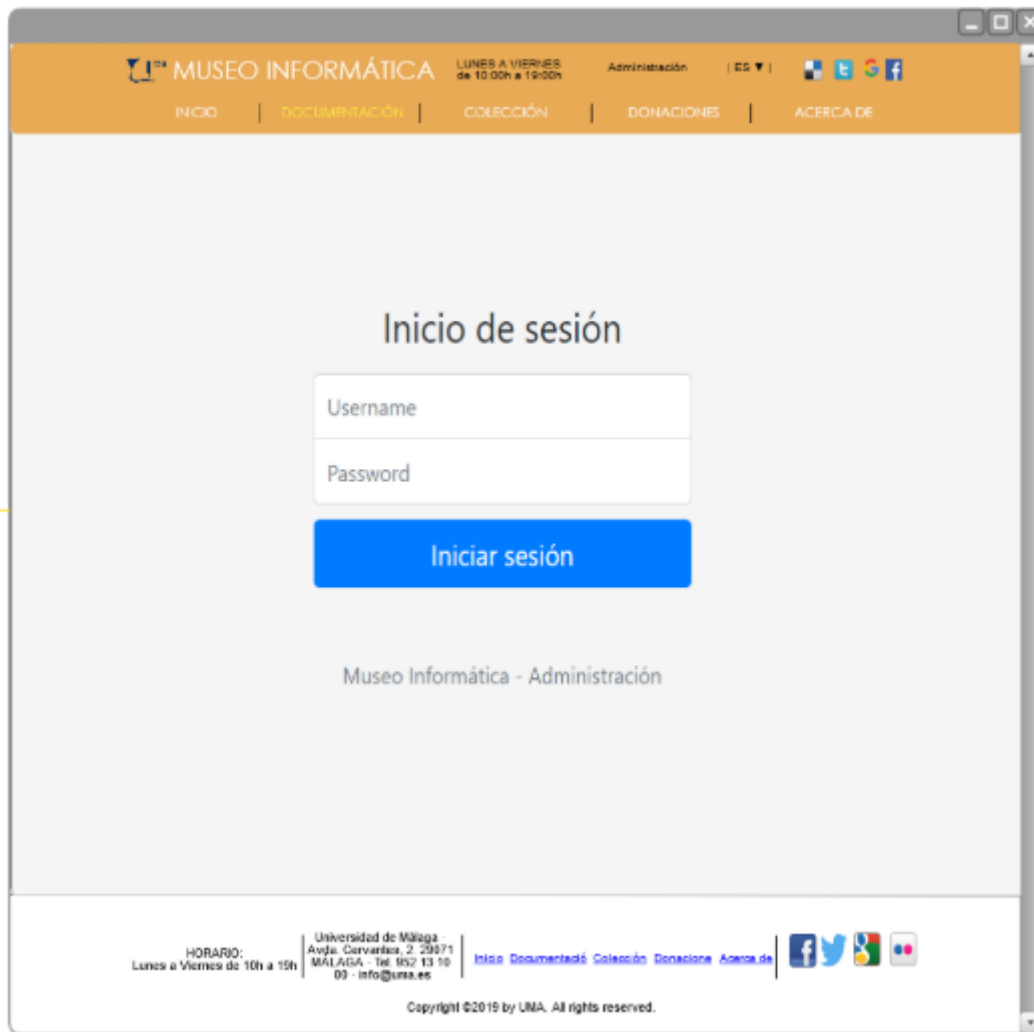


Figura 17. Boceto funcional. Vista Inicio Sesión.

Nombre (id)	Componente	Obligatorio	Validación	Notas
labelTitle	label	No	No	
username	inputText	Si	No	
password	inputText	Si	No	
buttonSignIn*	button	No	Si	Comprueba que username y password no estén vacíos
labelBottom	label	No	No	

Figura 18. Tabla de componentes. Vista Inicio Sesión.

*Este botón ejecutará el servicio `findByUsernameAndPassword(username, password)`. Si la respuesta es nula, se mostrará el mensaje de error. Si la respuesta es un usuario, se guardará su id la sesión del navegador(`sessionStorage.setItem("user_id", id)`) y se transitará a la página correspondiente.

- **NOTA:** La contraseña almacenada en base de datos será un codificada mediante Hash md5. Para realizar la comprobación será necesario obtener el código Hash de la contraseña introducida por el usuario.

```
import { Md5 } from 'ts-md5/dist/md5';  
  
Md5.hashStr("password");
```

Figura 19. Código TypeScript para obtener el Hash de la contraseña.

2. Documentación CRUD

Requisitos alcanzados

RF 2 - Crear documento: Un documento debe tener los atributos nombre, imagen y lista de artículos que contiene.

RF 2.1 - Editar documento: El sistema permite al administrador la edición de documentos que ya habían sido creadas anteriormente.

RF 2.2 - Eliminar documento: El sistema debe permitir al administrado eliminar un documento que ya había sido creado anteriormente.

RF 3 - Crear artículo: Un artículo debe tener los atributos nombre, texto y lista de imágenes asociadas.

RF 3.1 - Editar artículo: El sistema permite al administrador la edición de artículos que ya habían sido creadas anteriormente.

RF 3.2 - Eliminar artículo: El sistema debe permitir al administrador eliminar un artículo que ya había sido creado anteriormente.

2.1 Vista Lista de documentos - Zona administrador

En esta ventana se muestran todos los documentos que hay en la base de datos. Disponemos de un botón “Editar” para editar los datos de un documento determinado y otro botón “Borrar” para eliminar un documento. Además, podemos encontrar un botón “Crear Documento” que dirige a un formulario

donde podemos rellenar los campos con la información del documento que se va a crear.

Modelo y Repositorio

- Document.java – DocumentRepository.java

Servicios

- findAll() - Devuelve una lista con todos los documentos de la base de datos.
- **Al buscar documento:** findByNameLikeOrTextLike(String word) – Busca coincidencias entre la palabra proporcionada por el usuario y los nombres de los documentos y sus descripciones.

Ficheros Front-End

DocumentationCRUD.component.css
DocumentationCRUD.component.html
DocumentationCRUD.component.spec.ts
DocumentationCRUD.component.ts
DocumentationCRUD-routing.module.ts
DocumentationCRUD.module.ts

Providers

DocumentProvider.ts

Boceto



Figura 20. Boceto funcional. Vista Documentación CRUD

Nombre (id)	Componente	Obligatorio	Validación	Notas
documentsTable	table	No	No	Consta de un botón editar y otro eliminar en cada una de sus filas
createButton	button	No	Si	Redirige a la siguiente vista

Figura 21. Tabla de componentes. Vista Documentación CRUD

2.2 Vista Crear/Editar Documento - Zona administrador

En esta ventana se muestra un formulario con todos los campos de un documento. Además, se permitirá crear artículos asociados a ese documento. El botón “Crear Documento” actualizará la base de datos añadiendo el documento creado y redirigirá a la vista anterior donde se muestran todos los documentos.

La vista en modo edición será similar, la única diferencia es que aparecerán los datos rellenos al entrar a la ventana.

Modelo y Repositorio

- Document.java – DocumentRepository.java
- Article.java – DocumentRepository.java

Servicios

- create(Document document) - Crea un nuevo documento en la base de datos.
- update(Document document) - Actualiza el documento en la base de datos.

Ficheros Front-End

MaintenanceDocumentation.component.css
MaintenanceDocumentation.component.html
MaintenanceDocumentation.component.spec.ts
MaintenanceDocumentation.component.ts
MaintenanceDocumentation -routing.module.ts
MaintenanceDocumentation.module.ts

Providers

DocumentProvider.ts

Boceto

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

INICIO | DOCUMENTACIÓN | COLECCIÓN | DONACIONES | ACERCA DE

Crear Documento

Nombre

Descripción

Imagen

Artículos		<input type="button" value="Crear Artículo"/>
Artículo 1		
Artículo 2		
Artículo 3		
Artículo 4		

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.

Figura 22. Boceto funcional. Vista Mantenimiento Documentación.

Nombre (id)	Componente	Obligatorio	Validación	Notas
nameLabel	label	No	No	
inputName	input	Si	No	
textLabel	label	No	No	
inputText	input	No	No	
imageLabel	label	No	No	
inputImage	input	No	No	URL de la imagen
addButton	button	No	Si	Si inputImage está vacío, saltará la validación
imagesList	gallery	No	No	Al clicar sobre el símbolo "X" la imagen se borrará
articlesLabel	label	No	No	
createArticleButton	button	No	No	Redirige a la siguiente vista
articlesTable	table	No	No	Consta de un botón de editar y otro de eliminar en cada una de sus filas
createButton	button	No	Si	Comprobará que todos los campos obligatorios estén rellenos

Figura 23. Tabla de componentes. Vista Mantenimiento Documentación.

2.3 Vista Crear/Editar Artículo - Zona administrador

Esta vista será una ventana emergente que se abrirá sobre la vista anterior. En ella se muestra un formulario con todos los campos de un artículo. Además, se permitirá crear artículos asociados a ese documento. El botón "Crear Artículo" actualizará la lista de artículos que se encuentra en la vista anterior y redirigirá a la misma.

La vista en modo edición será similar, la única diferencia es que aparecerán los datos rellenos al entrar a la ventana.

Modelo y Repositorio

N/A. La información respectiva a los artículos se encuentran embebidos en la tabla del Documento al que pertenecen.

Servicios

N/A

Ficheros Front-End

MaintenanceArticleDialog.component.css
MaintenanceArticleDialog.component.html
MaintenanceArticleDialog.component.spec.ts
MaintenanceArticleDialog.component.ts
MaintenanceArticleDialog -routing.module.ts
MaintenanceArticleDialog.module.ts

Providers

N/A

Boceto

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

INICIO | DOCUMENTACIÓN | COLECCIÓN | DONACIONES | ACERCA DE

Crear Artículo

Nombre

Descripción

Imagen
 Añadir

Crear Artículo

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.

Figura 24. Boceto funcional. Vista Mantenimiento Artículo

Nombre (id)	Componente	Obligatorio	Validación	Notas
nameLabel	label	No	No	
inputName	input	Si	No	
textLabel	label	No	No	
inputText	input	No	No	
imageLabel	label	No	No	
inputImage	input	No	No	URL de la imagen
addButton	button	No	Si	Si inputImage está vacío, saltará la validación
imagesList	gallery	No	No	Al clicar sobre el símbolo "X" la imagen se borrará
createButton	button	No	Si	Comprobará que todos los campos obligatorios estén rellenos

Figura 25. Tabla de componentes. Vista Mantenimiento Documentación.

El documento de pruebas de sistema se puede encontrar en el [Apéndice E](#). En él se incluye una pequeña descripción del caso de prueba, se muestran los pasos seguidos con sus respectivas capturas de pantalla y los cambios reflejados en la base de datos si corresponde.

En las siguientes figuras podemos ver el resultado final de estas vistas.

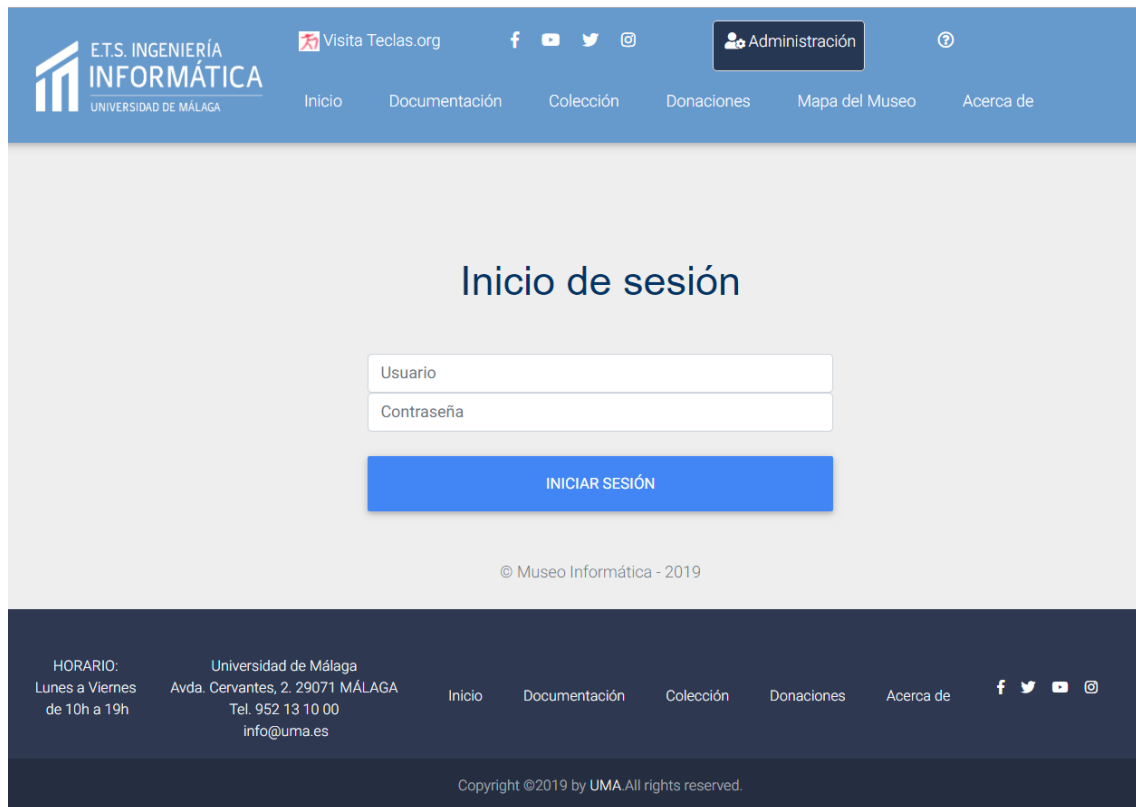


Figura 26. Resultado final. Vista Inicio Sesión.



Figura 27. Resultado final. Vista Documentación CRUD.

ETS. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Visita Teclas.org

f y v t i

Administración

Cerrar Sesión

Inicio Documentación Colección Donaciones Mapa del Museo Acerca de

Editar Documento

Nombre


Memorias

Descripción

Puedes conocer los distintos tipos de memoria que se han utilizado en las distintas máquinas desde sus comienzos.


Imágenes

Imágenes **AÑADIR**



Artículos

3 resultados encontrados **CREAR ARTÍCULO**

Memorias en calculadoras mecánicas	 
Memoria de Tambor	 
Memorias de Línea de retardo	 

VOLVER **EDITAR DOCUMENTO**

Figura 28. Resultado final. Vista Editar Documento.

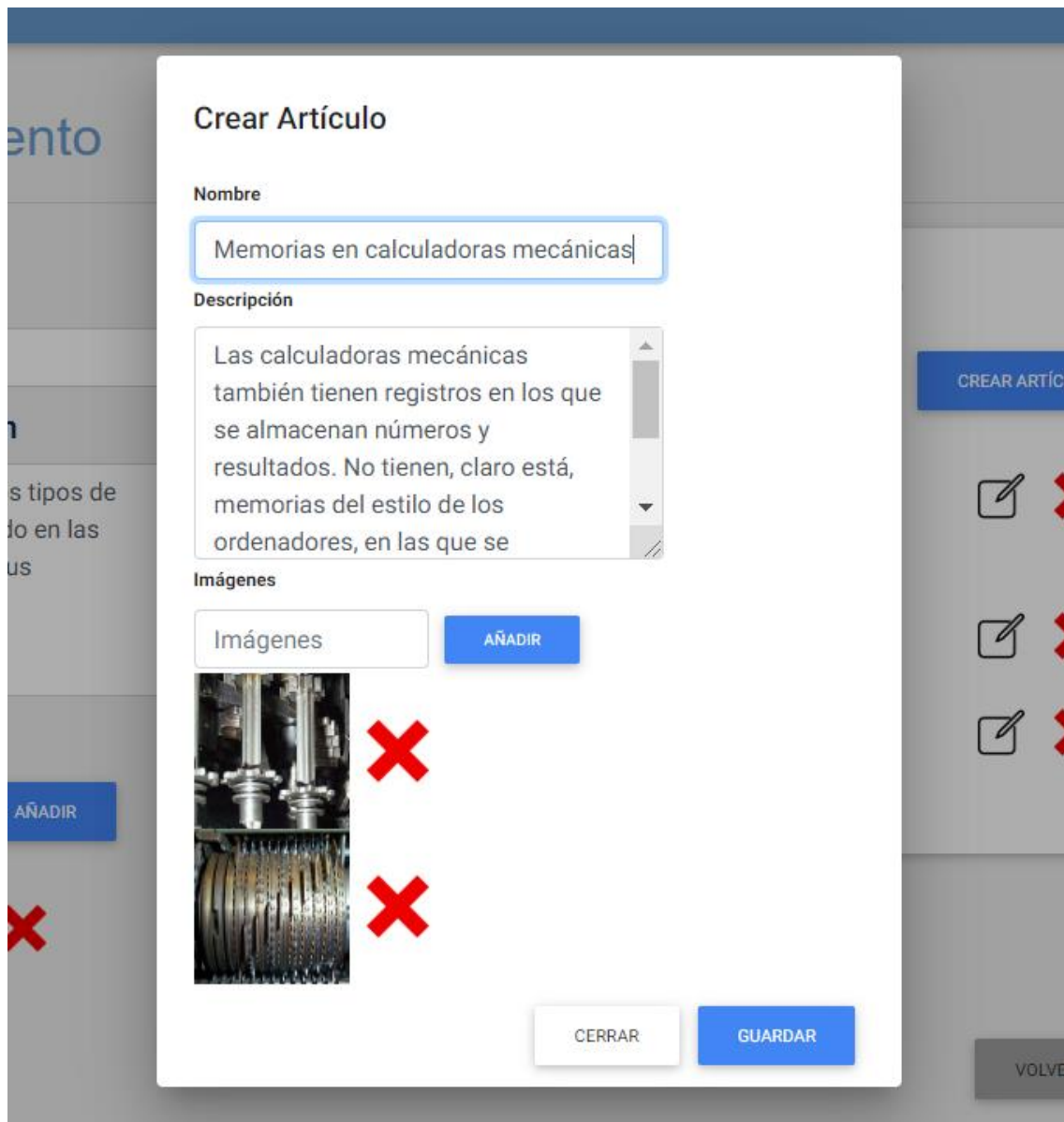


Figura 29. Resultado final. Vista Editar Artículo.

2.4 Iteración 3

En este punto ya contamos con una página con amplia funcionalidad. Tenemos una vista de usuario donde visualizar toda la información y, también, una zona restringida para administradores donde crear, eliminar o editar los datos ya existentes. Entonces, esta fase se planteó para perfeccionar lo anteriormente desarrollado y añadir ampliaciones con el objetivo de hacer el sitio web más moderno, seguro y completa.

2.4.1 Vista donaciones

Siguiendo la página oficial del dueño de museo (teclas.org) hemos querido incluir el apartado de donaciones, donde el fundador da las gracias a los donantes.



Agradecimientos a los donantes al museo

Alberto Lozano Barcia	Hermann Schittenhelm	Luis C Avellan (Mr.Electroplastic)
Alejandro Pequeño Boter	Holger Klee	Luis Nogueira Díaz
Alfonso Carlos Rodríguez Barragán	Miguel Ángel Pascual Pérez	Luiz Alberto Feijó Jr.
Antonio López	Miguel Angel Romero Castillo	Luiz Cláudio Vieira
Carlos Lomeña	Pilar Durán	Manolo González
Casimiro Población	Regalo	Manolo Mallorca
Ciudad de los Niños	S. Pequeño Ciaurritz	María Luisa de Arroita
Claudio Bonilla Toyos	ICET	María Mayans
Enrique Peregrín Pitto	Ingrid Graf	Marido de Nuria
Erna Ehlert	Javier (Cazorla)	Sonja Blaschek
Fali	Javier Moreno y Bernardo	Thomas Heinrich
Fernando Agudo Lencero	Jordi Ferré	Toñi Moreno Navas -APA Valle Inclán
Fernando...	José Sánchez Arjona	Virginia Cañas
Francisco Luis Medina Montoya	Lourdes Martínez Reina	Werner Schneider
Heine Ehlert	Luis Arce Matute	

Donaciones

[¿Quieres contribuir a la causa?](#)

¿Tienes un viejo ordenador y no sabes qué hacer con él?, pues ámate y escríbeme a museo@teclas.org, o llámame por teléfono al **639 11 97 41** preguntando por **Daniel Sancho**.

Por el momento he recibido donaciones desde España (Málaga, Mallorca, Barcelona, Salamanca, Sevilla), Alemania y Brasil.

Figura 30. Vista Donaciones.

Para completar y añadir algún componente interactivo, se decidió hacer un podio dinámico con las tres personas que más piezas han donado.

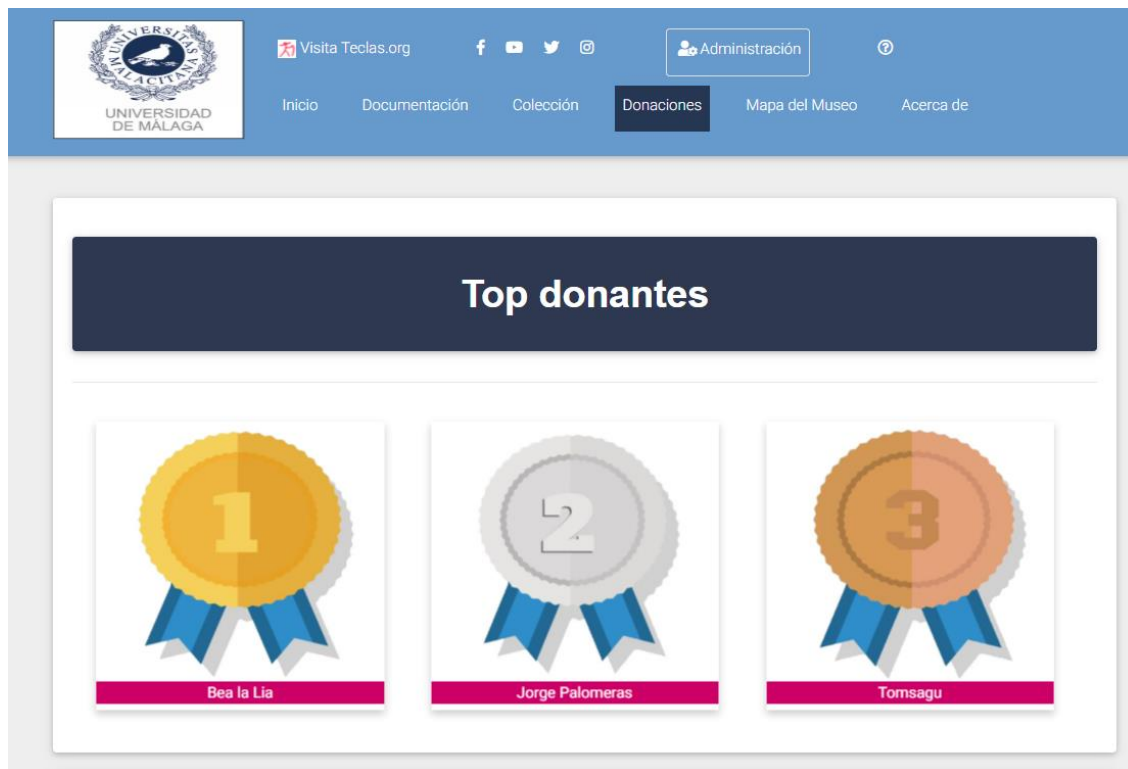


Figura 31. Vista Donaciones. Top Donantes.

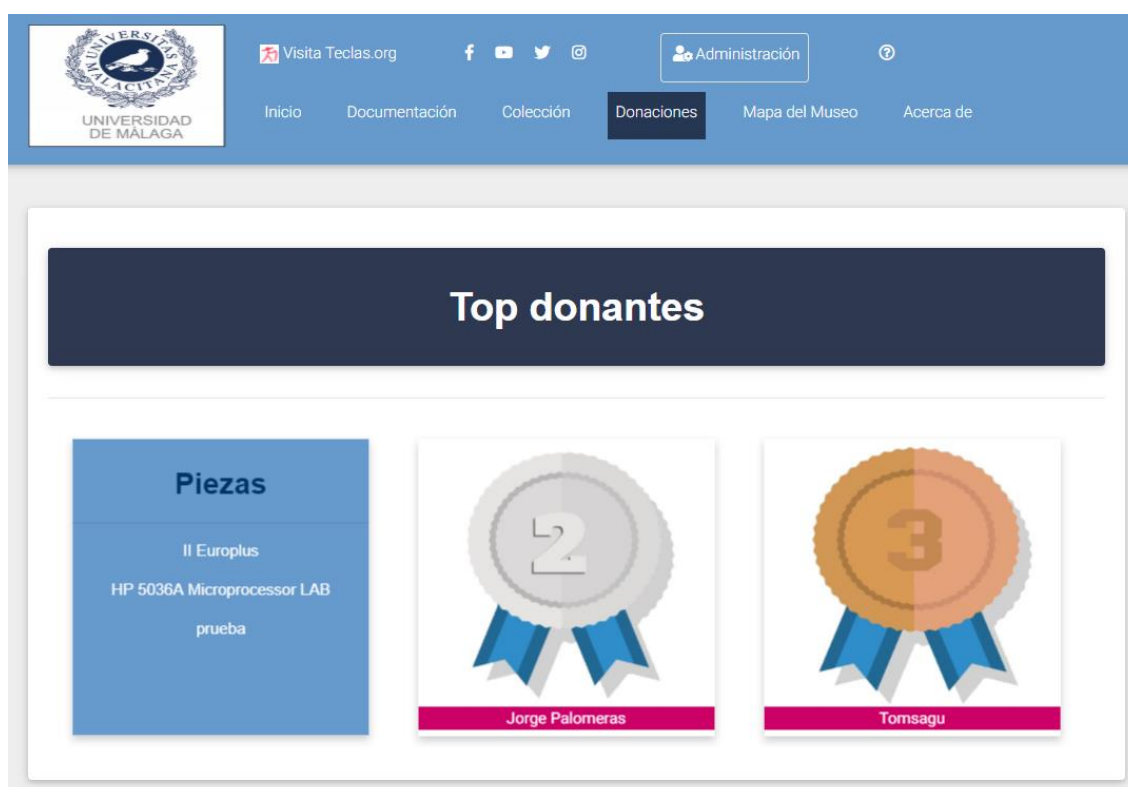


Figura 32. Vista Donaciones. Top Donantes volteado

2.4.2 Seguridad del servidor

Hasta este punto cualquier usuario podría realizar peticiones a nuestra base de datos si dispone de la dirección correcta. Esto significa que podrían hacer llamadas POST, DELETE, etc. y modificar la información almacenada. Para solucionar esto, se ha implementado un sistema de seguridad basado en JSON Web Tokens (JWT). De esta manera, el cliente envía los credenciales en una llamada POST a una dirección específica (/login), y el servidor responde con un token en su cabecera. Así, cada llamada que se realice tiene que incluir este token en su cabecera bajo la etiqueta *Authorization*.

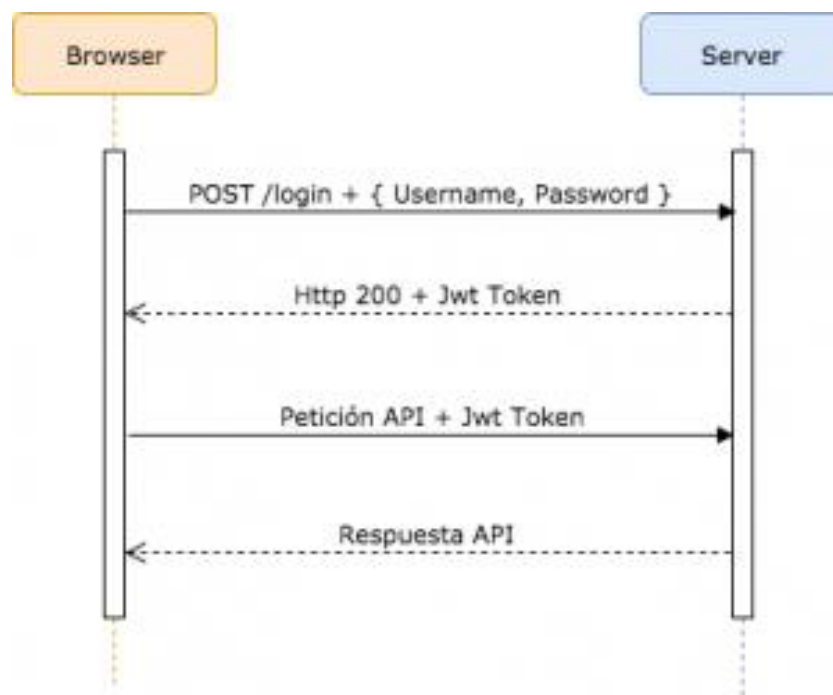


Figura 33. Diagrama de flujo de una aplicación securizada.

Para añadir más seguridad, las contraseñas han sido encriptadas y guardadas en la base de datos de esta manera:

```
@Bean
public BCryptPasswordEncoder passwordEncoder() {
    BCryptPasswordEncoder bCryptPasswordEncoder = new BCryptPasswordEncoder();
    return bCryptPasswordEncoder;
}
```

Figura 34. Fragmento de código del servidor. Encripta la contraseña.

Uno de los problemas más costoso de solucionar de todo el proyecto ha sido controlar el CORS del servidor sobre los navegadores. El Intercambio de Recursos de Origen Cruzado (CORS) es un mecanismo de seguridad que tienen los navegadores para acceder a recursos seleccionados desde un servidor con un dominio no conocido. Utiliza cabeceras HTTP adicionales que restringen las solicitudes externas. De este modo, el explorador nos bloqueaba todas las peticiones de nuestro servidor. Tras mucho esfuerzo e investigación, descubrimos que debíamos añadir unas cabeceras determinadas a todos los envíos desde nuestro servidor. En la siguiente figura podemos ver el fragmento de código que logra este propósito.

```
@Bean
CorsConfigurationSource corsConfigurationSource() {
    CorsConfiguration configuration = new CorsConfiguration();
    configuration.setAllowedOrigins(Arrays.asList("http://localhost:4200", "http://localhost:8080"));
    configuration.setAllowedMethods(Arrays.asList("GET", "POST", "OPTIONS", "DELETE", "PUT", "PATCH"));
    configuration.setAllowedHeaders(Arrays.asList("*", "Authorization"));
    configuration.setExposedHeaders(Arrays.asList("Authorization"));
    configuration.setAllowCredentials(true);
    UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
    source.registerCorsConfiguration( path: "**", configuration);
    return source;
}
```

Figura 35. Fragmento de código del servidor. Soluciona el problema del CORS.

De esta forma añadimos las etiquetas y los dominios que vamos a utilizar, de modo que el navegador las añada a su lista de direcciones de confianza y no las bloquee.

3

Conclusiones y líneas futuras

3.1 Conclusiones

Al finalizar este trabajo de varios meses de duración, es conveniente hacer una comparación entre los objetivos presentados en un principio y los resultados obtenidos.

En primer lugar, se ha conseguido proveer al museo de la ETSI Informática de un sitio web con funcionalidades para visualizar y modificar la información almacenada en una base de datos. Este era el objetivo principal y es nuestro mayor logro. Sin embargo, se han sufrido diferentes dificultades que han producido cambios en el planteamiento inicial.

Por un lado, se pretendió realizar una aplicación web totalmente adaptable a dispositivos móviles. Para ello intentamos realizar la web usando Ionic Framework, pero esta herramienta está orientada al desarrollo móvil por lo que decidimos cambiar a Angular que nos ofrecía componentes adaptados a webs de navegadores. A pesar de esto, nuestro sitio web es totalmente adaptable a diferentes tamaños de pantallas, incluidas tablets y dispositivos móviles.

Por otro lado, se pensaron diseños como el de un cronograma donde se viesen las piezas por antigüedad. Esta idea se sustituyó por la de un boceto de las diferentes salas de la Escuela, el cual se puede hacer clic en cada una de ellas mostrando las piezas que se encuentran en esa sala.

Por último, mencionar a mi tutor del TFG Eduardo Guzmán por ofrecernos esta línea de trabajo y proponernos la idea de realizar esta página. También, agradecer a todos los compañeros que me han ayudado tanto con el desarrollo como con la redacción de la memoria. Y para finalizar, a mis compañeros de TFG por formar parte de este proyecto y por conformar un gran equipo.

3.2 Líneas futuras

Tras hacer una reflexión y un repaso al proyecto, es conveniente hacer una autocrítica y exponer las ideas y posibles mejoras que pueden enriquecer este trabajo.

1. Usar un WYSIWYG para la creación/edición personalizada de un artículo: "*What you see if what you get*" es una herramienta que provee un editor de texto. Así, se podrá crear artículos muchos más personalizados.

2. Crear una adaptación para dispositivos móviles: es cierto que nuestra aplicación dispone de un diseño adaptable a este tipo de pantallas. Aún así, se puede mejorar usando componentes preparados para móviles empleando, por ejemplo, la herramienta *Ionic Framework*.

3. Añadir imágenes desde tu equipo: permitir subir fotos desde el explorador de archivos de tu ordenador. Estas fotos se deberían almacenar además en un servidor de imágenes.

4. Caducar la sesión iniciada por tiempo: como se ha detallado en la memoria, la sesión del usuario administrador se almacena en el navegador. Ésta debería caducar por inactividad, pidiéndole al cliente que vuelva a introducir sus credenciales. De esta manera, se aumentaría la seguridad de la aplicación web.

5. Añadir internalización: para hacer que la página sea visitable por un mayor número de usuarios, es necesario permitir al visitante elegir el idioma de visionado.

Referencias

- [1] Pivotal Software (2019). Spring Official Web. Building a RESTful Web Service. <https://spring.io/guides/gs/rest-service/>
- [2] Pivotal Software (2019). Spring Official Web. Accessing MongoDB Data with REST. <https://spring.io/guides/gs/accessing-mongodb-data-rest/>
- [3] 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>
- [4] 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/>
- [5] Innova Engineers (2018, January 13). Introduction to Spring Data MongoDB Cloud. <https://www.youtube.com/watch?v=FbIAs8278h4>
- [6] Innova Engineers (2018, April 15). Spring Boot + Spring Data MongoDB How to perform CRUD Operations. <https://www.youtube.com/watch?v=A5NWMpp4H7s>
- [7] 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>
- [8] MuleSoft (2019). Advanced Rest Client. <https://install.advancedrestclient.com/install>
- [9] JetBrains s.r.o. (2019, February 1). Creating a Remote Server Configuration. <https://www.jetbrains.com/help/idea/creating-a-remote-server-configuration.html>
- [10] JetBrainsTV (2013, June 13). Deployment and Remote Hosts in PhpStorm - PhpStorm Video Tutorial. <https://www.youtube.com/watch?v=AHK20LWEWXQ>
- [11] 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/>

[12] 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>

[13] Angular Official Web (2010-2019). Angular - Getting Started.

<https://angular.io/guide/quickstart>

[14] Web oficial Universidad de Málaga (2019, June 20). Repositorio de documento.

https://www.uma.es/etsi-informatica/navegador_de_ficheros/Informatica/listado/Logos/

[15] GitHub (2019, June 26). Repositorio del servidor del proyecto.

<https://github.com/tomsagu/museumServer>

[16] GitHub (2019, June 26). Repositorio del cliente del proyecto.

<https://github.com/tomsagu/museumClient>

Apéndice A

Manual de Instalación

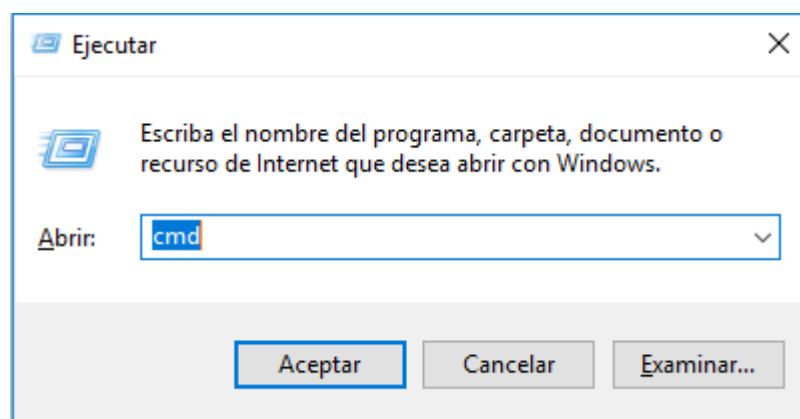
1. Requerimientos:

Para poder ejecutar la aplicación web en un equipo local es necesario tener en cuenta varios puntos:

1.1 Compilación y despliegue del servidor

Partiendo del proyecto descomprimido que podemos encontrar en el CD asociado a esta memoria, es necesario compilar el servidor. Para ello:

1. Abrir un símbolo de sistema: Atajo de teclado Windows + R, escribir “cmd” y pulsar Aceptar.



2. Navegar hasta la dirección donde tiene los ficheros del servidor con el comando cd.

3. Ejecutar desde consola el comando `gradlew build` para compilar.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.17134.829]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumServer>gradlew build
Starting a Gradle Daemon (subsequent builds will be faster)

BUILD SUCCESSFUL in 10s
3 actionable tasks: 1 executed, 2 up-to-date
C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumServer>
```

4. Navegar hasta la carpeta libs donde se ha creado el jar del servidor:
`cd build/libs` y ejecutar el comando `java -jar <archivo.jar>` para desplegar el servidor de forma local.

```
C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumServer>cd build
C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumServer\build>cd libs
C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumServer\build\libs>java -jar museumServer-0.0.1-SNAPSHOT.jar
```

```
C:\Windows\System32\cmd.exe - java -jar museumServer-0.0.1-SNAPSHOT.jar
WireVersion=7, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=46403556, setName='Cluster0-shard-0', canonicalAddress=cluster0-shard-00-02-vkdib.mongodb.net:27017, hosts=[cluster0-shard-00-02-vkdib.mongodb.net:27017, cluster0-shard-00-01-vkdib.mongodb.net:27017, cluster0-shard-00-00-vkdib.mongodb.net:27017], passives=[], arbiters=[], primary='cluster0-shard-00-00-vkdib.mongodb.net:27017', tagSet=TagSet{[Tag{name='region', value='EU_CENTRAL_1'}, Tag{name='provider', value='AWS'}, Tag{name='nodeType', value='ELECTABLE'}]}, electionId=null, setVersion=1, lastWriteDate=Thu Jun 20 19:49:52 CEST 2019, lastUpdateTimeNanos=269417851170765}
2019-06-20 19:49:51.885 INFO 11520 --- [ngodb.net:27017] org.mongodb.driver.cluster : Setting max election id to 7fffffff0000000000000007 from replica set primary cluster0-shard-00-00-vkdib.mongodb.net:27017
2019-06-20 19:49:51.887 INFO 11520 --- [ngodb.net:27017] org.mongodb.driver.cluster : Setting max set version to 1 from replica set primary cluster0-shard-00-00-vkdib.mongodb.net:27017
2019-06-20 19:49:51.887 INFO 11520 --- [ngodb.net:27017] org.mongodb.driver.cluster : Discovered replica set primary cluster0-shard-00-00-vkdib.mongodb.net:27017
2019-06-20 19:49:52.798 INFO 11520 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter@2dc9b0f5, org.springframework.security.web.context.SecurityContextPersistenceFilter@163370c2, org.springframework.security.web.header.HeaderWriterFilter@13df2a8c, org.springframework.web.filter.CorsFilter@6531a794, org.springframework.security.web.authentication.logout.LogoutFilter@c667f46, com.museum.museumServer.main.security.LoginFilter@3b5fad2d, com.museum.museumServer.main.security.JwtFilter@5e17553a, org.springframework.security.web.savedrequest.RequestCacheAwareFilter@7905a0b8, org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter@13d9cbf5, org.springframework.security.web.authentication.AnonymousAuthenticationFilter@3eb91815, org.springframework.security.web.session.SessionManagementFilter@72d6b3ba, org.springframework.security.web.access.ExceptionTranslationFilter@1169afef, org.springframework.security.web.access.intercept.FilterSecurityInterceptor@a1f72f5]
2019-06-20 19:49:53.127 INFO 11520 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2019-06-20 19:49:53.725 INFO 11520 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2019-06-20 19:49:53.728 INFO 11520 --- [main] c.m.m.main.MuseumServerApplication : Started MuseumServerApplication in 8.334 seconds (JVM running for 8.942)
Server is running!
```

1.2 Cliente

Este proyecto usa la versión de Angular CLI 7.3.8.

Para poder hacer uso de los comandos de Angular, es necesario tener instalado previamente Node.js (<https://nodejs.org/es/download/>).

1. Instalar Angular CLI en su equipo: `npm i -g @angular/cli@latest`
2. Acceder a la carpeta contenedora del proyecto cliente desde consola y descargar las dependencias de los componentes: `npm install` en la carpeta principal.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.17134.829]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumClient>npm install
npm WARN @angular/animations@7.2.12 requires a peer of @angular/core@7.2.12 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/common@7.2.12 requires a peer of @angular/core@7.2.12 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/forms@7.2.12 requires a peer of @angular/core@7.2.12 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/http@7.2.14 requires a peer of @angular/core@7.2.14 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/http@7.2.14 requires a peer of @angular/platform-browser@7.2.14 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/platform-browser@7.2.12 requires a peer of @angular/core@7.2.12 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/platform-browser-dynamic@7.2.12 requires a peer of @angular/core@7.2.12 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/router@7.2.14 requires a peer of @angular/core@7.2.14 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/router@7.2.14 requires a peer of @angular/common@7.2.14 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/router@7.2.14 requires a peer of @angular/platform-browser@7.2.14 but none is installed. You must install peer dependencies yourself.
npm WARN bootstrap@4.3.1 requires a peer of popper.js^1.14.7 but none is installed. You must install peer dependencies yourself.
npm WARN ngx-image-gallery@1.3.1 requires a peer of @angular/core@^4.0.0 but none is installed. You must install peer dependencies yourself.
npm WARN ngx-image-gallery@1.3.1 requires a peer of rxjs@^5.1.0 but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 12 packages from 3 contributors and audited 42653 packages in 17.711s
found 0 vulnerabilities

C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumClient>
```

3. Compilar y desplegar el cliente: ejecutar el comando `ng serve` y navegar a la dirección <http://localhost:4200> en su navegador.

```
C:\Users\Tomas\Desktop\UMA\4Software\TFG_personal\museumClient>ng serve
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

Date: 2019-06-20T17:52:26.964Z
Hash: 3bb8b75e6776cddc5bd3
Time: 18486ms
chunk {es2015-polyfills} es2015-polyfills.js, es2015-polyfills.js.map (es2015-polyfills) 284 kB [initial] [rendered]
chunk {main} main.js, main.js.map (main) 496 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 236 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {scripts} scripts.js, scripts.js.map (scripts) 86.1 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 200 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 8.29 MB [initial] [rendered]
i @wdm@: Compiled successfully.
```


Apéndice B

Manual de Usuario

Requerimientos:

1. Navegación por la web

Al acceder a la web, comenzaremos en la página [Inicio](#). En la parte superior siempre encontraremos una cabecera con links 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

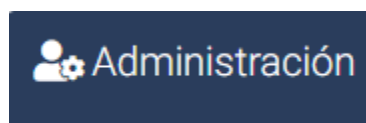
En la ventana “acerca de” aparece información de dónde visitar este museo, así como información sobre la motivación de realizar la misma y agradecimientos.

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:



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>
	

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 Páginas: **09**

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

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

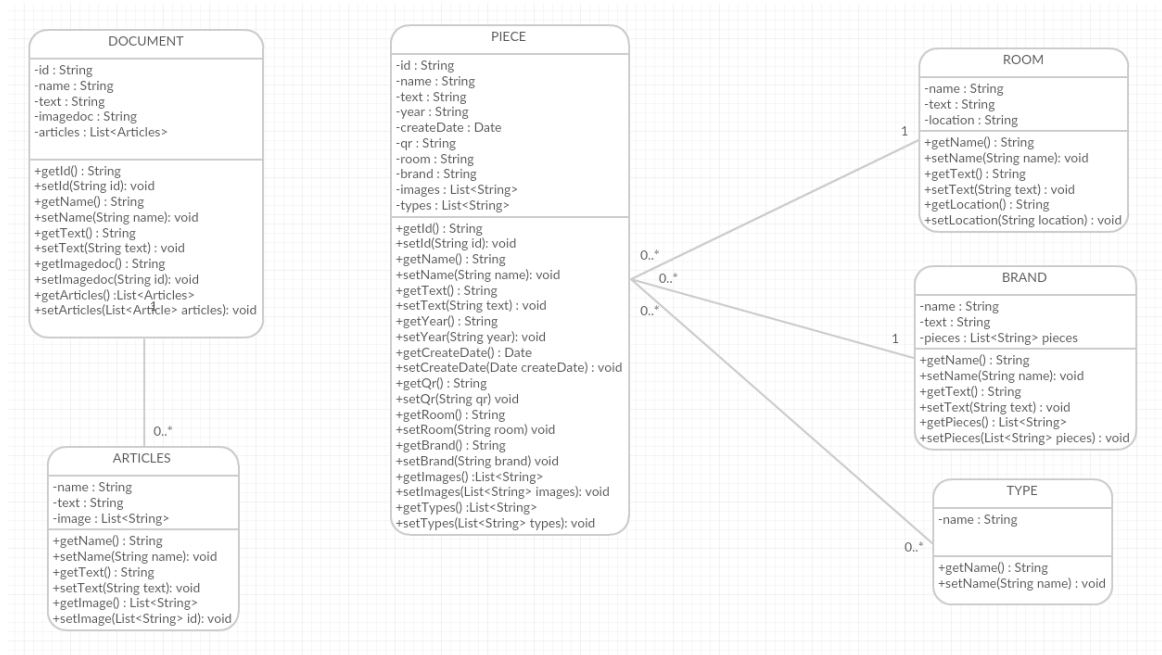
2.1. MANUAL DE USUARIO

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.

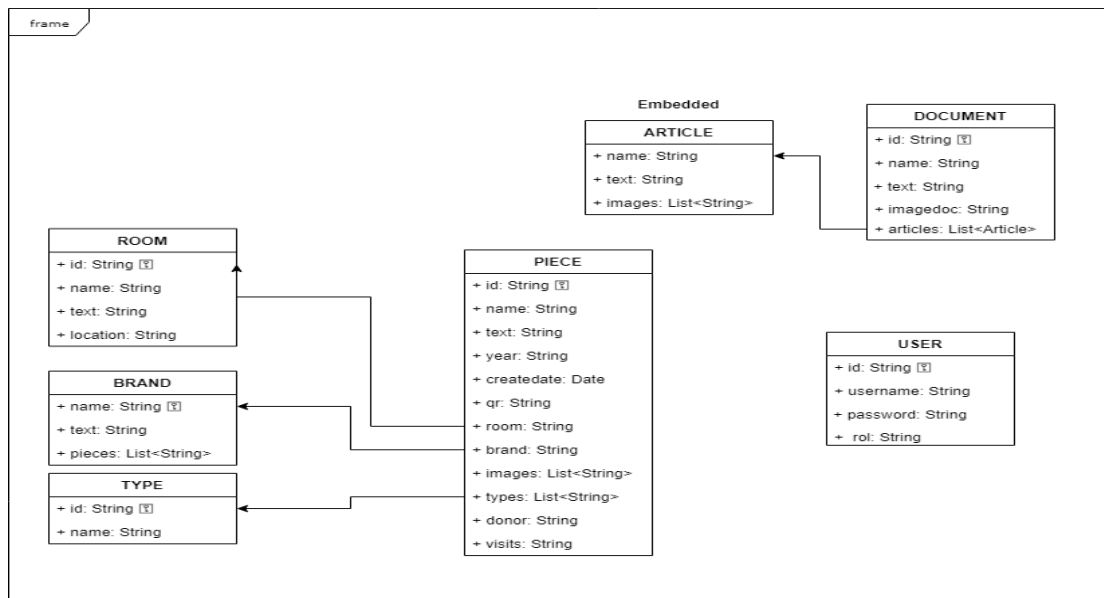
Más específicamente, se podrá encontrar información para gestionar la información de la base de datos a través de una herramienta con interfaz gráfica como es la misma aplicación web.

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

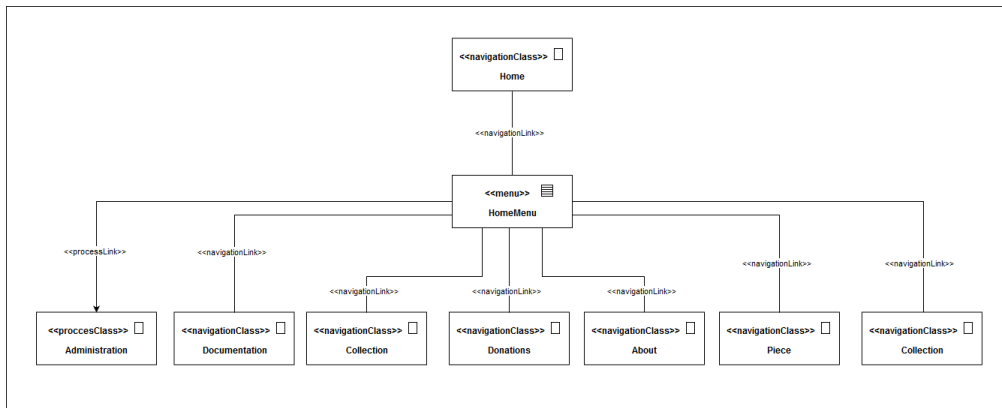


2.4. MODELOS DE NAVEGACIÓN

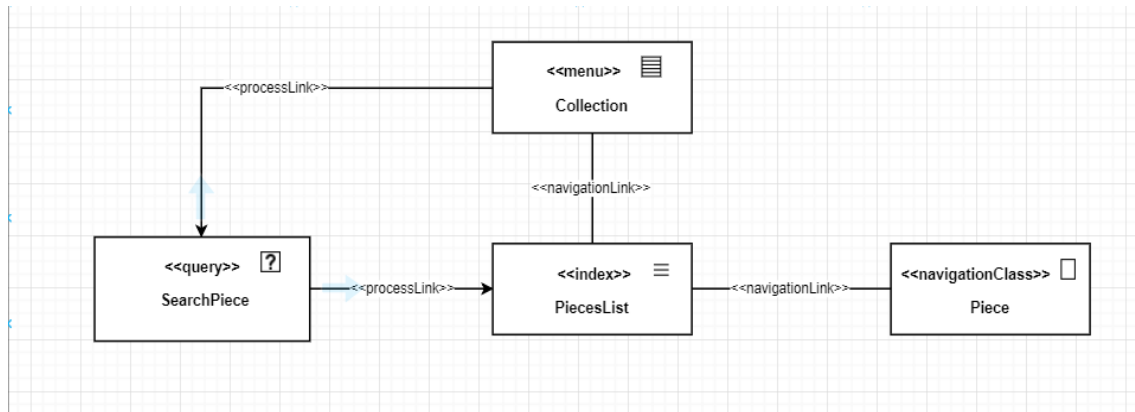
- HOME:

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.

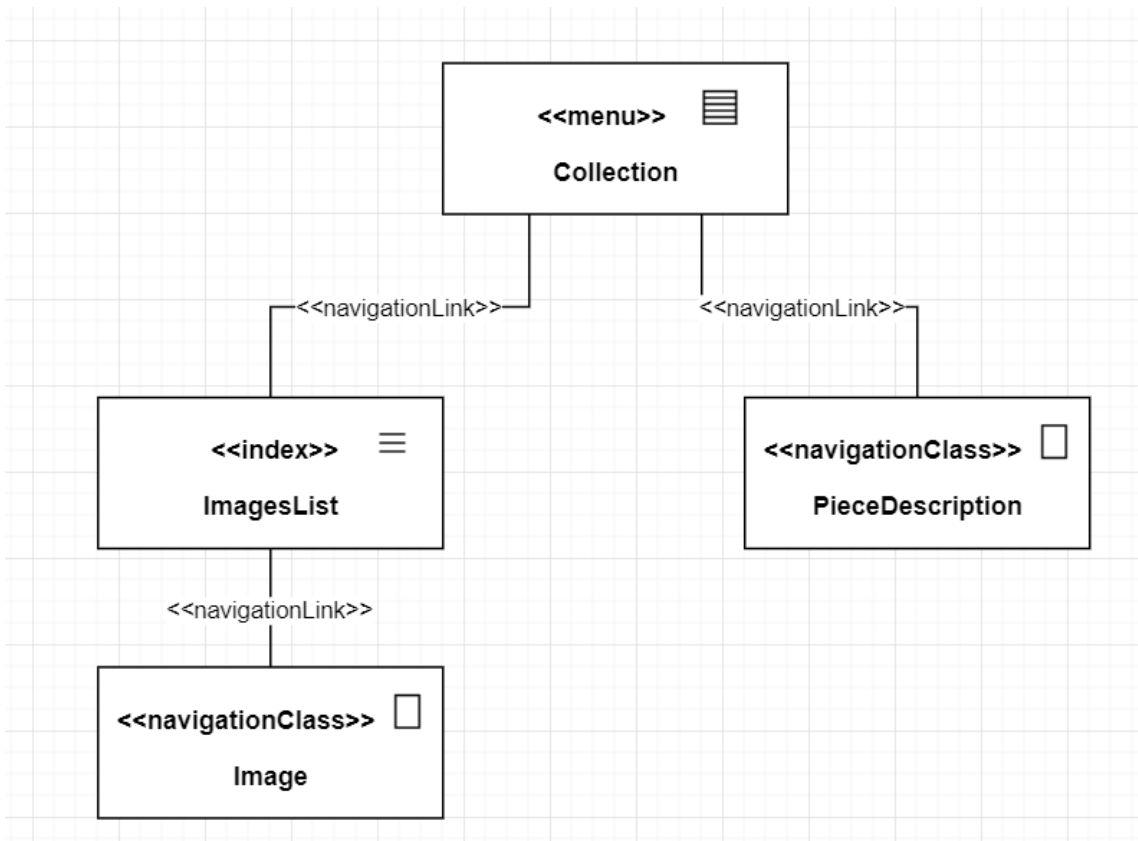
Navegación Home



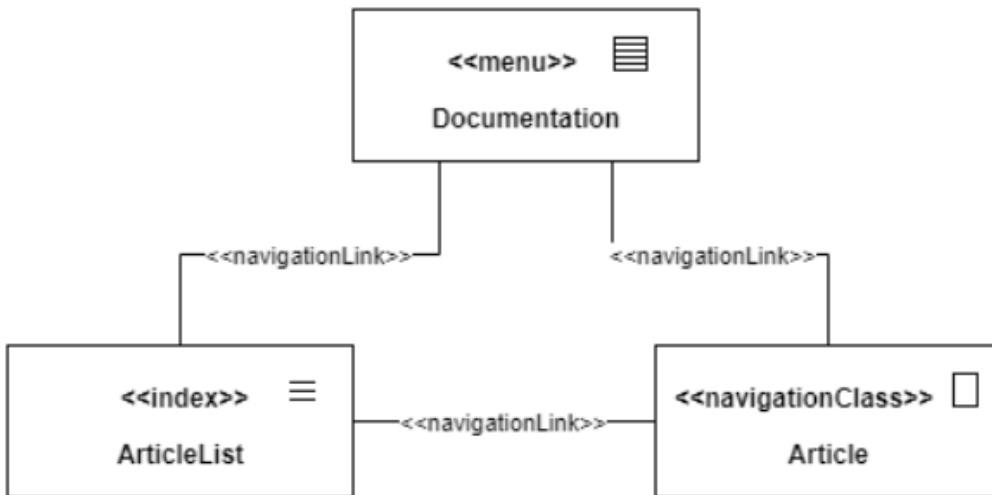
• **COLECCIÓN:**



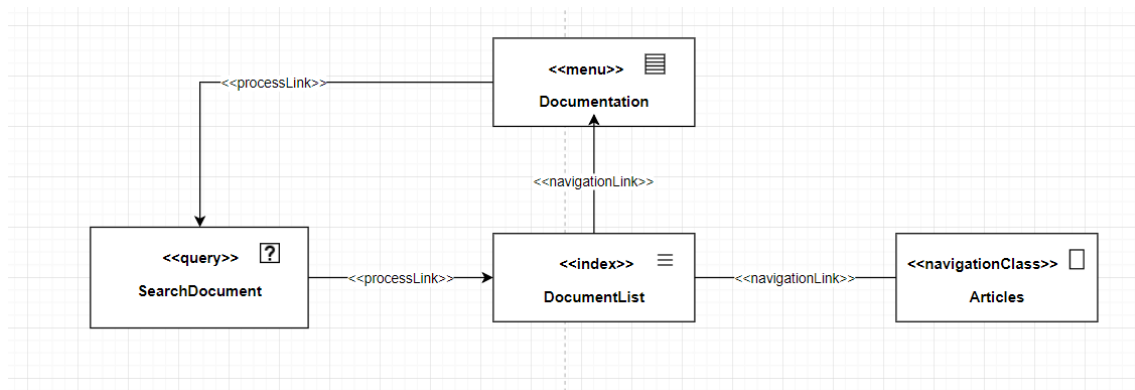
- **PIEZA:**



- **ARTÍCULO:**



- **DOCUMENTACIÓN:**



3. 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

Pruebas de Sistema

- Iteración 1 - Inicio

1. Contenido

Pruebas del sistema para el desarrollo de la Iteración 1.

2. Control de versión

2.1. Control de cambios

<i>Fecha</i>	<i>Autor</i>	<i>Versión</i>	<i>Comentario</i>
<i>29/04/2019</i>	<i>Tomás Sayago Gutiérrez</i>	<i>1.0</i>	<i>Versión inicial, estructura de cada escenario</i>

3. Pruebas de sistema

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

Para este test, no hay ningún usuario con la 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 Documentación.

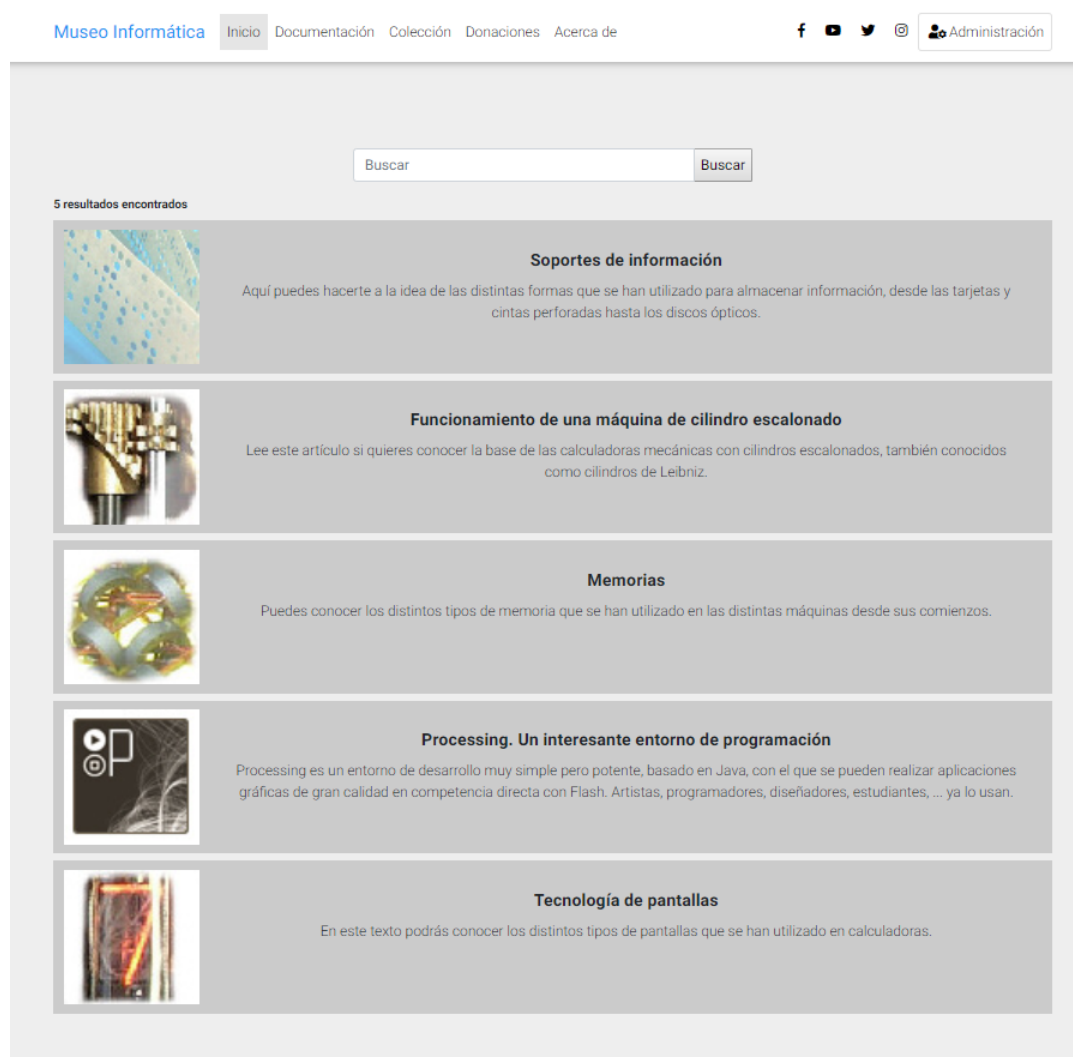
Realizar una búsqueda, apreciar cómo la información mostrada en pantalla se actualiza al igual que el número de resultados encontrados.

3.1.2. Resultado

Passed.






3.1.3. Evidencias

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




The screenshot shows the website interface for Museo Informática. At the top, there is a navigation bar with the site name and menu items: Inicio, Documentación, Colección, Donaciones, and Acerca de. On the right side of the navigation bar, there are social media icons for Facebook, YouTube, Twitter, and Instagram, along with an 'Administración' button. Below the navigation bar is a search bar with the placeholder text 'Buscar' and a 'Buscar' button. The search results section displays '5 resultados encontrados'. The first result is titled 'Soportes de información' and includes a thumbnail image of a blue patterned surface. The second result is 'Funcionamiento de una máquina de cilindro escalonado' with a thumbnail of a mechanical device. The third result is 'Memorias' with a thumbnail of a colorful, abstract image. The fourth result is 'Processing. Un interesante entorno de programación' with a thumbnail showing the letters 'p' and 'P' on a dark background. The fifth result is 'Tecnología de pantallas' with a thumbnail of a glowing screen. Each result card contains a brief description of the topic.

2. Insertamos una palabra en el buscador.

Museo Informática Inicio Documentación Colección Donaciones Acerca de      Administración


calculadoras

5 resultados encontrados




Soportes de información

Aquí puedes hacerte a la idea de las distintas formas que se han utilizado para almacenar información, desde las tarjetas y cintas perforadas hasta los discos ópticos.




Funcionamiento de una máquina de cilindro escalonado

Lee este artículo si quieres conocer la base de las calculadoras mecánicas con cilindros escalonados, también conocidos como cilindros de Leibniz.



Memorias






Puedes conocer los distintos tipos de memoria que se han utilizado en las distintas máquinas desde sus comienzos.



Processing. Un interesante entorno de programación


Processing es un entorno de desarrollo muy simple pero potente, basado en Java, con el que se pueden realizar aplicaciones gráficas de gran calidad en competencia directa con Flash. Artistas, programadores, diseñadores, estudiantes, ... ya lo usan.

3. Seleccionamos el botón *Buscar*.

Museo Informática Inicio Documentación Colección Donaciones Acerca de      Administración


calculadoras

2 resultados encontrados



Funcionamiento de una máquina de cilindro escalonado

Lee este artículo si quieres conocer la base de las calculadoras mecánicas con cilindros escalonados, también conocidos como cilindros de Leibniz.



Tecnología de pantallas

En este texto podrás conocer los distintos tipos de pantallas que se han utilizado en calculadoras.

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 Documentación.

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 documentos aparece.

3.2.2. Resultado

Passed.

3.2.3. Evidencias

- 1. Seleccionamos la pestaña Documentación en la barra de navegación.*

3 resultados encontrados



Soportes de información

Aquí puedes hacerte a la idea de las distintas formas que se han utilizado para almacenar información, desde las tarjetas y cintas perforadas hasta los discos ópticos.



Funcionamiento de una máquina de cilindro escalonado

Lee este artículo si quieres conocer la base de las calculadoras mecánicas con cilindros escalonados, también conocidos como cilindros de Leibniz.



Memorias

Puedes conocer los distintos tipos de memoria que se han utilizado en las distintas máquinas desde sus comienzos.



2. Insertamos una palabra en el buscador.

3 resultados encontrados



Soportes de información

Aquí puedes hacerte a la idea de las distintas formas que se han utilizado para almacenar información, desde las tarjetas y cintas perforadas hasta los discos ópticos.



Funcionamiento de una máquina de cilindro escalonado

Lee este artículo si quieres conocer la base de las calculadoras mecánicas con cilindros escalonados, también conocidos como cilindros de Leibniz.

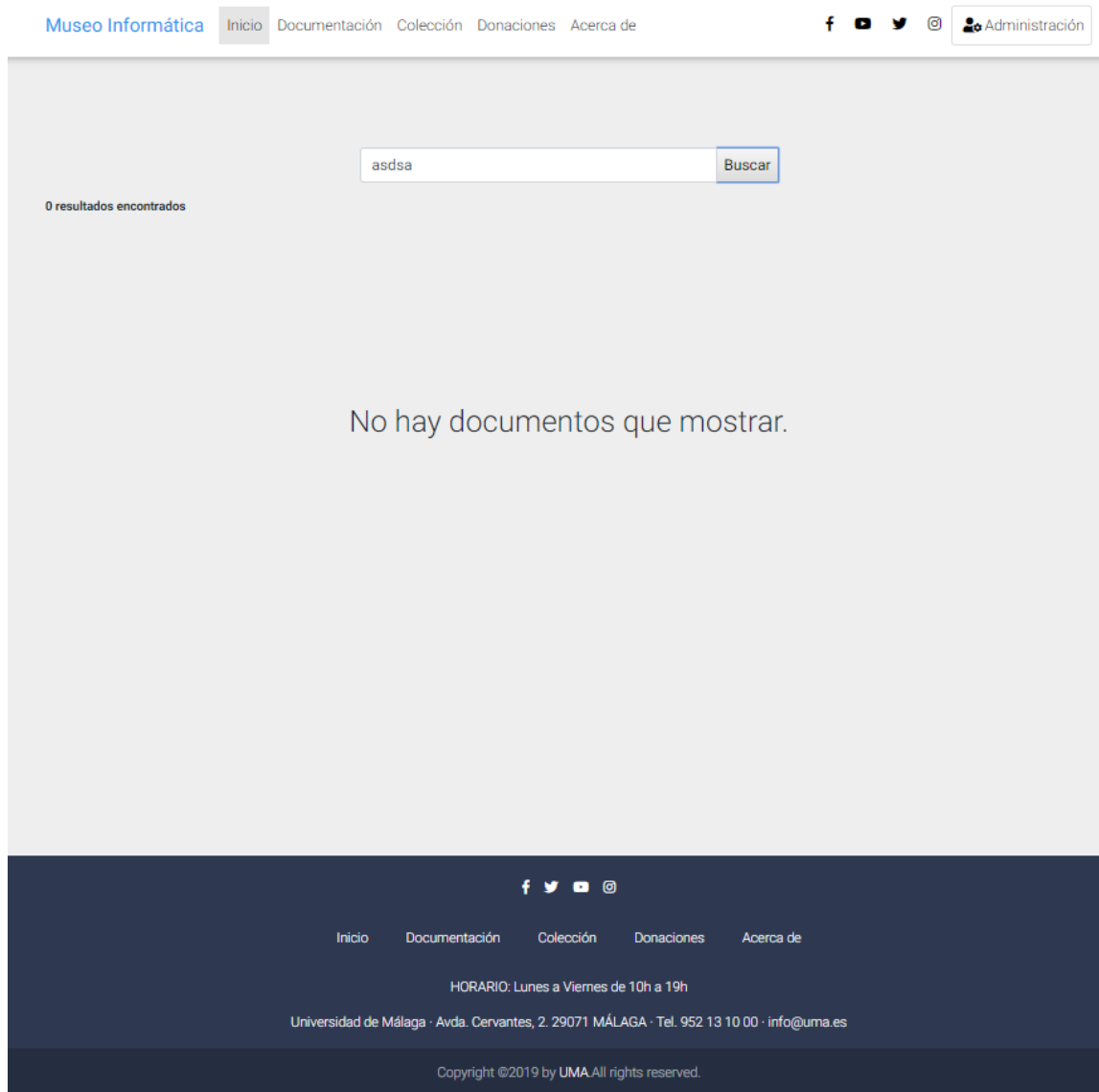


Memorias

Puedes conocer los distintos tipos de memoria que se han utilizado en las distintas máquinas desde sus comienzos.



*3. Seleccionamos el botón **Buscar**.*



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 Documentación y Artículo.

Seleccionar un documento y acceder a los artículos de dicho documento.

Navegar entre los diferentes artículo usando el menú lateral o la paginación.

Seleccionar una foto de la galería y navegar entre ellas.

3.3.2. Resultado

Passed.

3.3.3. Evidencias

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

The screenshot shows the website interface for Museo Informática. At the top, there is a navigation bar with the logo 'Museo Informática' and links for 'Inicio', 'Documentación', 'Colección', 'Donaciones', and 'Acerca de'. On the right side of the navigation bar, there are social media icons for Facebook, YouTube, Twitter, and Instagram, along with a button for 'Administración'. Below the navigation bar is a search bar with the text 'Buscar' and a 'Buscar' button. The search results section shows '3 resultados encontrados'. The first result is titled 'Soportes de información' and includes a thumbnail image of a blue and white patterned surface. The second result is titled 'Funcionamiento de una máquina de cilindro escalonado' and includes a thumbnail image of a mechanical cylinder. The third result is titled 'Memorias' and includes a thumbnail image of a colorful, abstract pattern. The footer of the website contains social media icons, the same navigation links as the top bar, the opening hours 'HORARIO: Lunes a Viernes de 10h a 19h', the contact information 'Universidad de Málaga · Avda. Cervantes, 2. 29071 MÁLAGA · Tel. 952 13 10 00 · info@uma.es', and the copyright notice 'Copyright ©2019 by UMA. All rights reserved.'

2. Seleccionamos un documento.

Documento: Memorias

Memorias en calculadoras mecánicas

Las calculadoras mecánicas también tienen registros en los que se almacenan números y resultados. No tienen, claro está, memorias del estilo de los ordenadores, en las que se almacenan tanto datos como programas. Las memorias (o registros) de las calculadoras mecánicas son muy variados.

Previous Next

Artículos

[Memorias en calculadoras mecánicas](#)

Memoria de Tambor

Memorias de Línea de retardo

Galería de imágenes



3. Navegamos al siguiente documento usando el botón Next.

Documento: Memorias

Memoria de Tambor

No, no tenemos ninguna en el museo. Imaginad un tambor con superficie magnética girando de 3600 a 10000 revoluciones por minuto. Imaginad ahora un conjunto de cabezales similares a los de una unidad de disco, un cabezal por cada pista (o dos cabezales). En los comienzos, se fabricaron ordenadores que como memoria principal tenían este tipo de dispositivos. Bueno, si lo comparamos con los discos, nos damos cuenta que cuantas más pistas, más cabezales tiene el tambor, ya que éstos son fijos.

[Previous](#) [Next](#)

Artículos

Memorias en calculadoras mecánicas

[Memoria de Tambor](#)

Memorias de Línea de retardo



4. Navegamos a un documento usando el menú lateral.

Documento: Memorias

Memorias de Línea de retardo

Las líneas de retardo son dispositivos que almacenan información durante un corto periodo de tiempo. Mejor poner un ejemplo: ¿nunca habéis golpeado una varilla metálica, o una varandilla?, ésta vibra durante un tiempo hasta que deja de hacerlo. Pues esa vibración es la información que se almacena. Claro, que esta vibración se va atenuando, se va degradando, por lo que es necesario amplificarla y realimentarla. Si se quiere almacenar información, se aplican conjuntos de impulsos a un material que vibra. Por el otro extremo del material se leen dichos impulsos y además, se realimentan para mantener dicha información. Líneas de retardo de mercurio El material que vibra es una varilla de mercurio encerrada en un tubo Líneas de retardo de cuarzo El material que vibra es un cristal de cuarzo. Líneas de retardo de este tipo utilizan hoy en día los televisores para sincronizar la imagen con el sonido. Líneas de retardo acústicas :)) De esas si tenemos. El elemento que vibra es un alambre metálico.

Previous Next

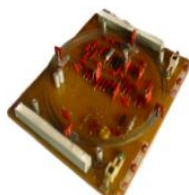
Artículos

Memorias en calculadoras mecánicas

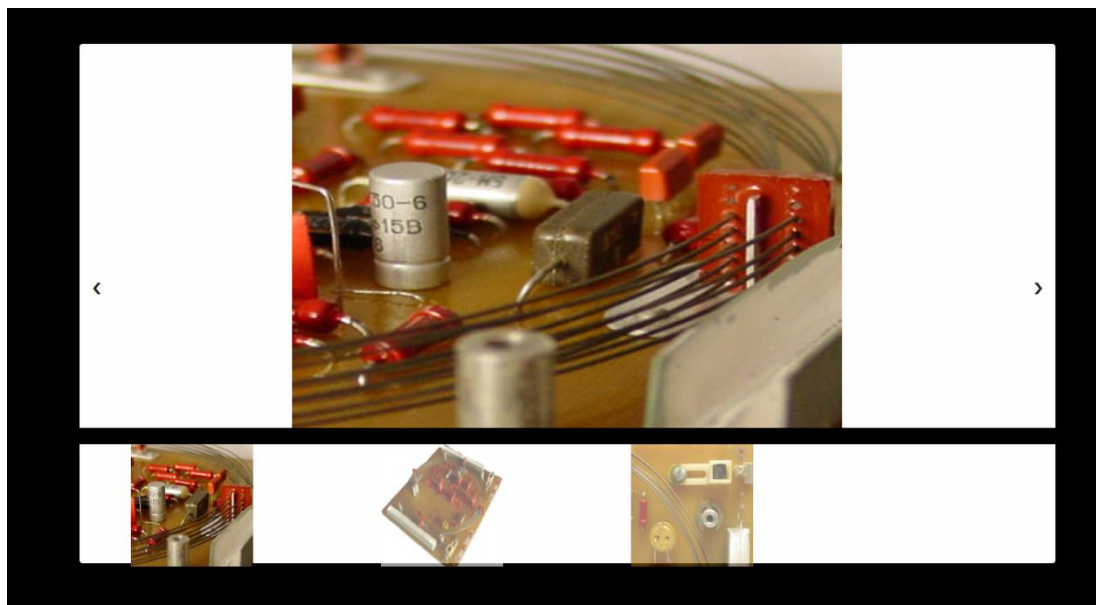
Memoria de Tambor

[Memorias de Línea de retardo](#)

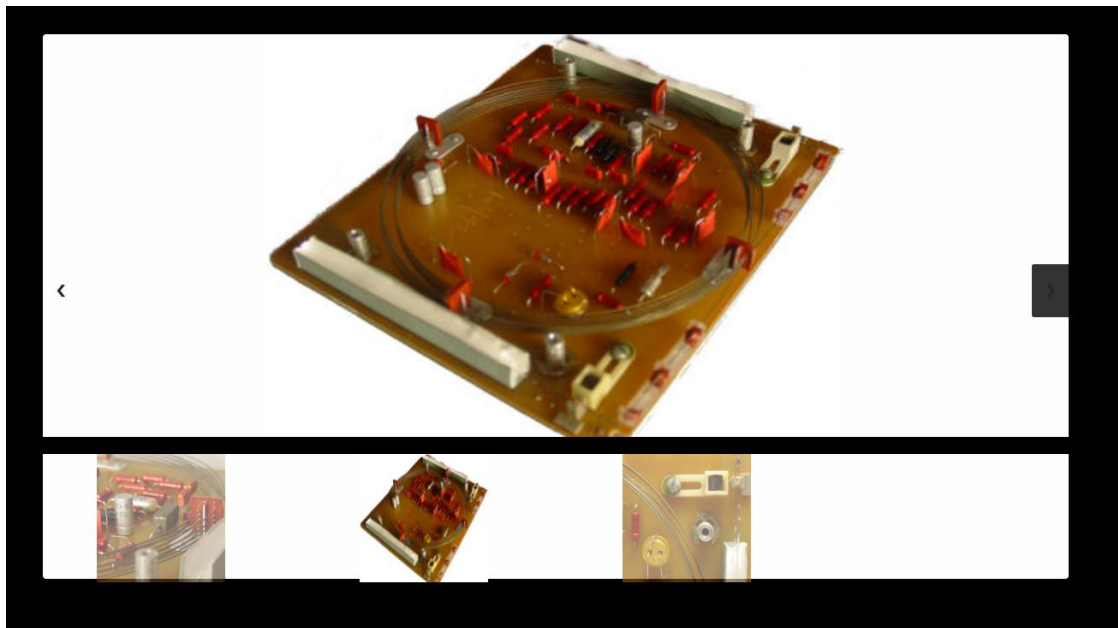
Galería de imágenes



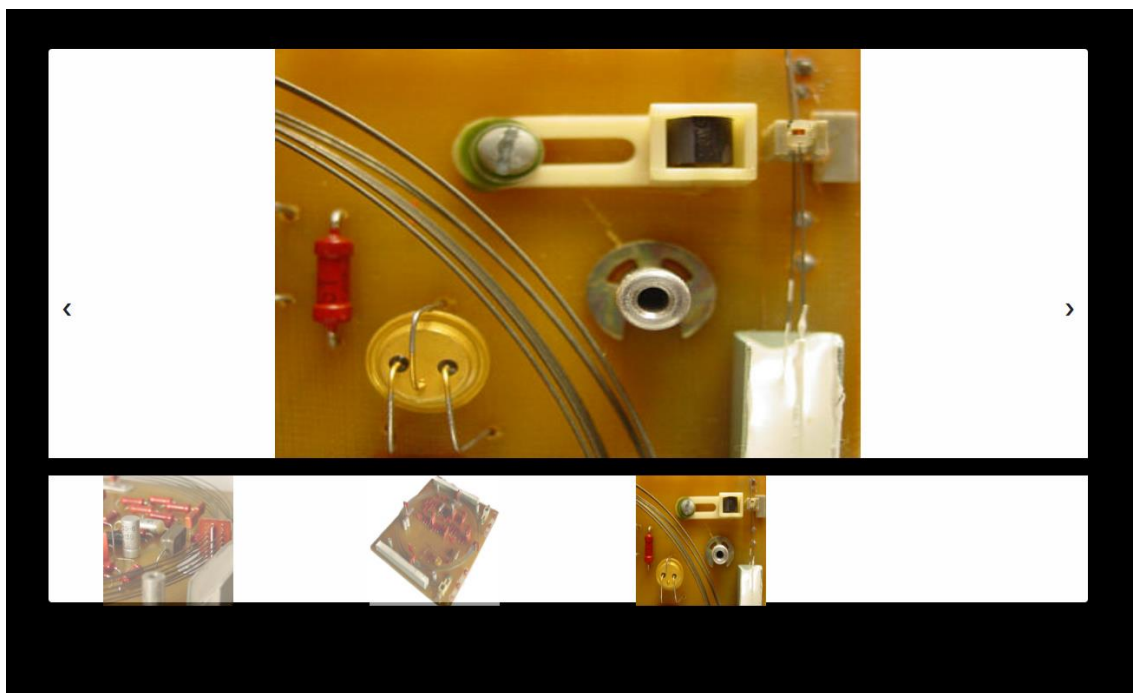
5. Seleccionamos una imagen de la galería.



6. Navegamos a la siguiente imagen usando la flecha derecha.



7. Navegamos a una de las imágenes seleccionándola en el menú inferior.



8. Cerramos la galería.

Documento: Memorias

Memorias de Línea de retardo

Las líneas de retardo son dispositivos que almacenan información durante un corto periodo de tiempo. Mejor poner un ejemplo: ¿nunca habéis golpeado una varilla metálica, o una varandilla?, ésta vibra durante un tiempo hasta que deja de hacerlo. Pues esa vibración es la información que se almacena. Claro, que esta vibración se va atenuando, se va degradando, por lo que es necesario amplificarla y realimentarla. Si se quiere almacenar información, se aplican conjuntos de impulsos a un material que vibra. Por el otro extremo del material se leen dichos impulsos y además, se realimentan para mantener dicha información. Líneas de retardo de mercurio El material que vibra es una varilla de mercurio encerrada en un tubo Líneas de retardo de cuarzo El material que vibra es un cristal de cuarzo. Líneas de retardo de este tipo utilizan hoy en día los televisores para sincronizar la imagen con el sonido. Líneas de retardo acústicas :-)) De esas si tenemos. El elemento que vibra es un alambre metálico.

Previous Next

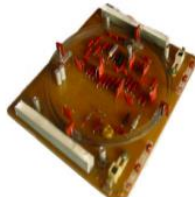
Artículos

Memorias en calculadoras mecánicas

Memoria de Tambor

[Memorias de Línea de retardo](#)

Galería de imágenes



9. Volvemos a la pestaña documentación clicando en el hipervínculo “Documento”.

3 resultados encontrados



Soportes de información

Aquí puedes hacerte a la idea de las distintas formas que se han utilizado para almacenar información, desde las tarjetas y cintas perforadas hasta los discos ópticos.



Funcionamiento de una máquina de cilindro escalonado

Lee este artículo si quieres conocer la base de las calculadoras mecánicas con cilindros escalonados, también conocidos como cilindros de Leibniz.



Memorias

Puedes conocer los distintos tipos de memoria que se han utilizado en las distintas máquinas desde sus comienzos.



3.3.4. Base de datos

N/A

3.3.5. Ficheros

N/A

Apéndice E

Pruebas de Sistema

- Iteración 2 -

Administración

1. Contenidos

Pruebas del sistema para el desarrollo de la Iteración 2.

2. Control de versión

2.1. Control de cambios

<i>Fecha</i>	<i>Autor</i>	<i>Versión</i>	<i>Comentario</i>
<i>23/05/2019</i>	<i>Tomás Sayago Gutiérrez</i>	<i>1.0</i>	<i>Versión inicial, estructura de cada escenario</i>

3. Pruebas de sistema

Estas son las pruebas del sistema para las ventanas Inicio sesión y Documentación CRUD (zona de administración de documentos).

Para estos tests, 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 Login.

Iniciar sesión introduciendo correctamente los credenciales de administrador.

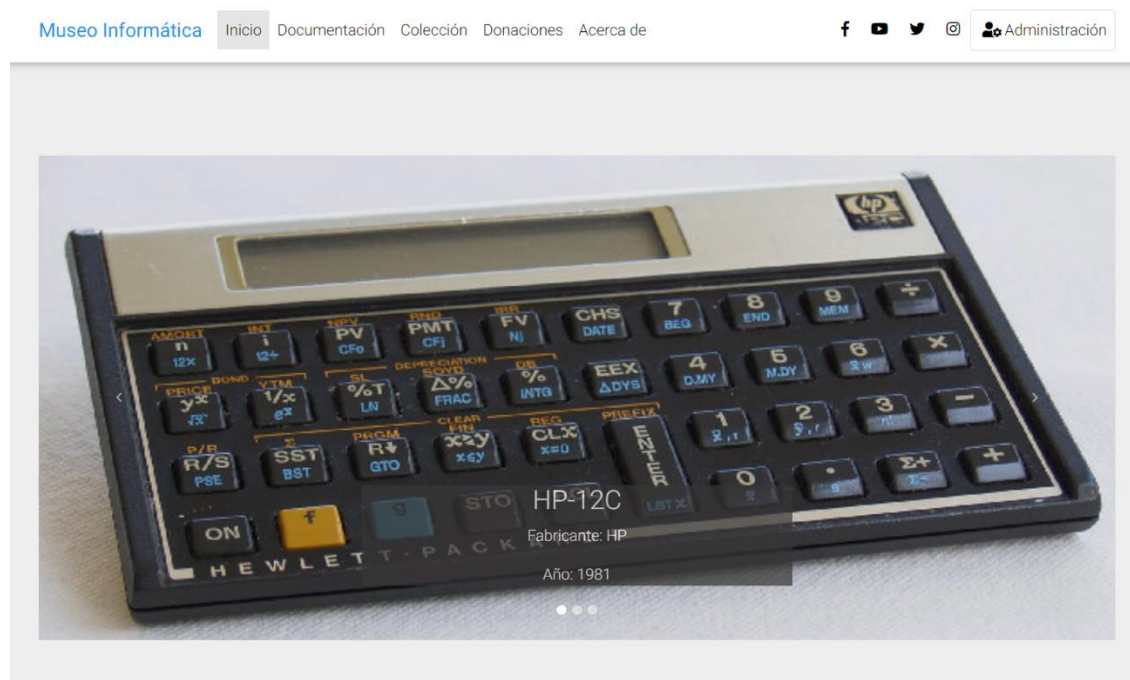
Accederemos a la zona restringida para administradores.

Finalmente, se cerrará sesión.

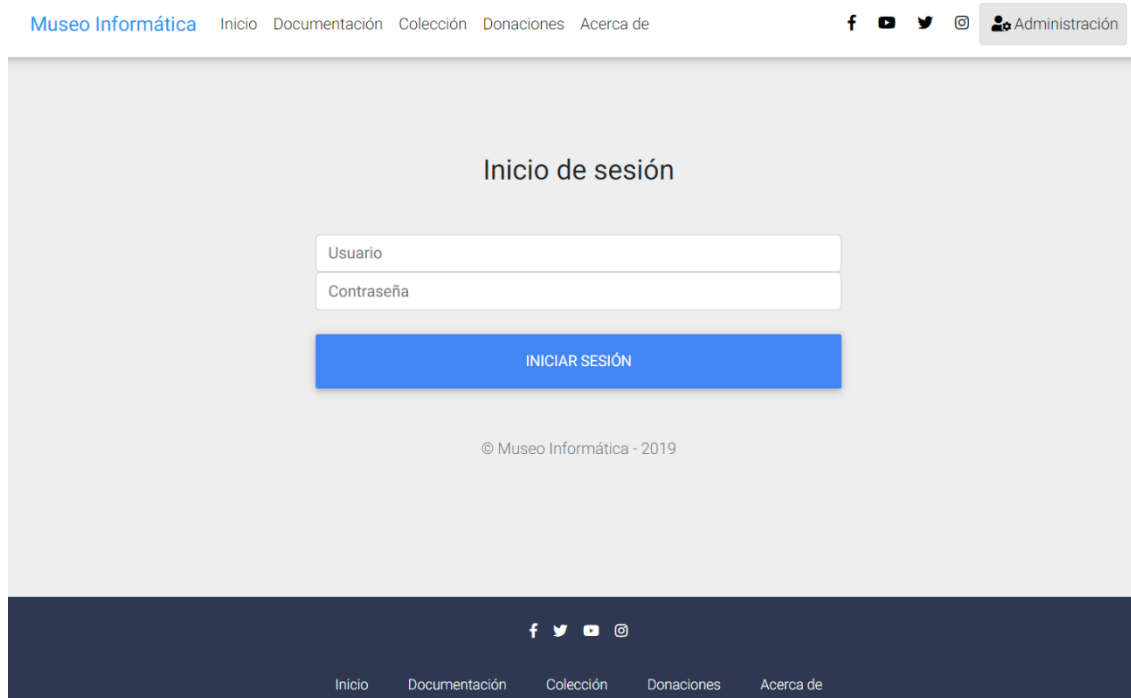
3.1.2. Resultado

Passed.

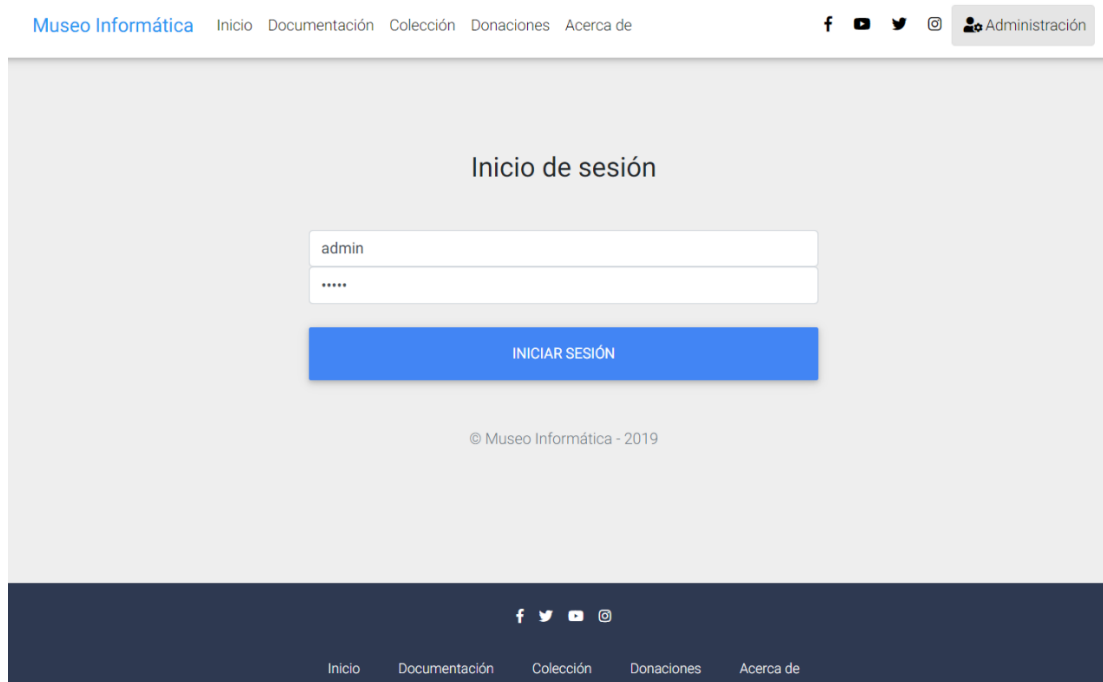
3.1.3. Evidencias



1. *Seleccionamos el botón de Administración situado en la barra de navegación superior.*



2. *Insertamos los credenciales.*



3. *Pulsamos el botón Iniciar Sesión.*

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y o t w i Administration 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		

4. Pulsar botón Cerrar Sesión.

Museo Informática Inicio Documentación Colección Donaciones Acerca de f y o t w i Administration

18C Business Consultant
Fabricante: HP
Año: 1986

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 Login.

Iniciar sesión introduciendo credenciales de administrador vacíos e incorrectos.

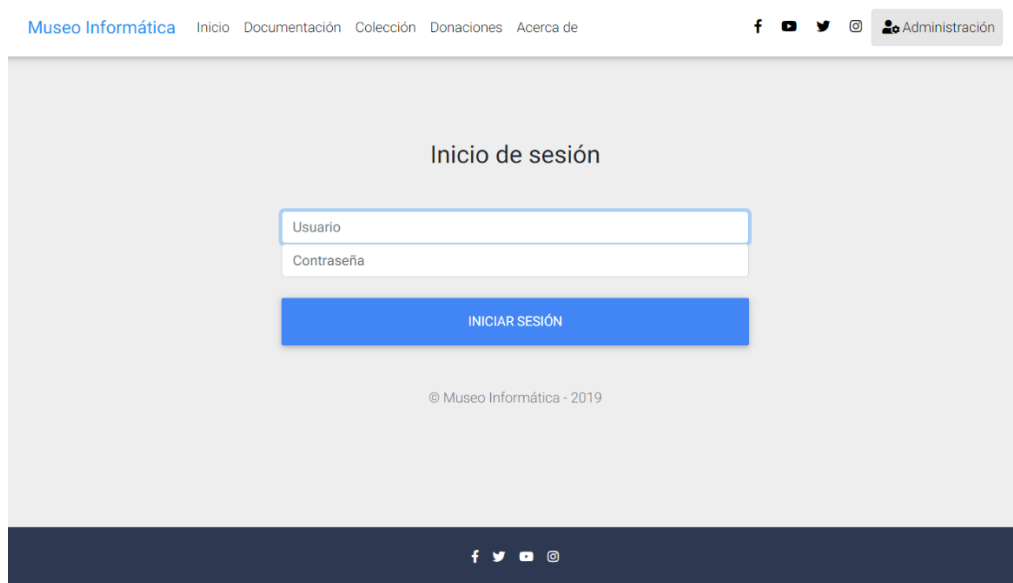
Se mostrarán mensajes de error.

3.2.2. Resultado

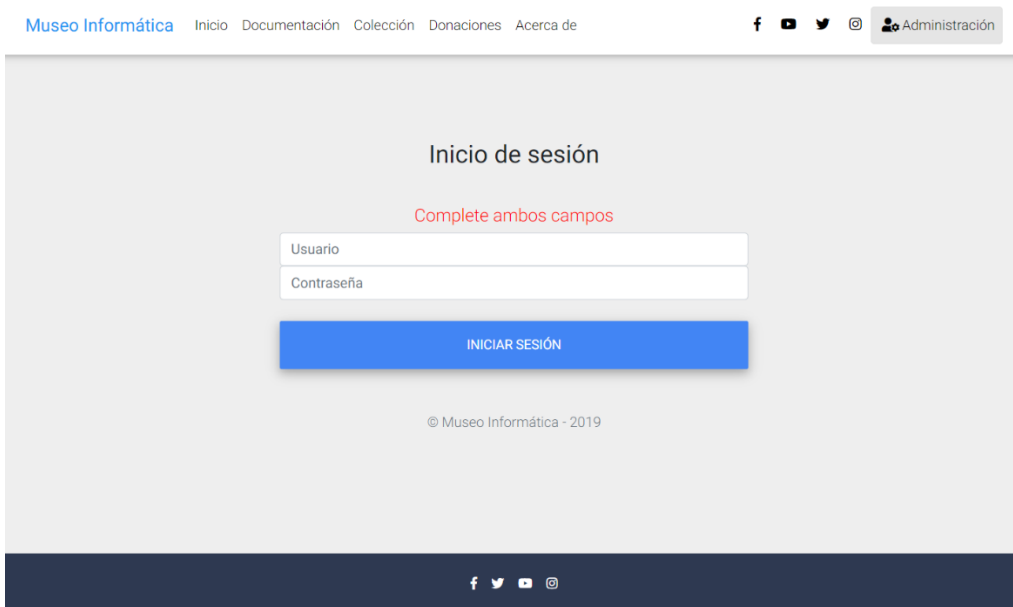
Passed.

3.2.3. Evidencias

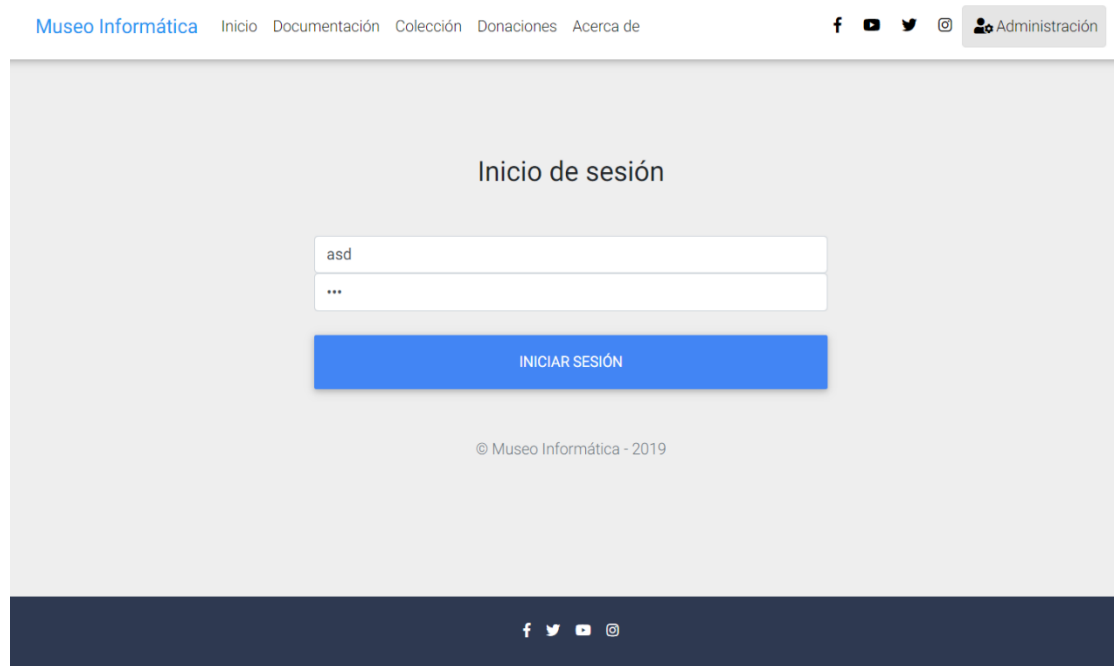
1. **Seleccionamos el botón de Administración situado en la barra de navegación superior.**



2. **Pulsamos el botón Iniciar Sesión.**



3. *Introducimos credenciales incorrectas.*



4. *Pulsamos el botón Iniciar Sesión.*

Inicio de sesión

Usuario o contraseña incorrecta

INICIAR SESIÓN

© Museo Informática - 2019

f t i s

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 documentación CRUD.

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

Posteriormente crearemos un documento.

3.3.2. Resultado

Passed.

3.3.3. Evidencias











1. Pulsamos el botón Administración.

Museo Informática Inicio Documentación Colección Donaciones Acerca de [f](#) [v](#) [t](#) [@](#) [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. Realizamos una búsqueda de un documento.

Museo Informática Inicio Documentación Colección Donaciones Acerca de [f](#) [v](#) [t](#) [@](#) [Administración](#) [Cerrar Sesión](#)

Documentación Colección Marca Tipo Salas

entorno Buscar

1 resultados encontrados [CREAR DOCUMENTO](#)

Processing. Un interesante entorno de programación	 
--	---

[f](#) [v](#) [t](#) [@](#)

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. Pulsamos el botón Crear Documento.

Crear Documento

Nombre

Descripción

Imágenes

 [AÑADIR](#)

Artículos

0 resultados encontrados [CREAR ARTÍCULO](#)

No hay artículos que mostrar.

[VOLVER](#) [CREAR DOCUMENTO](#)

4. Introducimos todos los datos.

Museo Informática Inicio Documentación Colección Donaciones Acerca de [f](#) [v](#) [t](#) [@](#) [Administración](#) [Cerrar Sesión](#)


Crear Documento

Nombre

Descripción

Imágenes

 [AÑADIR](#)



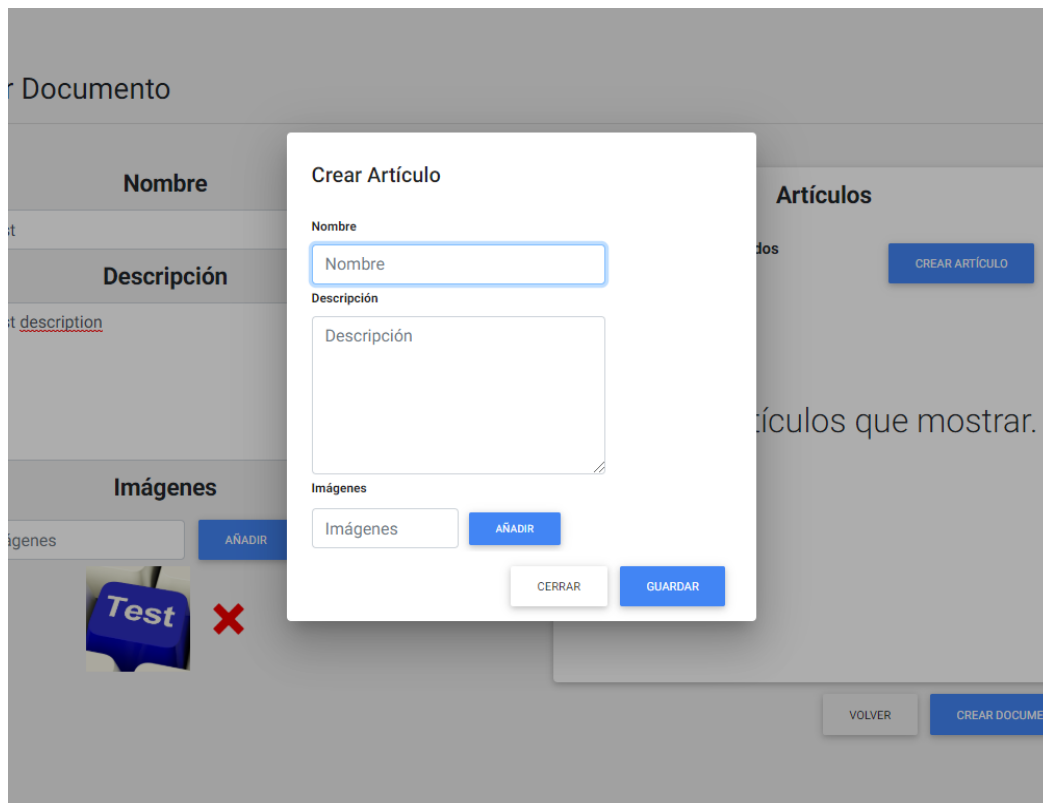
Artículos

0 resultados encontrados [CREAR ARTÍCULO](#)

No hay artículos que mostrar.

[VOLVER](#) [CREAR DOCUMENTO](#)

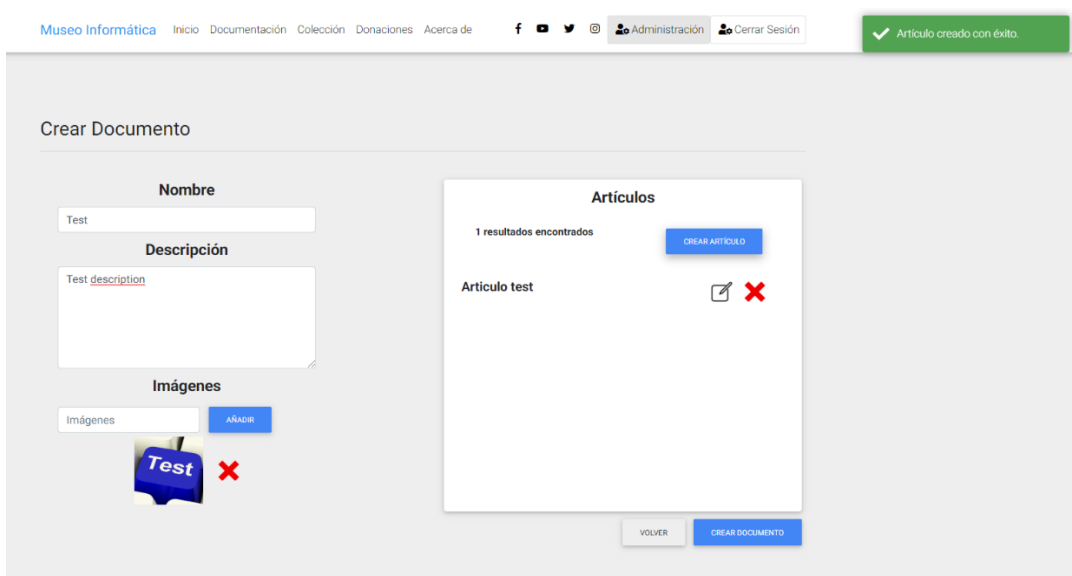
5. Pulsamos el botón Crear Artículo.



6. Introducimos los datos.



7. Pulsamos el botón Guardar.



8. Pulsamos el botón Crear Documento.

Después de la creación:

```
QUERY RESULTS 1-6 OF 6

  _id: "Test"
  name: "Test"
  imagedoc: "http://techengineers.com/wp-content/uploads/2018/07/Testing.jpg"
  text: "Test description"
  articles: Array
    0: Object
      name: "Articulo test"
      text: "articulo test description"
      images: Array
        0: "https://cdn-images-1.medium.com/max/1600/0*xMaFF2hSxpf_kIfG.jpg"
        1: "https://www.capgemini.com/mx-es/wp-content/uploads/sites/24/2019/02/Te..."
      _class: "com.museum.museumServer.main.models.Document"

  _id: ObjectId("5cbe0270b288d4327c4153d5")
  name: "Soportes de información"
  imagedoc: "http://www.teclas.org/logos/art_soporte1.jpg"
  text: "Aquí puedes hacerte a la idea de las distintas formas que se han utili..."
  articles: Array
  _class: "com.museum.museumServer.main.models.Document"

  _id: ObjectId("5cbf4f7cb288d44b7c1e08da")
  name: "Funcionamiento de una máquina de cilindro escalonado"
  imagedoc: "http://www.teclas.org/logos/art_leibniz.jpg"
  text: "Lee este artículo si quieres conocer la base de las calculadoras mecán..."
  articles: Array
  _class: "com.museum.museumServer.main.models.Document"

  _id: ObjectId("5cbf51d2b288d44b7c1e08db")
  name: "Memorias"
  imagedoc: "http://www.teclas.org/logos/art_memoria.jpg"
  text: "Puedes conocer los distintos tipos de memoria que se han utilizado en ..."
  articles: Array
```

3.3.5. Ficheros

N/A

3.4. TC004

3.4.1. Descripción

Pruebas de sistema para la ventana Documentación CRUD.

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

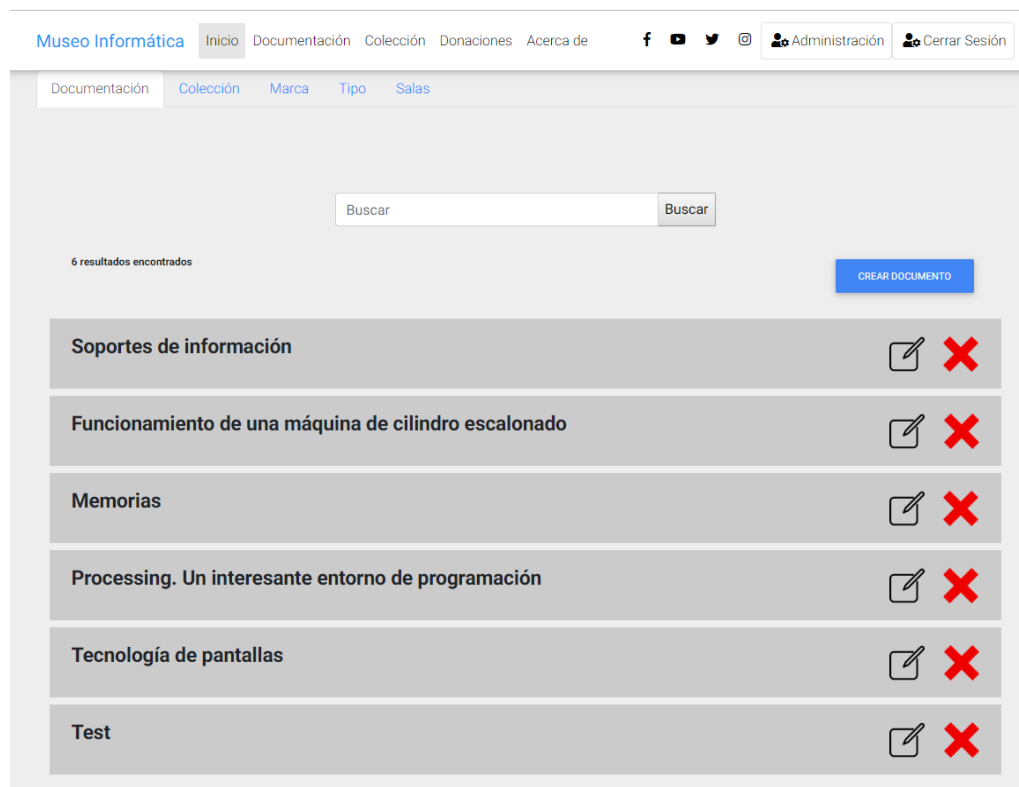
Saltarán mensajes de error y validaciones.

3.4.2. Resultado

Passed.

3.4.3. Evidencias

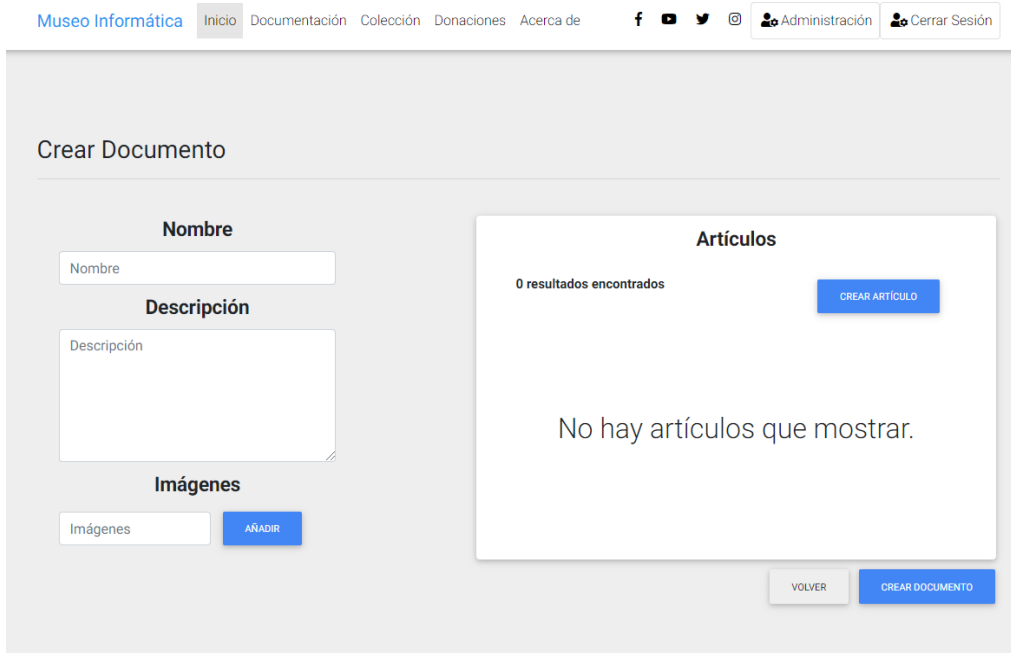
1. Pulsamos el botón Administración.



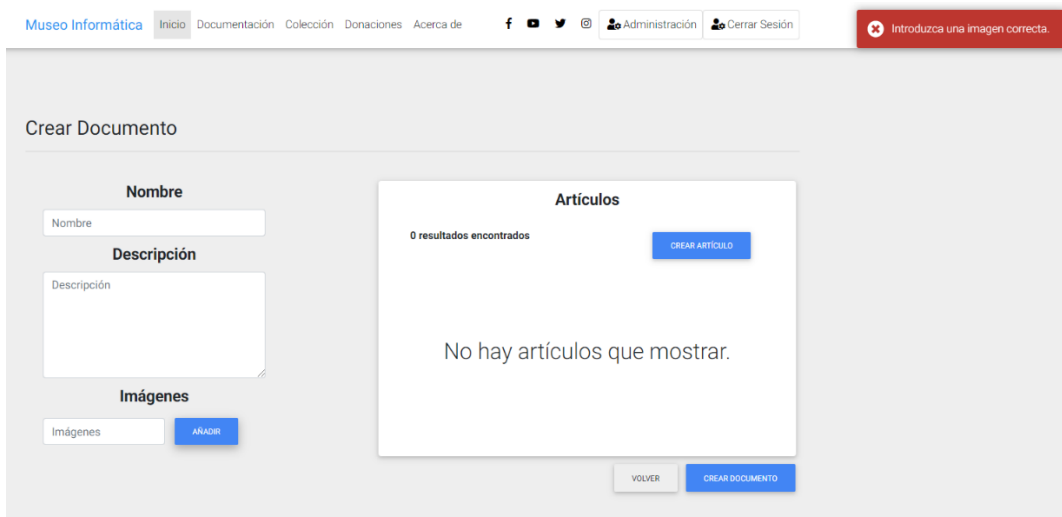
The screenshot shows the website interface for Museo Informática. At the top, there is a navigation bar with the logo 'Museo Informática' and several menu items: 'Inicio', 'Documentación', 'Colección', 'Donaciones', and 'Acerca de'. To the right of these are social media icons for Facebook, YouTube, Twitter, and Instagram, followed by 'Administración' and 'Cerrar Sesión' buttons. Below the navigation bar, there is a secondary menu with 'Documentación', 'Colección', 'Marca', 'Tipo', and 'Salas'. A search bar with the text 'Buscar' and a 'Buscar' button is positioned below the menu. The main content area displays '6 resultados encontrados' and a blue button labeled 'CREAR DOCUMENTO'. Below this, there is a list of six search results, each with a title, a pencil icon, and a red 'X' icon:

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

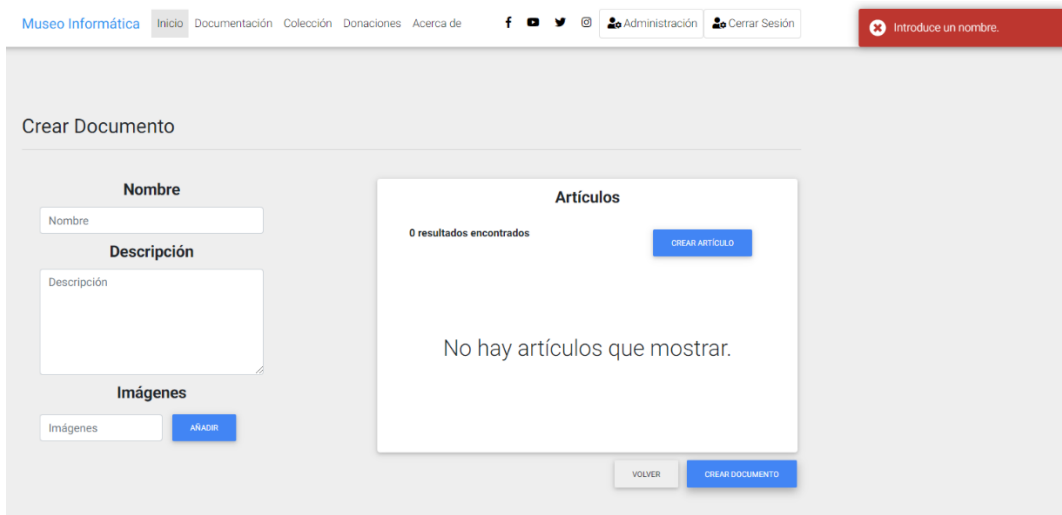
2. Pulsamos el botón Crear Documento.



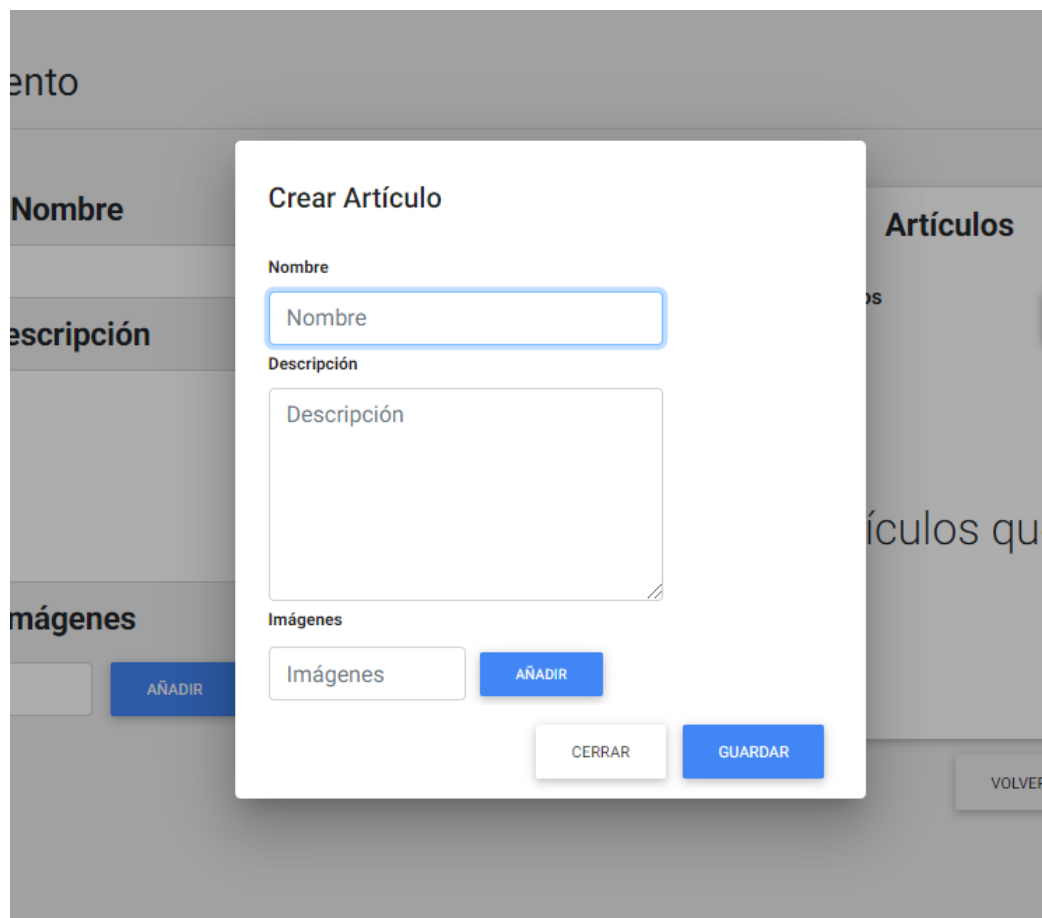
3. Pulsamos el botón Añadir.



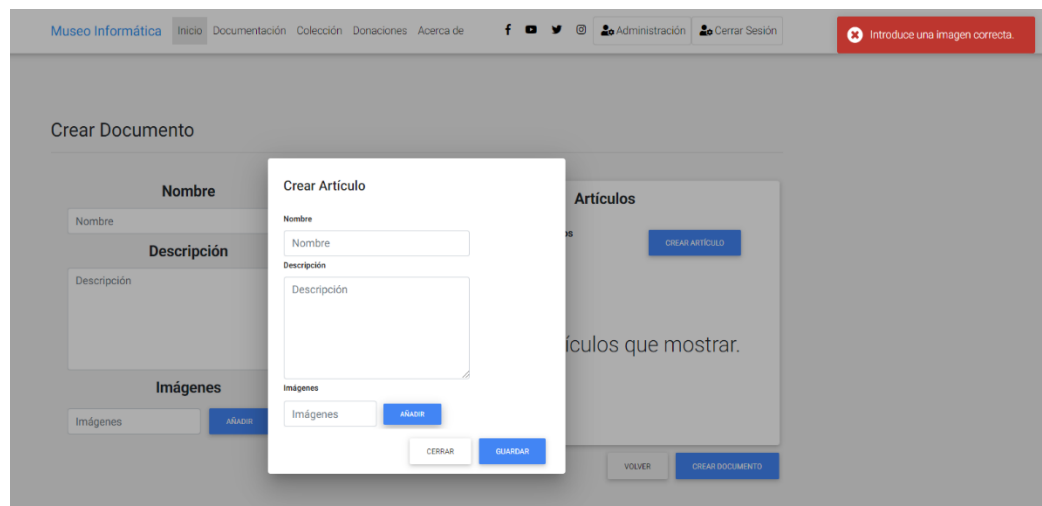
4. Pulsamos el botón Crear Documento.



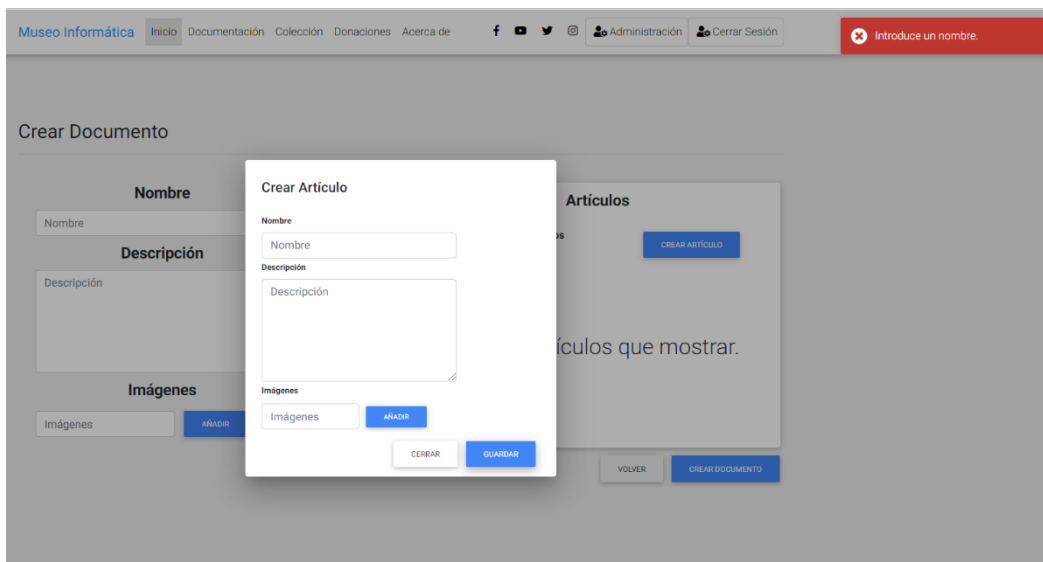
5. Pulsamos el botón Crear Artículo.



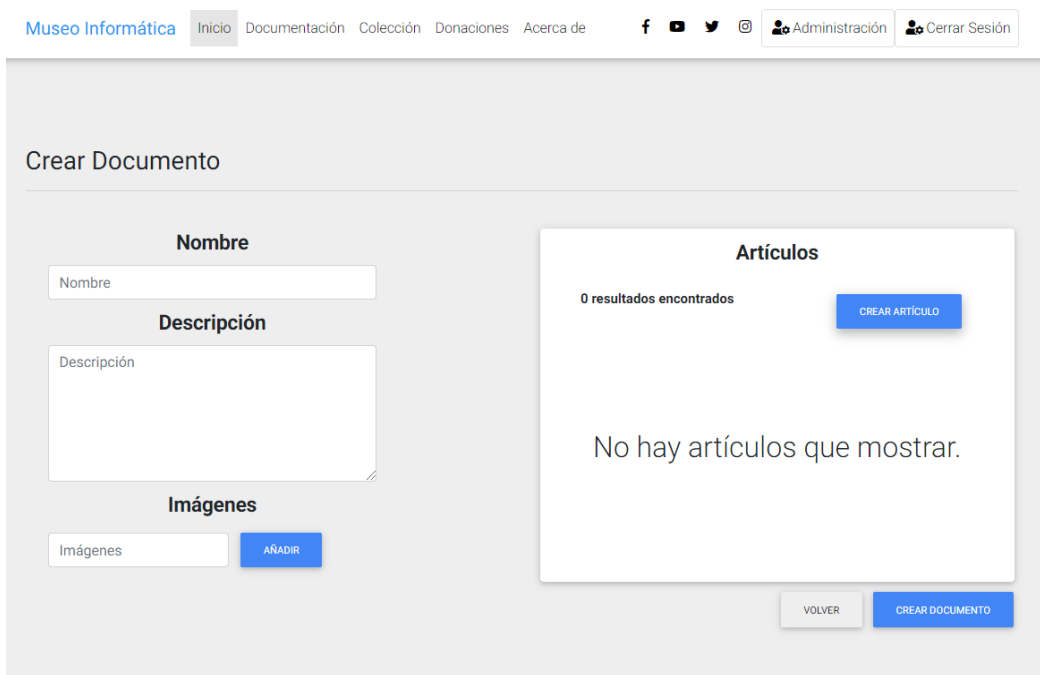
6. Pulsamos el botón Añadir.



7. Pulsamos el botón Guardar.



8. Pulsamos el botón Cerrar.















9. Pulsamos el botón Volver.

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

Documentación Colección Marca Tipo Salas

Buscar Buscar

6 resultados encontrados [CREAR DOCUMENTO](#)

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

3.4.4. Base de datos

N/A

3.4.5. Ficheros

N/A

3.5. TC005

3.5.1. Descripción

Pruebas de sistema para la ventana Documentación CRUD.

Con una sesión de administrador iniciada, editaremos el documento anteriormente creado.

3.5.2. Resultado

Passed.

3.5.3. 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

6 resultados encontrados CREAR DOCUMENTO

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

2. Pulsamos el icono de editar del documento creado anteriormente.


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

Editar Documento



Nombre
Test

Descripción
Test description

Imágenes
Imágenes AÑADIR



Artículos
1 resultados encontrados CREAR ARTÍCULO

Articulo test  

VOLVER EDITAR DOCUMENTO

3. Pulsamos el botón Añadir.


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



Eliminar la foto actual antes de introducir una nueva.

Editar Documento

Nombre
Test

Descripción
Test description

Imágenes
Imágenes


Artículos
1 resultados encontrados
Articulo test  


4. Modificamos la información.



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

Editar Documento

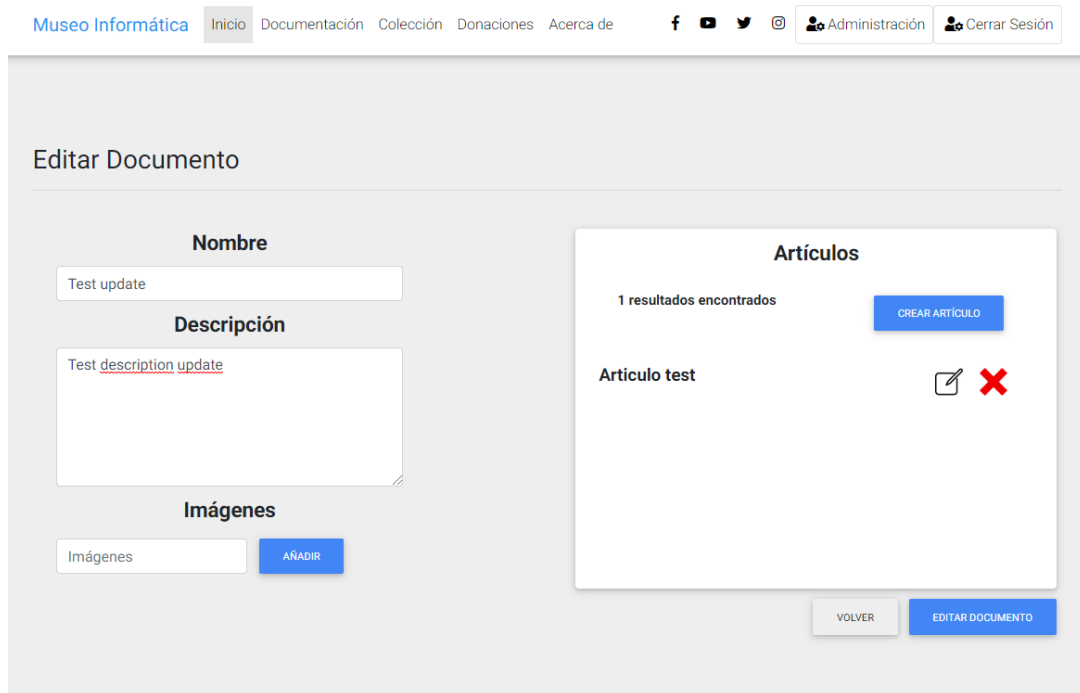
Nombre
Test update

Descripción
Test description update

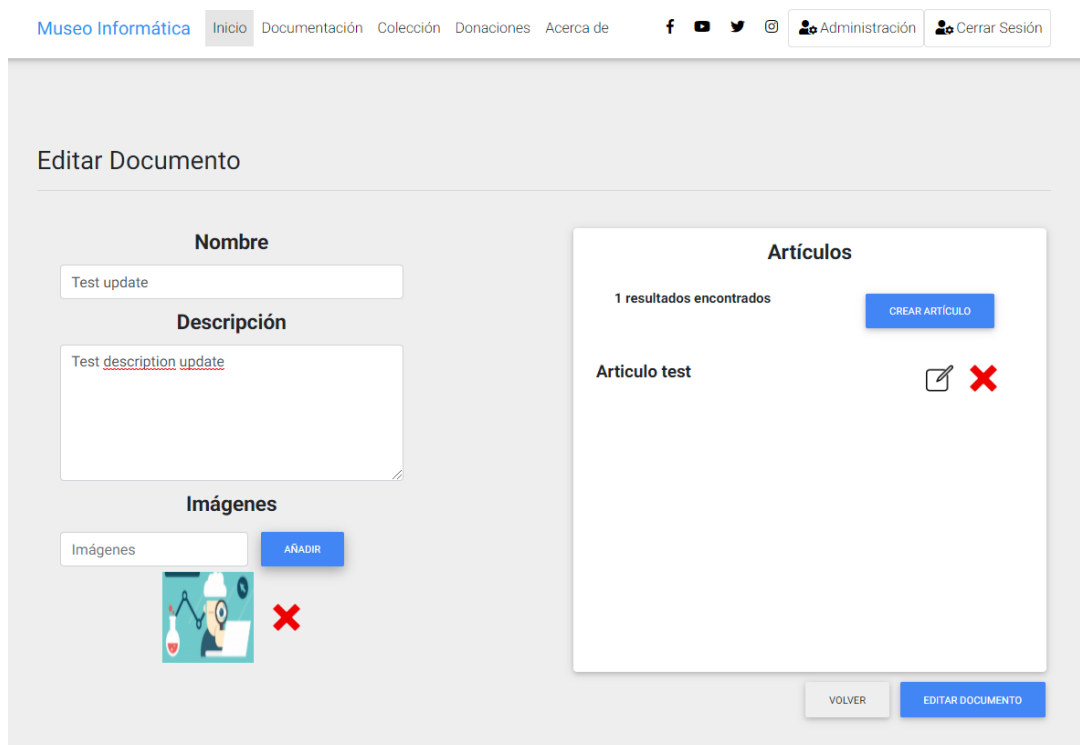
Imágenes
Imágenes


Artículos
1 resultados encontrados
Articulo test  

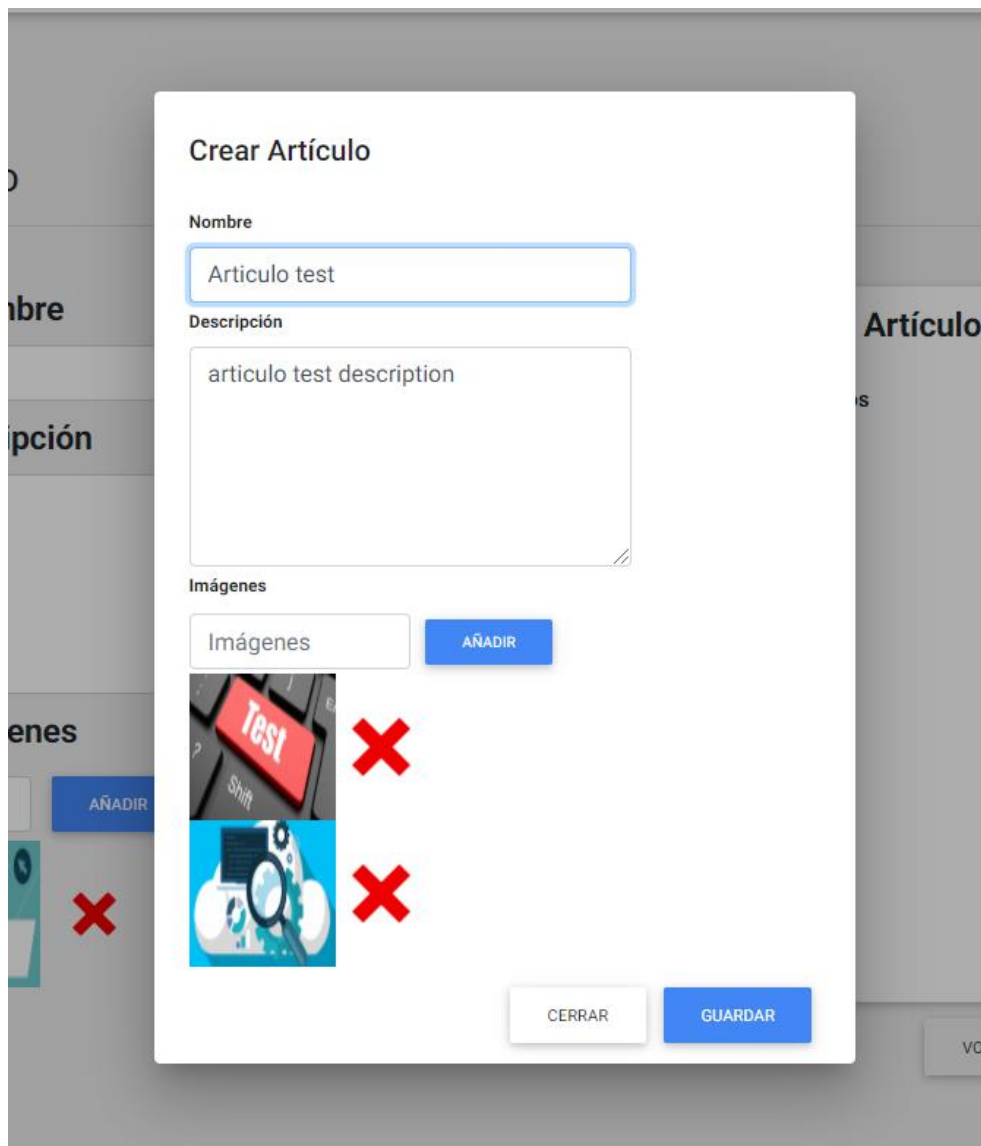
5. Eliminamos la imagen.



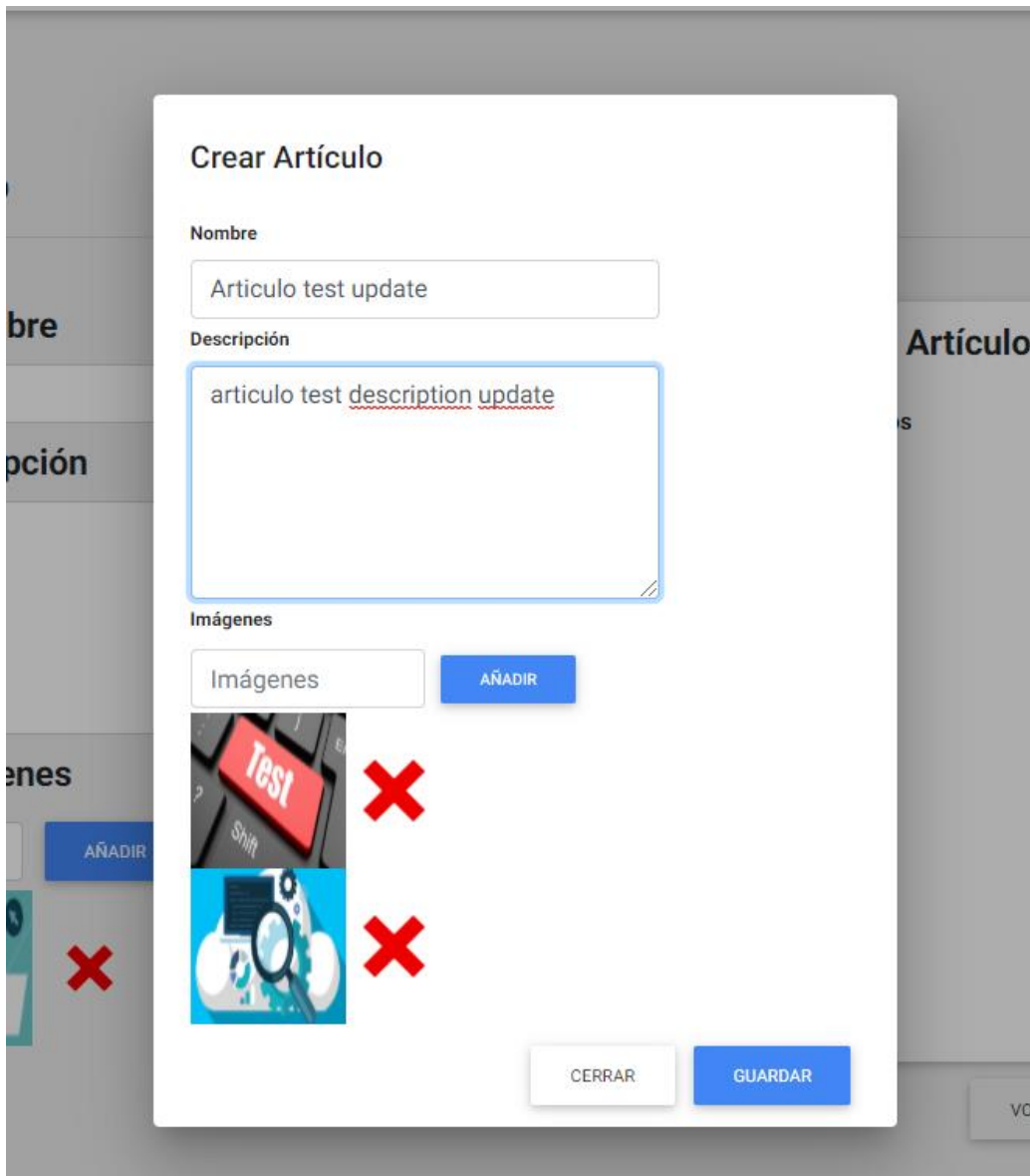
6. Añadimos otra imagen aportando su dirección.



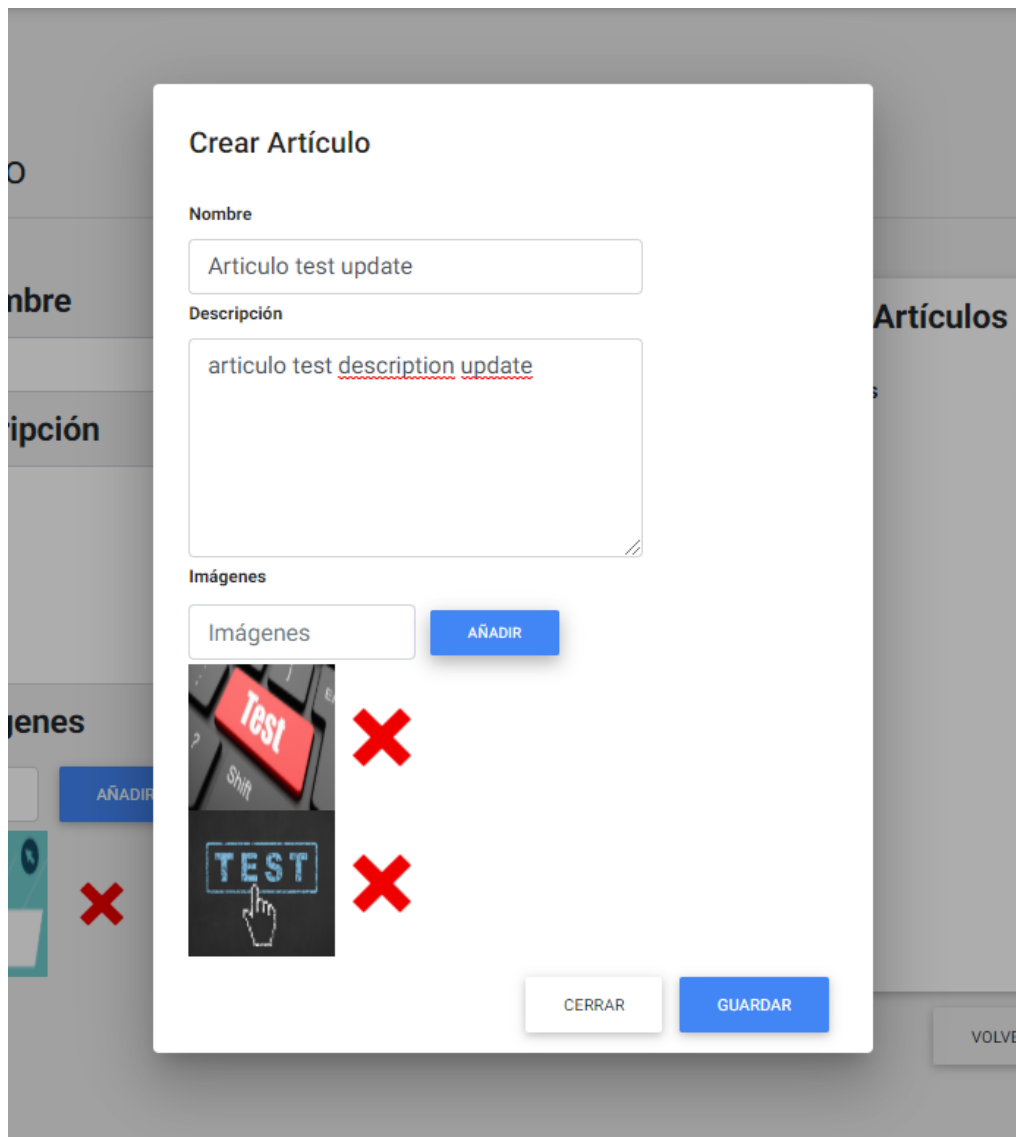
7. Pulsamos en el icono de editar un artículo.



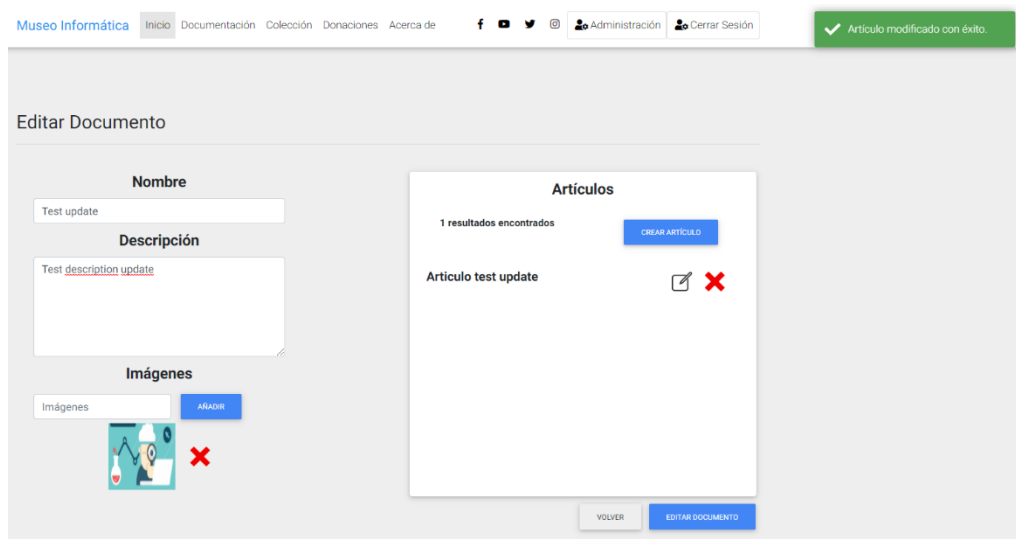
8. Modificamos la información.



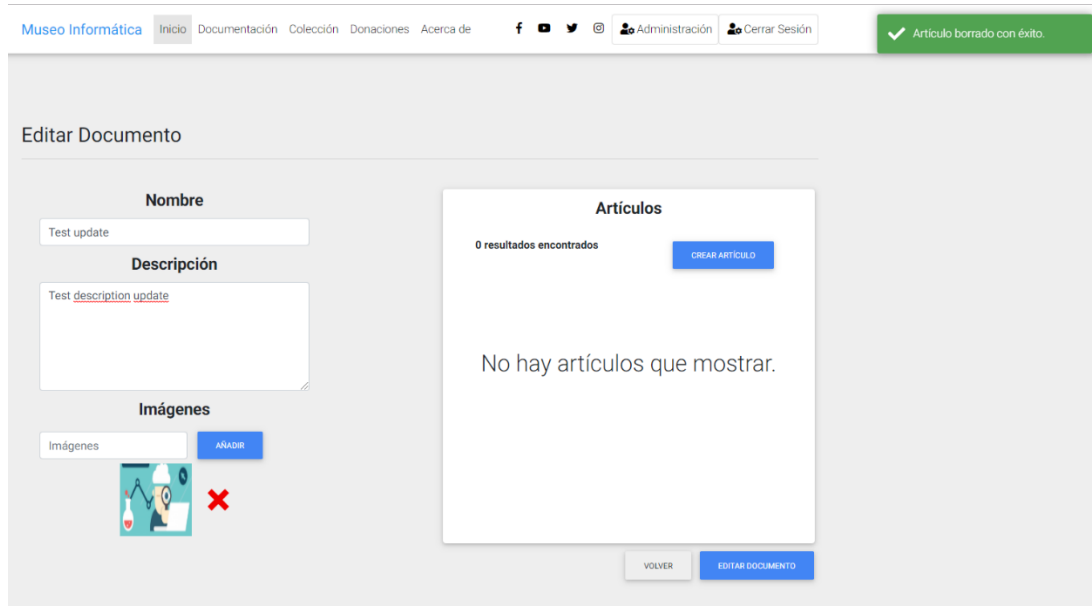
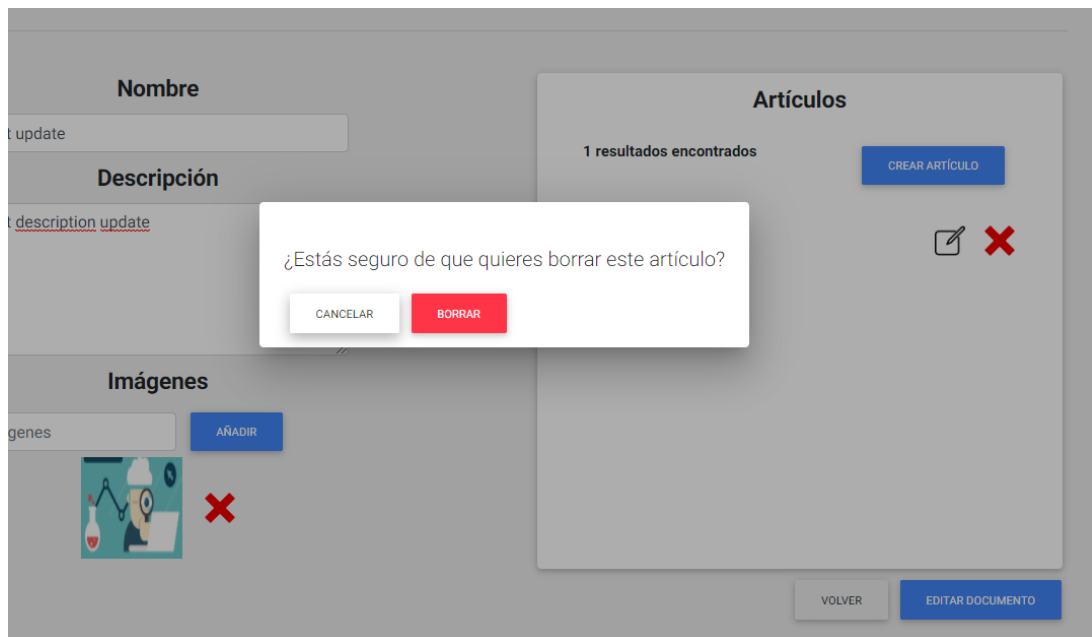
9. Borrarnos alguna imagen y añadimos otras.



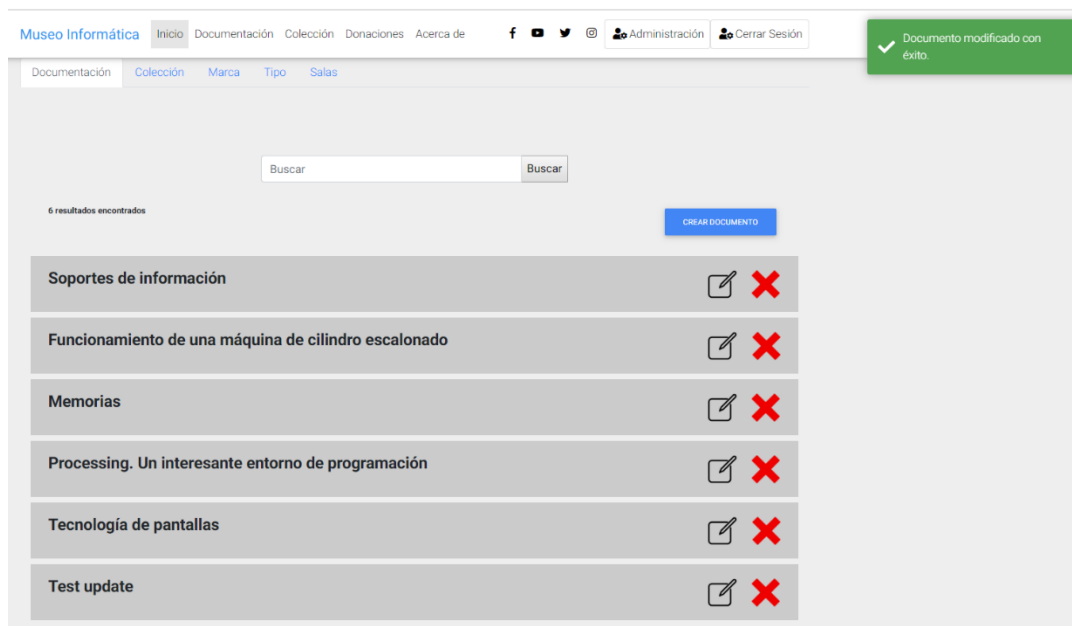
10. Pulsamos el botón Guardar.



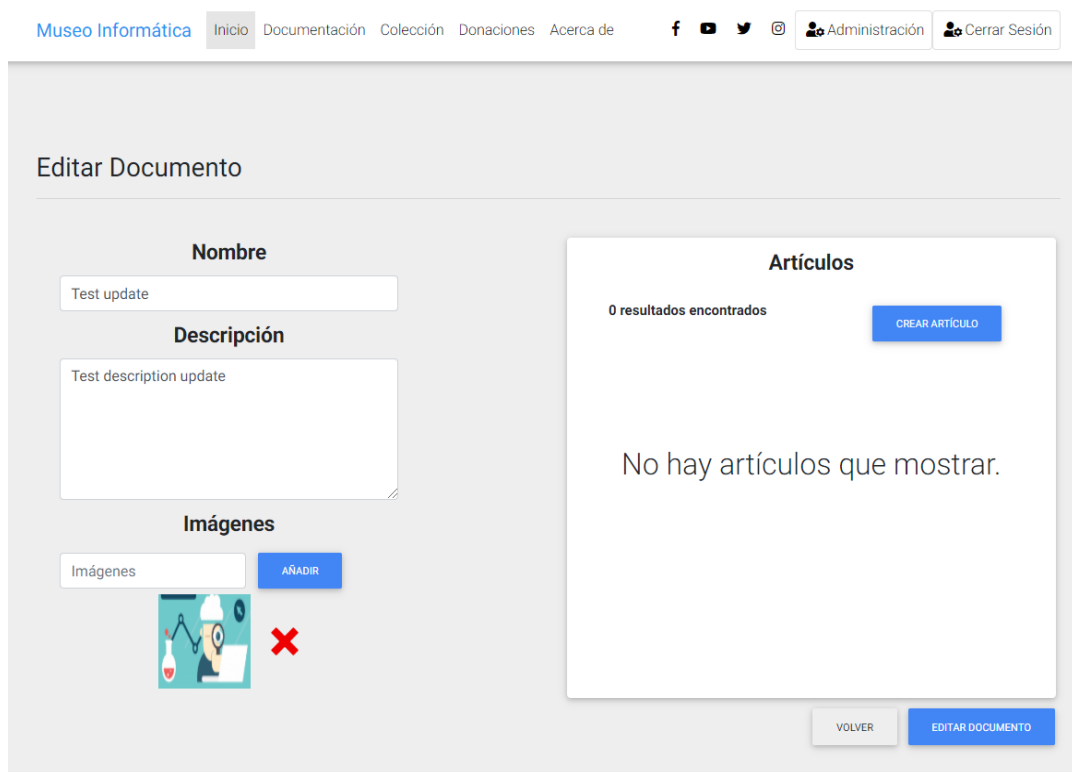
11. Pulsamos el icono de borrar un artículo.



12. Pulsamos el botón *Editar Documento*.



13. Seleccionamos el icono de editar para ese mismo documento y vemos la información modificada.



3.5.4. Base de datos

Antes de la edición:

QUERY RESULTS 1-6 OF 6

```
_id: "Test"
name: "Test"
imagedoc: "http://techengineers.com/wp-content/uploads/2018/07/Testing.jpg"
text: "Test description"
articles: Array
  0: Object
    name: "Articulo test"
    text: "articulo test description"
  images: Array
    0: "https://cdn-images-1.medium.com/max/1600/0*xMaF2hSXpf_kIfG.jpg"
    1: "https://www.cappgemini.com/mx-es/wp-content/uploads/sites/24/2019/02/Te..."
_class: "com.museum.museumServer.main.models.Document"
```

```
_id: ObjectId("5cbe0270b288d4327c4153d5")
name: "Soportes de información"
imagedoc: "http://www.teclas.org/logos/art_soporte1.jpg"
text: "Aquí puedes hacerte a la idea de las distintas formas que se han utili..."
articles: Array
_class: "com.museum.museumServer.main.models.Document"
```

```
_id: ObjectId("5cbf4f7cb288d44b7c1e08da")
name: "Funcionamiento de una máquina de cilindro escalonado"
imagedoc: "http://www.teclas.org/logos/art_leibniz.jpg"
text: "Lee este artículo si quieres conocer la base de las calculadoras mecán..."
articles: Array
_class: "com.museum.museumServer.main.models.Document"
```

```
_id: ObjectId("5cbf51d2b288d44b7c1e08db")
name: "Memorias"
imagedoc: "http://www.teclas.org/logos/art_memoria.jpg"
text: "Puedes conocer los distintos tipos de memoria que se han utilizado en ..."
articles: Array
```

Después de la edición:

QUERY RESULTS 1-6 OF 6

```
> _id: ObjectId("5cbf4f7cb288d44b7c1e08da")
name: "Funcionamiento de una máquina de cilindro escalonado"
imagedoc: "http://www.teclas.org/logos/art_leibniz.jpg"
text: "Lee este artículo si quieres conocer la base de las calculadoras mecán..."
articles: Array
_class: "com.museum.museumServer.main.models.Document"
```

```
_id: "Test"
name: "Test update"
imagedoc: "https://adrianalonso.es/wp-content/uploads/2017/06/testing.jpg"
text: "Test description update"
articles: Array
_class: "com.museum.museumServer.main.models.Document"
```

```
_id: ObjectId("5cbe0270b288d4327c4153d5")
name: "Soportes de información"
imagedoc: "http://www.teclas.org/logos/art_soporte1.jpg"
text: "Aquí puedes hacerte a la idea de las distintas formas que se han utili..."
articles: Array
_class: "com.museum.museumServer.main.models.Document"
```

```
_id: ObjectId("5cbf51d2b288d44b7c1e08db")
name: "Memorias"
imagedoc: "http://www.teclas.org/logos/art_memoria.jpg"
text: "Puedes conocer los distintos tipos de memoria que se han utilizado en ..."
articles: Array
```

3.5.5. Ficheros

N/A

3.6. TC006

3.6.1. Descripción

Pruebas de sistema para la ventana Documentación CRUD.

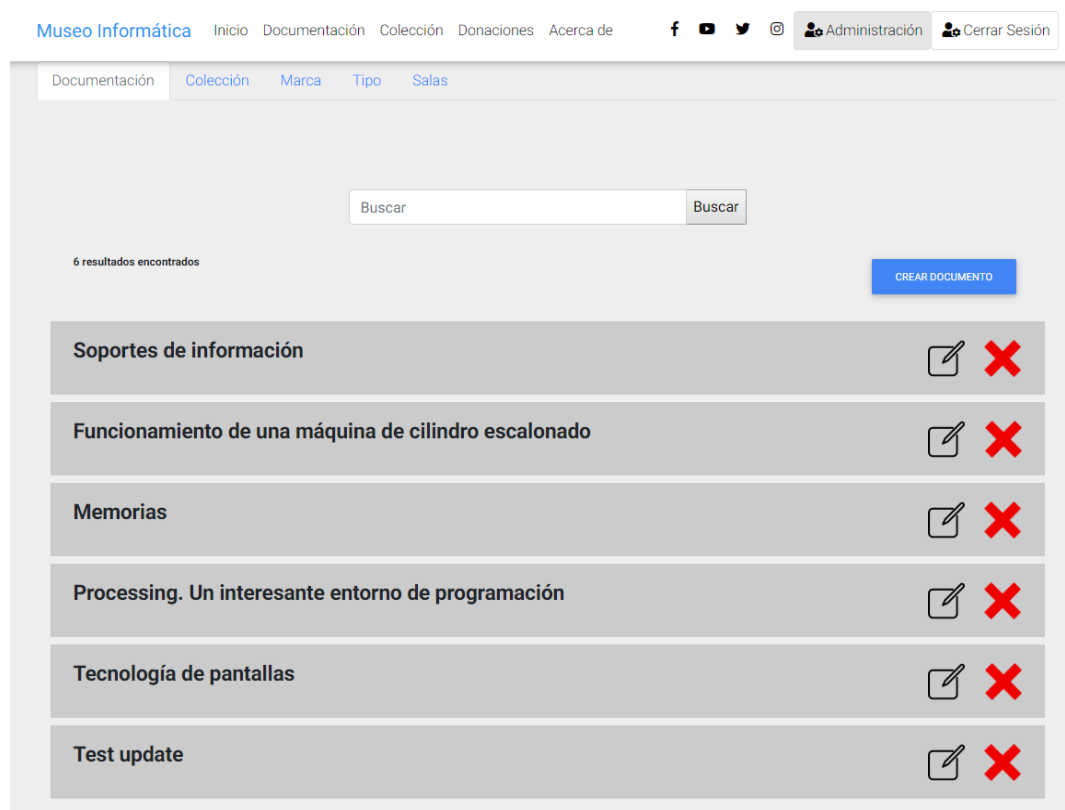
Con una sesión de administrador iniciada, borraremos el documento creado para estas pruebas.

3.6.2. Resultado

Passed.

3.6.3. Evidencias

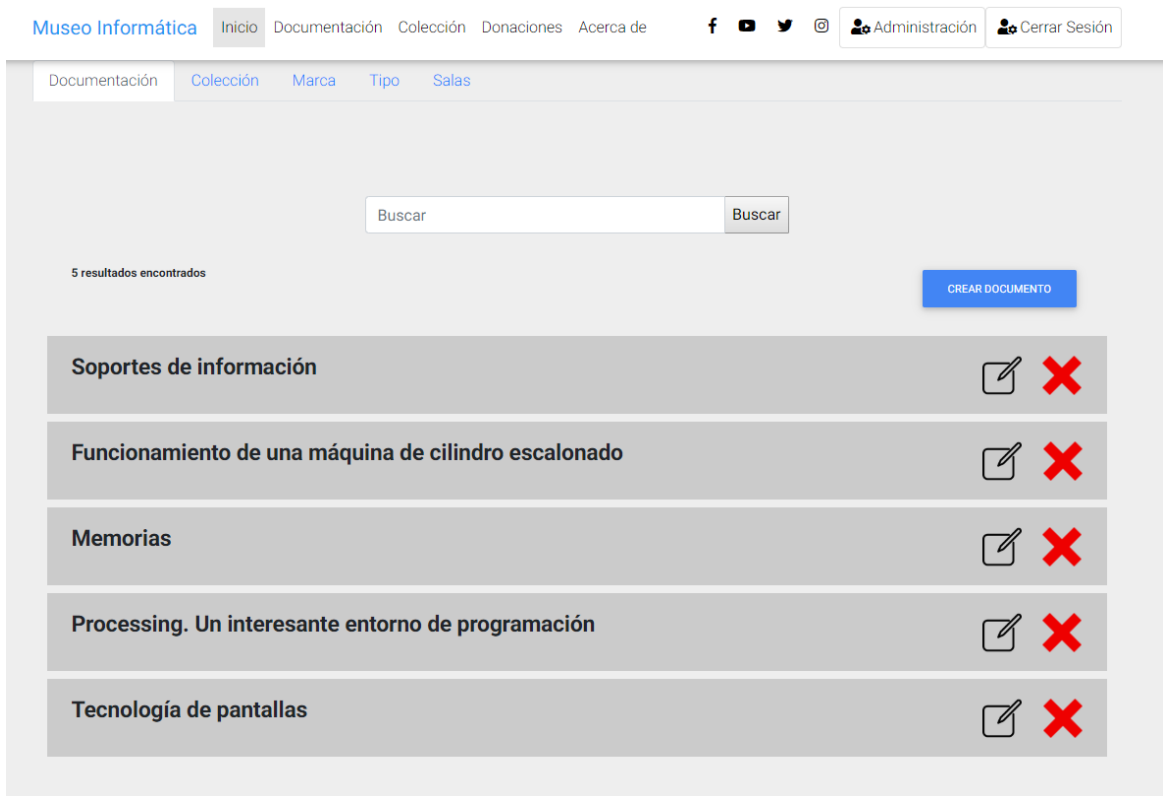
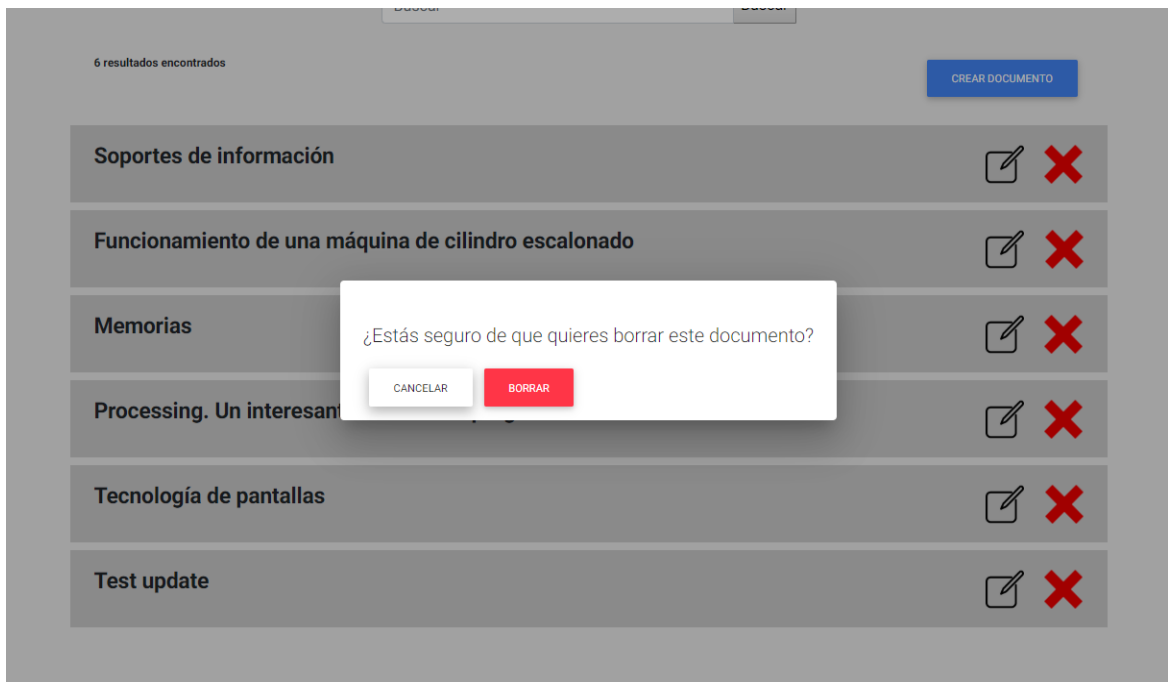
1. Pulsamos el botón Administración.



The screenshot shows the top navigation bar of the Museo Informática website. The 'Administración' button is highlighted. Below the navigation bar, there is a search bar with the text 'Buscar' and a 'Buscar' button. The search results show 6 results found. A blue button labeled 'CREAR DOCUMENTO' is visible. The list of documents includes:

Documento	Editar	Borrar
Soportes de información		
Funcionamiento de una máquina de cilindro escalonado		
Memorias		
Processing. Un interesante entorno de programación		
Tecnología de pantallas		
Test update		

2. Pulsamos el icono de borrar un documento.



3.6.4. Base de datos

Antes del borrado:

QUERY RESULTS 1-6 OF 6

```
> _id: ObjectId("5cbf4f7cb288d44b7c1e08da")
  name: "Funcionamiento de una máquina de cilindro escalonado"
  imagedoc: "http://www.teclas.org/logos/art_leibniz.jpg"
  text: "Lee este artículo si quieres conocer la base de las calculadoras mecán..."
  > articles: Array
  _class: "com.museum.museumServer.main.models.Document"
```

```
{
  _id: "Test"
  name: "Test update"
  imagedoc: "https://adrianalonso.es/wp-content/uploads/2017/06/testing.jpg"
  text: "Test description update"
  > articles: Array
  _class: "com.museum.museumServer.main.models.Document"
}
```

```
_id: ObjectId("5cbe0270b288d4327c4153d5")
name: "Soportes de información"
imagedoc: "http://www.teclas.org/logos/art_soporte1.jpg"
text: "Aquí puedes hacerte a la idea de las distintas formas que se han utili..."
> articles: Array
_class: "com.museum.museumServer.main.models.Document"
```

```
_id: ObjectId("5cbf51d2b288d44b7c1e08db")
name: "Memorias"
imagedoc: "http://www.teclas.org/logos/art_memoria.jpg"
text: "Puedes conocer los distintos tipos de memoria que se han utilizado en ..."
> articles: Array
_class: "com.museum.museumServer.main.models.Document"
```

Después del borrado:

QUERY RESULTS 1-5 OF 5

> `_id: ObjectId("5cbe0270b288d4327c4153d5")`
`name: "Soportes de información"`
`imagedoc: "http://www.teclas.org/logos/art_soporte1.jpg"`
`text: "Aquí puedes hacerte a la idea de las distintas formas que se han utili..."`
> `articles: Array`
`_class: "com.museum.museumServer.main.models.Document"`

`_id: ObjectId("5cbf4f7cb288d44b7c1e08da")`
`name: "Funcionamiento de una máquina de cilindro escalonado"`
`imagedoc: "http://www.teclas.org/logos/art_leibniz.jpg"`
`text: "Lee este artículo si quieres conocer la base de las calculadoras mecán..."`
> `articles: Array`
`_class: "com.museum.museumServer.main.models.Document"`

`_id: ObjectId("5cbf51d2b288d44b7c1e08db")`
`name: "Memorias"`
`imagedoc: "http://www.teclas.org/logos/art_memoria.jpg"`
`text: "Puedes conocer los distintos tipos de memoria que se han utilizado en ..."`
> `articles: Array`
`_class: "com.museum.museumServer.main.models.Document"`

`_id: "Processing. Un interesante entorno de programación"`
`name: "Processing. Un interesante entorno de programación"`
`imagedoc: "http://www.teclas.org/logos/art_processing.jpg"`
`text: "Processing es un entorno de desarrollo muy simple pero potente, basado..."`
> `articles: Array`
`_class: "com.museum.museumServer.main.models.Document"`

`_id: "Tecnología de pantallas"`
`name: "Tecnología de pantallas"`
`imagedoc: "http://www.teclas.org/logos/art_pantalla.jpg"`
`text: "En este texto podrás conocer los distintos tipos de pantallas que se h..."`
> `articles: Array`
`_class: "com.museum.museumServer.main.models.Document"`

3.6.5. *Ficheros*

N/A