



Universidad
Zaragoza

Trabajo Fin de Grado

Sistema de Información para el Museo de
Informática Histórica (MIH) de la Universidad de
Zaragoza

Information System for the Historical Computer
Science Museum (MIH) of the University of
Zaragoza

Autor:

David Alloza Tejero

Director:

Eduardo Mena Nieto

Titulación del autor:

Graduado en Ingeniería Informática

Escuela de Ingeniería y Arquitectura, Zaragoza

2021

TÍTULO

Sistema de Información para el Museo de Informática Histórica (MIH) de la Universidad de Zaragoza.

RESUMEN

En este trabajo se ha desarrollado un sistema completamente funcional que se utilizará para catalogar todos aquellos objetos que sean donados al Museo de Informática Histórica (MIH) de la Universidad de Zaragoza, así como para catalogar progresivamente todos aquellos objetos que ya pertenecen al museo.

Este sistema consta de una base de datos MySQL en la que almacenar toda la información y una aplicación web desarrollada en Java y JavaScript que permite consultar el catálogo del museo de forma abierta para cualquier usuario o realizar acciones de administración en ella, esta última accesible solo para el usuario administrador con identificación previa.

El desarrollo de este trabajo puede dividirse en cuatro etapas:

- 1) Análisis previo al desarrollo del sistema de los gestores del mercado y decisión de cuál es mejor opción para resolver este problema.
- 2) Diseño del modelo de datos.
- 3) Desarrollo de la aplicación web.
- 4) Incorporación de pequeñas mejoras a la aplicación, solución de errores y catalogación de varias donaciones reales del museo.

AGRADECIMIENTOS

Agradecer a Eduardo Mena Nieto por acompañarme en el desarrollo en este trabajo fin de grado, por darme a conocer los entresijos del Museo de Informática Histórica de la Universidad de Zaragoza y por ofrecerme su disponibilidad en todo momento.

Agradecer también a mi familia por apoyarme en todo momento durante estos 10 meses en los que he desarrollado mi trabajo fin de grado.

Índice

1. Introducción	7
1.1. El Museo de Informática Histórica (MIH)	7
1.2. Objetivos del proyecto	8
1.3. Tecnologías utilizadas	11
1.4. Estructura de la memoria	12
2. Diseño de la base de datos	12
2.1. Análisis de los SGBD del mercado	12
2.2. Elección del SGBD	14
2.3. Esquema de la base de datos	15
2.3.1. Esquema Entidad-Relación	18
2.3.2. Esquema Relacional	21
2.3.3. Evolución del esquema	22
3. Arquitectura del sistema de información para el MIH	24
3.1. Diseño de la aplicación	24
3.2. Funcionalidades de la aplicación	29
4. Pruebas efectuadas al sistema	34
4.1. Validación	34
4.2. Sobrecarga y eficiencia	35
5. Conclusiones	37
5.1. Cronograma	38
5.2. Posibles ampliaciones	39
5.3. Opinión personal	40
Bibliografía	42
A. Anexo: Análisis individual de los SGBD del mercado	43

A.1. MySQL	43
A.2. MariaDB	43
A.3. Oracle	43
A.4. PostgreSQL	43
A.5. DB2	43
A.6. Informix	44
A.7. CACHE	44
A.8. DB4O	44
A.9. MongoDB	45
A.10. Cuadro comparativo de los gestores anteriores	45
B. Anexo: Tablas de la Base de Datos	46
B.1. Tabla TIPOS	46
B.2. Tabla DONANTES	46
B.3. Tabla DONACIONES	47
B.4. Tabla ITEMS	48
B.5. Tabla IMAGENES.ITEMS	49
B.6. Tabla IMAGENES.TIPOS	50
B.7. Tabla ATTR.TIPOS	50
B.8. Tabla ATTR.VALORES	52
B.9. Tabla ATTR.ITEMS.STRING	53
B.10. Tabla ATTR.ITEMS.REF	54
B.11. Tabla ZONAS	54
B.12. Tabla LOCALIZACIONES	55
B.13. Tabla UBICACIONES	56
C. Anexo: Manual de usuario de las funcionalidades de la aplicación	57
C.1. Funcionalidades de usuario	57
C.1.1. Barra de navegación	57
C.1.2. Visión global del museo	58
C.1.3. Visualización de la jerarquía de tipos de ítems	59

C.1.4.	Vista detallada de un tipo de ítem	60
C.1.5.	Consultar datos variados con información detallada	62
C.1.6.	Consultar datos estadísticos del sistema	66
C.1.7.	Vista detallada de un ítem	68
C.1.8.	Búsqueda en el sistema	70
C.2.	Funcionalidades específicas de administración	71
C.2.1.	Visualización de la jerarquía de tipos de ítems	71
C.2.2.	Vista detallada de un tipo de ítem	74
C.2.3.	Consultar datos variados con información detallada	84
C.2.4.	Vista detallada de un ítem	92

D. Anexo: Contenido del fichero XML de configuración *configuracion_tablas.xml* 101

E. Anexo: Contenido del fichero XML de configuración *variables constantes.xml* 107

F. Anexo: Diagrama de Gantt completo. 109

1. Introducción

El Museo de Informática Histórica de la Universidad de Zaragoza, situado en el edificio Ada Byron de la EINA, lleva recibiendo centenares de donaciones de objetos anuales desde que se creara en año 2003. Todos estos objetos se guardan en el almacén del sótano y después, aquellos que son más relevantes y están en mejor estado se seleccionan para ser expuestos las vitrinas de los pasillos del edificio.

Eduardo Mena Nieto, profesor del Dpto. de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, miembro de la asociación RetroAcción, y coordinador del museo y director de este Trabajo Fin de Grado, se encarga de catalogar las donaciones de todos aquellos productos que recibe. Para ello utiliza un único documento Google Drive donde anota todas las características de los objetos e información relevante sobre el donante.

Después de 18 años con el museo en funcionamiento, ese documento se ha vuelto tan voluminoso que resulta muy complicado trabajar con él y la búsqueda que ofrece es solo a partir de cadenas de texto, por lo que resulta difícil por ejemplo hacer un recuento de ítems catalogados. Por estos principales motivos se ha considerado adecuado informatizar este registro de información.

1.1. El Museo de Informática Histórica (MIH)

El Museo de Informática Histórica surge en 2003 por la motivación de un grupo de personas del Centro Politécnico Superior (actualmente Escuela de Ingeniería y Arquitectura, EINA) de la Universidad de Zaragoza por recuperar parte del patrimonio de las tecnologías y sus conocimientos, precursores de la breve y trepidante era informática, completando así unos fondos representativos del avance tecnológico en nuestra sociedad.

El MIH resulta una iniciativa atractiva y pionera en nuestra comunidad autónoma, siendo éste un reto que la asociación RetroAcción afronta en toda su extensión e intensidad en colaboración con la EINA desde que en 2009 dicho centro confió en RetroAcción la gestión técnica del MIH.

En estos momentos los fondos de MIH incluyen varios miles de artículos tecnológicos (desde sistemas completos, dispositivos de almacenamiento, microprocesadores, periféricos...), buena parte de su software más significativo, y sus correspondientes fuentes de documentación en distintos soportes (manuales originales, guías de usuario, fotografías, documentos digitales, etc.). Se recogen también, siempre que es

posible, el testimonio de las personas que han utilizado dichas tecnologías. La parte más importante de estos fondos se encuentra en la exposición permanente que se distribuye a lo largo de las distintas plantas del edificio Ada Byron del Campus Río Ebro. Se comentan a continuación cada una de las zonas temáticas del museo que se distribuyen a lo largo de las tres plantas del edificio:

- **Planta 0:** Se incluyen 6 zonas centradas en los componentes de un ordenador, los ordenadores domésticos, los ordenadores Apple, los compatibles PC, la computación portátil, y los servidores y terminales. De esta manera se muestra un completo recorrido a través de las distintas familias de ordenadores de la historia de la Informática.
- **Planta 1:** Se incluyen cuatro zonas centradas en algunas de las principales aplicaciones y objetivos de los ordenadores: una doble zona dedicada a los videojuegos (de máquinas recreativas, consolas, y ordenadores), la creación de gráficos y música por ordenador, y las aplicaciones más empresariales.
- **Planta 2:** Incluye una zona situada a lo largo de las cristaleras de la sala de estudios del edificio Ada Byron y que sirve de complemento a la zona 6 de la planta 0, incluyendo algún servidor más y sobre todo estaciones de trabajo.

Para ampliar la información sobre el museo, visitar la página del MIH <http://mih.unizar.es/> o consultar el dossier [10] del museo accesible también desde esa web.

1.2. Objetivos del proyecto

Se listan a continuación los objetivos principales que se pretenden abordar con la realización de este proyecto:

1. El objetivo principal es elaborar un sistema de información que permita llevar un registro de todos los elementos que forman parte del Museo de Informática Histórica (MIH) situado en la EINA.
2. Se desarrollará una aplicación Web que permita consultar dicha información por parte de los usuarios, así como catalogar la información sobre los elementos por parte del personal administrador del museo.
3. Este proyecto pretende solucionar un problema de la vida real que existe desde que se creó el museo y que todavía no ha sido solucionado. En consecuencia, el

principal reto de este trabajo es manejar un modelado de la información que puede ser cambiante a lo largo del tiempo conforme el museo crece.

4. Toda la aplicación girará entorno a los objetos físicos que se donan al museo con el fin de ser almacenados y catalogados, a dichos objetos se les denominará ítems físicos.
5. Para facilitar la catalogación de los ítems físicos, se ha creado el concepto de tipo de ítem. Un tipo será una abstracción de un conjunto de ítems físicos en el que se representarán las características comunes que todos ellos comparten.
6. Los tipos se representarán mediante una jerarquía, lo que permitirá a algunos tipos heredar características de tipos con mayor nivel de abstracción. Esta idea de almacenamiento jerárquico tiene como principales objetivos el poder catalogar un ítem de forma más rápida y evitar información repetida en el sistema. Conforme se cataloguen objetos irán apareciendo nuevos tipos de ítems, y será común que la clasificación inicial hecha de los tipos no sea la más adecuada pasado un tiempo y que haya que reorganizarla, por este motivo resulta de vital importancia ofrecer una jerarquía dinámica de tipos, en la que poder añadir nuevos tipos, eliminar tipos no deseados o reubicar otros en diferentes ramas.
7. Las características de un tipo estarán representadas en el sistema mediante atributos. Estos atributos de los tipos tendrán un rango y un valor asociado. El rango determinará el tipo de información que representan, pudiendo ser un valor literal o una referencia a otro ítem. El valor asociado podrá ser asignado por el propio tipo, por los subtipos o por los ítems asociados a dicho tipo (o cuyo tipo asociado es descendiente).
8. Cada donación realizada al museo será registrada en la aplicación, indicando el donante que la realizó, así como información adicional como la fecha, el tipo de donación, etc.
9. El museo se dividirá en zonas, y cada zona tendrá una o varias localizaciones en las que se podrán encontrar ubicados los ítems del museo. La aplicación permitirá consultar la información de las zonas, las localizaciones y los ítems ubicados en ellas.
10. Se pondrá también a disposición del administrador del museo una cuenta de usuario administrador para el sistema. Con ella, además de poder interactuar con la aplicación como un usuario invitado normal, podrá llevar a cabo diversas

tareas en el sistema para catalogar y almacenar la información relativa a los objetos donados en el museo. Podrá realizar las siguientes acciones:

- Modificar la jerarquía de tipos de ítems. Se le permitirá añadir nuevos tipos, eliminar tipos existentes y reubicar tipos en otras ramas de la jerarquía.
- Modificar la información relativa tanto a ítems físicos como a tipos de ítems. Se le permitirá añadir, eliminar y modificar atributos de tipos, valores asignados en atributos y ubicaciones de un ítem.
- Incorporar en el sistema nuevos donantes. También se le permite eliminar o modificar la información relativa a otros donantes ya existentes.
- Incorporar nuevas donaciones al sistema y asociarlas a un donante y a un conjunto de ítems físicos. También podrá eliminar o modificar lo relativo a una donación existente.
- Añadir nuevas zonas y localizaciones al museo. También podrá eliminar o editar la información de alguna ya existente. Esta funcionalidad será la menos utilizada del sistema ya que las zonas y localizaciones del museo apenas cambiarán con el tiempo.

Se muestra a continuación un diagrama de las distintas vistas de las que dispondrá la aplicación web y como se relacionan entre ellas:

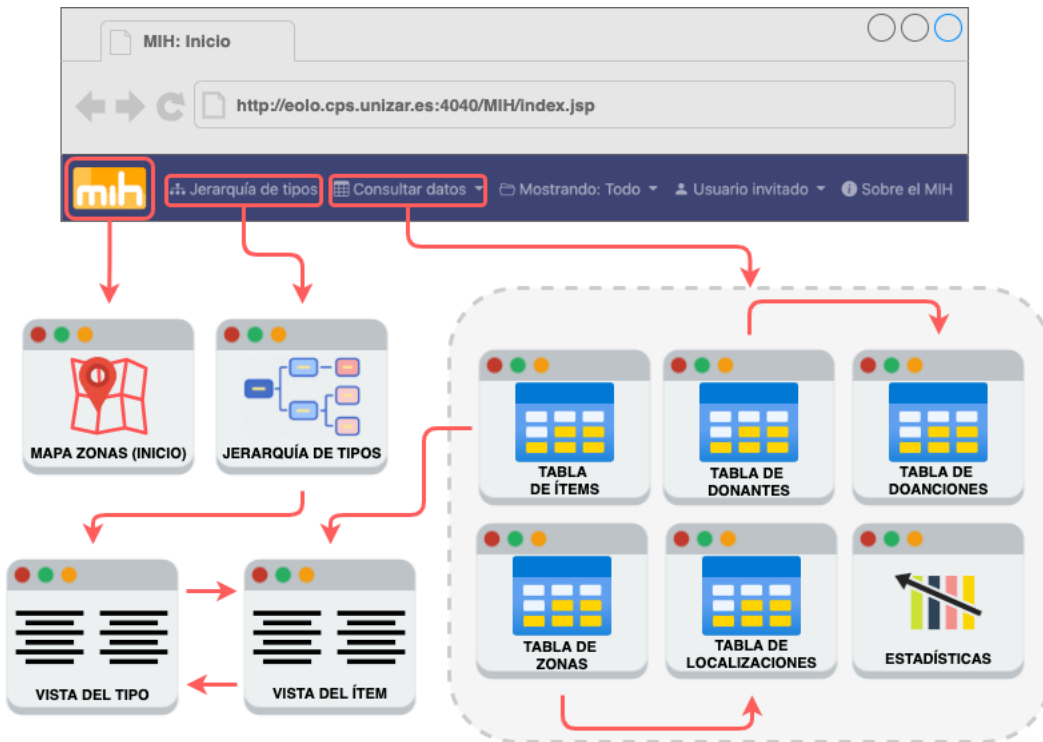


Figura 1: Diagrama con las vistas de aplicación web y su relación entre ellas

1.3. Tecnologías utilizadas

- **MySQL:** Gestor de base de datos utilizado por el sistema. En posteriores secciones se explican los motivos por los que se ha optado por esta solución.
- **Java:** Lenguaje de programación en el que está desarrollada la mayor parte de la aplicación web [7].
- **JavaScript:** Lenguaje de programación utilizado para hacer las distintas vistas de la aplicación más dinámicas [8].
- **Bootstrap:** Es un framework o biblioteca de código abierto basada en HTML, CSS y JavaScript. Contiene plantillas de diseño web y componentes que permiten agilizar el proceso de construcción del front-end [1].
- **JSPs y Servlets:** Tecnología utilizada para la construcción del back-end de la aplicación.

- **Datatables:** Se trata de un plug-in para JQuery de JavaScript que permite dar formato a una tabla HTML y añadirle funcionalidades como paginación, filtrado, ordenación, buscador, etc [2].

1.4. Estructura de la memoria

En primer lugar, en la sección 2 se trata todo lo relacionado con la base de datos. Se comenta el análisis realizado a los diferentes gestores del mercado, los motivos de la elección final y se comenta el esquema final de la base de datos y su evolución. En la sección 3, se habla de la arquitectura del sistema, es decir, como se ha diseñado y desplegado y que funcionalidades presenta. La sección 4 comenta las diferentes pruebas efectuadas al sistema. En la sección 5, se comentan las conclusiones extraídas tras la finalización del proyecto, se incluye un cronograma de tareas y se habla de posibles ampliaciones en la aplicación.

Para terminar, existen 6 anexos que contienen información detallada acerca de: análisis individual de los gestores (A), tablas SQL de la Base de datos (B), manual de usuario de las funcionalidades de la aplicación (C), contenido del fichero XML de configuración de las tablas visibles en la aplicación (D), contenido del fichero XML de variables constantes (E) y diagrama de Gantt completo con la evolución temporal de las tareas (F).

2. Diseño de la base de datos

En esta sección se analizan las características que ofrecen los diferentes gestores del mercado en el contexto de este problema antes de seleccionar uno, después se comentan los motivos de la elección y finalmente se comentan aspectos sobre el esquema final de la base de datos y su evolución.

2.1. Análisis de los SGBD del mercado

El problema principal del modelado de datos es la representación de los tipos de objetos como una jerarquía para que los tipos hijos heredasen los atributos de los padres. Por tanto, el objetivo del análisis fue encontrar un gestor que fuese capaz de crear una jerarquía de tablas donde cada tabla representaría un tipo, contendría sus atributos propios como campos y heredaría los atributos de las tablas superiores

gracias a las propiedades de la herencia. En la imagen 2 se muestra un esquema entidad-relación de como se pretendía modelar la herencia de tipos, donde la idea era que cada entidad que figura se convirtiese en una tabla en la base de datos. En este esquema cada ítem se relaciona con un único tipo de ítem, que este a su vez puede ser de varios subtipos, y los subtipos tendrían acceso a los atributos de toda la rama a la que perteneciesen.

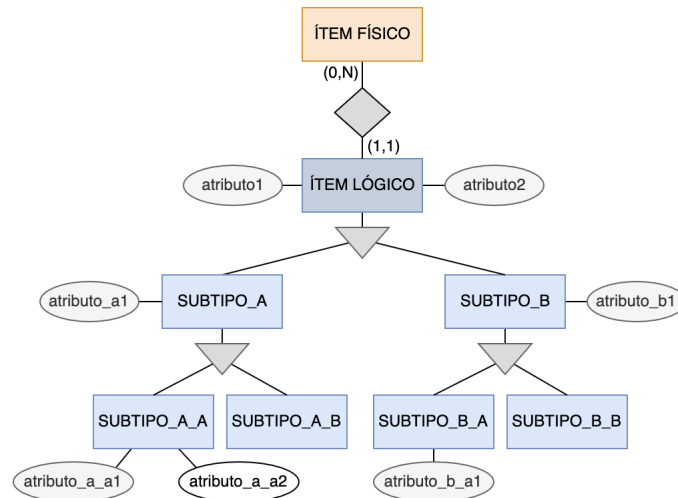


Figura 2: Ejemplo de un esquema entidad-relación para modelar la herencia de tipos de ítems con una tabla por tipo.

El gestor elegido, además de ofrecer un mecanismo de herencia de tablas propiamente dicho, debía tener los siguientes requisitos:

- Desde la tabla de un tipo hijo, poder acceder a sus atributos propios y a los atributos de las tablas de los ancestros.
- Las restricciones definidas en la tabla de un tipo se deben heredar para todas las tablas de los tipos herederos.

A continuación, se muestran en una tabla comparativa las distintas características que presentan respecto a la gestión de la herencia cada uno de los gestores analizados. En el Anexo A puede verse con más detalle un análisis individualizado de cada gestor.

Gestor	Tipo	Software libre	Herencia	Propagación restricciones	Integridad referencial
MySQL	Relacional	Sí	No tiene	—	—
MariaDB	Relacional	Sí	No tiene	—	—
Oracle	Relacional	No	Objetos	Sí	Sí
PostgreSQL	Relacional	Sí	Tablas	Parcial	No
DB2	Relacional	No	Objetos	Sí	Parcial
Informix	Relacional	No	Objetos	Sí	Parcial
CACHE	Orientado a Objetos	No	Clases	—	—
DB4O	Orientado a Objetos	Sí	Clases	—	—
MongoDB	Orientado a Documentos	Sí	No tiene	—	—

Cuadro 1: Comparativa de características de los principales gestores del mercado

2.2. Elección del SGBD

En este punto, finalizado el análisis de los gestores y habiendo visto lo que ofrecían, se tenía que elegir entre una de estas dos soluciones para solventar el tema de la herencia:

1. Optar por una implementación orientada a objetos que permita heredar restricciones entre la jerarquía a cambio de que dicha jerarquía sea más rígida en cuanto a cambios. Los gestores candidatos eran Oracle y DB2.
2. Optar por una implementación que permita una jerarquía de tablas flexible pero a cambio de perder la herencia de restricciones entre tablas. El mejor gestor candidato era PostgreSQL.

Ambas implementaciones tenían problemas que solventar y había que pensar cual supondría menos impacto. Con la primera solución estaba el problema de que cada vez que se quisiera modificar la jerarquía habría que hacer una copia de seguridad de los datos de las tablas, modificar el esquema y después rehacer la base de datos. Con la segunda solución aparecía el problema de que para solventar la no herencia de restricciones entre tablas habría que implementar triggers que se encargasen de

propagarlas de forma automática cada vez que se añadiese una tabla nueva a la jerarquía.

El objetivo de la herencia era poder establecer una jerarquía de tipos de ítems almacenados en el museo, pudiendo así cada tipo tener una serie de atributos a los que dar valor y que los heredasen sus descendientes. El fin no es otro que a la hora de registrar un nuevo ítem donado, el encargado del museo tenga que hacer el menor esfuerzo posible para registrarlo al poder utilizar la información jerárquica.

Como ninguna de las dos opciones resultaba convincente y ningún gestor ofrecía un soporte a la herencia de tablas con garantías, surgió una nueva idea para modelar la jerarquía de tipos sin necesidad de hacer uso de los mecanismos de herencia en el gestor, que tras estudiarla se acabó considerando la mejor opción. La idea era poder mantener una jerarquía a partir de una tabla que almacenase los meta-datos de dicha jerarquía y sin necesidad de usar lo ofrecido por el gestor para controlar la herencia, es decir, una tabla que guardasen toda la información necesaria para inferir la herencia.

Por ejemplo, para modelar un tipo de ítem con dos tipos hijos, en lugar de tener tres tablas relacionadas con el mecanismo de herencia del gestor, tener una única tabla donde insertar tuplas [*nombre_del_tipo, referencia_al_tipo_padre*] que indiquen esta relación, y después guardar los atributos de cada tipo en otra tabla de manera similar con tuplas [*nombre_del_tipo, nombre_del_atributo, valor_del_atributo*].

Finalmente, en lo que respecta al gestor utilizado, como con la nueva solución ya no era necesario el uso de mecanismos de herencia propios de los gestores, se optó por usar MySQL. Los motivos los siguientes:

- Es uno de los gestores SQL más utilizados y que cuenta con una gran comunidad de desarrolladores que lo utilizan.
- Es software libre, ya dispone de una licencia GPL (General Public License).
- A nivel personal ya había trabajado con dicho gestor en diversas ocasiones.
- El servidor donde se iba a alojar tanto la aplicación como el gestor de bases de datos ya tenía instalado y configurado MySQL.

2.3. Esquema de la base de datos

En esta sección se va a mostrar en primer lugar la versión final del esquema de la base de datos y después se comentará la evolución del mismo desde que se tuvo la

primera idea y se comenzó a trabajar con él. Pero antes, destacar que para diseñar el esquema se partió de una antigua versión del mismo diseñada por Daniel Martínez García. Daniel fue un alumno que comenzó a desarrollar este mismo TFG en el año 2012 pero que no llegó a terminarlo y lo abandonó. Dicho esquema sirvió como punto de partida y de él se obtuvieron ideas importantes como: tener ítems y tipos de ítems (Daniel los denominaba ítem físico e ítem lógico), tener un histórico de ubicaciones para un ítem o la idea de establecer una relación jerárquica para los tipos de ítems. Hubo otras ideas de este esquema que se acabaron desechando finalmente ya que no se consideraban muy adecuadas como por ejemplo: Tener triggers para asegurar la integridad referencial en la herencia (ya que utilizaba el gestor PostgreSQL) o tener atributos propios en la entidad Ítem Físico. Con todo este análisis se comenzó a desarrollar el nuevo esquema.

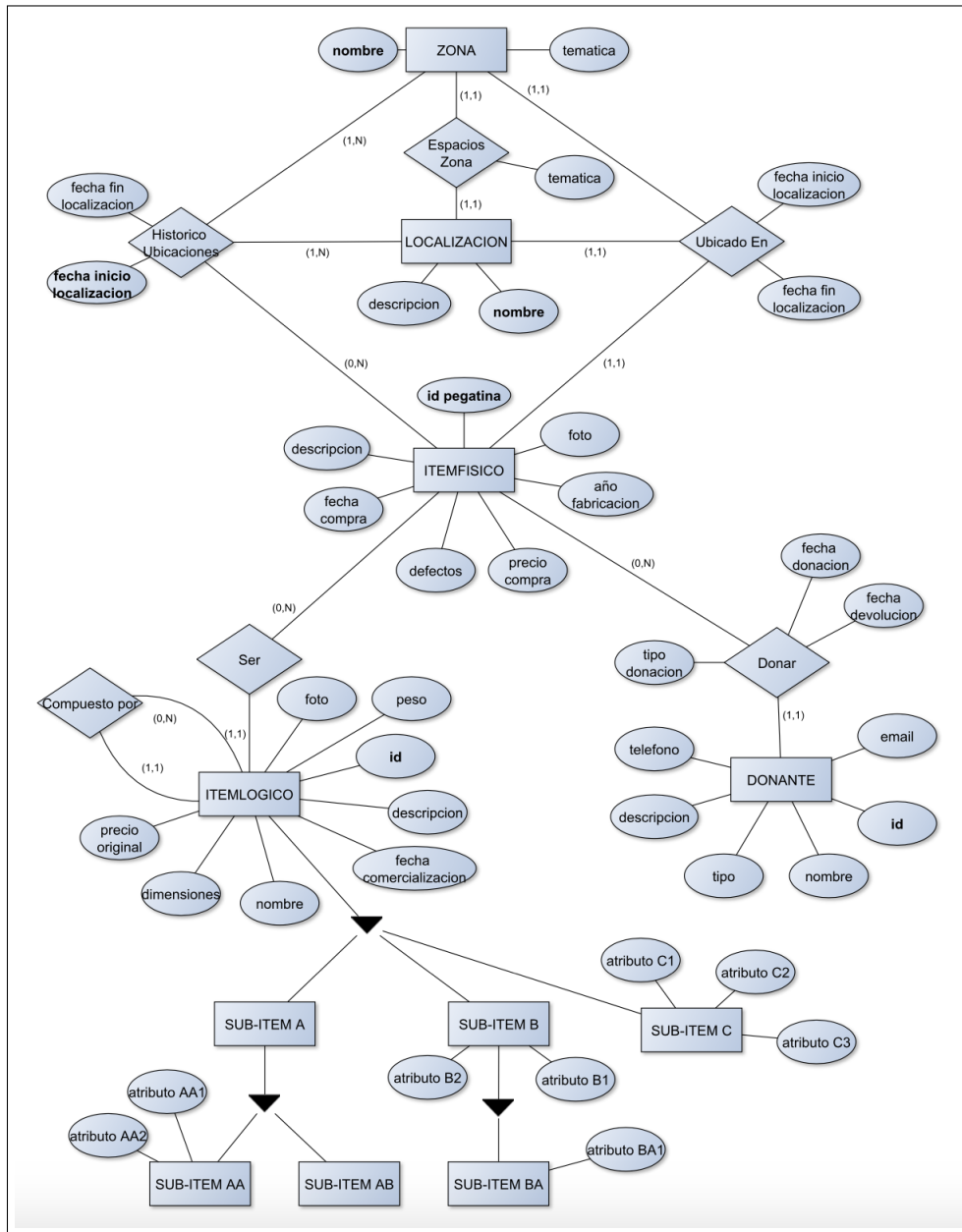


Figura 3: Esquema Entidad-Relación diseñado por Daniel en 2012

En las siguientes subsecciones se muestra el esquema de la base de datos completo en su versión actual. También se comentan las distintas etapas por las que este ha pasado desde que se comenzó a trabajar con él.

2.3.1. Esquema Entidad-Relación

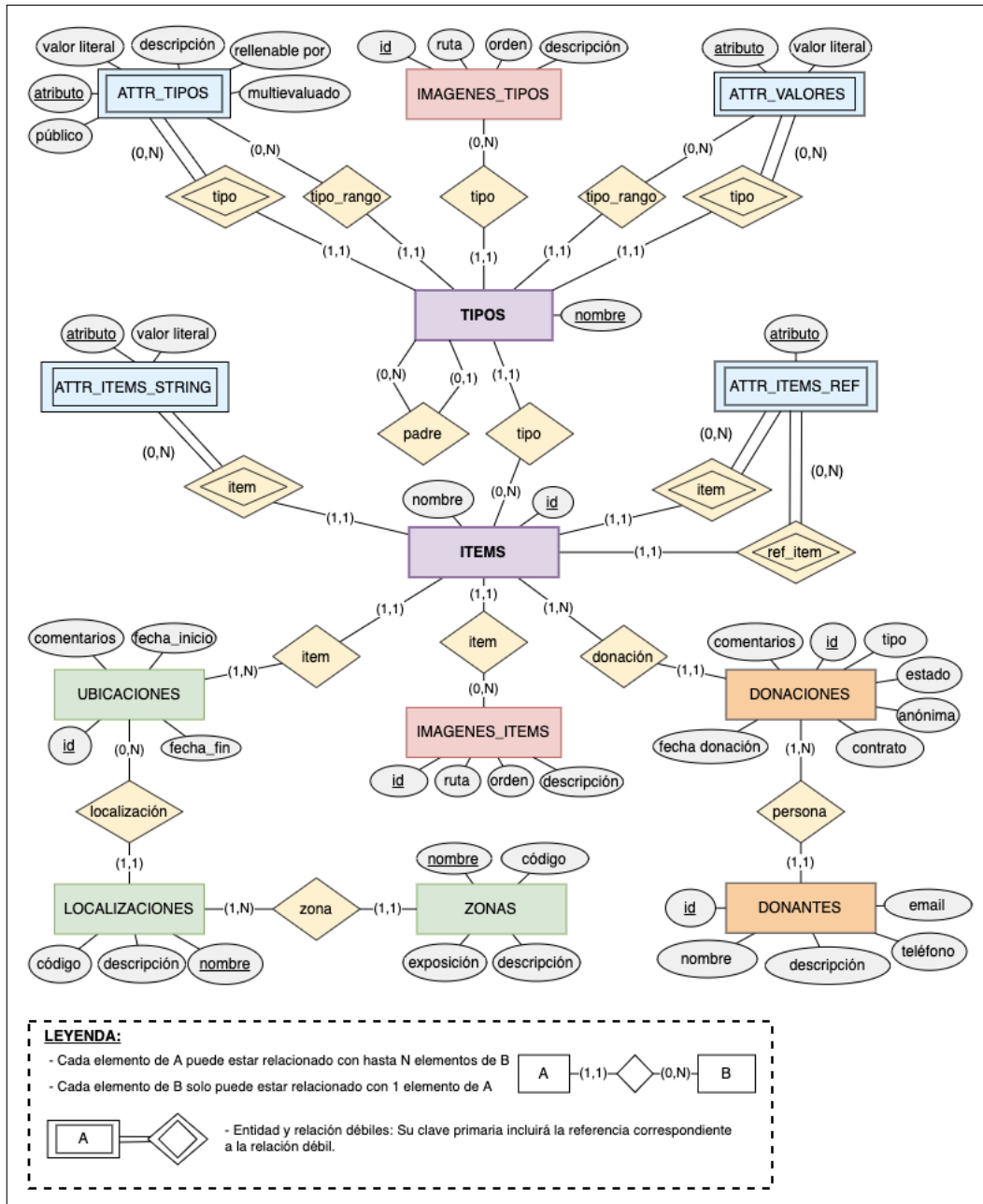


Figura 4: Esquema Entidad-Relación de la Base de Datos

A continuación se aporta una breve explicación de cada entidad:

- **Tipos:** Almacena de forma jerárquica los diferentes tipos de ítems que se declaran en la aplicación, guardando su nombre y una referencia a su tipo que actúa como padre jerárquico.
- **Donantes:** Almacena información relativa a las personas que han realizado una donación al museo al menos una vez.
- **Donaciones:** Almacena información relativa a una donación de uno o varios ítems realizada por un donante.
- **Ítems:** Almacena cada uno de los ítems físicos donados al museo y lo asocia a un tipo concreto.
- **Imágenes Ítems:** Almacena las rutas de aquellas imágenes asociadas a un ítem.
- **Imágenes Tipos:** Almacena las rutas de aquellas imágenes asociadas a un tipo.
- **Attr Tipos:** Almacena los atributos definidos por cada tipo. Recalcar que en esta entidad solo se almacenan los atributos propios junto con su valor si es que lo tienen, los heredados nunca estarán aquí.
- **Attr Valores:** Almacena únicamente los valores que un tipo le asigna a aquellos atributos que ha heredado de otros supertipos de forma jerárquica.
- **Attr Ítems String:** Almacena únicamente los valores dados por un ítem a un atributo heredado que sea de rango 'STRING', es decir, cuando el valor que puede adoptar el atributo represente valores literales.
- **Attr Ítems Ref:** Almacena únicamente los valores dados por un ítem a un atributo heredado que sea de rango 'Tipo', es decir, cuando el atributo tenga el rango restringido por un tipo de ítem y el valor que puede adoptar es una referencia a otro ítem del tipo del rango.
- **Zonas:** Almacena las distintas zonas en las que se divide el museo.
- **Localizaciones:** Almacena los distintos puntos en los que se pueden localizar ítems en el museo (cada localización pertenece a una zona concreta).

- **Ubicaciones:** Almacena el historial de localizaciones por las que ha pasado un ítem físico desde que se donó.

Comentar también algunas restricciones que presenta el esquema y que se solucionan en el lado del servidor:

- El periodo temporal de cada ubicación asociada a un ítem no debe interferir en el periodo de otra ubicación antigua a la hora de añadir nuevas o modificar existente.
- Cada ítem tendrá una ubicación inicial por defecto localizada en el almacén.
- Existirá un donante predefinido denominado 'Donante desconocido' asociará por defecto a aquellas donaciones en las que no se indique donante.
- No se podrá repetir el mismo nombre de atributo en la rama jerárquica de un tipo.
- Los nombres de atributos almacenados en las entidades 'Attr Valores', 'Attr Ítems String' y 'Attr Ítems Ref' deberán coincidir siempre con los nombres definidos en 'Attr Tipos'.

2.3.2. Esquema Relacional

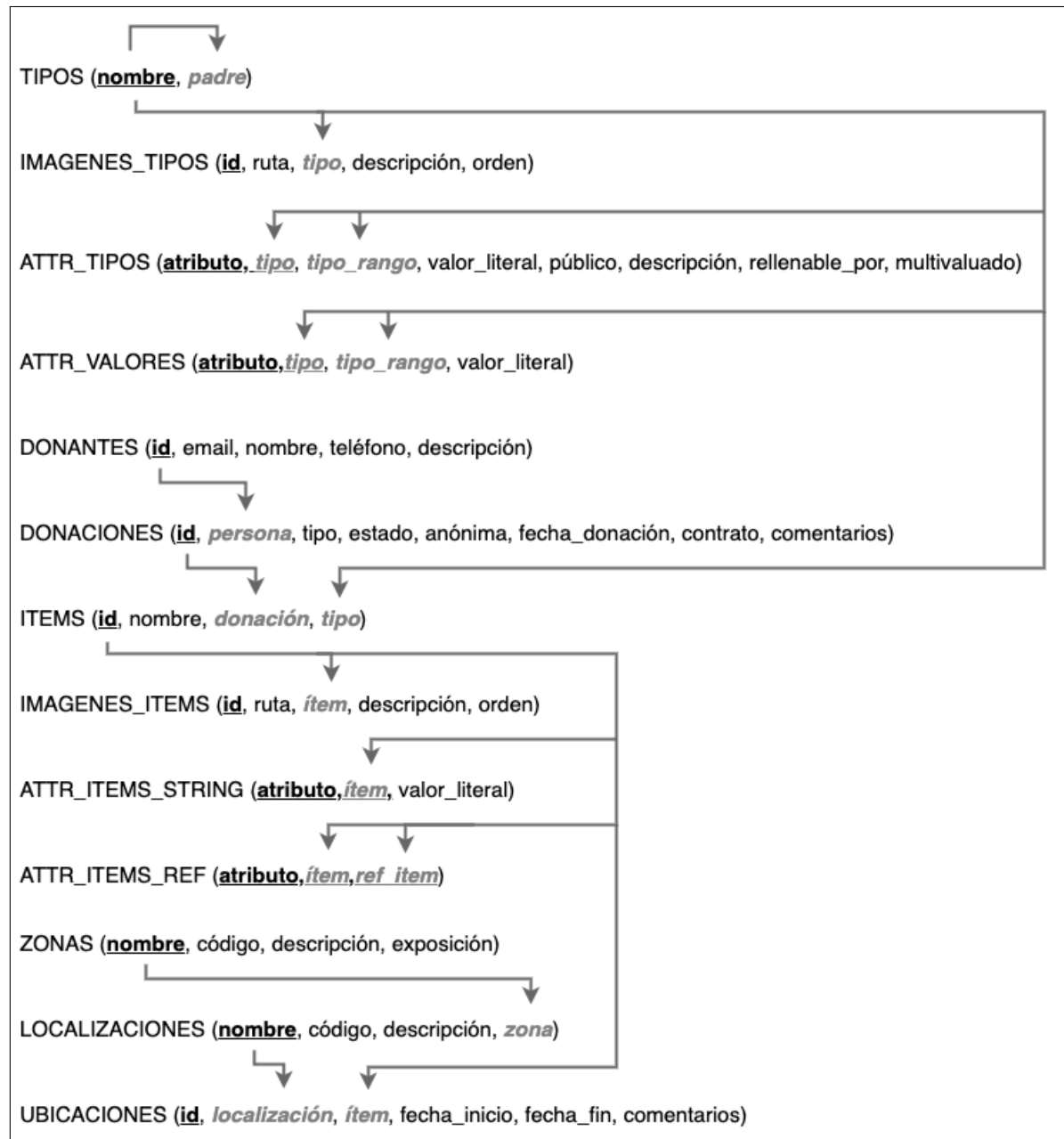


Figura 5: Esquema Entidad-Relación de la Base de Datos

Respecto a la normalización del esquema:

- En primer lugar, se ha comprobado que cumple la primera forma normal ya que ninguna relación presenta atributos multivaluados.
- El modelo no presenta dependencias de parte de la clave en las relaciones ya que en las relaciones con claves compuestas, ninguno de sus atributos puede ser obtenido con solo una parte de la clave primaria, por estos motivos se puede afirmar que cumple la segunda forma normal.
- Se ha comprobado también que el modelo no presente dependencias de atributos que no sean clave con otros de la misma relación, de esta manera se confirma que cumple la tercera forma normal.
- Para finalizar la normalización, se ha comprobado que cumple forma normal de Boyce-Codd ya que en todas las relaciones del modelo las únicas dependencias existentes son las de la clave primaria.

Como resultado se obtiene que el modelo relacional final ya está normalizado. Puede verse una explicación más detallada de cada una de las tablas generadas para la base de datos en el Anexo B.

2.3.3. Evolución del esquema

El esquema anterior representa la versión final de la base de datos utilizada por la aplicación, pero como se puede suponer, previamente hubo otras versiones que se acabaron descartando o que evolucionaron hasta llegar a la actual.

Las entidades que representan los ítems, las donaciones, los donantes, las zonas, las localizaciones, las ubicaciones de un ítem y las imágenes de tipos e ítems se tuvieron claras desde el inicio como se querían modelar y no han sufrido apenas cambios desde entonces. El cambio más significativo fue la incorporación de un campo 'código' en las localizaciones y donaciones y de un campo 'orden' en las imágenes, con el fin de mostrar la información en la web con un orden preestablecido y configurable.

La entidad de los tipos tuvo dos etapas: La primera, en la que se pretendía usar un gestor que permitiese herencia de tablas y por tanto existiría una tabla por tipo creado en el sistema. La segunda y la actual, en la que decidió inferir la herencia de tablas guardando los meta-datos en una única tabla, dando lugar a una única tabla donde guardar tuplas con el nombre del tipo y una referencia al tipo padre.

El tema de los atributos de los tipos fue sin duda, junto con la herencia de los tipos, lo más difícil de modelar y lo que supuso más tiempo. Al inicio, cuando se tenía la idea de tener una tabla por tipo, modelar los atributos era relativamente sencillo ya que cada uno sería un campo de la tabla del tipo que lo definiese y las tablas hijas los heredarían por las propiedades de la herencia del gestor. El problema estaba en definir campos en las tablas que referenciaban a otras tablas (necesarios para los atributos con rango 'Tipo') ya que las restricciones de integridad referencial no eran soportadas por el gestor elegido, este fue uno de los motivos que propició el cambio en la forma de modelar la herencia. Con la solución de la herencia de tipos modelada con una sola tabla, el modelado de los atributos pasó por varias versiones:

1. Una única tabla donde almacenar tanto los atributos que se definían como los valores asignados a ellos por su tipo o por un subtipo.
2. Se decidió dividir la tabla anterior en dos, en una se almacenaría la definición de atributos y en otra la asignación de valores.
3. Cuando se decidió que los ítems también pudiesen asignar valores a atributos de su tipo o de un tipo ancestro jerárquicamente, la tabla para almacenar la asignación de valores se quedaba pequeña, por tanto se destino esta para la asignación en tipos y se creó otra para la asignación en ítems.
4. Con la implementación del sistema bastante avanzada, se decidió que un ítem no solo podía asignar valores a atributos de rango STRING sino que también podía hacerlo con atributos de rango Tipo (el valor pasaba de ser un literal a ser una referencia a un ítem). Este cambio supuso añadir una nueva tabla para almacenar solo la asignación en ítems de atributos de rango Tipo y dejar la que había para almacenar solo la asignación en ítems de atributos de rango STRING.

Destacar también que cuando la aplicación estaba bastante avanzada surgió un debate acerca del uso de tablas MyIsam o InnoDB en MySQL. En un principio se construyó la base de datos con tablas MyIsam ya que están más optimizadas para recibir una gran cantidad de lecturas y pocas escrituras, que es lo que hace esta aplicación la mayor parte del tiempo. El problema vino al descubrir que este tipo de tablas no aseguraban la integridad referencial. Por tanto se tuvo que elegir entre mantener MyIsam y asegurar la integridad con triggers o con código en el backend, o pasarse a tablas InnoDB que si ofrecen integridad referencial a cambio de perder un poco de eficiencia. Se optó finalmente por la última opción.

3. Arquitectura del sistema de información para el MIH

La arquitectura del sistema es sencilla, tanto el servidor web con la aplicación como el gestor de la base de datos se encuentran desplegados en una misma máquina ubicada en el edificio Ada Byron, a la cuál el director de este proyecto (y administrador del museo) tiene acceso tanto de forma física como remota.

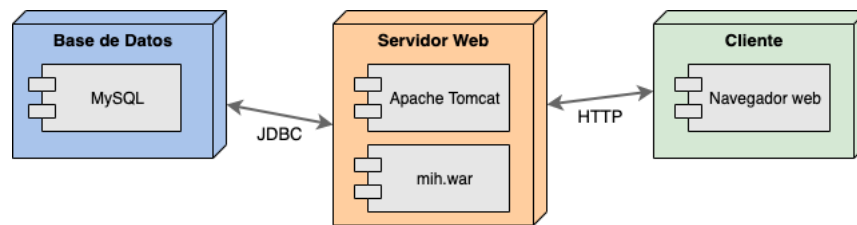


Figura 6: Diagrama de despliegue del sistema

El gestor de la base de datos utilizado es MySQL y se ha optado por usar la versión 5.1 ya que esta se encontraba ya preinstalada en la máquina. El servidor web utilizado es Apache Tomcat y se utiliza la versión 7.0 ya que también se encontraba preinstalada. Al desarrollar la aplicación en Java, la conexión entre la base de datos y la aplicación se realiza mediante JDBC. En lo referente a los clientes de la aplicación, interactuarán con ella mediante HTTP desde un navegador web.

Se trata de una aplicación que a nivel de interfaz gráfico es sencilla, en la que los datos almacenados no serán muy pesados y que la mayor cantidad de peticiones realizadas por clientes serán consultas a la base de datos, por tanto, la potencia requerida a nivel hardware no es elevada. Por estos motivos se ha considerado suficiente utilizar la máquina nombrada y las versiones citadas tanto de gestor como servidor, pese a tener varios años y estar ligeramente obsoletos. Además, dichas versiones obsoletas son más ligeras que las modernas y consumen menos recursos.

3.1. Diseño de la aplicación

El lenguaje utilizado ha sido Java tanto para el lado del servidor como del cliente y HTML y JavaScript para el lado del cliente. Además, para desarrollar las interfaces se ha hecho uso del framework Bootstrap.

Para desarrollar la aplicación se ha hecho uso del patrón MVC (Modelo Vista Controlador). Con este patrón, la vista (lo que ve el usuario) interactúa de forma indirecta con el modelo (lo que representa a los datos almacenados) a través de controladores, que recibe peticiones del cliente a través de las vistas y las redirige al modelo en la forma correspondiente.

Para el acceso a datos se ha hecho uso del patrón DAO (Objeto de Acceso a Datos). Este patrón crea una capa intermedia entre la aplicación y la base de datos, proporcionando una abstracción del acceso a la base de datos. Esto permite independizar la base de datos de aplicación, haciendo más fácil por ejemplo la labor de migrar los datos entre distintos gestores.

Los fuentes Java de la aplicación se dividen en 3 paquetes distintos:

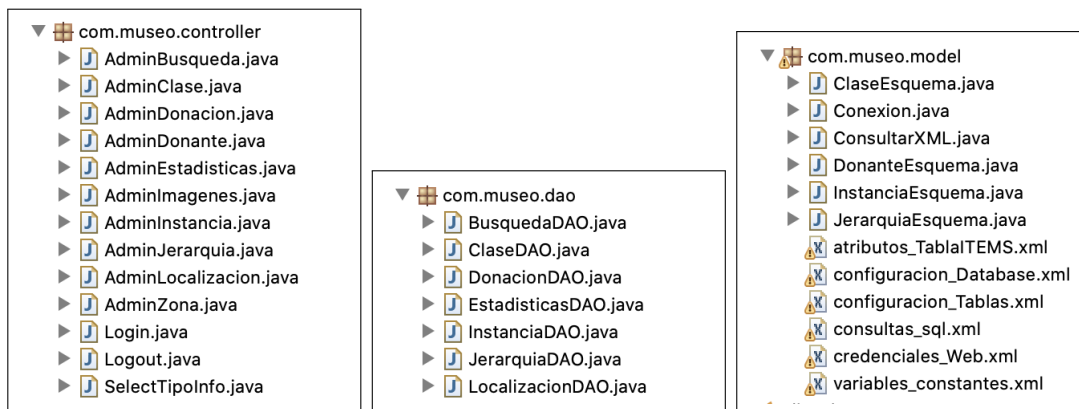


Figura 7: Paquetes de la aplicación Java y sus ficheros.

- **com.museo.controller:** Contiene todas las clases Java que actúan como servlets en la aplicación. Estas clases se encargan de recibir peticiones de los clientes y procesarlas. Existe un servlet para cada vista de la aplicación.
- **com.museo.dao:** Contiene las clases encargadas de acceder a los datos, es decir, los DAOs. Se invocan desde los servlets y permiten interactuar con la base de datos. Se ha generado un DAO para cada conjunto de información similar.
- **com.museo.model:** Contiene clases que representan una determinada información de la base de datos que quiere tratarse en conjunto. Se generan en los servlets o en los DAO y permiten crear objetos que agrupan información y que después se pasan a las vistas para mostrarlos al usuario. En este paquete se

encuentra también la clase encargada de establecer la conexión con la base de datos a través de JDBC, una clase con todas las consultas en formato texto que se utilizan en el sistema y por último, diversos ficheros XML que contienen información de configuración. Estos ficheros permiten cambiar aspectos de configuración de la aplicación sin necesidad de detenerla y volverla a lanzar ya que la aplicación obtiene la información de ellos cada vez que la necesita. Los ficheros XML son los siguientes:

1. ***atributos_TablaItems.xml***: Fichero que contenía el nombre de aquellos campos definidos en la tabla ITEMS de la base de datos y que actuaban como atributos comunes para cualquier ítem. Contenía también para cada atributo un indicador de si era público o privado. Actualmente este fichero no se usa en la aplicación pero se ha querido mantener por si acaso.
2. ***credenciales_Web.xml***: Fichero que contiene el nombre de usuario y la contraseña del administrador de la aplicación. Es el fichero que se consulta cada vez que se trata de iniciar sesión como administrador.
3. ***configuracion_Database.xml***: Fichero que contiene diversas variables de configuración para establecer la conexión con la base de datos. Almacena el nombre del tipo de JDBC, el nombre del driver de MySQL, la IP y puerto de la base de datos, el nombre de la base de datos y el nombre de usuario y contraseña para acceder a ella. También se ha creado una variable en este fichero llamada 'Modo' (puede tener los valores 'desarrollo' o 'produccion') que permite cambiar entre dos bases de datos distintas, lo que permite cambiar rápidamente la fuente de datos de la aplicación. La utilización de esta variable fue muy útil durante el desarrollo, ya que para desarrollar se usaba una base de datos local y después al desplegar la aplicación en el servidor se cambiaba a la base de datos del servidor.
4. ***configuracion_Tablas.xml***: Guarda información de configuración relativa a las tablas con información detallada de ítems, donantes, donaciones, localizaciones, zonas, resumen de donantes y resumen de donaciones (las 2 últimas se encuentran en la vista de estadísticas). Se puede ver el fichero completo en el Anexo D. Las variables disponibles para cada tabla son las siguientes:
 - nombresAtrPrincipales: Nombres de los atributos principales.
 - nombresAtrSecundarios: Nombres atributos secundarios.
 - inputTypesAtr: Tipo de input de todos los atributos para el formulario.

- `infoAdicionalInputTypes`: Información adicional para cada input anterior.
 - `comentariosCampos`: Comentarios explicativos acerca de cada uno de los atributos.
 - `columnSize`: Tamaño de cada columna visible de la tabla.
 - `inputSize`: Tamaño de cada input del formulario.
 - `columnaOrden`: Indica que índice de columna es el que marca el orden de la tabla.
 - `tipoOrden`: Indica si el orden es ascendente (`asc`) o descendente (`desc`).
5. ***consultas_sql.xml***: Contiene todas las consultas en lenguaje SQL que se utilizan en la aplicación.
 6. ***variables_constantes.xml***: Almacena el valor de diferentes constantes que se utilizan en la aplicación, principalmente constantes usadas en los JSPs. El objetivo es que cualquier constante que aparezca en el futuro al realizar alguna mejora en la interfaz pueda ubicarse aquí. Se puede ver el fichero completo en el Anexo E.

Para desarrollar las vistas de la aplicación se ha utilizado la tecnología JSP (Java Server Page). Que permite combinar el lenguaje HTML con el lenguaje Java, además de poder incorporar JavaScript. Esto permite generar páginas para el usuario que sean dinámicas. Existe un JSP por cada una de las vistas, excepto en el caso de las tablas de ítems, donantes, donaciones, zonas y localizaciones, para ellas se ha utilizado un JSP común en todas ellas.

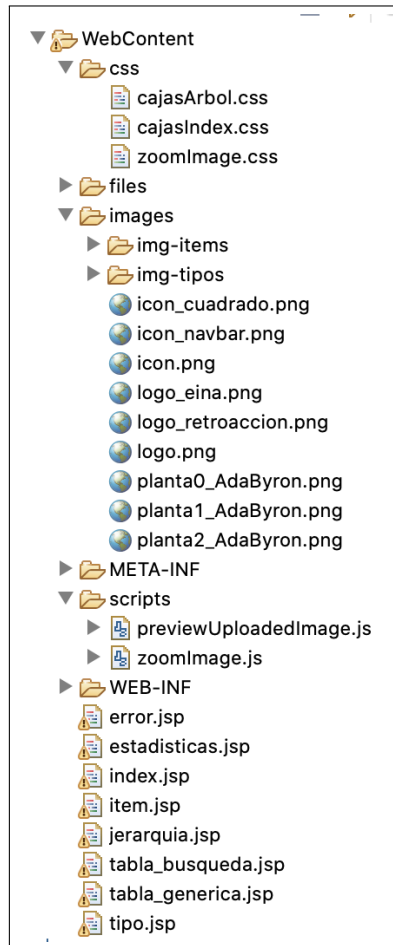


Figura 8: Directorio de la aplicación Java con los JSPs y otros subdirectorios importantes.

Otros subdirectorios destacables de la aplicación:

- **css:** Almacena las fuentes de estilo CSS que pueden ser utilizadas en la aplicación en más de un punto. Por ejemplo: Para las cajas del árbol jerárquico de tipos, para las zonas y vitrinas de la página de inicio o para el zoom con el que se muestra una imagen al pulsar en ella.
- **scripts:** Contiene el código JavaScript que se utiliza tanto para mostrar una imagen con zoom como para previsualizar una imagen en el formulario de subida.

- **files:** Almacena los ficheros PDF subidos que representan el contrato de una donación.
- **images:** Contiene las imágenes que aparecen en la web, como los iconos del MIH por ejemplo. También se subdivide en los directorios 'img-items' y 'img-tipos', donde se almacenan las imágenes subidas que se asocian o bien a un tipo o bien a un ítem.

3.2. Funcionalidades de la aplicación

Podemos dividir las funcionalidades que la aplicación ofrece en dos grupos:

- **Funcionalidades de usuario:** Aquellas que pueda realizar cualquier visitante de la web (incluido el administrador). Son las siguientes:
 1. **Desde la barra de navegación:** Redirigir a la página de inicio, a la jerarquía de tipos, a la ventana de inicio de sesión, a la página original del MIH y a las tablas con información detallada (con información de ítems, zonas, localizaciones y tablas y gráficos estadísticos). También hay accesible un buscador, que busca por nombres o por valores de atributos, y un selector del tipo de información a consultar en el sistema (cambiar entre exposición, en almacén o todo).
 2. **Desde la página de inicio:** Ver el plano de cada planta, acceder a la página detallada de una zona y ver el listado de ítems de una vitrina.
 3. **Desde la vista del árbol de tipos:** Visualizar una sola rama, visualizar el árbol completo, ocultar/mostrar subtipos y acceder a la vista detallada de un tipo.
 4. **Desde la vista de un tipo:** Observar sus imágenes, su tabla de atributos, el árbol de tipos herederos y el listado de ítems (propios y de los herederos).
 5. **Desde las vistas de tablas con información detallada:** Observar toda la información referente a ítems, localizaciones y zonas en formato de tabla con un formulario para ver información adicional.
 6. **Desde la vista de información estadística:** Observar una tabla con el resumen de donaciones de cada donante, otra tabla con el resumen de donaciones anual y una gráfica que muestra una evolución del año de lanzamiento de cada ítem catalogado.

7. **Desde la vista de un ítem:** Observar las imágenes, la tabla de atributos del tipo, información sobre la donación y una tabla con el histórico de ubicaciones.
- **Funcionalidades específicas de administrador:** Aquellas que únicamente podrán llevarse a cabo por el administrador del sistema. Requerirán de un proceso de identificación previo (inicio de sesión en la web como administrador a través de usuario y contraseña) y son las siguientes:
 1. **Desde la vista del árbol de tipos:** Añadir un tipo hijo, eliminar un tipo o tipos y reubicar un tipo o tipos bajo un nuevo padre.
 2. **Desde la vista de un tipo:** Cambiar su nombre, añadir/eliminar imágenes, añadir nuevo ítem, añadir/eliminar/editar un atributo propio, añadir/editar/eliminar un valor literal a un atributo y añadir/editar/eliminar una restricción de rango para un atributo (si es de rango Tipo y no STRING).
 3. **Desde las vistas de tablas con información detallada:** Añadir/actualizar/eliminar una de las entradas de la tabla a través del formulario. También se puede desde la lista de ítems de una donación vincular/desvincular uno de los ítems de dicha donación.
 4. **Desde la vista de un ítem:** Cambiar su nombre, cambiar su tipo asociado, eliminar dicho ítem del sistema, añadir/eliminar imágenes, añadir/editar/eliminar un valor literal a un atributo, añadir/editar/eliminar una referencia a otro ítem para un atributo (si es de rango Tipo), añadir/eliminar una ubicación, editar la localización de una ubicación, editar la fecha de inicio/fin de una ubicación y editar los comentarios de una ubicación.

Todas las funcionalidades pueden verse de forma más detallada (con capturas de pantalla incluidas) en el Anexo C, pero cabe hacer una mención especial en esta sección dada su complejidad, al tema de los rangos de los atributos, su configuración, y toda la casuística que estos generan, que es la siguiente:

1. **Si el rango es STRING:** Hay que indicar quien asignará valores (que serán literales) al atributo, para esto hay dos opciones:
 - a) Si se elige '**Tipo**' entonces podrá asignar un valor literal en el tipo o en un subtipo.
Por ejemplo: el atributo 'año de comercialización' del tipo 'Ítem lógico'.

- b) Si se elige '**Ítem**' solo se le podrá asignar un valor literal desde un ítem del propio tipo o desde un ítem de los tipos herederos.
Por ejemplo: el atributo 'defectos' del tipo 'Ítem lógico'.

AÑADIR UN ATRIBUTO

Esta acción añadirá un nuevo atributo al tipo actual. Una vez confirmada la acción, el cambio será persistente.

Indique el nuevo nombre del atributo:

Seleccione el rango del atributo:

String Tipo

Indique quién asignará el valor de este atributo:

Tipo Ítem Ninguno

Indique ahora un valor o hágalo más tarde en este u otros subtipos:

Añada si lo desea una descripción semántica para este atributo:

Cancelar Aceptar

Figura 9: Opciones de configuración de un atributo con rango STRING.

2. **Si el rango es un tipo:** Hay que seleccionar del árbol de tipos uno para que actúe como rango raíz. Después hay que indicar quien asignará valores (que serán referencias a ítems) al atributo, para esto hay dos opciones:

a) Si se elige '**Ninguno**' el atributo no tendrá valores asociados ya que se considera que el rango y sus posibles restricciones aportan ya suficiente información.

Por ejemplo: el atributo 'microprocesador' del tipo 'Ordenador', ya que no interesa guardar todos los microprocesadores internos como ítems aparte, solo interesa conocer el tipo al cual pertenece.

b) Si se elige '**Ítem**' se podrán asignar valores únicamente en los ítems (dicho valor consistirá en la referencia aun ítem del tipo marcado por el rango o su restricción). Habrá que indicar también si será **multivaluado** o no (en caso de ser multivaluado, se permitirá referenciar a uno o más ítems desde el mismo atributo).

Por ejemplo: el atributo multivaluado 'contenido' del tipo 'Software', ya que interesa relacionar por ejemplo el ítem de un videojuego con su soporte físico (unos floppies), su caja, sus manuales, etc.

AÑADIR UN ATRIBUTO

Esta acción añadirá un nuevo atributo al tipo actual. Una vez confirmada la acción, el cambio será persistente.

Indique el nuevo nombre del atributo:

Seleccione el rango del atributo:

String Tipo

[Seleccionar Tipo](#)

Indique quién asignará el valor de este atributo:

Tipo Ítem Ninguno

Indique si el atributo será multivaluado:

Si No

Añada si lo desea una descripción semántica para este atributo:

[Cancelar](#) [Aceptar](#)

Figura 10: Opciones de configuración de un atributo con rango Tipo.

4. Pruebas efectuadas al sistema

A continuación, se comentan las pruebas realizadas sobre el sistema para asegurar su correcto funcionamiento futuro:

4.1. Validación

Durante el proceso de desarrollo de la aplicación que comenzó en marzo y ha terminado en noviembre, se han ido realizando pruebas de validación manuales para cada una de las funcionalidades. La metodología al acabar la implementación de una nueva funcionalidad era probarla a través de la interfaz y corregir los errores detectados. Con esto se conseguía que la aplicación desplegada en el servidor de la Universidad siempre estuviese probada. Si se detectaban errores, no se lanzaba una nueva versión de la aplicación hasta que se solucionasen o se mitigasen de la mejor forma.

En las semanas finales del desarrollo de la aplicación, también se ha probado la aplicación mediante la incorporación al sistema de datos reales del museo, con el objetivo de ver como se comportaba la aplicación en su entorno real y poder detectar posibles fallos no vistos hasta entonces. Se catalogaron un total de 21 ítems correspondientes a la vitrina central 0.2 de la zona 'Ordenadores domésticos', dedicando un total de 5 horas y media para completar la catalogación en el nuevo sistema. Esto demuestra lo tedioso y costoso en tiempo que puede llegar a ser la catalogación.

Destacar un problema surgió al catalogar la caja de cartón vacía del juego 'Lemmings' para el ordenador 'Commodore Amiga', que se encontraba expuesto en la vitrina: La caja representa el juego de cara al público, ya que cuando alguien la observa la relaciona directamente con el juego, pero lo que realmente contiene el juego físico son los floppies de 3.5 que vienen dentro de la caja y que actúan como soporte informático del videojuego (además en esta ocasión no estaban dentro de la caja sino guardados en el almacén). Ni la caja ni los floppies son el videojuego, ya que el videojuego realmente es un software, pero se utilizan para representarlo debido a sus características físicas (portada, título, serigrafía, etc).

Por tanto al catalogar este ítem aparece el debate entre guardar la caja como un ítem de algún subtipo de 'videojuego' o crear un nuevo tipo 'caja de cartón' para este ítem y referenciar la caja desde otro ítem más abstracto creado en un tipo específico para el videojuego denominado 'Lemmings (Amiga)' que es heredero del tipo 'Videojuego'. En esta ocasión se optó por la segunda opción, pero ambas podrían

ser válidas y es común que para unos objetos interese mejor una opción y para otros la otra dadas sus características.

Por tanto, la catalogación muchas veces conlleva tiempo debido a los problemas conceptuales propios de la catalogación y a la reflexión que hay que hacer respecto a la jerarquía de los tipos. ¿Cuándo se debería crear un nuevo tipo? ¿cuál es el sitio más adecuado para un cierto atributo en la jerarquía de tipos? son preguntas que no tienen fácil respuesta en algunos casos. También será común que la clasificación inicial que se hace de los tipos no sea la más adecuada pasado un tiempo y que haya que reorganizarla, por este motivo resulta tan importante el tener una jerarquía dinámica.

4.2. Sobrecarga y eficiencia

Una de las partes que se ha detectado como más crítica en cuanto a sobrecarga de datos ha sido el mostrar el árbol jerárquico de tipos. El crecimiento que este puede llegar a tener con el paso de los años puede ser elevado y conviene que la forma de generarlo sea eficiente.

La primera versión (V1) de la función generadora del árbol consultaba para el tipo raíz los hijos, después para cada hijo consultaba de nuevo sus hijos, y así lo hacía de forma recursiva hasta obtener todo el árbol haciendo un recorrido en profundidad. El problema que presentaba esta solución era el elevado número de consultas que se hacían contra la Base de Datos, lo que hacía que mostrar el árbol fuese más lento conforme crecía.

La segunda versión (V2) de la función generadora del árbol solucionó el problema de la anterior. Esta versión obtiene mediante una sola consulta a la base de datos tuplas *[padre,hijos]* y las almacena en una tabla Hash. Después recorre dicha tabla Hash de forma recursiva para generar el árbol como en la versión anterior. Con esta nueva versión se elimina el problema del número elevado de consultas y hace que mostrar el árbol sea casi instantáneo.

Las pruebas de rendimiento realizadas con ambas versiones de la función generadora han consistido en mostrar árboles jerárquicos con distintos niveles (el nivel de la raíz no se cuenta) y con distinto número de hijos por nivel y calcular el tiempo en segundos que tarda la aplicación en mostrarlo en pantalla. Los resultados obtenidos han sido los mostrados en el cuadro 4.2.

Niveles	Hijos/Nivel	Total nodos	Tiempo V1 (seg.)	Tiempo V2 (seg.)
5	3	364	1	0,2
5	5	3.906	9	0,2
5	6	9.331	23	0,2
6	5	19.531	118	0,2

Cuadro 2: Resultados de las pruebas de rendimiento realizadas a las dos funciones generadoras del árbol de tipos.

Destacar que aunque el tiempo de visualización del árbol completo usando la versión 2 es siempre mínimo y constante, el tiempo real de carga de la página es algo más elevado debido a algunos procesos que ocurren internamente (estamos hablando de unos 10 segundos adicionales para la última de las pruebas). No obstante, esta pequeña demora tampoco es apreciable por el usuario ya que mientras termina la carga, el árbol ya está visible.

Otra de las partes críticas en cuanto a sobrecarga es mostrar la tabla completa de ítems. Todas las tablas de la aplicación se muestran a través del plug-in de JQuery Datatables, que permite formatear las tablas HTML y añadirles funcionalidades extra (filtrado, buscado, paginación, ordenación, etc). En las primeras versiones de los JSPs se generaba primero la tabla en HTML y después se formateaba con Datatables, esto hacía que el tiempo de carga de la vista fuese muy elevado ya que tenía generar todo el HTML y después formatearlo todo a la vez. Por este motivo se optó por generar el contenido HTML de las tablas de forma dinámica mediante una opción el propio plug-in Datatables. De esta forma el contenido de la tabla se obtenía en formato JSON dinámicamente, lo que permitía generar la tabla y formatearla de forma progresiva.

Volviendo con la tabla de ítems, se probó a mostrar el contenido de las dos formas anteriores, aplicando Datatables directamente sobre la tabla primero y usando Datatables para obtener el contenido en formato JSON de forma dinámica después. El número de ítems almacenados en la base de datos que utilizó para la prueba fue de 50.000, se estimó que este sería el número aproximado que la aplicación tendría pasados 50 años de uso con una media de 1.000 ítems donados al año (desde el 2007 en tan solo un año se ha registrado una cifra de ítems donados superior a 1.000). Los resultados obtenidos fueron:

- Con la primera de las configuraciones de Datatables, el tiempo de carga de la

tabla rozaba el minuto desde que se accedía a la vista hasta que se formateaba completamente.

- Con la segunda configuración de Datatables, la tabla se mostraba de forma instantánea al acceder a la vista y tardaba aproximadamente 2-3 segundos en cargar el contenido.

5. Conclusiones

Como resultado de este trabajo, se ha desarrollado un sistema completamente funcional que se utilizará desde este momento para catalogar todos aquellos objetos que sean donados al Museo de Informática Histórica (MIH) de la Universidad de Zaragoza, así como para catalogar progresivamente todos aquellos objetos que ya pertenecen al museo. Este sistema consta de: una base de datos MySQL en la que almacenar toda la información y una aplicación web que permite consultar el catálogo del museo de forma abierta para cualquier usuario o realizar acciones de administración en ella solo para el usuario administrador (con identificación previa).

El desarrollo de este trabajo puede dividirse en cuatro etapas:

- 1) Análisis de los gestores del mercado y decisión de cuál es mejor opción para resolver este problema.
- 2) Diseño del modelo de datos.
- 3) Desarrollo de la aplicación web.
- 4) Incorporación de pequeñas mejoras a la aplicación, solución de errores y catalogación de varias donaciones reales del museo.

El proceso de desarrollo no ha estado exento de dificultades como era de esperar. Se podría afirmar que la mayor dificultad, sin duda alguna, ha sido el modelado a nivel de base de datos de la jerarquía de tipos de ítems y el permitir gestionarla después a través de una interfaz web. Y más concretamente, ha supuesto muchos problemas la gestión de los atributos de estos tipos, ya que la cantidad de tipos de atributos que pueden coexistir en el sistema según su naturaleza es mucho mayor de lo que se planteo al inicio del trabajo.

Destacar también que resulta de vital importancia y es un aspecto clave tener una jerarquía dinámica de tipos en la que poder añadir nuevos tipos, eliminar tipos no deseados o reubicar otros en diferentes ramas, ya que será común que la clasificación inicial hecha de los tipos no sea la más adecuada pasado un tiempo y que haya que reorganizarla.

5.1. Cronograma

Se muestra una evolución temporal de las tareas que se han llevado a cabo en este trabajo. Para ello se ha elaborado un diagrama de Gantt, en esta sección se muestra una versión reducida, la versión completa está en el Anexo F.

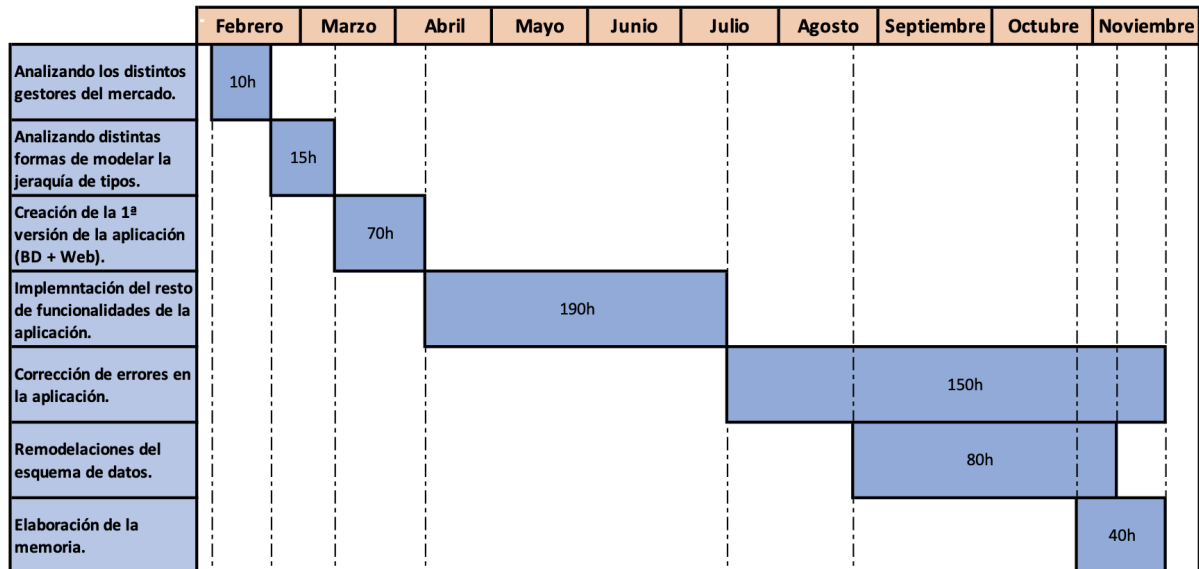


Figura 11: Versión reducida del diagrama de Gantt con la evolución temporal del desarrollo de este trabajo.

Destacar que en este trabajo el director ha jugado un doble papel, puesto que ha hecho tanto de director aconsejando sobre como afrontar el trabajo, como de cliente que será quien utilice la aplicación y quien tiene las necesidades reales de una buena gestión y acceso a esa información. Por tanto la mayoría de las reuniones mantenidas han sido con roles de cliente y empleado, en lugar de alumno y director. Este es el motivo principal de que se supere la veintena de reuniones y de que, desde el mes de julio, la media de horas dedicadas a cada reunión haya sido de unas 4 horas. En la tabla 5.1 se muestran la cronología de las diferentes reuniones mantenidas durante toda la etapa de desarrollo.

Fecha	Lugar	Motivación
09/02/2021	Despacho	Primera toma de contacto con el tema del TFG.
22/02/2021	Online	Repasar análisis de las características de los gestores del mercado.
26/02/2021	Online	Repasar el análisis de la herencia que ofrecen los gestores.
08/03/2021	Online	Discusión sobre el modelado de la jerarquía de tipos y discusión sobre la primera versión del esquema de la Base de datos.
06/04/2021	Online	Probar el primer prototipo de la aplicación.
28/04/2021	Online	Mirar las nuevas funcionalidades de la aplicación.
21/05/2021	Online	Mirar nuevas funcionalidades de la App y revisar propuesta del TFG.
17/06/2021	Online	Mirar las nuevas funcionalidades de la aplicación.
29/06/2021	Online	Mirar las nuevas funcionalidades de la aplicación.
19/07/2021	Online	Mirar las nuevas funcionalidades de la aplicación.
29/07/2021	Online	Mirar correcciones hechas en la aplicación.
23/08/2021	Online	Mirar correcciones hechas en la App y revisar el índice de la memoria.
06/09/2021	Teléfono	Debate sobre uso de tablas MyIsam o InnoDB en la base de datos.
07/09/2021	Online	Mirar correcciones hechas en la aplicación.
09/09/2021	Online	Mirar correcciones hechas en la aplicación.
14/09/2021	Online	Analizar el estado del TFG y debatir aplazamiento a Noviembre.
27/09/2021	Online	Mirar correcciones hechas en la aplicación.
08/10/2021	Online	Mirar correcciones hechas en la aplicación.
15/10/2021	Online	Mirar correcciones hechas en la aplicación.
21/10/2021	Online	Mirar correcciones hechas en la aplicación.
26/10/2021	Online	Mirar correcciones hechas en la aplicación.
09/11/2021	Online	Mirar correcciones hechas en la aplicación.
15/11/2021	Edif. Ada Byron	Dos horas y media cargando datos reales de ítems en la aplicación para observar como se comporta (registrar los 21 ítems de la vitrina central 0.2 del museo).
17/11/2021	Edif. Ada Byron	Tres horas cargando datos reales de ítems en la aplicación para observar como se comporta (registrar los 21 ítems de la vitrina central 0.2 del museo).

Cuadro 3: Cronología de las reuniones mantenidas entre el director y el alumno durante este trabajo.

5.2. Posibles ampliaciones

A continuación se listan las posibles ampliaciones que podrían llevarse a cabo en la aplicación en un futuro próximo:

- Además de tener imágenes asociadas a tipos y a ítems, poder tener también vídeos asociados. Se tratarían de forma idéntica a las imágenes a nivel de base de

datos pero que habría que adaptar las interfaces para mostrarlos correctamente.

- Ampliar el rango de búsqueda del buscador de la aplicación. Permitiendo buscar cualquier cadena en cualquier campo de la base de datos.
- Tratar el atributo 'Descripción' declarado en el tipo 'Ítem lógico' de forma especial en las vistas de tipos e ítems. Mostrando su valor asociado debajo de la tabla ocupando todo el ancho en vez de contenido en su celda. Esto permitiría que al incorporar código HTML como valor de este atributo se tuviese más espacio para mostrarlo, como por ejemplo, al insertar el plug-in HTML de un vídeo.
- Actualmente, a los atributos que presentan un rango tipo solo les pueden asignar valor (una referencia a ítem) los ítems. Se podría ampliar para que se permitiese también asignar una referencia a ítem desde un tipo.
- Dado que la jerarquía de tipos crecerá bastante con el paso del tiempo, sería interesante incorporar algún mecanismo de zoom que permita ampliar/reducir el tamaño en el que se muestran los nodos del árbol para hacerlo más manejable.
- Añadir más tablas y gráficas con información estadística. Hay mucha información en el sistema y podrían mostrarse una gran cantidad de datos estadísticos interesantes.
- Fusionar la web de este sistema de catalogación con actual web del museo MIH (<http://mih.unizar.es/>), con el fin de tener todo accesible desde un mismo punto.

5.3. Opinión personal

Personalmente considero que el resultado obtenido tras la realización de este trabajo es bastante satisfactorio. Se ha logrado implementar todas aquellas funcionalidades que se consideraban imprescindibles y otras más que no lo eran tanto pero que mejoran la aplicación. También se ha conseguido que la aplicación sea bastante eficiente en cuanto a recursos utilizados y tiempos de carga de los datos en la web. Además, se trata de un sistema que va a ser utilizado en la vida real de forma instantánea tras su finalización, por lo que resulta todavía más gratificante haberlo desarrollado.

Con este trabajo he aprendido que los gestores de bases de datos del mercado dejan mucho que desear en algunos aspectos (en referencia a la gestión de la herencia

de estos). También he aprendido a analizar alternativas y ofrecer la mejor solución posible cuando ningún gestor es el adecuado para tu problema de modelado de datos (en referencia al modelado de la jerarquía de tipos y sus atributos). Por último, he aprendido también a trabajar en un entorno donde el cliente de la aplicación no te fija los requisitos funcionales desde el inicio y en cambio te los va proponiendo conforme avanza el desarrollo de la misma, o donde algún requisito que resultaba imprescindible al inicio deja de serlo y viceversa.

Respecto a la metodología de trabajo y relacionado con el aspecto de trabajar con unos requisitos cambiantes constantemente, ha resultado complicado sobrellevarlo en muchas ocasiones, pero también considero que es uno de los motivos por los que el resultado final de la aplicación ha sido finalmente satisfactorio para el cliente y que hacen que este trabajo fin de grado sea más interesante. Destacar también pese a todo la buena relación mantenida con Eduardo (director de este trabajo fin de grado y futuro usuario administrador de la aplicación) durante los 10 meses en los que he estado desarrollando este sistema.

Bibliografía

- [1] Bootstrap. *Getting started to Bootstrap V5.1*. URL: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>. (accedido: 26.11.2021).
- [2] Datatables. *Datatables Manual*. URL: <https://datatables.net/manual/>. (accedido: 26.11.2021).
- [3] Merche Marqués (Universitat Jaume I). *Tema 2. Bases de datos orientadas a objetos*. URL: <https://www3.uji.es/~mmarques/e16/teoria/cap2.pdf>. (accedido: 26.11.2021).
- [4] IBM. *DB2: Creating typed tables*. URL: https://www.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.structypes.doc/doc/t0006634.html. (accedido: 26.11.2021).
- [5] IBM. *Informix: Table inheritance*. URL: https://www.ibm.com/docs/en/informix-servers/12.10?topic=SSGU8G_12.1.0/com.ibm.ddi.doc/ids_ddi_124.htm. (accedido: 26.11.2021).
- [6] Intersystems. *Cache Documentation: Relationships Between Tables*. URL: https://docs.intersystems.com/latest/csp/docbook/Doc.View.cls?KEY=GSQF_foreignkeys#GSQF_foreignkeys_child. (accedido: 26.11.2021).
- [7] Java. *Java API Specifications*. URL: <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>. (accedido: 26.11.2021).
- [8] JavaScript. *JavaScript Documentation*. URL: <https://developer.mozilla.org/es/docs/Web/JavaScript>. (accedido: 26.11.2021).
- [9] PostgreSQL. *PostgreSQL: Inheritance*. URL: <https://www.postgresql.org/docs/9.1/ddl-inherit.html>. (accedido: 26.11.2021).
- [10] Asociación RetroAcción. *Dossier del Museo de Informática Histórica (MIH) de la Universidad de Zaragoza*. URL: <http://mih.unizar.es/DossierMIH.pdf>. (accedido: 26.11.2021).

A. Anexo: Análisis individual de los SGBD del mercado

A continuación, se detallan para cada gestor analizado, los aspectos más importantes respecto a la gestión de la herencia. Por último se muestra una tabla comparativa con las características que ofrece cada gestor a modo de resumen.

A.1. MySQL

Uno de los gestores más populares del mercado que sorprende por no soportar ningún tipo de herencia. Por esto se descartó en primera instancia, aunque finalmente se optó por el a consecuencia de la solución final adoptada en el modelado de los datos que se comentará más adelante.

A.2. MariaDB

Es un gestor derivado de MySQL, por lo que presenta las mismas características citadas anteriormente.

A.3. Oracle

No soporta herencia de tablas, ya que las trata como tablas individuales sin ningún tipo de relación entre ellas. En cambio presenta una buena herencia de objetos, aunque el tratamiento de los mismos es complicado [3].

A.4. PostgreSQL

Probablemente el gestor con mejor gestión de la herencia del mercado. Dispone de una herencia de tablas muy completa. Su único fallo es que las restricciones de tipo clave primaria y clave extranjera no se heredan para las tablas hijas [9].

A.5. DB2

Se parece bastante a Oracle, aunque este permite tener al mismo tiempo herencia en los tipos y herencia en tablas tipadas, que son tablas que se han generado a partir

de objetos. Las restricciones de clave primaria y clave extranjera si que se propagan entre los hijos de la jerarquía, lo que lo hacía un buen gestor candidato. También permite consultar sobre una tabla padre y obtener las tuplas también de sus hijas. El problema es que para las referencias a objetos no existe la integridad referencial, por tanto hay que solucionarlo a nivel de tabla tipada añadiendo la restricción de clave extranjera. Además los tipos presentan varias restricciones importantes [4]:

- No se pueden modificar ni eliminar atributos heredados desde un tipo/tabla hija.
- No se pueden crear tipos sin atributos, aunque este heredando de otro tipo.
- No se puede ni añadir ni eliminar atributos a un tipo si se está utilizando en una tabla tipada.
- No se ha encontrado nada sobre modificar la herencia tipos ya creados.

A.6. Informix

Es muy parecida a DB2, ya que ambas pertenecen a IBM. Ofrece herencia pero solo para tablas tipadas. Al igual que ocurre con DB2, está la importante limitación de que no permite ni añadir ni eliminar atributos de un tipo que se esté usando en una tabla [5].

A.7. CACHE

Es un gestor de los denominados Orientado a Objetos. En el se declaran clases que luego se transforman en tablas. Pese a ser Orientado a Objetos soporta el lenguaje SQL. Al ser un gestor de pago se descartó y no se pudieron hacer pruebas con él [6].

A.8. DB4O

Es un gestor Orientado a Objetos puro. Por tanto, se trabaja con él a través de clases Java y la herencia es bastante buena. Pero presenta dos problemas importantes, uno que el lenguaje de consultas no es SQL (lo que hace complicado el tratamiento de referencias) y otro que al trabajar con clases Java el modelo de datos debe ser fijo (generar código Java para crear una nueva clase de forma dinámica sería complicado).

A.9. MongoDB

Es un gestor de los denominados NoSQL. Los datos no siguen un esquema fijo y se almacenan en documentos independientes, por tanto, es un gestor que no ofrece integridad referencial. Para este sistema, la información estaría repartida en muchos documentos y hacer consultas que impliquen varios documentos resulta muy tedioso en este tipo de gestor.

A.10. Cuadro comparativo de los gestores anteriores

La tabla siguiente compara las características generales de cada gestor y las soluciones que cada gestor ofrece respecto a la herencia.

Gestor	Tipo	Software libre	Herencia	Propagación restricciones	Integridad referencial
MySQL	Relacional	Sí	No tiene	—	—
MariaDB	Relacional	Sí	No tiene	—	—
Oracle	Relacional	No	Objetos	Sí	Sí
PostgreSQL	Relacional	Sí	Tablas	Parcial	No
DB2	Relacional	No	Objetos	Sí	Parcial
Informix	Relacional	No	Objetos	Sí	Parcial
CACHE	Orientado a Objetos	No	Clases	—	—
DB4O	Orientado a Objetos	Sí	Clases	—	—
MongoDB	Orientado a Documentos	Sí	No tiene	—	—

Cuadro 4: Comparativa de características de los principales gestores del mercado

B. Anexo: Tablas de la Base de Datos

En las siguientes secciones se comenta para cada tabla cual es su funcionalidad principal, se detallan los campos que cada una tiene y finalmente, se muestra el código SQL.

B.1. Tabla TIPOS

Almacena los diferentes tipos de ítems que se donan al museo. Esta misma tabla se utiliza también para establecer la relación jerárquica existente entre todos ellos. Dispone de dos campos:

- **nombre:** Utilizado para asignarle un nombre al tipo de ítem y que será único en toda la jerarquía.
- **padre:** Referencia al tipo concreto que hace de padre a nivel jerárquico.

```
CREATE TABLE TIPOS (  
  padre VARCHAR(100),  
  nombre VARCHAR(100) PRIMARY KEY,  
  CONSTRAINT fk_padre_tipos FOREIGN KEY (padre)  
    REFERENCES TIPOS(nombre) ON DELETE CASCADE ON UPDATE CASCADE  
)ENGINE=InnoDB;
```

B.2. Tabla DONANTES

Almacena información de todas aquellas personas que han realizado alguna vez una donación al museo. Dispone de 5 campos:

- **id:** Es el identificador único de cada donante.
- **email:** Dirección de correo electrónico del donante (es opcional).
- **nombre:** Nombre completo del donante (es obligatorio indicarlo).
- **telefono:** Teléfono de contacto del donante (es opcional).

- **descripcion:** Campo en el que poder indicar otros aspectos relevantes acerca de la persona.

```
CREATE TABLE DONANTES (  
  id SERIAL PRIMARY KEY,  
  email VARCHAR(100),  
  nombre VARCHAR(100) NOT NULL,  
  telefono VARCHAR(100),  
  descripcion TEXT  
)ENGINE=InnoDB;
```

B.3. Tabla DONACIONES

Guarda información específica sobre una donación de uno o varios ítems realizada por una persona. Permite relacionar los ítems donados con su donante concreto. Dispone de 8 campos:

- **id:** Es el identificador único de cada donación.
- **persona:** Referencia al donante concreto que efectuó la donación.
- **tipo:** Es obligatorio elegir entre 'Donación a fondo perdido', 'Cesión indefinida' o 'Cesión temporal'.
- **estado:** Es obligatorio elegir entre 'Pendiente' o 'Finalizada'. Este indicador facilita al administrador del museo la identificación de donaciones incompletas.
- **anonima:** Es obligatorio elegir entre 'Sí' o 'No'. Indica si el donante a dado su consentimiento para aparecer en la web de forma pública. Si se marca la donación como anónima solo el administrador podrá conocer quién es el donante, para el resto aparecerá como 'Donante anónimo'.
- **fecha_donacion:** Indica la fecha en la que se ha efectuado la donación. Es obligatorio almacenar dicha fecha en el sistema.
- **contrato:** Este campo se utiliza para almacenar la ruta a un fichero PDF que representa un contrato con el donante sobre la donación que haya realizado. No es obligatorio hacerlo.

- **comentarios:** Permite indicar otros aspectos adiciones relevantes acerca de la donación.

```
CREATE TABLE DONACIONES (  
  id SERIAL PRIMARY KEY,  
  persona BIGINT UNSIGNED NOT NULL,  
  CONSTRAINT fk_persona_donacion FOREIGN KEY (persona)  
    REFERENCES DONANTES(id) ON DELETE CASCADE ON UPDATE CASCADE,  
  tipo VARCHAR(100),  
  estado VARCHAR(100),  
  anonima VARCHAR(2) NOT NULL,  
  fecha_donacion VARCHAR(100) NOT NULL,  
  contrato VARCHAR(200),  
  comentarios TEXT  
)ENGINE=InnoDB;
```

B.4. Tabla ITEMS

Registra cada uno de los diferentes ítems físicos únicos que recibe el museo como parte de una donación y los relaciona con un tipo concreto. Dispone de 4 campos:

- **id:** Es el identificador único de cada ítem.
- **nombre:** Nombre asociado a cada ítem. Por defecto el nombre lo conformarán el nombre del tipo asociado y el identificador numérico.
- **tipo:** Referencia al tipo de ítem concreto de la jerarquía con el que se le asocia.
- **donacion:** Referencia a la donación concreta a través de la cuál este ítem llego al museo por primera vez.


```

CREATE TABLE ITEMS (
  id SERIAL PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  tipo VARCHAR(100) NOT NULL,
  CONSTRAINT fk_tipo_items FOREIGN KEY (tipo)
    REFERENCES TIPOS(nombre) ON DELETE CASCADE ON UPDATE CASCADE,
  donacion BIGINT UNSIGNED,
  CONSTRAINT fk_donacion_items FOREIGN KEY (donacion)
    REFERENCES DONACIONES(id) ON DELETE SET NULL ON UPDATE CASCADE
)ENGINE=InnoDB;

```

B.5. Tabla IMAGENES_ITEMS

Almacena las rutas de todas las imágenes asociadas a ítems. Dispone de 5 campos:

- **id:** Es el identificador único de cada imagen.
- **ruta:** Es la URL del servidor donde se almacena la imagen. El nombre de dicha imagen siempre está compuesto por el identificador del ítem y por su propio identificador como imagen. Por ejemplo: idItemY_imgIdX.png.
- **item:** Referencia al ítem físico concreto con el que se asocia la imagen.
- **descripcion:** Campo que permite asociar una descripción semántica para la imagen.
- **orden:** Número que indica el orden de visualización de las imágenes de un mismo ítem.

```

CREATE TABLE IMAGENES_ITEMS (
  id SERIAL PRIMARY KEY,
  ruta VARCHAR(200) NOT NULL,
  item BIGINT UNSIGNED NOT NULL,
  CONSTRAINT fk_item_imagenes_items FOREIGN KEY (item)
    REFERENCES ITEMS(id) ON DELETE CASCADE ON UPDATE CASCADE,
  descripcion TEXT,
  orden INT
)ENGINE=InnoDB;

```

B.6. Tabla IMAGENES_TIPOS

Almacena las rutas de todas las imágenes asociadas a tipos de ítems. Dispone de 5 campos:

- **id:** Es el identificador único de cada imagen.
- **ruta:** Es la URL del servidor donde se almacena la imagen. El nombre de dicha imagen siempre está compuesto por el nombre único del tipo y por su propio identificador como imagen. Por ejemplo: nombre_del_tipo_imgIdX.png.
- **tipo:** Referencia al tipo de ítem concreto con el que se asocia la imagen.
- **descripcion:** Campo que permite asociar una descripción semántica para la imagen.
- **orden:** Número que indica el orden de visualización de las imágenes de un mismo tipo.

```
CREATE TABLE IMAGENES_TIPOS (  
  id SERIAL PRIMARY KEY,  
  ruta VARCHAR(200) NOT NULL,  
  tipo VARCHAR(100) NOT NULL,  
  CONSTRAINT fk_tipo_imagenes_tipos FOREIGN KEY (tipo)  
    REFERENCES TIPOS(nombre) ON DELETE CASCADE ON UPDATE CASCADE,  
  descripcion TEXT,  
  orden INT  
)ENGINE=InnoDB;
```

B.7. Tabla ATTR_TIPOS

Almacena los atributos propios de cada tipo de ítem, es decir, para cada tipo en la jerarquía, se almacenarán aquí solo aquellos atributos que el propio tipo haya definido. Dispone de 8 campos:

- **tipo:** Referencia al tipo de ítem concreto que ha definido este atributo.

- **atributo:** Es el nombre del atributo. En una misma rama de la jerarquía de tipos nunca se repetirá dos veces el mismo nombre, en ramas diferentes nada lo impide.
- **tipo_rango:** Toma valor NULL cuando estemos ante un atributo de rango STRING, es decir, aquellos atributos en los que el valor asociado es un valor literal. Referenciará a un tipo de ítem concreto cuando estemos ante un atributo en el que su rango lo define otro tipo de la jerarquía.
- **valor_literal:** Cuando el atributo sea de rango STRING tomará valor cuando este sea asignado desde el propio tipo que definió el atributo. Tomará valor NULL cuando el rango lo defina un tipo.
- **publico:** Indica si el atributo se muestra a cualquier usuario (1) o solo es visible para el administrador (0). El valor por defecto es 1.
- **descripcion:** Permite indicar una descripción semántica que explique más en detalle el significado del atributo.
- **rellenable_por:** Indica desde donde se podrán asignar los valores para el atributo, desde un ítem concreto, desde un tipo de ítem o desde ninguno de los dos. Si el rango del atributo es STRING su valor siempre será "Tipo". Si el rango es un tipo, será obligatorio elegir entre "Ítem." "Ninguno". Marcar la opción de "Ninguno" significa que dicho atributo ya aporta suficiente información solo con el tipo indicado en el rango.
- **multievaluado:** Es obligatorio elegir entre "Si." "No". Por defecto todos los atributos serán multivaluados, excepto en los atributos con rango tipo y rellenables por un ítem en los que se indique lo contrario.

```

CREATE TABLE ATTR_TIPOS (
  tipo VARCHAR(100) NOT NULL,
  CONSTRAINT fk_tipo_attr_tipos FOREIGN KEY (tipo)
  REFERENCES TIPOS(nombre) ON DELETE CASCADE ON UPDATE CASCADE,
  atributo VARCHAR(100) NOT NULL,
  tipo_rango VARCHAR(100),
  CONSTRAINT fk_tipo_rango_attr_tipos FOREIGN KEY (tipo_rango)
  REFERENCES TIPOS(nombre) ON DELETE CASCADE ON UPDATE CASCADE,
  valor_literal TEXT,
  publico BOOLEAN NOT NULL DEFAULT 1,
  descripcion TEXT,
  rellenable_por VARCHAR(7) NOT NULL,
  multievaluado VARCHAR(2) NOT NULL,
  PRIMARY KEY (tipo , atributo)
)ENGINE=InnoDB;

```

B.8. Tabla ATTR_VALORES

Almacena para un tipo el valor que le ha asignado a un atributo que este ha heredado jerárquicamente de sus tipos ancestros. Dispone de 4 campos:

- **tipo:** Referencia al tipo de ítem concreto que ha heredado este atributo y que le ha asignado un valor.
- **atributo:** Es el nombre del atributo original. Siempre coincidirá con el nombre que aparezca en ATTR_TIPOS asociado a un ancestro de este tipo.
- **tipo_rango:** Toma valor NULL cuando estemos ante un atributo de rango STRING. Cuando estemos ante un atributo de rango tipo se utilizará este campo para restringir el rango definido por el tipo original, es decir, referenciará a algún tipo heredero del último tipo definido como rango o como restricción del rango.
- **valor_literal:** Cuando el atributo sea de rango STRING tomará el valor que el tipo le asigne a este atributo heredado. Tomará valor NULL cuando el rango lo defina un tipo.

```

CREATE TABLE ATTR_VALORES (
  tipo VARCHAR(100) NOT NULL,
  CONSTRAINT fk_tipo_attr_valores FOREIGN KEY (tipo)
    REFERENCES TIPOS(nombre) ON DELETE CASCADE ON UPDATE CASCADE,
  atributo VARCHAR(100) NOT NULL,
  tipo_rango VARCHAR(100),
  CONSTRAINT fk_tipo_rango_attr_valores FOREIGN KEY (tipo_rango)
    REFERENCES TIPOS(nombre) ON DELETE CASCADE ON UPDATE CASCADE,
  valor_literal TEXT NOT NULL,
  PRIMARY KEY (tipo , atributo)
)ENGINE=InnoDB;

```

B.9. Tabla ATTR_ITEMS_STRING

Almacena para un ítem el valor que este le ha asignado a un atributo de rango STRING que es propio de su tipo asociado o heredado por su tipo asociado. Dispone de 3 campos:

- **item:** Referencia al ítem que asigna el valor.
- **atributo:** Es el nombre del atributo original. Siempre coincidirá con el nombre que aparezca en ATTR_TIPOS asociado al tipo de este ítem o a los ancestros.
- **valor_literal:** Toma el valor literal que el ítem le ha asignado al atributo.

```

CREATE TABLE ATTR_ITEMS_STRING (
  item BIGINT UNSIGNED NOT NULL,
  CONSTRAINT fk_item_attr_items_string FOREIGN KEY (item)
    REFERENCES ITEMS(id) ON DELETE CASCADE ON UPDATE CASCADE,
  atributo VARCHAR(100) NOT NULL,
  valor_literal TEXT,
  PRIMARY KEY (item , atributo)
)ENGINE=InnoDB;

```

B.10. Tabla ATTR_ITEMS_REF

Almacena para un ítem el valor que este le ha asignado a un atributo de rango tipo que es propio de su tipo asociado o heredado por su tipo asociado. Dicho valor no será un valor literal sino que será una referencia a otro ítem concreto cuyo tipo asociado coincida con el tipo indicado en el rango del atributo o con sus herederos. Dispone de 3 campos:

- **item:** Referencia al ítem que asigna el valor.
- **atributo:** Es el nombre del atributo original. Siempre coincidirá con el nombre que aparezca en ATTR_TIPOS asociado al tipo de este ítem o a los ancestros.
- **ref.item:** Referencia al ítem que actúa como valor definido para este atributo por el ítem asignador.

```
CREATE TABLE ATTR_ITEMS_REF (  
  item BIGINT UNSIGNED NOT NULL,  
  CONSTRAINT fk_item_attr_items_ref FOREIGN KEY (item)  
    REFERENCES ITEMS(id) ON DELETE CASCADE ON UPDATE CASCADE,  
  atributo VARCHAR(100) NOT NULL,  
  ref_item BIGINT UNSIGNED,  
  CONSTRAINT fk_ref_item_attr_items_ref FOREIGN KEY (ref_item)  
    REFERENCES ITEMS(id) ON DELETE CASCADE ON UPDATE CASCADE,  
  PRIMARY KEY (item , atributo , ref_item)  
)ENGINE=InnoDB;
```

B.11. Tabla ZONAS

Guarda las distintas zonas en las que se divide el museo. Dispone de 4 campos:

- **nombre:** Define el nombre propio de una zona, cada nombre es único.
- **codigo:** Es un código numérico único que permite identificar a cada zona. El formato es "X.Y", donde la X es el número de planta y la Y el orden de aparición.
- **descripcion:** Guarda cualquier información adicional acerca de la zona.

- **exposicion:** Es obligatorio elegir entre "Sí." "No". Indica si la zona está visible al público, tanto física como virtualmente, o si por el contrario solo tiene acceso el administrador el museo a su contenido.

```
CREATE TABLE ZONAS (
  nombre VARCHAR(100) PRIMARY KEY,
  codigo VARCHAR(4) NOT NULL UNIQUE,
  descripcion TEXT,
  exposicion VARCHAR(2) NOT NULL
)ENGINE=InnoDB;
```

B.12. Tabla LOCALIZACIONES

Distintos puntos en los que se depositan los ítems físicos donados. Cada localización siempre tiene una zona a la que pertenece. Dispone de 4 campos:

- **nombre:** Define el nombre propio de una localización, cada nombre es único.
- **codigo:** Es un código numérico único que permite identificar la zona y también establecer un orden de visita de las mismas.
- **descripcion:** Guarda cualquier información adicional acerca de la localización.
- **zona:** Referencia a la zona concreta a la que pertenece.

```
CREATE TABLE LOCALIZACIONES (
  nombre VARCHAR(100) PRIMARY KEY,
  codigo INTEGER NOT NULL UNIQUE,
  descripcion TEXT,
  zona VARCHAR(100) NOT NULL,
  CONSTRAINT fk_zona_localizacion FOREIGN KEY (zona)
  REFERENCES ZONAS(nombre) ON DELETE CASCADE ON UPDATE CASCADE
)ENGINE=InnoDB;
```

B.13. Tabla UBICACIONES

Guarda para cada ítem físico concreto el historial de localizaciones por las que este ha pasado desde que llegó al museo por primera vez. Pudiéndose repetir una localización en el tiempo para un mismo ítem. Dispone de 6 campos:

- **id:** Es el identificador único de cada ubicación.
- **item:** Referencia al ítem físico concreto asociado a la ubicación.
- **localizacion:** Referencia a la localización concreta en la que se ubica el ítem.
- **fecha_inicio:** Indica la fecha en la que un ítem comenzó a estar ubicado en una localización. Dicha fecha siempre ha de indicarse para cualquier ubicación.
- **fecha_fin:** Indica la fecha en la que un ítem dejó de estar ubicado en una localización. Si toma valor NULL es porque el ítem sigue ubicado allí en la actualidad.
- **comentarios:** Guarda cualquier tipo de información adicional acerca de la ubicación.

```
CREATE TABLE UBICACIONES (  
  id SERIAL PRIMARY KEY,  
  item BIGINT UNSIGNED NOT NULL,  
  CONSTRAINT fk_item_ubicacion FOREIGN KEY (item)  
    REFERENCES ITEMS(id) ON DELETE CASCADE ON UPDATE CASCADE,  
  localizacion VARCHAR(100) NOT NULL,  
  CONSTRAINT fk_localizacion_ubicacion FOREIGN KEY (localizacion)  
    REFERENCES LOCALIZACIONES(nombre) ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  fecha_inicio VARCHAR(100) NOT NULL,  
  fecha_fin VARCHAR(100),  
  comentarios TEXT  
)ENGINE=InnoDB;
```


C. Anexo: Manual de usuario de las funcionalidades de la aplicación

A continuación se muestran las funcionalidades de las que dispone la aplicación, indicando para cada una de ellas las instrucciones para que el usuario (ya sea administrador o invitado) las ejecute.

C.1. Funcionalidades de usuario

Funcionalidades disponibles para cualquier visitante de la web. Se enumeran a continuación todas ellas haciendo una categorización según la vista a la que pertenecen:

C.1.1. Barra de navegación

La aplicación muestra siempre desde cualquier vista una barra de navegación fija con las siguientes opciones, según orden de aparición:

1. **Botón de inicio:** El icono del MIH actúa como un botón del pánico, ya que redirige siempre hacia la página de inicio.
2. **Jerarquía de tipos:** Redirige hacia la vista del árbol jerárquico de tipos. Se hablará más en detalle posteriormente.
3. **Consultar datos:** Menú desplegable que permite visualizar en formato tabla datos acerca de los ítems, zonas, localizaciones e información estadística general. Se hablará más en detalle posteriormente.
4. **Mostrado de datos:** Menú desplegable que permite seleccionar el tipo de información con la que se quiere que trabaje la aplicación de forma global. Actúa como un filtro global y se puede elegir entre ".Exposición", ".Almacén" "Todo", esta última es la opción por defecto.
5. **Tipo de usuario:** Menú desplegable que indica que tipo de usuario está utilizando la aplicación, pudiendo ser usuario invitado (por defecto) o administrador (requiere de identificación con credenciales). Desde este menú se puede acceder a la ventana de inicio de sesión como administrador o se puede cerrar dicha sesión y pasar a usuario invitado.

6. **Sobre el MIH:** Acceso directo a la página original del MIH (<http://mih.unizar.es/>).
7. **Buscador:** Caja de texto donde indicar lo que se desee buscar en el sistema. Se hablará posteriormente más en detalle de su funcionamiento y de sus limitaciones.



Figura 12: Barra de navegación de la aplicación.

C.1.2. Visión global del museo

Desde la página principal de la aplicación (<http://eolo.cps.unizar.es:4040/MIH/>) se podrá observar el esquema de vitrinas de las diferentes plantas del edificio dividido por zonas. Las acciones disponibles son las siguientes:

1. **Ver plano de la planta:** Se puede observar para cada planta un plano del edificio donde aparecen dibujadas las vitrinas en su ubicación.
2. **Ver página detallada de la zona:** Para cada zona del museo, se podrá acceder a una página donde observar información detallada de esa zona así como ver imágenes de las vitrinas que contiene.
3. **Ver listado de ítems de una vitrina:** Cada vitrina que aparece dibujada contiene un enlace en forma de lupa que muestra el listado de ítems contenidos en dicha vitrina.

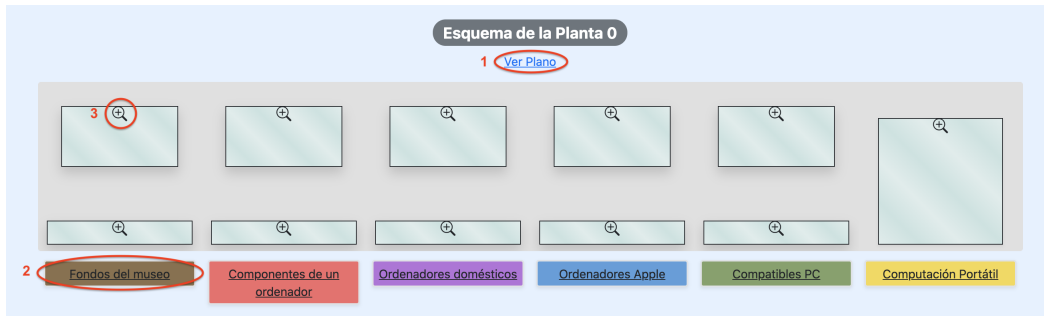


Figura 13: Zonas y localizaciones de una planta del museo.

C.1.3. Visualización de la jerarquía de tipos de ítems

Visualización, en formato árbol, de la jerarquía de tipos de ítems. Las acciones disponibles son las siguientes:

1. **Visualizar solo una rama:** Marcar uno de los tipos en el árbol habilita el botón 'Mostrar el sub-árbol'. Este botón recarga la vista del árbol pero mostrando solo los tipos hijos del seleccionado, es decir, la rama de ese tipo.
2. **Visualizar árbol completo:** El botón 'Árbol completo' recarga la vista del árbol para mostrarlo en su forma completa, con todos sus tipos visibles.
3. **Ocultar/Mostrar subtipos:** En la caja de cada tipo padre existe un botón -/+ que permite de forma dinámica ocultar los subtipos del padre o mostrarlos si estaban ocultos previamente.
4. **Acceder a la vista detallada de un tipo:** En cada caja hay un enlace con el nombre del tipo que redirige a la vista con información detallada. Se hablará mas en detalle en la siguiente sección.
5. **Recuento de ítems:** En cada caja del árbol puede observarse un resumen de los ítems asociados a ese tipo, tanto los propios (indicado como propios entre paréntesis) como el global sumando los de los tipos hijos.



Figura 14: Árbol jerárquico de los tipos (vista del usuario invitado).

C.1.4. Vista detallada de un tipo de ítem

En la vista completa de un tipo podrá observarse la siguiente información:

- El conjunto de imágenes asociadas al tipo. Además, el usuario podrá clicar en cualquiera de ellas para visualizarlas a tamaño real y en pantalla completa. Ver imagen 15.
- Una tabla con los atributos propios y heredados de la rama a la cual el tipo pertenece jerárquicamente, junto con su rango, su valor asociado y quién le ha asignado el valor. Ver imagen 16.
- Un árbol con los tipos hijos. El árbol presenta la misma estética visual que el árbol de la vista de la jerarquía completa. Ver imagen 17.
- Una tabla que lista tanto los ítems propios del tipo como los ítems de los tipos hijos en formato tabla, indicando el nombre del ítem y el tipo en caso de estar asociado a un subtipo del de la vista. Dispone de un filtro seleccionable para mostrar solo los ítems propios (Opción 'Propios'), o los propios y los de los hijos (Opción 'Todos'). Ver imagen 18.



Figura 15: Imágenes asociadas al tipo 'ZX Spectrum +2'.

JERARQUÍA TIPOS	ATRIBUTO*	RANGO	VALOR	ASIGNADOR
Ítem lógico	año de comercialización	STRING	1982	Sinclair ZX Spectrum 16K
	Comentarios ítem	STRING	Valor ya asignado	1 ítems
	defectos	STRING	Sin valor	
	Descripción	STRING	Sin valor	
	fecha de compra	STRING	Sin valor	
	número de ejemplares	STRING	Valor ya asignado	1 ítems
	País	STRING	Reino Unido	Sinclair ZX Spectrum
	Precio de compra	STRING	Sin valor	
	precio original	STRING	Sin valor	
Hardware	fabricante	STRING	Sinclair Research	Sinclair ZX Spectrum
	funciona	STRING	Valor ya asignado	1 ítems
	modelo	STRING	Sin valor	
	número de serie	STRING	Valor ya asignado	1 ítems
	número de serie Unizar	STRING	Sin valor	
	Otras características	STRING	Sin valor	
Ordenador	audio	Zumbador ZX Spectrum (Dispositivo de audio)	No aplica	Sinclair ZX Spectrum 16K
	dispositivo de almacenamiento (*)	Dispositivo de almacenamiento	No aplica	Ordenador
	gráficos	STRING	16 colores, 32x24 (texto), 256x192	Sinclair ZX Spectrum
	memoria central	STRING	16K	Sinclair ZX Spectrum 16K
	Microprocesador	Zilog Z80 (CPU)	No aplica	Sinclair ZX Spectrum
	velocidad	STRING	3,54 MHz	Sinclair ZX Spectrum
Ordenador personal	No existen atributos propios			
Sinclair ZX Spectrum	No existen atributos propios			
Sinclair ZX Spectrum 16K	No existen atributos propios			

Figura 16: Tabla de atributos del tipo 'ZX Spectrum 16K'.

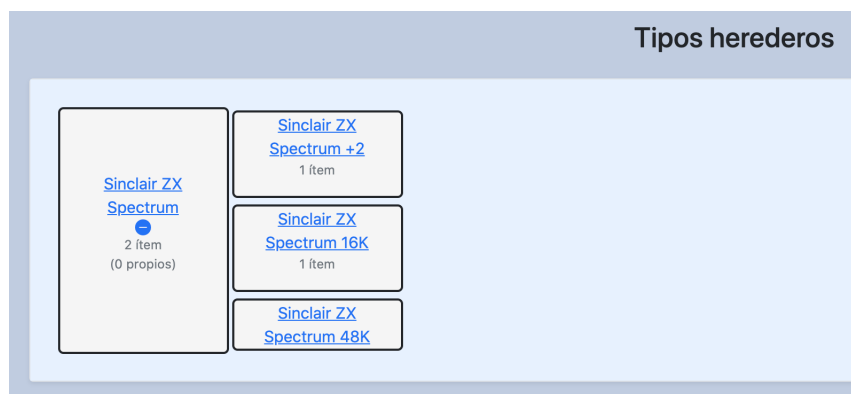


Figura 17: Árbol con los subtipos del tipo 'ZX Spectrum'.

Ítems del tipo

Propios Todos

Mostrar registros Buscar:

Nombre	Tipo
Spectrum+2-12345678 #3	Sinclair ZX Spectrum +2
ZX Spectrum 16K #1	Sinclair ZX Spectrum 16K

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior Siguiente

Figura 18: Tabla con los ítems del tipo 'ZX Spectrum'.

C.1.5. Consultar datos variados con información detallada

Desde la opción 'Consultar Datos' de la barra de navegación se despliega un menú que permite visualizar en formato tabla la siguiente información:

- **Ítems físicos:** La tabla muestra el ID, el nombre, el tipo asociado, información sobre la donación (nombre del donante en caso de ser donación pública y fecha

de donación) e información sobre la ubicación actual (zona y localización). No se permite al usuario invitado acceder a información detallada sobre un donante ni tampoco a información detallada sobre localizaciones que no estén en exposición.

- **Zonas:** La tabla muestra el nombre, un código de ordenación, un indicador de si está en exposición y un enlace a las localizaciones concretas de la zona (si no está expuesta se indica 'La zona no está en exposición' en lugar del enlace).
- **Localizaciones:** La tabla muestra, solo para localizaciones pertenecientes a zonas expuestas, el nombre, el código de ordenación, la zona y un enlace al listado de ítems que se encuentran expuestos en ella (tabla con el mismo formato que la comentada anteriormente para los ítems).

Toda la información mostrada en la tabla estará enlazada una con otra en caso de ser posible. La tabla contendrá filtros de texto a nivel de columna (escribiendo la palabra a buscar en la caja), buscador global en la tabla, capacidad para ordenar columnas (clicando en la cabecera de la columna para cambiar el orden) y paginación (pudiendo indicar el número de entradas a mostrar en la tabla).

Además, todas las tablas (excepto la de ítems) contendrán un formulario inferior con información detallada que se autocompletará al clicar en una fila de la tabla (la fila seleccionada se marcará con fondo azul). En el caso de la tabla de ítems, la información detallada de cada ítem se mostrará a través de una vista única para cada uno de ellos, la cual será accesible a través de un enlace y que se hablará de ella más adelante.

Zonas

Mostrar registros Buscar:

Nombre	Código	En exposición	Localizaciones
Aplicaciones empresariales	1.4	Sí	Ver listado
Estaciones de trabajo	2.1	Sí	Ver listado

Mostrando registros del 11 al 12 de un total de 12 registros

Anterior 1 **2** Siguiente

Información detallada

Nombre	Código	En exposición	Descripción
Aplicaciones empresariales	1.4	Sí	Se presentan las que son probablemente las principales funciones de los ordenadores: la gestión de cantidades enormes de datos y las aplicaciones ofimáticas.

Figura 19: Tabla con información detallada de las zonas del museo.

Ítems físicos

Mostrar registros Buscar:

ID	Nombre	Tipo de ítem	Donación	Localización
28	AMIGA Workbech #1	AMIGA Workbech	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)
24	Amstrad CPC 464 #1	Amstrad CPC 464	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)
40	Amstrad GT 65 #1	Amstrad GT 65	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)
19	BBC Micro #19	BBC Micro	Eduardo Meso (donación: 25/10/2017)	Vitrina central 0.2 (zona: Ordenadores domésticos)
33	Caja del Defender of the Crown (Amiga)	Caja de cartón	José Luis Torres (donación: 13/09/2011)	Vitrina central 0.2 (zona: Ordenadores domésticos)
37	Caja del Lemmings (Amiga)	Caja de cartón	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)
23	Cassette Computone CRC-1001 #1	Cassette Computone CRC-1001	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)
15	CBM Model 3040 Dual Drive Floppy Disk #15	CBM Model 3040 Dual Drive Floppy Disk	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)
27	Commodore 1084 #1	Commodore 1084	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)
25	Commodore Amiga 500 #1	Commodore Amiga 500	No registrada	Vitrina central 0.2 (zona: Ordenadores domésticos)

Mostrando registros del 1 al 10 de un total de 36 registros Anterior Siguiente

Figura 20: Tabla con información detallada de los ítems del museo.

Localizaciones

Mostrar registros Buscar:

Nombre	Código	Zona	Listado de ítems
Vitrina lateral 0.1	1	Componentes de un ordenador	Ver listado
Vitrina central 0.1	2	Componentes de un ordenador	Ver listado
Vitrina lateral 0.2	3	Ordenadores domésticos	Ver listado
Vitrina central 0.2	4	Ordenadores domésticos	Ver listado
Vitrina lateral 0.3	5	Ordenadores Apple	Ver listado
Vitrina central 0.3	6	Ordenadores Apple	Ver listado
Vitrina lateral 0.4	7	Compatibles PC	Ver listado
Vitrina central 0.4	8	Compatibles PC	Ver listado
Vitrina lateral 0.5	9	Computación Portátil	Ver listado
Vitrina central 0.5	10	Computación Portátil	Ver listado

Mostrando registros del 1 al 10 de un total de 21 registros

Anterior 2 3 Siguiente

Información detallada

Nombre	Código	Zona	Descripción
<input type="text" value="Almacen"/>	<input type="text" value="21"/>	Fondos del museo	<input type="text"/>

Figura 21: Tabla con información detallada de las localizaciones del museo.

C.1.6. Consultar datos estadísticos del sistema

Visualización de tablas y gráficas que ofrecen una visión global de la información almacenada por el sistema. Está accesible desde la opción 'Estadísticas' del menú desplegable 'Consultar datos' en la barra de navegación. Se pueden observar los siguientes elementos:

- **Resumen de los donantes:** Tabla que muestra para cada donante registrado en el sistema, el total de donaciones que ha realizado y el total de ejemplares de ítems que ha donado. Ver imagen 22.
- **Resumen anual:** Tabla que muestra el total de donaciones realizadas y el total de ejemplares de ítems donados cada año. Tanto esta tabla como la anterior, utilizan para sacar los datos estadísticos los valores dados a los distintos ítems

en el atributo 'número de ejemplares' del tipo 'Ítem lógico'. Ver imagen 22.

- Evolución de la comercialización:** Gráfica que muestra una evolución anual del total de ejemplares de ítems del sistema que fueron lanzados cada año. Si no se tiene información de ítems en el sistema para un año concreto, se muestra con valor 0. Para sacar esta información, se utilizan los valores dados a los distintos ítems/tipos en los atributos 'número de ejemplares' y 'año de comercialización' del tipo 'Ítem lógico'. Ver imagen 23.

Las tablas contendrán filtros de texto a nivel de columna (escribiendo la palabra a buscar en la caja), buscador global en la tabla, capacidad para ordenar columnas (clicando en la cabecera de la columna para cambiar el orden) y paginación (pudiendo indicar el número de entradas a mostrar en la tabla). En la gráfica, al pasar el cursor sobre cada uno de los puntos aparecerá una pequeña ventana donde se indicará a forma de resumen el año y el número de ejemplares.

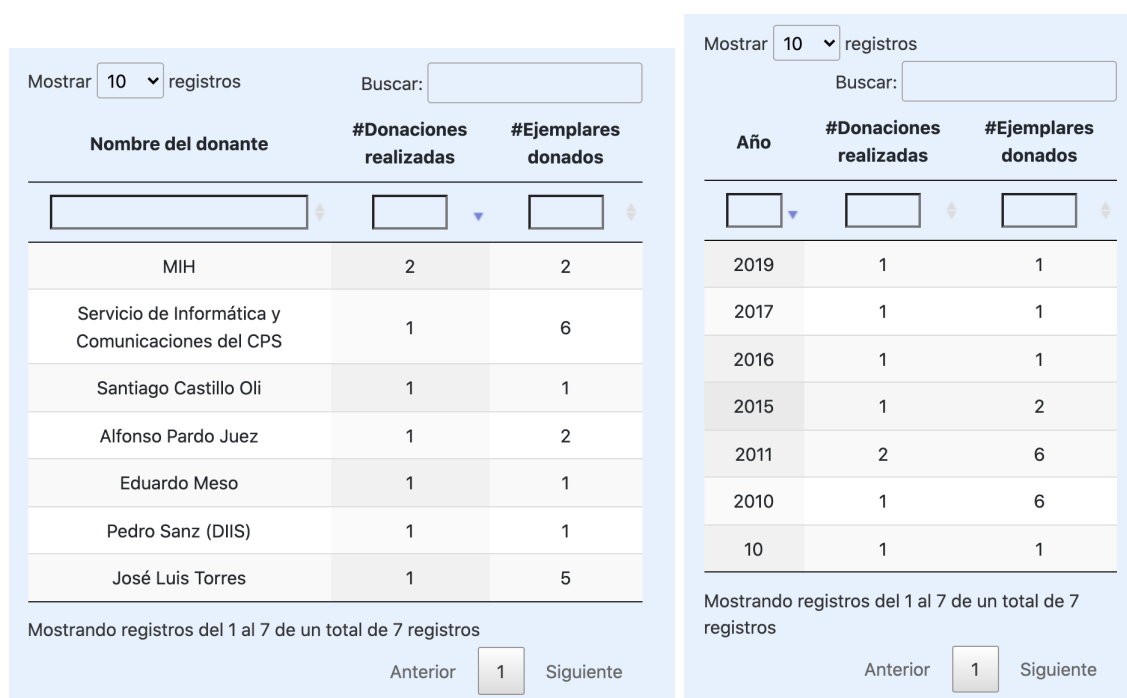


Figura 22: Tablas estadísticas que resumen la información de los donantes y la información anual del sistema.

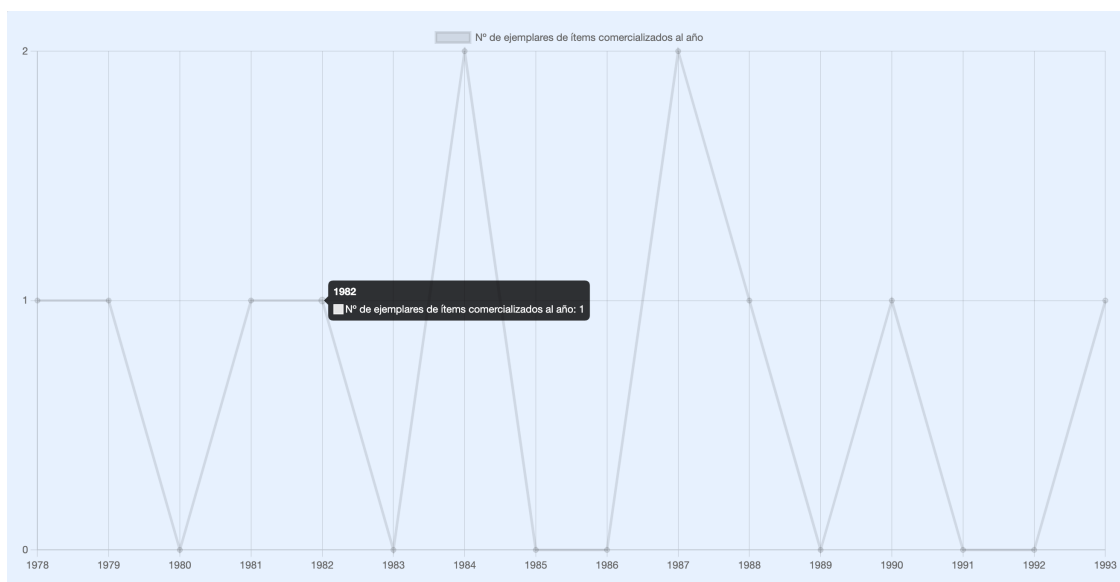


Figura 23: Gráfica que muestra la evolución temporal de la comercialización de ítems del sistema.

C.1.7. Vista detallada de un ítem

La vista con información detallada de un ítem mostrará la siguiente información:

- Enlace a su tipo asociado debajo del nombre del ítem.
- Conjunto de imágenes asociadas al ítem (de forma idéntica a la mostrada anteriormente para los tipos).
- Tabla con los atributos de la rama del tipo al que pertenece el ítem junto con los valores asociados (que podrán ser dados por tipos o por el propio ítem). Dicha tabla es visualmente idéntica a la mostrada en la vista de los tipos.
- Información acerca de la donación a la cual pertenece el ítem. Se muestra el tipo de donación, la fecha y el nombre del donante (este último solo en el caso de que la donación no sea anónima). En caso de no pertenecer a ninguna donación, se indicará con el texto 'Este ítem no pertenece a ninguna donación'. Ver imagen 25.
- Tabla con el histórico de ubicaciones por la que ha pasado el ítem. Se muestra para cada ubicación la zona, la localización, la fecha de inicio y de finalización

y comentarios adicionales. Siempre aparecerá en verde la ubicación actual en la que se encuentre el ítem. Si hay un periodo temporal en el que no existe ubicación, se indicará en la tabla como una nueva entrada con el número de días de ese periodo. Destacar también que en esta tabla solo aparecerán para usuarios invitados aquellas que estén en una zona en exposición, en caso contrario se informará con el mensaje "No expuesto públicamente". Ver imagen 26.

JERARQUÍA TIPOS	ATRIBUTO*	RANGO	VALOR	ASIGNADOR
Ítem lógico	año de comercialización	STRING	Sin valor	
	Comentarios ítem	STRING	Sin valor	
	defectos	STRING	Sin valor	
	Descripción	STRING	Sin valor	
	fecha de compra	STRING	Sin valor	
	número de ejemplares	STRING	1	Defender of the Crown (Amiga) #1
	País	STRING	Sin valor	
	Precio de compra	STRING	Sin valor	
	precio original	STRING	Sin valor	
Software	Contenido (*)	Ítem lógico	Caja del Defender of the Crown (Amiga) Defender of the crown manuals	Defender of the Crown (Amiga) #1 (Software)
	Plataforma Hardware	Commodore Amiga (Ordenador)	Sin referencia	Defender of the Crown (Amiga)
	soporte (*)	Floppy 3.5 (Almacenamiento extraíble)	Floppies Defender of the Crown (Amiga) Rel 1 Floppies Defender of the Crown (Amiga) Rel 2	Defender of the Crown (Amiga) #1 (Defender of the Crown (Amiga))
Videojuego	No existen atributos propios			
Defender of the Crown	No existen atributos propios			
Defender of the Crown (Amiga)	No existen atributos propios			

Figura 24: Tabla con los atributos del ítem 'Defender of the Crown (Amiga) #1'.

Info. de la donación
<ul style="list-style-type: none"> • Tipo: Donación a fondo perdido • Fecha: 2011-09-13 • Donante: José Luis Torres

Figura 25: Información sobre la donación visible desde la vista del ítem para usuarios invitados.

Histórico de ubicaciones en el museo				
Zona	Localización	Fecha de inicio	Fecha de finalización	Comentarios
Computación Portátil	Vitrina central 0.5	2021-03-12	2021-04-12	No hay comentarios
Existen 221 días sin ubicación conocida o sin exponer públicamente				
Ordenadores domésticos	Vitrina central 0.2	2021-11-20	Actualidad	No hay comentarios

Figura 26: Historial de ubicaciones visible desde la vista del ítem para usuarios invitados.

C.1.8. Búsqueda en el sistema

Buscador devuelve aquellas cadenas que coinciden con el substring indicado para nombres de ítems, nombres de un tipos, nombres de localizaciones o nombres de zonas (en estos dos últimos casos, siempre que la zona se encuentre en exposición). También busca coincidencias entre los valores de un atributo perteneciente a cualquier tipo (siempre que dicho atributo sea público cuando se opera como usuario invitado). Cuando se accede como administrador se permite la búsqueda también de nombres de donantes y de valores de atributos privados. Los resultados se muestran en una tabla donde se indica la coincidencia encontrada y su categoría. Las categorías existentes son:

- *Nombre del tipo de ítem.*
- *Nombre del ítem físico.*
- *Nombre de la zona.*
- *Nombre de la localización.*
- *Valor del atributo 'xxxx' (del tipo 'XXXX').*
- *Nombre del donante.* (solo mostrado para el administrador)

La tabla, como las citadas en otras secciones, contendrá filtros de texto a nivel de columna (escribiendo la palabra a buscar en la caja), buscador global en la tabla, capacidad para ordenar columnas (clicando en la cabecera de la columna para cambiar el orden) y paginación (pudiendo indicar el número de entradas a mostrar en la tabla).

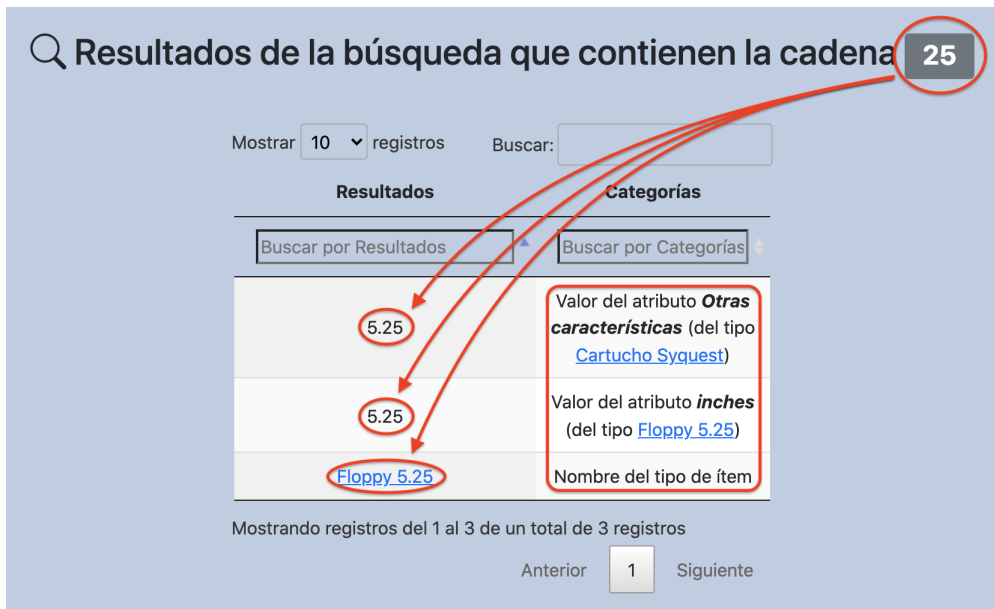


Figura 27: Visualización de los resultados de una búsqueda en el sistema.

C.2. Funcionalidades específicas de administración

Funcionalidades disponibles únicamente para el administrador con inicio de sesión previo. Se enumeran a continuación todas ellas haciendo una categorización según la vista a la que pertenecen:

C.2.1. Visualización de la jerarquía de tipos de ítems

Se permite administrar la jerarquía de tipos mediante tres posibles acciones ubicadas en la parte superior del árbol. Las acciones son las siguientes:

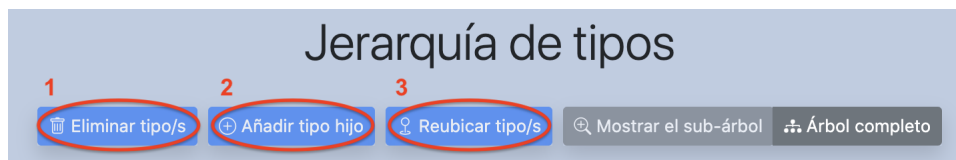


Figura 28: Acciones de administrador disponibles en la jerarquía de tipos.

1. **Eliminar tipo/s:** Elimina uno o varios tipos del sistema, pudiéndose elegir la

opción de borrado en cascada para eliminar también los descendientes. Para ello el administrador seleccionará en el árbol el tipo o los tipos a eliminar, pulsará el botón 'Eliminar tipo/s' y después en la ventana emergente podrá marcar la casilla de borrado en cascada (borrar tipos y subtipos en cascada, o borrar solo el tipo marcado y reubicar los subtipos bajo su abuelo si no es en cascada) antes de confirmar la acción si lo desea. Además, eliminar un tipo conlleva las siguientes consecuencias directas que se efectúan de forma automática: Borrado de los atributos del tipo y valores asociados, borrado de las imágenes asociadas al tipo y el borrado de los ítems propios del tipo (con las consecuencias propias que esto conlleva y que detallan más adelante). Si borrar un tipo conlleva alguna de estas consecuencias se indicará en la ventana emergente con un mensaje de alerta.

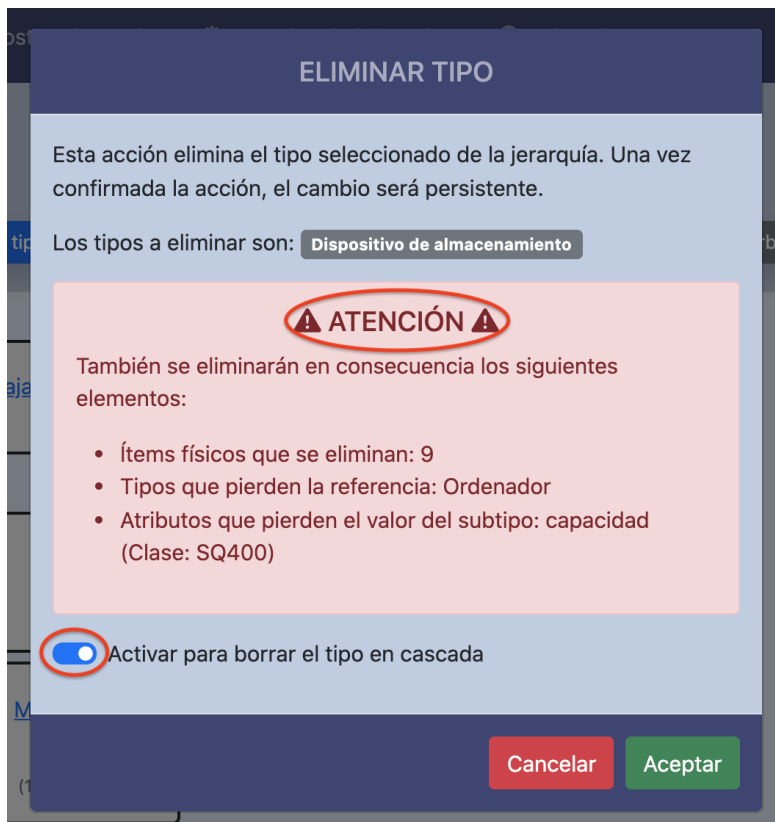


Figura 29: Ventana emergente de la acción de eliminar un tipo.

2. **Añadir tipo hijo:** Añade un nuevo tipo hijo bajo el tipo seleccionado. Para

ello el administrador seleccionará primero el tipo sobre el que añadir el hijo, pulsará el botón 'Añadir tipo hijo' y después indicará en la ventana emergente el nuevo nombre antes de aceptar la acción.

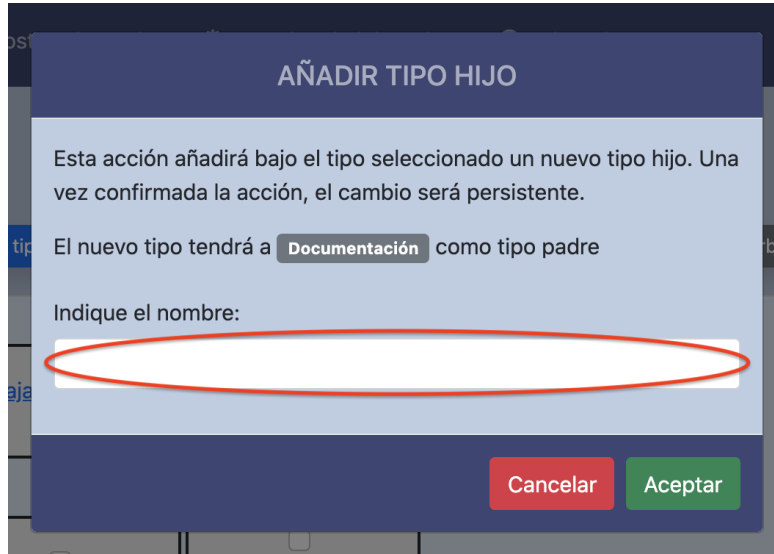


Figura 30: Ventana emergente de la acción de añadir un tipo.

- 3. Reubicar tipo/s:** Reubica uno o varios tipos sobre otro en la jerarquía. Para ello el administrador seleccionará dos o más tipos del árbol, pulsará el botón 'Reubicar tipo/s' y después indicará en la ventana emergente cual de ellos actuará como nuevo padre del resto, también podrá marcar la casilla de reubicar en cascada antes de confirmar la acción si lo desea para reubicar el tipo junto con los subtipos en cascada o solo el tipo y reubicar los subtipos bajo el abuelo. Reubicar un tipo puede suponer la pérdida de los valores asignados por los subtipos a atributos propios del tipo a reubicar. En estos casos se alerta mediante un mensaje en la ventana emergente.

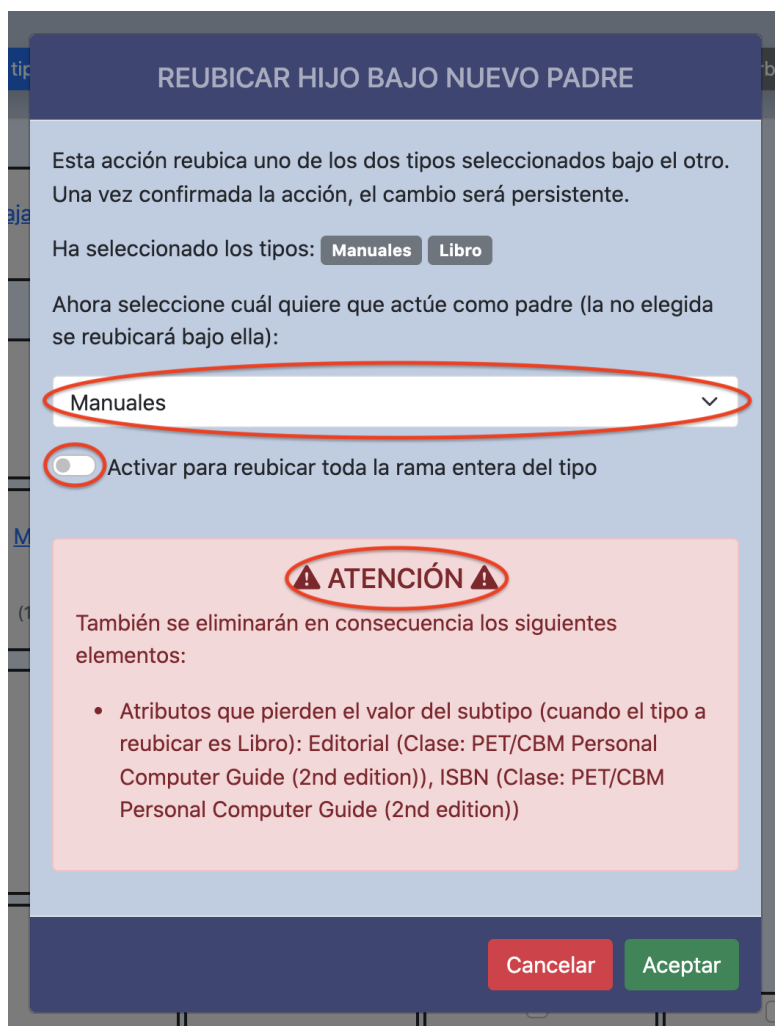


Figura 31: Ventana emergente de la acción de reubicar un tipo.

C.2.2. Vista detallada de un tipo de ítem

Se permiten realizar las siguientes acciones relacionadas con el tipo mostrado:

- **Cambiar el nombre:** El administrador podrá indicar un nuevo nombre para el tipo, siempre que este no haya sido utilizado previamente por otro tipo existente. Para ello pulsará el botón ubicado junto al nombre y en la ventana emergente lo editará antes de confirmar la acción.

Tipo de ítem **Sinclair ZX Spectrum +2** 

Figura 32: Ventana emergente de la acción de editar el nombre de un tipo.

- **Añadir imágenes:** El administrador podrá subir al sistema imágenes ubicadas en su dispositivo y asociarlas al tipo de la vista. Para ello pulsará el botón '+' situado detrás de la última imagen (o debajo del nombre del tipo si no hay imágenes), después en la ventana emergente seleccionará la imagen a subir en el selector de archivos e indicará una descripción textual si lo desea antes de confirmar.

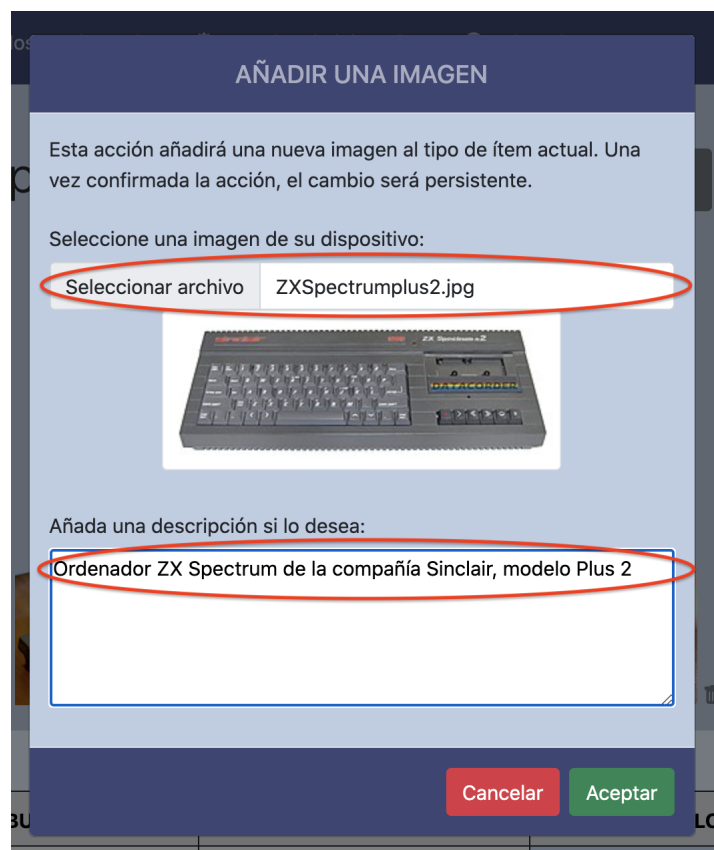


Figura 33: Ventana emergente de la acción de subir una imagen.

- **Ordenar imágenes:** El administrador podrá cambiar el orden de aparición de las imágenes de un tipo editando la información directamente en la base

de datos (no está implementado desde la interfaz de la aplicación). Para ello deberá cambiar el valor del campo 'orden' en la tabla 'IMAGENES_TIPOS'.

- **Eliminar imágenes:** El administrador podrá eliminar una imagen del sistema de forma permanente y quedar disociada del tipo al que pertenecía. Para ello pulsará sobre el icono de la papelera que aparece en la esquina inferior derecha de cada imagen, después deberá confirmar la acción en la ventana emergente.



Figura 34: Ventana emergente de la acción de eliminar una imagen.

- **Añadir un nuevo ítem:** El administrador podrá añadir un nuevo ítem para el tipo de la vista actual. Dicho ítem será genérico, es decir, tendrá como única ubicación por defecto el almacén del museo, no tendrá donación asociada, no dispondrá de imágenes, no tendrá valores propios asociados a atributos y dispondrá de un nombre generado automáticamente que consistirá en el nombre del tipo junto con el identificador de dicho ítem nuevo. Para ello deberá pulsar sobre el botón 'Añadir nuevo ítem' y se redirigirá de forma automática a la vista del ítem creado.

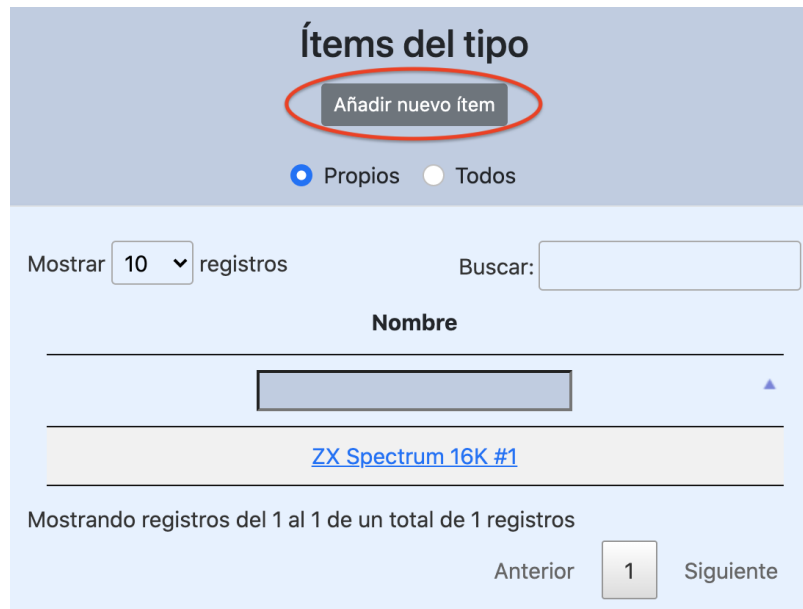


Figura 35: Botón para añadir directamente un ítem genérico asociado a un tipo.

- Añadir atributo:** El administrador podrá definir un nuevo atributo para el tipo de la vista, el cual automáticamente será heredado por los tipos herederos de este. Para ello deberá pulsar el botón '+' que aparece junto a la cabecera 'ATRIBUTO' de la tabla de atributos, después en la ventana emergente deberá indicar el nombre del atributo (no debe coincidir con ningún nombre de atributo de la rama jerárquica a la que pertenece el tipo), podrá indicar una descripción si lo desea y deberá seleccionar el rango del atributo (STRING o Tipo). Según el rango seleccionado se dan los siguientes escenarios:

JERARQUÍA TIPOS	ATRIBUTO*	RANGO
Ítem lógico	año de comercialización	STRING
	Comentarios ítem	STRING

Figura 36: Botón para añadir un nuevo atributo al tipo.

- Si el rango es STRING, deberá indicar quién asignará valores al atributo (Tipo o Ítem), si se elige 'Tipo' entonces podrá asignar un valor en ese

mismo instante si así lo desea o hacerlo más tarde en el tipo o en un subtipo, si se elige "Ítem" solo se le podrá asignar valor desde un ítem del propio tipo o desde un ítem de los tipos herederos.

AÑADIR UN ATRIBUTO

Esta acción añadirá un nuevo atributo al tipo actual. Una vez confirmada la acción, el cambio será persistente.

Indique el nuevo nombre del atributo:

Seleccione el rango del atributo:

String Tipo

Indique quién asignará el valor de este atributo:

Tipo Ítem Ninguno

Indique ahora un valor o hágalo más tarde en este u otros subtipos:

Añada si lo desea una descripción semántica para este atributo:

Cancelar Aceptar

Figura 37: Ventana emergente de la acción de añadir un atributo con rango STRING.

2. Si el rango marcado es Tipo, se deberá seleccionar del árbol de tipos uno de ellos para que actúe como rango raíz. Después deberá indicar quien asignará valores al atributo, hay dos opciones:
 - a) Si se elige "Ninguno" el atributo no tendrá valores asociados ya que se considera que el rango y sus posibles restricciones aportan ya suficiente información.
 - b) Si se elige "Ítem" se podrán asignar valores únicamente en los ítems (dicho valor consistirá en la referencia aun ítem del tipo marcado por el rango o su restricción) y habrá que indicar también será multivaluado o no (en caso de ser multivaluado, se permitirá en un ítem referenciar a uno o más ítems con el mismo atributo).

AÑADIR UN ATRIBUTO

Esta acción añadirá un nuevo atributo al tipo actual. Una vez confirmada la acción, el cambio será persistente.

Indique el nuevo nombre del atributo:

Seleccione el rango del atributo:

String Tipo

Tipo Ítem Ninguno

Indique si el atributo será multivaluado:

Si No

Añada si lo desea una descripción semántica para este atributo:

Cancelar Aceptar

Figura 38: Ventana emergente de la acción de añadir un atributo con rango Tipo.

- **Eliminar atributo:** Si el tipo dispone de atributos propios, es decir, atributos que han sido definidos por él y que no son heredados, el administrador podrá eliminarlo, eliminando en consecuencia todos valores dados por el tipo o por los subtipos herederos. Para ello deberá pulsar sobre el botón 'X' que aparece al lado del nombre del atributo en la tabla y confirmar la acción en la ventana emergente.

- Editar atributo:** El administrador podrá modificar para un atributo propio del tipo de la vista, su nombre, su descripción, el valor asociado si lo tuviera (y solo en caso de ser de rango STRING), el rango, quién puede asignarle valores (varía en función de la selección del rango), el tipo que hace rango (solo cuando sea un tipo de rango Tipo”) y si es multivaluado o no. Para ello deberá pulsar sobre el botón 'X' que aparece al lado del nombre del atributo en la tabla y después en la ventana emergente, la cuál es idéntica a la mostrada para añadir, podrá realizar los cambios oportunos antes de confirmar.

JERARQUÍA TIPOS	ATRIBUTO* <input data-bbox="901 619 925 651" type="button" value="+"/>	RANGO	
ítem lógico	año de comercialización <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> 1	STRING	Valo
	Comentarios ítem <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> 2	STRING	Valo

Figura 39: Botones de para borrar (1) y editar (2) un atributo.

- Asignar valor literal:** El administrador podrá asignar un valor literal a un atributo de rango STRING que sea propio del tipo de la vista o que sea heredado y al que ningún tipo ancestro le haya dado valor todavía. Para ello deberá pulsar el botón '+' que aparece junto al texto 'Sin valor' en la fila del atributo de la tabla, después en la ventana emergente podrá indicar en la caja de texto dicho valor. Destacar que el valor puede ser cualquier tipo de texto, incluso se podrá añadir texto HTML que se interpretará después como corresponda al mostrarse en la tabla.
- Editar valor literal:** El administrador podrá editar el valor asignado con anterioridad a un atributo por el mismo. Para ello deberá pulsar el botón con el icono del lápiz que aparece junto al valor del atributo en la tabla, después podrá modificar el valor en la ventana emergente antes de confirmar la acción.
- Eliminar valor literal:** El administrador podrá eliminar un valor literal desde un tipo a un atributo de rango STRING, ya sea el atributo propio o heredado. Después de esta acción el atributo aparecerá con la cadena 'Sin valor'. Para ello deberá pulsar el botón 'X' y confirmar después la acción en la ventana emergente.





JERARQUÍA TIPOS	ATRIBUTO* 	RANGO	VALOR ¹	ASIGNADOR
Ítem lógico	año de comercialización	STRING	Sin valor 	
	Comentarios ítem	STRING	Sin valor	
	defectos	STRING	Sin valor	
	Descripción	STRING	Hardware generador de audio  	Dispositivo de audio

Figura 40: Botones de para añadir (1), editar (2) y borrar (3) el valor literal de un atributo.

- Añadir restricción de rango:** Cuando un atributo heredado sea de rango "Tipo", el administrador podrá restringir su rango en el tipo de la vista actual, para la restricción el administrador elegirá un tipo que sea heredero del tipo que actuaba hasta el momento como rango de ese atributo (sea una restricción previa o el rango inicial). Para ello deberá pulsar el botón '+' que aparece junto al tipo que marca el rango para el atributo en la tabla, después en la ventana emergente deberá pulsar en 'Seleccionar tipo' para ser redirigido al árbol, allí marcará el tipo y pulsará seleccionar, después en la ventana emergente inicial aparecerá el tipo cargado y podrá confirmar la acción.



Figura 41: Ventana emergente de la acción de añadir una restricción de rango (primera imagen) y vista de la selección de un tipo en el árbol (segunda imagen).

- **Editar restricción de rango:** El administrador podrá modificar el tipo que actúa como restricción de rango en el tipo de la vista actual seleccionando un tipo que sea heredero del tipo marcado como rango inicial (el cual aparece entre paréntesis en la celda de la tabla). Para ello deberá pulsar el botón con el icono del lápiz que aparece junto al tipo que marca el rango para el atributo en la tabla, los siguientes pasos son idénticos al proceso de añadir anterior.
- **Eliminar restricción de rango:** El administrador podrá eliminar una restricción de rango de un atributo dada por el tipo de la vista actual. Como consecuencia el rango efectivo para este tipo en este tipo volverá a ser el here-

dado de la restricción del tipo padre, o si no hubiese restricciones intermedias, heredaría el rango original del atributo (el que aparece entre paréntesis). Para ello deberá pulsar el botón 'X' que aparece junto al tipo que marca el rango para el atributo en la tabla y en la ventana emergente confirmar la acción.




	Otras características	STRING
Ordenador	audio	Dispositivo de audio  1
	dispositivo de almacenamiento (*)	HP 6000 SCSI SE   Dispositivo de almacenamiento 2 3

Figura 42: Botones para añadir (1), editar (2) y eliminar (3) un tipo como restricción de rango a un atributo.

C.2.3. Consultar datos variados con información detallada

Se permite al administrador interactuar con las tablas de zonas, localizaciones, donantes y donaciones. La visualización de estas dos últimas es exclusiva para el administrador (ver imágenes 43 y 44).

Donantes

Mostrar registros Buscar:

ID	eMail	Nombre	Teléfono	Donaciones realizadas
<input type="text" value="4"/>	<input type="text" value="alfpardo@svalero.com"/>	<input type="text" value="Alfonso Pardo Juez"/>	<input type="text" value=""/>	<input type="text" value=""/>
4	alfpardo@svalero.com	Alfonso Pardo Juez		Ver listado
1		Donante desconocido		Ver listado
6		Eduardo Meso		Ver listado
8	multitercia@hotmail.com	José Luis Torres	656746475	Ver listado
5	mih@unizar.es	MIH		Ver listado
7		Pedro Sanz (DIIS)		Ver listado
3	scastillo@aragon.es	Santiago Castillo Oli		Ver listado
2		Servicio de Informática y Comunicaciones del CPS		Ver listado

Mostrando registros del 1 al 8 de un total de 8 registros Anterior Siguiente

Figura 43: Vista de la tabla de donantes.

Donaciones

Mostrar registros Buscar:

ID	Tipo	Estado	Anónima	Fecha de donación	Nombre del donante	Lista de items
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	Donación a fondo perdido	Pendiente	No	2019-05-07	Santiago Castillo Oli	Ver listado
5	Donación a fondo perdido	Pendiente	No	2017-10-25	Eduardo Meso	Ver listado
7	Donación a fondo perdido	Pendiente	No	2016-11-02	Pedro Sanz (DIIS)	Ver listado
3	Cesión indefinida	Pendiente	No	2015-04-22	Alfonso Pardo Juez	Ver listado
8	Donación a fondo perdido	Pendiente	No	2011-09-13	José Luis Torres	Ver listado
6	Donación a fondo perdido	Finalizada	No	2011-08-29	MIH	Ver listado
1	Cesión indefinida	Pendiente	No	2010-11-05	Servicio de Informática y Comunicaciones del CPS	Ver listado
4	Donación a fondo perdido	Finalizada	No	0010-03-01	MIH	Ver listado

Mostrando registros del 1 al 8 de un total de 8 registros Anterior Siguiente

Figura 44: Vista de la tabla de donaciones.

A través de los formularios de información detallada que aparecen en la parte inferior de cada tabla, podrá llevar a cabo las siguientes acciones sobre la tabla:

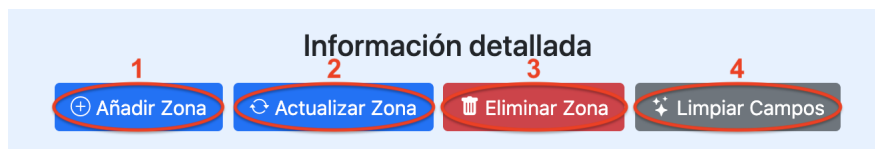


Figura 45: Acciones disponibles para interactuar con las entradas de una tabla.

1. **Añadir nueva entrada:** El administrador deberá cumplimentar aquellos campos del formulario que sean requeridos y opcionalmente cumplimentar el resto, después, tras pulsar el botón 'Añadir X' (donde X será: Zona, Localización,

Donación o Donante) el sistema validará que los datos introducidos sean adecuados para campo de la tabla y creará una nueva entrada con dicha información. Destacar que el campo del formulario que representa el ID de una tabla no permite la asignación de valor ya que este se genera de forma automática.

2. **Actualizar entrada seleccionada:** El administrador seleccionará un entrada de la tabla y automáticamente los datos se cargarán en el formulario para que puedan ser editados. Cuando el administrador pulse el botón 'Actualizar X' el sistema validará que la nueva información de los campos modificados sea adecuada antes de realizar la actualización de dicha entrada de la tabla. Destacar que el campo ID de una tabla no es editable.
3. **Eliminar entrada seleccionada:** El sistema eliminará de forma permanente aquella entrada de la tabla que el administrador haya seleccionado de forma previa a realizar la acción. Para ello deberá pulsar después de seleccionar la entrada el botón 'Eliminar X'. En ocasiones, eliminar una entrada conllevará la modificación o eliminación de otros datos del sistema:
 - Al eliminar un donante se eliminarán sus donaciones asociadas, al eliminar una donación sus ítems no se eliminan pero dejan de pertenecer a ella.
 - Al eliminar una zona se eliminan sus localizaciones asociadas.
 - Al eliminar una localización se eliminarán del historial de ubicaciones de un ítem aquellas que implicasen a dicha localización eliminada.
4. **Limpiar campos:** Limpia el contenido de los campos del formulario. Es una acción útil cuando se ha seleccionado una entrada en la tabla y se quiere dejar de seleccionarla para añadir una nueva con datos nuevos.

Los formularios de información detallada tienen los siguientes tipos de campos:

The figure shows three examples of 'Información detallada' forms. Each form has a header with buttons: 'Añadir', 'Actualizar', 'Eliminar', and 'Limpiar Campos'. The fields are numbered as follows:

- Form 1 (Donación):** ID (1) is 7; Tipo (2) is 'Donación a fondo perdido'; Estado is 'Pendiente'; Anónima is 'No'; Fecha de donación (3) is 02/11/2016; Donante (6) is 'Pedro Sanz (D.I.S.)'; Contrato (10) is 'Ningún archivo seleccionado'; Comentarios (5) is 'procedente de despacho de prof. asociados'.
- Form 2 (Localización):** Nombre is 'Vitrina lateral 0.4'; Código (7) is 7; Zona (6) is 'Compatibles PC'; Descripción is empty.
- Form 3 (Donante):** ID is 3; eMail (8) is 'scastillo@aragon.es'; Nombre (4) is 'Santiago Castillo Oli'; Teléfono (9) is empty; Descripción is 'Biblioteca de Aragón (c/ Doctor Cerrada)'.

Figura 46: Campos existentes en los formularios de visualización de información detallada de una tabla.

1. **ID:** Campo no editable, su valor se genera de forma automática por el sistema.
2. **Selección:** Campo desplegable con opciones seleccionables.
3. **Fecha:** Se podrá indicar escribiendo los números o eligiendo la fecha en un calendario desplegable.
4. **Nombre:** Campo para indicar nombres en una sola línea.

5. **Texto:** Campo para indicar un texto largo, se utiliza para escribir comentarios o descripciones.
6. **Selección:** Botón que redirige a otra tabla para seleccionar allí una entrada y después cargarla en el formulario. Existen dos, uno para seleccionar el donante de una donación y otro para seleccionar la zona de una localización. La vista de una tabla cuando se accede en modo selección es la siguiente:

Seleccionar un donante

Mostrar registros Buscar:

ID	eMail	Nombre	Teléfono	Donaciones realizadas
<input type="text" value="4"/>	<input type="text" value="alfpardo@svalero.com"/>	<input type="text" value="Alfonso Pardo Juez"/>	<input type="text"/>	Ver listado
1		Donante desconocido		Ver listado
6		Eduardo Meso		Ver listado
8	multitercia@hotmail.com	José Luis Torres	656746475	Ver listado
5	mih@unizar.es	MIH		Ver listado
7		Pedro Sanz (DIIS)		Ver listado
3	scastillo@aragon.es	Santiago Castillo Oli		Ver listado
2		Servicio de Informática y Comunicaciones del CPS		Ver listado

Mostrando registros del 1 al 8 de un total de 8 registros Anterior Siguiente

Información detallada

Figura 47: Vista de la tabla de donantes cuando se accede en modo selección desde el botón de selección en el formulario de una donación.

7. **Número:** Se puede indicar el número manualmente o hacerlo incrementado y disminuyendo una unidad con las flechas del campo.
8. **eMail:** Campo para indicar correos electrónicos, valida que sintácticamente este escrito correctamente.
9. **Teléfono:** Campo para indicar números telefónicos, valida que lo introducido tenga exactamente nueve números comprendidos entre el 0 y el 9.

10. **Fichero:** Campo para subir ficheros al sistema. Se utiliza en el formulario de una donación para poder asociarle el PDF de un contrato que acredite la donación. Cuando se selecciona una entrada que contiene un contrato asociado, aparece un nuevo botón para visualizar el contrato y aparece también un checkbox que si se marca se elimina el fichero al actualizar la entrada con el botón 'Actualizar Donación' (Ver imagen 48).



Figura 48: Botón de visualización de un contrato de una donación y checkbox para indicar su eliminación al actualizar la donación.

Destacar también que en la tabla con el listado de ítems de una donación, existen tres acciones más exclusivas del administrador:



Figura 49: Acciones de administrador en el listado de ítems de una donación.

1. **Añadir nuevo ítem:** El administrador pulsará el botón 'Añadir nuevo ítem' y se le redirigirá al árbol de tipos, seleccionará uno y pulsará el botón 'seleccionar', después en la ventana emergente indicará en nombre del ítem (si no

lo hace se asignará uno por defecto) y confirmará la acción. El ítem añadido tendrá la donación desde la que se añadió como donación asociada.

2. **Añadir ítem existente:** El administrador pulsará el botón 'Añadir ítem existente' y se desplegará una ventana emergente con el listado de ítems sin donación asociada, marcará uno de la lista y confirmará la acción. El ítem marcado pasará a estar vinculado con dicha donación.

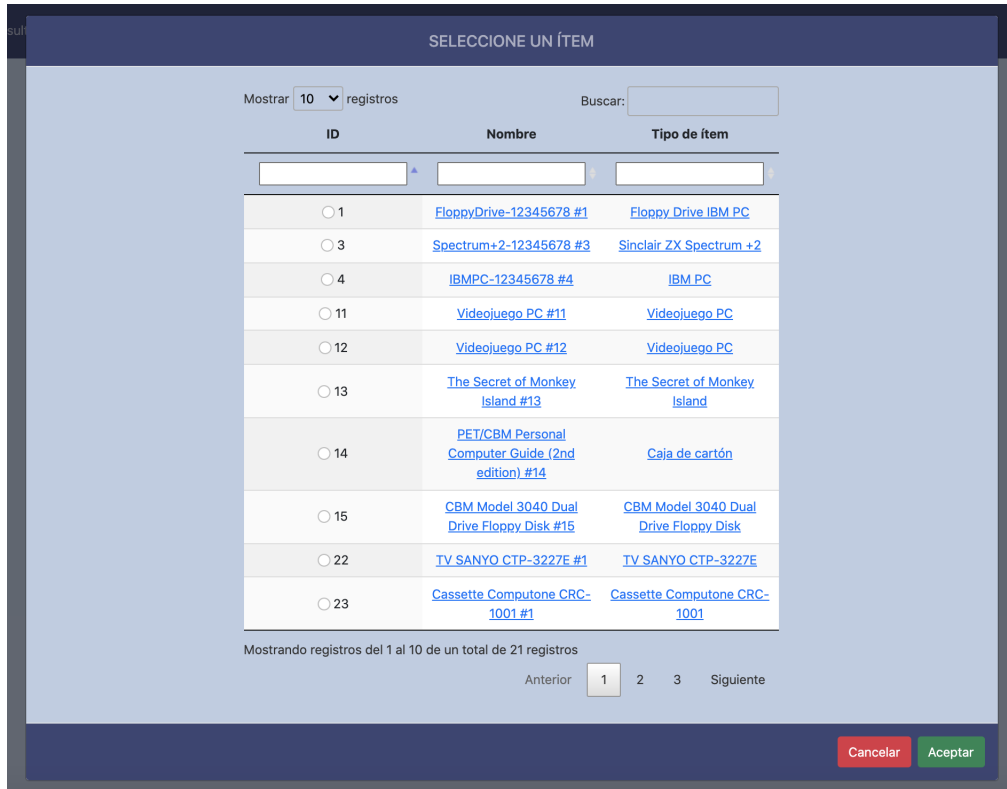


Figura 50: Ventana emergente con los ítems que no están asociados a ninguna donación.

3. **Desvincular ítem de la donación:** El administrador pulsará el botón '-' que aparece junto al ID de cada entrada de la tabla de ítems de la donación, después confirmará la acción en la ventana emergente. El ítem pasará a no tener una donación asociada y desaparecerá de la lista.

C.2.4. Vista detallada de un ítem

Se permiten realizar las siguientes acciones relacionadas con el ítem mostrado:

- **Cambiar el nombre:** El administrador podrá indicar un nuevo nombre para el tipo pulsando el icono del lápiz junto al nombre. Mismo procedimiento que el explicado para los tipos.
- **Cambiar el tipo:** El administrador podrá cambiar el tipo asociado al ítem por otro de los existentes en el árbol jerárquico de tipos. Para ello deberá pulsar sobre el icono del lápiz que aparece junto a la referencia del tipo, después se le redirigirá a la vista del árbol de tipos, marcará uno, pulsará el botón 'seleccionar' y confirmará la acción.
- **Eliminar el ítem:** El administrador podrá eliminar el ítem de forma permanente del sistema. Para ello deberá pulsar el icono de la papelera situado junto al nombre y confirmar la acción en la ventana emergente. Esta acción conlleva el borrado de todas sus imágenes asociadas, el borrado de los valores asignados en atributos que impliquen a dicho ítem y el borrado del historial de ubicaciones de dicho ítem.








Figura 51: Botones para editar el nombre del ítem, eliminar el ítem y cambiar el tipo asociado al ítem.

- **Añadir imágenes:** El administrador podrá subir al sistema imágenes ubicadas en su dispositivo y asociarlas al ítem de la vista. También podrá indicar previamente a subirlas una descripción para cada imagen. Mismo procedimiento que el explicado para los tipos.
- **Eliminar imágenes:** El administrador podrá eliminar una imagen del sistema de forma permanente y quedar disociada del ítem al que pertenecía. Mismo procedimiento que el explicado para los tipos.
- **Asignar valor literal en atributo:** El administrador podrá asignar un valor literal a un atributo perteneciente al tipo asociado o a los ancestros, siempre y

cuando sea de rango STRING y se haya indicado la opción de que los valores de dicho atributo los asignarán los ítems. Mismo procedimiento que el explicado para los tipos.

- **Editar valor literal en atributo:** El administrador podrá editar un valor literal previamente asignado a un atributo. Mismo procedimiento que el explicado para los tipos.
- **Eliminar valor literal en atributo:** El administrador podrá eliminar un valor literal asignado por él mismo a un atributo perteneciente al tipo asociado o a los ancestros, siempre y cuando sea de rango STRING y se haya indicado la opción de que los valores de dicho atributo los asignarán los ítems. Mismo procedimiento que el explicado para los tipos.
- **Asignar referencia a ítem en atributo:** El administrador podrá asignar un ítem como valor a un atributo perteneciente al tipo asociado o a los ancestros, siempre y cuando sea de rango Tipo y se haya indicado la opción de que los valores de dicho atributo los asignarán los ítems. Si además el atributo se marcó como multivaluado, se podrá asignar más de un ítem como valor en dicho atributo. Para ello deberá pulsar el botón '+' que parece en la celda de valores de un atributo en la tabla, después en la ventana emergente deberá marcar el ítem deseado como valor y confirmar la acción.
- **Eliminar referencia a ítem en atributo:** El administrador podrá eliminar un ítem que actúa como valor asignado por él mismo a un atributo perteneciente al tipo asociado o a los ancestros, siempre y cuando sea de rango Tipo y se haya indicado la opción de que los valores de dicho atributo los asignarán los ítems. Para ello debe pulsar el botón 'X' situado junto a la referencia a un ítem y confirmar la acción en la ventana emergente.
- **Editar referencia a ítem en atributo:** Combina la acción de eliminar la referencia a un ítem y añadir una nueva a otro en un solo paso. Para ello debe pulsar el icono del lápiz situado junto a la referencia, y como al añadir, se despliega una ventana emergente donde aparece el listado de ítems que está permitido referenciar.

Contenido (*)	Ítem lógico	Caja del Defender of the Crown (Amiga)   Defender of the crown manuals   
---------------	-----------------------------	---

AÑADIR UNA REFERENCIA CON RANGO *Ítem lógico* AL ATRIBUTO: *Contenido*

Esta acción añadirá una nueva referencia al atributo específica para el ítem físico. Una vez confirmada la acción, el cambio será persistente.

Seleccione el ítem al que desea hacer referencia:

Mostrar registros Buscar:

ID	Nombre	Tipo
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="radio"/> 1	FloppyDrive-12345678 #1	Floppy Drive IBM PC
<input type="radio"/> 3	Spectrum+2-12345678 #3	Sinclair ZX Spectrum +2
<input type="radio"/> 4	IBMPc-12345678 #4	IBM PC
<input type="radio"/> 5	Macintosh SE/30 #5	Macintosh SE/30
<input type="radio"/> 6	Macintosh Classic II #6	Macintosh Classic II
<input type="radio"/> 7	HP 6000 SCSI SE #7	HP 6000 SCSI SE
<input type="radio"/> 8	HP 9000 E35 (ibon) #8	HP 9000 E35
<input type="radio"/> 9	HP 28645 #9	HP 28645
<input type="radio"/> 10	SQ400 #10	SQ400
<input type="radio"/> 11	Videojuego PC #11	Videojuego PC

Mostrando registros del 1 al 10 de un total de 39 registros

Anterior 2 3 4 Siguiente

Figura 52: Botones para añadir, editar y eliminar una referencia a un ítem en un atributo (primera imagen) y ventana emergente con el listado de ítems referenciables (segunda imagen).

- **Añadir nueva ubicación:** El administrador podrá añadir al historial de ubicaciones nuevas localizaciones para el ítem de la vista. Para ello deberá pulsar el botón '+' que aparece junto al título 'Histórico de ubicaciones en el museo', después en la ventana emergente deberá indicar obligatoriamente una fecha de inicio y de finalización (si no se indica fecha de finalización se tratará esta nueva ubicación como la actual) y seleccionar una localización de entre las existentes en el museo, hecho esto podrá confirmar la acción. También podrá indicar de forma optativa un comentario explicativo de la ubicación previo a la confirmación. Antes de hacer efectiva la inclusión de la ubicación en el historial, el sistema comprobará que las fechas introducidas no interfieran en otros periodos de ubicaciones ya existentes. Si se permitirá que haya periodos donde el ítem esté sin localización y se mostrarán visualmente dichos periodos en la tabla de forma especial.

Histórico de ubicaciones en el museo +

Zona	Localización	Fecha de inicio	Fecha de finalización	Comentarios
Fondos del museo	Almacen ✎ ✕	2021-11-17 ✎	Actualidad ✎	Ubicación inicial por defecto ✎

AÑADIR NUEVA UBICACIÓN

Esta acción añadirá una nueva ubicación para el ítem físico actual. Una vez confirmada la acción, el cambio será persistente.

Indique la fecha de inicio:

 📅

Indique la fecha de finalización (o déjelo en blanco si será su ubicación actual):

 📅

Indique algún comentario si así lo desea:

Seleccione una localización:

Mostrar 10 registros

Buscar:

Nombre	Zona
<input type="radio"/> Almacen	Fondos del museo
<input type="radio"/> Sala acristalada	Servidores y Terminales
<input type="radio"/> Sala de estudios	Estaciones de trabajo
<input type="radio"/> Vitrina central 0.1	Componentes de un ordenador
<input type="radio"/> Vitrina central 0.2	Ordenadores domésticos
<input type="radio"/> Vitrina central 0.3	Ordenadores Apple
<input type="radio"/> Vitrina central 0.4	Compatibles PC
<input type="radio"/> Vitrina central 0.5	Computación Portátil
<input type="radio"/> Vitrina central 1.1	Videojuegos I
<input type="radio"/> Vitrina central 1.2	Videojuegos II

Mostrando registros del 1 al 10 de un total de 21 registros

Anterior 1 2 3 Siguiente

Cancelar
Aceptar

Figura 53: Botón (primera imagen) y ventana emergente para añadir una nueva ubicación en un ítem (segunda imagen).

- **Eliminar una ubicación:** El administrador podrá eliminar cualquier entrada de la tabla del historial de ubicaciones. En consecuencia, el periodo temporal liberado se marcará en la tabla como libre de ubicaciones. Para ello deberá pulsar el botón 'X' que aparece al lado de la localización de una entrada de la tabla de ubicaciones, después deberá confirmar la acción en la ventana emergente.

Histórico de ubicaciones en el museo +

Zona	Localización	Fecha de inicio	Fecha de finalización	Comentarios
Fondos del museo	Almacen X	2021-11-17 ↗	Actualidad ↗	Ubicación inicial por defecto ↗

Figura 54: Botón para eliminar una ubicación existente en un ítem.

- **Editar una localización de una ubicación:** El administrador podrá cambiar la localización asociada a una entrada de la tabla del historial de ubicaciones por cualquier otra localización del museo. Para ello deberá pulsar el icono del lápiz situado junto a una localización de la tabla de ubicaciones, después en la ventana emergente deberá seleccionar la nueva localización y confirmar la acción.

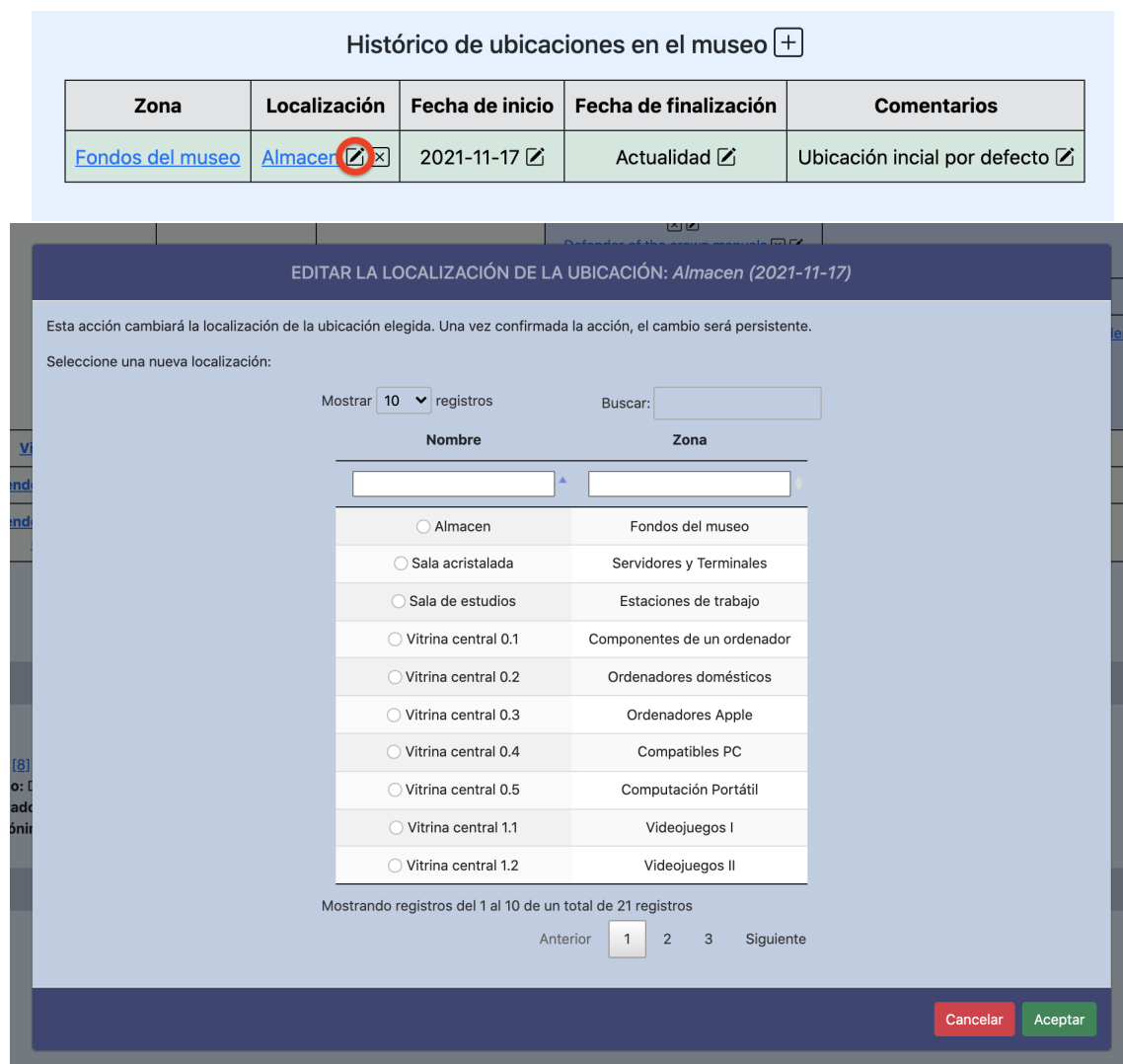


Figura 55: Botón (primera imagen) y ventana emergente para editar la localización de una ubicación (segunda imagen).

- Editar la fecha de inicio/finalización de una ubicación:** El administrador podrá cambiar tanto la fecha de inicio como la de finalización de una ubicación. En el caso de está última, si al cambiarla no especifica una concreta y se deja el campo vacío pasará a considerarse esa ubicación como la actual. Además, al igual que ocurre al añadir una ubicación, antes de hacer efectivo el cambio el

sistema comprobará que las fechas introducidas no interfieran en otros periodos de ubicaciones ya existentes. Para ello deberá pulsar el icono del lápiz junto a cada una de las fechas y en la ventana emergente seleccionar la nueva fecha antes de confirmar la acción.

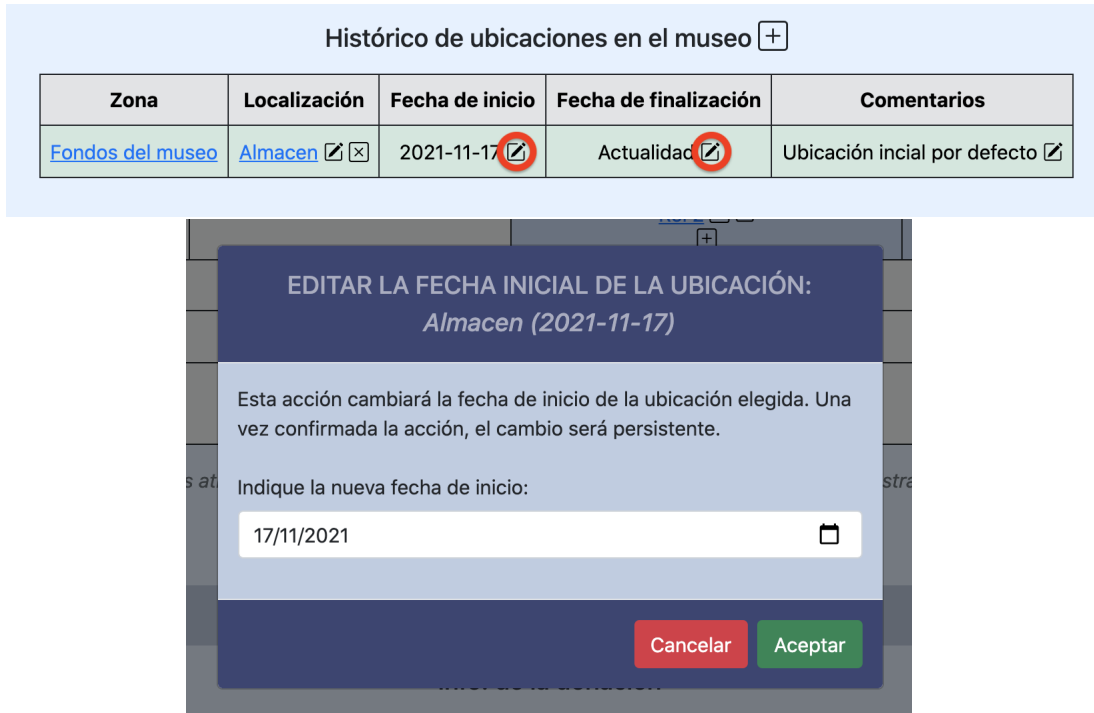


Figura 56: Botón (primera imagen) y ventana emergente para editar la fecha de inicio o de fin de una ubicación (segunda imagen).

- **Editar comentarios de una ubicación:** El administrador podrá cambiar los comentarios opcionales asociados a cada entrada del historial de ubicaciones. Para ello deberá pulsar el icono del lápiz situado junto al comentario y después en la ventana emergente modificar el texto antes de confirmar la acción.

Histórico de ubicaciones en el museo +

Zona	Localización	Fecha de inicio	Fecha de finalización	Comentarios
Fondos del museo	Almacen ✎ ✕	2021-11-17 ✎	Actualidad ✎	Ubicación inicial por defecto ✎

EDITAR LOS COMENTARIOS DE LA UBICACIÓN:
Almacen (2021-11-17)

Esta acción modificará los comentarios de la ubicación elegida. Una vez confirmada la acción, el cambio será persistente.

Modifique o indique los nuevos comentarios:

Ubicación inicial por defecto

Cancelar
Aceptar

Figura 57: Botón (primera imagen) y ventana emergente para editar los comentarios de una ubicación (segunda imagen).


```

<comentariosCampos></comentariosCampos>

<columnSize>5</columnSize>
<columnSize>11</columnSize>
<columnSize>10</columnSize>
<columnSize>2</columnSize>
<columnSize>10</columnSize>
<columnSize>20</columnSize>
<columnSize>>null</columnSize>
<columnSize>>null</columnSize>
<columnSize>>null</columnSize>
<columnSize>10</columnSize>

<inputSize>70px</inputSize>
<inputSize>>null</inputSize>
<inputSize>>null</inputSize>
<inputSize>>null</inputSize>
<inputSize>165px</inputSize>
<inputSize>160px</inputSize>
<inputSize>>null</inputSize>
<inputSize>475px</inputSize>
<inputSize>350px</inputSize>

<columnaOrden>4</columnaOrden>
  <tipoOrden>desc</tipoOrden>
</donacion>

<!-- ***** -->
<!-- Info sobre la tabla DONANTE -->
<!-- ***** -->

<donante>
  <nombresAtrPrincipales>ID</nombresAtrPrincipales>
  <nombresAtrPrincipales>eMail</nombresAtrPrincipales>
  <nombresAtrPrincipales>Nombre</nombresAtrPrincipales>
  <nombresAtrPrincipales>Teléfono</nombresAtrPrincipales>

  <nombresAtrSecundarios>Descripción</nombresAtrSecundarios>

  <inputTypesAtr>id</inputTypesAtr>
  <inputTypesAtr>email</inputTypesAtr>
  <inputTypesAtr>text</inputTypesAtr>
  <inputTypesAtr>tel</inputTypesAtr>
  <inputTypesAtr>textarea</inputTypesAtr>

  <infoAdicionalInputTypes>>null</infoAdicionalInputTypes>
  <infoAdicionalInputTypes></infoAdicionalInputTypes>
  <infoAdicionalInputTypes>required</infoAdicionalInputTypes>
  <infoAdicionalInputTypes>pattern="[0-9]{9}"</infoAdicionalInputTypes>
  <infoAdicionalInputTypes>>null</infoAdicionalInputTypes>

  <comentariosCampos></comentariosCampos>
  <comentariosCampos></comentariosCampos>
  <comentariosCampos></comentariosCampos>
  <comentariosCampos></comentariosCampos>
  <comentariosCampos></comentariosCampos>

  <columnSize>5</columnSize>

```

```

<columnSize>35</columnSize>
<columnSize>20</columnSize>
<columnSize>9</columnSize>
<columnSize>>null</columnSize>
<columnSize>10</columnSize>

<inputSize>70px</inputSize>
<inputSize>350px</inputSize>
<inputSize>350px</inputSize>
<inputSize>110px</inputSize>
<inputSize>350px</inputSize>

<columnaOrden>2</columnaOrden>
<tipoOrden>asc</tipoOrden>
</donante>

<!-- ***** -->
<!-- Info sobre la tabla ZONA -->
<!-- ***** -->

<zona>
<nombrsAtrPrincipales>Nombre</nombrsAtrPrincipales>
<nombrsAtrPrincipales>Código</nombrsAtrPrincipales>
<nombrsAtrPrincipales>En exposición</nombrsAtrPrincipales>

<nombrsAtrSecundarios>Nombre</nombrsAtrSecundarios>
<nombrsAtrSecundarios>Descripción</nombrsAtrSecundarios>

<inputTypesAtr>text</inputTypesAtr>
<inputTypesAtr>text</inputTypesAtr>
<inputTypesAtr>select</inputTypesAtr>
<inputTypesAtr>oculto</inputTypesAtr>
<inputTypesAtr>textarea</inputTypesAtr>

<infoAdicionalInputTypes>required</infoAdicionalInputTypes>
<infoAdicionalInputTypes>required</infoAdicionalInputTypes>
<infoAdicionalInputTypes>Sí;No</infoAdicionalInputTypes>
<infoAdicionalInputTypes>>null</infoAdicionalInputTypes>
<infoAdicionalInputTypes>>null</infoAdicionalInputTypes>

<comentariosCampos></comentariosCampos>
<comentariosCampos></comentariosCampos>
<comentariosCampos></comentariosCampos>
<comentariosCampos></comentariosCampos>
<comentariosCampos></comentariosCampos>

<columnSize>25</columnSize>
<columnSize>3</columnSize>
<columnSize>2</columnSize>
<columnSize>>null</columnSize>
<columnSize>>null</columnSize>
<columnSize>14</columnSize>

<inputSize>260px</inputSize>
<inputSize>60px</inputSize>
<inputSize>>null</inputSize>
<inputSize>>null</inputSize>
<inputSize>500px</inputSize>

```

```

        <columnaOrden>1</columnaOrden>
        <tipoOrden>asc</tipoOrden>
</zona>

<!-- ***** -->
<!-- Info sobre la tabla LOCALIZACION -->
<!-- ***** -->

<localizacion>
    <nombresAtrPrincipales>Nombre</nombresAtrPrincipales>
    <nombresAtrPrincipales>Código</nombresAtrPrincipales>
    <nombresAtrPrincipales>Zona</nombresAtrPrincipales>

    <nombresAtrSecundarios>Zona</nombresAtrSecundarios>
    <nombresAtrSecundarios>Nombre</nombresAtrSecundarios>
    <nombresAtrSecundarios>Descripción</nombresAtrSecundarios>

    <inputTypesAtr>text</inputTypesAtr>
    <inputTypesAtr>number</inputTypesAtr>
    <inputTypesAtr>no_crear_input</inputTypesAtr>
    <inputTypesAtr>ref</inputTypesAtr>
    <inputTypesAtr>oculto</inputTypesAtr>
    <inputTypesAtr>textarea</inputTypesAtr>

    <infoAdicionalInputTypes>required</infoAdicionalInputTypes>
    <infoAdicionalInputTypes>required</infoAdicionalInputTypes>
    <infoAdicionalInputTypes>null</infoAdicionalInputTypes>
    <infoAdicionalInputTypes>null</infoAdicionalInputTypes>
    <infoAdicionalInputTypes>null</infoAdicionalInputTypes>
    <infoAdicionalInputTypes>null</infoAdicionalInputTypes>

    <comentariosCampos></comentariosCampos>
    <comentariosCampos></comentariosCampos>
    <comentariosCampos></comentariosCampos>
    <comentariosCampos></comentariosCampos>
    <comentariosCampos></comentariosCampos>
    <comentariosCampos></comentariosCampos>

    <columnSize>25</columnSize>
    <columnSize>2</columnSize>
    <columnSize>25</columnSize>
    <columnSize>null</columnSize>
    <columnSize>null</columnSize>
    <columnSize>null</columnSize>
    <columnSize>10</columnSize>

    <inputSize>225px</inputSize>
    <inputSize>60px</inputSize>
    <inputSize>160px</inputSize>
    <inputSize>null</inputSize>
    <inputSize>null</inputSize>
    <inputSize>500px</inputSize>

    <columnaOrden>1</columnaOrden>
    <tipoOrden>asc</tipoOrden>
</localizacion>

```



```

<!-- ***** -->
<!-- Info sobre la tabla INSTANCIAS -->
<!-- ***** -->

<instancia>
  <nombrsAtrPrincipales>ID</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>Nombre</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>Tipo de ítem</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>Donación</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>Localización</nombrsAtrPrincipales>

  <nombrsAtrPrincipales_donacion>ID</nombrsAtrPrincipales_donacion>
  <nombrsAtrPrincipales_donacion>Nombre</nombrsAtrPrincipales_donacion>
  <nombrsAtrPrincipales_donacion>Tipo de ítem</nombrsAtrPrincipales_donacion>
  <nombrsAtrPrincipales_donacion>Localización</nombrsAtrPrincipales_donacion>

  <columnSize>5</columnSize>
  <columnSize>25</columnSize>
  <columnSize>25</columnSize>
  <columnSize>25</columnSize>
  <columnSize>25</columnSize>

  <columnSize_donacion>5</columnSize_donacion>
  <columnSize_donacion>25</columnSize_donacion>
  <columnSize_donacion>25</columnSize_donacion>
  <columnSize_donacion>25</columnSize_donacion>

  <columnaOrden>1</columnaOrden>
  <tipoOrden>asc</tipoOrden>
</instancia>

<!-- ***** -->
<!-- Info sobre la tabla RESUMEN DONANTES -->
<!-- ***** -->

<resumenDonantes>
  <nombrsAtrPrincipales>Nombre del donante</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>#Donaciones realizadas</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>#Ejemplares donados</nombrsAtrPrincipales>

  <columnSize>25</columnSize>
  <columnSize>6</columnSize>
  <columnSize>6</columnSize>

  <columnaOrden>1</columnaOrden>
  <tipoOrden>desc</tipoOrden>
</resumenDonantes>

<!-- ***** -->
<!-- Info sobre la tabla RESUMEN DONACIONES -->
<!-- ***** -->

<resumenDonaciones>
  <nombrsAtrPrincipales>Año</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>#Donaciones realizadas</nombrsAtrPrincipales>
  <nombrsAtrPrincipales>#Ejemplares donados</nombrsAtrPrincipales>

  <columnSize>4</columnSize>

```

```
<columnSize>6</columnSize>
<columnSize>6</columnSize>
    <columnOrden>0</columnOrden>
    <tipoOrden>desc</tipoOrden>
</resumenDonaciones>
</tablas>
```

E. Anexo: Contenido del fichero XML de configuración *variables_constantes.xml*

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<variables>
  <altura_max_imagen_principal><valor>275px</valor></altura_max_imagen_principal>
  <altura_max_imagenes_secundarias><valor>100px</valor></altura_max_imagenes_secundarias>
</variables>
```

F. Anexo: Diagrama de Gantt completo.

