



Creación del sitio web para el Museo de la Informática de la Escuela de Ingeniería Informática de Oviedo

GRADO EN INGENIERÍA INFORMÁTICA DEL SOFTWARE

TRABAJO FIN DE GRADO

AUTOR

M^a Isabel Fernández Pérez

TUTOR

José Manuel Redondo López

Julio 2021

Copyright (C) 2020 **ELENA ALLEGUE GONZÁLEZ, JOSÉ MANUEL REDONDO LÓPEZ**

Teaching Innovation Project: PINN-19-A-029 (University of Oviedo)

This work has been published in [1] [2]

Esta versión de la plantilla para Trabajos de Fin de Grado ha sido posible gracias a la donación de la ex-alumna Elena Allegue González de su documentación de Trabajo de Fin de Grado, que ha servido como base para elaborar esta versión. Aquí podréis encontrar todos los títulos y subtítulos de las secciones, pero las explicaciones se mantendrán en la versión *Word* de la plantilla (se proporciona una versión PDF de la misma para facilitar el acceso a las mismas). No obstante, del trabajo de Elena se han conservado ejemplos de como hacer elementos clave como imágenes, tablas, etc.

Desarrollar una versión *Latex* de la plantilla desde cero es una trabajo bastante largo, pero gracias al trabajo de Elena se ha podido equiparar esta versión con las de *Word* mucho más rápidamente.

Agradecimientos

Índice general

1. PSI: PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN	11
1.1. PSI 1: INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN	12
1.1.1. PSI 1.1: Análisis de la Necesidad del PSI	12
1.1.2. PSI 1.2: Identificación del Alcance del PSI	12
1.1.3. PSI 1.3: Determinación de Responsables	12
1.2. PSI 7: DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA	14
1.2.1. PSI 7.2: Selección de la Arquitectura Tecnológica	14
2. ESTUDIO DE VIABILIDAD DEL SISTEMA	15
2.1. EVS 4, 5, 6: ESTUDIO Y VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN. SELECCIÓN DE ALTERNATIVA FINAL	16
2.1.1. Evaluación de alternativas de desarrollo	16
2.1.2. Evaluación de alternativas de gestor de bases de datos	17
3. PLANIFICACIÓN Y GESTIÓN DEL TFG	18
3.1. PLANIFICACIÓN DEL PROYECTO	19
3.1.1. Identificación de Interesados	19
3.1.2. OBS y PBS	19
3.1.3. Planificación Inicial. WBS	19
3.1.4. Riesgos	19
3.1.5. Presupuesto Inicial	19
3.2. EJECUCIÓN DEL PROYECTO	20
3.2.1. Plan Seguimiento de Planificación	20
3.2.2. Bitácora de Incidencias del Proyecto	20
3.2.3. Riesgos	20
3.3. CIERRE DEL PROYECTO	21
3.3.1. Planificación Final	21
3.3.2. Informe Final de Riesgos	21

3.3.3. Presupuesto Final de Costes	21
3.3.4. Informe de Lecciones Aprendidas	21
4. ANÁLISIS DEL SISTEMA DE INFORMACIÓN	22
4.1. ASI 2: ESTABLECIMIENTO DE REQUISITOS	23
4.1.1. Obtención de los Requisitos del Sistema	23
4.1.2. Identificación de Actores del Sistema	27
4.1.3. Especificación de Casos de Uso	27
4.2. ASI 4: ANÁLISIS DE LOS CASOS DE USO	31
4.2.1. Caso de Uso 1	31
4.2.2. Caso de Uso 2	31
4.2.3. Caso de Uso 3	32
4.2.4. Caso de Uso 4	32
4.2.5. Caso de Uso 5	33
4.2.6. Caso de Uso 6	33
4.2.7. Caso de Uso 7	34
4.2.8. Caso de Uso 8	34
4.2.9. Caso de Uso 9	35
4.2.10. Caso de Uso 10	35
4.2.11. Caso de Uso 11	36
4.3. ASI 5: ANÁLISIS DE CLASES	37
4.3.1. Diagrama de Clases	37
4.3.2. Descripción de las Clases	38
4.4. ASI 8: DEFINICIÓN DE INTERFACES DE USUARIO	48
4.4.1. Definición del aspecto de la interfaz	48
4.4.2. Diagrama de Navegabilidad	53
4.5. ASI 10: ESPECIFICACIÓN DEL PLAN DE PRUEBAS	55
4.5.1. Pruebas unitarias	55
4.5.2. Pruebas del sistema	57
4.5.3. Pruebas de usabilidad	58
5. DISEÑO DEL SISTEMA DE INFORMACIÓN	59
5.1. DSI 4: DISEÑO DE CLASES	60
5.1.1. Diagrama de Clases	60
5.2. DSI 5: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA	62
5.2.1. Diagrama de paquetes	62

5.2.2. Diagrama de componentes	62
5.2.3. Diagrama de despliegue	62
5.3. DSI 6: DISEÑO FÍSICO DE DATOS	63
5.3.1. Descripción del SGBD Usado	63
5.3.2. Integración del SGBD en Nuestro Sistema	63
5.3.3. Diagrama E-R	63
5.4. DSI 10: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS	65
5.4.1. Pruebas Unitarias y del Sistema	65
5.4.2. Pruebas de Usabilidad y Accesibilidad	68
6. CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN	73
6.1. CSI 1: PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN	74
6.1.1. Estándares y normas seguidos	74
6.1.2. Lenguajes de programación	74
6.1.3. Herramientas y programas usados para el desarrollo	75
6.2. CSI 2: GENERACIÓN DEL CÓDIGO DE LOS COMPONENTES Y PROCEDIMIENTOS	77
6.3. CSI 3: EJECUCIÓN DE LAS PRUEBAS UNITARIAS	78
6.4. CSI 4: EJECUCIÓN DE LAS PRUEBAS DE INTEGRACIÓN	79
6.5. CSI 5: EJECUCIÓN DE LAS PRUEBAS DEL SISTEMA	82
6.5.1. Prueba de Usabilidad	82
6.6. CSI 6: ELABORACIÓN DE LOS MANUALES DE USUARIO	93
6.6.1. Manual de Instalación	93
6.6.2. Manual de Ejecución	93
6.6.3. Manual de Usuario	94
6.6.4. Manual del Programador	102
7. IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA	104
7.1. IAS 1: ESTABLECIMIENTO DEL PLAN DE IMPLANTACIÓN	105
7.2. IAS 4: CARGA DE DATOS AL ENTORNO DE OPERACIÓN	106
7.3. IAS 5: PRUEBAS DE IMPLANTACIÓN DEL SISTEMA	107
7.4. IAS 7: PREPARACIÓN DEL MANTENIMIENTO DEL SISTEMA	108
8. APÉNDICES	109
8.1. PROBLEMAS ENCONTRADOS DURANTE EL DESARROLLO	110
8.2. CONCLUSIONES	111

8.3. AMPLIACIONES	112
ANEXOS	115
PLAN DE GESTIÓN DE RIESGOS	116
CONTENIDO ENTREGADO EN LOS ANEXOS	117

Índice de figuras

1.1.	Diagrama de la arquitectura tecnológica	14
2.1.	Logos de JavaScript y Node.js	16
2.2.	Logos de Angular, TypeScript y PHP	17
2.3.	Logo de MySQL	17
4.1.	Diagrama de casos de uso del museo	27
4.2.	Diagrama de casos de uso de la administración del museo	28
4.3.	Análisis de clases: diagrama de clases del museo	37
4.4.	Análisis de clases: diagrama de clases de la administración	38
4.5.	Prototipo: Página de inicio	48
4.6.	Prototipo: Página de la vista general del museo	49
4.7.	Prototipo: Página de detalles del periodo (museo)	49
4.8.	Prototipo: Página de detalles del componente (museo)	50
4.9.	Prototipo: Página de inicio de sesión	50
4.10.	Prototipo: Página de listado de periodos	51
4.11.	Prototipo: Página de detalles de un periodo (admimnistración)	51
4.12.	Prototipo: Página de detalles de un componente (admimnistración)	52
4.13.	Prototipo: Formulario para añadir o editar un periodo	52
4.14.	Prototipo: Formulario para añadir o editar un componente	53
4.15.	Diagrama de navegabilidad del museo	53
4.16.	Diagrama de navegabilidad de la administración del museo	54
5.1.	Diseño de clases: diagrama de clases del museo	60
5.2.	Diseño de clases: diagrama de clases de la administración del museo	61
5.3.	Diagrama Entidad-Relación de la base de datos creada	64
6.1.	Logo de Visual Studio Code	75
6.2.	Logo de XAMPP	75

6.3.	Logo de MobaXTerm	75
6.4.	Logo de Git	76
6.5.	Instalación de Angular CLI	93
6.6.	Instalación de los paquetes del proyecto del museo	93
6.7.	Instalación de los paquetes del proyecto de administración	93
6.8.	Ejecución de la aplicación del museo	94
6.9.	Ejecución de la aplicación de administración	94
6.10.	Manual de usuario: Inicio	95
6.11.	Manual de usuario: Vista general del museo	96
6.12.	Manual de usuario: Detalles del periodo (museo)	96
6.13.	Manual de usuario: Detalles del componente (museo)	97
6.14.	Manual de usuario: Inicio de sesión	97
6.15.	Manual de usuario: Cambio de contraseña	98
6.16.	Manual de usuario: Listado de periodos	98
6.17.	Manual de usuario: Detalles de un periodo (administración)	99
6.18.	Manual de usuario: Detalles de un componente (administración)	99
6.19.	Manual de usuario: Formulario para añadir un periodo	100
6.20.	Manual de usuario: Formulario para editar un periodo	100
6.21.	Manual de usuario: Formulario para añadir un componente	101
6.22.	Manual de usuario: Formulario para editar un componente	102

Índice de tablas

4.1. Especificación Caso de Uso 1	27
4.2. Especificación Caso de Uso 2	28
4.3. Especificación Caso de Uso 3	29
4.4. Especificación Caso de Uso 4	29
4.5. Especificación Caso de Uso 5	29
4.6. Especificación Caso de Uso 6	29
4.7. Especificación Caso de Uso 7	29
4.8. Especificación Caso de Uso 8	30
4.9. Especificación Caso de Uso 9	30
4.10. Especificación Caso de Uso 10	30
4.11. Especificación Caso de Uso 11	30
4.12. Análisis del Caso de Uso 1	31
4.13. Análisis del Caso de Uso 2	31
4.14. Análisis del Caso de Uso 3	32
4.15. Análisis del Caso de Uso 4	32
4.16. Análisis del Caso de Uso 5	33
4.17. Análisis del Caso de Uso 6	33
4.18. Análisis del Caso de Uso 7	34
4.19. Análisis del Caso de Uso 8	34
4.20. Análisis del Caso de Uso 9	35
4.21. Análisis del Caso de Uso 10	35
4.22. Análisis del Caso de Uso 11	36
4.23. Descripción de la clase Period	39
4.24. Descripción de la interfaz MyComponent	39
4.25. Descripción de la clase Cpu	40
4.26. Descripción de la clase PeriodService (museo)	40
4.27. Descripción de la clase CompService (museo)	41

4.28. Descripción de la clase TimelineComponent	41
4.29. Descripción de la clase PeriodDetailsComponent	42
4.30. Descripción de la clase CompDetailsComponent	42
4.31. Descripción de la clase UserService	43
4.32. Descripción de la clase PeriodService (administración)	43
4.33. Descripción de la clase CompService (administración)	44
4.34. Descripción de la clase LoginComponent	44
4.35. Descripción de la clase ListPeriodsComponent	45
4.36. Descripción de la clase PeriodComponent	45
4.37. Descripción de la clase AddPeriodComponent	45
4.38. Descripción de la clase EditPeriodComponent	46
4.39. Descripción de la clase MyCompComponent	46
4.40. Descripción de la clase AddCompComponent	46
4.41. Descripción de la clase EditCompComponent	47
4.42. Pruebas unitarias: Caso de uso 1	55
4.43. Pruebas unitarias: Caso de uso 2	55
4.44. Pruebas unitarias: Caso de uso 3	55
4.45. Pruebas unitarias: Caso de uso 4	55
4.46. Pruebas unitarias: Caso de uso 5	56
4.47. Pruebas unitarias: Caso de uso 6	56
4.48. Pruebas unitarias: Caso de uso 7	56
4.49. Pruebas unitarias: Caso de uso 8	56
4.50. Pruebas unitarias: Caso de uso 9	57
4.51. Pruebas unitarias: Caso de uso 10	57
4.52. Pruebas unitarias: Caso de uso 11	57
5.1. Diseño de pruebas: Consultar periodos (museo)	65
5.2. Diseño de pruebas: Consultar componentes (museo)	65
5.3. Diseño de pruebas: Iniciar sesión	65
5.4. Diseño de pruebas: Consultar periodos (administración)	66
5.5. Diseño de pruebas: Añadir periodo	66
5.6. Diseño de pruebas: Modificar periodo	66
5.7. Diseño de pruebas: Eliminar periodo	67
5.8. Diseño de pruebas: Consultar componentes (administración)	67
5.9. Diseño de pruebas: Añadir componente	67
5.10. Diseño de pruebas: Modificar componente	68

5.11. Diseño de pruebas: Eliminar componente	68
5.12. Pruebas de usabilidad: preguntas de carácter general	69
5.13. Pruebas de usabilidad: preguntas sobre la aplicación	71
5.14. Pruebas de usabilidad: cuestionario para el responsable de las pruebas	72
6.1. Ejecución de pruebas: Consultar periodos (museo)	79
6.2. Ejecución de pruebas: Consultar componentes (museo)	79
6.3. Ejecución de pruebas: Iniciar sesión	79
6.4. Ejecución de pruebas: Consultar periodos (administración)	79
6.5. Ejecución de pruebas: Añadir periodo	80
6.6. Ejecución de pruebas: Modificar periodo	80
6.7. Ejecución de pruebas: Eliminar periodo	80
6.8. Ejecución de pruebas: Consultar componentes (administración)	81
6.9. Ejecución de pruebas: Añadir componente	81
6.10. Ejecución de pruebas: Modificar componente	81
6.11. Ejecución de pruebas: Eliminar componente	81
6.12. Checklist de pruebas de Usabilidad	82
6.13. Cuestionario de carácter general (Usuario 1)	87
6.14. Cuestionario sobre la aplicación (Usuario 1)	88
6.15. Actividades guiadas (Usuario 1)	89
6.16. Cuestionario de carácter general (Usuario 2)	90
6.17. Cuestionario sobre la aplicación (Usuario 2)	91
6.18. Actividades guiadas (Usuario 2)	92

Capítulo 1

PSI: PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE PLANIFICACIÓN

PSI

1.1. PSI 1: INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN

1.1.1. PSI 1.1: Análisis de la Necesidad del PSI

El tutor de este trabajo de fin de grado, José Manuel Redondo, ha propuesto el desarrollo de una aplicación web para el Museo de la Informática de Asturias, que contenga toda la información disponible sobre los componentes del museo y la muestre de forma ordenada para que las personas interesadas puedan acceder a ella fácilmente. El sistema será gestionado directamente por el tutor del trabajo.

El sistema debe identificar cada componente y mostrar la información disponible del mismo,. El software permitirá añadir la información de las nuevas piezas que puedan ser incluidas en la exposición en un futuro gracias a donaciones o compras. Los componentes serán ordenados según su tipo y la época a la que pertenecen.

1.1.2. PSI 1.2: Identificación del Alcance del PSI

Actualmente los carteles informativos sobre las piezas del museo se encuentran expuestos en la Escuela de Ingeniería Informática. Los objetivos de este proyecto son los siguientes:

- Recopilar los datos disponibles de las piezas que se encuentran actualmente en el Museo e introducirlos en una base de datos.
- Mostrar una linea temporal con los diferentes periodos a los que pertenecen los componentes del Museo.
- Permitir acceder a cada periodo para ver los componentes del mismo.
- Organizar las diferentes piezas en función de su tipo y de la familia de la que forman parte.
- Presentar la información disponible de cada pieza, así como imágenes de la misma y otras curiosidades.

En definitiva, estos objetivos se pueden resumir en:

- Permitir a los usuarios visitar el Museo de la Informática de forma online, ofreciendo la misma información que se encuentra disponible en la exposición física.
- Facilitar al administrador la inserción de nuevos periodos y componentes.

1.1.3. PSI 1.3: Determinación de Responsables

- **El proyectante** se encargará del desarrollo del software descrito y de realizar la carga de los datos disponibles a la base de datos correspondientes.

- **El tutor del proyecto** se encargará de la supervisión de las fases del proyecto y de su validación.
- **Una serie de usuarios escogidos aleatoriamente** realizará pruebas del software para comprobar su correcto funcionamiento.

1.2. PSI 7: DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

1.2.1. PSI 7.2: Selección de la Arquitectura Tecnológica

Al tratarse de una aplicación Angular, seguirá el patrón Modelo Vista Vista-Modelo (MVVM), una variación del Modelo Vista Controlador, patrón arquitectónico que separa los datos y la lógica de una aplicación de su representación. En la variación MVVM, la vista y el modelo son muy dependientes entre sí.

En este caso la vista estará compuesta por las *templates* de Angular, que son componentes HTML.

El modelo se corresponde con las clases de Angular que modelan las entidades de la base de datos MySQL, a la que se accede desde un servidor Apache que aloja los archivos PHP necesarios.

Por último, la vista-modelo es la propia aplicación de Angular.

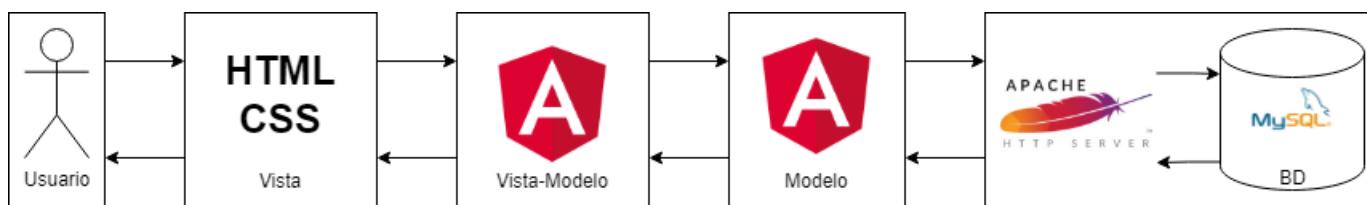


Figura 1.1: Diagrama de la arquitectura tecnológica

Capítulo 2

ESTUDIO DE VIABILIDAD DEL SISTEMA

FASE DE DESARROLLO

EVS

2.1. EVS 4, 5, 6: ESTUDIO Y VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN. SELECCIÓN DE ALTERNATIVA FINAL

2.1.1. Evaluación de alternativas de desarrollo

2.1.1.1. JavaScript y Node.js

JavaScript es uno de los lenguajes más populares actualmente. Está basado en el estándar ECMAScript. Se trata un lenguaje interpretado, se compila en tiempo de ejecución. Es orientado a objetos, débilmente tipado y dinámico[3].

Node.js es un entorno de ejecución de JavaScript orientado a eventos asíncronos, en el que no hace falta utilizar hilos. Utiliza un modelo de entrada y salida sin bloqueo, lo que asegura un rendimiento más eficiente de la aplicación y evita que se produzca una gran sobrecarga del lado del servidor. Por ello, es muy apropiado para desarrollar sistemas escalables[4]. Además, puede ser utilizado tanto en el lado del cliente como en el servidor, por lo que no se necesitaría una tecnología adicional para el back-end.

Esta fue la primera opción barajada, ya que había utilizado anteriormente estas tecnologías y podría aprovechar este proyecto para profundizar en su aprendizaje.



Figura 2.1: Logos de JavaScript y Node.js

2.1.1.2. Angular, TypeScript y PHP

La otra opción considerada fue Angular y TypeScript, debido a su popularidad. No había trabajado con ellas antes, y esta sería una buena oportunidad para conocerlas.

Angular es un framework desarrollado en TypeScript y utilizado habitualmente para crear aplicaciones de una sola página. Se basa en la utilización de componentes web reutilizables para crear aplicaciones web fácilmente escalables. Angular extiende la sintaxis de HTML y actualiza automáticamente el árbol DOM cuando el estado de un componente cambia. Cuenta con gran cantidad de librerías y es uno de los frameworks más utilizados en la industria actual[5].

TypeScript es un lenguaje de programación que extiende JavaScript añadiendo la definición de tipos estáticos. Al compilarlo se transforma en código JavaScript siguiendo todos los estándares, y puede ejecutarse en cualquier lugar que execute JavaScript[6].

En este caso, Angular y TypeScript son ambas tecnologías de front-end, por tanto necesitamos una tercera tecnología para el back-end de esta aplicación. Para ello consideré

como opción PHP, lenguaje que se ejecuta en el servidor y envía el resultado generado al cliente, y que es otra de las tecnologías más reconocidas y usadas en el desarrollo web actualmente y desde su creación.



Figura 2.2: Logos de Angular, TypeScript y PHP

Ambas opciones son de código abierto, lo que me parece un punto positivo ya que, gracias a la colaboración de la comunidad, se consigue una alta calidad en el software.

Finalmente, me decidí por Angular, TypeScript y PHP, principalmente por la razón de profundizar en el aprendizaje de estas tecnologías tan importantes actualmente en el desarrollo de aplicaciones web.

2.1.2. Evaluación de alternativas de gestor de bases de datos

2.1.2.1. MySQL

MySQL es un SGBD relacional de código abierto con un modelo cliente-servidor. Ha sido la única opción considerada al tratarse de la base de datos relacional que es comúnmente utilizada con Angular, y no se ha encontrado ninguna necesidad o ventaja de usar un sistema no relacional.



Figura 2.3: Logo de MySQL

Capítulo 3

PLANIFICACIÓN Y GESTIÓN DEL TFG

3.1. PLANIFICACIÓN DEL PROYECTO

3.1.1. Identificación de Interesados

3.1.2. OBS y PBS

3.1.3. Planificación Inicial. WBS

3.1.4. Riesgos

3.1.4.1. Plan de Gestión de Riesgos

3.1.4.2. Identificación de Riesgos

3.1.4.3. Registro de Riesgos

3.1.5. Presupuesto Inicial

3.1.5.1. Presupuesto de Costes

3.1.5.2. Presupuesto de Cliente

3.2. EJECUCIÓN DEL PROYECTO

3.2.1. Plan Seguimiento de Planificación

3.2.2. Bitácora de Incidencias del Proyecto

3.2.3. Riesgos

3.3. CIERRE DEL PROYECTO

- 3.3.1. Planificación Final**
- 3.3.2. Informe Final de Riesgos**
- 3.3.3. Presupuesto Final de Costes**
- 3.3.4. Informe de Lecciones Aprendidas**

Capítulo 4

ANÁLISIS DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

ASI

4.1. ASI 2: ESTABLECIMIENTO DE REQUISITOS

4.1.1. Obtención de los Requisitos del Sistema

4.1.1.1. Requisitos de interfaces externas

Interfaces de usuario

RIE-IU-1. El sistema será accesible desde cualquier dispositivo que cuente con conexión a internet y un navegador web.

RIE-IU-2. El sistema estará disponible en diferentes idiomas.

RIE-IU-2.1. Español

RIE-IU-2.2. Inglés

RIE-IU-3. El sistema deberá ser accesible para todos los usuarios a través de los navegadores más comunes.

RIE-IU-3.1. Google Chrome

RIE-IU-3.2. Mozilla Firefox

RIE-IU-3.3. Microsoft Edge

RIE-IU-4. El usuario podrá utilizar todas las funcionalidades desarrolladas de la aplicación sin inconvenientes.

RIE-IU-5. El usuario no necesitará de conocimientos tecnológicos avanzados.

Interfaces hardware

RIE-IH-1. El sistema dispondrá de una base de datos para almacenar la información necesaria.

Interfaces de comunicaciones

RIE-IC-1. El sistema contendrá enlaces a diferentes sitios web.

RIE-IC-2. El sistema mostrará por defecto enlaces a los siguientes sitios web.

RIE-IC-2.1. Twitter oficial de la Escuela de Ingeniería Informática

RIE-IC-2.2. Página web de la Escuela de Ingeniería Informática

RIE-IC-2.3. Página web de la Universidad de Oviedo

4.1.1.2. Requisitos funcionales

RF-1. El sistema estará constituido por dos aplicaciones web diferentes.

RF-1.1. El museo.

RF-1.2. La administración del museo.

Museo

RF-MU-1. El sistema mostrará los periodos existentes.

RF-MU-1.1. Los periodos estarán ordenados por año de inicio.

RF-MU-1.2. Se presentarán en una línea temporal incluyendo los siguientes datos.

RF-MU-1.2.1. Nombre.

RF-MU-1.2.2. Año de inicio.

RF-MU-1.2.3. Año de fin.

RF-MU-1.2.4. Nombres de los componentes pertenecientes al periodo.

RF-MU-2. El sistema permitirá realizar búsquedas.

RF-MU-2.1. Se podrá buscar por los siguientes campos.

RF-MU-2.1.1. Por nombre.

RF-MU-2.1.2. Por un intervalo de años.

RF-MU-2.2. Al realizar la búsqueda se mostrará la línea temporal filtrada con los resultados.

RF-MU-3. El sistema mostrará los detalles de un periodo.

RF-MU-3.1. Nombre.

RF-MU-3.2. Características.

RF-MU-3.3. Listado de datos curiosos (Sabías qué...).

RF-MU-3.4. Eventos ocurridos durante el periodo.

RF-MU-3.5. Sistemas famosos que contienen componentes de este periodo.

RF-MU-3.6. Listado de los componentes pertenecientes al periodo.

RF-MU-4. El sistema mostrará los detalles de un componente.

RF-MU-4.1. Nombre.

RF-MU-4.2. Descripción.

RF-MU-4.3. Imágenes.

RF-MU-4.4. Año de inicio.

RF-MU-4.5. Año de fin.

RF-MU-4.6. Precio.

RF-MU-4.7. Tipo de dispositivos en los que se encuentra.

RF-MU-4.7.1. Portátiles.

RF-MU-4.7.2. De escritorio.

RF-MU-4.8. Detalles específicos del tipo de componente.

RF-MU-4.8.1. Detalles de CPU.

RF-MU-4.8.1.1. Memoria ROM (obligatorio).

RF-MU-4.8.1.2. Memoria RAM (obligatorio).

RF-MU-4.8.1.3. Velocidad de reloj (obligatorio).

RF-MU-4.8.1.4. Potencia (obligatorio).

- RF-MU-4.8.1.5.** Tamaño de palabra (obligatorio).
- RF-MU-4.8.1.6.** Nanómetros de los transistores (obligatorio).
- RF-MU-4.8.1.7.** Passmark (obligatorio).
- RF-MU-4.8.1.8.** Número de transistores (obligatorio).

Administración del museo

RF-ADM-1. El sistema permitirá al usuario iniciar sesión mediante un formulario.

RF-ADM-1.1. El formulario se solicitan los siguientes campos.

- RF-ADM-1.1.1.** Correo electrónico (obligatorio).
- RF-ADM-1.1.2.** Contraseña (obligatorio).

RF-ADM-1.2. Si los campos no son correctos, se mostrará de nuevo el inicio de sesión.

RF-ADM-1.3. Si los campos son válidos, se accederá al sistema como administrador.

RF-ADM-2. El sistema mostrará un listado de los períodos existentes.

RF-ADM-3. El sistema mostrará los detalles de un periodo.

RF-ADM-3.1. Especificados en **RF-MU-3**.

RF-ADM-4. El sistema mostrará los detalles de un componente.

RF-ADM-4.1. Especificados en **RF-MU-4**.

RF-ADM-5. El sistema permitirá añadir un periodo.

RF-ADM-5.1. Se introducirán los datos mediante un formulario con los siguientes campos.

- RF-ADM-5.1.1.** Nombre (obligatorio).
- RF-ADM-5.1.2.** Características (obligatorio).
- RF-ADM-5.1.3.** Listado de datos curiosos (Sabías qué...) (obligatorio).
- RF-ADM-5.1.4.** Eventos ocurridos durante el periodo (obligatorio).

RF-ADM-6. El sistema permitirá añadir un componente a un periodo existente.

RF-ADM-6.1. Se introducirán los datos mediante un formulario con los siguientes campos.

- RF-ADM-6.1.1.** Nombre (obligatorio).
- RF-ADM-6.1.2.** Tipo de componente (obligatorio).
 - RF-ADM-6.1.2.1.** CPU.
- RF-ADM-6.1.3.** Descripción (obligatorio).
- RF-ADM-6.1.4.** Familia de componente (obligatorio).
- RF-ADM-6.1.5.** Imágenes.
- RF-ADM-6.1.6.** Año de inicio (obligatorio).
- RF-ADM-6.1.7.** Año de fin (obligatorio).
- RF-ADM-6.1.8.** Precio (obligatorio).
- RF-ADM-6.1.9.** Tipo de dispositivos en los que se encuentra.
 - RF-ADM-6.1.9.1.** Portátiles.

RF-ADM-6.1.9.2. De escritorio.

RF-ADM-6.1.10. Sistema famoso que lo contiene.

RF-ADM-6.1.10.1. Nombre del sistema.

RF-ADM-6.1.10.2. Imagen del sistema.

RF-ADM-6.1.11. Detalles específicos del tipo de componente.

RF-ADM-6.1.11.1. Detalles de CPU.

RF-ADM-6.1.11.1.1. Memoria ROM (obligatorio).

RF-ADM-6.1.11.1.2. Memoria RAM (obligatorio).

RF-ADM-6.1.11.1.3. Velocidad de reloj (obligatorio).

RF-ADM-6.1.11.1.4. Potencia (obligatorio).

RF-ADM-6.1.11.1.5. Tamaño de palabra (obligatorio).

RF-ADM-6.1.11.1.6. Nanómetros de los transistores (obligatorio).

RF-ADM-6.1.11.1.7. Passmark (obligatorio).

RF-ADM-6.1.11.1.8. Número de transistores (obligatorio).

RF-ADM-7. El sistema permitirá editar períodos.

RF-ADM-7.1. Se introducirán los datos mediante un formulario.

RF-ADM-7.1.1. Campos especificados en **RF-ADM-5.1**.

RF-ADM-8. El sistema permitirá editar componentes.

RF-ADM-8.1. Se introducirán los datos mediante un formulario.

RF-ADM-8.1.1. Campos especificados en **RF-ADM-6.1**.

RF-ADM-9. El sistema permitirá eliminar un periodo.

RF-ADM-9.1. Se pedirá confirmación antes de eliminarlo.

RF-ADM-9.2. Se eliminarán los componentes pertenecientes a dicho periodo.

RF-ADM-10. El sistema permitirá eliminar un componente.

RF-ADM-10.1. Se pedirá confirmación antes de eliminarlo.

4.1.1.3. Atributos del sistema

Seguridad

RNF-SEG-1. La parte de administración del sistema se asegurará de que el usuario se identifica para acceder a ella.

RNF-SEG-1.1. El usuario se identificará mediante un email y una contraseña.

RNF-SEG-2. El sistema cifrará la contraseña para almacenarla en la base de datos.

4.1.2. Identificación de Actores del Sistema

4.1.2.1. Usuario estándar

Actor que interactúa con el sistema. Tiene acceso de lectura a la página web del museo. Solo debe tener un conocimiento básico para navegar por internet.

4.1.2.2. Usuario administrador

Actor que interactúa con el sistema. Debe iniciar sesión en la parte de administración del sistema, es el único actor con acceso a esta. Es responsable de gestionar el sistema y su mantenimiento. Debe tener amplios conocimientos sobre el sistema.

4.1.3. Especificación de Casos de Uso

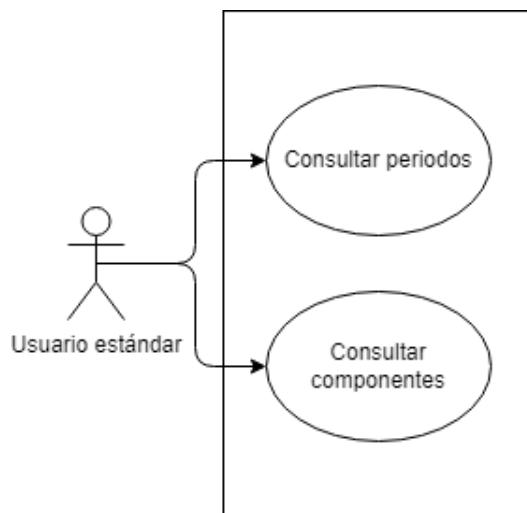


Figura 4.1: Diagrama de casos de uso del museo

Tabla 4.1: Especificación Caso de Uso 1

Nombre del caso de uso
Consultar periodos (museo)
Descripción
Un usuario estándar puede visualizar los períodos existentes en el museo.

Tabla 4.2: Especificación Caso de Uso 2

Nombre del caso de uso
Consultar componentes (museo)
Descripción
Un usuario estándar puede visualizar los componentes pertenecientes a cada periodo del museo.

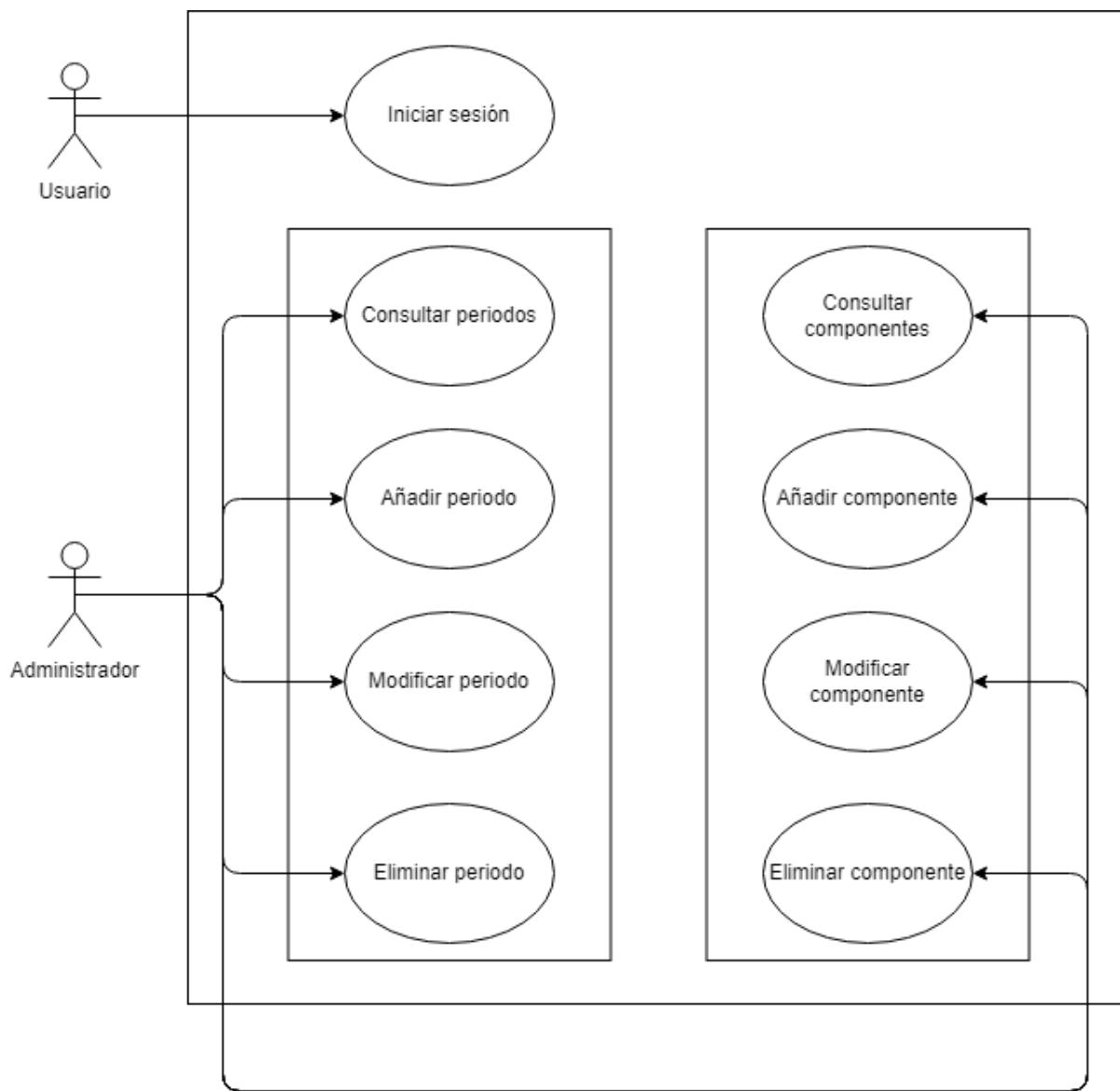


Figura 4.2: Diagrama de casos de uso de la administración del museo

Tabla 4.3: Especificación Caso de Uso 3

Nombre del caso de uso
Iniciar sesión
Descripción
Un usuario puede iniciar sesión en la aplicación para acceder a esta como administrador.

Tabla 4.4: Especificación Caso de Uso 4

Nombre del caso de uso
Consultar periodos (administración)
Descripción
El administrador puede consultar los períodos existentes en el sistema.

Tabla 4.5: Especificación Caso de Uso 5

Nombre del caso de uso
Añadir periodo
Descripción
El administrador puede añadir períodos a la base de datos del sistema.

Tabla 4.6: Especificación Caso de Uso 6

Nombre del caso de uso
Modificar periodo
Descripción
El administrador puede modificar los períodos existentes en la base de datos del sistema.

Tabla 4.7: Especificación Caso de Uso 7

Nombre del caso de uso
Eliminar periodo
Descripción
El administrador puede eliminar los períodos existentes en la base de datos del sistema.

Tabla 4.8: Especificación Caso de Uso 8

Nombre del caso de uso
Consultar componentes (administración)
Descripción
El administrador puede consultar los componentes existentes en el sistema.

Tabla 4.9: Especificación Caso de Uso 9

Nombre del caso de uso
Añadir componente
Descripción
El administrador puede añadir componentes a los períodos existentes en la base de datos del sistema.

Tabla 4.10: Especificación Caso de Uso 10

Nombre del caso de uso
Modificar componente
Descripción
El administrador puede modificar los componentes existentes en la base de datos del sistema.

Tabla 4.11: Especificación Caso de Uso 11

Nombre del caso de uso
Eliminar componente
Descripción
El administrador puede eliminar los componentes existentes en la base de datos del sistema.

4.2. ASI 4: ANÁLISIS DE LOS CASOS DE USO

4.2.1. Caso de Uso 1

Tabla 4.12: Análisis del Caso de Uso 1

Consultar periodos (museo)	
Precondiciones	-
Postcondiciones	-
Actores	Usuario estándar
Descripción	El usuario accederá a la vista principal del museo y podrá visualizar los períodos existentes. Podrá acceder a los períodos. Podrá acceder a los componentes de los períodos. Podrá realizar una búsqueda.
Escenarios Secundarios	

4.2.2. Caso de Uso 2

Tabla 4.13: Análisis del Caso de Uso 2

Consultar componentes (museo)	
Precondiciones	-
Postcondiciones	-
Actores	Usuario estándar
Descripción	El usuario accederá a la vista de un periodo y podrá visualizar los componentes pertenecientes al mismo. Podrá acceder a otros períodos. Podrá acceder a los otros componentes de ese periodo.
Escenarios Secundarios	

4.2.3. Caso de Uso 3

Tabla 4.14: Análisis del Caso de Uso 3

Iniciar sesión	
Precondiciones	El usuario no debe haber iniciado sesión.
Postcondiciones	-
Actores	Usuario
Descripción	El usuario accederá a la página principal de la aplicación de administración e introducirá su email y contraseña para iniciar sesión en el sistema.
Escenarios Secundarios	Los datos introducidos no se corresponden con los datos de un usuario con permiso de administrador. Se muestra un error y de nuevo se solicita iniciar sesión.

4.2.4. Caso de Uso 4

Tabla 4.15: Análisis del Caso de Uso 4

Consultar períodos (administración)	
Precondiciones	El usuario debe haber iniciado sesión.
Postcondiciones	-
Actores	Usuario administrador
Descripción	El usuario accederá al listado de períodos. Podrá acceder a cada uno de ellos.
Escenarios Secundarios	Aún no existe ningún periodo en el sistema. Se muestra una tabla vacía.

4.2.5. Caso de Uso 5

Tabla 4.16: Análisis del Caso de Uso 5

Añadir periodo	
Precondiciones	El usuario debe haber iniciado sesión.
Postcondiciones	El periodo añadido se guardará en la base de datos.
Actores	Usuario administrador
Descripción	El usuario accede al formulario para añadir un periodo. Rellena los campos necesarios. Pulsa el botón de guardar.
Escenarios Secundarios	<ul style="list-style-type: none"> - Se pulsa el botón cancelar. El formulario se restablece. - Se intenta acceder a otra página de la aplicación sin haber guardado los cambios. Se avisa de la situación y se pide una confirmación para continuar. - No se puede añadir el periodo. Se mostrará un error avisando de la situación.

4.2.6. Caso de Uso 6

Tabla 4.17: Análisis del Caso de Uso 6

Modificar periodo	
Precondiciones	El usuario debe haber iniciado sesión. Debe existir al menos un periodo.
Postcondiciones	Los cambios realizados al periodo se guardarán en la base de datos.
Actores	Usuario administrador
Descripción	El usuario accede al periodo deseado y selecciona la opción de editar. Se mostrará el formulario correspondiente. Se realizan los cambios en el formulario. Pulsa el botón de guardar.
Escenarios Secundarios	<ul style="list-style-type: none"> - Se pulsa el botón cancelar. El formulario se restablece. - Se intenta acceder a otra página de la aplicación sin haber guardado los cambios. Se avisa de la situación y se pide una confirmación para continuar. - No se puede modificar el periodo. Se mostrará un error avisando de la situación.

4.2.7. Caso de Uso 7

Tabla 4.18: Análisis del Caso de Uso 7

Eliminar periodo	
Precondiciones	El usuario debe haber iniciado sesión. Debe existir al menos un periodo.
Postcondiciones	El periodo eliminado y los componentes que pertenecen al mismo se borrarán de la base de datos y dejarán de mostrarse en la aplicación.
Actores	Usuario administrador
Descripción	El usuario accede al periodo deseado y selecciona la opción de eliminar. Se pide confirmación para eliminarlo. Se acepta esta confirmación.
Escenarios Secundarios	<ul style="list-style-type: none"> - No se acepta la confirmación para eliminarlo. El periodo y sus componentes permanecen en la base de datos. - No se puede eliminar el periodo. Se mostrará un error avisando de la situación.

4.2.8. Caso de Uso 8

Tabla 4.19: Análisis del Caso de Uso 8

Consultar componentes (administración)	
Precondiciones	El usuario debe haber iniciado sesión. Debe existir al menos un periodo.
Postcondiciones	-
Actores	Usuario administrador
Descripción	El usuario accederá a un periodo existente y visualizará los componentes pertenecientes a este. Podrá acceder a cada uno de ellos.
Escenarios Secundarios	Aún no existen componentes para el periodo que se consulta. Se mostrará una tabla vacía.

4.2.9. Caso de Uso 9

Tabla 4.20: Análisis del Caso de Uso 9

Añadir componente	
Precondiciones	El usuario debe haber iniciado sesión. Debe existir al menos un periodo.
Postcondiciones	El componente añadido se guardará en la base de datos y se asociará al periodo correspondiente.
Actores	Usuario administrador
Descripción	El usuario accede al formulario para añadir un componente. Rellena los campos necesarios. Pulsa el botón de guardar.
Escenarios Secundarios	<ul style="list-style-type: none"> - Se pulsa el botón cancelar. El formulario se restablece. - Se intenta acceder a otra página de la aplicación sin haber guardado los cambios. Se avisa de la situación y se pide una confirmación para continuar. - No se puede añadir el componente. Se mostrará un error avisando de la situación.

4.2.10. Caso de Uso 10

Tabla 4.21: Análisis del Caso de Uso 10

Modificar componente	
Precondiciones	El usuario debe haber iniciado sesión. Debe existir al menos un componente.
Postcondiciones	Los cambios realizados en el componente se guardarán en la base de datos.
Actores	Usuario administrador
Descripción	El usuario accede al componente deseado y selecciona la opción de editar. Se mostrará el formulario correspondiente. Se realizan los cambios en el formulario. Pulsa el botón de guardar.
Escenarios Secundarios	<ul style="list-style-type: none"> - Se pulsa el botón cancelar. El formulario se restablece. - Se intenta acceder a otra página de la aplicación sin haber guardado los cambios. Se avisa de la situación y se pide una confirmación para continuar. - No se puede modificar el componente. Se mostrará un error avisando de la situación.

4.2.11. Caso de Uso 11

Tabla 4.22: Análisis del Caso de Uso 11

Eliminar componente	
Precondiciones	El usuario debe haber iniciado sesión. Debe existir al menos un componente.
Postcondiciones	El componente eliminado se borrará de la base de datos y dejará de mostrarse en la aplicación.
Actores	Usuario administrador
Descripción	El usuario accede al componente deseado y selecciona la opción de eliminar. Se pide confirmación para eliminarlo. Se acepta esta confirmación.
Escenarios Secundarios	<ul style="list-style-type: none"> - No se acepta la confirmación para eliminarlo. El componente permanece en la base de datos. - No se puede eliminar el componente. Se mostrará un error avisando de la situación.

4.3. ASI 5: ANÁLISIS DE CLASES

4.3.1. Diagrama de Clases

4.3.1.1. Museo

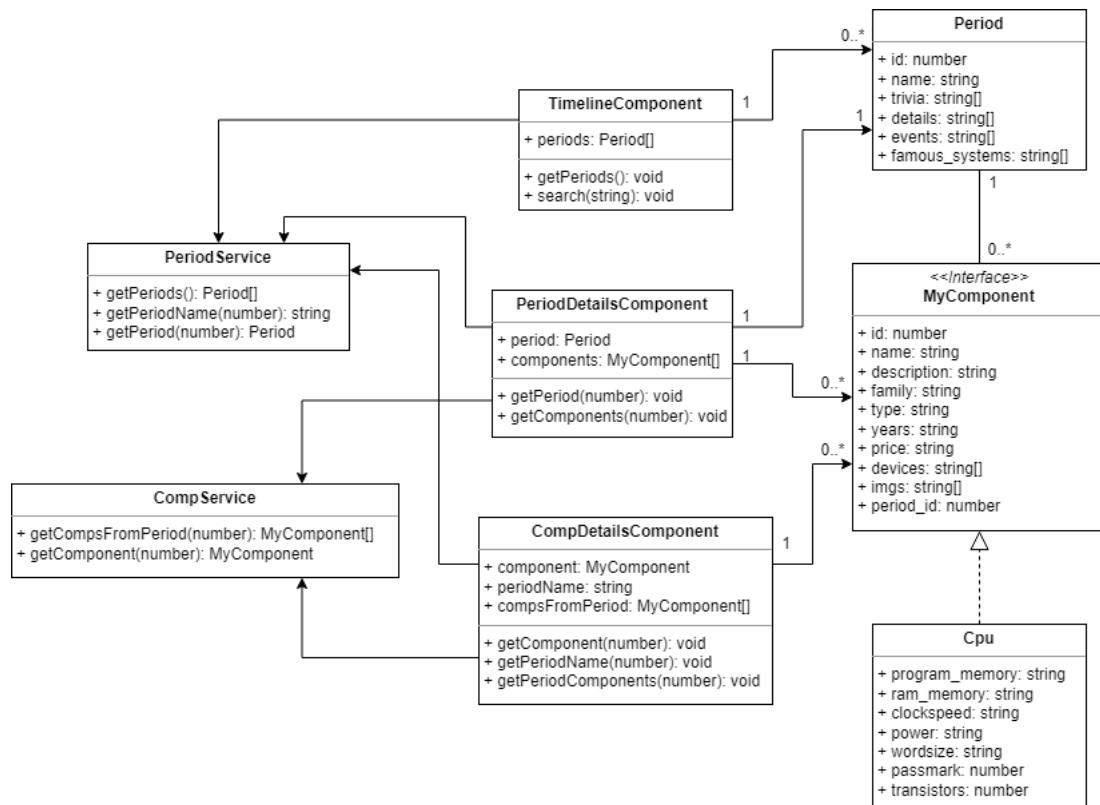


Figura 4.3: Análisis de clases: diagrama de clases del museo

4.3.1.2. Administración del museo

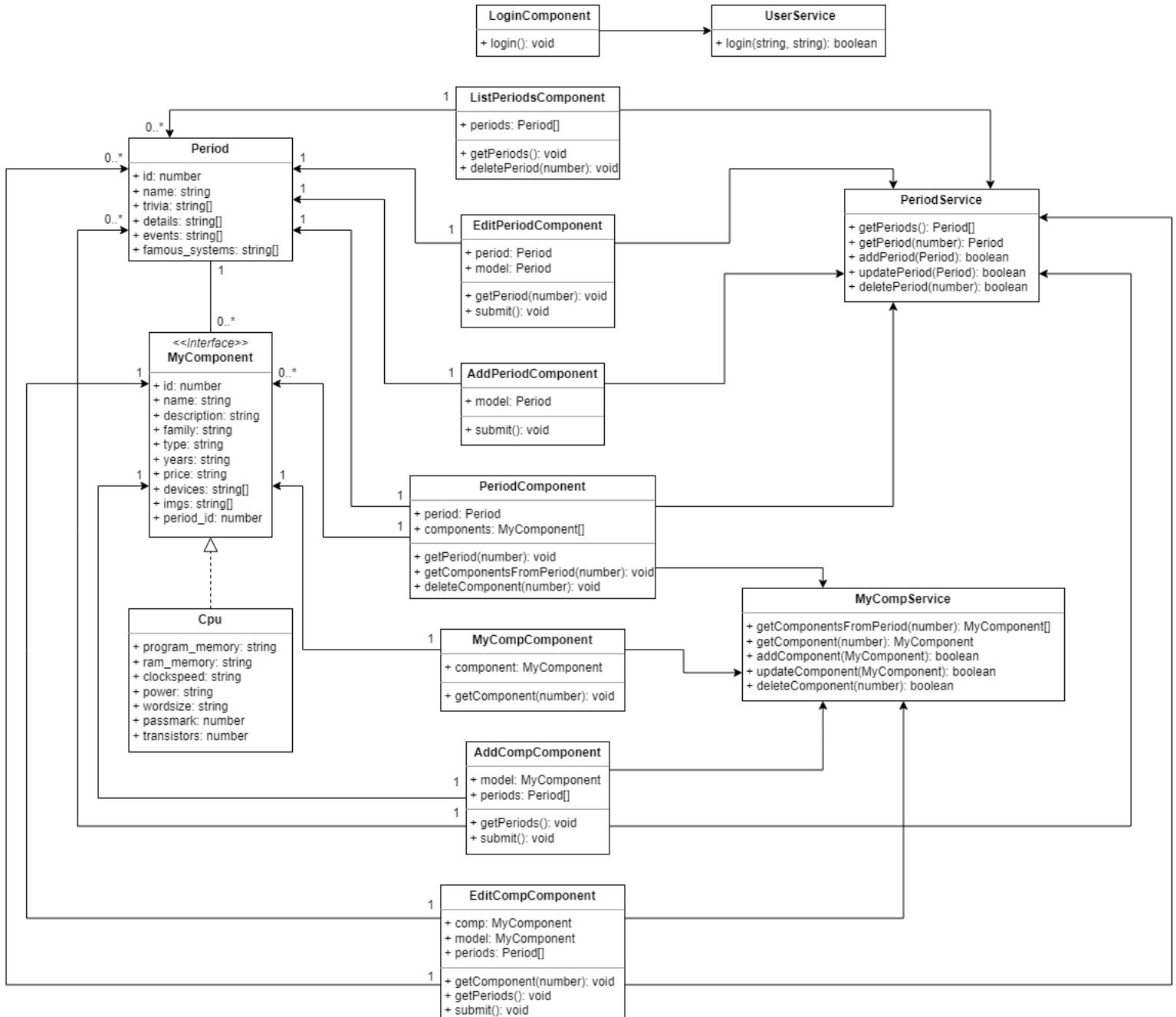


Figura 4.4: Análisis de clases: diagrama de clases de la administración

4.3.2. Descripción de las Clases

Period, *MyComponent* y *Cpu* son iguales en el proyecto del museo y en el de la administración, por lo tanto se describen una única vez a continuación:

Tabla 4.23: Descripción de la clase Period

Periodo		
Descripción		
Atributos propuestos		
id number Identificador del periodo		
name string Nombre del periodo		
trivia string[] Curiosidades del periodo		
details string[] Detalles del periodo		
events string[] Eventos ocurridos durante este periodo		
famous_systems string[] Sistemas famosos que llevaban componentes pertenecientes al periodo		
Métodos propuestos		
-		

Tabla 4.24: Descripción de la interfaz MyComponent

MyComponent		
Descripción		
Atributos propuestos		
id number Identificador del componente		
name string Nombre del componente		
description string Descripción del componente		
family string Familia a la que pertenece		
type string Tipo de componente (CPU, genérico...)		
years string Rango de años en los que se utilizó		
price string Precio de venta del componente		
devices string[] Tipo de dispositivos en los que se usaba el componente (portátiles o de escritorio)		
imgs string[] Nombres de las imágenes del componente		
period_id number Identificador del periodo al que pertenece el componente		
Métodos propuestos		
-		

Tabla 4.25: Descripción de la clase Cpu

Cpu		
Descripción		
Atributos propuestos		
program_memory	string	Memoria ROM de la CPU
ram_memory	string	Memoria RAM de la CPU
clockspeed	string	Velocidad de reloj de la CPU
power	string	Potencia de la CPU
wordsize	string	Tamaño de palabra de la CPU
passmark	number	Passmark de la CPU
transistors	number	Número de transistores de la CPU
Métodos propuestos		
-		

4.3.2.1. Museo

Tabla 4.26: Descripción de la clase PeriodService (museo)

PeriodService	
Descripción	
Atributos propuestos	
getPeriods	Devuelve todos los periodos existentes.
getPeriodName	Devuelve el nombre del periodo correspondiente al identificador pasado por parámetro.
getPeriod	Devuelve el periodo cuyo identificador se pasa como parámetro.
Métodos propuestos	

Tabla 4.27: Descripción de la clase CompService (museo)

CompService
Descripción
Atributos propuestos
Servicio que conecta con el back-end de la aplicación para realizar las operaciones relacionadas con los componentes.
-
Métodos propuestos
getCompsFromPeriod Devuelve los componentes pertenecientes al periodo cuyo identificador se pasa como parámetro.
getComponent Devuelve el componente cuyo identificador se pasa como parámetro.

Tabla 4.28: Descripción de la clase TimelineComponent

TimelineComponent
Descripción
Atributos propuestos
Clase asociada a la vista 4.4.1.1.
periods Period[] Listado de todos los periodos existentes.
Métodos propuestos
getPeriods Obtiene los periodos y los asigna a <i>periods</i> .
search Filtra los periodos según el texto introducido en la búsqueda.

Tabla 4.29: Descripción de la clase PeriodDetailsComponent

PeriodDetailsComponent		
Descripción		
Clase asociada a la vista 4.4.1.1.		
Atributos propuestos		
period	Period	Periodo del que se muestran los detalles.
components	MyComponent[]	Componentes pertenecientes al periodo.
Métodos propuestos		
getPeriod	Obtiene el periodo y lo asigna a <i>period</i> .	
getComponents	Obtiene los componentes del periodo y los asigna a <i>components</i> .	

Tabla 4.30: Descripción de la clase CompDetailsComponent

CompDetailsComponent		
Descripción		
Clase asociada a la vista 4.4.1.1.		
Atributos propuestos		
component	MyComponent	Componente del que se muestran los detalles.
periodName	string	Nombre del periodo al que pertenece el componente.
components	MyComponent[]	Otros componentes pertenecientes al periodo.
Métodos propuestos		
getComponent	Obtiene el componente y lo asigna a <i>component</i> .	
getPeriodName	Obtiene el nombre del periodo y lo asigna a <i>periodName</i> .	
getPeriodComponents	Obtiene los componentes del periodo y los asigna a <i>components</i> .	

4.3.2.2. Administración del museo

Tabla 4.31: Descripción de la clase UserService

UserService
Descripción
Atributos propuestos
-
Métodos propuestos
login Comprueba si el usuario y la contraseña introducidos se corresponden con los existentes en la base de datos.

Tabla 4.32: Descripción de la clase PeriodService (administración)

PeriodService
Descripción
Atributos propuestos
-
Métodos propuestos
getPeriods Devuelve todos los periodos existentes. getPeriod Devuelve el periodo cuyo identificador se pasa como parámetro. addPeriod Añade el periodo pasado como parámetro a la base de datos. updatePeriod Actualiza el periodo pasado como parámetro en la base de datos. getPeriod Elimina de la base de datos el periodo cuyo identificador se pasa como parámetro.

Tabla 4.33: Descripción de la clase CompService (administración)

CompService	
Descripción	
Servicio que conecta con el back-end de la aplicación para realizar las operaciones relacionadas con los componentes.	
Atributos propuestos	
-	
Métodos propuestos	
getCompsFromPeriod	Devuelve los componentes pertenecientes al periodo cuyo identificador se pasa como parámetro.
getComponent	Devuelve el componente cuyo identificador se pasa como parámetro.
addComponent	Añade el componente pasado como parámetro a la base de datos.
updateComponent	Actualiza el componente pasado como parámetro en la base de datos.
getComponent	Elimina de la base de datos el componente cuyo identificador se pasa como parámetro.

Tabla 4.34: Descripción de la clase LoginComponent

LoginComponent	
Descripción	
Clase asociada a la vista 4.4.1.2.	
Atributos propuestos	
-	
Métodos propuestos	
login	Comprueba los datos introducidos para iniciar sesión.

Tabla 4.35: Descripción de la clase ListPeriodsComponent

ListPeriodsComponent		
Descripción		
Clase asociada a la vista 4.4.1.2.		
Atributos propuestos		
periods	Period[]	Listado de todos los periodos existentes.
Métodos propuestos		
getPeriods	Obtiene los periodos y los asigna a <i>periods</i> .	
deletePeriod	Elimina el periodo seleccionado.	

Tabla 4.36: Descripción de la clase PeriodComponent

PeriodComponent		
Descripción		
Clase asociada a la vista 4.4.1.2.		
Atributos propuestos		
period	Period	Periodo del que se muestran los detalles.
components	MyComponent[]	Componentes pertenecientes al periodo.
Métodos propuestos		
getPeriod	Obtiene el periodo y lo asigna a <i>period</i> .	
getComponentsFromPeriod	Obtiene los componentes del periodo y los asigna a <i>components</i> .	
deleteComponent	Elimina el componente seleccionado.	

Tabla 4.37: Descripción de la clase AddPeriodComponent

AddPeriodComponent		
Descripción		
Clase asociada a la vista 4.4.1.2.		
Atributos propuestos		
model	Period	Periodo asociado al formulario en el que se introducen los datos.
Métodos propuestos		
submit	Añade el periodo con los datos introducidos en el formulario.	

Tabla 4.38: Descripción de la clase EditPeriodComponent

EditPeriodComponent		
Descripción		
Clase asociada a la vista 4.4.1.2.		
Atributos propuestos		
period	Period	Periodo que se va a editar, con los datos iniciales.
model	Period	Periodo asociado al formulario en el que se editan los datos.
Métodos propuestos		
getPeriod	Obtiene el periodo y lo asigna a <i>period</i> .	
submit	Actualiza el periodo con los datos introducidos en el formulario.	

Tabla 4.39: Descripción de la clase MyCompComponent

MyCompComponent		
Descripción		
Clase asociada a la vista 4.4.1.2.		
Atributos propuestos		
component	MyComponent	Componente del que se muestran los detalles.
Métodos propuestos		
getComponent	Obtiene el componente y lo asigna a <i>component</i> .	

Tabla 4.40: Descripción de la clase AddCompComponent

AddCompComponent		
Descripción		
Clase asociada a la vista 4.4.1.2.		
Atributos propuestos		
model	MyComponent	Componente asociado al formulario en el que se introducen los datos.
periods	Period[]	Listado de periodos existentes.
Métodos propuestos		
getPeriods	Obtiene los periodos y los asigna a <i>periods</i> .	
submit	Añade el componente con los datos introducidos en el formulario.	

Tabla 4.41: Descripción de la clase EditCompComponent

EditCompComponent		
Descripción		
Clase asociada a la vista 4.4.1.2.		
Atributos propuestos		
comp	MyComponent	Componente que se va a editar, con los datos iniciales.
model	MyComponent	Componente asociado al formulario en el que se editan los datos.
periods	Period[]	Listado de periodos existentes.
Métodos propuestos		
getComponent	Obtiene el componente y lo asigna a <i>comp</i> .	
getPeriods	Obtiene los periodos y los asigna a <i>periods</i> .	
submit	Actualiza el componente con los datos introducidos en el formulario.	

4.4. ASI 8: DEFINICIÓN DE INTERFACES DE USUARIO

4.4.1. Definición del aspecto de la interfaz

4.4.1.1. Museo

A continuación se presentan los prototipos de interfaces diseñados para la página web del museo. Todas ellas tienen en común la barra de navegación, que contiene el logo de la EII, un enlace a la vista general del museo y un selector de idioma.

Inicio

En la página de inicio del museo se muestra un mensaje de bienvenida y un botón que conduce a la vista general del museo.

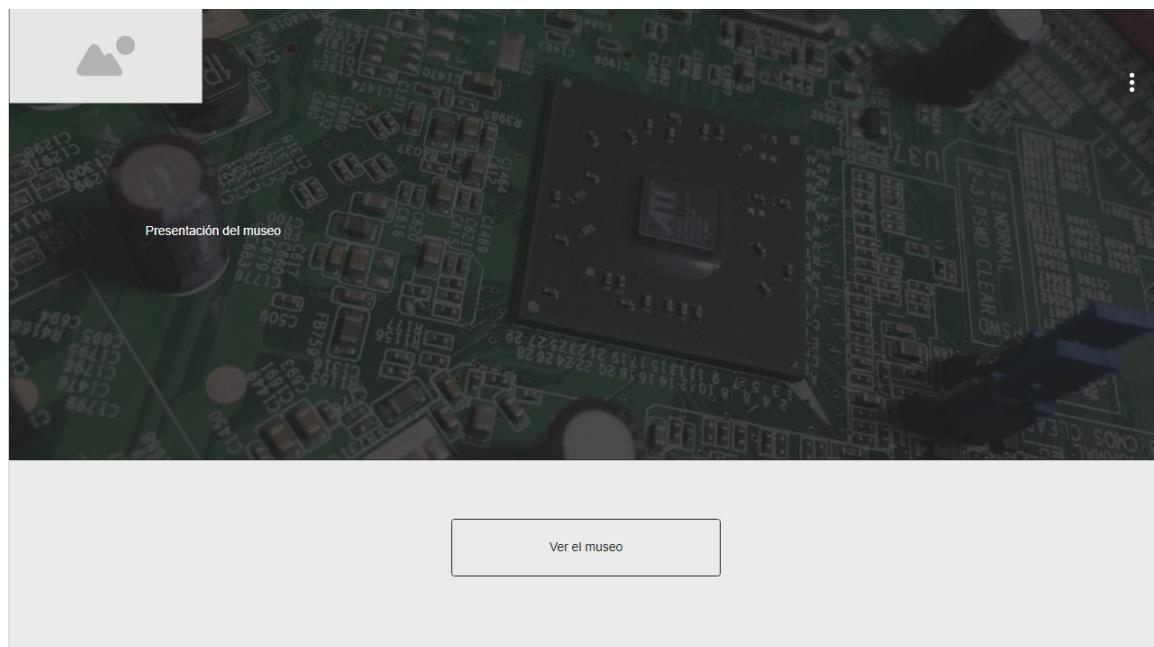


Figura 4.5: Prototipo: Página de inicio

Vista general del museo

En la vista general del museo encontramos un menú lateral con filtros de búsqueda, y una sección principal que contiene una línea temporal con los períodos en los que se divide la historia de las CPUs.

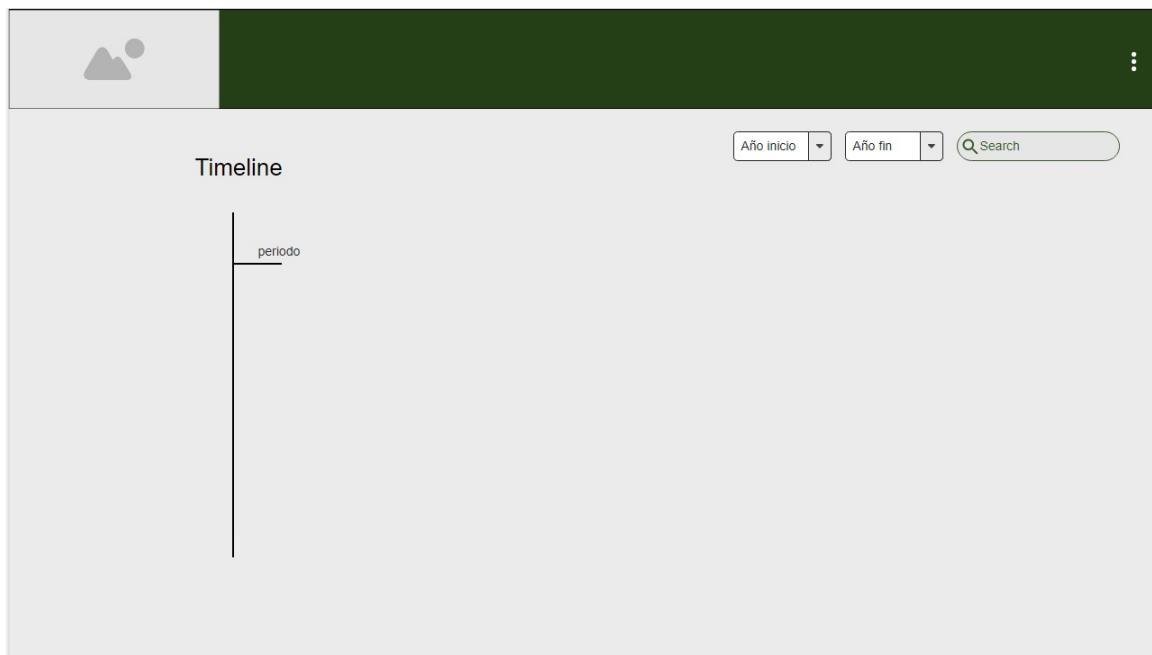


Figura 4.6: Prototipo: Página de la vista general del museo

Detalles del periodo

En esta página hay un menú para volver a la vista general, y se muestran todos los detalles de un periodo (nombre, características, sistemas famosos de dicho periodo, etc.) y los componentes que pertenecen al mismo.



Figura 4.7: Prototipo: Página de detalles del periodo (museo)

Detalles del componente

En esta página se muestra una galería de fotos del componente, la descripción del mismo, y un listado de características. En el menú de esta página hay un listado de componentes pertenecientes al mismo periodo.

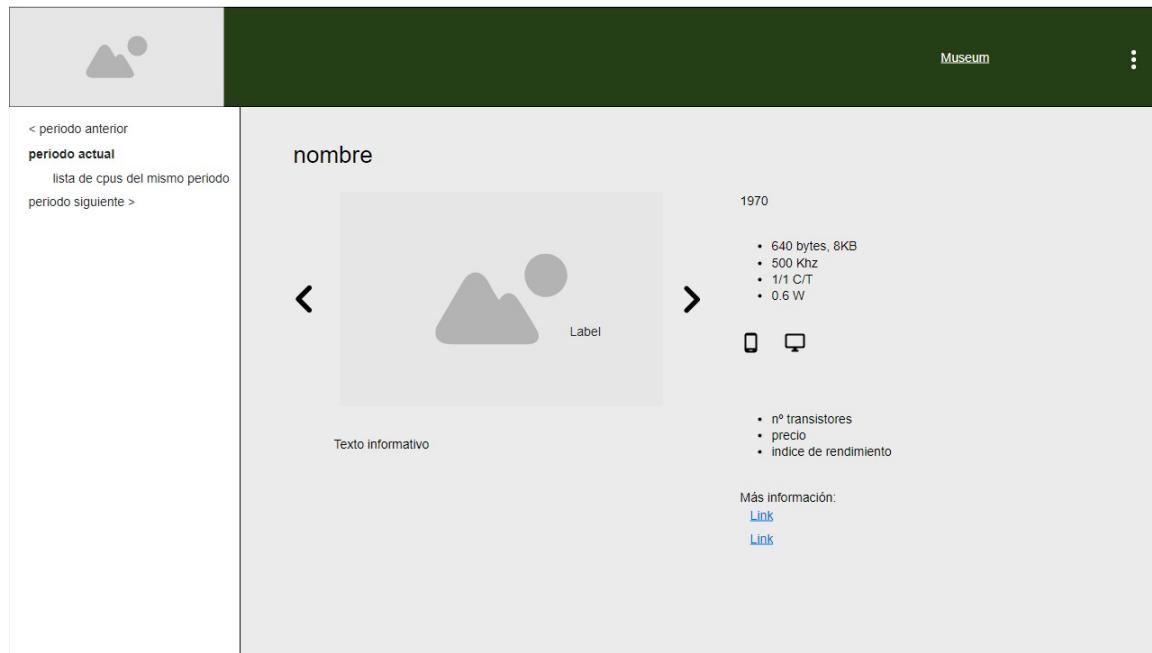


Figura 4.8: Prototipo: Página de detalles del componente (museo)

4.4.1.2. Administración del museo

A continuación, se muestran los prototipos inciales para la aplicación de administración del museo. En todas ellas, salvo en la de inicio de sesión, hay un menú lateral de navegación.

Iniciar sesión

En esta página el administrador del sistema deberá introducir su usuario y contraseña para acceder al mismo.



Figura 4.9: Prototipo: Página de inicio de sesión

Listado de periodos

En esta página se muestra un listado de los períodos existentes, con las opciones de acceder a cada uno, editarlo o eliminarlo.

Periodo 1	Editar	Eliminar
Periodo 2	Editar	Eliminar
Periodo 3	Editar	Eliminar
Periodo 4	Editar	Eliminar

Figura 4.10: Prototipo: Página de listado de períodos

Periodo

Esta página contiene los detalles de un período así como un listado de los componentes pertenecientes al mismo, ofreciendo la opción de acceder a ellos, editarlos o eliminarlos.

Componente 1	Editar	Eliminar
Componente 2	Editar	Eliminar
Componente 3	Editar	Eliminar
Componente 4	Editar	Eliminar

Figura 4.11: Prototipo: Página de detalles de un período (administración)

Componente

En esta página se muestran los detalles correspondientes al componente.

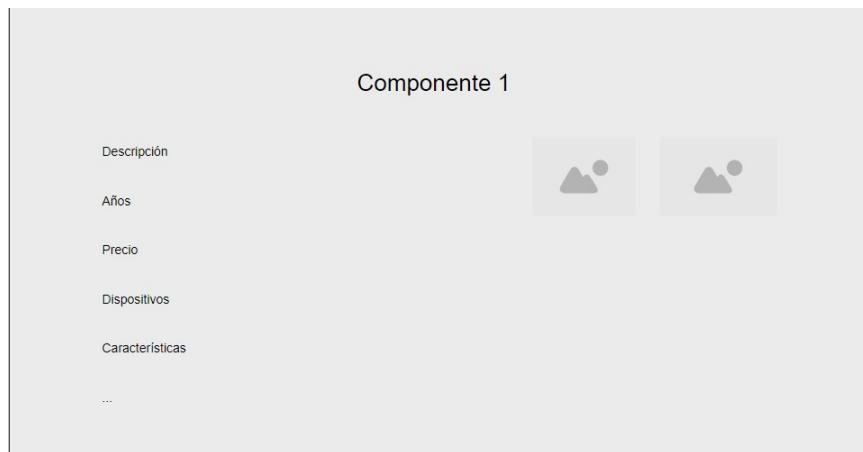


Figura 4.12: Prototipo: Página de detalles de un componente (admnimnistración)

Añadir/editar periodo

Los formularios para añadir o editar un periodo son idénticos, con la única diferencia de que el formulario para editar ya tiene los campos completados con los valores existentes del periodo, por tanto solo se muestra una captura representando ambos.

Este prototipo muestra un formulario para 'Añadir periodo'. Los campos incluyen:

- Nombre: Un cuadro de texto con placeholder 'Placeholder'.
- Detalles: Un cuadro de texto con placeholder 'Type here'.
- Eventos: Un cuadro de texto con placeholder 'Type here'.
- Sabías qué: Un cuadro de texto con placeholder 'Type here'.

Figura 4.13: Prototipo: Formulario para añadir o editar un periodo

Añadir/editar componente

Con los formularios para añadir o editar un componente ocurre igual que con los del periodo ya mencionados.

El prototipo muestra un formulario titulado "Añadir componente". A la izquierda, una barra lateral incluye enlaces: "Añadir periodo", "Añadir componente", "Editar periodo" y "Editar componente". El formulario principal contiene los siguientes campos:

- Periodo:** Un select desplegable con el valor "Select".
- Nombre:** Un campo de texto con placeholder "Placeholder".
- Descripción:** Un cuadro de texto con placeholder "Type here".
- Propiedad:** Sección que incluye un campo "Valor" y un select "Uds.".
- Años:** Un campo de tipo spinner.
- Precio:** Un campo de tipo spinner.
- Dispositivos:** Una sección con checkboxes: "Escritorio" (check) y "Portátiles" (check).
- Propiedad:** Sección que incluye un campo "Valor" y un select "Uds.".
- Propiedad:** Sección que incluye un campo "Valor" y un select "Uds.".
- Propiedad:** Sección que incluye un campo "Valor" y un select "Uds.".

En la parte inferior, hay botones "Cancelar" y "Guardar".

Figura 4.14: Prototipo: Formulario para añadir o editar un componente

4.4.2. Diagrama de Navegabilidad

A continuación se presentan dos diagramas de navegabilidad, correspondientes a las dos aplicaciones web que constituyen el sistema.

4.4.2.1. Museo

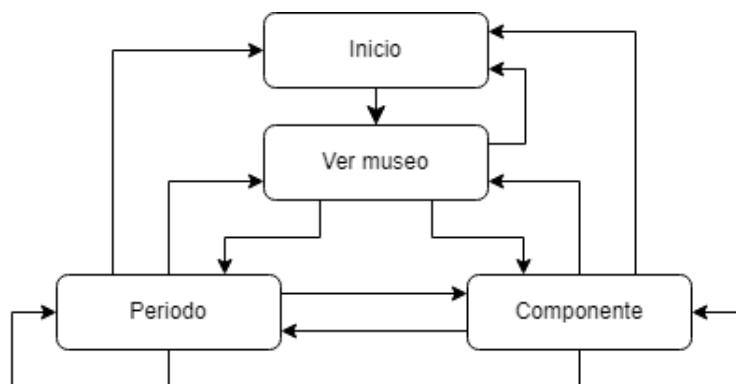


Figura 4.15: Diagrama de navegabilidad del museo

4.4.2.2. Administración del museo

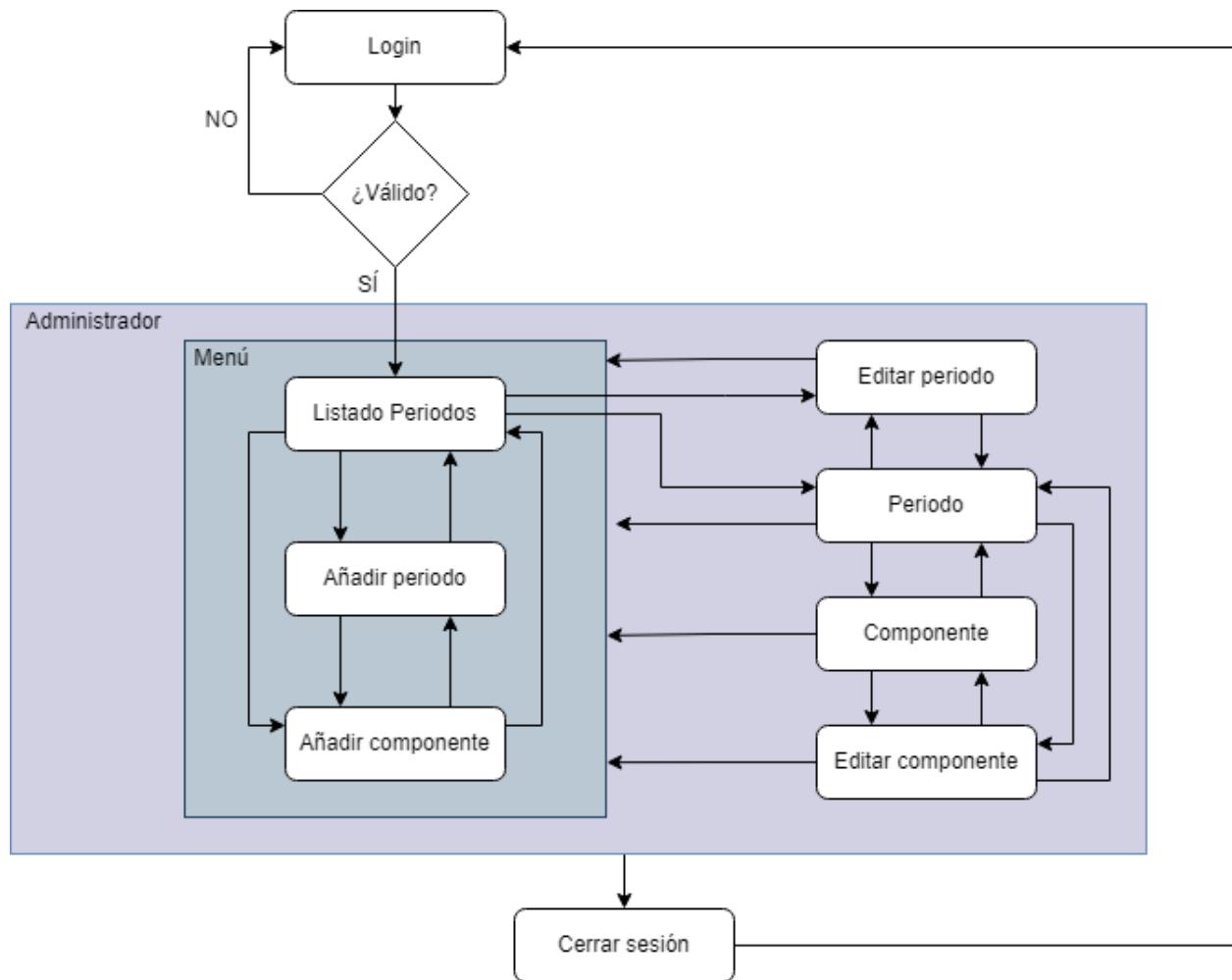


Figura 4.16: Diagrama de navegabilidad de la administración del museo

4.5. ASI 10: ESPECIFICACIÓN DEL PLAN DE PRUEBAS

4.5.1. Pruebas unitarias

Se realizarán pruebas unitarias del sistema utilizando Jasmine y Karma, herramientas incluidas en Angular para probar el correcto funcionamiento de los diferentes componentes.

Tabla 4.42: Pruebas unitarias: Caso de uso 1

Caso de uso 1: Consultar periodos (museo)	
Prueba	Resultado esperado
Obtener periodos	El sistema devolverá una lista de todos los periodos existentes.
existentes	
Obtener periodo	El sistema devolverá una lista de los periodos cuyo nombre
por nombre	contenga el texto introducido.
Obtener periodo	El sistema devolverá una lista de los periodos cuyos años coin-
por años	ciden con los introducidos.

Tabla 4.43: Pruebas unitarias: Caso de uso 2

Caso de uso 2: Consultar componentes (museo)	
Prueba	Resultado esperado
Obtener componentes	El sistema devolverá una lista de todos los componentes
de un periodo	pertenecientes al periodo.

Tabla 4.44: Pruebas unitarias: Caso de uso 3

Caso de uso 3: Iniciar sesión	
Prueba	Resultado esperado
Iniciar sesión con	El sistema permitirá el acceso a la página de administración.
datos válidos	
Iniciar sesión con	El sistema no permitirá el acceso y se mostrará un error.
datos incorrectos	

Tabla 4.45: Pruebas unitarias: Caso de uso 4

Caso de uso 4: Consultar periodos (administración)	
Prueba	Resultado esperado
Obtener periodos	El sistema devolverá una lista de todos los periodos existentes.
existentes	

Tabla 4.46: Pruebas unitarias: Caso de uso 5

Caso de uso 5: Añadir periodo	
Prueba	Resultado esperado
Añadir nuevo periodo	El sistema tendrá un periodo más.
Añadir periodo que ya existe	El sistema no añadirá el periodo y responderá con un error.
Añadir periodo con campos vacíos	El sistema no añadirá el periodo y responderá con un error.

Tabla 4.47: Pruebas unitarias: Caso de uso 6

Caso de uso 6: Modificar periodo	
Prueba	Resultado esperado
Modificar periodo existente	El sistema actualizará los datos del periodo.
Modificar un periodo que no existe	El sistema responderá con un error.
Modificar periodo dejando campos vacíos	El sistema no actualizará el periodo y responderá con un error.

Tabla 4.48: Pruebas unitarias: Caso de uso 7

Caso de uso 7: Eliminar periodo	
Prueba	Resultado esperado
Eliminar un periodo existente	El sistema tendrá un periodo menos.
Eliminar un periodo que no existe	El sistema responderá con un error.

Tabla 4.49: Pruebas unitarias: Caso de uso 8

Caso de uso 8: Consultar componentes (administración)	
Prueba	Resultado esperado
Obtener componentes de un periodo	El sistema devolverá una lista de todos los componentes pertenecientes al periodo.

Tabla 4.50: Pruebas unitarias: Caso de uso 9

Caso de uso 9: Añadir componente	
Prueba	Resultado esperado
Añadir nuevo componente	El sistema tendrá un componente más.
Añadir componente que ya existe	El sistema no añadirá el componente y responderá con un error.
Añadir componente a un periodo que no existe	El sistema no añadirá el componente y responderá con un error.
Añadir componente con campos obligatorios vacíos	El sistema no añadirá el componente y responderá con un error.

Tabla 4.51: Pruebas unitarias: Caso de uso 10

Caso de uso 10: Modificar componente	
Prueba	Resultado esperado
Modificar componente existente	El sistema actualizará los datos del componente.
Modificar un componente que no existe	El sistema responderá con un error.
Modificar componente dejando campos obligatorios vacíos	El sistema no actualizará el componente y responderá con un error.

Tabla 4.52: Pruebas unitarias: Caso de uso 11

Caso de uso 11: Eliminar componente	
Prueba	Resultado esperado
Eliminar un componente existente	El sistema tendrá un componente menos.
Eliminar un componente que no existe	El sistema responderá con un error.

4.5.2. Pruebas del sistema

Se probará el conjunto completo del sistema, es decir, si la aplicación, el servidor y la base de datos se conectan de forma correcta, sin errores en tiempo de ejecución y obteniendo las respuestas esperadas. Estas pruebas se realizarán de forma manual sobre la aplicación.

4.5.3. Pruebas de usabilidad

Para realizar estas pruebas se hará uso de una serie de cuestionarios, cuyos resultados se obtendrán tras observar como usuarios con diferentes perfiles interactúan con el sistema.

Capítulo 5

DISEÑO DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

DSI

5.1. DSI 4: DISEÑO DE CLASES

5.1.1. Diagrama de Clases

5.1.1.1. Museo

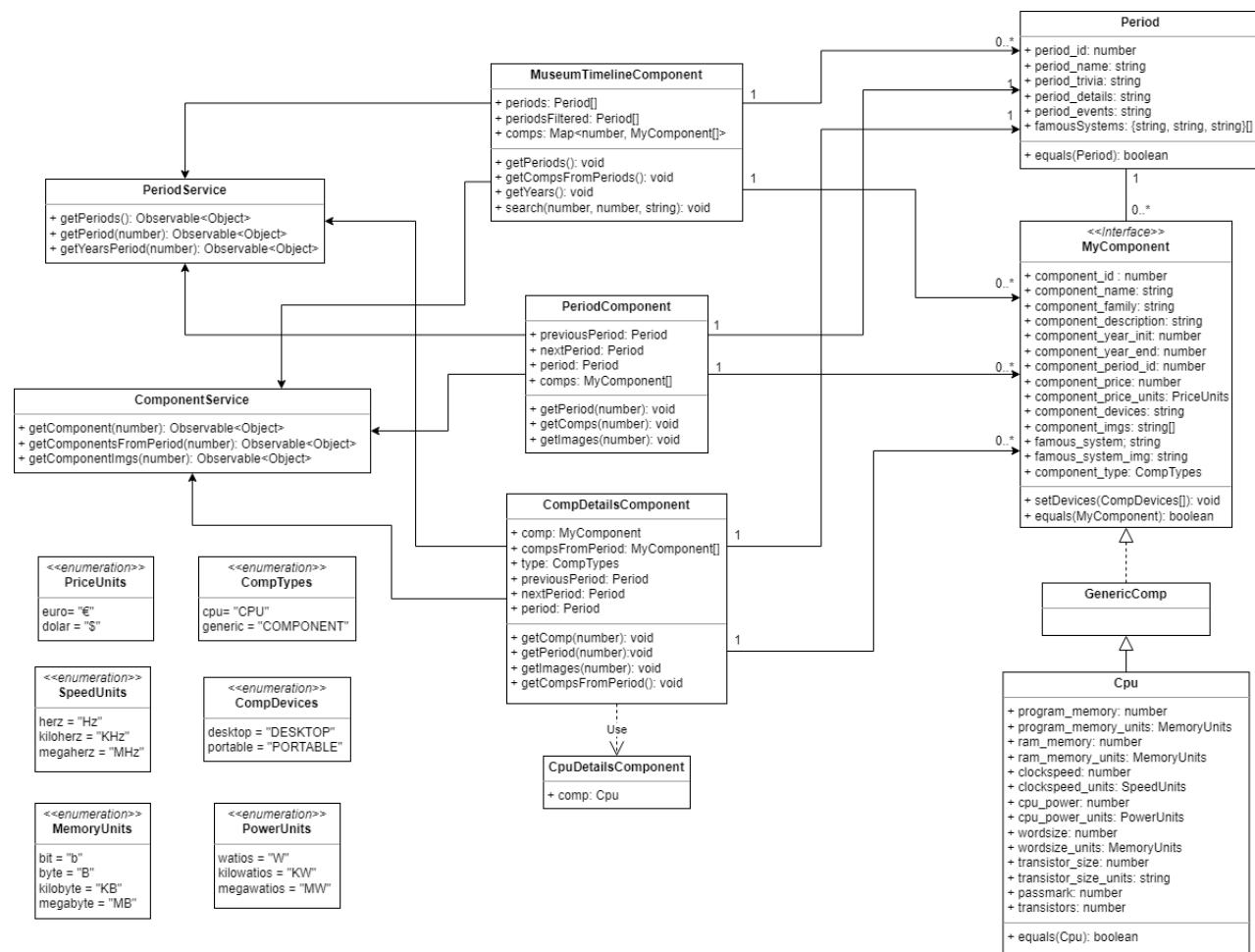


Figura 5.1: Diseño de clases: diagrama de clases del museo

5.1.1.2. Administración del museo

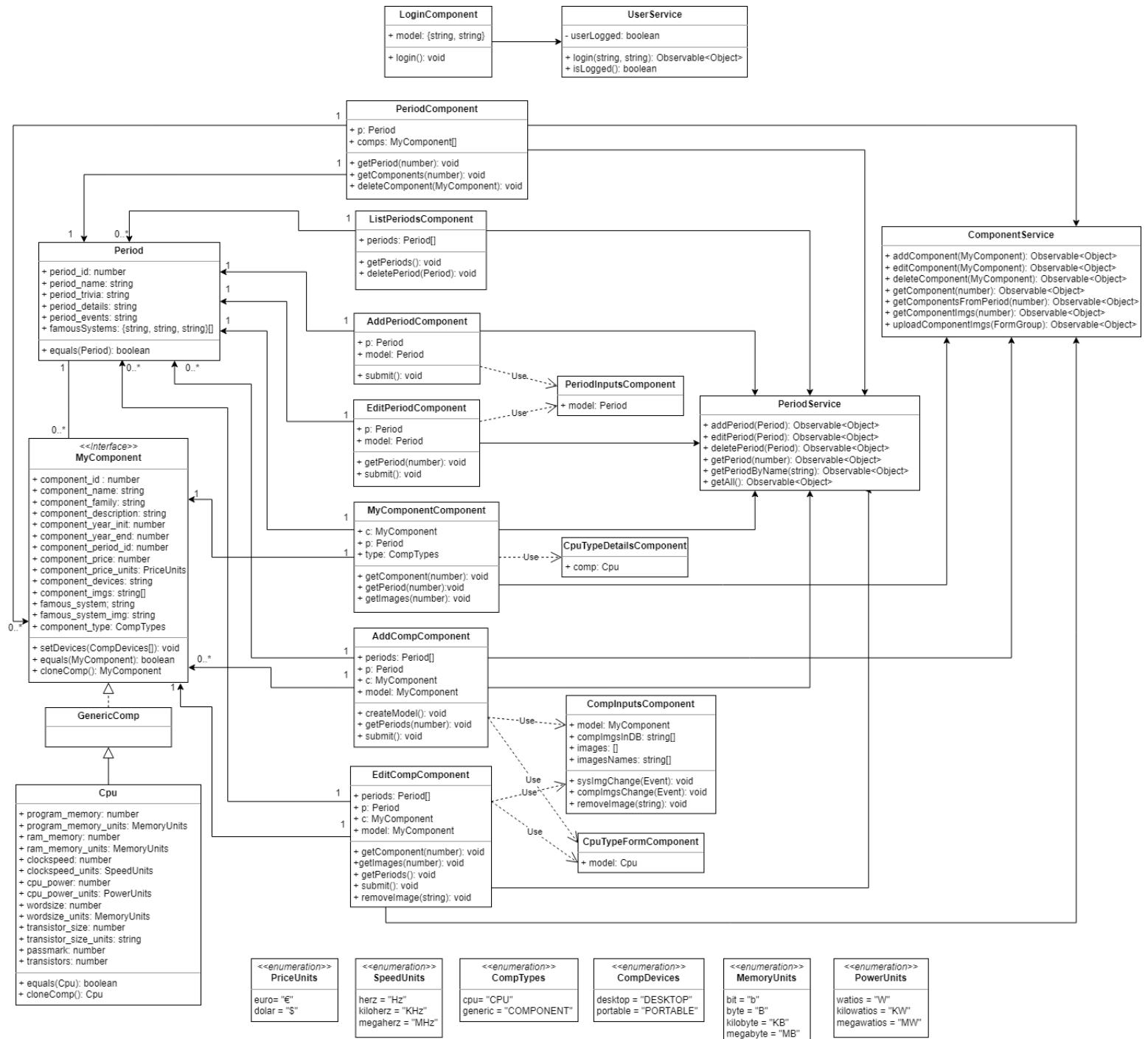


Figura 5.2: Diseño de clases: diagrama de clases de la administración del museo

5.2. DSI 5: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA

- 5.2.1. Diagrama de paquetes
- 5.2.2. Diagrama de componentes
- 5.2.3. Diagrama de despliegue

5.3. DSI 6: DISEÑO FÍSICO DE DATOS

5.3.1. Descripción del SGBD Usado

Se ha creado una base de datos relacional, utilizando MySQL 8 como sistema gestor de bases de datos, debido a su gran popularidad en todo el mundo y en el desarrollo de aplicaciones con Angular y PHP.

La base de datos creada es museo-eii, y se compone de cinco tablas:

- periods: almacena toda la información relativa a un periodo.
- components: contiene los datos que serían comunes a cualquier tipo de componente independientemente de su tipo, como nombre, año de salida, precio, ect.
- components_images: asigna el conjunto de imágenes de cada componente.
- cpus: almacena los datos específicos de una CPU, como la memoria RAM, la velocidad de reloj o el tamaño de palabra. El identificador de esta tabla referencia al elemento correspondiente de la tabla components, simulando así una herencia en la base de datos. Esta herencia simulada hace que la base de datos sea fácilmente ampliable si se añade un tipo de componente que no sea una CPU, ya que bastaría con añadir una nueva tabla con los campos necesarios que también refiriese a components.
- administrator: almacena los datos de inicio de sesión del administrador (email y contraseña cifrada). En caso de que hubiera más usuarios que tuviesen que iniciar sesión se habría creado una base de datos exclusiva para almacenarlos, pero como en este caso solo existe un administrador, he decidido incluir esta tabla en la base de datos existente para este proyecto.

5.3.2. Integración del SGBD en Nuestro Sistema

Para integrar el sistema con la base de datos se han creado tres servicios en angular: uno para periodos, otro para componentes y el último para el usuario. Estos tres servicios utilizan la librería HttpClient de Angular, que permite realizar peticiones HTTP para obtener o enviar datos al lado del servidor, donde se encuentran los archivos PHP que contienen las consultas que han de realizarse sobre la base de datos.

5.3.3. Diagrama E-R

A continuación se muestra el diagrama entidad-relación de la base de datos del sistema, museo-eii:

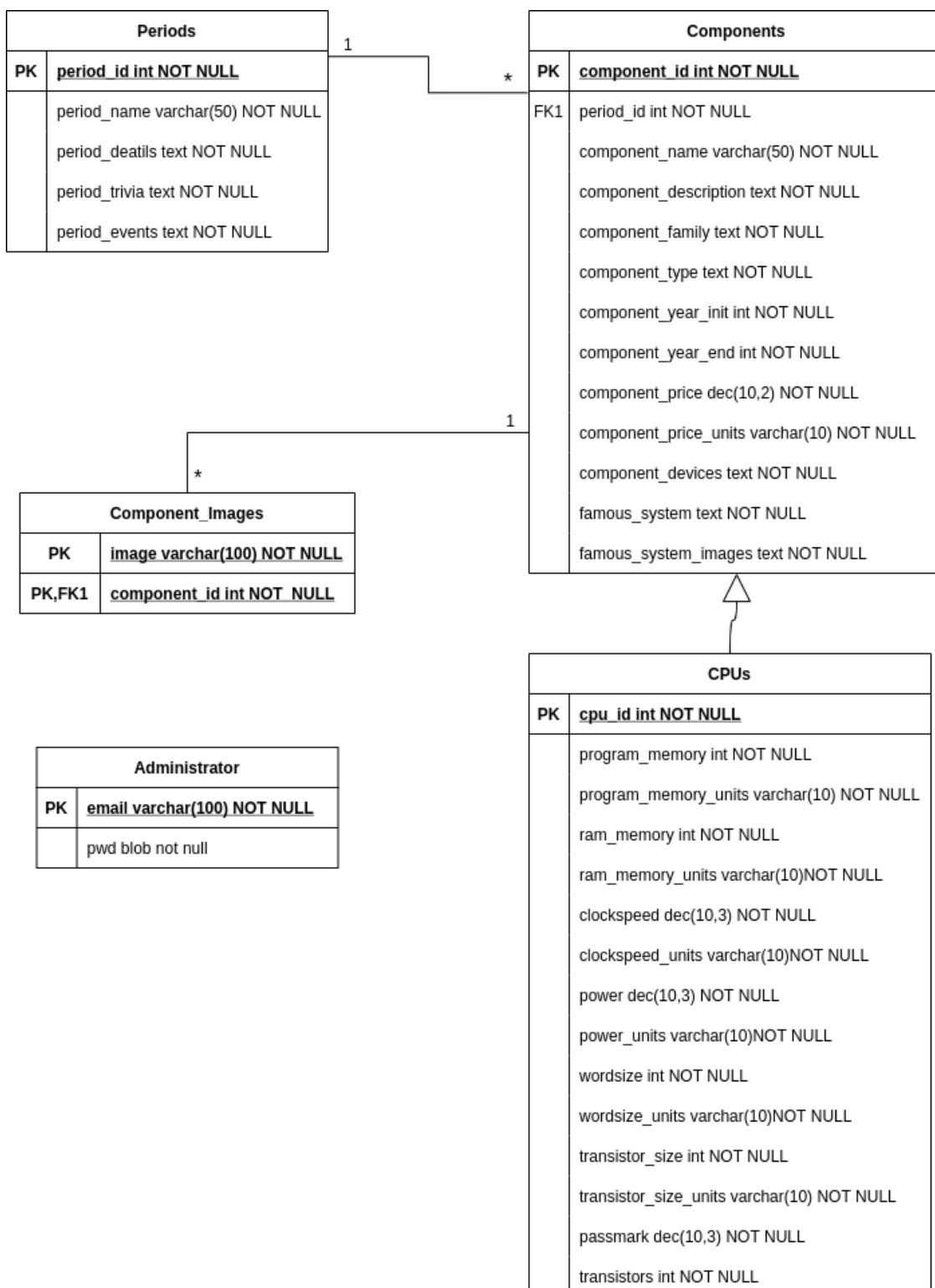


Figura 5.3: Diagrama Entidad-Relación de la base de datos creada

5.4. DSI 10: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS

5.4.1. Pruebas Unitarias y del Sistema

Tabla 5.1: Diseño de pruebas: Consultar períodos (museo)

Consultar períodos (museo)		
Prueba	Pasos	Resultado esperado
Obtener períodos existentes	- Acceder a la vista general del museo	El sistema devolverá una lista de todos los períodos existentes.
Obtener periodo por nombre	- Acceder a la vista general del museo. - Introducir un texto en la barra de búsqueda.	El sistema devolverá una lista de los períodos cuyo nombre contenga el texto introducido.
Obtener periodo por años	- Acceder a la vista general del museo. - Cambiar los años especificados en la barra deslizadora.	El sistema devolverá una lista de los períodos cuyos años coincidan con los introducidos.

Tabla 5.2: Diseño de pruebas: Consultar componentes (museo)

Consultar componentes (museo)		
Prueba	Pasos	Resultado esperado
Obtener componentes de un periodo	- Acceder a un periodo.	El sistema devolverá una lista de todos los componentes pertenecientes al periodo.

Tabla 5.3: Diseño de pruebas: Iniciar sesión

Iniciar sesión		
Prueba	Pasos	Resultado esperado
Iniciar sesión con datos válidos	- Introducir el usuario <i>uo257829@uniovi.es</i> . - Introducir la contraseña <i>museoinfo2022</i> .	El sistema permitirá el acceso a la página de administración.
Iniciar sesión con datos incorrectos	- Introducir el usuario <i>uo257829@uniovi.es</i> . - Introducir la contraseña <i>123456</i> .	El sistema no permitirá el acceso y se mostrará un error.

Tabla 5.4: Diseño de pruebas: Consultar periodos (administración)

Consultar periodos (administración)		
Prueba	Pasos	Resultado esperado
Obtener periodos existentes	- Acceder al listado de periodos.	El sistema devolverá una lista de todos los periodos existentes.

Tabla 5.5: Diseño de pruebas: Añadir periodo

Añadir periodo		
Prueba	Pasos	Resultado esperado
Añadir nuevo periodo (periodo 2)	- Acceder a añadir un periodo. - Introducir los datos <i>Periodo 2, detalles, eventos, curiosidades.</i> - Guardar periodo.	El sistema tendrá un periodo más.
Añadir periodo que ya existe (periodo 1)	- Acceder a añadir un periodo. - Introducir los datos <i>Periodo 1, detalles, eventos, curiosidades.</i> - Guardar periodo.	El sistema no añadirá el periodo y responderá con un error.
Añadir periodo con campos vacíos	- Acceder a añadir un periodo. - Introducir los datos <i>Periodo 2, ' , eventos, curiosidades.</i> - Guardar periodo.	El sistema no añadirá el periodo y responderá con un error.

Tabla 5.6: Diseño de pruebas: Modificar periodo

Modificar periodo		
Prueba	Entrada	Resultado esperado
Modificar periodo existente (periodo 1)	- Introducir los datos <i>Periodo 1 modificado, detalles, eventos, curiosidades, 1.</i> - Guardar periodo.	El sistema actualizará los datos del periodo.
Modificar un periodo que no existe (periodo 25)	- Introducir los datos <i>Periodo 25 modificado, detalles, eventos, curiosidades, 25.</i> - Guardar periodo.	El sistema responderá con un error.
Modificar periodo dejando campos vacíos	- Introducir los datos <i>Periodo 1 modificado, detalles, eventos, ' , 1.</i> - Guardar periodo.	El sistema no actualizará el periodo y responderá con un error.

Tabla 5.7: Diseño de pruebas: Eliminar periodo

Eliminar periodo		
Prueba	Entrada	Resultado esperado
Eliminar un periodo existente (periodo 1)	- Eliminar periodo 1.	El sistema tendrá un periodo menos.
Eliminar un periodo que no existe (periodo 25)	- Eliminar periodo 25.	El sistema responderá con un error.

Tabla 5.8: Diseño de pruebas: Consultar componentes (administración)

Consultar componentes (administración)		
Prueba	Entrada	Resultado esperado
Obtener componentes de un periodo	- Acceder a un periodo.	El sistema devolverá una lista de todos los componentes pertenecientes al periodo.

Tabla 5.9: Diseño de pruebas: Añadir componente

Añadir componente		
Prueba	Entrada	Resultado esperado
Añadir nuevo componente (CPU 2)	- Acceder a añadir componente. - Introducir los datos <i>CPU 2, familia, descripción, años, periodo 1, etc.</i> - Guardar componente.	El sistema tendrá un componente más.
Añadir componente que ya existe (CPU 1)	- Acceder a añadir componente. - Introducir los datos <i>CPU 1, familia, descripción, años, periodo 1, etc.</i> - Guardar componente.	El sistema no añadirá el componente y responderá con un error.
Añadir componente a un periodo que no existe (CPU 3, periodo 25)	- Acceder a añadir componente. - Introducir los datos <i>CPU 3, familia, descripción, años, periodo 25, etc.</i> - Guardar componente.	El sistema no añadirá el componente y responderá con un error.
Añadir componente con campos obligatorios vacíos	- Acceder a añadir componente. - Introducir los datos <i>CPU 3, familia, , años, periodo 1, etc.</i> - Guardar componente.	El sistema no añadirá el componente y responderá con un error.

Tabla 5.10: Diseño de pruebas: Modificar componente

Modificar componente		
Prueba	Entrada	Resultado esperado
Modificar componente existente (CPU 1)	<ul style="list-style-type: none"> - Introducir los datos <i>CPU 1 modificada, familia, descripción, años, periodo 1, 1, etc.</i> - Guardar componente. 	El sistema actualizará los datos del componente.
Modificar un componente que no existe (CPU 30)	<ul style="list-style-type: none"> - Introducir los datos <i>CPU 30 modificada, familia, descripción, años, periodo 1, 30, etc.</i> - Guardar componente. 	El sistema responderá con un error.
Modificar componente dejando campos obligatorios vacíos	<ul style="list-style-type: none"> - Introducir los datos <i>CPU 1 modificada, familia, '', años, periodo 1, 1, etc.</i> - Guardar componente. 	El sistema no actualizará el componente y responderá con un error.

Tabla 5.11: Diseño de pruebas: Eliminar componente

Eliminar componente		
Prueba	Entrada	Resultado esperado
Eliminar un componente existente (CPU 1)	- Eliminar CPU 1.	El sistema tendrá un componente menos.
Eliminar un componente que no existe (CPU 30)	- Eliminar CPU 30.	El sistema responderá con un error.

5.4.2. Pruebas de Usabilidad y Accesibilidad

Las pruebas de usabilidad y accesibilidad de la aplicación web del museo serán realizadas por usuarios con distintos perfiles, mientras que las de la aplicación de administración solo las realizará el usuario administrador, ya que será el único que interactúe con este sistema. Los usuarios interactuarán con el sistema y llenarán un cuestionario, que será detallado a continuación. También habrá un cuestionario en el que el responsable de las pruebas anotará las observaciones realizadas durante las pruebas.

5.4.2.1. Cuestionario de evaluación

Preguntas de carácter general

Tabla 5.12: Pruebas de usabilidad: preguntas de carácter general

¿Usa un ordenador frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el ordenador?

1. Es parte de mi trabajo o profesión
2. Lo uso básicamente para ocio
3. Solo empleo aplicaciones estilo Office
4. Únicamente leo el correo y navego ocasionalmente

¿Qué busca Vd. principalmente en una aplicación web?

1. Que sea fácil de navegar
2. Que sea intuitiva
3. Que sea rápida

Actividades guiadas

Las actividades que realizarán los diferentes usuarios en la aplicación web del museo son las siguientes:

- Navegar por la linea temporal presente en la vista general del museo.
- Realizar una búsqueda por años.
- Realizar una búsqueda por nombre.
- Consultar los detalles de un periodo.
- Consultar los detalles de un componente.

El administrador realizará las actividades mencionadas y también las correspondientes a la aplicación de administración, que se muestran a continuación:

- Añadir un periodo.
- Añadir un componente.
- Editar un periodo.
- Editar un componente.
- Eliminar un periodo.
- Eliminar un componente.

Preguntas cortas sobre la aplicación y observaciones

Tabla 5.13: Pruebas de usabilidad: preguntas sobre la aplicación

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>				
<i>¿Necesita ayuda para utilizar la aplicación?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
<i>¿Identifica fácilmente la información que se le presenta?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. espera?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
<i>El tipo y el tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				
Diseño de la interfaz	Sí	A veces	No	
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Es coherente el diseño general del sitio web?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
Observaciones				

5.4.2.2. Cuestionario para el responsable de las pruebas

Tabla 5.14: Pruebas de usabilidad: cuestionario para el responsable de las pruebas

<i>Nombre de la actividad</i>
Tiempo empleado:
Problemas encontrados:
Observaciones:

Capítulo 6

CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

CSI

6.1. CSI 1: PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN

6.1.1. Estándares y normas seguidos

6.1.1.1. Angular Style Guide

La guía de estilos de Angular[7] es un conjunto de recomendaciones sobre la sintaxis, estructura y convenciones de código en proyectos de Angular.

6.1.1.2. HTML5

HTML5 es la versión más reciente y la actualmente usada de HTML, y está estandarizada por el W3C (World Wide Web Consortium).

6.1.1.3. CSS

Hojas de estilos estandarizadas por el W3C.

6.1.1.4. PHP Code Style Guide

La guía de estilos de PHP[8] contiene normas de código y buenas prácticas.

6.1.2. Lenguajes de programación

6.1.2.1. TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft. Extiende JavaScript añadiendo la definición de tipos estáticos.

6.1.2.2. HTML

HTML (HyperText Markup Language) es un lenguaje de marcado utilizado en la elaboración de páginas web.

6.1.2.3. CSS

CSS (Cascading Style Sheets) es un lenguaje de diseño gráfico que permite modificar la presentación de los elementos definidos en los documentos HTML.

6.1.2.4. PHP

PHP es un lenguaje de programación utilizado en el desarrollo web que es procesado en el lado del servidor.

6.1.2.5. SQL

SQL (Structured Query Language) es un lenguaje de consultas utilizado para leer, insertar, actualizar o eliminar datos de la base de datos relacional utilizada.

6.1.3. Herramientas y programas usados para el desarrollo

6.1.3.1. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft, gratuito y de código abierto. Tiene soporte integrado para TypeScript y Node.js, extensiones para otros lenguajes como PHP. También cuenta con soporte para depuración, control integrado de Git e *IntelliSense*, una función de autocompletado de código[9].



Figura 6.1: Logo de Visual Studio Code

6.1.3.2. XAMPP

XAMPP es una distribución de Apache gratuita que contiene MariaDB, PHP y Perl[10]. Fue usado inicialmente para trabajar con la base de datos y los PHP necesarios, pero una vez configurado el servidor Ubuntu 20.04 dejó de ser necesario.



Figura 6.2: Logo de XAMPP

6.1.3.3. MobaXTerm

MobaXTerm permite trabajar con herramientas de red remotas, como SSH, utilizando una terminal Unix desde Windows. Se ha usado para configurar el servidor Ubuntu 20.04 que aloja el servidor MySQL con la base de datos del sistema, y el servidor Apache con los PHP que se utilizan para trabajar con esta base de datos.



Figura 6.3: Logo de MobaXTerm

6.1.3.4. Git

Git es un software de control de versiones gratuito y de código abierto, diseñado para gestionar los cambios de un repositorio[11].



Figura 6.4: Logo de Git

6.2. CSI 2: GENERACIÓN DEL CÓDIGO DE LOS COMPONENTES Y PROCEDIMIENTOS

Una vez desarrolladas las aplicaciones del museo y la administración, se han documentado las clases de ambos proyectos. Esta documentación se ha generado utilizando *Compodoc*, una herramienta similar a la conocida JavaDoc para Angular. Debido a la extensión de la documentación los dos proyectos, se adjunta como archivos externos, como se indica más adelante en el Contenido entregado en los anexos (sección 8.3).

6.3. CSI 3: EJECUCIÓN DE LAS PRUEBAS UNITARIAS

6.4. CSI 4: EJECUCIÓN DE LAS PRUEBAS DE INTEGRACIÓN

Tabla 6.1: Ejecución de pruebas: Consultar periodos (museo)

Consultar periodos (museo)		
Prueba	Resultado esperado	¿Superado?
Obtener periodos existentes	El sistema devolverá una lista de todos los periodos existentes.	Sí
Obtener periodo por nombre	El sistema devolverá una lista de los periodos cuyo nombre contenga el texto introducido.	Sí
Obtener periodo por años	El sistema devolverá una lista de los periodos cuyos años coincidan con los introducidos.	Sí

Tabla 6.2: Ejecución de pruebas: Consultar componentes (museo)

Consultar componentes (museo)		
Prueba	Resultado esperado	¿Superado?
Obtener componentes de un periodo	El sistema devolverá una lista de todos los componentes pertenecientes al periodo.	Sí

Tabla 6.3: Ejecución de pruebas: Iniciar sesión

Iniciar sesión		
Prueba	Resultado esperado	¿Superado?
Iniciar sesión con datos válidos	El sistema permitirá el acceso a la página de administración.	Sí
Iniciar sesión con datos incorrectos	El sistema no permitirá el acceso y se mostrará un error.	Sí

Tabla 6.4: Ejecución de pruebas: Consultar periodos (administración)

Consultar periodos (administración)		
Prueba	Resultado esperado	¿Superado?
Obtener periodos existentes	El sistema devolverá una lista de todos los periodos existentes.	Sí

Tabla 6.5: Ejecución de pruebas: Añadir periodo

Añadir periodo		
Prueba	Resultado esperado	¿Superado?
Añadir periodo nuevo (periodo 2)	El sistema tendrá un periodo más.	Sí
Añadir periodo que ya existe (periodo 1)	El sistema no añadirá el periodo y responderá con un error.	Sí
Añadir periodo con campos vacíos	El sistema no añadirá el periodo y responderá con un error.	Sí

Tabla 6.6: Ejecución de pruebas: Modificar periodo

Modificar periodo		
Prueba	Resultado esperado	¿Superado?
Modificar periodo existente (periodo 1)	El sistema actualizará los datos del periodo.	Sí
Modificar un periodo que no existe (periodo 25)	El sistema responderá con un error.	Sí
Modificar periodo dejando campos vacíos	El sistema no actualizará el periodo y responderá con un error.	Sí

Tabla 6.7: Ejecución de pruebas: Eliminar periodo

Eliminar periodo		
Prueba	Resultado esperado	¿Superado?
Eliminar un periodo existente (periodo 1)	El sistema tendrá un periodo menos.	Sí
Eliminar un periodo que no existe (periodo 25)	El sistema responderá con un error.	Sí

Tabla 6.8: Ejecución de pruebas: Consultar componentes (administración)

Consultar componentes (administración)		
Prueba	Resultado esperado	¿Superado?
Obtener componentes de un periodo	El sistema devolverá una lista de todos los componentes pertenecientes al periodo.	Sí

Tabla 6.9: Ejecución de pruebas: Añadir componente

Añadir componente		
Prueba	Resultado esperado	¿Superado?
Añadir nuevo componente (CPU 2)	El sistema tendrá un componente más.	Sí
Añadir componente que ya existe (CPU 1)	El sistema no añadirá el componente y responderá con un error.	Sí
Añadir componente a un periodo que no existe (CPU 3, periodo 25)	El sistema no añadirá el componente y responderá con un error.	Sí
Añadir componente con campos obligatorios vacíos	El sistema no añadirá el componente y responderá con un error.	Sí

Tabla 6.10: Ejecución de pruebas: Modificar componente

Modificar componente		
Prueba	Resultado esperado	¿Superado?
Modificar componente existente (CPU 1)	El sistema actualizará los datos del componente.	Sí
Modificar un componente que no existe (CPU 30)	El sistema responderá con un error.	Sí
Modificar componente dejando campos obligatorios vacíos	El sistema no actualizará el componente y responderá con un error.	Sí

Tabla 6.11: Ejecución de pruebas: Eliminar componente

Eliminar componente		
Prueba	Resultado esperado	¿Superado?
Eliminar un componente existente (CPU 1)	El sistema tendrá un componente menos.	Sí
Eliminar un componente que no existe (CPU 30)	El sistema responderá con un error.	Sí

6.5. CSI 5: EJECUCIÓN DE LAS PRUEBAS DEL SISTEMA

6.5.1. Prueba de Usabilidad

Una vez desarrollada la aplicación, podemos pasar la guía de usabilidad desarrollada por Yusef Hassan Montero[12].

Tabla 6.12: Checklist de pruebas de Usabilidad

Criterios	¿Cumplido?
Generales	
¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos? ¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?	Sí
¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?	Sí
¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web? El diseño de la página de inicio debe ser diferente al resto de páginas y cumplir la función de 'escaparate' del sitio.	Sí
¿La estructura general del sitio web está orientada al usuario? Los sitios web deben estructurarse pensando en el usuario, sus objetivos y necesidades. La estructura interna de la empresa u organización, cómo funciona o se organiza no interesan al usuario.	Sí
¿El look & feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web? Ciertas combinaciones de colores ofrecen imágenes más o menos formales, serias o profesionales.	Sí
¿Es coherente el diseño general del sitio web? Se debe mantener una coherencia y uniformidad en las estructuras y colores de todas las páginas. Esto sirve para que el usuario no se desoriente en su navegación.	Sí
¿Es reconocible el diseño general del sitio web? Cuanto más se parezca el sitio web al resto de sitios web, más fácil será de usar.	Sí
¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza? Las fechas que se muestren en la página deben corresponderse con actualizaciones, noticias, eventos...no con la fecha del sistema del usuario.	-
Identidad e información	
¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?	Sí
El Logotipo, ¿es significativo, identifiable y suficientemente visible?	Sí
El eslogan o tagline, ¿expresa realmente qué es la empresa y qué servicios ofrece?	-

¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster'...?	Sí
¿Se proporciona mecanismos para ponerse en contacto con la empresa? (email, teléfono, dirección postal, fax...)	-
¿Se proporciona información sobre la protección de datos de carácter personal de los clientes o los derechos de autor de los contenidos del sitio web?	-
En artículos, noticias, informes... ¿Se muestra claramente información sobre el autor, fuentes y fechas de creación y revisión del documento?	-
Lenguaje y redacción	
¿El sitio web habla el mismo lenguaje que sus usuarios? Se debe evitar usar un lenguaje corporativista. Así mismo, hay que prestarle especial atención al idioma, y ofrecer versiones del sitio en diferentes idiomas cuando sea necesario.	Sí
¿Emplea un lenguaje claro y conciso?	Sí
¿Es amigable, familiar y cercano? Es decir, lo contrario a utilizar un lenguaje constantemente imperativo, mensajes crípticos, o tratar con "desprecio. ^a l usuario.	Sí
¿1 párrafo = 1 idea? Cada párrafo es un objeto informativo. Trasmita ideas, mensajes...Se deben evitar párrafos vacíos o varios mensajes en un mismo párrafo.	Sí
Rotulado	
Los rótulos, ¿son significativos? Ejemplo: evitar rótulos del tipo "haga clic aquí".	Sí
¿Usa rótulos estándar? Siempre que exista un .estándar.comúnmente aceptado para el caso concreto, como "Mapa del Sitio. ^o ^acerca de...".	Sí
¿Usa un único sistema de organización, bien definido y claro? No se deben mezclar diferentes. Los sistemas de organización son: alfabetico, geográfico, cronológico, temático, orientado a tareas, orientado al público y orientado a metáforas.	Sí
¿Utiliza un sistema de rotulado controlado y preciso? Por ejemplo, si un enlace tiene el rótulo "Quiénes somos", no puede dirigir a una página cuyo encabezamiento sea ^acerca de"	Sí
El título de las páginas, ¿Es correcto? ¿Ha sido planificado? Relacionado con la capacidad para poder buscar y encontrar el sitio web.	Sí
Estructura y navegación	
La estructura de organización y navegación, ¿Es la más adecuada? Hay varios tipos de estructuras: jerárquicas, hipertextual, facetada...	Sí
En el caso de estructura jerárquica, ¿Mantiene un equilibrio entre Profundidad y Anchura?	-
En el caso de ser puramente hipertextual, ¿Están todos los clústeres de nodos comunicados? Aquí se mide la distancia entre nodos.	Sí

¿Los enlaces son fácilmente reconocibles como tales? ¿Su caracterización indica su estado (visitados, activos...)? Los enlaces no sólo deben reconocerse como tales, sino que su caracterización debe indicar su estado, y ser reconocidos como una unidad	No
En menús de navegación, ¿Se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística? No se deben superar los 7 ± 2 elementos, ni los 2 o, como mucho, 3 términos por elemento.	Sí
¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace? Relacionado con el nivel de significación del rótulo del enlace, aunque también con: el uso de globos de texto, información contextual, la barra de estado del navegador...	Sí
¿Se ha controlado que no haya enlaces que no llevan a ningún sitio? Enlaces que no llevan a ningún sitio: Los enlaces rotos, y los que enlazan con la misma página que se está visualizando (por ejemplo, enlaces a la "home" desde la misma página de inicio)	Sí
¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación? ...como breadcrumbs, enlaces a la página de inicio...recuerde que el logo también es recomendable que enlace con la página de inicio.	Sí
Las imágenes enlace, ¿se reconocen como clicables? ¿Incluyen un atributo 'title' describiendo la página de destino? En este sentido, también hay que cuidar que no haya imágenes que parezcan enlaces y en realidad no lo sean.	No
¿Se ha evitado la redundancia de enlaces?	Sí
¿Se ha controlado que no haya páginas "huérfanas"? Páginas huérfanas: que, aun siendo enlazadas desde otras páginas, éstas no enlacen con ninguna.	Sí
Layout de la página	
¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia? (como por ejemplo la zona central)	Sí
¿Se ha evitado la sobrecarga informativa? Esto se consigue haciendo un uso correcto de colores, efectos tipográficos y agrupaciones para discriminar información. Los grupos diferentes de objetos informativos de una página deben ser 7 ± 2 .	Sí
¿Es una interfaz limpia, sin ruido visual?	Sí
¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?	Sí
¿Se hace un uso correcto del espacio visual de la página? Es decir, que no se desaproveche demasiado espacio con elementos de decoración, o grandes zonas en "blanco", y que no se adjudique demasiado espacio a elementos de menor importancia.	Sí
¿Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página? (La jerarquía visual se utiliza para orientar al usuario)	Sí

¿Se ha controlado la longitud de página? Se debe evitar en la medida de lo posible el scrolling. Si la página es muy extensa, se debe fraccionar.	No
Elementos multimedia	
¿Las fotografías están bien recortadas? ¿Son comprensibles?	Sí
¿Se ha cuidado su resolución?	-
¿Las metáforas visuales son reconocibles y comprensibles por cualquier usuario? (prestar especial atención a usuarios de otros países y culturas)	-
¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?	Sí
¿Se ha evitado el uso de animaciones cíclicas?	Sí
Accesibilidad	
¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?	Sí
¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleadas facilitan la lectura?	Sí
¿Existe un alto contraste entre el color de fuente y el fondo?	Sí
¿Incluyen las imágenes atributos 'alt' que describan su contenido?	Sí
¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla? Se debe prestar atención a: JScript, CSS, tablas, fuentes...	Sí
¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar plugins adicionales?	Sí
¿Se ha controlado el peso de la página? Se deben optimizar las imágenes, controlar el tamaño del código JScript...	No
¿Se puede imprimir la página sin problemas? Leer en pantalla es molesto, por lo que muchos usuarios preferirán imprimir las páginas para leerlas. Se debe asegurar que se puede imprimir la página (no salen partes cortadas), y que el resultado es legible.	Sí
Control y retroalimentación	
¿Tiene el usuario todo el control sobre el interfaz? Se debe evitar el uso de ventanas pop-up, ventanas que se abren a pantalla completa, banners intrusivos...	Sí
¿Se informa constantemente al usuario acerca de lo que está pasando? Si el usuario tiene que esperar hasta que se termine una operación, se debe mostrar un mensaje indicándoselo y que debe esperar, con el tiempo de espera estimado o una barra de progreso.	-
¿Se informa al usuario de lo que ha pasado? Por ejemplo, cuando un usuario valora un artículo o responde a una encuesta, se le debe informar de que su voto ha sido procesado correctamente.	Sí

Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema? Siempre es mejor intentar evitar que se produzcan errores a tener que informar al usuario del error.	Sí
¿Posee el usuario libertad para actuar? NO restringir la libertad del usuario: Uso de animaciones que no pueden ser "saltadas", páginas en las que desaparecen los botones de navegación, no impida al usuario poder usar el botón derecho de su ratón...	Sí
¿Se ha controlado el tiempo de respuesta? Esto tiene que ver con el peso de cada página (accesibilidad) y tiene relación con el tiempo que tarda el servidor en finalizar una tarea y responder. El tiempo máximo que esperará un usuario son 10 segundos	Sí
Aclaraciones	
¿Se ha evaluado adecuadamente la orientación del usuario? (Donde estoy, como volver, que he visitado, que va a pasar)	Sí
¿Se ha usado correctamente la publicidad?	-

A continuación, se mostrarán las respuestas de los usuarios a los cuestionarios diseñados anteriormente (sección 5.4.2.1)

Usuario 1

Tabla 6.13: Cuestionario de carácter general (Usuario 1)

¿Usa un ordenador frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el ordenador?

1. Es parte de mi trabajo o profesión
2. Lo uso básicamente para ocio
3. Solo empleo aplicaciones estilo Office
4. Únicamente leo el correo y navego ocasionalmente

¿Qué busca Vd. principalmente en una aplicación web?

1. Que sea fácil de navegar
 2. Que sea intuitiva
 3. Que sea rápida
-

Tabla 6.14: Cuestionario sobre la aplicación (Usuario 1)

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe dónde está dentro de la aplicación?	X			
¿Necesita ayuda para utilizar la aplicación?				X
¿Le resulta sencillo el uso de la aplicación?	X			
¿Identifica fácilmente la información que se le presenta?	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. espera?	X			
¿El tiempo de respuesta de la aplicación es muy grande?				X
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
El tipo y el tamaño de letra es	X			
Los iconos e imágenes usados son	X			
Los colores empleados son	X			
Diseño de la interfaz	Sí	A veces	No	
¿Le resulta fácil de usar?	X			
¿El diseño de las pantallas es claro y atractivo?				X
¿Es coherente el diseño general del sitio web?				X
¿Cree que el programa está bien estructurado?	X			
Observaciones				
Hacer más visible la división de los períodos en la línea temporal.				

Tabla 6.15: Actividades guiadas (Usuario 1)

<i>Navegar por la línea temporal</i>	
Tiempo empleado:	1 minuto
Problemas encontrados:	No se identificó bien el criterio de división de los componentes en periodos.
Observaciones:	Se propuso hacer una división de periodos por etapas (años 70, 80,...)
<i>Búsqueda por años</i>	
Tiempo empleado:	1 minuto
Problemas encontrados:	Ninguno
Observaciones:	Ninguno
<i>Búsqueda por nombre</i>	
Tiempo empleado:	1 minuto
Problemas encontrados:	Ninguno
Observaciones:	Ninguno
<i>Ver detalles de un periodo</i>	
Tiempo empleado:	1 minuto
Problemas encontrados:	Ninguno
Observaciones:	Ninguno
<i>Ver detalles de un componente</i>	
Tiempo empleado:	1 minuto
Problemas encontrados:	Ninguno
Observaciones:	Ninguno

Usuario 2

Tabla 6.16: Cuestionario de carácter general (Usuario 2)

¿Usa un ordenador frecuentemente?

1. Todos los días
2. Varias veces a la semana
3. Ocasionalmente
4. Nunca

¿Qué tipo de actividades realiza con el ordenador?

1. Es parte de mi trabajo o profesión
2. Lo uso básicamente para ocio
3. Solo empleo aplicaciones estilo Office
4. Únicamente leo el correo y navego ocasionalmente

¿Qué busca Vd. principalmente en una aplicación web?

1. Que sea fácil de navegar
 2. Que sea intuitiva
 3. Que sea rápida
-

Tabla 6.17: Cuestionario sobre la aplicación (Usuario 2)

Facilidad de uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe dónde está dentro de la aplicación?</i>				
<i>¿Necesita ayuda para utilizar la aplicación?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
<i>¿Identifica fácilmente la información que se le presenta?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. espera?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad del interfaz				
Aspectos gráficos	Muy adecuado	Adecuado	Poco adecuado	Nada adecuado
<i>El tipo y el tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				
Diseño de la interfaz	Sí	A veces	No	
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Es coherente el diseño general del sitio web?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
Observaciones				

Tabla 6.18: Actividades guiadas (Usuario 2)

Navegar por la línea temporal
Tiempo empleado:
Problemas encontrados:
Observaciones:
Búsqueda por años
Tiempo empleado:
Problemas encontrados:
Observaciones:
Búsqueda por nombre
Tiempo empleado:
Problemas encontrados:
Observaciones:
Ver detalles de un periodo
Tiempo empleado:
Problemas encontrados:
Observaciones:
Ver detalles de un componente
Tiempo empleado:
Problemas encontrados:
Observaciones:

Una vez terminadas estas pruebas, se realizaron los siguientes cambios para mejorar la aplicación web en base a los resultados:

- Se han modificado los enlaces para identificar aquellos con estado activo.
- Se ha añadido el atributo *title* y un borde sombreado a las imágenes enlace para distinguirlas de las imágenes que no lo son.
- Se ha enfatizado el borde de las cajas que contienen cada periodo en la línea temporal del museo.

6.6. CSI 6: ELABORACIÓN DE LOS MANUALES DE USUARIO

6.6.1. Manual de Instalación

En este manual se detallarán los pasos necesarios para realizar las instalaciones necesarias para la ejecución del sistema.

En primer lugar, es necesario instalar NodeJS (se puede descargar en <https://nodejs.org/en/download/>) y reiniciar el sistema, ya que con esta instalación se ha cambiado la configuración de variables del PATH.

Para los siguientes pasos, es necesario el uso de la terminal del sistema.

Usando npm, el gestor de paquetes de NodeJS, hay que instalar Angular CLI. Para ello hay que ejecutar el comando `npm install -g @angular/cli`.

```
C:\Users\isabe>npm install -g @angular/cli
```

Figura 6.5: Instalación de Angular CLI

Por último, desde la carpeta que ubica tanto el proyecto del museo (museo-eii) como el de la administración (museo-eii-admin), se ejecuta el comando `npm install` para instalar los paquetes necesarios.

```
C:\Users\isabe\Documents\TFG\museo-eii\museo-eii>npm install
```

Figura 6.6: Instalación de los paquetes del proyecto del museo

```
C:\Users\isabe\Documents\TFG\museo-eii\museo-eii-admin>npm install
```

Figura 6.7: Instalación de los paquetes del proyecto de administración

6.6.2. Manual de Ejecución

Una vez completada la instalación siguiendo los pasos descritos en el apartado anterior, se pueden ejecutar ambas aplicaciones utilizando el comando `ng serve -o`, `npm start` o `npm run ng serve -o`. Esto hará que la aplicación esté disponible en `http://localhost:4200`.

```
C:\Users\isabe\Documents\TFG\museo-eii\museo-eii>ng serve -o
10% building 3/3 modules 0 active(node:14092) [DEP0111] DeprecationWarning: Access to process.binding('http_parser') is
deprecated.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:14092) [DEP0148] DeprecationWarning: Use of deprecated folder mapping "./" in the "exports" field module resolutio
n of the package at C:\Users\isabe\Documents\TFG\museo-eii\museo-eii\node_modules\tslib\package.json.
Update this package.json to use a subpath pattern like "./".
chunk {main} main.js, main.js.map (main) 142 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 150 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 964 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.38 MB [initial] [rendered]
Date: 2022-04-27T12:24:42.301Z - Hash: f4defa2709d0765124e - Time: 6678ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

Figura 6.8: Ejecución de la aplicación del museo

```
C:\Users\isabe\Documents\TFG\museo-eii\museo-eii-admin>ng serve -o
10% building 3/3 modules 0 active(node:56048) [DEP0111] DeprecationWarning: Access to process.binding('http_parser') is
deprecated.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:56048) [DEP0148] DeprecationWarning: Use of deprecated folder mapping "./" in the "exports" field module resolutio
n of the package at C:\Users\isabe\Documents\TFG\museo-eii\museo-eii-admin\node_modules\tslib\package.json.
Update this package.json to use a subpath pattern like "./".
chunk {main} main.js, main.js.map (main) 255 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 150 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 1.11 MB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 5 MB [initial] [rendered]
Date: 2022-04-27T12:25:28.123Z - Hash: 6832c2cf5609987bce1d - Time: 8925ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.
```

Figura 6.9: Ejecución de la aplicación de administración

6.6.3. Manual de Usuario

6.6.3.1. Museo

Al acceder a la web observamos la página de inicio. La parte superior de esta página está presente en toda la aplicación web y, por orden de izquierda a derecha, observamos:

- El logo de la escuela, que redirige a esta página de inicio.
- Museo, que redirige a la vista general del museo.
- Acerca de.
- Un selector de idioma, que permite cambiar entre inglés y español.

La parte inferior, que contiene enlaces a las redes sociales de la escuela, también está presente en toda la aplicación web.

En la parte central se encuentra el contenido específico de la página de inicio: una bienvenida a la página web y un botón que nos dirige a la vista general del museo.

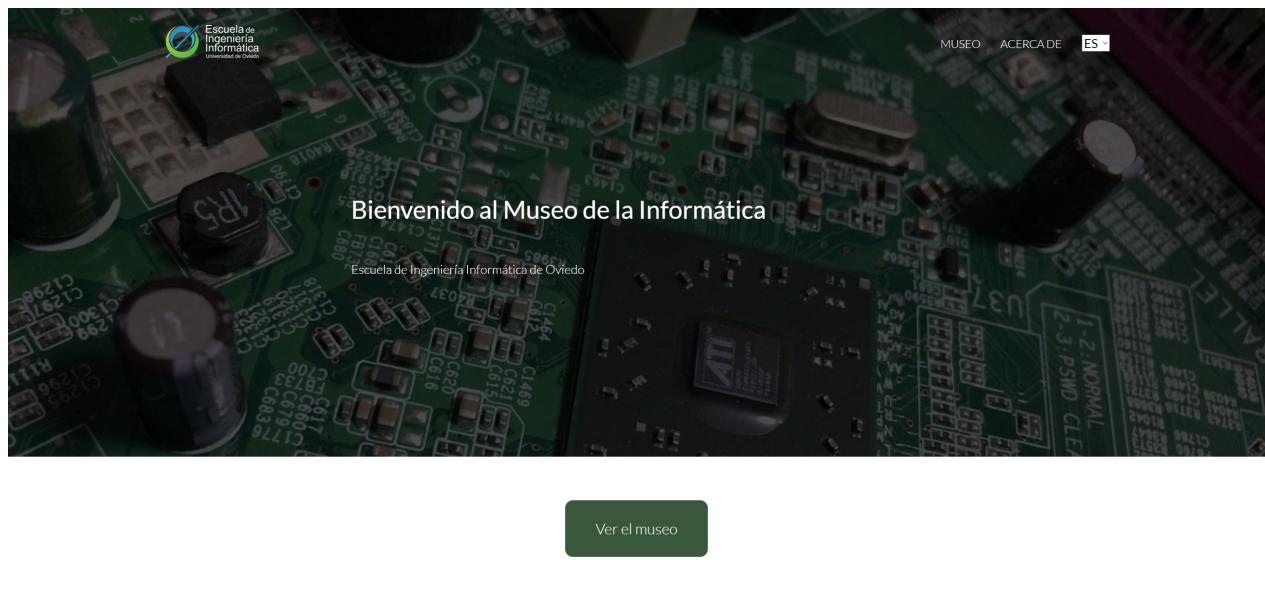


Figura 6.10: Manual de usuario: Inicio

En la vista general del museo hay una línea temporal y filtros de búsqueda.

En cada elemento de la línea temporal se muestra el nombre del periodo con un enlace al mismo, los años que comprende dicho periodo, y los nombres de los componentes pertenecientes al periodo, también con enlaces a cada uno de ellos.

La búsqueda puede realizarse por años o por nombre. Se puede filtrar por años mediante la barra deslizadora, y se mostrarán entonces todos aquellos períodos que, parcialmente o en su totalidad, tengan componentes pertenecientes a esos años. La búsqueda por nombre se realiza tras escribir en el recuadro de búsqueda y pulsar la tecla *Enter*, y el resultado será aquellos períodos cuyo nombre o el nombre de alguno de sus componentes contenga el texto buscado.

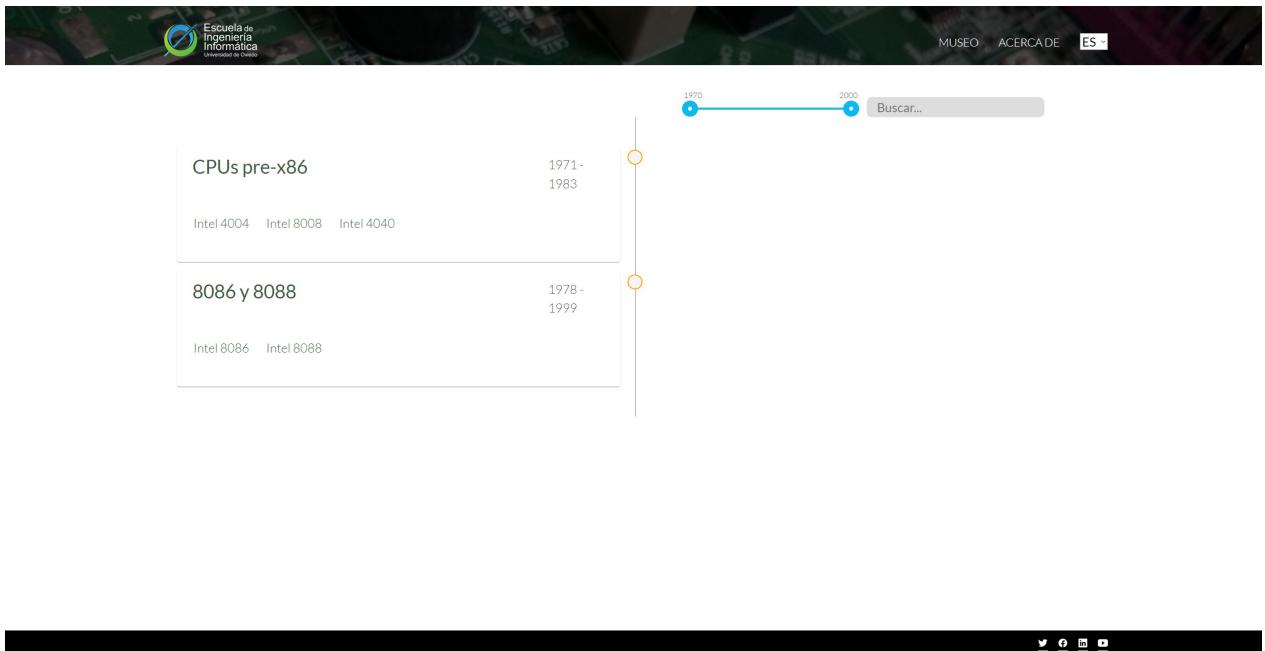


Figura 6.11: Manual de usuario: Vista general del museo

Al entrar en un periodo, en la parte superior podemos ver un menú, en la izquierda se mostraría el periodo anterior cronológicamente, y en la izquierda el periodo siguiente (si existen). El contenido principal de la página son los detalles del periodo: nombre, características, una lista de curiosidades (sabías que...), eventos informáticos ocurridos en esos años, los componentes pertenecientes al periodo (mostrando una imagen y el nombre, con un enlace al componente), y una serie de sistemas famosos que llevan esos componentes.

The screenshot shows the details for the 'CPUs pre-x86' period (1971-1983). At the top right, there is a link '8086 y 8088 >'. The main title is 'CPUs pre-x86'. Below it, there is a section titled 'Características:' with the following text:
 El 4004 formaba parte de la familia de chips MCS, que complementaban sus funciones.
 Sus sucesores 4040, 8080 y 8085 tenían su propia familia de chips de soporte.

On the right, there is a 'Sabías que...' box with the following text:
 > Intel originalmente fabricaba solo chips de memoria y se inició en la fabricación de CPUs con estos modelos.
 > Estos chips no tienen memoria caché ni tiene sentido hablar de velocidad de bus.

Below the characteristics section, there are images and names for three chips: Intel 4004, Intel 8008, and Intel 4040. To the right, there are images and names for four systems: Busicom 141PF, Datapoint 2200, Flying Carpet, and a small image of a floppy disk labeled '1971: Floppy, pantallas LCD'. At the bottom, there are three small boxes with the text '1970: Apollo 13', '1971: email', and '1971: Floppy, pantallas LCD'.

Figura 6.12: Manual de usuario: Detalles del periodo (museo)

Por último, al acceder a un componente, podemos ver una galería de fotos que se abrirán en grande al pulsar sobre ellas, una descripción del componente y un listado de características. Además, en el lateral izquierdo hay un menú que permite navegar entre períodos (ver el anterior, el actual y el siguiente) y entre los componentes del periodo actual.

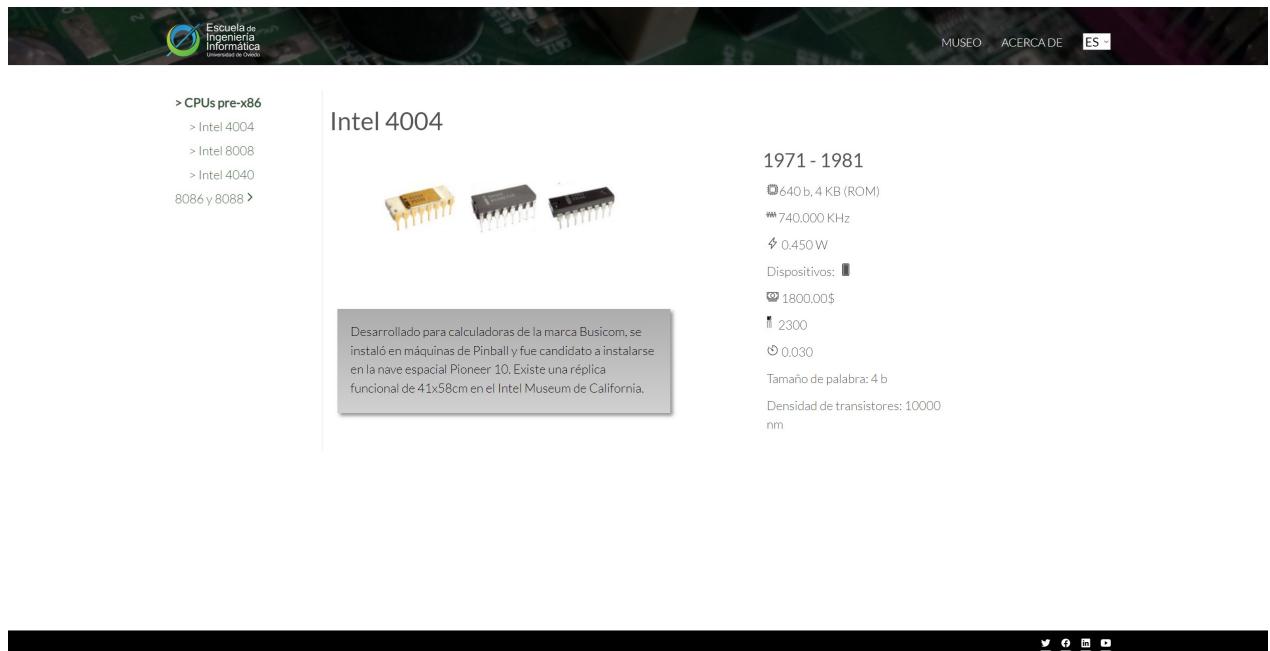


Figura 6.13: Manual de usuario: Detalles del componente (museo)

6.6.3.2. Administración del museo

Al entrar a la web de administración del museo nos encontramos con el inicio de sesión. Es necesario indicar el correo electrónico y la contraseña para acceder.

The screenshot shows a login form titled 'Museo de la Informática - Administrador'. It has two input fields: 'Email' and 'Contraseña', both with placeholder text. Below the fields is a large green button labeled 'INICIAR SESIÓN'.

Figura 6.14: Manual de usuario: Inicio de sesión

Si el administrador necesita cambiar la contraseña, podrá hacerlo accediendo a la máquina Ubuntu 20.04 donde se encuentra alojado el servidor Apache que contiene los archivos PHP y la base de datos del sistema, ya que habrá un fichero ejecutable que solicitará la nueva contraseña y realizará el cambio. Este fichero solo estará disponible en esta máquina ya que, al haber un solo usuario administrador, es más sencillo hacer el cambio de contraseña de esta forma en lugar de enviar un correo con un enlace temporal para realizarlo, y es seguro ya que el administrador es el único usuario con acceso a la máquina.

```
redondo@xampp:~/Documentos$ php changePassword.php
Introduzca el usuario: uo257829@uniovi.es
Introduzca una nueva contraseña: NewPass
Repita la contraseña: NewPass
Contraseña actualizada con éxito.
```

Figura 6.15: Manual de usuario: Cambio de contraseña

Lo primero que se muestra una vez iniciada la sesión es un listado de los períodos existentes, mostrando sus nombres con un enlace a cada uno de ellos, y permitiendo editar y eliminar cada periodo. Eliminar un periodo borrará también los componentes asociados al mismo, para ello se mostrará un aviso y se pedirá confirmación. En el lateral izquierdo hay un menú que se incluye en todas las páginas de la aplicación, desde el que se puede acceder a este listado de períodos, y a los formularios para añadir períodos y componentes.

Listado de períodos		
Nombre	Editar	Eliminar
CPUUs pre-x86		
8086 y 8088		

Figura 6.16: Manual de usuario: Listado de períodos

Los detalles de un periodo y del componente muestran los mismos datos explicados anteriormente en el manual de usuario del museo, con la diferencia de en que cada una de estas páginas se muestra una opción para editar el periodo o el componente que estemos visualizando, y en el listado de componentes del periodo también se da la opción de editar o eliminar cada uno de ellos.

- > Listado de periodos
- > Añadir periodo
- > Añadir componente a periodo existente

CPUs pre-x86

[Editar](#)

Detalles

El 4004 formaba parte de la familia de chips MCS, que complementaban sus funciones.

Sus sucesores 4040, 8080 y 8085 tenían su propia familia de chips de soporte.

Sabías qué...

Intel originalmente fabricaba solo chips de memoria y se inició en la fabricación de CPUs con estos modelos.

Estos chips no tienen memoria caché ni tiene sentido hablar de velocidad de bus.

Eventos

1970: Apollo 131971: email1971: Floppy, pantallas LCD

Sistemas famosos



Componentes

Nombre	Editar	Eliminar
Intel 4004	/	
Intel 8008	/	
Intel 4040	/	

Figura 6.17: Manual de usuario: Detalles de un periodo (administración)

- > Listado de periodos
- > Añadir periodo
- > Añadir componente a periodo existente

[←](#)

Intel 4004

[Editar](#)

Periodo

CPUs pre-x86

Familia

Intel

Años

1971 - 1981

Descripción

Desarrollado para calculadoras de la marca Busicom, se instaló en máquinas de Pinball y fue candidato a instalarse en la nave espacial Pioneer 10. Existe una réplica funcional de 41x58cm en el Intel Museum de California.



Memoria ROM

4 KB

Memoria RAM

640 b

Frecuencia de reloj

740.000 KHz

Consumo energético

0.450 W

Tamaño de palabra

4 b

Nanómetros de los transistores

10000

Passmark

0.030

Número de transistores

2300

Figura 6.18: Manual de usuario: Detalles de un componente (administración)

En el formulario de añadir un periodo hay cuatro entradas de texto para nombre,

detalles, curiosidades y eventos del periodo. Todos ellos deben rellenarse obligatoriamente para poder guardar el periodo. Si se pulsa el botón *Cancelar*, el formulario se vaciará de nuevo. Al pulsar *Guardar y continuar* con el formulario completo, se añadirá el periodo a la base de datos del sistema y nos redirirá al formulario para añadir componentes. En cambio, si el formulario no es válido se mostrará un error y no se añadirá.

Añadir periodo

> Listado de períodos
> Añadir periodo
> Añadir componente a periodo existente

Nombre:

Detalles:

Sabías que...
Introduzca cada característica en una línea

Eventos:
Introduzca cada evento en una línea

Eventos:

Figura 6.19: Manual de usuario: Formulario para añadir un periodo

A la hora de editar un periodo, el formulario funcionará igual que al añadirlo, con la diferencia de que los valores iniciales serán los del periodo que se está editando.

Editar periodo

<

> Listado de períodos
> Añadir periodo
> Añadir componente a periodo existente

Nombre: CPUs pre-x86

Detalles: El 4004 formaba parte de la familia de chips MCS, que complementaban sus funciones. Sus sucesores 4040, 8080 y 8085 tenían su propia familia de chips de soporte.

Sabías que...
Introduzca cada característica en una línea

Eventos:
Introduzca cada evento en una línea

Eventos: 1970: Apollo 13
1971: email
1971: Floppy, pantallas LCD

Figura 6.20: Manual de usuario: Formulario para editar un periodo

Los formularios para añadir y editar componentes funcionan de la misma forma que los de añadir y editar períodos, pero en este caso, hay campos que no son obligatorios,

como la subida de imágenes y el sistema famoso. Además, al añadir o editar componentes se puede seleccionar su tipo: CPU o componente genérico. Al seleccionar CPU se muestran los campos de memoria ROM, memoria RAM, frecuencia de reloj, consumo energético, tamaño de palabra, nanómetros de transistores, passmark y número de transistores.

The screenshot shows a user interface for adding a component. On the left, there's a sidebar with navigation links: 'Listado de períodos', 'Añadir periodo', and 'Añadir componente a periodo existente'. The main area is titled 'Añadir componente' and contains the following fields:

- Selección del periodo al que pertenece:** A dropdown menu set to 'CPUs pre-x86'.
- Tipo:** A dropdown menu set to 'CPU'.
- Nombre:** An input field labeled 'Nombre'.
- Descripción:** A large text area labeled 'Descripción'.
- Familia:** An input field labeled 'Familia'.
- Año de salida:** A dropdown menu set to '1970'.
- Año de fin:** A dropdown menu set to '1990'.
- Precio:** An input field set to '100' with a '\$' symbol.
- Utilizado en dispositivos:** Checkboxes for 'Portátiles' (unchecked) and 'De escritorio' (unchecked).
- Sistema famoso que lo utiliza:** An input field labeled 'Nombre del sistema' and a 'Subir imagen' button.
- Subir imágenes del componente:** A button.
- Memoria ROM:** Two input fields, both set to '0'.
- Frecuencia de reloj:** Two input fields, both set to '0'.
- Tamaño de palabra:** Two input fields, both set to '0'.
- Passmark:** An input field set to '0'.
- Memoria RAM:** Two input fields, both set to '0'.
- Consumo energético:** Two input fields, both set to '0'.
- Nanómetros de los transistores:** An input field set to '0' with 'nm' suffix.
- Número de transistores:** An input field set to '0'.

At the bottom are 'Cancelar' and 'Guardar' buttons.

Figura 6.21: Manual de usuario: Formulario para añadir un componente

> Listado de períodos
 > Añadir periodo
 > Añadir componente a periodo existente

← Editar componente

Seleccione el periodo al que pertenece: CPUs pre-x86

Nombre: Intel 4004

Descripción: Desarrollado para calculadoras de la marca Busicom, se instaló en máquinas de Pinball y fue candidato a instalarse en la nave espacial Pioneer 10. Existe una réplica funcional de 41x58cm en el Intel Museum de California.

Familia: Intel

Año de salida: 1971 Año de fin: 1981 Precio: 1800,00 \$

Utilizado en dispositivos: Portátiles De escritorio

Sistema famoso que lo utiliza: Busicom 141PF

Memoria ROM:	4 KB	Memoria RAM:	640 b
Frecuencia de reloj:	740,000 KHz	Consumo energético:	0,450 W
Tamaño de palabra:	4 b	Nanómetros de los transistores:	10000 nm
Passmark:	0,030	Número de transistores:	2300

Figura 6.22: Manual de usuario: Formulario para editar un componente

6.6.4. Manual del Programador

En este manual se explicará cómo ampliar la aplicación añadiendo nuevos tipos de componentes además de CPUs. Primero habría que crear una nueva clase para cada tipo que se desee añadir. Cada una de estas clases implementarán la interfaz *MyComponent* y heredarán de *GenericComp*. También habría que actualizar la enumeración *CompTypes*. Estos tres elementos mencionados se encuentran en el archivo *comp.ts*, que forma parte tanto del proyecto del museo (museo-eii) como de la administración (museo-eii-admin). Una vez realizado esto, común a ambos proyectos, se explicará qué debe añadirse a cada uno de ellos en específico, así como a la base de datos.

6.6.4.1. Museo

En el proyecto del museo (museo-eii) deberá generarse un componente de Angular para cada tipo añadido, se llamará ‘*new type*’*DetailsComponent* y será hijo de *CompDetailsComponent*, del que recibirá como input el atributo *comp*. Este solo se mostrará cuando *comp* sea una instancia del tipo correspondiente a ‘*new type*’. En ‘*new type*-details.component.html’ se listarán las características de *comp*.

Además, en el método *getComp* de *CompDetailsComponent* habrá que añadir las comprobaciones necesarias para mostrar los nuevos tipos definidos.

6.6.4.2. Administración del museo

En el proyecto de la administración (museo-eii-admin) habrá que generar dos componentes de Angular nuevos por cada tipo añadido:

- ‘new type’*FormComponent*, hijo de *AddCompComponent* y de *EditCompComponent*. De ambos recibe como input el atributo *model*. En ‘new type’-*form.component.html* se incluirán los campos del formulario que se correspondan con las características del tipo creado. Se mostrará cuando *model* sea una instancia del tipo correspondiente a ‘new type’.

En el método *createModel* de *AddCompComponent* habrá que añadir la opción de crear un objeto de este nuevo tipo, y también se añadirán las comprobaciones necesarias en los métodos *isValid* y *cloneComp* de *AddCompComponent* y *EditCompComponent*.

- ‘new type’*DetailsComponent*, hijo de *MyComponentComponent*, del que recibe como input el atributo *c*. En este caso, se hará exactamente lo mismo que lo mencionado anteriormente al añadir ‘new type’*DetailsComponent* en el proyecto del museo, ya que ambos componentes son para mostrar las características de cada tipo.

6.6.4.3. Base de datos

En la base de datos habría que crear una tabla por cada nuevo tipo de componente, con los campos necesarios para este que no estén ya incluidos en la tabla *components*. La clave primaria de esta tabla sería también una clave foránea, el identificador del componente en la tabla *components*. Una vez creadas las tablas correspondientes, habría que modificar las comprobaciones y consultas realizadas en los archivos *getComp.php*, *updateComp.php* y *postComp.php* para incluir los nuevos tipos creados.

Capítulo 7

IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA

FASE DE DESARROLLO

IAS

7.1. IAS 1: ESTABLECIMIENTO DEL PLAN DE IMPLANTACIÓN

7.2. IAS 4: CARGA DE DATOS AL ENTORNO DE OPERACIÓN

7.3. IAS 5: PRUEBAS DE IMPLANTACIÓN DEL SISTEMA

7.4. IAS 7: PREPARACIÓN DEL MANTENIMIENTO DEL SISTEMA

Capítulo 8

APÉNDICES

8.1. PROBLEMAS ENCONTRADOS DURANTE EL DESARROLLO

8.2. CONCLUSIONES

8.3. AMPLIACIONES

En esta sección se proponen una serie de ampliaciones del sistema para realizar en un futuro.

Internacionalización de la base de datos

Actualmente, la aplicación web permite escoger entre dos idiomas, español e inglés, pero no está completamente internacionalizada. Solo se encuentran traducidos los elementos que se definen directamente en el HTML (títulos, menú de navegación, texto presentación). Los datos de los períodos y los componentes (descripciones, características, curiosidades, ...) se encuentran solo en español, ya que se leen directamente de la base de datos, en la que se han introducido los datos en este idioma. Por tanto, una posible ampliación sería terminar la internacionalización de la aplicación traduciendo también estos campos leídos de la base de datos.

Más tipos de componentes

En la aplicación desarrollada solo se puede añadir y visualizar componentes de tipo genérico o CPUs. Una ampliación a realizar en un futuro será añadir nuevos componentes como, por ejemplo, GPUs. El proyecto actual se ha desarrollado teniendo en cuenta esta posible ampliación, utilizando la interfaz *MyComponent* y dividiendo los componentes de Angular de forma que se puedan añadir nuevas clases realizando los mínimos cambios posibles.

Generación automática de carteles

Una ampliación que sería interesante para el proyecto del museo es la generación automática de carteles desde la parte de administración. Esto ayudaría a crear los carteles para el museo físico, ya que pulsando un único botón se aprovecharían los datos ya presentes en la aplicación para generar un cartel informativo del periodo seleccionado y sus respectivas CPUs.

Bibliografía

- [1] Jose Manuel Redondo, “Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo.” https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo, 2019. Online; accessed 13 Jul 2020.
- [2] Jose Manuel Redondo, “Creación y evaluación de plantillas para trabajos de fin de grado como buena práctica docente,” *Revista de Innovación y Buenas Prácticas Docentes*, vol. pp, no. pp, p. pp, 2020.
- [3] MDN contributors, “JavaScript.” <https://developer.mozilla.org/es/docs/Web/JavaScript>, 2020. Online; accessed 10 Oct 2021.
- [4] Jesús Lucas, “Qué es NodeJS y para qué sirve.” <https://openwebinars.net/blog/que-es-nodejs/>, 2019. Online; accessed 10 Oct 2021.
- [5] “What is Angular?” <https://angular.io/guide/what-is-angular>. Online; accessed 10 Oct 2021.
- [6] “Typed JavaScript at Any Scale..” <https://www.typescriptlang.org/>. Online; accessed 10 Oct 2021.
- [7] “Angular Style Guide.” <https://angular.io/guide/styleguide>. Online; accessed 13 Abr 2022.
- [8] Ryan Sechrest, “PHP Code Style Guide.” <https://gist.github.com/ryansechrest/8138375>. Online; accessed 13 Abr 2022.
- [9] “Visual Studio Code.” <https://code.visualstudio.com/docs>. Online; accessed 13 Abr 2022.
- [10] “XAMPP.” <https://www.apachefriends.org/es/index.html>. Online; accessed 13 Abr 2022.
- [11] “Git.” <https://git-scm.com/>. Online; accessed 13 Abr 2022.
- [12] Yusef Hassan Montero, “Guía de Evaluación Heurística de Sitios Web.” <http://www.nosolousabilidad.com/articulos/heuristica.htm>, 2003. Online; accessed 02 Jun 2022.
- [13] J. M. Requena, “El consejero de Universidad pide apostar por la innovación y generar conocimiento.” <https://www.lne.es/asturias/2019/08/13/consejero-univers>

[idad-pide-apostar-innovacion/2514937.html](#), 2019. Online; accessed 13 Jul 2020.

ANEXOS

PLAN DE GESTIÓN DE RIESGOS

CONTENIDO ENTREGADO EN LOS ANEXOS

Contenidos

Además de este documento, se hace entrega de una carpeta comprimida “.zip” en la que ahora se describirán sus contenidos. Se estructurará también la organización del código fuente.

- **Planificación_TFG.mpp** → Archivo de Microsoft Project que contiene la planificación del proyecto entera.
- **Presupuesto_TFG.xlsx** → Archivo Microsoft Excel que contiene los cálculos del presupuesto del proyecto.
- **Documentación_Compodoc** → Carpeta que contiene la documentación de los proyectos (museo y administración) generada con Compodoc.
 - **Documentación_Museo** → Contiene la documentación del proyecto del museo (museo-eii).
 - **Documentación_Admin** → Contiene la documentación del proyecto de la administración del museo (museo-eii-admin).

Cada una de estas carpetas contiene los archivos HTML con la documentación generada. Abriendo el archivo *index.html* de cada una se puede ver la documentación al completo de su respectivo proyecto.

- **Diagramas** → Carpeta que contiene todos los diagramas utilizados en este documento.
 - *Diagrama_arquitectura_tecnologica.png*
 - *Diagrama_casos_uso_museo.png*
 - *Diagrama_casos_uso_admin.png*
 - *Diagrama_clases_museo-Análisis.png*
 - *Diagrama_clases_admin-Análisis.png*
 - *Diagrama_navegabilidad_museo.png*
 - *Diagrama_navegabilidad_admin.png*
 - *Diagrama_clases_museo-Diseño.png*
 - *Diagrama_clases_admin-Diseño.png*
 - *Diagrama_E-R.png*
- **TFG_codigo.zip** → Carpeta comprimida con todo el código fuente.

Ejemplo de como especificar los contenidos entregados

Ahora se mostrará el contenido de dicha carpeta comprimida que contiene todo el código fuente de la aplicación la cual esta dividida a su vez en dos carpetas:

AuthServerGuardMe

Contiene el código que se aloja en *Heroku* para darle funcionalidad al servidor. La clase principal es la llamada `mainAuthServer.js`.

GuardMe

Contiene el código fuente de la aplicación y se compone de las siguientes carpetas:

- **assets** -*Carpetas que contiene los elementos gráficos usados en la aplicación. Se subdivide en una carpeta llamada *images* que contiene todas las imágenes utilizadas para la construcción de la aplicación.*
- **components** -*Carpetas que contiene el código para todos los componentes creados.*
- **constants** -*Carpetas que contiene el código*
- **docs** -*Carpetas que contiene los archivos html generados por JSDoc.*
- **files** -*Carpetas en la que se encuentran los futuros archivos de Términos y Condiciones y Política de Privacidad entre otros.*
- **modules_LICENSES** -*Carpetas que contiene una por una todas las licencias de las librerías utilizadas en el desarrollo.*
- **navigation** -*Carpetas que contiene las clases relativas a la navegación de la aplicación.*
- **objects** -*Carpetas que contiene los objetos utilizados en el desarrollo que en este caso ha sido solo Fire.js.*
- **screens** -*Carpetas que contiene todas las pantallas, agrupadas a su vez en subcarpetas que identifican la pantalla sobre la que están relacionadas.*
- **styles** -*Carpetas que contiene todos los estilos de las pantallas, agrupadas a su vez en subcarpetas que siguen la misma estructura que screens.*
- **App.js** -*Clase principal y encargada de que comience la aplicación entera.*
- **LICENSE** -*Licencia sobre el código fuente.*
- **README.md** -*Archivo con la descripción del proyecto para la documentación y el repositorio de GitHub.*
- **package.json** -*Archivo que contiene las librerías utilizadas en el proyecto.*
- **app.json** -*Archivo que contiene la configuración de la aplicación.*
- **configJSDoc.json** -*Archivo de configuración para la creación de documentación por parte de JSDoc.*
- **Otros archivos** -*Los demás archivos no son relevantes ya que muchos se generan por defecto y los demás son configuraciones propias de expo.*