



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una plataforma web para la gestión y
visualización de contenidos de una galería digital

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Serra Talens, Roberto

Tutor/a: Valderas Aranda, Pedro José

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una plataforma web para la gestión de una galería digital

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Roberto Serra Taléns

Tutor: Valderas Aranda, Pedro José

Curso 2022-2023

Resum

El treball present consisteix en desenvolupar una plataforma web tant per a escriptori com per a mòbil i que serveix per a gestionar els continguts d'una galeria digital. On no només hi ha visitants que contemplen les obres exposades sinó també un portal d'empleats perquè facin aquesta gestió. Per al seu desenvolupament s'ha seguit una planificació que va des de l'anàlisi de requisits i de mercat, passant pel disseny dels components de la web ('front-end' i 'back-end') i la corresponent implementació fins a arribar a un procés de validació per a aquesta.

Paraules clau: Plataforma web, Escriptori, Mòbil, Galeria digital, Disseny, Desenvolupament web, 'front-end', 'back-end', Validació

Resumen

El trabajo presente consiste en el desarrollo de una plataforma web tanto para escritorio como para móvil y que sirve para gestionar los contenidos de una galería digital. Dónde no sólo hay visitantes que contemplan las obras expuestas sino también un portal de empleados para que realicen dicha gestión. Para su desarrollo se ha seguido una planificación que va desde el análisis de requisitos y de mercado, pasando por el diseño de los componentes de la web ('front-end' y 'back-end') y su correspondiente implementación hasta llegar a un proceso de validación para la solución lograda.

Palabras clave: Plataforma web, Escritorio, Móvil, Galería digital, Diseño, Desarrollo web, 'front-end', 'back-end', Validación

Abstract

The present work consists of the development of a web platform for both desktop and mobile and that serves to manage the contents of a digital gallery. Where there are not only visitors who contemplate the exhibited works but also a portal for employees to carry out said management. For its development, a planning has been followed that goes from the analysis of requirements and the market, going through the design of the web components ('front-end' and 'back-end') and their corresponding implementation until reaching a validation process for it.

Key words: Web platform, Desktop, Mobile, Digital gallery, Design, Web development, 'front-end', 'back-end', Validation

Índice general

Índice general	V
Índice de figuras	VII
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Estructura del documento	2
2 Estado del arte	5
2.1 Galería tradicional	5
2.2 Galería Virtual	6
2.3 Galería Digital	7
2.4 Galería propuesta	8
3 Metodología	9
4 Análisis y Especificación de Requisitos	11
4.1 Requisitos iniciales	11
4.2 Casos de Uso	12
4.2.1 Escenario 1: El empleado confecciona una exposición	13
4.2.2 Escenario 2: El empleado confecciona una obra	14
4.2.3 Escenario 3: El empleado descataloga una obra	14
4.2.4 Escenario 4: El empleado edita una obra	14
4.2.5 Escenario 5: El administrador gestiona una sustitución de empleados	14
4.2.6 Escenario 6: El visitante decide registrarse en la web	15
4.2.7 Escenario 6: El visitante reordena su espacio	15
4.3 Diagrama de clases	15
5 Diseño	17
5.1 Modelo de la BBDD	17
5.2 Bocetos de las interfaces	18
5.2.1 Vista usuario visitante	19
5.2.2 Vista usuario empleado	25
6 Desarrollo de la solución	29
6.1 Arquitectura	29
6.2 Contexto tecnológico	30
6.2.1 HTML	30
6.2.2 CSS	31
6.2.3 JavaScript	31
6.2.4 PHP	32
6.2.5 IntelliJ	32
6.2.6 Postman	33
6.2.7 XAMPP	34
6.2.8 Chrome	34
6.3 Ejemplos de código	35
6.3.1 Interfaz con HTML, CSS y Javascript	35

6.3.2	API REST y PHP	41
6.3.3	Consultas en BBDD	43
6.3.4	Despliegue	44
7	Producto desarrollado	49
7.1	Empleado artístico	49
7.2	Empleado de administración	56
7.3	Visitante	59
7.4	Interfaces móviles	63
7.4.1	Interfaz empleado artístico	64
7.4.2	Interfaz empleado administrativo	64
7.4.3	Interfaz usuario visitante	65
8	Validación	67
8.1	Método 1º: Validación mediante prueba experimental	67
8.2	Método 2º: Validación teórica	71
9	Conclusiones y trabajo futuro	75
	Bibliografía	77
A	Objetivos de Desarrollo Sostenible	79
B	Caso de estudio	81
C	API Obra	83
D	Formulario usuario visitante	87
E	Formulario usuario empleado	89

Índice de figuras

2.1	Ejemplo de galería tradicional.	5
2.2	Ejemplo de galería tradicional (2).	6
2.3	Ejemplo de galería con experiencias virtuales.	6
2.4	Ejemplo de galería con experiencias virtuales (2).	7
2.5	Ejemplo de galería digital.	7
2.6	Ejemplo de galería digital (2).	8
3.1	Esquema del modelo incremental.	9
4.1	Diagrama de casos de uso.	12
4.2	Diagrama de clases.	15
5.1	Modelo de la base de datos.	17
5.2	Prototipo de la página inicial.	19
5.3	Prototipo de la página de exposiciones.	20
5.4	Prototipo de la vista de una exposición.	21
5.5	Prototipo de la vista de información.	22
5.6	Prototipo de la vista del login.	23
5.7	Prototipo de la vista del espacio del visitante.	24
5.8	Prototipo, vista continua de las obras.	25
5.9	Prototipo, dashboard del empleado.	26
5.10	Prototipo, vista del personal a través de la web.	27
5.11	Prototipo, vista de las obras creadas.	28
6.1	Esquema de arquitectura API REST	29
6.2	Ejemplo de objeto JSON devuelto por el servidor.	30
6.3	Ejemplo visual de HTML.	30
6.4	Ejemplo de utilidad de CSS.	31
6.5	Uso de JavaScript en REST API.	31
6.6	Ejemplo de resaltado de código HTML.	32
6.7	Uso del refactor para la producción de código limpio.	33
6.8	Uso de Postman para probar el funcionamiento de la API.	33
6.9	Uso de XAMPP para desplegar los servidores necesarios.	34
6.10	Uso de las herramientas de desarrollador de chrome.	35
6.11	Ejemplo de código HTML.	36
6.12	Archivo manifest.json	37
6.13	HTML de una etiqueta <select>	38
6.14	Función JavaScript de filtrado	38
6.15	Ejemplo Document ready.	39
6.16	Ejemplo de petición AJAX.	39
6.17	Ejemplo de construcción de elementos HTML en JavaScript.	40
6.18	Almacenamiento en sesión.	40
6.19	Inserción de datos en 'session storage'.	40
6.20	Ejemplo de estilo CSS para el 'header' de la página web.	41

6.21	Métodos HTTP manejados en PHP.	42
6.22	Manejo de una petición PUT en PHP, 2.	42
6.23	Creación de cookies en PHP.	42
6.24	Conexión a BDD mediante PHP.	43
6.25	Función de autenticación en PHP.	44
6.26	XAMPP: herramienta de código abierto para entorno web.	45
6.27	XAMPP: web de gestión del servidor local.	46
7.1	Página inicial de la aplicación web.	50
7.2	Página de inicio de sesión.	50
7.3	Página de bienvenida del empleado.	51
7.4	Vista de creaciones del empleado.	51
7.5	Vista de edición de obra.	52
7.6	Vista de creaciones del empleado.	52
7.7	Vista de creación de obra.	53
7.8	Opciones de cambio entre apartados.	54
7.9	Vista de apartado de exposiciones.	54
7.10	Edición de exposición.	55
7.11	Creación de una exposición.	56
7.12	Página de inicio del empleado administrativo.	56
7.13	Vista del personal.	57
7.14	Opciones de cambio de apartados.	57
7.15	Vista del personal (2).	58
7.16	Alta de empleado.	58
7.17	Edición de un empleado.	59
7.18	Inicio de sesión visitante.	60
7.19	Registro de un visitante.	60
7.20	Página del espacio del visitante.	61
7.21	Opciones de filtrado de obras.	61
7.22	Vista de las exposiciones de la galería.	62
7.23	Vista de detalle de una exposición.	62
7.24	Vista de las obras de una exposición.	63
7.25	63
7.26	64
7.27	64
7.28	Vista del personal del administrador en móvil.	65
7.29	65
7.30	Vista del espacio del visitante en móvil.	66
8.1	Encuesta usuario visitante (1).	68
8.2	Encuesta usuario visitante (2).	68
8.3	Encuesta usuario visitante (3).	69
8.4	Encuesta usuario visitante (4).	69
8.5	Resultados de la tarea 3 del formulario empleado.	70
8.6	Pregunta sobre la ayuda recibida del usuario.	70
8.7	Pregunta de respuesta múltiple para el empleado.	71
8.8	Indicaciones contextuales de los botones de la barra de navegación.	71
8.9	Control de 'quitar' elemento.	72
8.10	Alerta de error al guardar cambios en un formulario.	72
8.11	Alerta de error al iniciar sesión con usuario no registrado.	73
8.12	Alerta de error al iniciar sesión con contraseña incorrecta.	73

CAPÍTULO 1

Introducción

Las galerías ofrecen un espacio donde disfrutar y apreciar el arte. Donde se puede no sólo contemplar, si no también aprender. Un espacio físico al que acudir y visualizar los resultados de las mentes más audaces, ingeniosas y creativas. Un lugar al que si uno no se presenta, apenas se enteraría de las intrigantes producciones visuales que puede contener. Si sólo hubiera una plataforma donde llegar a más rincones del mundo sin tener que mover dichas creaciones. Pues es aquí donde se presenta una plataforma, donde no sólo ofrece una experiencia digital para el 'visitante', si no una herramienta para que la propia galería pueda gestionar sus contenidos y que tenga un efecto inmediato sobre la misma.

Esta herramienta da el control total a la galería sobre qué exponer a sus visitantes, y éstos, no sólo la posibilidad de aprender y contemplar las obras, si no que también son capaces de 'guardar' en su espacio personal de la galería aquellas que sean sus preferidas. Herramienta creada teniendo en mente el gran empuje e influencia de las redes sociales sobre la manera de interactuar con las aplicaciones y la accesibilidad que éstas ofrecen.

Por lo tanto, este documento contempla la creación de una plataforma web para trasladar el trabajo de una galería de arte al mundo digital. Para conseguir el desarrollo de este proyecto ha sido necesario apoyarse en los conocimientos y aprendizajes recibidos del propio grado, que va desde la planificación del trabajo, pasando por crear un diseño centrado en el usuario hasta llegar a la parte más técnica como puede ser el desarrollo y la integración de los distintos componentes (web, base de datos, servidor, comunicaciones, etc).

1.1 Motivación

La decisión de construir una herramienta a través de una plataforma web viene dada por la flexibilidad que aporta este tipo de aplicación. Es decir, la web es compatible con múltiples tipos de dispositivos y de sistemas operativos, además de que el acceso a la misma no obliga a pasar previamente por una tienda o mercado de 'apps', y por tanto adaptarla según sus requisitos.

Adicionalmente, los usuarios finales pueden recibir actualizaciones y mejoras sin que éstos tengan que solicitarlo, pues nada más acceder al servicio ya cargan la última versión que ofrece el servidor.

Finalmente, una característica a favor de crear la aplicación como una plataforma web es su facilidad de prototopado y experimentación, permitiendo avanzar con mayor rapidez en el desarrollo de la misma.

1.2 Objetivos

El objetivo a nivel conceptual consiste en el desarrollo de una herramienta Web para una modesta galería de arte. Se tendrán en cuenta dos tipos de usuarios, los visitantes y los empleados de la galería, los cuales se encargarán de gestionar los contenidos de la web a través de la interfaz proporcionada. En cambio, los visitantes podrán visualizar obras a través de una experiencia digital y además podrán registrarse a través de la web para guardar aquellas obras que más les guste.

Así, el resultado debería dar soporte a las siguientes acciones:

- La aplicación debe permitir visualizar correctamente tanto obras como exposiciones
- La aplicación debe permitir el registro de visitantes
- La aplicación debe permitir tanto marcar como desmarcar una obra como favorita según las preferencias de usuario visitante
- La aplicación debe permitir al visitante filtrar sus obras preferidas según las opciones que se le ofrece
- La aplicación debe permitir la gestión de obras digitales
- La aplicación debe permitir la gestión de exposiciones
- La aplicación debe mostrar un acceso rápido a los elementos manejados por el empleado

Respecto al proyecto y en referencia al grado cursado, el objetivo trata de desarrollar una aplicación web con las tecnologías existentes y previamente utilizadas en el curso académico. Algunas de las materias con las que se maneja en este trabajo serían: el desarrollo web, bases de datos, desarrollo centrado en el usuario y integración de aplicaciones.

1.3 Estructura del documento

Para obtener una noción de en qué consiste este proyecto, a continuación se presentan una serie de capítulos o secciones junto a una breve descripción de lo que se realiza en cada una de ellas.

1. Estado del arte: aquí se realiza una revisión de aplicaciones o sistemas existentes similares a la propuesta, sacando el valor de la misma.
2. Metodología: donde se explica acerca de la metodología incremental junto con los sprints planteados.
3. Análisis y Especificación de Requisitos: aquí se extraen y explican los requisitos que debe soportar la aplicación, además de incluir los casos de uso y el diagrama de clases, que determinan la funcionalidad y elementos de la 'app'.
4. Diseño: tanto de la interfaz a través del prototipado, como del modelo de base de datos y su configuración.
5. Desarrollo de la solución: donde se explica la arquitectura de la aplicación, su contexto tecnológico y ejemplos de cómo se ha desarrollado.

6. Producto desarrollado: aquí se presenta la aplicación de forma visual permitiendo observar qué resultado se ha obtenido.
7. Validación: el tipo de validación que se ha utilizado y cómo se ha llevado a cabo.
8. Conclusiones y trabajo futuro: resumen del trabajo junto a una opinión, además se identifican una serie de mejoras o extensiones que se podrían aplicar.

CAPÍTULO 2

Estado del arte

Antes de comenzar con cualquier tarea es interesante ver qué servicios similares se ofrecen a través de Internet, para así ver los elementos a imitar, a evitar y también el aporte de valor de la propuesta de este trabajo. Principalmente se pueden dividir en tres clases: una de ámbito tradicional, la otra con unas características más virtuales y finalmente de ámbito puramente digital.

2.1 Galería tradicional

Un ejemplo resultaría ser 'Galería Benlliure', con actividad física pero que también muestra a través de su web las exposiciones que se producirán o se han producido. Como se puede observar en las figuras 2.1 y 2.2, la web no parece ofrecer una experiencia digital suficiente, pues muestra imágenes de las obras con poco cuidado, además de ser escasas en cuanto a número de obras. Otro elemento que se observa es que enseguida tratan de advertirte de que acudas al espacio físico para no perder detalle de su arte.

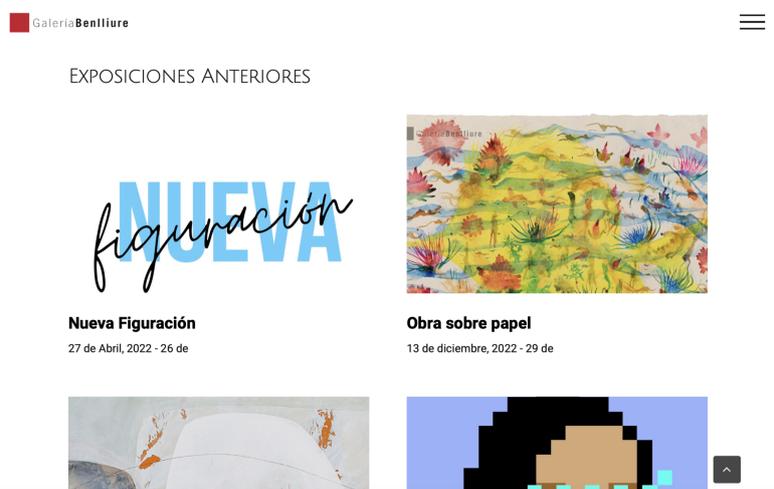


Figura 2.1: Ejemplo de galería tradicional.

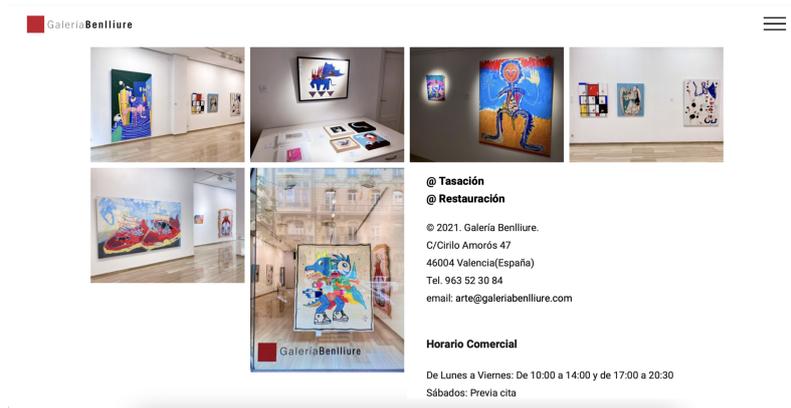


Figura 2.2: Ejemplo de galería tradicional (2).

Con estas razones se puede llegar a entender que esta aplicación web no está a la altura del usuario y sus necesidades, el cual ya está acostumbrado a acceder a sus servicios desde donde quiera y cuando quiera, teniendo al alcance de la mano una variedad de posibilidades.

2.2 Galería Virtual

Por otro lado, como galería virtual se encuentra 'Gozar Magazine', que trata de llevar la experiencia física de un visitante a una virtual, mostrando un espacio casi tridimensional junto con las obras a exponer. La propuesta de esta revista, aunque original, no ofrece al espectador un rendimiento y control suficientes como para que el mismo sienta la necesidad de volver a la web. Esto es debido a los controles poco intuitivos y a la manera limitada de observar las imágenes de las obras.

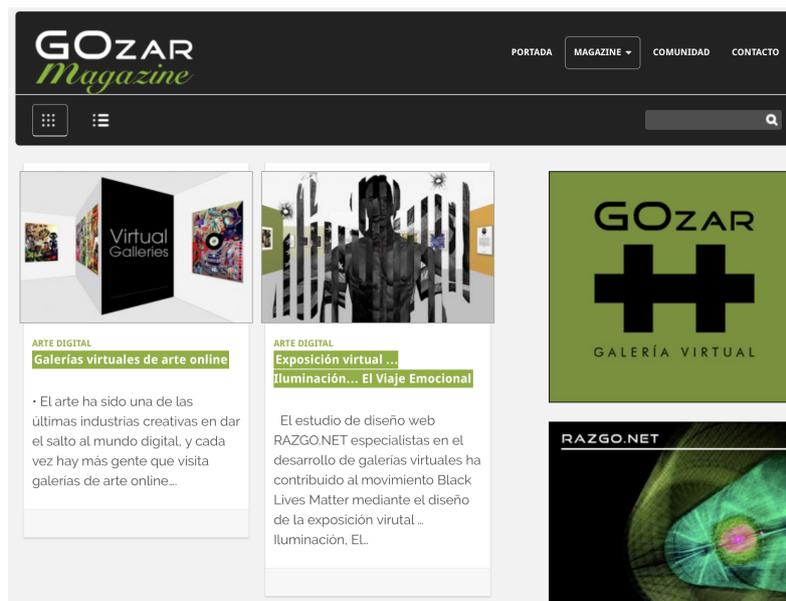


Figura 2.3: Ejemplo de galería con experiencias virtuales.

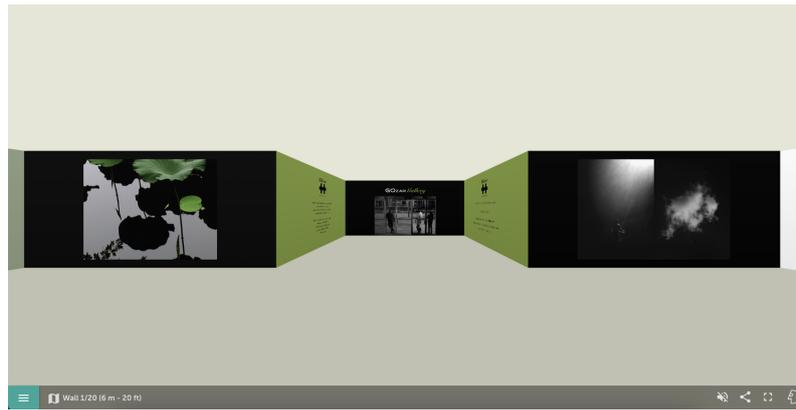


Figura 2.4: Ejemplo de galería con experiencias virtuales (2).

2.3 Galería Digital

Finalmente, esta última galería ofrece la posibilidad de explorar las exposiciones y ver las obras que contienen de una manera más estándar y cómoda respecto a las anteriores, cuidando un poco más la manera de mostrar las obras. Este tipo de galerías funcionan únicamente a modo de expositor, llegando a tener un fin comercial en algunos casos, pero que no presenta ninguna funcionalidad extra al usuario o potencial cliente.



¿Buscas un espacio donde presentar tu trabajo? **ARTEVIVO** es la respuesta

Figura 2.5: Ejemplo de galería digital.

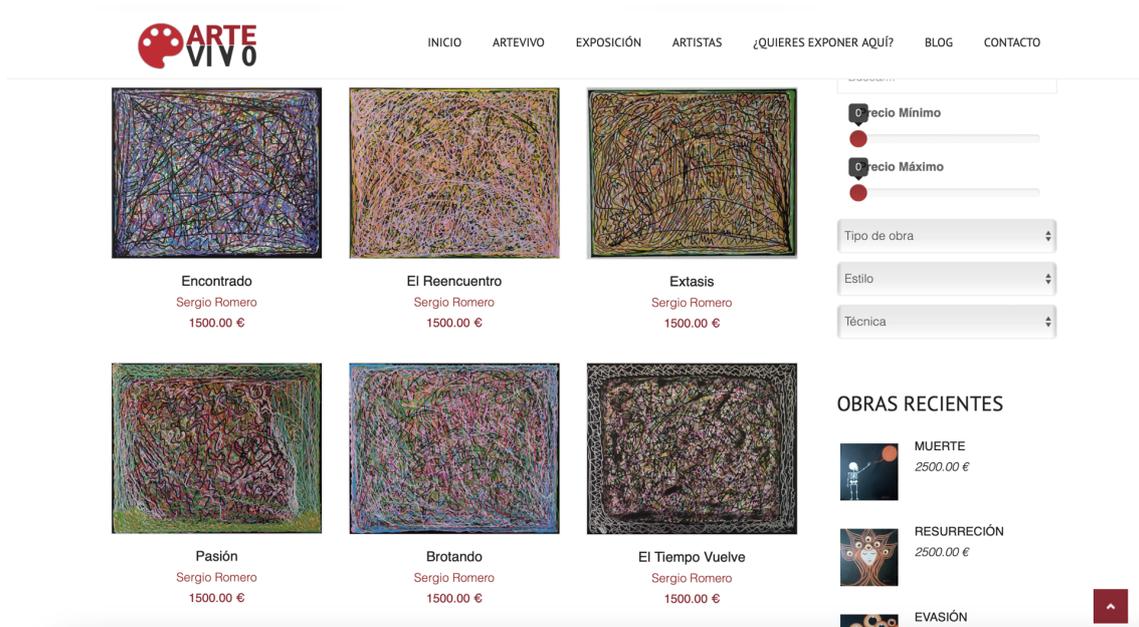


Figura 2.6: Ejemplo de galería digital (2).

2.4 Galería propuesta

Sin embargo, la aplicación propuesta y como se ha mencionado anteriormente permite a los visitantes no sólo visualizar las obras sino que también interactuar con ellas teniendo la posibilidad de guardarlas como favoritas para así contemplarlas posteriormente. Además la web contiene un portal de empleados que les permite, tras iniciar sesión, gestionar los contenidos de la propia galería digital y sin necesidad de poseer conocimientos de desarrollo web. A resumidas cuentas, la aplicación se centra en el usuario.

Más adelante, en el capítulo 7 se mostrarán ejemplos en forma de captura que dan a entender el funcionamiento de la solución propuesta y los aspectos clave comentados.

CAPÍTULO 3

Metodología

En el planteamiento del proyecto se ha utilizado una metodología incremental [1], donde se identifica una parte de la funcionalidad, se desarrolla y se validan los resultados en cada incremento. Salvo una mejora no contemplada previamente o algún error encontrado posteriormente, no se modifica el trabajo realizado en cada incremento.

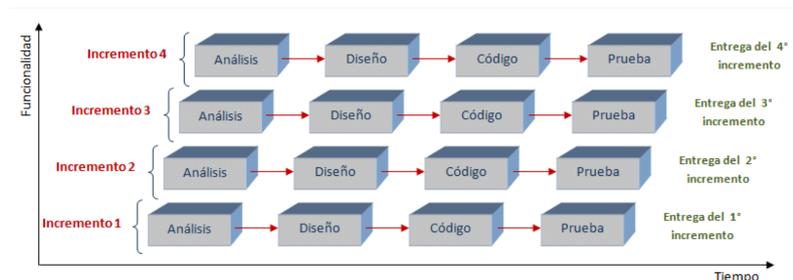


Figura 3.1: Esquema del modelo incremental.

En el caso de las funcionalidades, éstas se han determinado siguiendo el diagrama de casos de uso junto a los requisitos. Y gracias a esta metodología es posible validar la funcionalidad sin necesidad de esperar a que el sistema quede terminado.

Un ejemplo de ello sería el incremento a la hora de crear y poner en marcha la base de datos (BBDD), en este caso relacional. Para ello habría que basarse en el diagrama de clases y los casos de uso, incrementos anteriores, para determinar aquellas relaciones y datos de las clases y objetos a persistir. Más adelante se crearía el código SQL que construiría la base de datos junto con sus tablas, relaciones y restricciones, todo ello dentro de un mismo servidor. Finalmente se validaría la creación de la BBDD con pruebas y datos realistas para comprobar que las consultas y actualizaciones en la misma son correctas.

Con este incremento finalizado, se puede avanzar con otro creando por ejemplo una primera vista web que consulta en la base de datos las obras catalogadas y mostrarlas al usuario, para que éste sepa qué contenido presenta la galería. Este nuevo incremento también se apoyaría en otros como son el de la especificación de requisitos, los casos de uso y el prototipado de la interfaz de usuario.

Comentar que en posteriores incrementos no será necesario cambiar el estado de la base de datos para adaptarse a nuevas funcionalidades o casos de uso, puesto que en su respectivo incremento ya se habrá tenido en cuenta a qué funcionalidades y requisitos va a satisfacer.

Por tanto, los incrementos realizados con una vista general serían:

1. Caso de estudio y especificación de requisitos: donde se determina a nivel historia de usuario qué debe permitir la aplicación hacer al usuario.
2. Casos de uso: aquí se contemplan los escenarios y acciones de los usuarios.
3. Diagrama de clases: donde se identifican los objetos con los que el sistema y el usuario manejarán e interactuarán.
4. Diseño y prototipado de las interfaces de usuario: donde se trata de identificar aquellos elementos visuales más apropiados para los distintos usuarios de la aplicación.
5. Diseño de la Base de de Datos: aquí se determina aquella información que el sistema deberá persistir y para ello se muestran las características de la base de datos y también de los objetos a guardar, como tablas de objetos y las relaciones entre éstos.
6. Desarrollo e implementación de la solución: aquí se crea el código necesario para soportar los casos de uso o escenarios determinados previamente de forma secuencial, además de la creación de las interfaces de usuario, y con pruebas de validación para cada uno de esos procesos.
7. Validación y calidad: donde se conforma y ejecuta la validación por medio de pruebas, cuestionarios y comentarios acerca de el uso de la aplicación para así determinar la calidad de la misma.

CAPÍTULO 4

Análisis y Especificación de Requisitos

4.1 Requisitos iniciales

Para extraer los requisitos de la aplicación web es necesario leer el caso de estudio que se encuentra en el apéndice B. Tras ello se han determinado una serie de requisitos para los distintos usuarios de la plataforma.

En cuanto a los empleados artísticos de la galería, que gestionarán sus contenidos pudiendo hacer las siguientes acciones:

- El sistema debe permitir catalogar una exposición (o varias): indicando toda la información necesaria junto con las obras catalogadas que incluirá .
- El sistema debe permitir catalogar una obra (o varias).
- El sistema debe permitir modificar una exposición u obra: editando alguna de sus propiedades u obras.
- El sistema debe permitir catalogar y descatalogar obras y exposiciones: seleccionando el elemento y confirmando la acción deseada.
- El sistema debe permitir recuperar elementos descatalogados.

Los empleados administrativos gestionarán la plantilla de la galería pudiendo realizar las siguientes acciones:

- El sistema debe permitir dar de alta a un nuevo empleado (de administración o de artes).
- El sistema debe permitir dar de baja a un empleado ya existente.
- El sistema debe permitir editar información de un empleado (nombre, telf, nº cuenta, salario...).

Y finalmente el usuario visitante podrá realizar las siguientes acciones:

- El sistema debe permitir la correcta visualización de las obras y exposiciones.
- El sistema debe permitir registrar o iniciar la sesión de un usuario.
- El sistema debe permitir asignar o desmarcar una o varias obras como favoritas.

- El sistema debe permitir visualizar las obras favoritas.
- El sistema debe permitir filtrar la búsqueda de sus obras favoritas.

4.2 Casos de Uso

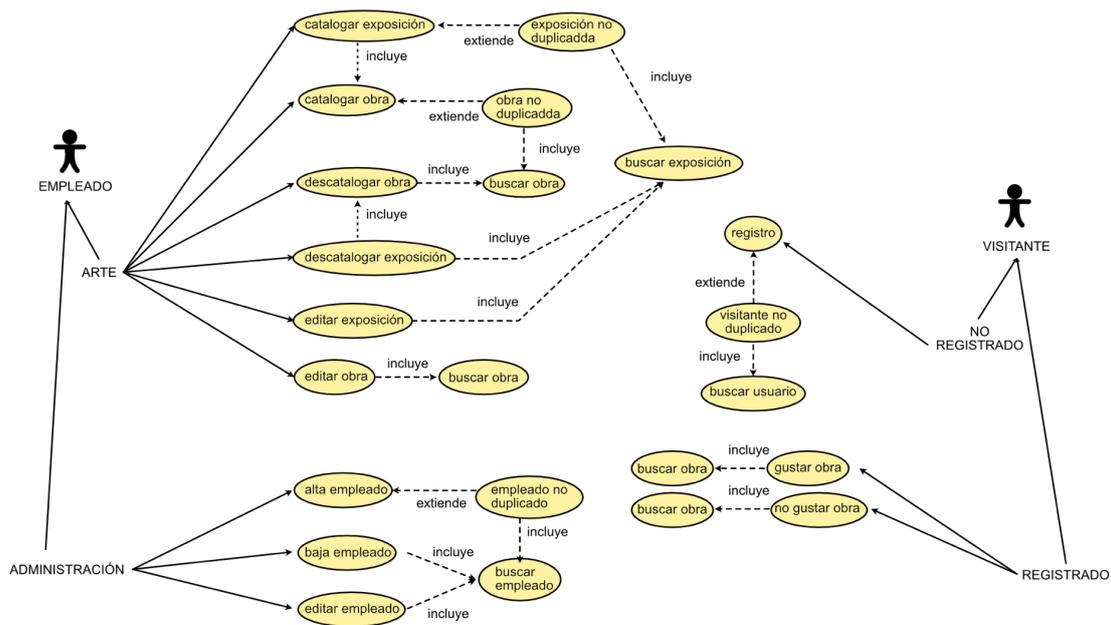


Figura 4.1: Diagrama de casos de uso.

Empezando con el usuario empleado artístico y como se ve en la figura 4.1 podrá catalogar una exposición en el caso de que la exposición no se haya creado anteriormente, lo que requiere al sistema de su consulta en BBDD. Para ello deberá indicar y completar todos los campos de información de una exposición.

Adicionalmente, como una exposición debe contener al menos una obra, el empleado deberá catalogar una de ellas también completando sus campos de información correspondientes. De nuevo, la obra creada no debe estar duplicada según su código identificador, por lo que el sistema tendrá que consultarlo en la BBDD. Anotar que el empleado puede en cualquier momento catalogar una obra, pero siempre indicando la exposición a la que pertenece, para que su publicación sea correcta.

En el caso de que el usuario no esté conforme con el estado actual de una obra o exposición podrá en cualquier instante editar la información del elemento. En cualquiera de los dos casos, el usuario deberá modificar la información deseada y el sistema guardar los cambios en caso de que no hayan códigos identificadores repetidos.

También, por los motivos que considere el empleado, éste podrá descatalogar una obra o exposición. En el primer caso, la obra simplemente no será mostrada al usuario visitante en su correspondiente exposición. Sin embargo, en el caso de descatalogar una exposición, ni ésta ni sus correspondientes obras serán mostradas a los visitantes. Por lo que el sistema deberá encontrar aquellos contenidos a descatalogar y marcarlos correspondientemente.

Pero si el empleado se arrepiente, siempre podrá recuperar las obras o exposiciones descatalogadas. Por lo que el sistema deberá encontrar y mostrar aquellos contenidos descatalogados y tras indicar el usuario el contenido a recuperar, volver a mostrarlo al usuario como elementos catalogados.

En referencia al empleado administrativo, éste podrá realizar acciones similares al artístico pero con sus correspondientes contenidos. Un escenario es cuando el usuario quiere dar de alta a un nuevo empleado, sea administrativo también o artístico. Para ello deberá incluir todos los datos necesarios y se podrá crear el nuevo empleado en caso de que el nombre de usuario no coincida con el de algún otro, por lo que el sistema deberá realizar la búsqueda y comprobación correspondiente.

A su vez, este tipo de empleado podrá dar de baja a cualquier otro indicándoselo al sistema, el cual realizará una búsqueda del mismo y lo eliminará.

Y por último, un administrador podrá editar en cualquier momento cualquier tipo de información de los empleados, guardando los cambios el sistema en el caso de que no se repitan los nombres de usuario. En este caso el sistema deberá previamente consultar y mostrar todos los empleados registrados en el mismo.

Acabando con los usuarios, el visitante podrá visualizar los contenidos de la web una vez el sistema se los presente, por lo que deberá consultar toda la información correspondiente en BBDD.

En el caso de que el visitante quiera guardarse obras como favoritas, deberá primeramente registrarse en el sistema indicando su nombre de usuario y su contraseña. El sistema guardará la información en caso de que el nombre de usuario no se haya guardado anteriormente.

Más adelante, para que el usuario pueda acceder a sus obras favoritas, éste deberá iniciar sesión con sus credenciales, las cuales el sistema deberá comprobar realizando una búsqueda. En el caso de que coincidan las credenciales con las de un usuario registrado, el sistema iniciará la sesión del usuario y le mostrará sus obras preferidas.

Otro escenario que contempla este usuario es el de poder filtrar la búsqueda o visualización de sus obras favoritas según las opciones ofrecidas. En este caso el sistema deberá consultar y filtrar los resultados según las opciones de filtro indicadas, y así mostrárselas al visitante.

El último escenario del visitante resultaría ser el de poder marcar o desmarcar una o varias obras como favoritas. El sistema deberá persistir los cambios solicitados y posteriormente mostrárselos con alguna indicación al usuario.

A continuación se presentan una serie de escenarios los cuales contienen los casos de uso ya comentados y que reflejan de manera ejemplificadora las acciones que pueden realizar los usuarios.

4.2.1. Escenario 1: El empleado confecciona una exposición

Paco, el humilde trabajador de la galería digital ha de preparar una nueva exposición, que trata de aquellas cosas que hacen sentirse a gusto a las personas cuando están de viaje de vacaciones y que se representa a través de fotografías de lugares u elementos que evocan esas buenas sensaciones y recuerdos a las personas. Por lo tanto, Paco iniciará su sesión como empleado y se dirigirá a la sección de creaciones donde se encuentran las exposiciones ya publicadas. En esa página se le dará la opción de crear una exposición y donde deberá indicar las características de la exposición, como por ejemplo de temática vacacional o con una breve descripción como la que se ha comentado anteriormente, y también deberá crear por tanto las obras de la exposición, que corresponderán a las

fotografías, e indicar también sus propiedades. Por tanto, el empleado catalogará tanto una exposición como una o varias obras.

4.2.2. Escenario 2: El empleado confecciona una obra

A Paco le han asignado la tarea de gestionar el contenido de la exposición 'Lo mejor de 2023', por lo que ya existen obras dignas de aparecer en esta exposición y otras que llegarán a lo largo del año, lo que ocurre justamente con la nueva obra del autor contemporáneo Don Alfredo: 'Lágrimas de amor', que hace referencia a la cultura del esfuerzo y al entendimiento de uno mismo como herramientas para el desarrollo personal, según la opinión de los expertos. Claramente la galería digital no se quiere perder la oportunidad de mostrar esta gran obra en su exposición anual y por tanto Paco deberá catalogar dicha obra con sus correspondientes detalles, todo ello siguiendo las mismas acciones que con la exposición pero en la sección de obras.

4.2.3. Escenario 3: El empleado descataloga una obra

Paco sigue trabajando en 'Lo mejor de 2023', cuando se da cuenta, navegando por internet y curioseando sobre los procesos creativos de las obras de grandes autores contemporáneos, que una de ellas es muy similar, sino idéntica, a una de las que contiene la exposición. Por lo que decide contactar con el autor y comunicarle lo sucedido. Tras una exhaustiva revisión, Paco confirma lo último que quería oír: ha habido un caso de plagio de obras. Por tanto, con rapidez se dirige a la sección de creaciones y donde se encuentran las obras selecciona la opción de descatalogar obra para hacer justicia y así no engañar al espectador final.

4.2.4. Escenario 4: El empleado edita una obra

Como se ha comentado antes, 'Lo mejor de 2023' aún no está acabada, pues apenas ha comenzado el año y queda mucho todavía por hacer. De nuevo, nos encontramos a Paco, que acaba de recuperarse de un resfriado e intuye que se le viene el trabajo encima. Efectivamente, se pone manos a la obra y comienza a revisar lo poco que pudo adelantar en los días que se encontraba mal para poder así retomar lo que dejó. Entonces se da cuenta de un despiste que tuvo al colocar la fecha en la obra 'Lágrimas de amor', donde indicó que la obra había sido creada en el año anterior, algo que suele ocurrir en éstas fechas tempranas del año pero que ha de corregirse. Por ello, Paco, una vez encontrada dicha obra, selecciona la opción de editar obra y corrige su error indicando la fecha correcta y guardando los cambios en el sistema.

4.2.5. Escenario 5: El administrador gestiona una sustitución de empleados

Ahora se encuentra Maria del Carmen con la tarea de dar de alta a un empleado y por lo tanto dar de baja a otro. Tras haber cumplido y excedido con su trabajo, Paco decide jubilarse para dedicarse a otros proyectos de vida, sin dejar de lado todo aquello que ha aprendido en la galería. Ahora, Carmen deberá, una vez encontrado a Paco como empleado artístico en la sección de personal en la web, seleccionar la opción de eliminar empleado, efectuándose por lo tanto la baja. Por otro lado se encuentra Pablo, un entusiasta del arte y gran seguidor de la Galería digital, que ha decidido presentarse a trabajar como empleado artístico en la misma. Tras una entrevista, la cual ha ido muy bien, Carmen deberá solicitar la información necesaria para poder rellenar el alta de Pablo como nuevo empleado de la galería. Una vez obtenida, tendrá que seleccionar la opción de

crear empleado de la sección de personal e indicarla en los campos correspondientes. Cuando Carmen guarde los cambios, el alta se habrá efectuado.

4.2.6. Escenario 6: El visitante decide registrarse en la web

Un tiempo atrás, Pablo, que apenas conocía el mundo del arte, se encontraba investigando acerca de galerías a las que podía asistir o acceder para aprender más del mundo. Una de las webs con las que se topó es 'Galería digital' y, tras un rato husmeando sus contenidos, decidió registrarse en la web ya que le había gustado lo que vió. Para ello, debió acceder a la pestaña 'Mi espacio' y tras mostrarse dos opciones según el usuario (empleado o visitante) indicar que es visitante y seleccionar la opción de registrarse. En ese momento Pablo rellenaría el nombre de usuario y la contraseña con la que utilizaría en la web.

4.2.7. Escenario 6: El visitante reordena su espacio

A Pablo, de vez en cuando, le gusta volver a visitar sus obras favoritas de la galería en su espacio, pero como ya hacía tiempo que no volvía a esta sección de la web, decidió echarle un vistazo. Mientras filtraba las obras según las opciones que se le ofrecía, vió que una obra ya no resonaba tanto con él mismo, por lo que decidió desmarcarla como favorita seleccionando el botón correspondiente. Más tarde, Pablo se encontraba merodeando por la sección de exposiciones de la web, donde encontró una que le interesó tanto que marcó todas las obras que contenía como favoritas seleccionando para cada una de las obras el botón correspondiente. Este escenario contempla los casos de uso de 'gustar obra' y 'no gustar obra'.

4.3 Diagrama de clases

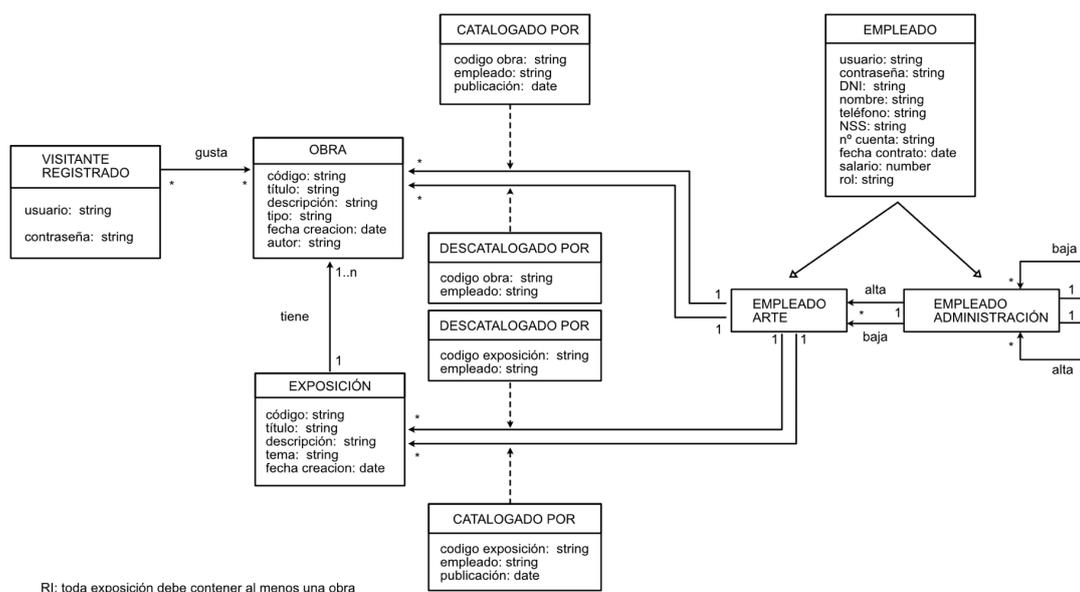


Figura 4.2: Diagrama de clases.

En la figura 4.2 se presenta el diagrama de clases que dan soporte a los requisitos y que, a través del lenguaje UML [2], se puede describir el sistema. Este lenguaje permite representar la funcionalidad del sistema y la interactividad entre los elementos de la misma haciendo uso de clases y sus relaciones.

Como se puede observar en la imagen anterior son necesarias las clases de empleado, más concretamente de arte, obra y exposición para poder completar todos los requerimientos que permiten todas las acciones de este tipo de usuario. Además, debido a la restricción que se encuentra también en la figura anterior, la clase obra se relaciona con la clase exposición. Finalmente las acciones que ejecuta el empleado artístico sobre las obras o exposiciones se determinan como clases asociación, porque contienen información de ambas clases para formar una propia.

En cuanto al empleado administrativo, éste se encarga de manejar el personal de la galería, por lo que las clases necesarias que dan soporte a sus acciones definidas en los requisitos son el empleado artístico y el empleado de administración. Debido a los requisitos especificados anteriormente, no es necesario establecer nuevas relaciones o crear nuevas clases para que este usuario pueda efectuar correctamente sus acciones.

Finalmente se encuentra el usuario visitante y para que pueda realizar sus acciones son necesarias las clases visitante, obra y exposición. Así, este usuario es capaz de visualizar los contenidos de las obras y como también indica la relación 'gusta', podrá indicar como favorita la obra deseada.

CAPÍTULO 5

Diseño

En este capítulo, debido a que no se ha seguido una metodología estrictamente de diseño centrado en el usuario, no será posible ver con todo lujo de detalle los procesos más técnicos de esta rama del conocimiento. Sin embargo sí se utilizan algunos conceptos aprendidos de esta materia. Adicionalmente también se trata de explicar el diseño del modelo de la base de datos que servirá de información al sistema y que permitirá la correcta interacción y uso del sistema por parte del usuario objetivo. Cabe comentar que el prototipado de las interfaces permite adaptar su vista a la de una versión móvil, como se verá en el capítulo 7.

5.1 Modelo de la BBDD

ACTUALIZAR LA IMAGEN

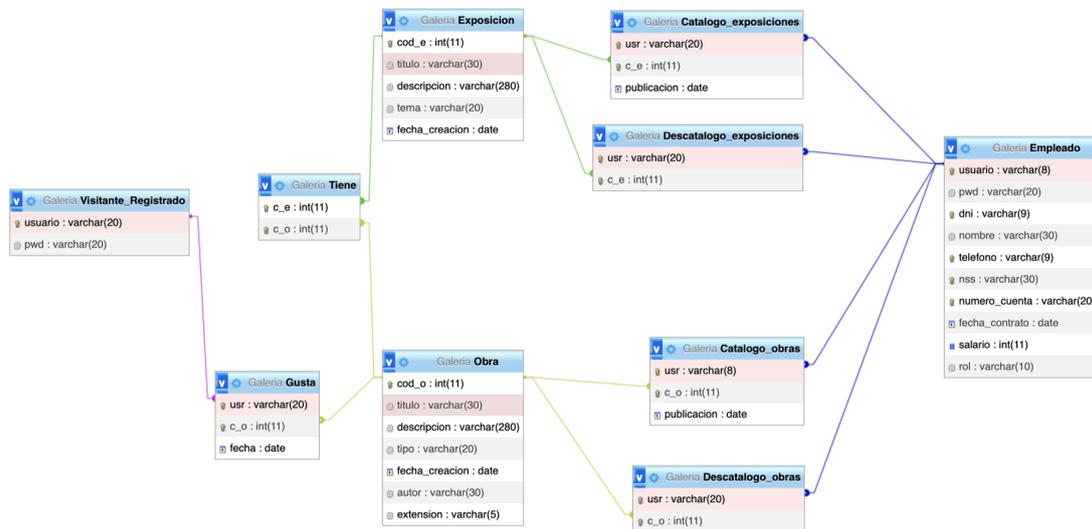


Figura 5.1: Modelo de la base de datos.

La base de datos utilizada en la aplicación es de tipo relacional e implementada en SQL. Sin embargo, antes de comenzar a construirla hay que plantearse a qué funcionalidad servirá como también que información es clave que persista.

Por lo tanto, el modelo de la BBDD se ha apoyado del caso de estudio y del diagrama de clases para poder determinar los datos concretos a guardar y también todas las

relaciones que se dan entre las clases y así guardar nueva información generada de estas relaciones, potencialmente útiles para el funcionamiento de la aplicación.

En el caso del usuario visitante, sobre éste sólo es necesario guardar su nombre de usuario y contraseña, una vez registrado. No es necesario ningún dato más pues los casos de uso ya se completarían con esta información.

Por otro lado, se encuentran las obras y exposiciones, de las cuales, éstas últimas contienen a las primeras, por ello se guarda la relación 'Tiene' en BBDD. En cuanto a los datos propios de obras y exposiciones, se trata de información sobre las propiedades o características de las mismas.

Comentar que en el caso de uso 'gustar obra' es necesario guardar esta relación de obra y visitante para que posteriormente el sistema pueda mostrar las obras de un visitante marcadas como favoritas, de ahí la tabla 'Gusta'.

Acabando con el usuario empleado, éste puede tener la función de entre dos roles distintos: administración y arte. Sólo se ha incluido una tabla correspondiente con el objeto 'empleado' debido a que con el campo 'rol' de la tabla 'Empleado' ya se pueden acometer los casos de uso de ambos roles. En cuanto al administrador, como éste maneja única y exclusivamente empleados, sólo será necesario persistir en base de datos este objeto.

Finalmente se encuentra el empleado artístico, el cual realizará como casos de uso catalogar o descatalogar una obra o exposición. Aquí es interesante guardar las relaciones entre el empleado y estos objetos pues es necesario para satisfacer los requisitos en los cuales el empleado descataloga una obra o exposición y también cuando se le permite recuperar una obra o exposición descatalogada. Como se puede observar en las tablas de 'catálogo' de la figura 5.1, se le añade un campo 'fecha' que será útil para el sistema que le servirá para mostrarle al usuario de forma ordenada los contenidos de la galería.

5.2 Bocetos de las interfaces

Aunque no se ha hecho una investigación profunda de los usuarios de la plataforma, sí es posible deducir algunos aspectos de los usuarios después de analizar los requisitos funcionales de los mismos.

En cuanto al visitante, su perfil se conformaría de la siguiente manera: es un usuario 'general' que estaría centrado en un público objetivo más joven, que serán los futuros y principales usuarios de la app. Sin embargo, las obras pueden ser de tipología muy variada, esto es porque cualquier persona a través de la subjetividad puede disfrutar del arte y así contentar al mayor número de personas.

Por ello se ha querido construir una interfaz cercana y que resuene con las redes sociales más populares de la actualidad. Para este caso será necesario incluir elementos familiares a este tipo de plataformas. Es decir, se busca un perfil tecnológico y digital para contentar a los usuarios jóvenes pero de contenido diverso para no dejar de lado al resto del público.

Comprendiendo las tareas y función de los empleados, se puede extraer el siguiente perfil: es un usuario general y serio, pues se trata de un puesto de trabajo y con responsabilidad. El perfil por tanto no es tan joven por lo que también puede implicar un uso ocasional de Internet como las redes sociales.

Por ello se busca claridad y conocimiento de la información que se está manejando, dando la posibilidad de mantener actualizado al usuario en el momento que desee, pues es su herramienta de trabajo.

A continuación se presentan las interfaces que dan soporte a los escenarios en los cuales los usuarios podrán interactuar con la aplicación web.

Comenzando con la interfaz que representa la primera vista de la aplicación y que corresponde con la página inicial (5.2), se observa cómo se le muestran las obras al usuario y así tener una primera idea del contenido que puede esperar de la galería, como también las distintas acciones que puede realizar para navegar por la aplicación.



Figura 5.2: Prototipo de la página inicial.

5.2.1. Vista usuario visitante

Para seguir comentando las interfaces sería mejor centrarse en algún usuario para visualizar correctamente las acciones que puede realizar y cómo las puede ejecutar. Por lo que ahora el usuario objetivo será el visitante, el cual una de las primeras acciones que se le muestra en la barra superior de navegación sería la de explorar las exposiciones con sus respectivas obras. Una vez dirigido a la página de exposiciones, el usuario puede contemplar una vista como la que muestra la imagen 5.3, donde puede ver una previsualización de las mismas junto con algunos detalles que le pueden causar interés para decidir si ver la exposición o no.

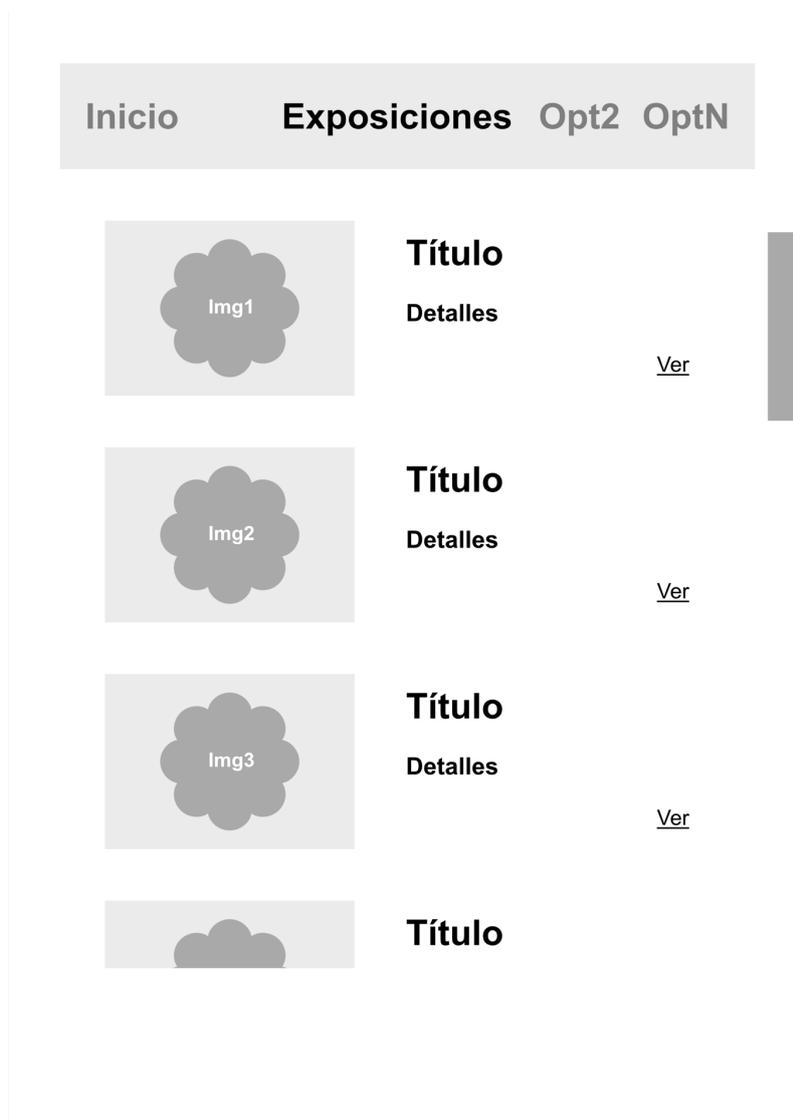


Figura 5.3: Prototipo de la página de exposiciones.

En el caso de que el visitante quiera visualizar los contenidos de una de ellas, se le mostrará una interfaz basada en la figura 5.4. Así como sucede con las exposiciones, el usuario podrá ver las obras junto con sus detalles, así como poder indicar una obra como favorita. Adicionalmente, si el usuario desea ver la obra con detalle, podrá hacerlo seleccionándola y verla a pantalla completa.

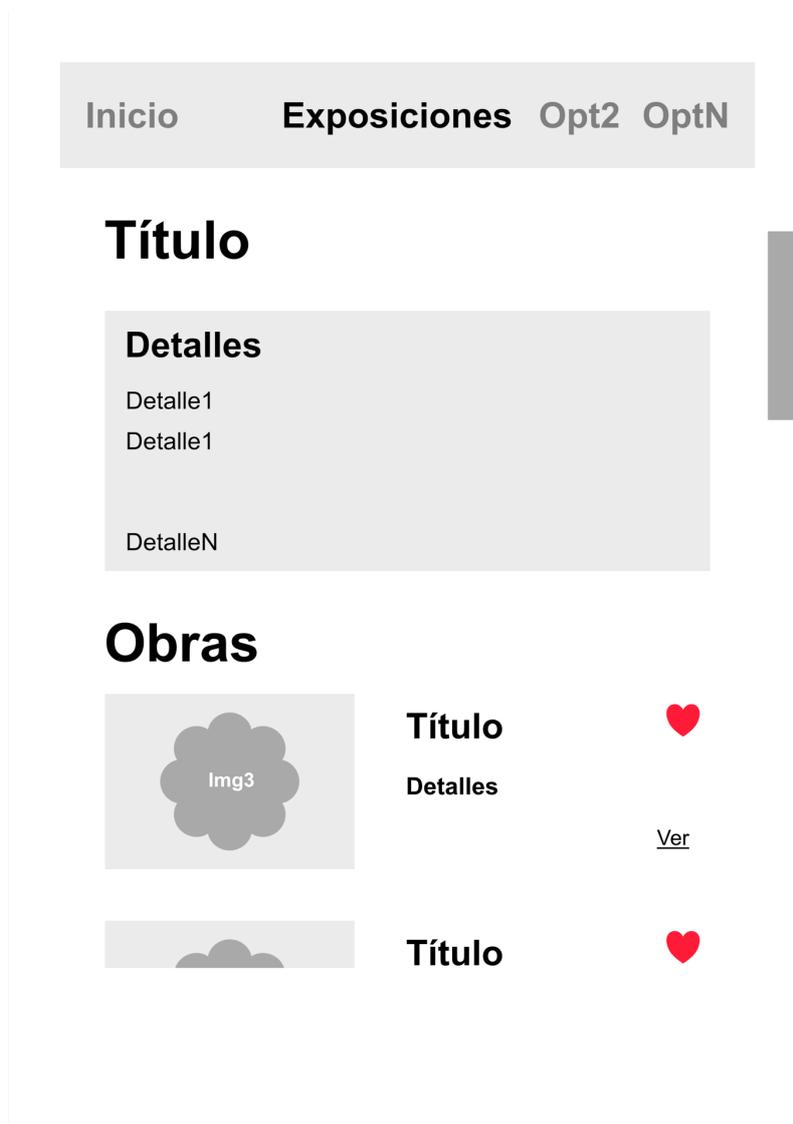


Figura 5.4: Prototipo de la vista de una exposición.

Por otro lado, aunque no aparezca en la especificación de requisitos, el usuario tendrá una vista con información acerca de la galería, como muestra la figura 5.5. Accediendo a la misma a través de la barra superior de navegación ya comentada.



Figura 5.5: Prototipo de la vista de información.

Continuando con los requisitos del usuario visitante, para que éste pueda ver las obras que ha marcado como favoritas en su espacio personal, antes deberá iniciar sesión como se puede observar en la figura 5.6 o en su defecto registrarse con una vista similar a la de inicio de sesión pero indicando al usuario que se está registrando.

Inicio Opt1 Opt2 Login

Visitante

Usuario

Contraseña

[Registrarse](#)

Empleado

Usuario

Contraseña

Figura 5.6: Prototipo de la vista del login.

Por último, una vez completado el inicio de sesión, ahora el usuario será capaz de visualizar las obras que ha indicado como preferidas (figura 5.7). Además se le mostrará la opción de desmarcar la decisión de obra favorita. Una mejora incorporada para la experiencia de usuario del visitante es la posibilidad de visualizar el contenido de manera continua o en forma de 'parrila', para así ver de un vistazo las obras (figura 5.8). Adicionalmente, el usuario podrá filtrar las obras según las opciones que se le ofrezcan, permitiéndole concretar en una serie de obras determinadas.

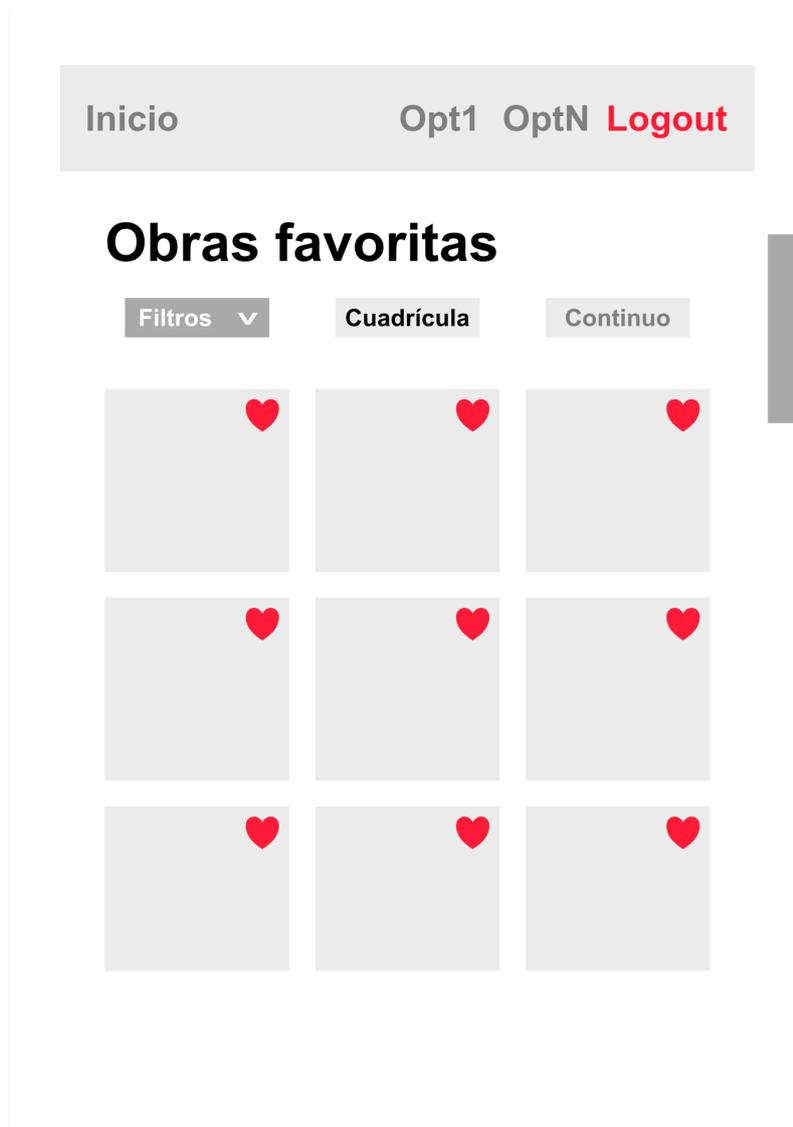


Figura 5.7: Prototipo de la vista del espacio del visitante.

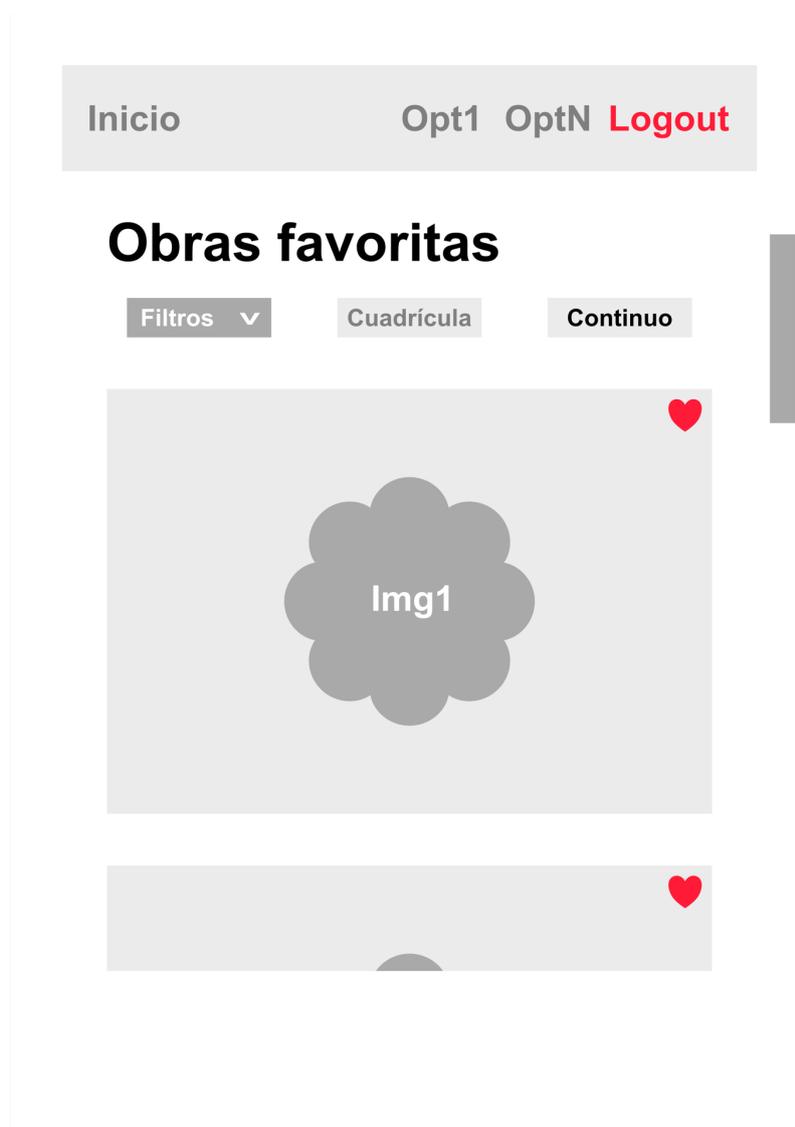


Figura 5.8: Prototipo, vista continua de las obras.

5.2.2. Vista usuario empleado

En primera instancia, comentar que el usuario en este caso deberá acceder a su sección en la aplicación a través del inicio de sesión (figura 5.6), por lo que se encontrará con una interfaz y experiencia prácticamente idéntica al usuario visitante.

Una vez autenticado, el usuario recibirá una primera vista (figura 5.9) que consiste en un 'dashboard' con información del propio usuario, una estadística de su rendimiento de trabajo respecto al mes anterior y un conjunto de elementos recientes el cual el usuario ha manejado.



Figura 5.9: Prototipo, dashboard del empleado.

Como se puede observar en esta primera vista, la barra superior de navegación ha cambiado para adaptarse a las nuevas vistas o acciones que puede realizar el usuario. Ahí encontrará la posibilidad de cerrar la sesión y también un acceso directo a la vista de edición de los elementos con los que el empleado trabajará. En caso del empleado administrativo, éste verá un conjunto de empleados, y para el empleado artístico las creaciones, sean obras o exposiciones.

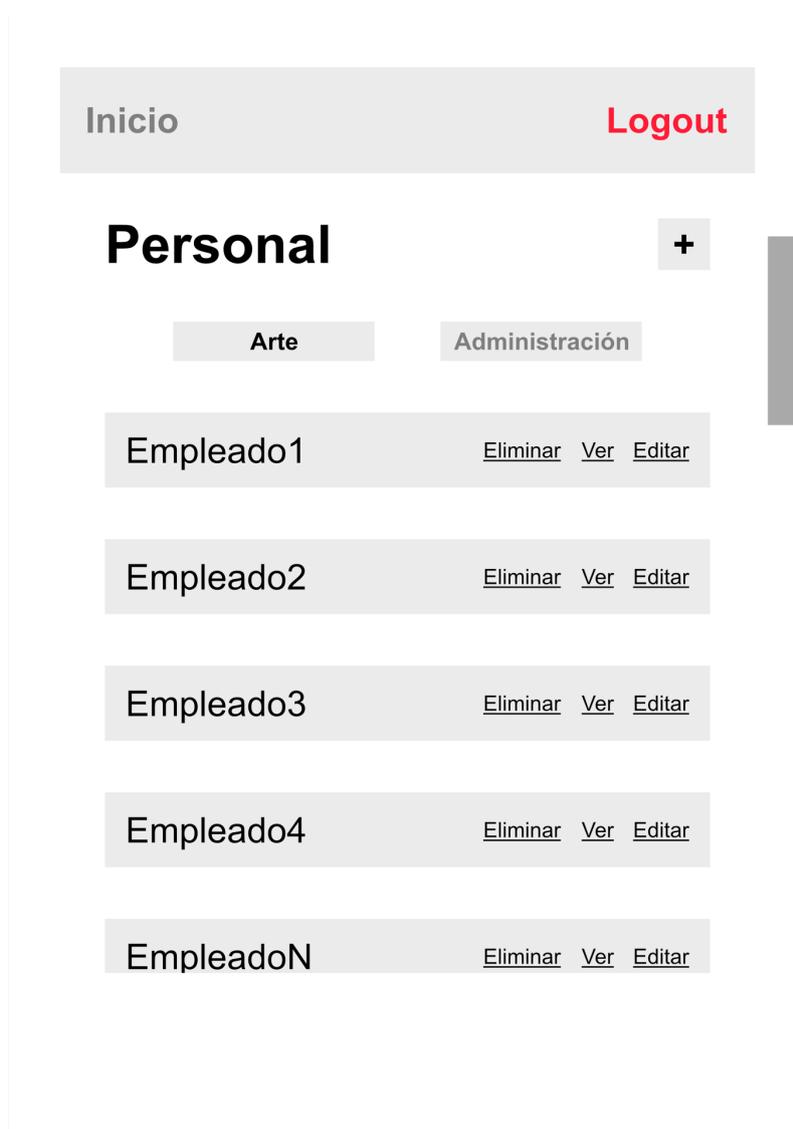


Figura 5.10: Prototipo, vista del personal a través de la web.

Continuando con la vista de edición del administrativo, que se corresponde con el nombre "Personal", se puede observar un conjunto de empleados en forma de lista (figura 5.10). A su vez, para cada empleado de esa lista el usuario podrá editar, ver en detalle y eliminar. En caso de añadir un nuevo empleado, el usuario lo podrá realizar a través del acceso directo en la zona superior derecha. Por último, en la zona superior de la sección de personal el usuario tiene la posibilidad de cambiar la vista entre empleados artísticos y empleados administrativos.

Finalmente, siguiendo con el caso del empleado artístico, éste tendrá una serie de dinámicas similares al empleado administrativo. Sin embargo los contenidos que manejará serán distintos. Su vista de edición ahora se corresponde con 'Creaciones', que muestra una lista de elementos divididos entre obras y exposiciones (figura 5.11). Igualmente podrá realizar las mismas acciones de edición, vista de detalles y eliminación, como también la posibilidad de crear o recuperar elementos descatalogados.



Figura 5.11: Prototipo, vista de las obras creadas.

CAPÍTULO 6

Desarrollo de la solución

6.1 Arquitectura

La arquitectura principal de la aplicación web gira alrededor de servicios REST, donde nos encontramos los siguientes componentes: el navegador que muestra la interfaz web con la que permite interactuar al usuario con la galería, el servidor web que satisface las peticiones del navegador y finalmente la base de datos, que persiste la información necesaria para mantener un correcto estado del sistema.

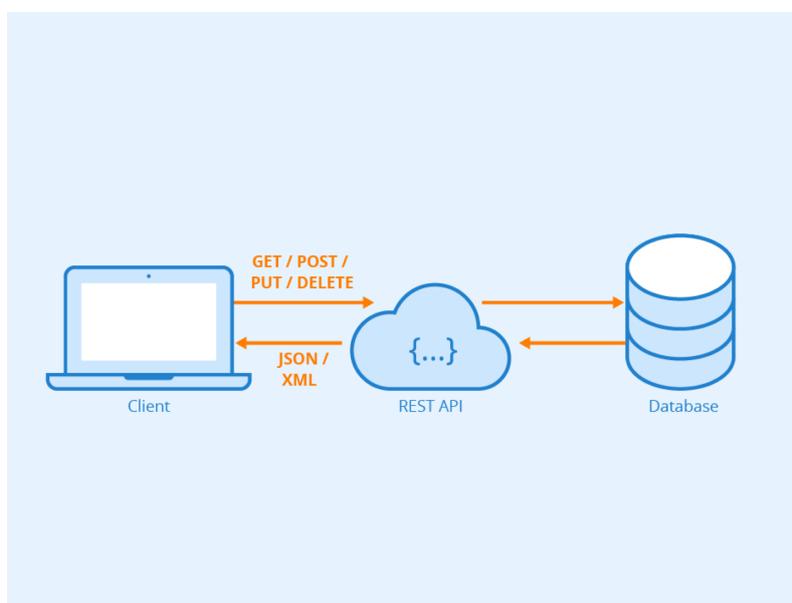


Figura 6.1: Esquema de arquitectura API REST

Por lo tanto, la comunicación entre el navegador y el servidor web se produce a través de una API, y más adelante, la comunicación entre el servidor y la base de datos se produce a través de una conexión directa mediante PHP, para así ejecutar las correspondientes consultas. En ambos extremos de la arquitectura se generan datos compuestos en formato JSON (ejemplo en figura 6.2).

```
1  {
2    "code": 1,
3    "data": [
4      {
5        "cod_o": 3,
6        "titulo": "Obra3",
7        "descripcion": "Prueba",
8        "tipo": "abstracto",
9        "fecha_creacion": "2023-03-28",
10       "autor": "test",
11       "extension": "jpg",
12       "exposicionid": 1,
13       "exposicion": "expo1"
14     }
15   ]
16 }
```

Figura 6.2: Ejemplo de objeto JSON devuelto por el servidor.

6.2 Contexto tecnológico

El desarrollo de la aplicación se ha llevado a cabo en dos partes principales: la primera de ellas consiste en la que permite al usuario interactuar con el sistema, es decir, la interfaz, y la otra parte es la que el usuario final nunca ve, es decir, todos los procesos que ocurren 'detrás' y que sirven para servir las peticiones del usuario que genera.

A continuación se presentarán las tecnologías empleadas para este trabajo dividiéndolas en dos clases: lenguajes de programación y herramientas de desarrollo.

6.2.1. HTML

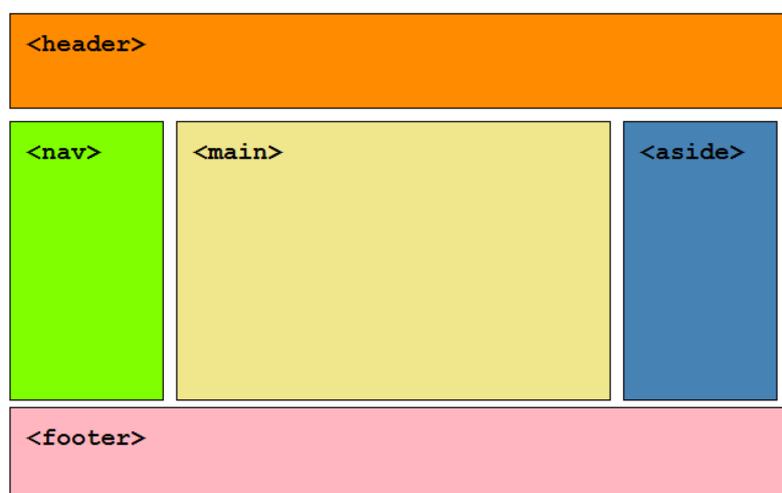


Figura 6.3: Ejemplo visual de HTML.

Puesto que se ha construido una aplicación web, es esencial este lenguaje para su desarrollo. Como se observa en la figura 6.3, este lenguaje se utiliza para otorgar de contenido y estructura a la página web. Otra de las razones de utilizar esta tecnología, ya

comentada anteriormente, es la compatibilidad entre plataformas y navegadores, permitiendo al usuario navegar por Internet sin importar el dispositivo o sistema operativo que esté utilizando. Además HTML es compatible con otras tecnologías web como PHP, JavaScript y CSS, entre otros, que se verán a continuación.

6.2.2. CSS

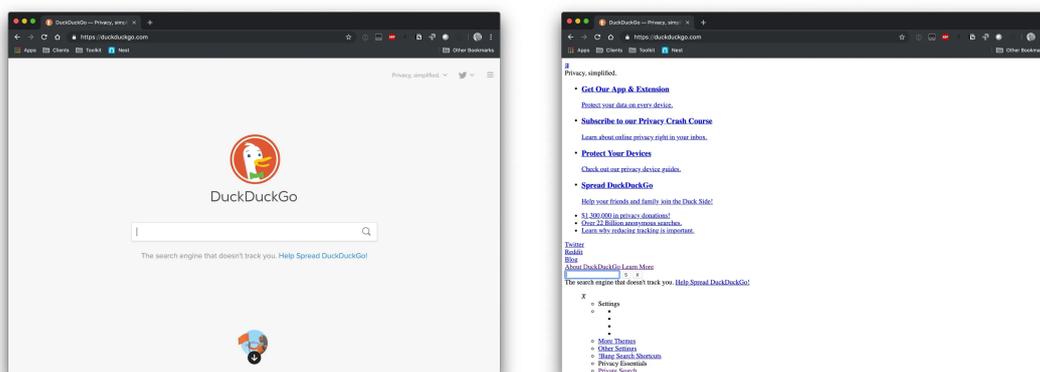


Figura 6.4: Ejemplo de utilidad de CSS.

Si bien puede ser suficiente mostrarle directamente al usuario el contenido, no lo sería en el caso de crear una interfaz centrada en el mismo, como ocurre en esta propuesta. Por lo tanto, ésta tecnología permitirá presentarle de una manera más cómoda e intuitiva al usuario la página web. Un ejemplo se puede observar en la figura 6.4. Otros usos de CSS que permiten lograr estos objetivos son: la posibilidad de crear una interfaz 'responsive', pudiendo adaptar la web en función del tamaño de la ventana o dispositivo y finalmente animaciones y transiciones, para otorgar interactividad e interés visual al usuario.

6.2.3. JavaScript

Lenguaje para la interactividad del usuario con la interfaz y que sus acciones tengan efecto visual y práctico. Por ejemplo, este lenguaje se utiliza para: la validación de formularios cuando el usuario inicia sesión o para la carga dinámica de contenido, como por ejemplo cuando el empleado crea una nueva obra que aparecerá posteriormente en la lista de obras catalogadas. Otro de los usos de este lenguaje para la usabilidad de la interfaz es manejar objetos DOM y actualizar su contenido o propiedades de manera directa.

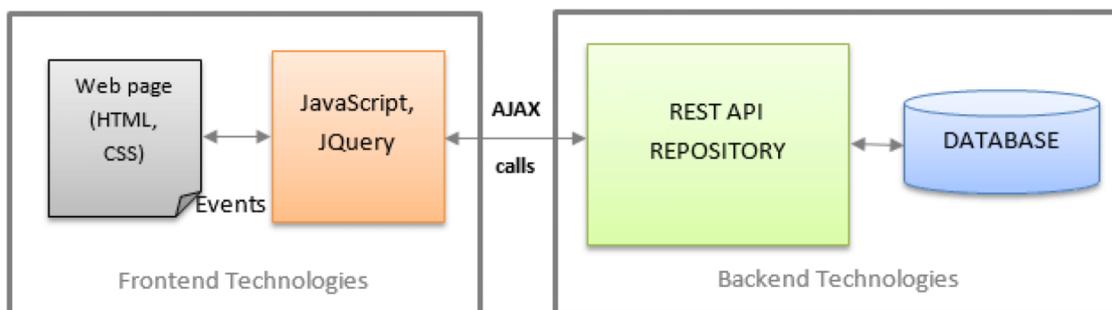


Figura 6.5: Uso de JavaScript en REST API.

Por otro lado, JavaScript se utiliza para realizar peticiones AJAX mediante JQuery a la API y así solicitar la información necesaria para mostrarle correctamente al usuario el contenido adecuado.

6.2.4. PHP

Este lenguaje [3] se utiliza principalmente para el procesamiento de datos en el lado del servidor, el cual recibe las peticiones del cliente por medio de un API y las procesa según sus requerimientos proporcionando en última instancia una respuesta compuesta por un objeto en formato JSON. En paralelo al procesamiento de datos, es necesario persistir los cambios del estado del sistema en base de datos, por lo que también se utilizará PHP para realizar las consultas apropiadas y así cumplir con el funcionamiento lógico de la aplicación.

6.2.5. IntelliJ

Para ampliar el contexto tecnológico, es interesante comentar el uso de herramientas para el desarrollo de la solución. Por lo tanto, a continuación se presentan las utilizadas en este proyecto.

Esta herramienta se ha empleado principalmente para el desarrollo de código y se trata de un IDE que puede entender la sintaxis de casi cualquier lenguaje de programación, hacer uso de plugins e incluso la ejecución de programas.

En concreto, de este IDE se ha utilizado el resaltado de código para HTML, JavaScript y PHP (figura 6.6).

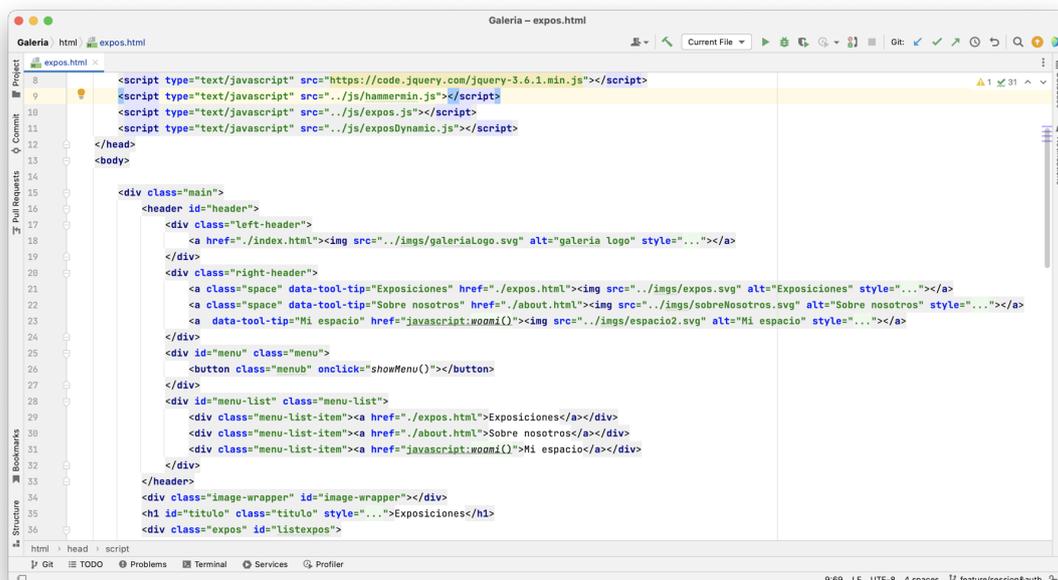


Figura 6.6: Ejemplo de resaltado de código HTML.

Y también se ha hecho uso de la utilidad 'refactor' del IDE para mejorar la legibilidad del código creado.

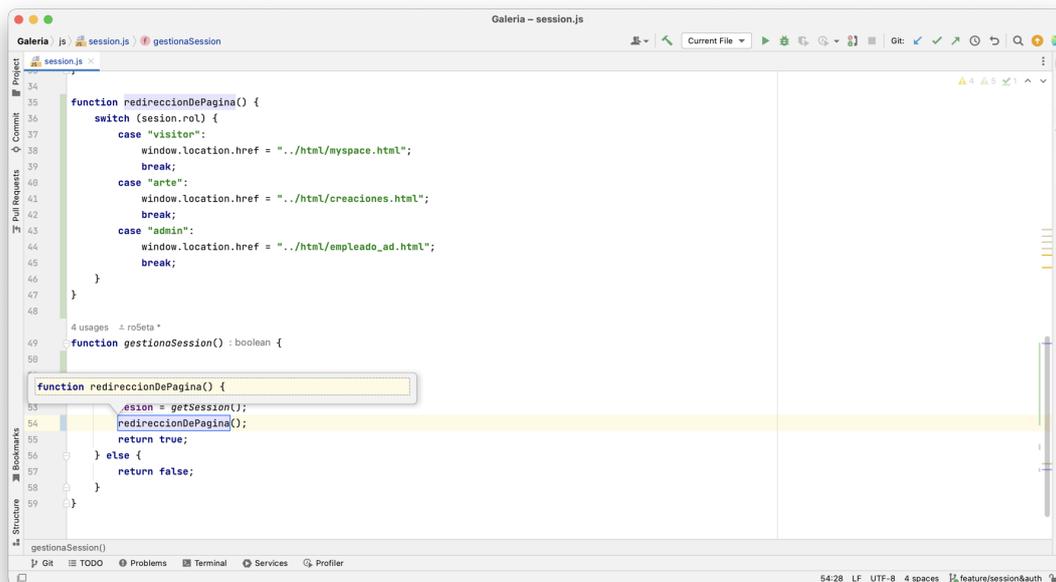


Figura 6.7: Uso del refactor para la producción de código limpio.

6.2.6. Postman

Herramienta para realizar peticiones HTTP y configurarlas según su url, método soportado y cuerpo del mensaje. Principalmente se ha utilizado este programa para realizar pruebas de caja negra en el lado del servidor y así validar el funcionamiento del entorno 'backend'. En la siguiente imagen se puede ver un ejemplo de petición a la API la cual trata de obtener las obras y sus detalles según el usuario visitante que las marcó como favoritas:

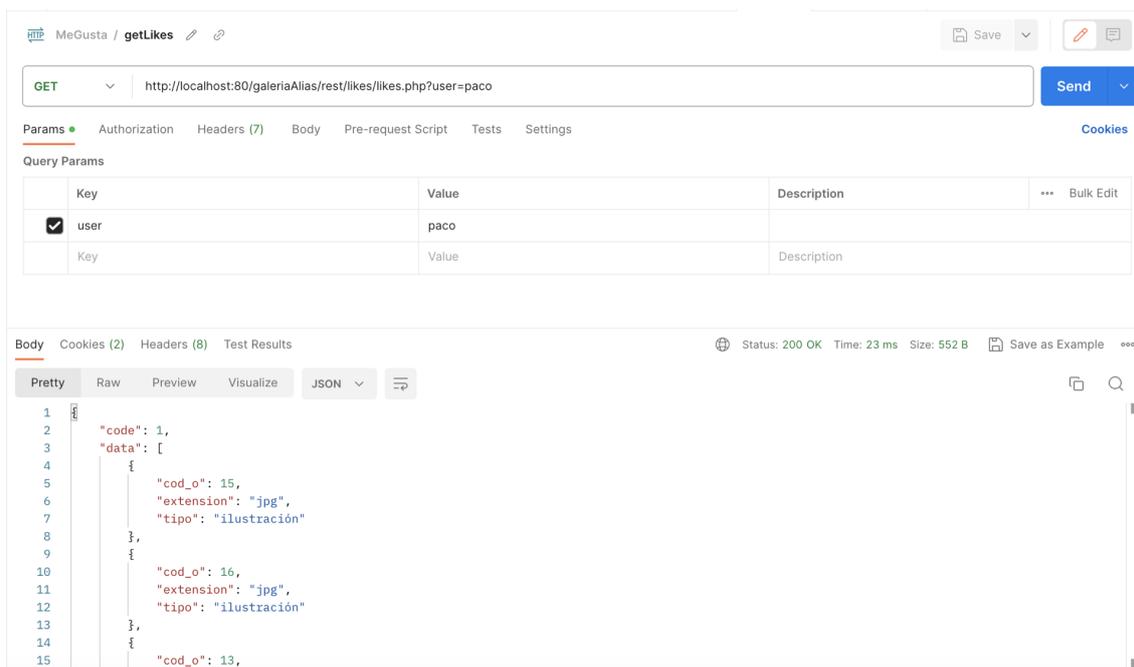


Figura 6.8: Uso de Postman para probar el funcionamiento de la API.

6.2.7. XAMPP

Este software de código abierto se ha utilizado para poner en marcha un servidor web apache y que es capaz de interpretar código PHP, además de uno de base de datos. Más adelante se verán más detalles de cómo se ha empleado esta herramienta en la sección de despliegue (6.3.4).

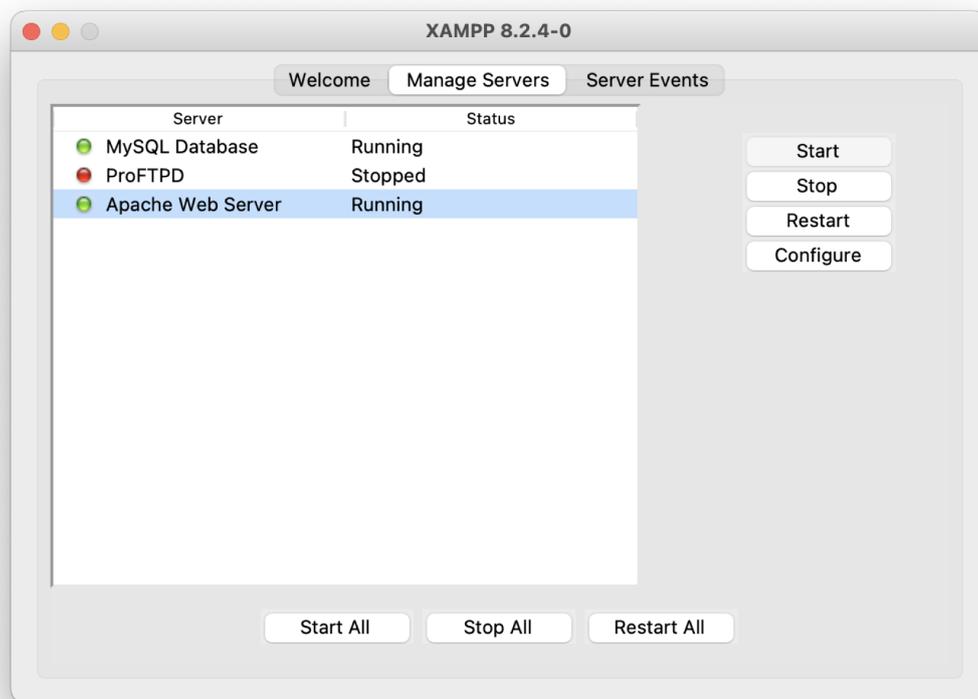


Figura 6.9: Uso de XAMPP para desplegar los servidores necesarios.

6.2.8. Chrome

Concretamente se han utilizado las herramientas de desarrollador del propio navegador, para la identificación y resolución de errores a la hora de integrar la funcionalidad de la web con el entorno 'backend'. Un ejemplo sería cuando se produce un error en el cual una tarea no genera los resultados esperados y por lo tanto se realiza una traza de la ejecución del código 'front-end' para determinar el origen del problema.

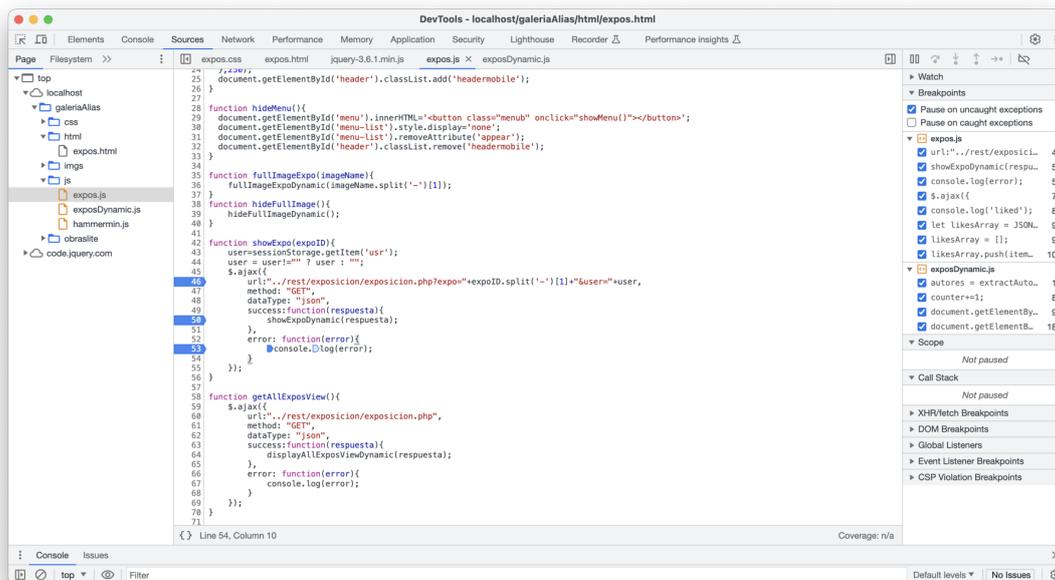


Figura 6.10: Uso de las herramientas de desarrollador de chrome.

6.3 Ejemplos de código

En este capítulo se van a dar una serie de explicaciones sobre cómo se ha desarrollado la aplicación web, es decir, qué tecnologías se han usado y cómo, a través de capturas que mostrarán de forma representativa los aspectos clave de la implementación de la solución.

6.3.1. Interfaz con HTML, CSS y Javascript

Comenzando con la interfaz web de la aplicación, se ha utilizado HTML como lenguaje de marcado para dar contenido a la web y también estructura a la misma. El código HTML se encuentra en archivos de extensión .html y son servidos por el servidor web. Un ejemplo de código HTML se puede observar en la figura 6.11, que muestra la estructura del lenguaje en forma de etiquetas. Este ejemplo corresponde a la página inicial de la aplicación y tiene como nombre 'index.html' (nombre común para el archivo de carga inicial).

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta name="viewport" content="width=device-width, initial-scale=1">
5   <title>Galería</title>
6   <link rel="icon" type="image/x-ico" href="./imgs/favicon.ico"/>
7   <link rel="stylesheet" href="./css/ini2.css">
8   <script type="text/javascript" src="https://code.jquery.com/jquery-3.6.1.min.js"></script>
9   <script type="text/javascript" src="./js/ini2.js"></script>
10  <script type="text/javascript" src="./js/session.js"></script>
11  <script type="text/javascript" src="./js/sessionDynamic.js"></script>
12  <link rel="apple-touch-icon" href="./imgs/galeriaAppIcon.png">
13  <meta name="apple-mobile-web-app-title" content="Galería">
14  <meta name="apple-mobile-web-app-capable" content="yes">
15  <link rel="manifest" href="./js/manifest.json">
16 </head>
17 <body>
18
19   <div class="main" style="...">
20     <header id="header" ...>
38     <div class="images" ...>
57   </div>
58
59 </body>
60 </html>

```

Figura 6.11: Ejemplo de código HTML.

Para ejemplificar cómo se ha utilizado este lenguaje a la hora de proveer contenido a la web, a continuación se presentan una serie de elementos que representan los objetos o etiquetas HTML más utilizadas. Primeramente, la etiqueta que más se utiliza para dar estructura a la aplicación es '`<div>`', que sirve de contenedor y permite agrupar y organizar otros elementos HTML, los cuales contiene. Un ejemplo de ello se muestra a continuación:

```

<div class="image-item">
  
</div>

```

Este elemento '`<div>`' contiene a una imagen ('``') y tiene como identificador de clase 'image-item', que sugiere que el usuario verá en la interfaz una obra según la disposición de su contenedor o elemento padre.

Otro ejemplo de etiqueta que es muy utilizada y que sirve para permitir al usuario iniciar la sesión es '`<form>`', que hace que el usuario pueda introducir datos (credenciales en este caso). Un ejemplo de ello se muestra a continuación:

```

<form onsubmit='return iniciarSession();' class='registro'>
  <input type='text' id='runV' value='' size='30' minlength='1' maxlength='10'
    required='required' placeholder='Inserta tu nombre de usuario'
    name='texto' />
  <br>
  <input type='password' id='rpasswordV' value='' size='30' minlength='1'
    maxlength='40' required='required' placeholder='Inserta tu contrase&ntilde;a'
    name='password' />
  <button class='ojo' id='ojoV' onclick='verPV()' type='button'>
  <img src='../imgs/eye.svg' width='25px' height='25px'
    onmouseover="this.src='../imgs/eye-2.svg'"
    onmouseout="this.src='../imgs/eye.svg'">
  </button>
  <br>
  <input type='submit' value='Iniciar' />
</form>

```

Donde se observa que este elemento agrupa también otro tipo de etiquetas como '`<input>`', que permite especificar qué tipo de entrada de datos será el elemento o '`<button>`', como elemento que el usuario puede clicar y así accionar una serie de eventos que realice la aplicación.

Algo a mencionar es que se ha habilitado la opción de que el usuario que esté utilizando un dispositivo móvil, ya sea un portátil, tablet o smartphone, tenga la posibilidad de 'instalarse' la web como una aplicación, siendo especialmente útil para tabletas y teléfonos móviles. Para ello se ha necesitado incluir los siguientes etiquetas dentro de la que se nombra como '`<head>`':

```
<link rel="apple-touch-icon" href="../imgs/galeriaAppIcon.png">
<meta name="apple-mobile-web-app-title" content="Galería">
<meta name="apple-mobile-web-app-capable" content="yes">
<link rel="manifest" href="../js/manifest.json">
```

También se ha hecho uso de un archivo 'manifest.json' que contiene parámetros de configuración para el correcto funcionamiento de esta 'web-app' y para que lo pueda interpretar correctamente el navegador (figura 6.12).



```
1 {
2   "name": "Galería",
3   "short_name": "Galería",
4   "start_url": "./index.html",
5   "display": "standalone",
6   "description": "Gestión de galería digital.",
7   "icons": [
8     {
9       "src": "../imgs/galeriaAppIcon.png",
10      "sizes": "512x512",
11      "type": "image/png"
12    }
13  ],
14  "scope": "/"
15 }
```

Figura 6.12: Archivo manifest.json

En referencia a JavaScript, éste lenguaje se ha utilizado para dos tareas en concreto: la primera de ellas para manejar las interacciones del usuario con la interfaz, y la segunda para manejar las peticiones al servidor y carga dinámica de contenido.

Un ejemplo para la primera tarea consistiría en el caso de uso en el que el usuario seleccionase una opción de filtrado de las obras que ha indicado como favoritas dentro de la vista 'Mi espacio' (figura 5.7). Según el código html de la figura 6.13, cuando el usuario seleccione una opción ejecutará la función 'filter(opción)'. Esta función se muestra en la figura 6.14.

```

<div class="filtrado">
  <select id="options" onchange="filter(this.value)">
    <option value="none">Sin filtros</option>
    <option id="date" value="date">Fecha publicada</option>
    <option id="photo" value="photo">Fotografía</option>
    <option id="draw" value="draw">Ilustración</option>
    <option id="abstract" value="abstract">Abstracto</option>
  </select>
</div>

```

Figura 6.13: HTML de una etiqueta <select>

```

102 function filterDynamic(filter) : void {
103     if(filter=="none"){
104         var clear : NodeListOf<Element> = document.querySelectorAll( selectors: '.fav-item');
105         for(var i : number = 0; i<clear.length; i++){
106             clear[i].style.display="inline-block";
107         }
108         fixGridViewDynamic();
109
110         var clear2 : NodeListOf<Element> = document.querySelectorAll( selectors: '.fav-cont-item');
111         for(var i : number = 0; i<clear2.length; i++){
112             clear2[i].style.display="inline-block";
113         }
114     } else{
115         var busc : NodeListOf<Element> = document.querySelectorAll( selectors: '.fav-item');
116         var cont : number = 0;
117         for(var i : number = 0; i<busc.length; i++){
118             if (!busc[i].classList.contains(filter)) {
119                 busc[i].style.display="none";
120             } else{
121                 busc[i].style.display="inline-block";
122                 cont+=1;
123             }
124         }
125         fixGridViewDynamic();
126
127         var busc2 : NodeListOf<Element> = document.querySelectorAll( selectors: '.fav-cont-item');
128         for(var i : number = 0; i<busc2.length; i++){
129             if (!busc2[i].classList.contains(filter)) {
130                 busc2[i].style.display="none";
131             } else{
132                 busc2[i].style.display="inline-block";
133             }
134         }
135     }
136 }

```

Figura 6.14: Función JavaScript de filtrado

Resumiendo el funcionamiento de este método, lo que trata de realizar es primeramente averiguar la opción seleccionada por el usuario, que se pasa como argumento en la función. Más adelante mostraría las obras que coincidan con la opción de filtrado. En caso de que la opción seleccionada sea la de 'none' (quitar filtro), el método dejaría a la vista todas las obras, pero en el caso de que el usuario marque una de las opciones, la función trataría de mostrar aquellas obras cuyo identificador de clase en HTML coincida con esa misma opción de filtrado.

Cabe mencionar la distinción de nombres de la función de filtrado, pues desde el código html se llama al método 'filter()' y posteriormente se ha mostrado el contenido del

mismo método pero con nombre distinto ('filterDynamic'). Aclarar que se le ha puesto esa coletilla 'Dynamic' porque a la hora del desarrollo software se ha seguido el principio de código limpio en el cual se separa la lógica esencial de aquella más concreta y que sirve de apoyo. Es decir, la lógica principal se encontraría en un archivo y aquella lógica concreta que no representa la funcionalidad de la aplicación se ubicaría en otro archivo javascript. Permitiendo así una mayor legibilidad y comprensión de lo que la lógica trataría de realizar.

Otro de los usos de JavaScript serviría para comunicarse con el servidor a través de la API y así poder cargar contenido en función de los datos recibidos. Por ejemplo, en el caso de uso en el que el usuario visitante accede a su espacio, tras iniciar sesión, habría que cargar todas aquellas obras, las cuales tienen que estar catalogadas, que haya dado con un 'me gusta'. Por ello hay que hacer una petición específica tras cargar el documento -DOM ready- (figura 6.15) a través de AJAX, por ejemplo (figura 6.16).

```
1  $(document).ready(function() : void {
2      checkSession();
3
4
5      // Getting obras
6      getLikedObras();
7  });
```

Figura 6.15: Ejemplo Document ready.

```
74  function getLikedObras() : void {
75      user = sessionStorage.getItem( key: "usr");
76      $.ajax({
77          url: "../rest/likes/likes.php?user="+ user,
78          method: "GET",
79          accept: "application/json",
80          success: function(respuesta) : void {
81              showLikedObrasDynamic(JSON.parse(respuesta));
82          },
83          error: function(error) : void {
84              showBad();
85          }
86      });
87  }
```

Figura 6.16: Ejemplo de petición AJAX.

Un ejemplo de código que utilizaría los datos recibidos del servidor para construir elementos en HTML y que sean mostrados por el navegador correspondería con el de la llamada a la función vista en la figura anterior, 'showLikedObrasDynamic(obras)' y que se puede ver en la figura 6.17 a continuación:

```

1 function showLikedObrasDynamic(respuesta) : void {
2   let pPlace : HTMLElement = document.getElementById( 'fav-grid' );
3   pPlace.innerHTML="";
4   let pPlace2 : HTMLElement = document.getElementById( 'fav-cont' );
5   pPlace2.innerHTML="";
6   if(respuesta.code===1){
7     var date_aux : string = new Date().toUTCString();
8     if (respuesta.data.length===0){
9       pPlace.innerHTML="<p style='margin-inline:auto;margin-top: 30px'>No hay obras para ver</p>";
10      pPlace.style.marginInline="auto";
11      pPlace.style.display="block";
12      pPlace2.innerHTML="<p style='margin-inline:auto;margin-top: 30px'>No hay obras para ver</p>";
13    } else {
14      respuesta.data.forEach(element => {
15        pPlace.innerHTML+="<div id='img-"+element.cod_o+"' class='fav-item '+getTipo(element.tipo)+"'> " +
16          "<a id='full-"+element.cod_o+"'"+element.extension+"' href='javascript:fullImage(\""+full-"+element.cod_o+"\""+element.extension+"')> " +
17            "<img src='../obrasLite/'"+element.cod_o+"\""+element.extension+"'"> " +
18              "<a>" +
19                "<button id='1-"+element.cod_o+"' class='heart' onclick='unFav(this.id)'></button>" +
20              "</div>";
21        pPlace2.innerHTML+="<div id='img2-"+element.cod_o+"' class='fav-cont-item '+getTipo(element.tipo)+"'><a id='full-"+element.cod_o+"\""+element.extension
22      });
23    }
24  } else {
25    console.log(respuesta.data);
26    pPlace.innerHTML="<p style='margin-inline:auto;margin-top: 30px'>No hay obras para ver</p>";
27    pPlace2.innerHTML="<p style='margin-inline:auto;margin-top: 30px'>No hay obras para ver</p>";
28  }
29 }

```

Figura 6.17: Ejemplo de construcción de elementos HTML en JavaScript.

En el que por cada elemento de la respuesta recibida (un array de obras en formato JSON) se crea un elemento '`<div>`' que contiene la imagen de la obra '``' y un botón '`<button>`' para poder indicar que esa obra ya no es favorita. También se hace uso de un elemento '`<a>`' que envuelve a la imagen para que cuando el usuario haga 'click' sobre la imagen pueda verla de forma completa y más grande.

Otro elemento de la web utilizado es el almacenamiento en sesión (session storage), que sirve a un caso concreto para ayudar al visitante cuando éste olvida iniciar sesión pero ya había marcado obras con un 'me gusta'. Por lo que se utiliza para almacenar los 'likes' cuando la sesión aún no ha sido iniciada y una vez el usuario se autentica en la web, se genera una nueva petición para guardar aquellas obras favoritas en caso de que no estuvieran previamente marcadas.

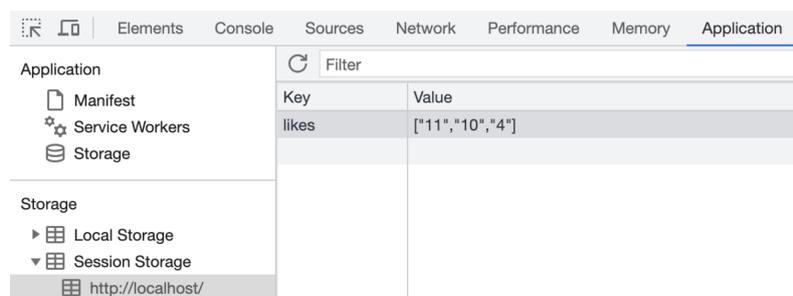


Figura 6.18: Almacenamiento en sesión.

```

21 function setSession(value) : void {
22   sessionStorage.setItem("usr", value);
23 }
24

```

Figura 6.19: Inserción de datos en 'session storage'.

Y para acabar esta sección, es importante destacar el efecto que produce el aspecto visual de una interfaz al usuario final, haciendo que éste se sienta más cómodo utilizando la web y que por lo tanto le haga querer volver a usarla. Para ello, siguiendo el prototi-

pado de las interfaces en la sección 5.2, se ha hecho un uso específico y personalizado de CSS y sin recurrir a otras herramientas como Bootstrap. El motivo de esto viene a través de tomar la decisión de querer practicar y aprender a manejar y estructurar elementos HTML a través de este lenguaje de estilo de texto en cascada.

```
27 header {
28     position: sticky;
29     top: 0;
30     width: 100vw;
31     height: 55px;
32     max-height: 400px;
33     background: rgba(255, 255, 255, 0.8);
34     -webkit-backdrop-filter: blur(10px);
35     backdrop-filter: blur(10px);
36     color: #000;
37     vertical-align: middle;
38     box-shadow: 0px 5px 15px #f1f1f1;
39     z-index: 9999;
40     transition: height ease-out 500ms;
41 }
```

Figura 6.20: Ejemplo de estilo CSS para el 'header' de la página web.

Un ejemplo de código CSS para dar estilo a la barra de navegación ('header') de la página web se puede observar en la figura 6.20. Donde se ha utilizado color, efectos 'blur', tamaño, posición, sombra, jerarquía de preferencia e incluso una transición para cuando el usuario en versión móvil despliegue las opciones de la barra de navegación.

6.3.2. API REST y PHP

Para poder comunicar la interfaz web con el servidor web ha sido necesario el uso de una API REST. Esto tiene como ventaja la escalabilidad, porque las distintas partes pueden evolucionar independientemente. Para la integración de estos componentes se ha creado una serie de métodos y formatos de datos para la correcta comunicación entre el navegador web y el servidor.

Un ejemplo de API puede verse en el apéndice C, que corresponde a las principales acciones que se pueden realizar sobre una obra: obtener una obra según su identificador, obtener obras catalogadas, según el empleado que las creó, editar una obra, crear una obra y descatalogar una obra.

Como se ha podido observar en los 'path' de las URL que solicitan las peticiones en el API Obras, aparece al final de los enlaces 'obra.php', y esto es así porque, como ya se había mencionado antes, se hace uso de PHP para gestionar las peticiones del cliente, procesar la información necesaria y construir el objeto de respuesta en formato JSON.

```

11 try{
12     switch($_SERVER['REQUEST_METHOD']){
13         case "GET": echo GET();break;
14         case "POST": echo POST();break;
15         case "PUT": echo PUT();break;
16         case "DELETE": echo DELETE();break;
17         default: echo getResponse( code: -1, data: "Método HTTP no soportado");
18     }
19 }catch(Exception $e){
20     die("Error de procesamient: $e");
21 }

```

Figura 6.21: Métodos HTTP manejados en PHP.

La figura 6.21 muestra un ejemplo de código en PHP de cómo gestionaría las peticiones del cliente según el método HTTP utilizado. Siguiendo con el caso de uso en el que un empleado crease una obra y el navegador realizase una petición de método PUT, de path '/rest/obra/obra.php' y en el cuerpo un objeto JSON con la información de la obra a crear, el servidor gestionaría la petición extrayendo los datos recibidos del cliente y llamando a un método para que inserte la obra en BBDD para este caso.

```

53 function PUT(){
54     $payload=file_get_contents( filename: 'php://input');
55     $obra = json_decode($payload);
56     return insertObra($obra);
57 }

```

Figura 6.22: Manejo de una petición PUT en PHP, 2.

Otro uso de PHP es para las cookies en el inicio de sesión. Cuando el usuario inicia sesión y se autentica correctamente, el servidor crea una cookie con el nombre se usuario registrado en el sistema y el rol al cual pertenece (6.23), permitiendo así a la web tener constancia del tipo de usuario que ha iniciado sesión. Este caso es especialmente útil cuando un empleado inicia su sesión, pues éste puede tener un rol artístico o administrativo y entonces, sin tener que generar dos formularios para el usuario empleado, la web podrá saber a qué página redirigir al usuario y qué contenido cargar según el mismo.

```

74 function setSession($user){
75     $cookie_name = "usr";
76     $cookie_value = $user['userName'];
77     setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
78     $cookie_name = "rol";
79     $cookie_value = $user['rol'];
80     setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
81     return getResponse( code: 1, $user);
82 }
83

```

Figura 6.23: Creación de cookies en PHP.

Por último, cabe mencionar el esfuerzo de mejora del rendimiento de la página web a través de la reducción de la resolución de las imágenes sin dejar de ofrecer una versión completa y con la calidad original. Este proceso se produce cuando el empleado artístico crea o edita una obra e introduce una nueva imagen, por lo que se realizaría una petición al servidor con los nuevos datos. Ahora, el servidor, mediante una función en PHP (ver figura ejemplo siguiente) crearía dos versiones ya mencionadas de la imagen recibida y las guardaría en destinos diferentes. Luego, cuando el empleado o el visitante visualicen obras de un primer vistazo, éstas son las de versión reducida. En cambio, las versiones originales y con mayor detalle sólo la podrán visualizar los usuarios visitantes cuando,

al ver una obra, la seleccionen y vean a tamaño completo, produciéndose una solicitud al servidor de dicha imagen.

```
<?php
$portada=$obra->cod_o.".".$obra->extension;
$portadaFile = fopen("../imagen_original/".$portada, "w")
    or
    die("Unable to open file: ".$portada);
fwrite($portadaFile, base64_decode($obra->portada));
fclose($portadaFile);
compressImage("../imagen_original/".$portada,
    "../imagen_reducida/".$portada,50);
?>
```

Resumiendo este código, primero se trata de obtener el nombre de archivo de la imagen de la obra (codigo + extensión). Luego, se obtiene la imagen original y se guarda en su destino. Finalmente, se comprime la imagen original según un porcentaje de reducción, en este caso 50 %, a un destino diferente.

6.3.3. Consultas en BBDD

Para que las peticiones puedan satisfacerse correctamente, el servidor web ha de consultar una serie de información en la BBDD para así generar el objeto de respuesta que el cliente necesita para su correcto funcionamiento. Sin ir más lejos, las consultas a la base de datos se ejecutan mediante PHP a través de una conexión MYSQL directa.

```
3      function conectar(){
4          $host="localhost";
5          $port="3307";
6          $user="root";
7          $pass="";
8          $bd="Galeria";
9
10         // Se conecta al SGBD
11         $conexion = mysqli_connect( hostname: $host.".".$port,$user,$pass,$bd);
12         if(!$conexion ) die("Error: No se pudo conectar");
13
14         mysqli_set_charset($conexion, charset: "utf8");
15
16         return $conexion;
17     }
```

Figura 6.24: Conexión a BDD mediante PHP.

Siguiendo con el caso de autenticación de la sección anterior, cuando el usuario clicla en el botón de iniciar sesión, se envía desde JavaScript una petición al servidor con las credenciales del usuario (nombre de usuario y contraseña), la cual recibe y procesa el servidor de la manera en que se puede observar en la figura 6.25.

```

3  function auth($user){
4      $conexion=conectar();
5
6      $userName = findUserName($user, $conexion);
7      $pwd = findPwd($user, $conexion);
8
9      $validation = validateUser($userName, $pwd);
10
11     if ($validation['ok']){
12         saveLikes($user->user, $user->obras);
13         return getSession($userName);
14     } else {
15         return getResponse( code: -1,$validation['error']);
16     }
17 }

```

Figura 6.25: Función de autenticación en PHP.

Resumido el método de autenticación, trata de hacer una primera validación de los datos recibidos y en el caso de que las credenciales recibidas son correctas la sesión se iniciaría y con las cookies necesarias vistas anteriormente. Pero para poder validar las credenciales, el sistema ha de consultar que la información existe y además concuerda con lo que hay registrado en la BBDD. A continuación se muestra un ejemplo de código en PHP que realizaría una consulta a la base de datos para así poder comprobar en primera instancia que el usuario indicado efectivamente existe:

```

<?php
    $conexion=conectar();
    $stmt = $conexion->prepare("SELECT e.usuario, e.rol
    FROM `Empleado` e
    WHERE e.usuario='$user->user'");
    $stmt->execute();
    $stmt->bind_result($userName, $rol);
    $stmt->fetch();
    desConectar($conexion, $stmt);
?>

```

6.3.4. Despliegue

Una vez está completado el código que permite al cliente visualizar la aplicación y también aquel que permite procesar y persistir la información el cual el usuario ha solicitado, es momento de la puesta en marcha de la aplicación. Para ello se ha utilizado una herramienta de código abierto llamada 'XAMPP', software que ofrece un entorno para levantar un servidor local y trabajar en desarrollo web.

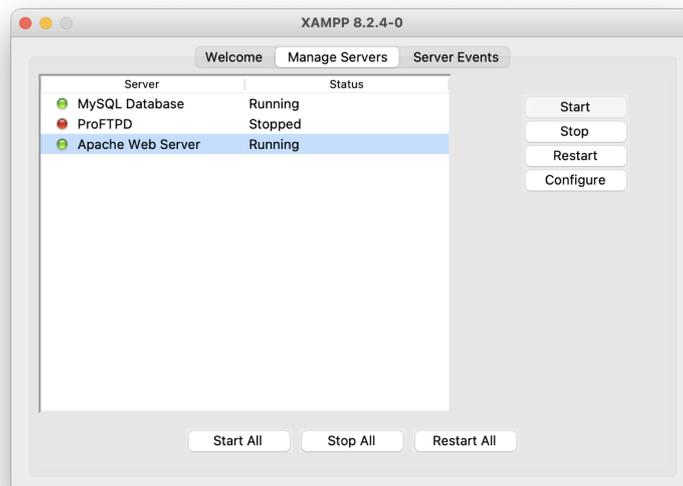


Figura 6.26: XAMPP: herramienta de código abierto para entorno web.

Los componentes de este software que se han utilizado para el despliegue de la aplicación son: Apache para el servidor; PHP para que el servidor pueda comprender y ejecutar código de este lenguaje y finalmente MySQL para configurar y utilizar una base de datos.

La configuración del servidor local es básica, y se muestra a continuación:

```
host: http://localhost
puerto: 80
ruta raíz: /galeria
```

Los archivos que componen la web y el servidor (html, css, js, php e imágenes) se encontrarían en el endpoint que marca la ruta raíz y seguirían una organización de ficheros tal que a la hora de enlazar a través de la url habría cierta coherencia.

Acabando con el servidor, al ser la aplicación de uso global, hay que abrir una entrada al mismo desde la red en la que se encuentre para que así los visitantes y empleados puedan acceder desde cualquier parte del mundo con acceso a Internet. Para ello habría que seguir los siguientes pasos:

- Redireccionar desde el punto de acceso de la red en la que se encuentre (por ejemplo el router) los paquetes que apunten al servidor y de puerto 80
- Utilizar un servicio DNS para traducir la IP del servidor a un nombre de dominio útil y fácil de recordar
- Por lo tanto, hacer saber al router de esta traducción, accediendo a la web de administración del mismo y en sus configuraciones indicarlo adecuadamente

En cambio, la configuración de la BBDD sería la siguiente:

```
user=user
port=3307
socket =/Applications/XAMPP/xamppfiles/var/mysql/mysql.sock
key_buffer=16M
```

```

max_allowed_packet=1M
table_open_cache=64
sort_buffer_size=512K
net_buffer_length=8K
read_buffer_size=256K
read_rnd_buffer_size=512K
myisam_sort_buffer_size=8M

```

Comentar que el nombre de usuario y contraseña para acceder a la base de datos ha de poseer una construcción fuerte por temas de seguridad.

Adicionalmente a la configuración inicial del servidor de BBDD, cómo realmente se crea la base de datos es mediante la ejecución de código SQL para así construir la BBDD, sus tablas, relaciones y restricciones, entre otros. Esto se hace a través de la web propia del servidor local creado mediante XAMPP (figura 6.27).

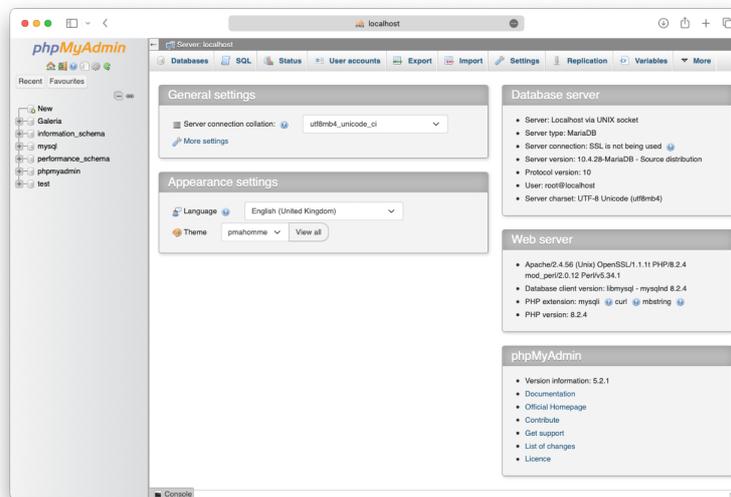


Figura 6.27: XAMPP: web de gestión del servidor local.

Como se observa en la figura anterior, añadir una nueva base de datos no es tarea compleja, pues arriba a la izquierda se ofrece la opción de añadir una BBDD. Aunque se pueda crear con código SQL, la interfaz web, con sólo indicar el nombre de la base de datos y el tipo de formato de caracteres admitidos, crea directamente la BBDD.

Sin embargo, resulta más productivo y eficaz crear el resto de elementos de la base de datos, como son las tablas y sus relaciones y los datos que contienen, con código SQL. Un ejemplo de creación de tabla y de inserción de datos en la misma se muestra a continuación:

```

CREATE TABLE `Catalogo_obras` (
  `usr` varchar(8) NOT NULL,
  `c_o` int(11) NOT NULL,
  `publicacion` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Claves primarias para la tabla `Catalogo_obras`

```

```
--  
ALTER TABLE `Catalogo_obras`  
ADD PRIMARY KEY (`usr`,`c_o`),  
ADD KEY `c_o3` (`c_o`);  
  
--  
-- Inserción de datos en la tabla `Catalogo_obras`  
--  
  
INSERT INTO `Catalogo_obras` (`usr`, `c_o`, `publicacion`)  
VALUES ('Pascual', 1, '2023-03-28')
```

CAPÍTULO 7

Producto desarrollado

Tras todo el trabajo realizado en los capítulos anteriores, se ha llegado a un resultado el cual servirá de herramienta a la galería para crear y exponer sus mejores obras al público, el cual, utilizará esta aplicación web para visualizar el contenido y guardar aquello que más le guste.

Por lo que la mejor manera de mostrar los resultados en un documento como tal es presentando un conjunto de capturas junto con sus explicaciones para tratar de ofrecer el mejor entendimiento de lo que la aplicación y, por lo tanto, el usuario son capaces de llegar a realizar.

7.1 Empleado artístico

Siguiendo los casos de uso de la sección 4.2 y comenzando con el usuario empleado, concretamente el artístico, deberá iniciar sesión en el sistema para poder comenzar a realizar alguna de las acciones comentadas en esa sección mencionada antes. Entonces, deberá acceder al inicio de sesión desde el icono de perfil en la barra de navegación como se muestra en la figura 7.1.

Una vez accedido a la página de inicio de sesión y seleccionado el rol propio, el usuario visualizará algo como la siguiente imagen:

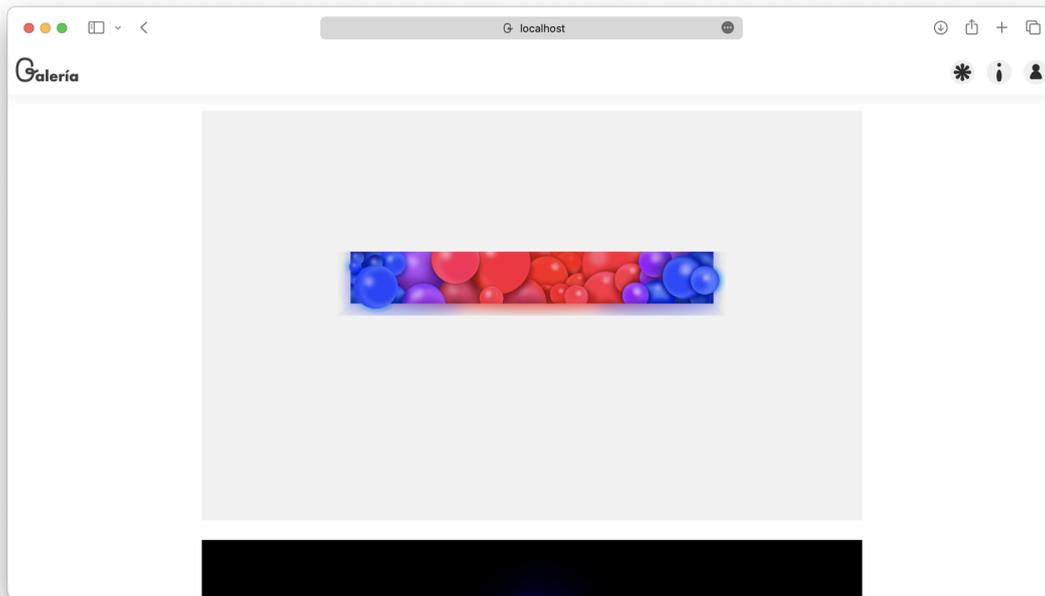


Figura 7.1: Página inicial de la aplicación web.



Figura 7.2: Página de inicio de sesión.

Tras autenticarse correctamente, el empleado ahora es bienvenido con la vista de la figura 7.3 y ya es capaz de realizar las acciones propias del usuario 'empleado artístico'.

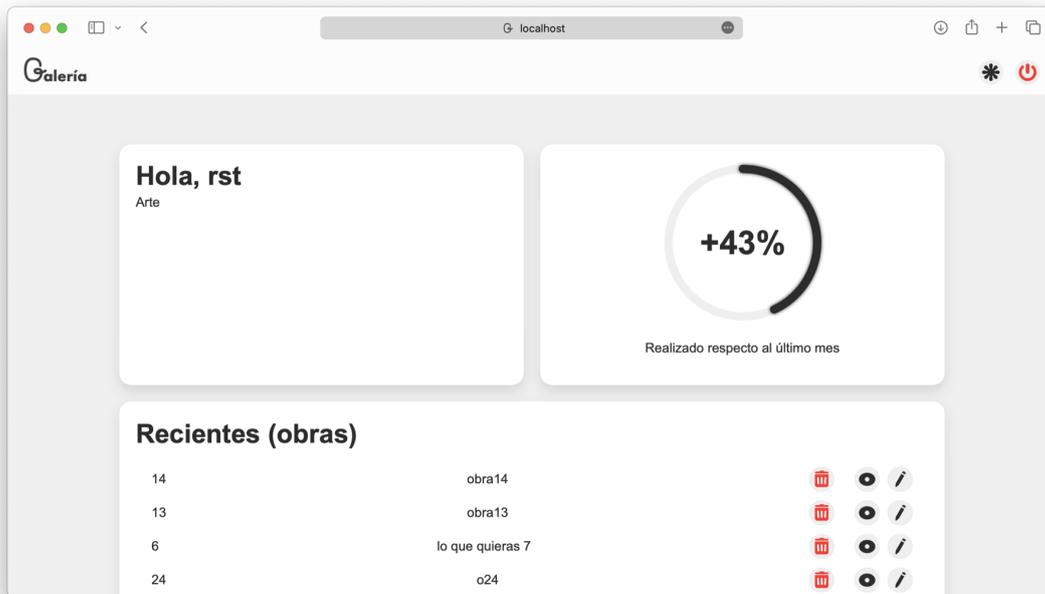


Figura 7.3: Página de bienvenida del empleado.

A partir de ahora, cuando el usuario vea una obra o exposición podrá realizar las acciones que se le ofrecen: descatalogar, ver detalles y editarlos. Aunque pueda el usuario desde esta vista realizar las acciones mencionadas anteriormente, se mostrarán a continuación en la sección de creaciones que se accede a través de la barra superior de navegación.

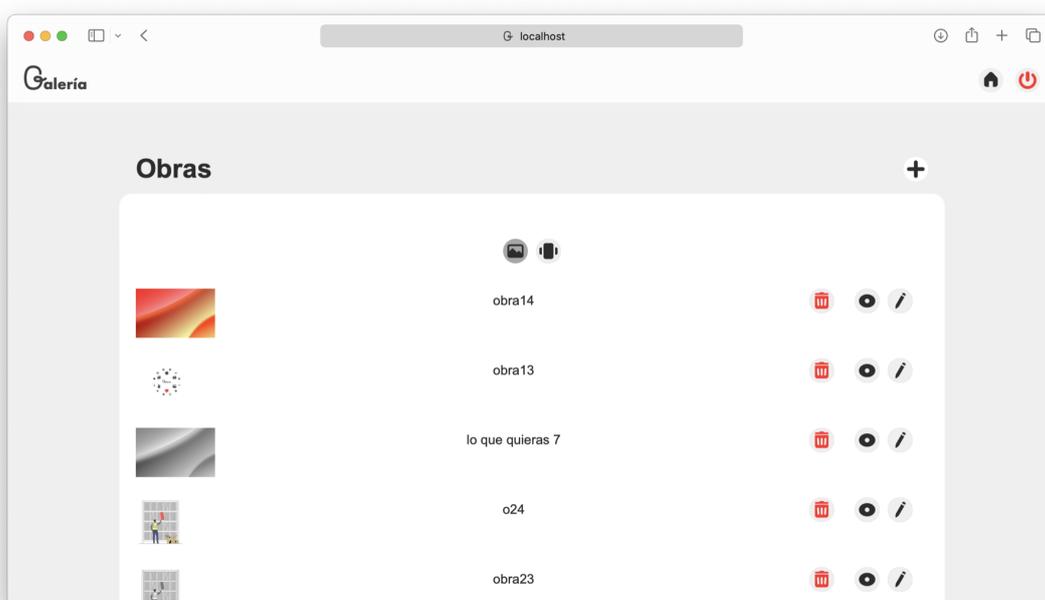
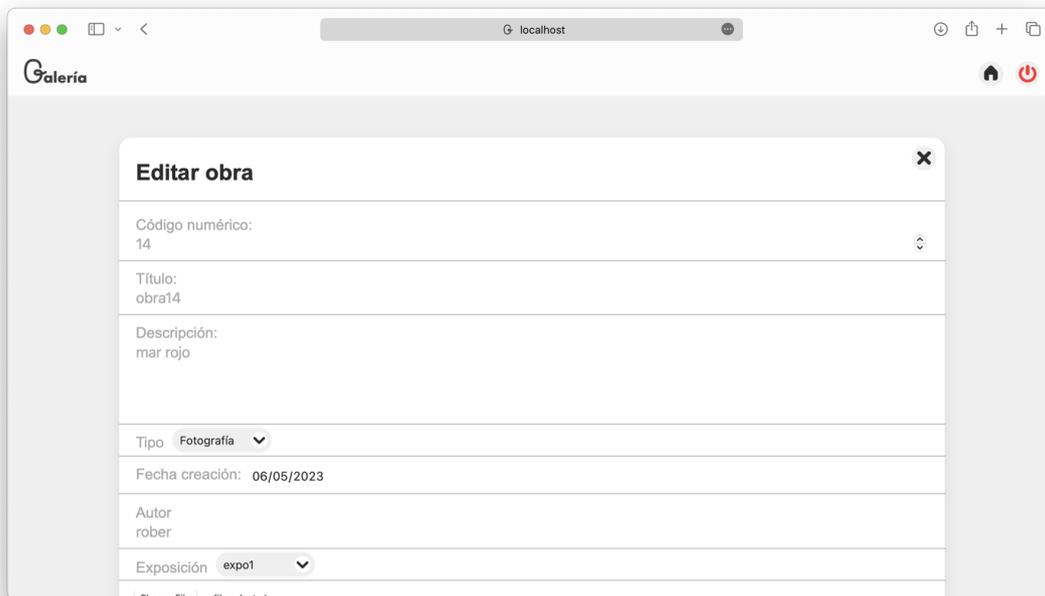


Figura 7.4: Vista de creaciones del empleado.

En el caso en que el empleado desee editar una obra, sólo tendrá que seleccionar el botón de edición correspondiente y le aparecerá un formulario como se muestra a continuación:



The screenshot shows a web browser window with the URL 'localhost'. The page title is 'Galería'. A modal window titled 'Editar obra' is open, containing the following form fields:

- Código numérico: 14
- Título: obra14
- Descripción: mar rojo
- Tipo: Fotografía
- Fecha creación: 06/05/2023
- Autor: rober
- Exposición: expo1

At the bottom of the form, there is a 'Choose File' button with the text 'no file selected'.

Figura 7.5: Vista de edición de obra.

Ahora el usuario podrá editar los campos deseados y una vez decida que ya es suficiente deberá guardar los cambios seleccionando el botón correspondiente y en caso de éxito verá una confirmación como lo que muestra la figura 7.6.

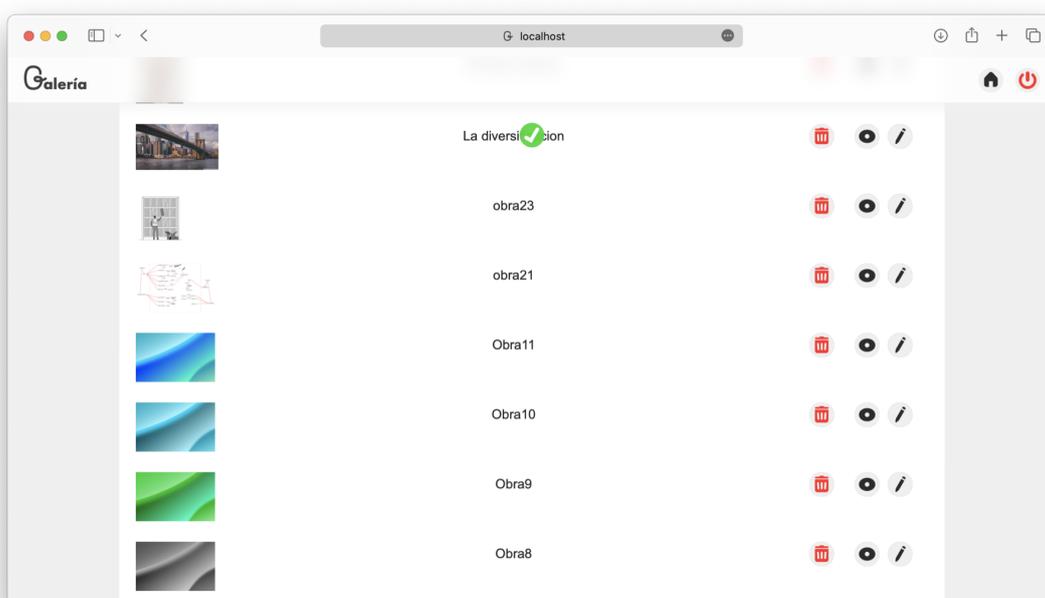


Figura 7.6: Vista de creaciones del empleado.

Por otro lado, como ya se ha comentado anteriormente, el empleado podrá descatalogar una obra siempre que desee. Para ello deberá seleccionar el botón de descatalogar en la obra correspondiente. Esto visualmente se reflejará con la obra desapareciendo de la página web.

Pero si el usuario quiere revertir esta acción, deberá, dentro de la misma vista de creaciones, seleccionar el botón de añadir, que le desplegará dos opciones: crear obra y recuperar obra. Cuando seleccione recuperar obra le aparecerá un listado de obras descatalogadas que podrá volver a catalogar 'clicando' la opción de recuperar en la obra deseada.

Acabando con las acciones que puede llevar a cabo el usuario empleado sobre las obras, éste podrá crear cuántas desee. Para ello deberá seleccionar en el desplegable comentado anteriormente la opción de crear obra y entonces se le mostrará un formulario en blanco como se muestra en la figura 7.7.

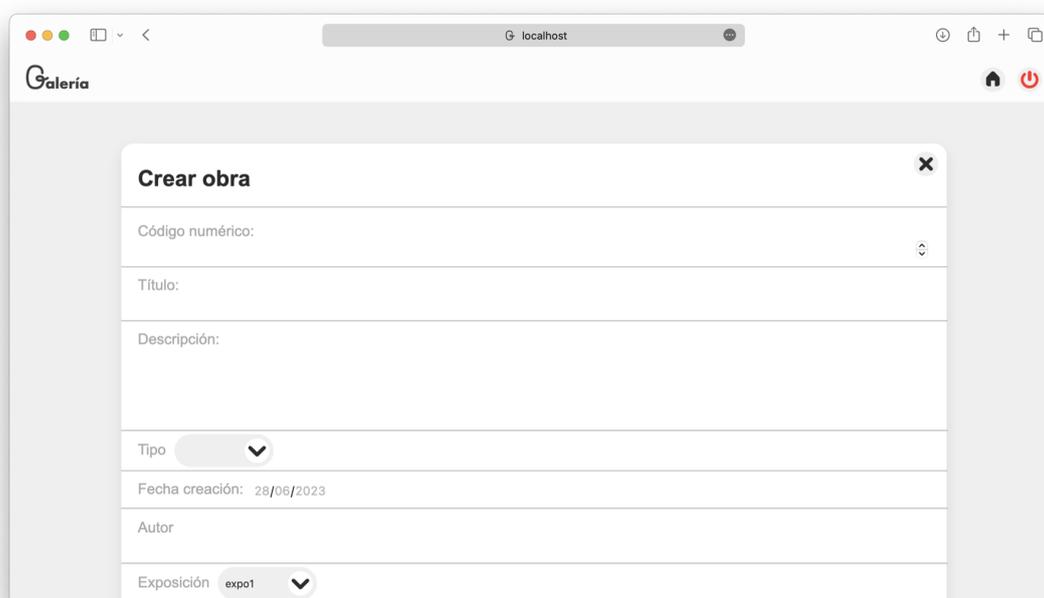


Figura 7.7: Vista de creación de obra.

Y una vez rellenado el mismo ya podrá guardar la nueva creación en el sistema para que sea mostrada en la galería digital.

Respecto a las exposiciones, el empleado deberá realizar un flujo de tareas muy similar al de la gestión de obras para acometer las respectivas acciones. Por tanto, para comenzar su trabajo, el empleado deberá iniciar sesión (figura 7.2), tras el cual, recibirá la misma bienvenida vista anteriormente (figura 7.3).

De nuevo, el empleado deberá acceder a la vista de creaciones para poder realizar todas sus tareas (figura 7.4). Sin embargo, como se tratan de exposiciones, el usuario deberá cambiar al apartado de exposiciones seleccionando la opción que se muestra en la figura siguiente:



Figura 7.8: Opciones de cambio entre apartados.

Quedando así la nueva vista de creaciones para las exposiciones:



Figura 7.9: Vista de apartado de exposiciones.

Ahora, una de las acciones que podrá realizar el empleado es la de editar los detalles de cualquiera de las exposiciones que se le muestren. Tras seleccionar el botón de editar en la creación deseada, el usuario verá un formulario para cambiar el contenido de la exposición, como el siguiente:

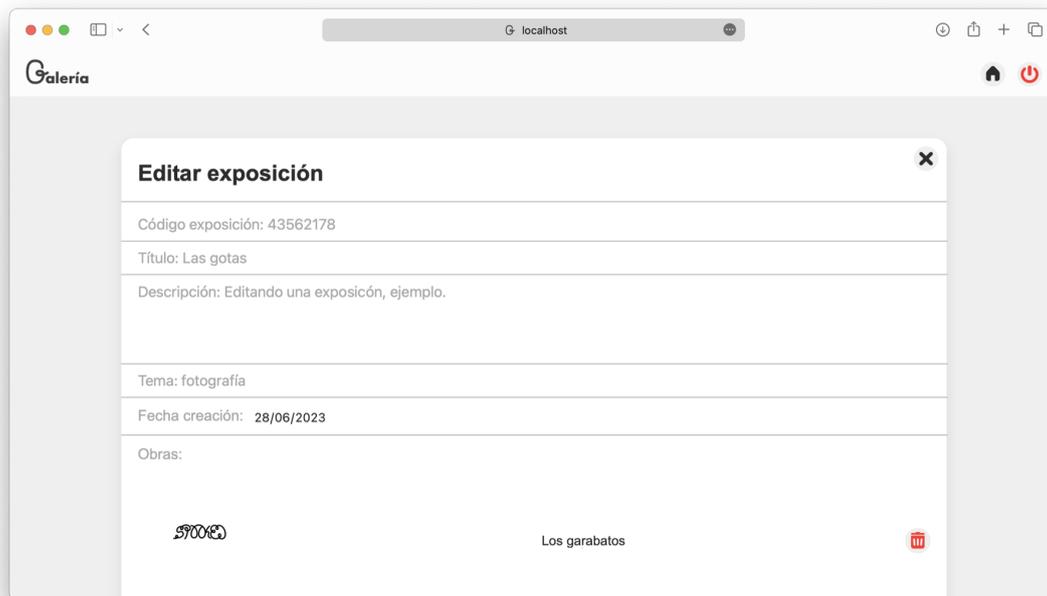
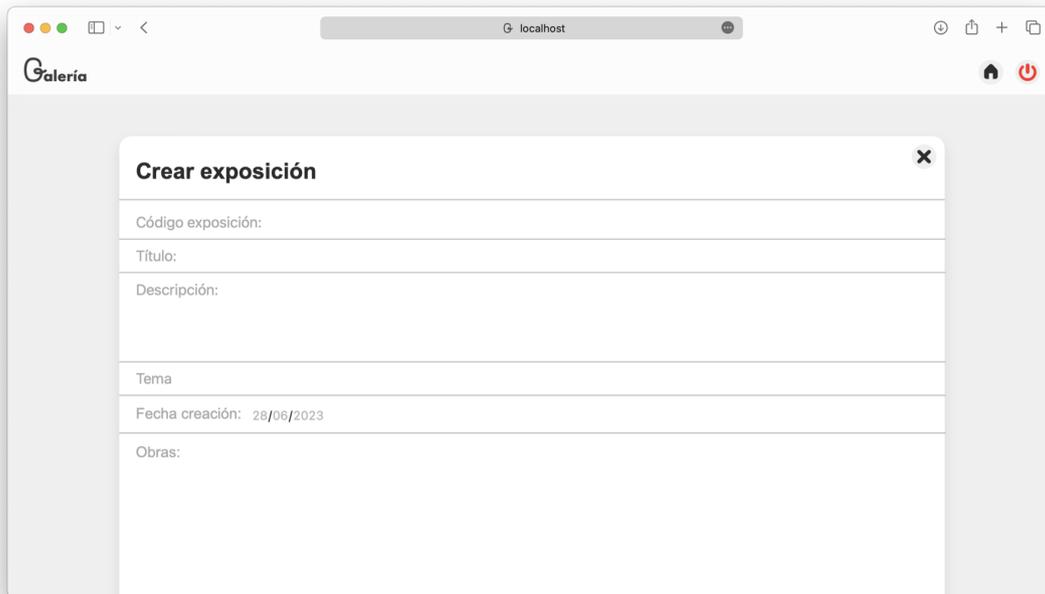


Figura 7.10: Edición de exposición.

Otra de las acciones que el empleado puede realizar sobre una exposición es la de descatalogarla, que para ello deberá seleccionar la opción de descataloga exposición. Cabe comentar que al descatalogarla sus correspondientes obras serán también automáticamente descatalogadas. Por el contrario, el usuario podrá también recuperarla accediendo de la misma manera que como en el caso de las obras. De nuevo, las obras asociadas a esa exposición serán recuperadas también.

Finalmente, el empleado tendrá la opción de crear una nueva exposición accediendo de la misma manera que en el caso de las obras. Más adelante, el usuario se topará con un formulario a rellenar como el de la figura siguiente:



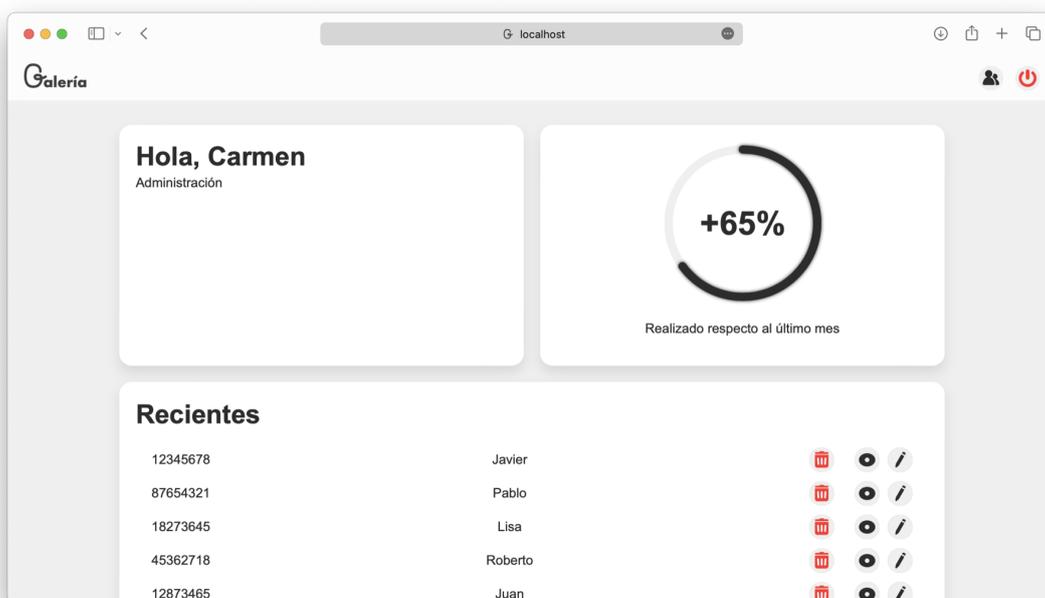
The screenshot shows a web browser window with the address bar set to 'localhost'. The page title is 'Galería'. A modal form titled 'Crear exposición' is displayed, containing the following fields:

- Código exposición:
- Título:
- Descripción:
- Tema:
- Fecha creación: 28/06/2023
- Obras:

Figura 7.11: Creación de una exposición.

7.2 Empleado de administración

Continuando con el siguiente usuario, el empleado de administración, éste podrá gestionar el personal de la galería digital. Como se ha visto antes, el usuario deberá iniciar sesión y, gracias al uso de cookies (sección 6.3.2), automáticamente accederá a su portal según su rol.



The screenshot shows the administrative dashboard for user Carmen. The page title is 'Galería'. The dashboard includes the following elements:

- Greeting: **Hola, Carmen** (Administración)
- Progress indicator: A circular gauge showing **+65%** Realizado respecto al último mes.
- Table of recent employees:

Recientes		
12345678	Javier	[Delete] [View] [Edit]
87654321	Pablo	[Delete] [View] [Edit]
18273645	Lisa	[Delete] [View] [Edit]
45362718	Roberto	[Delete] [View] [Edit]
12873465	Juan	[Delete] [View] [Edit]

Figura 7.12: Página de inicio del empleado administrativo.

Donde el administrador podrá realizar todas sus tareas es en la sección de personal que se puede acceder a través de la barra de navegación superior, en el que se puede observar que ha sido adaptado a las necesidades y usos de este usuario.

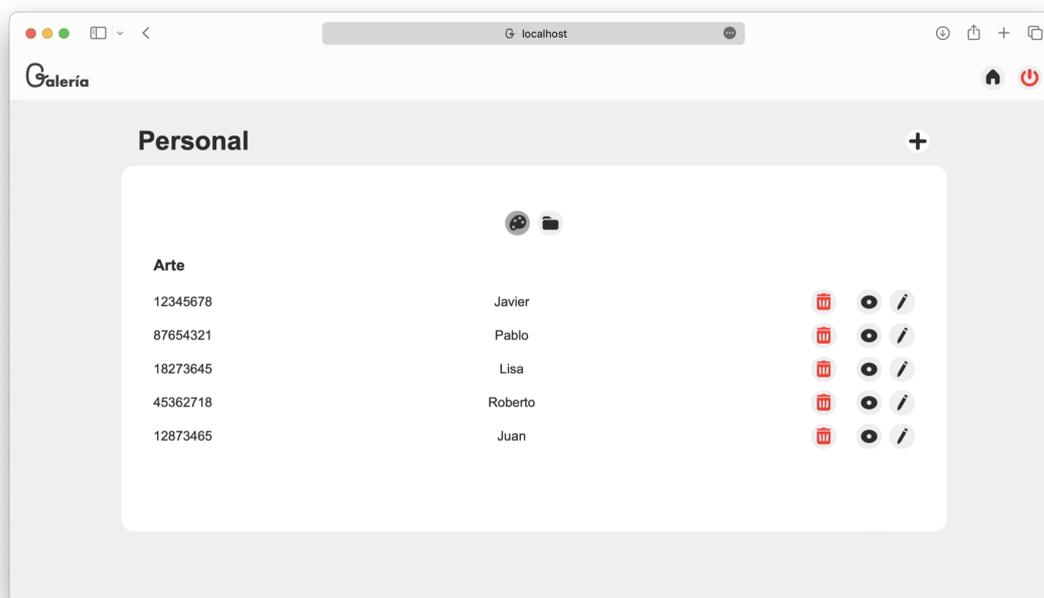


Figura 7.13: Vista del personal.

Como la galería digital está gestionada por dos empleados, esta sección permite cambiar la vista entre apartado de empleados artísticos y de empleados administrativos (figura 7.14).

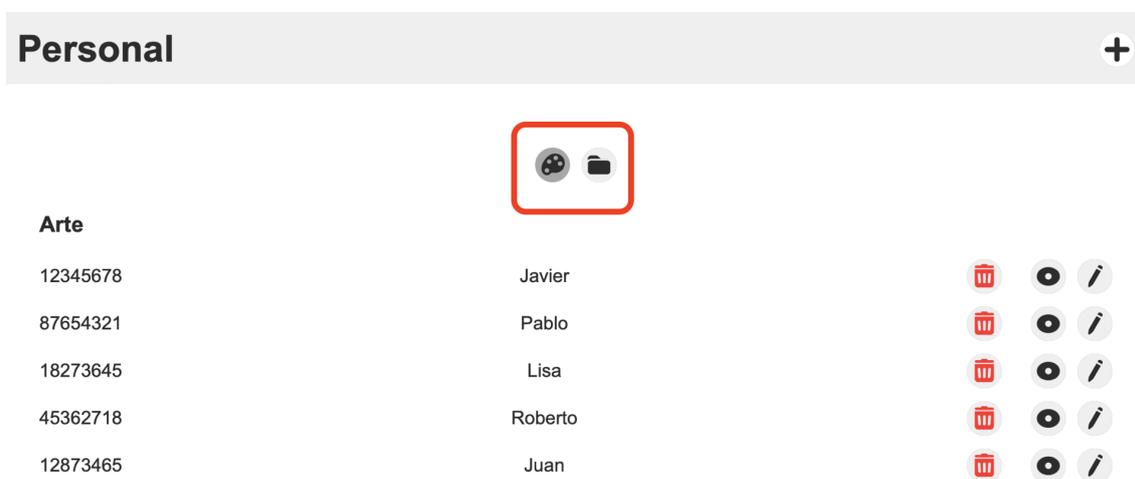


Figura 7.14: Opciones de cambio de apartados.

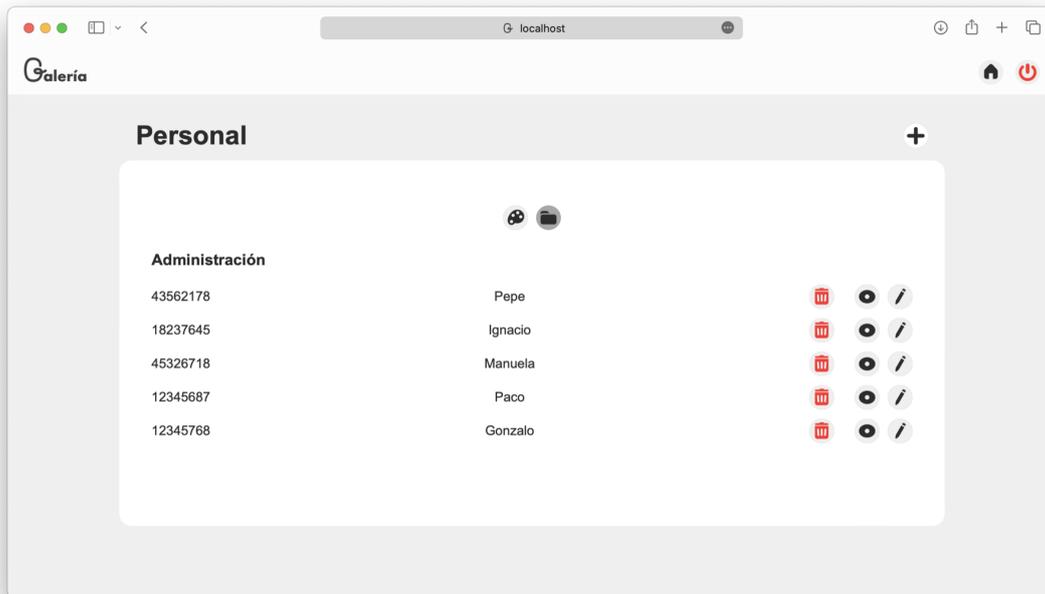


Figura 7.15: Vista del personal (2).

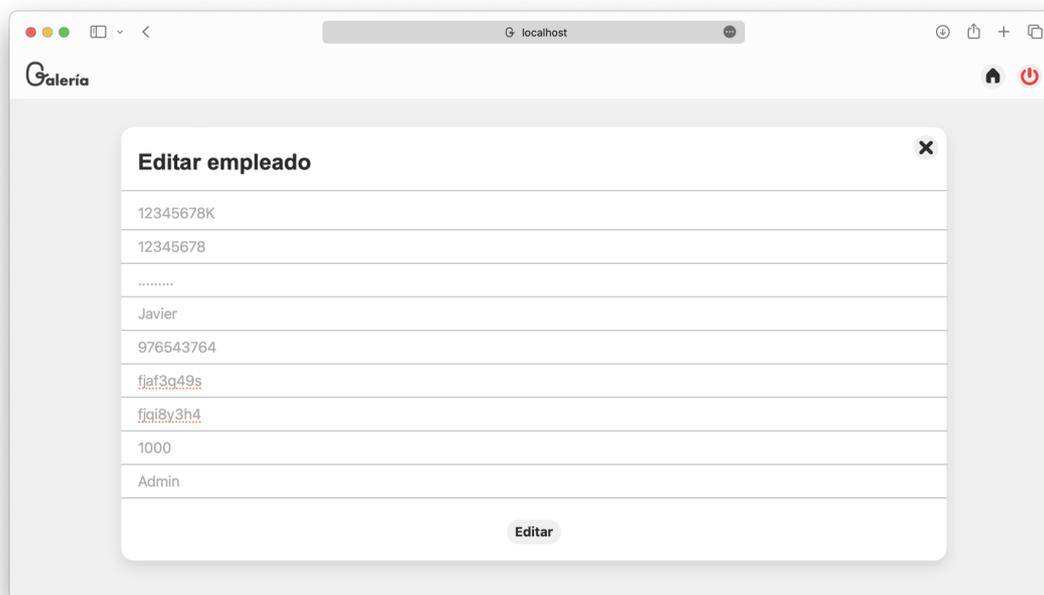
En cualquier caso, las acciones que se pueden realizar sobre los distintos empleados serán las mismas. Por lo que un primer caso sería el de dar de alta a un empleado, en el que habría que seleccionar la opción o botón de añadir empleado, entonces, se mostraría un formulario de creación como el siguiente:

Crear empleado	
DNI	
Usuario (DNI sin letra)	
Contraseña	
Nombre	
Teléfono	
NSS	
Número de cuenta	
Salario	
Arte	
<input type="button" value="Crear"/>	

Figura 7.16: Alta de empleado.

Por el contrario, el administrador podrá dar de baja a un empleado seleccionando la opción de eliminar empleado, borrando del sistema toda la información referente a ese empleado.

Por los motivos que sean, el administrador podrá editar los detalles del empleado y, una vez seleccionado el botón de editar en el empleado deseado, podrá modificar la información que aparece en el formulario de ejemplo siguiente:



The screenshot shows a web browser window with the address bar set to 'localhost'. The page title is 'Galería'. A modal window titled 'Editar empleado' is open, containing a form with the following fields:

12345678K
12345678
.....
Javier
976543764
fjaf3q49s
fjqj8y3h4
1000
Admin

At the bottom of the form is a button labeled 'Editar'.

Figura 7.17: Edición de un empleado.

7.3 Visitante

Terminando con el último usuario, el visitante es el motivo por el que se ha creado todo este sistema informático que permite la actividad de una galería. Una vez accede a la web, se le presenta una página de inicio (figura 7.1) con un conjunto de obras en orden aleatorio para que se pueda hacer una ligera idea de qué contenido se puede encontrar en la web.

Para navegar por el resto de la web, el usuario deberá seleccionar las opciones que se le ofrece en la barra superior de navegación. En el caso que el visitante sea nuevo en la galería, podrá registrarse accediendo a través del botón de perfil de la barra de navegación y, tras seleccionar su rol, podrá indicar sus credenciales para así poder iniciar su sesión posteriormente.



Figura 7.18: Inicio de sesión visitante.

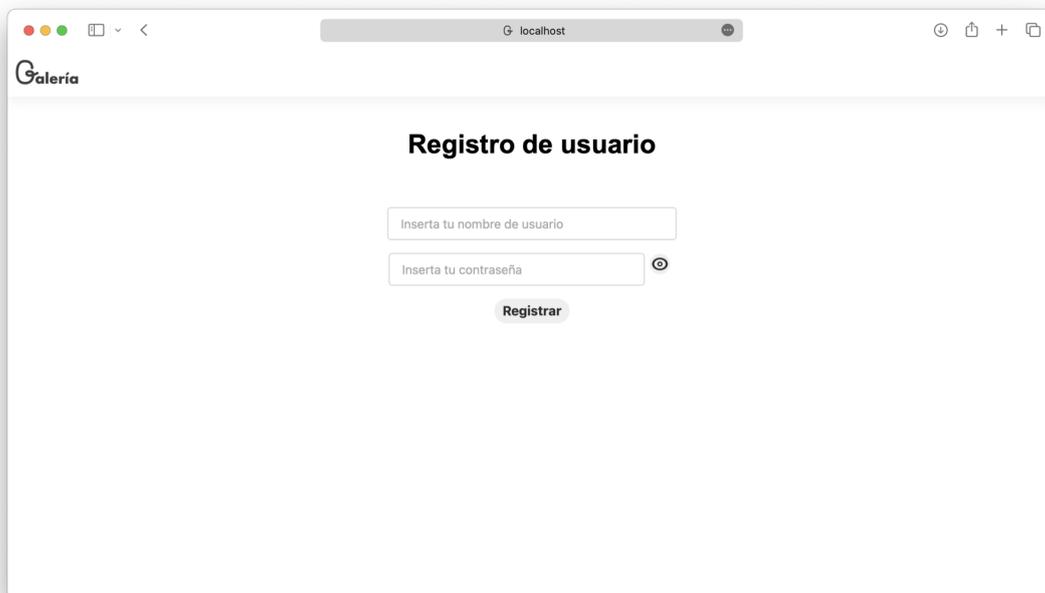


Figura 7.19: Registro de un visitante.

Ahora, tras el registro o inicio de sesión, el visitante podrá realizar sus tareas según los casos de uso de este tipo de usuario. Una de ellas y tras este proceso de iniciar la sesión, el usuario verá una página como la siguiente

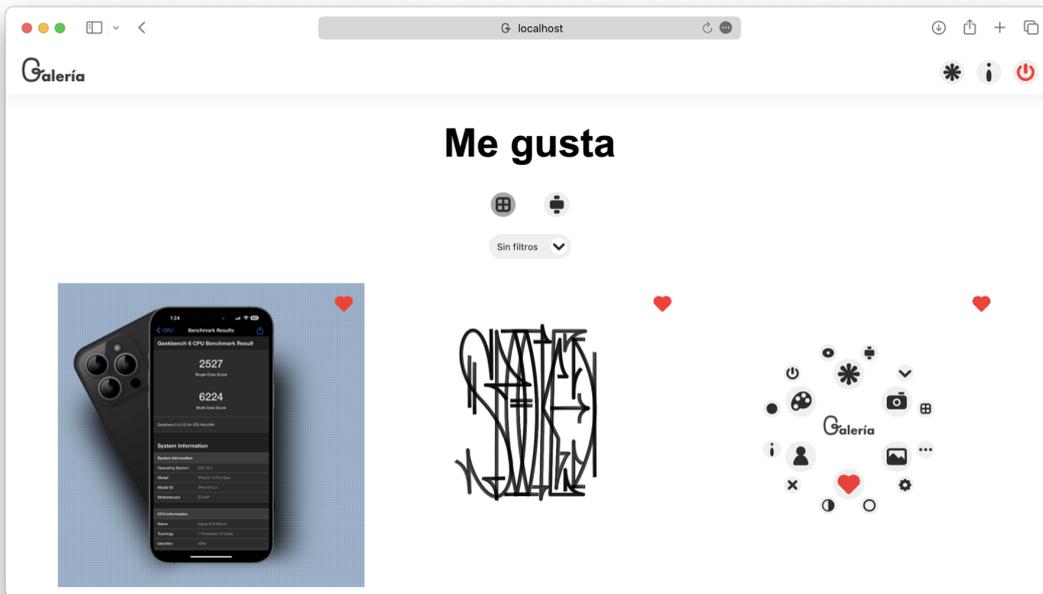


Figura 7.20: Página del espacio del visitante.

En la cual podrá visualizar todas aquellas obras que ha marcado como favoritas en otras ocasiones, además de buscarlas a través de las opciones de filtrado (figura 7.21), también podrá deshacer estas acciones desmarcando el botón de 'like' correspondiente.



Figura 7.21: Opciones de filtrado de obras.

Por otro lado, el visitante encontrará todo el catálogo de exposiciones y obras en la sección de exposiciones que se accede también a través de la barra superior de navegación. Una vez dentro, el usuario visualizará una lista de exposiciones con algunos detalles para así poder decidir cuál contemplar (figura 7.22).

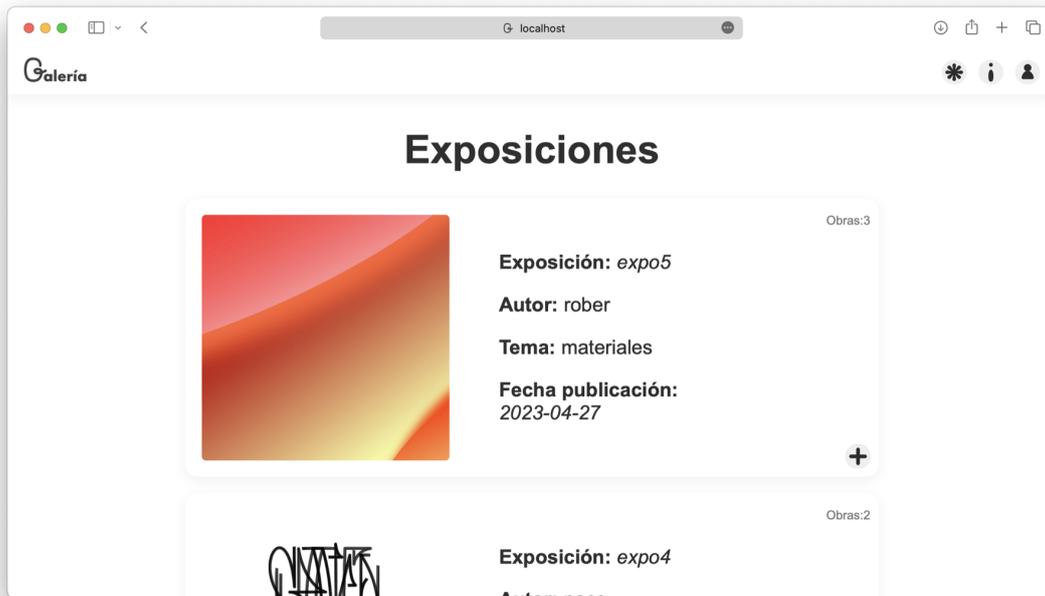


Figura 7.22: Vista de las exposiciones de la galería.

Por último, al entrar dentro de una exposición, podrá ver el resto de detalles de la misma junto con sus respectivas obras, con las que podrá ver cuáles de ellas ya había marcado con un 'like', marcar nuevas como favoritas o desmarcar esta opción.

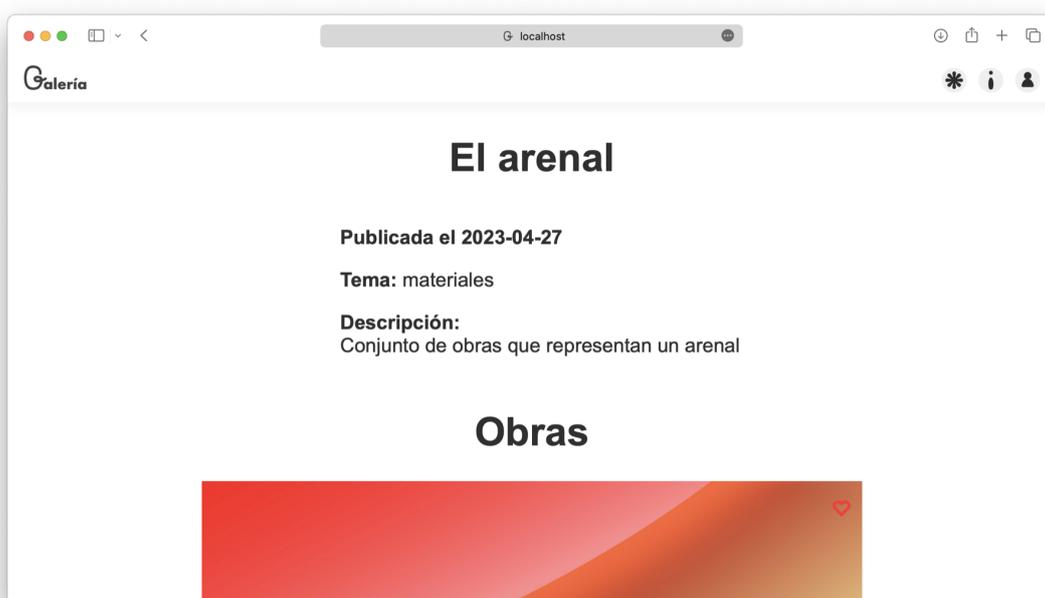


Figura 7.23: Vista de detalle de una exposición.

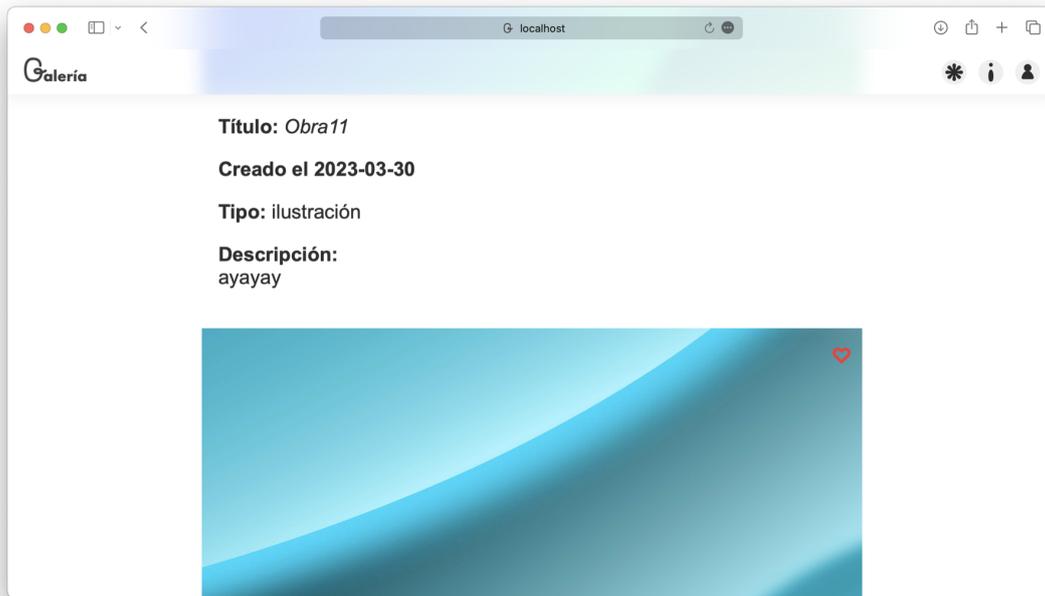
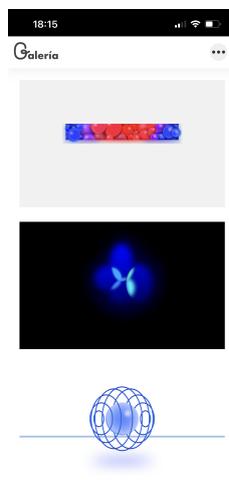


Figura 7.24: Vista de las obras de una exposición.

7.4 Interfaces móviles

Como bien es sabido, los dispositivos móviles son un elemento esencial en la vida de las personas a la hora de llevar a cabo las tareas del día a día, incluso para algunos supone su herramienta de trabajo con la que se ganan la vida. Es por ello que se ha adaptado toda la interfaz a una versión móvil para que sea accesible en más dispositivos y más empleados puedan seguir trabajando desde donde sea. Adicionalmente, otro de los motivos es el hecho de que el teléfono móvil es el dispositivo más utilizado de entre los demás en cuanto al mercado de 'masas' se refiere, por ello es una oportunidad que no se quería dejar escapar.



(a) Vista de inicio de la web en móvil.



(b) Vista de opciones de navegación en móvil.

Figura 7.25



(a) Vista de inicio de sesión móvil.



(b) Vista de inicio de sesión móvil (2).

Figura 7.26

7.4.1. Interfaz empleado artístico

Aquí se presentan un par de vistas que serían la de inicio y la de creación de obra, pudiendo visualizar cómo se ha adaptado el formulario o la selección de opciones a una versión móvil.



(a) Vista de inicio del empleado en móvil.



(b) Vista de formulario adaptado para móvil.

Figura 7.27

7.4.2. Interfaz empleado administrativo

Como el empleado de administración presenta gran parte de elementos que contiene también el empleado artístico en su interfaz, sólo se muestra la siguiente figura:

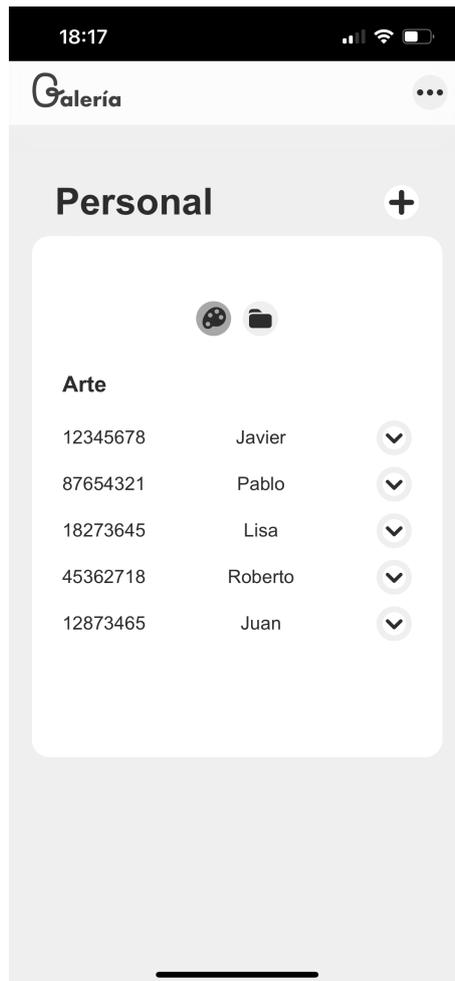
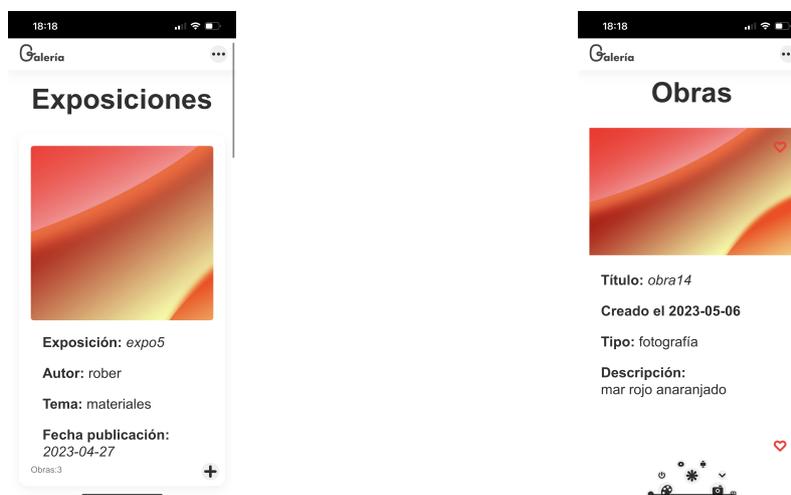


Figura 7.28: Vista del personal del administrador en móvil.

7.4.3. Interfaz usuario visitante



(a) Vista de exposiciones en móvil.

(b) Vista de las obras de una exposición en móvil.

Figura 7.29

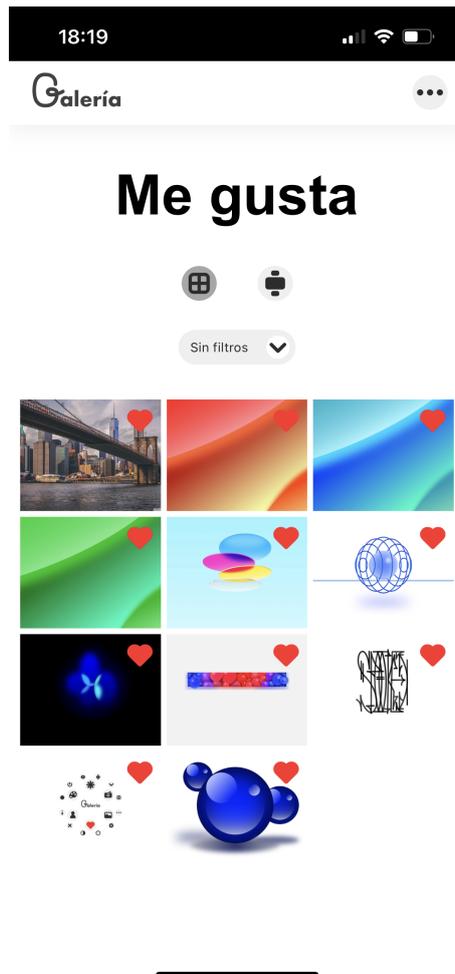


Figura 7.30: Vista del espacio del visitante en móvil.

CAPÍTULO 8

Validación

La validación vendría a ser un proceso de análisis que puede contener algún tipo de prueba, ya sea experimental o de otro tipo más teórico, que sirve para evaluar un sistema o aplicación y así determinar la calidad y aptitud del mismo. En este capítulo se explicarán dos métodos de evaluación de la aplicación, uno más informal y cercano al usuario y otro más teórico que sigue los principios de usabilidad de Jakob Nielsen.

8.1 Método 1º: Validación mediante prueba experimental

Como ya se ha mencionado, esta validación se centra en ir, en parte, de la mano del usuario para recibir una opinión del que podría ser el cliente final. El procedimiento seguido consiste en asignar un conjunto de tareas que sean suficientes para que el usuario de pruebas realice un recorrido completo de la aplicación y en función de su rol, en el que más adelante completará un formulario de valoración de la web y por último darán su opinión en forma de comentarios de manera presencial (en persona o a través de una llamada).

Un usuario en pasar por este proceso es el visitante, que en primera instancia ha realizado las siguientes tareas:

1. Visualizar algunas obras y exposiciones
2. Indicar como 'me gusta' alguna obra
3. Desmarcar como favorita alguna obra
4. Iniciar sesión
5. Buscar alguna obra gustada con los filtros ofrecidos en el espacio del usuario en la web
6. Cerrar sesión

Tras ello, el usuario completó un cuestionario, centrado en la usabilidad y con la posibilidad de proporcionar una opinión personal. A continuación se presenta el cuestionario.

El cuestionario trata de evaluar la usabilidad de la aplicación para el usuario visitante a través de una serie de preguntas que se pueden ver en el apéndice D. Comentando los resultados por encima, las cuatro primeras cuestiones, que tratan de averiguar si el usuario ha sido capaz de realizar sus tareas, han resultado en el cien por cien de completitud, es decir, todos los usuarios han podido completar las tareas asignadas. Por otro lado, las

siguientes cuestiones, que tratan de averiguar el grado de conformidad de los usuarios respecto a cómo han podido realizar dichas tareas, han surgido más variadas pero con un resultado positivo. A continuación se muestran los resultados de éstas últimas preguntas:



Figura 8.1: Encuesta usuario visitante (1).

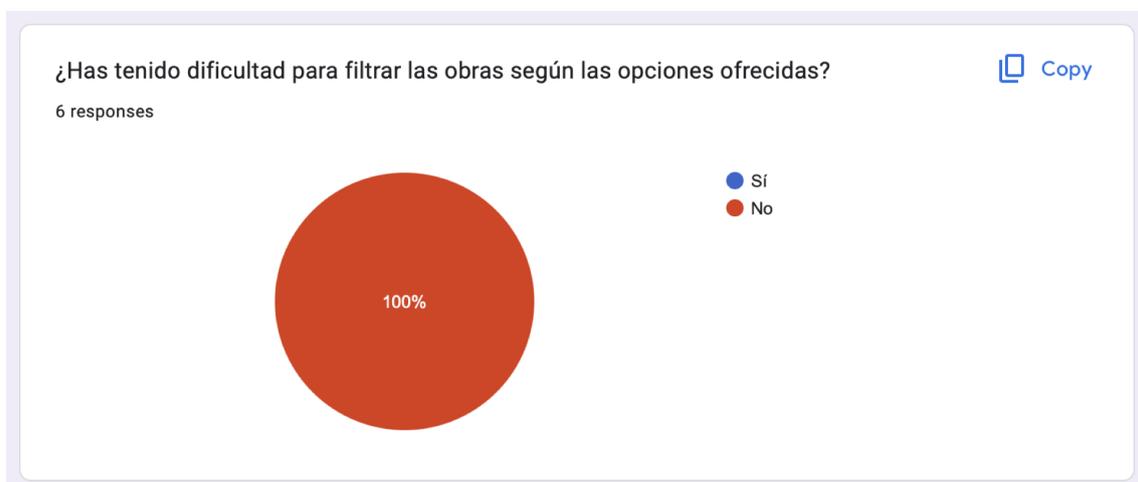


Figura 8.2: Encuesta usuario visitante (2).



Figura 8.3: Encuesta usuario visitante (3).

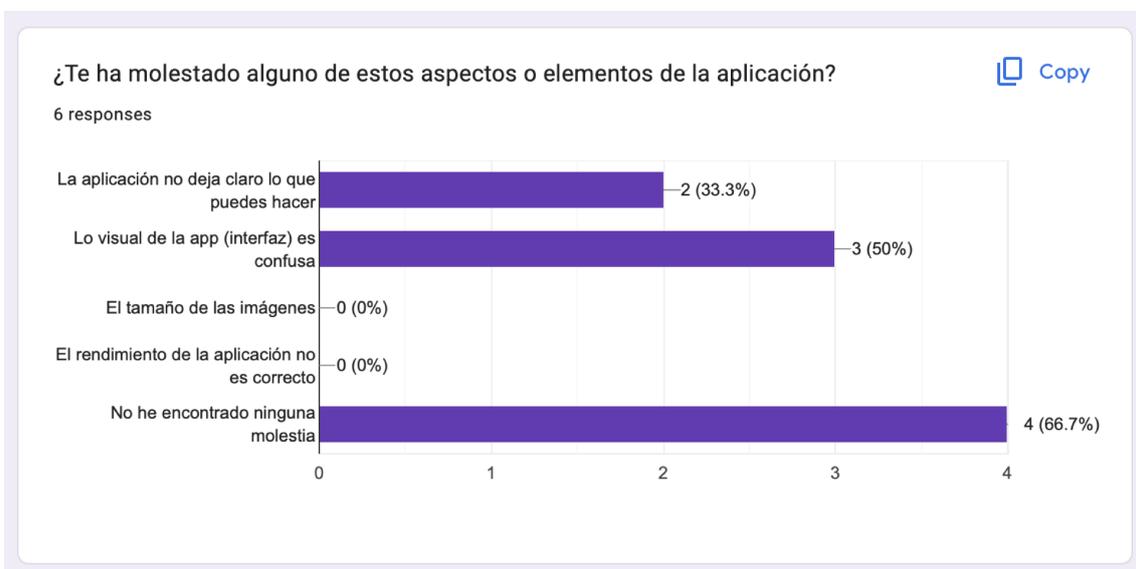


Figura 8.4: Encuesta usuario visitante (4).

Puediendo concluir un resultado mayormente positivo en cuanto a la eficacia de la aplicación, pero que hay que tener en cuenta que algún usuario sí se ha encontrado un poco desubicado realizando sus tareas. Lo que se vuleve a reflejar en los post comentarios que se podrían resumir de la siguiente manera: 'la aplicación está muy bien una vez te acostumbras a las acciones que puedes realizar'.

Por otro lado, se encuentra el usuario empleado, que deberá completar las siguientes tareas:

1. Iniciar sesión
2. Observa los detalles de una obra u exposición
3. Descataloga una obra
4. Crea una obra
5. Recupera una obra descatalogada

6. Cerrar sesión

Y tras completar las tareas, el empleado pasa por hacer el formulario que se puede ver en el apéndice E. Las primeras dos tareas las han podido realizar perfectamente todos los usuarios encuestados. En cambio la tercera tarea ha resultado en que la mayoría sí ha podido completarla pero que algún usuario ha tenido alguna dificultad para realizarla (ver figura 8.5).

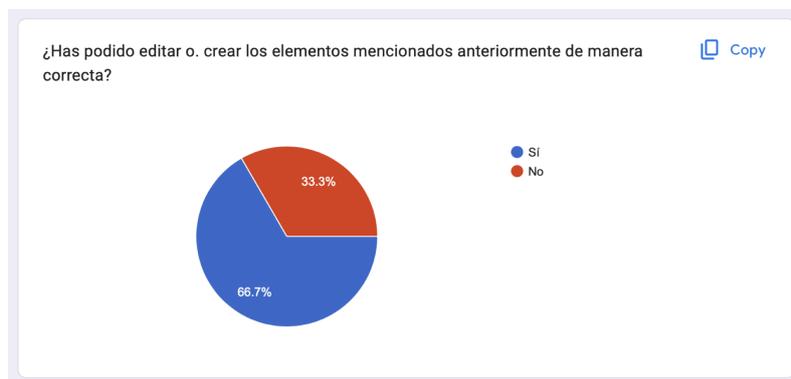


Figura 8.5: Resultados de la tarea 3 del formulario empleado.

Todos los empleados han podido navegar correctamente entre los menús y encontrar lo que buscaban. En cambio, puesto que anteriormente se ha visto que algún usuario no ha podido completar una tarea, sí ha necesitado ayuda para completarla.



Figura 8.6: Pregunta sobre la ayuda recibida del usuario.

Acabando con la última cuestión de respuesta múltiple y comentarios de los empleados (ver pregunta en la figura 8.7), se puede concluir algo muy similar que con el usuario visitante: se han podido completar las tareas pero hay que revisar la interfaz para no dejar de lado a ningún usuario.

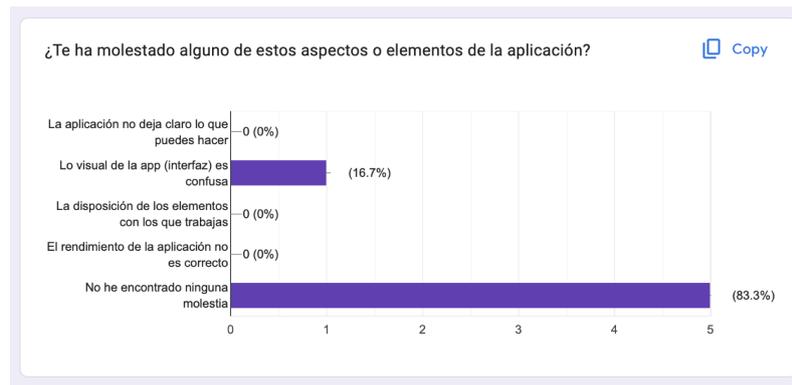


Figura 8.7: Pregunta de respuesta múltiple para el empleado.

8.2 Método 2º: Validación teórica

En esta sección se habla de una validación más teórica y que consiste en determinar la usabilidad por medio de unos principios heurísticos, en concreto los de Jakob Nielsen [4]. Estos principios son:

1. Visibilidad del estado del sistema: aquí el usuario debe estar al tanto de qué le ocurre al sistema. Aunque se dan ciertas indicaciones como en la figura 7.2 o la figura 8.8 en la barra superior de navegación, según los comentarios recibidos en la validación anterior, parece ser que no ha sido suficiente.

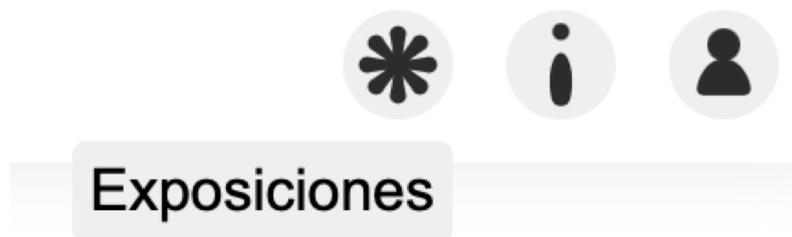
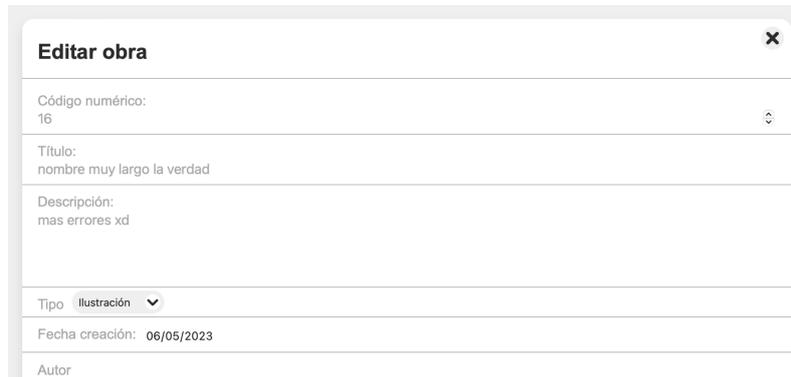


Figura 8.8: Indicaciones contextuales de los botones de la barra de navegación.

2. Coincidencia entre el sistema y el mundo real: el sistema debe mostrar un uso del lenguaje y elementos naturales y lógicos para el usuario. Aquí sí se ha tratado de diseñar centrado en el usuario tanto visitante como empleado. Para el primero se han utilizado elementos que ya conoce y se encuentran en el resto de redes sociales de la actualidad, y para el empleado se utilizan palabras o elementos en concordancia a las creaciones que gestionará en la aplicación.
3. Control y libertad para el usuario: puesto que de momento la aplicación no es muy compleja, los controles de navegación entre páginas o elementos están formados por los propios del navegador (atrás, adelante, recargar, etc) y otros de la web como

los ofrecidos en la barra de navegación u otros como la típica cruz de salir de un contexto (figura 8.9).



The image shows a modal window titled "Editar obra" with a close button (X) in the top right corner. The form contains the following fields:

- Código numérico: 16
- Título: nombre muy largo la verdad
- Descripción: mas errores xd
- Tipo: Ilustración (dropdown menu)
- Fecha creación: 06/05/2023
- Autor: (empty field)

Figura 8.9: Control de 'quitar' elemento.

4. Consistencia y estándares: según este principio se debe mantener a lo largo del sistema los mismos patrones de acciones para evitar que el usuario tenga que redescubrir varias maneras de hacer lo mismo. Un ejemplo de consistencia en la aplicación es el uso de un estilo de formularios similar para que el usuario tenga en cuenta siempre que ese elemento se utiliza para introducir datos (ver figura 7.5 Y 7.7 2). Otro ejemplo sería el uso de botones tanto con texto como con iconografía y que están contruidos de manera que el usuario tiene en cuenta que al hacer click sobre alguno de ellos podrá realizar una determinada acción, para ello manteniendo un mismo estilo, disposición y tamaños para su distinción.
5. Ayuda a los usuarios al reconocimiento, diagnóstico y recuperación de errores: permitiendo al usuario entender las causas del error y el apoyo para que lo pueda solucionar. Para ello, aunque de manera insuficiente, cuando el empleado se equivoca rellenando el formulario e intenta guardar los cambios de una obra, se le mostrará una alerta como la de la figura 8.10.

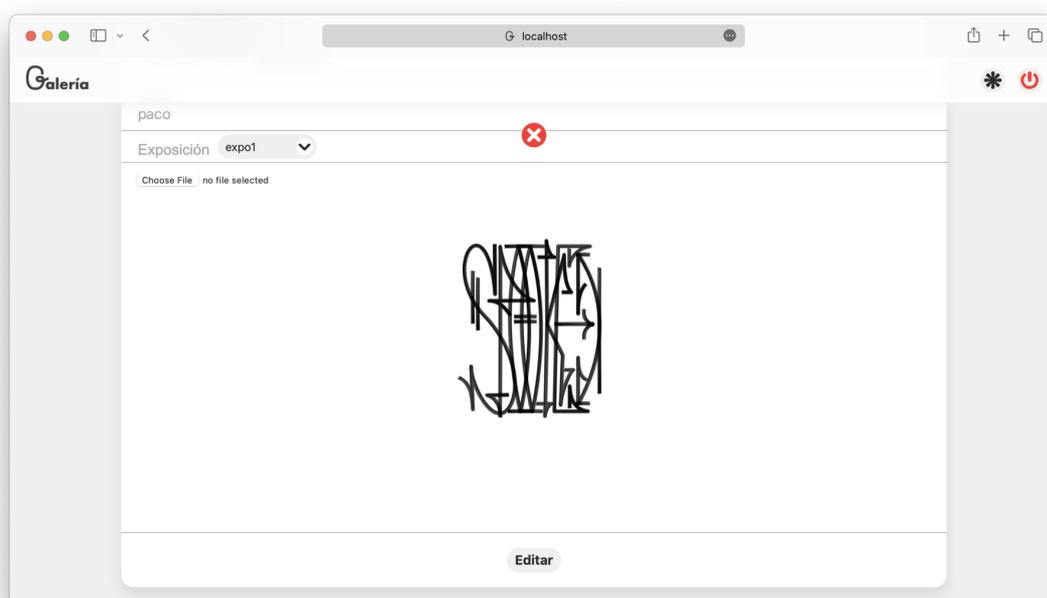


Figura 8.10: Alerta de error al guardar cambios en un formulario.

Por otro lado, en el inicio de sesión sí hay un mayor control de los errores y por tanto hay mejores indicaciones de por qué el usuario no puede iniciar su sesión. Las figuras siguientes ejemplifican este manejo de errores:

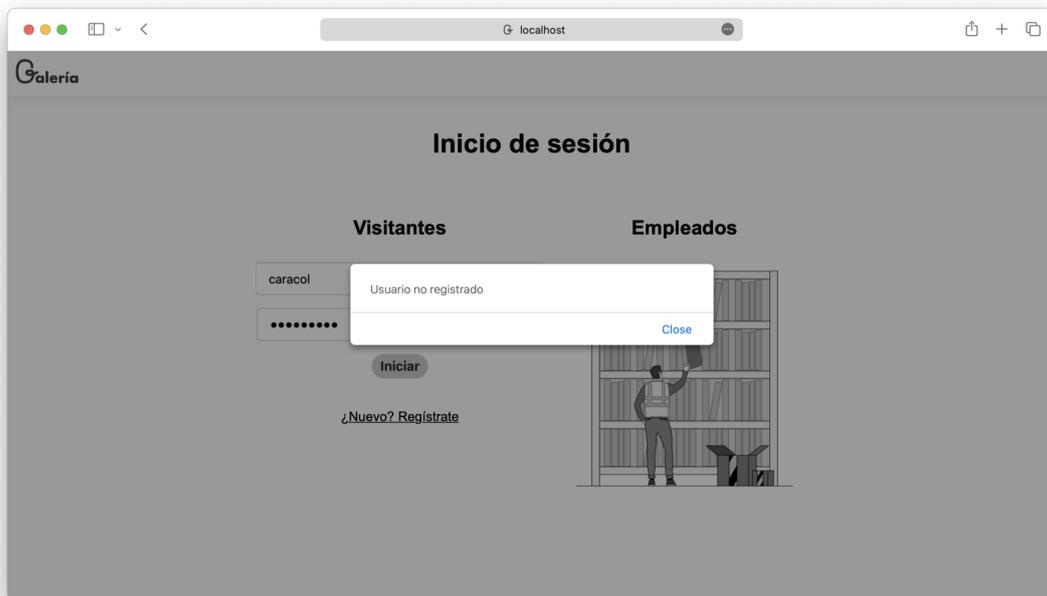


Figura 8.11: Alerta de error al iniciar sesión con usuario no registrado.



Figura 8.12: Alerta de error al iniciar sesión con contraseña incorrecta.

Por tanto, cabría mejorar el sistema de alertas cuando el empleado se equivoque rellenando un formulario, puesto que aunque tenga el conocimiento de cómo son las propiedades de una obra, puede cometer errores igualmente.

6. Prevención de errores: tratar de evitar que surgan errores aunque se hayan creado herramientas de apoyo al usuario una vez éstos ocurren. Como se han visto en los formularios, se hace uso de 'placeholders' en los campos de entrada de datos para indicar al usuario qué información debe indicar en cada campo. Un ejemplo concreto es la indicación de 'Código numérico' para hacer saber al usuario que el código de la obra es un valor numérico.
7. Reconocimiento antes que memoria: el usuario no debería estar recordando dónde se encuentra, de dónde viene o incluso qué acción puede realizar con las opciones que se le ofrece. Para cumplir con este principio se ha hecho uso de una iconografía en botones que ayudan a describir qué acción puede realizar o a dónde puede acceder el usuario.
8. Flexibilidad y eficiencia de uso: el sistema debe tener en cuenta a todo tipo de usuarios, tanto primeros visitantes como usuarios habituales. Por ello se ha hecho uso de elementos familiares como uso de iconos comunes a plataformas web populares y actuales.
9. Estética y diseño minimalista: se debe incluir lo estrictamente necesario, para que lo irrelevante no pueda confundir al usuario. Según las opiniones recibidas en la validación anterior, los usuarios consideran que la aplicación es minimalista, encontrando, tras familiarizarse con la interfaz, a mano las cosas necesarias para completar sus tareas. Además, cabe destacar el uso del color blanco en abundancia y la separación de elementos para tratar de evitar sobrecargar de información al usuario, mostrando con pocas palabras o símbolos el contexto necesario al usuario.
10. Ayuda y documentación: debe existir algún tipo de documentación que sirva de ayuda al usuario y le proporcione conocimiento acerca de cómo utilizar la aplicación. Por falta de tiempo y considerando que la aplicación no es un sistema demasiado complejo, no se ha podido crear una que sea fácil de buscar, no sea extensa y se centre en las tareas del usuario.

CAPÍTULO 9

Conclusiones y trabajo futuro

Como conclusión es interesante mencionar cómo, gracias a los conocimientos obtenidos en el grado cursado, se han hecho uso de los mismos para el desarrollo de una aplicación web. Conocimientos que van desde los técnicos, como los referidos a los que permiten manejar los distintos componentes de la arquitectura de la solución y también los que permiten el desarrollo de la aplicación, hasta aquellos conocimientos teórico prácticos que han permitido el análisis de información, uso de metodologías y planificación del proyecto para llevarlo a cabo de una forma organizada y con eficacia.

Por lo tanto, el objetivo de este trabajo es el de aportar valor más allá de realizar un ejercicio práctico que demuestre la capacidad de aplicar los aprendizajes adquiridos. Así pues, tras realizar un estudio de mercado, se ha llegado a una propuesta que no sólo ofrece un escaparate interactivo al usuario visitante, sino que también otorga una herramienta de creación de contenido a los empleados de la galería. Es decir, se ha utilizado un componente social y otro productivo como elementos diferenciadores de la propuesta respecto al resto de la competencia.

En cuanto al éxito de la aplicación, éste será posible sólo si la solución desarrollada va de la mano del usuario objetivo y que así pueda realizar sus tareas de la forma más natural y productiva posible. Para ello se ha realizado un análisis de los usuarios finales de la aplicación web y de sus requisitos. Luego, se han creado las interfaces con las que el usuario interactuará teniendo siempre en cuenta las tareas que éstos llevarán a cabo y finalmente, se ha construido la solución mediante los conocimientos técnicos adquiridos y desarrollando los distintos componentes de la arquitectura del servicio web como se ha visto en el capítulo 6.

Adicionalmente, atendiendo a las necesidades actuales de las personas que hacen uso de Internet, redes sociales y otras herramientas digitales, se ha creado también una versión adaptada a teléfonos y tabletas inteligentes, para así cubrir a un mayor número de potenciales usuarios de la aplicación. Ver apéndice XXXXXXXX.

Puesto que se ha presentado la primera versión de esta aplicación en el capítulo 7 y tras su validación en el capítulo 8, aún quedan por hacer muchas cosas interesantes, como ampliaciones o mejoras. Principalmente se pueden divisar dos tipos de mejoras o trabajo futuro: de ámbito técnico y de ámbito funcional.

En cuanto a las mejoras técnicas, que se refieren a las tareas de análisis e implementación de mejoras en los distintos componentes de la arquitectura del sistema, se encuentran las siguientes:

1. Uso de frameworks tanto en 'front-end' como en 'back-end' para una mejor estructura y organización del proyecto, además de las funcionalidades y productividad que puedan aportar.

2. Indexación de tablas en base de datos y aplicar más seguridad para evitar el ataque conocido como inyección SQL.
3. Uso de paginación para mejorar el rendimiento de la web, sobre todo para las interfaces del usuario empleado.
4. Uso de caches para almacenar información que el cliente (navegador) solicita frecuentemente
5. En vez de solicitar imágenes una a una de forma secuencial, hacer una petición que englobe un conjunto de imágenes y si hace falta apoyarse en la paginación

Por otro lado, en relación a las mejoras funcionales, que hacen referencia a la creación de nuevas herramientas o utilidades que mejoran la experiencia de uso de los visitantes y empleados como también la productividad de los mismos, se encuentran las siguientes:

1. Uso de más opciones de filtrado para concretar mejor la búsqueda tanto para visitantes como para empleados.
2. Uso de un buscador para acortar los tiempos de encontrar el elemento deseado mediante una búsqueda directa.
3. Más herramientas de personalización en la creación de contenido.
4. Colaboración en tiempo real entre empleados para la creación o edición de contenido.
5. Crear perfiles públicos o privados para los visitantes y crear un sistema de seguidores para la visualización de obras gustadas entre seguidores.
6. Indicaciones contextuales sobre las acciones que se pueden realizar.

Finalmente, como opinión, se han conseguido unos resultados esperados y que se han medido en el capítulo de validación. También, el trabajo realizado ha permitido reforzar conocimientos adquiridos en el grado cursado, como el desarrollo web, la integración de aplicaciones o el diseño centrado en el usuario, entre otros. Para cerrar las conclusiones, comentar la satisfacción obtenida tras completar el proyecto realizado el cual, aún siendo una primera versión, ya es posible desplegarla y utilizarla como un producto terminado y útil para el usuario final.

Bibliografía

- [1] Demián Gutierrez. *Métodos de Desarrollo de Software*. Universidad de los Andes, 2011.
- [2] A. Weitzenfeld. *Ingeniería del Software OO con UML. Java e Internet*. Thomson, 2005.
- [3] desarrolloweb.com. Manual de php. 2016.
- [4] José Juan Hernández Mora María Janaí Sánchez Hernández Higinio Nava Bautista César Alberto Collazos Ordóñez Julio Ariel Hurtado Alegría Guadalupe García Toribio, Yesenia Polvo Saldaña. *Medición de la usabilidad del diseño de interfaz de usuario con el método de evaluación heurística: dos casos de estudio*. 2019.

APÉNDICE A

Objetivos de Desarrollo Sostenible

La Organización de las Naciones Unidas aprobó en 2015 la Agenda 2030 sobre el Desarrollo Sostenible, con el objetivo de mejorar la vida de todas las personas y promover la sostenibilidad. Esta Agenda consta de 17 Objetivos de Desarrollo Sostenible (ODS) que abarcan áreas como el crecimiento económico, el bienestar social y la protección del medio ambiente. La Universitat Politècnica de València reconoce su responsabilidad social y busca contribuir a la consecución de estos objetivos desde sus distintas áreas de acción. A continuación se presentan dichos objetivos:

1. Fin de la pobreza
2. Hambre Cero
3. Salud y Bienestar
4. Educación de Calidad
5. Igualdad de género
6. Agua limpia y saneamiento
7. Energía asequible y no contaminante
8. Trabajo decente y crecimiento económico
9. Industria, innovación e infraestructura
10. Reducción de las desigualdades
11. Ciudades y comunidades sostenibles
12. Producción y consumos responsables
13. Acción por el clima
14. Vida submarina
15. Vida de ecosistemas terrestres
16. Paz, justicia e instituciones
17. Alianzas para lograr objetivos.

Es necesario marcar el grado de relación de este proyecto con los Objetivos de Desarrollo Sostenible, que se pueden observar en la tabla siguiente:

ODS	Alto	Medio	Bajo	No procede
1 Fin de la pobreza				x
2 Hambre Cero				x
3 Salud y Bienestar			x	
4 Educación de Calidad	x			
5 Igualdad de género				x
6 Agua limpia y saneamiento				x
7 Energía asequible y no contaminante				x
8 Trabajo decente y crecimiento económico	x			
9 Industria innovación e infraestructura		x		
10 Reducción de las desigualdades				x
11 Ciudades y comunidades sostenibles			x	
12 Producción y consumos responsables				x
13 Acción por el clima				x
14 Vida submarina				x
15 Vida de ecosistemas terrestres				x
16 Paz, justicia e instituciones	x			
17 Alianzas para lograr objetivos			x	

De entre estos objetivos sí se ha podido lograr al menos tres de ellos, los cuales se presentan a continuación:

1. Educación de Calidad, puesto que considero que el trabajo de una galería y su influencia son totalmente positivas para impulsar la cultura de la población, además de enriquecerla. Esto es debido a la intencionalidad puramente informativa, y que no impide la publicación de obras, tras una revisión de calidad, en la galería. Tampoco se permitiría el uso de la censura, pues iría en contra de los principios de divulgación de la galería.
2. Trabajo decente y crecimiento económico, como bien se comenta previamente en el trabajo, éste trata de incluir a todo tipo de usuarios, tanto a jóvenes como adultos, pues todos pueden interactuar con la aplicación sin la necesidad de poseer conocimientos en desarrollo web. Por otro lado, esta plataforma web trata de promover las actividades creativas propias del sector, lo que es muy positivo para su crecimiento y divulgación.
3. Paz, Justicia e Instituciones sólidas, como he dicho previamente, este proyecto aporta de conocimiento a sus visitantes, lo que permite mantener a la población más informada del tema y no causar daños como prejuicios o discriminación hacia el sector.

El resto de objetivos considero que no proceden analizar y por tanto relacionar con este proyecto debido a las características del mismo, el cual, como ya se ha mencionado, busca expandir y comunicar el arte a la población, lo que supone ser un objetivo meramente informativo. Podría haber un efecto positivo y en segundo plano que se enmarque en alguno de esos objetivos, como por ejemplo el de 'Industria innovación e infraestructura', ya que al promover este tipo de conocimiento sería posible que se estuviera impulsando también la propia industria del arte.

APÉNDICE B

Caso de estudio

Se desea gestionar una modesta galería de arte en formato digital, en la cual participarán dos tipos de usuarios, los visitantes y los empleados del servicio, los cuales se encargan de gestionar los contenidos de la misma.

La galería es un espacio donde se muestran obras artísticas, a través de imágenes, de muchos tipos, como fotografías, ilustraciones, arte abstracto, etc. que contienen un nombre de título, una breve descripción, tipo de obra, la fecha en la que se creó y el nombre del autor que la creó. Toda obra se muestra como parte de una exposición que contiene varias de las mismas con una temática en común, por lo que podemos esperar un nombre de título, un nombre de temática, la fecha en que se creó y una breve descripción. La galería puede mostrar todas las exposiciones catalogadas en la sección “colecciones” y se pueden filtrar por temática.

Tanto las obras como las exposiciones han de ser catalogadas previamente para su muestra en la galería. Así mismo, si a la galería no le interesa mostrar una obra o exposición, la podrá descatalogar si así desea. En caso de descatalogar una obra, ésta ya no se mostrará en la exposición correspondiente. En cambio, al descatalogar una exposición esto provocará que ya no se puedan visualizar las obras que contenía, ni ella misma.

Estas últimas tareas de gestión de los contenidos de la galería serán realizadas por los empleados de la misma, los cuales deberán proporcionar un nombre, un teléfono, un dni, el NSS, n° de cuenta en el que se les asigne la nómina, fecha de contratación y salario mensual. Para poder trabajar dentro de la aplicación, los empleados serán dados de alta de forma interna por un empleado administrativo. Todos los empleados iniciarán sesión mediante un nombre de usuario y una contraseña asignados en el proceso de alta del empleado.

Los visitantes, además de navegar por las exposiciones para visualizar las obras, podrán registrarse debiendo proporcionar un correo electrónico, un nombre de usuario y una contraseña. El usuario podrá ir a “su espacio” y ver las obras que ha indicado como preferidas (“me gusta”), además podrá filtrarlas por preferencias de búsqueda como son la fecha, el tipo de obra o por exposición. Además podrá indicar que quiere suscribirse al boletín de noticias de la galería a través del correo o si prefiere ser notificado desde la propia aplicación.

APÉNDICE C

API Obra

OBRA

METHOD: GET

```
path: /rest/obra/obra.php
description: Obtiene todas las obras catalogadas
response: 200 OK {
  "code": 1,
  "data": [
    {
      "cod_o": 3,
      "titulo": "Obra3",
      "descripcion": "Prueba",
      "tipo": "abstracto",
      "fecha_creacion": "2023-03-28",
      "autor": "test",
      "extension": "jpg",
      "exposicionid": 1,
      "exposicion": "expo1"
    }, ...
  ]
}
```

```
path: /rest/obra/obra.php?id=
parameter: id
description: Obtiene la obra según el identificador
response: 200 OK {
  "code": 1,
  "data": {
    "cod_o": 3,
    "titulo": "Obra3",
    "descripcion": "Prueba",
    "tipo": "abstracto",
    "fecha_creacion": "2023-03-28",
    "autor": "test",
    "extension": "jpg",
    "exposicionid": 1,
    "exposicion": "expo1"
  }
}
```

```
    }

    path: /rest/obra/obra.php?user=
    parameter: user
    description: Obtiene las obras según creadas por el empleado 'user'
    response: 200 OK {
      "code": 1,
      "data": [
        {
          "cod_o": 3,
          "titulo": "Obra3",
          "descripcion": "Prueba",
          "tipo": "abstracto",
          "fecha_creacion": "2023-03-28",
          "autor": "test",
          "extension": "jpg",
          "exposicionid": 1,
          "exposicion": "expo1"
        }, ...
      ]
    }
  }
```

METHOD: POST

```
    path: /rest/obra/obra.php
    body:
      {
        "cod_o": 6,
        "titulo": "Obra6",
        "desc": "Prueba",
        "tipo": "fotografía",
        "fecha": "2023-03-30",
        "autor": "rst",
        "extension": "jpg",
        "portada": null,
        "exposicion": 2,
        "exposicionAntigua":5,
        "empleado":"rst"
      }
    dataType: 'JSON'
    description: Actualiza los detalles de una obra
    response: 200 OK {
      "code":1,
      "data":"Obra modificada correctamente"
    }
  }
```

METHOD: PUT

```
    path: /rest/obra/obra.php
    body:
      {
        "cod_o": 27,
```

```
"titulo": "Obra 27",
"desc": "Prueba",
"tipo": "fotografía",
"fecha": "2023-03-30",
"autor": "rst",
"extension": "jpg",
"portada": null,
"exposicion": 2,
"empleado": "rst"
}
dataType: 'JSON'
description: Inserta una nueva obra
response: 200 OK {
  "code": 1,
  "data": "Obra creada correctamente"
}
```

METHOD: DELETE

```
path: /rest/obra/obra.php?id=
parameter: id
description: Descataloga la obra según el identificador
response: 200 OK {
  "code": 1,
  "data": "Obra eliminada correctamente"
}
```

APÉNDICE D

Formulario usuario visitante

Opciones: si y no.

1. ¿Has podido ver adecuadamente las obras y exposiciones?
2. ¿Has podido iniciar sesión correctamente?
3. ¿Has podido marcar obras como favoritas? (Dar 'like')
4. ¿Has podido desmarcar obras como favoritas? (Quitar 'like')
5. ¿Has tenido dificultad para navegar por los menús para encontrar lo que buscabas?
6. ¿Has tenido dificultad para filtrar las obras según las opciones ofrecidas?
7. ¿Has necesitado ayuda para realizar alguna acción?

Opción múltiple: ¿Te ha molestado alguno de estos aspectos o elementos de la aplicación?:

1. La aplicación no deja claro lo que puedes hacer
2. Lo visual de la app (interfaz) es confusa
3. El tamaño de las imágenes
4. El rendimiento de la aplicación no es correcto
5. No he encontrado ninguna molestia

APÉNDICE E

Formulario usuario empleado

Opciones: si y no.

1. ¿Has podido iniciar sesión correctamente?
2. ¿Has podido ver elementos como empleados, obras o exposiciones correctamente?
3. ¿Has podido editar o. crear los elementos mencionados anteriormente de manera correcta?
4. ¿Has tenido dificultad para navegar por los menús para encontrar lo que buscabas?
5. ¿Has necesitado ayuda para realizar alguna acción?

Opción múltiple: ¿Te ha molestado alguno de estos aspectos o elementos de la aplicación?:

1. La aplicación no deja claro lo que puedes hacer
2. Lo visual de la app (interfaz) es confusa
3. El tamaño de las imágenes
4. El rendimiento de la aplicación no es correcto
5. No he encontrado ninguna molestia