



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Sistema de Información para la Gestión de las Actividades
del Museo de Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Olmeda Gerena, Carlos

Tutor/a: Marzal Calatayud, Eliseo Jorge

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Sistema de Información para la Gestión de las Actividades del Museo de Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Carlos Olmeda Gerena

Tutor: Eliseo Jorge Marzal Calatayud

Curso: 2022-2023

Resumen

Este trabajo de fin de grado se basa en la creación de una aplicación web asociada a una base de datos para la gestión y la mejora del funcionamiento del museo de informática de la UPV.

Para ello, se ha creado un sistema de información que contenga todos los datos relevantes a cerca de los elementos partícipes del museo y sus actividades. A este sistema, se le incorporará una aplicación web que permitirá visualizar esta información.

El resultado final se basa en un sistema de información vinculado a una web, que permitirá gestionar los elementos del museo de informática y las reservas que se produzcan. Se dispone, asimismo, de la posibilidad de llevar un mantenimiento histórico para obtener datos estadísticos.

Además del modo administrador, la aplicación web dispone de la opción de reservar una actividad como usuario. Esta reserva se podrá realizar mediante la cumplimentación de un formulario.

Con esto, se mejoraría notablemente el funcionamiento del museo de informática y, a su vez, el personal vería automatizado gran parte del proceso manual de reserva de actividades empleado actualmente.

Palabras clave: Wordpress, MySQL, creación, conexión, plugin, automatizado.

Resum

Aquest treball de fi de grau es basa en la creació d'una aplicació web associada a una base de dades per a la gestió i la millora del funcionament del museu d'informàtica de la UPV.

Per a això s'ha creat un sistema d'informació que continga totes les dades rellevants i tots els elements participes del museu i les seues activitats. A aquest sistema, se l'incorporarà una aplicació web que permetrà visualitzar aquesta informació.

El resultat final es basa en un sistema d'informació vinculat a una web, que permetrà gestionar els elements del museu d'informàtica i les reserves que es produïsquen. Es disposa de la possibilitat de portar un manteniment històric de per a obtindre dades estadístiques.

A més de tindre un mode administrador, l'aplicació web disposa de l'opció de reservar una activitat com a usuari. Aquesta reserva es podrà realitzar amb l'ompliment d'un formulari.

Amb això es milloraria notablement el funcionament del museu d'informàtica i el personal veuria automatitzat gran part del procés manual de reserva d'activitats actual.

Paraules clau: Wordpress, MySQL, creació, connexió, plugin, automatitzat.

Abstract

This end-of-degree focuses the creation of a web application associated with a database for the better management of the museum of informatics of the UPV.

To this end, an information system has been created that contains all the relevant data as soon as all the elements participate in the museum and its activities. To this system, a web application will be incorporated that will allow you to visualize this information.

The final result is based on an information system linked to a website, which will allow managing the elements of the computer museum and the reservations that occur. It is possible to carry out a historical maintenance to obtain data.

In addition to the administrator mode, the web application has the option to reserve an activity as a user. This reservation can be done with the fulfillment of a form.

This would significantly improve the operation of the museum of informatics and its staff would see most of the reservation process automated.

Key words: Wordpress, MySQL, creation, connection, plugin, automated.

Tabla de contenidos

1.	Introducción	11
1.1.	Motivación.....	12
1.2.	Objetivos	13
1.3.	Estructura del trabajo.....	13
2.	Estado del arte	15
2.1.	Creadores de Web	15
2.2.	SGBD.....	17
2.3.	Plugin para conectar web y base de datos.....	19
3.	Análisis del sistema.....	22
3.1.	Especificación de requisitos	23
3.1.1.	Requisitos funcionales.....	23
3.1.2.	Requisitos no funcionales	25
3.2.	Modelo de análisis propuesta.....	27
3.3.	Diseño de la solución.....	29
3.3.1.	Diseño lógico.....	29
3.3.2.	Diseño físico.....	34
3.3.3.	Diseño web.....	34
3.4.	Desarrollo de la solución.....	35
3.4.1.	Creación de las tablas del sistema de información.....	35
3.4.2.	Inserción de datos	37
3.4.3.	Creación de la web	40
3.4.4.	Conexión de la web con la base de datos.....	44
4.	Ejemplo de prueba	51
4.1.	Envío correcto de formulario	51
4.2.	Recepción correcta de formulario	52
4.3.	Visualización correcta de la información en modo administrador.....	53
4.4.	Creación de nueva reserva.....	54
5.	Conclusiones	56
5.1.	Cumplimiento de objetivos.....	56
5.2.	A nivel personal	56
5.3.	Limitaciones	57
6.	Bibliografía	59



7.	Anexo	62
7.1.	Anexo 1.....	62
7.2.	Anexo 2	64
7.3.	Anexo 3	68

Índice de figuras

Imagen 1: Ejemplo del sistema usado con el calendario para el registro de actividades	11
Imagen 2: Logo de MySQL.....	17
Imagen 3: Logo de PostgreSQL.....	18
Imagen 4: Logo de Oracle Database.....	19
Imagen 5: Logo de WordPress	15
Imagen 6: Logo de Nicepage	16
Imagen 7: Logo de Squarespace	17
Imagen 8: Logo de WP Data Access	20
Imagen 9: Logo de WP DataTables	20
Imagen 10: Versión pro de Ninja Tables	21
Imagen 11: Formulario actual para la creación de una reserva.....	22
Imagen 12: UML de la base de datos realizado con draw.io	28
Imagen 14: Interfaz de MySQL para la creación de tablas.....	35
Imagen 15: Tablas añadidas correctamente en el sistema MySQL.....	37
Imagen 16: Horarios de la clase “calendario” con sus respectivos días de la semana asociados	38
Imagen 17: Interfaz de WP Admin Data.....	40
Imagen 18: Inicio de sesión de la web con el editor de elementor	42
Imagen 19: Secuencia de inserción del documento sql	45
Imagen 20: Parámetros de búsqueda y de exportación de las tablas creadas por WP Data Access.....	45
Imagen 21: Información asociada a los monitores y responsables del museo de informática	47
Imagen 22: Como añadir nuevas filas a las tablas de WP Data Access.....	48
Imagen 23: Envío correcto de formulario de la web.	51
Imagen 24: Correo recibido con la información relativa al formulario rellenado a modo de ejemplo en el punto de “Envío correcto de formulario”	52
Imagen 25: Filtración de datos en las tablas	53

1. Introducción

El museo de informática de la ETS de Ingeniería Informática realiza diversas actividades (visitas guiadas o talleres) para los institutos, asociaciones e incluso grupos particulares que lo soliciten.

Actualmente las visitas al museo para la realización de dichas actividades se gestionan mediante formularios que deben rellenar los responsables del grupo visitante.

En ellos, el encargado del grupo deberá indicar el número de alumnos a participar, el/los cursos en los que estos se encuentran, el centro de procedencia, la actividad a realizar y la información personal del responsable.

Tras rellenar los formularios, estos son enviados por la persona a cargo del grupo al correo del museo de informática, y tras ser recibidos, se revisa esta información. Con ella, los gestores del museo reservarán los espacios y se crearán las credenciales en la UPV.

Cuando se reciben estas credenciales se dejan a disposición de los monitores para que las utilicen durante la realización de la actividad.

Para indicar cuando se debe realizar cada actividad, el aula a asignar y el monitor asociado al grupo, se hace uso del calendario de Google.

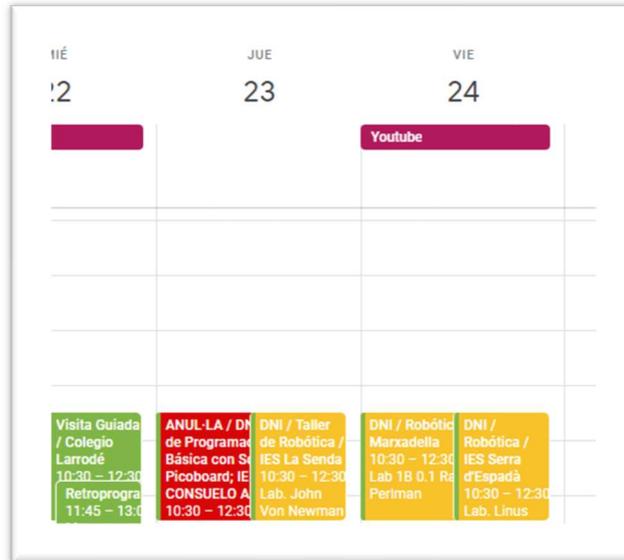


Imagen 1: Ejemplo del sistema usado con el calendario para el registro de actividades

Se marcará la actividad en amarillo mientras esté a la espera de información por enviar, en rojo si es anulada, y en verde si toda la información es correcta. Es entonces cuando se asigna un aula y un monitor.

Actualmente cuando hay una solicitud se realizan una serie de pasos:

1. Se busca un espacio adecuado para la actividad, hay actividades que requieren del uso de ordenadores, otras actividades requieren un aula para trabajo en grupo, etc.
2. Si hay espacio:
 - a. Se realiza la reserva utilizando la aplicación MONNENGRE de la UPV.
 - b. Se anotan los datos de la actividad en el calendario de la cuenta de Google del museo. Se utiliza color verde cuando la actividad está lista para su realización o color naranja cuando está pendiente de la recepción de información (listado de participantes para la creación de credenciales).
 - c. Hay actividades que son realizadas por algún profesor, en este caso se envía un correo al profesor para confirmar su disponibilidad en la fecha concreta. Si el profesor confirma su disponibilidad pasamos al apartado d. En caso negativo se pasaría al paso 3.
 - d. Se confirma al responsable de la solicitud de que su reserva se ha tramitado correctamente. Si se requiere información adicional se le solicita en este correo.
 - e. Se informa al profesor que tiene un taller (solo si es necesario).
3. Si no hay espacio, se pide al responsable que cambie de actividad o de día.

En este trabajo se tratará de cambiar la forma actual de reserva de actividades en el museo de informática a un modo más automático y práctico. Con esto se mejorará notablemente el procedimiento.

Se comenzará creando una base de datos con la información necesaria de los participantes en la reserva y los factores importantes en el proceso. Con esta base de datos, se podrán almacenar y gestionar la información de una forma mejor a la del uso de formularios y el calendario.

Se creará también una aplicación web que nos permitirá acceder como administrador y almacenar los datos correspondientes a las reservas. Esta aplicación será accesible desde cualquier dispositivo.

Con esto, la mejora en el sistema del museo de la informática será mucho mayor y se ahorrará esfuerzo a los trabajadores del centro para aprovechar el tiempo en otros factores importantes del museo.

1.1. Motivación

La motivación principal es la intención de mejorar la gestión del museo de informática.

Tras hablar con mi tutor, llegamos a la conclusión de que una mayor automatización en la creación de reservas, y en general, del proceso de registro de estas, traería muchos beneficios al museo. Por tanto, orientar mi trabajo de final de grado a mejorar y crear un sistema de información y una aplicación web para el museo fue una propuesta muy interesante a realizar.

En un primer momento, se planeó abordar de una manera ambiciosa el proyecto, y se intentó implantar en su totalidad el nuevo funcionamiento del museo. Esto es, un sistema de información complejo, con una aplicación web que permitiese diferentes tipos de inicio de sesión con sus respectivas características. Debido a que esto iba a llevar mucho más de las 120 horas de trabajo de un TFG, se decidió acortar el trabajo en cuanto a las funcionalidades de la web. Se decidió limitar la misma a un modo administrador y una posibilidad de reservar actividades, dado que al no haber nada creado para el proyecto planteado en primera instancia, lo principal sería crear el sistema de información y profundizar en la aplicación web en un futuro.

Los formularios y la asignación de eventos en el calendario permiten que se gestionen las actividades, pero con un sistema de información y una web asociada la mejora sería notable.

1.2. Objetivos

Los objetivos del proyecto serán mejorar la automatización de las reservas de actividades del museo de informática.

Se pretende crear una base de datos que contenga toda la información relevante en el proceso de registro de una solicitud y un histórico de las estadísticas que permitan obtener datos relevantes para el museo, como las aulas disponibles, los responsables y monitores, los grupos que intervienen, las actividades y las entradas sostenibles.

Con la aplicación web se pretende crear un sistema que permita autenticación de los gestores del museo para registrar la información necesaria y almacenarla.

1.3. Estructura del trabajo

La memoria está formada por los siguientes capítulos:

- Estado del arte: En esta parte se explican las alternativas adicionales a las seleccionadas para la realización del trabajo, y las causas por las que no fueron elegidas.
- Análisis del problema: Formado por la solución propuesta y su posterior diseño y desarrollo. En esta parte de la memoria se expondrán la mayoría de los conocimientos y pasos a realizar para llegar al resultado final del trabajo.
- Ejemplos de prueba: En esta parte se detallan los pasos a seguir para realizar las tareas principales para las que se ha diseñado la aplicación web, con su correspondiente demostración de que su funcionamiento es correcto.
- Conclusiones: Se detallan las conclusiones obtenidas tras la finalización del proyecto. Además, se detallan las limitaciones encontradas en el proceso.

2. Estado del arte

Para la realización de una base de datos hay una gran cantidad de opciones posibles. En la actualidad, la decisión en cuanto a la elección del programa seleccionado para crear un sistema de información, en la mayoría de los casos, se debe a la compatibilidad con las aplicaciones adicionales que intervengan en el proyecto o web a desarrollar.

En este apartado, se comentarán las razones por las cuales se han utilizado ciertos programas, y alternativas que en un principio se consideraron, pero que finalmente fueron descartadas.

2.1. Creadores de Web

En cuanto a los creadores de web, hay muchísimos en el mercado y cada uno dispone de una interfaz que se adapta mejor a diferentes tipos de páginas. La mayoría tienen una versión gratuita muy limitada, y para poder hacer cualquier cosa mínimamente avanzada se necesita la versión de pago. WordPress ofrece un servicio gratuito para crear una web en local, y la gran cantidad de plugin de los cuales dispone, han hecho de este sistema el elegido para la realización de este proyecto.

Los principales creadores de web analizados fueron:

1. WordPress

WordPress es una plataforma para crear webs muy populares. Dispone de una interfaz fácil de utilizar y muy gráfica. Permite un grado de personalización avanzado, en el cual es posible adaptar la edición a cualquier estilo de web deseado. Cuenta con una gran cantidad de plugins, los cuales facilitan ampliar esta personalización al máximo.

Gracias a estos plugins, es posible adaptar la web a distintos tipos de dispositivos electrónicos, generar formularios avanzados, conectar sistemas de información con webs, etc.

Por tanto, dadas estas características, WordPress es una gran elección para crear una web.



Imagen 2: Logo de WordPress

2. Nicepage

Nicepage es una herramienta de diseño y creación de sitios web que ofrece varias características útiles para ayudarte a construir sitios web de manera rápida y efectiva.

Proporciona un editor visual fácil de usar que permite diseñar y personalizar un sitio web de manera visual.

Ofrece una amplia selección de plantillas y bloques predefinidos que se pueden utilizar como punto de partida para el diseño, y permite crear sitios web que se adapten automáticamente a diferentes dispositivos y tamaños de pantalla.

Una característica muy importante es que los sitios web creados se pueden ver bien tanto en computadoras como en dispositivos móviles, lo que es esencial en el mundo actual donde la mayoría de las personas acceden a Internet desde sus teléfonos inteligentes y tabletas. La edición en tiempo real es posible, por tanto, se puede actualizar la web muy rápidamente.

Por todas estas virtudes, Nicepage es una gran elección para crear webs. El único problema es que para acceder a la mayoría de ellas hay que pagar (GPT, 2023).



Imagen 3: Logo de Nicepage

3. Squarespace

Squarespace es una plataforma de creación de sitios web que ofrece una serie de características y funcionalidades que permiten construir un sitio web atractivo y profesional.

Proporciona una amplia selección de plantillas elegantes y modernas diseñadas por profesionales y ofrece un editor intuitivo que te permite personalizar fácilmente el diseño y contenido.

Además, incluye una variedad de funcionalidades integradas, tales como herramientas de blogs, galerías de imágenes, integración con redes sociales, formularios de contacto, etc.

Puede verse perfectamente tanto en computadoras como en móviles y dispone de unas características óptimas para la seguridad de las webs creadas.

Con todo esto, Squarespace también es una buena opción, pero al igual que Nicepage, muchas de sus características solo se pueden disfrutar mediante la versión premium de pago (Enlinea.biz).



SQUARESPACE

Imagen 4: Logo de Squarespace

El creador web elegido finalmente para realizar el trabajo ha sido WordPress. La principal razón por la que se escogió WordPress fue que la página original del museo de informática está realizada en este sistema.

Dado que el objetivo a largo plazo sería “fusionar” la página original con la que se va a realizar, por temas de compatibilidad la mejor elección era usar WordPress para tal proceso. Otra razón fue el tema económico. Al ser gratuitas la mayoría de las funcionalidades de WordPress, esto contribuyó a que fuese el creador de web elegido.

2.2. SGBD

En un primer momento, se creó la base de datos usando PostgreSQL, y tras comprobar que era incompatible con WordPress, se cambió a MySQL.

Por tanto, algunos ejemplos para la creación de una base de datos son:

1. MySQL

MySQL es un sistema de gestión de bases de datos muy utilizado. Entre sus principales características, destaca el gran rendimiento que ofrece. Es posible manejar grandes cantidades de datos y presenta una gran compatibilidad con la mayoría de los sistemas operativos. Además, cuenta con muchas características avanzadas y una escalabilidad a cualquier tipo de sistema de información.

Su interfaz es muy clara y permite una familiarización muy rápida con las diferentes opciones del programa.

Por estas características, MySQL es un SGBD muy utilizado y recomendado.



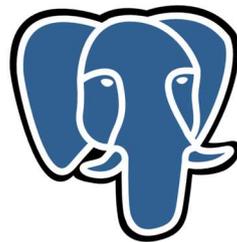
Imagen 5: Logo de MySQL

2. PostgreSQL

PostgreSQL ha sido diseñado para ser un sistema de gestión de bases de datos robusto y confiable. Su arquitectura interna y sus algoritmos de almacenamiento garantizan la integridad y consistencia de los datos. Dispone de una capacidad de manejar grandes volúmenes de datos y un gran rendimiento.

Además, ofrece una variedad de características de seguridad, como autenticación avanzada, cifrado de datos, control de acceso basado en roles, auditoría y encriptación, que son fundamentales para proteger la información confidencial.

Por tanto, estas características junto con la sencillez y la interfaz intuitiva que tiene PostgreSQL lo colocan como una gran opción a la hora de ser usado para crear una base de datos (Borges, 2019).



PostgreSQL

Imagen 6: Logo de PostgreSQL

3. Oracle Database

Oracle Database es conocido por su capacidad de manejar grandes volúmenes de datos y cargas de trabajo exigentes. Su arquitectura escalable permite distribuir la carga en múltiples servidores y utilizar técnicas como la partición de tablas y la indexación avanzada para optimizar el rendimiento de las consultas.

Ofrece una amplia gama de características de seguridad que ayudan a proteger los datos confidenciales y cumple con numerosas regulaciones y estándares de seguridad. También dispone de opciones de recuperación ante desastres, como copias de seguridad y recuperación, la cual es una característica realmente importante y útil.

Todos estos detalles hacen de Oracle Database un gran sistema para crear una base de datos. Su equipo de soporte y su comunidad son amplios, así que, en caso de duda, la ayuda disponible es muy grande y segura (Rootstack, 8).

ORACLE®
D A T A B A S E



Imagen 7: Logo de Oracle Database

De estos SGBD se eligió MySQL debido a que es totalmente compatible con WordPress. Al elegir Wordpress para crear la web asociada al sistema de información, la compatibilidad que tiene con MySQL en cuanto a los plugin fue el principal motivo de la elección.

2.3. Plugin para conectar web y base de datos

En cuanto a cómo conectar la web con la base de datos, para poder visualizar la información de esta en modo administrador, hay varias opciones disponibles en WordPress.

Las que más llamaron la atención fueron:

1. WP Data Access

WP Data Access es un plugin de WordPress que permite crear tablas a partir de un sistema de información.

Cuenta con un editor de tablas y formularios, un visualizador de información local, y múltiples opciones de conexión con bases de datos externas.

A diferencia de todos los demás plugins usados para este cometido, WP Data Access permite realizar la conexión con sistemas de información locales de forma gratuita. Esto ha hecho que haya sido el plugin elegido para realizar el trabajo.

El plugin cuenta con un editor visual fácil de usar y una gran cantidad de opciones para exportar información a diferentes formatos. No dispone de una opción de añadir información desde el visualizador de tablas en Web, pero permite añadir datos en tiempo real desde el administrador del programa.



Imagen 8: Logo de WP Data Access

2. WP Datatables

WPDataTables es un complemento de WordPress que permite crear y gestionar tablas de datos interactivas en un sitio web.

Dispone de un editor visual muy fácil de usar basado en elementos de arrastrar y soltar. Por tanto, su modo de uso resulta muy intuitivo. Ofrece una gran cantidad de plantillas muy variadas adaptables a muchos tipos distintos de tablas y diseños.

Permite, a su vez, una adaptabilidad tanto a computadoras como a dispositivos móviles. Se puede realizar la edición en línea.

Es posible conectarse con cualquier tipo de bases de datos.

Con esto WP Datatables es una gran opción. El único problema es que la única forma de conectar con bases de datos externas es pagando. La opción gratuita solo permite crear tablas (Ciroapp, 2023).



Imagen 9: Logo de WP DataTables

3. Ninja Tables

Ninja tables es un plugin muy usado en Wordpress para crear tablas profesionales y muy atractivas para un sitio web. Permite crear estructuras de datos de manera muy organizada y fácil de entender. Dispone de una interfaz muy intuitiva y fácil de usar y no requiere muchos conocimientos de programación para poder utilizarlo, ya que la creación de tablas se basa mayoritariamente en comandos de arrastrar y soltar los distintos elementos.

Pueden crearse tablas con un nivel de personalización muy grande y proporcionar funcionalidades avanzadas tales como la opción de añadir filas en tiempo real desde la web. Se pueden filtrar datos y exportarlos mediante múltiples opciones, como PDF, Excel, etc. (KME, 2018).

El problema de este plugin es que es de pago.



Imagen 10: En esta imagen se puede observar como la mayoría de sus funciones son con la versión pro. Por tanto, si no se puede realizar ninguna opción avanzada a menos que sea pagando, no merece la pena utilizar el plugin para un tfg sin presupuesto.

El plugin elegido fue WP Data Access, ya que, a diferencia de los otros plugin, permite realizar la conexión con bases de datos y la creación personalizada de tablas de forma gratuita.

3. Análisis del sistema

Para analizar el problema existente con la gestión de actividades del museo de informática, hay que observar los métodos actualmente usados para ello.

En primer lugar, el uso de formularios para la introducción de la información es demasiado manual. Sería beneficioso automatizar ciertas tareas para mejorar la productividad.

El Museo de informática por dentro

Actividades del Museo de Informática

Su historia es reciente, pero hoy sería impensable un mundo sin Informática. En la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València, la ETSINF – UPV, tenemos propuestas didácticas que pueden contribuir a que sus alumnos y alumnas conozcan mucho más de lo que imaginan sobre el pasado, presente y futuro de la Informática.

Nombre del centro u organización *	<input type="text"/>
Nombre de la persona responsable del grupo *	<input type="text"/>
Móvil de contacto *	<input type="text"/>
Email de contacto *	<input type="text"/>
Localidad *	<input type="text"/>
Código Postal *	<input type="text"/>
Titulación y curso del grupo *	<input type="text"/>
Actividad *	<input type="text"/>
Idioma de la visita *	<input type="text"/>
Número de alumnos (Máximo 40 plazas, Retroinformática 20) *	<input type="text"/>
Actividades pedagógicas realizadas antes de la visita	<input type="checkbox"/> Videos educativos <input type="checkbox"/> Películas <input type="checkbox"/> Cuestiones a resolver <input type="checkbox"/> Diagrama cronológico

Imagen 11: Formulario actual para la creación de una reserva

Tras el envío del formulario, este se recibe en el correo del museo de informática para ser revisado por los gestores del mismo.

Otro problema relacionado con los formularios es que el responsable debe rellenar varios al iniciarse la actividad. Al ser cumplimentados en papel, y no disponer de una base de datos del museo, el almacenamiento de esta información resulta realmente difícil para su gestión.

3.1. Especificación de requisitos

En este apartado se nombrarán los diferentes requisitos funcionales y no funcionales que se buscan conseguir con la realización del proyecto.

3.1.1. Requisitos funcionales

Identificación del requerimiento:	RFO1
Nombre del requerimiento:	Inicio de sesión de administrador
Características	Los administradores del museo de informática deben poder iniciar sesión en la web.
Descripción del requerimiento	El sistema pedirá nombre de usuario y una contraseña para así poder ver las tablas del museo.
Prioridad del requerimiento:	Muy Alta

Identificación del requerimiento:	RFO2
Nombre del requerimiento:	Enviar formularios
Características	Los usuarios deben poder enviar formularios para reservar actividades del museo de informática
Descripción del requerimiento	El sistema contará con un apartado “reservar”, en el cual los usuarios podrán rellenar información relevante, para luego enviarla como un formulario clicando el botón “enviar” al final de este.
Prioridad del requerimiento:	Media

Identificación del requerimiento:	RF03
Nombre del requerimiento:	Mostrar tablas
Características	El sistema ofrecerá al administrador todas las tablas referentes al sistema de información del museo de informática.
Descripción del requerimiento	Se mostrará la información relativa al personal, los centros, grupos, horarios, equipamiento, material, reservas de actividad, entrada sostenible y actividades del museo de informática
Prioridad del requerimiento:	Muy alta

Identificación del requerimiento:	RF04
Nombre del requerimiento:	Modificar tablas
Características	El sistema permitirá modificar las tablas del sistema de información del museo en tiempo real.
Descripción del requerimiento	Se podrán modificar los datos relativos a las tablas de la base de datos en el modo administrador.
Prioridad del requerimiento:	Media-Alta

Identificación del requerimiento:	RF05
Nombre del requerimiento:	Exportación y búsqueda
Características	El sistema permitirá exportar la información de la tabla escogida a diferentes formatos. Además, se podrán buscar datos de las tablas mediante un filtrado de estos.
Descripción del requerimiento	Habrán iconos que permitirán exportar la información de las tablas. También habrá iconos de búsqueda avanzada y rápida para filtrar los datos.
Prioridad del requerimiento:	Alta

3.1.2. Requisitos no funcionales

Identificación del requerimiento:	RNFO1
Nombre del requerimiento:	Seguridad de la información
Características	El sistema ofrecerá privacidad a las tablas del modo administrador, para que no puedan ser accedidas desde otro modo que no sea iniciando sesión.
Descripción del requerimiento	No se podrán ver las páginas relativas a las tablas desde modo usuario en la web.
Prioridad del requerimiento:	Muy alta

Identificación del requerimiento:	RFNFO2
Nombre del requerimiento:	Interfaz clara del sistema
Características	La interfaz del sistema será fácil de utilizar y de memorizar para los administradores y usuarios.
Descripción del requerimiento	La información y las opciones de la web serán muy gráficas y sencillas.
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RNFO3
Nombre del requerimiento:	Compatibilidad con dispositivos
Características	La web podrá ser visualizada correctamente desde ordenadores, móviles y tablets.
Descripción del requerimiento	El sistema será visible desde cualquier dispositivo electrónico
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNFO4
Nombre del requerimiento:	Mantenimiento
Características	La web podrá ser actualizada y su mantenimiento será regular.
Descripción del requerimiento	El sistema podrá actualizarse desde la opción de “actualizar plugin”.
Prioridad del requerimiento: Alta	

3.2. Modelo de análisis propuesta

Para solucionar el problema y mejorar el proceso de visualización de las reservas y el almacenamiento de los datos importantes en estas, se creará un sistema de información que permita almacenar los datos relevantes de las actividades realizadas. Para permitir la visualización adecuada de las reservas, se creará un sistema de información enlazado a una web, que hará posible la visualización de las reservas de una mejor forma a la del calendario de Google vinculado al mail del museo.

Para la realización de este sistema de información será necesario realizar un diseño UML para analizar las clases y atributos que participaran en la base de datos. Será muy importante la realización de este UML, ya que, a partir de él, se podrá diseñar el resto del sistema de información mucho más fácilmente.

Draw.io es un software usado para la creación de diagramas, y debido a su clara interfaz y el fácil funcionamiento de este, ha sido elegido para realizar el UML.

A continuación, se insertará una imagen del modelo en la siguiente página.

3.3. Diseño de la solución

La arquitectura del diseño de la solución vendrá dividida en dos grandes bloques.

El primero será el de la creación del sistema de información. Para ello, se realizará un diseño lógico en el cual se verán de forma clara las relaciones entre las clases del sistema y su estructura de forma detallada (UPV). En este esquema se mostrarán las claves ajenas que relacionan las diferentes tablas y se diseñará el “esqueleto” del sistema de información.

Se ha definido el borrado en cascada para toda la base de datos, en lugar de uno restrictivo.

3.3.1. Diseño lógico

Persona (dni: char(10), nombre_propio: char(25), apellidos: char(50), telefono: int, correo: char(50), localidad: char(25), codigo_postal: int)

CP: {dni}

VNN: {nombre_propio, apellidos, teléfono, correo}

Responsable (dni: char(10))

CP: {dni}

CAj: {dni} -> Persona

BORRADO CASCADA

Monitor (dni: char(10))

CP: {dni}

CAj: {dni} -> Persona

BORRADO CASCADA

Restricciones de integridad:

RIParcial: no todos los valores que aparezcan en el atributo dni de Persona deben aparecer en el atributo dni de Responsable o Monitor

RISolapada: puede haber un mismo valor en el atributo dni de Responsable y en el atributo dni de Monitor.

Tiene_conocimiento (dni: char(10), codigo_conocimiento: char(20))

CP: {dni, codigo_conocimiento}

CAj: {dni} -> Monitor {dni}

BORRADO CASCADA

CAj: {codigo_conocimiento} -> Conocimiento {codigo_conocimiento}

BORRADO CASCADA

RI2: Todo valor que aparezca en el atributo dni de Monitor debe aparecer en el atributo dni de Tiene.

Conocimiento (codigo_conocimiento: char(20), competencia: char(25), descripción: char(100))

CP: {codigo_conocimiento}
VNN: {competencia}

Necesita (codigo_actividad: char(50), codigo_conocimiento: char(20))

CP: {codigo_actividad, codigo_conocimiento}
CAj: {codigo_actividad} -> Actividad {codigo_actividad}
BORRADO CASCADA
CAj: {codigo_conocimiento} -> Conocimiento {codigo_conocimiento}
BORRADO CASCADA
RI3: Todo valor que aparezca en el atributo codigo_actividad de Actividad debe aparecer en el atributo codigo_actividad de Necesita.

Dispone de (dni: char(10), codigo_horario: char(10))

CP: {dni, codigo_horario}
CAj: {dni} -> Monitor {dni}
BORRADO CASCADA
CAj: {codigo_horario} -> Horario {codigo_horario}
BORRADO CASCADA
RI1: Todo valor que aparezca en el atributo dni de Monitor debe aparecer en el atributo dni de Dispone de.

Horario (codigo_horario: char(10), ocupado: char(2))

CP: {codigo_horario}
VNN: {ocupado}
CAj: {codigo_horario} -> Calendario {codigo_horario}
BORRADO CASCADA

Calendario (codigo_horario: char(10), fecha_inicio: int, fecha_fin: int, hora_inicio: int, hora_fin: int, dia_semana: char(10))

CP: {codigo_horario}
VNN: {fecha_inicio, fecha_fin, hora_inicio, hora_fin, ocupado, dia_semana}

Disponibilidad (codigo_horario: char(10), codigo_actividad: char(50))

CP: {codigo_horario, codigo_actividad}
CAj: {codigo_horario} -> Calendario {codigo_horario}
BORRADO CASCADA
CAj: {codigo_actividad} -> Actividad {codigo_actividad}
BORRADO CASCADA
RI4: Todo valor que aparezca en el atributo codigo_actividad de Actividad debe aparecer en el atributo codigo_actividad de Disponibilidad.

Tiene_un_espacio (codigo_clase: char(25), codigo_horario: char(10))

CP: {codigo_clase, codigo_horario}

CAj: {codigo_espacio} -> Espacio {codigo_espacio}
BORRADO CASCADA

CAj: {codigo_horario} -> Horario {codigo_horario}
BORRADO CASCADA

RI2: Todo valor que aparezca en el atributo codigo_clase de Espacio debe aparecer en el atributo codigo_clase de Tiene un.

Espacio (codigo_espacio: char(25), edificio: char(10), nombre: char(50), capacidad: int)

CP: {codigo_espacio}

VNN: {edificio, nombre, capacidad}

Dirige (dni: char(10), codigo_actividad: char(50))

CP: {dni, codigo_actividad}

CAj: {dni} -> Monitor{dni}
BORRADO CASCADA

CAj: {codigo_reserva} -> Reserva actividad {codigo_reserva}
BORRADO CASCADA

Reserva actividad (codigo_reserva: char(50), codigo_actividad: char(50), fecha: date, hora_inicio: date, duraci3n: int, num_participantes: int, codigo_clase: char(25), codigo_grupo: char(25))

CP: {codigo_reserva}

VNN: {fecha, hora_inicio, duraci3n, num_participantes, codigo_espacio, codigo_actividad, codigo_grupo}

CAj: {codigo_espacio} -> Espacio{codigo_espacio}
BORRADO CASCADA

CAj: {codigo_actividad} -> Actividad{codigo_actividad}
BORRADO CASCADA

CAj: {codigo_grupo} -> Grupo{codigo_grupo}
BORRADO CASCADA

RI3: Todo valor que aparezca en el atributo codigo_grupo de Grupo debe aparecer en el atributo codigo_grupo de Reserva actividad.

Actividad (codigo_actividad: char(50), tipo: char(50), requisitos: char(50))

CP: {codigo_actividad}

VNN: {tipo, requisitos}

Grupo (codigo_grupo: char(25), /total, chicos alumnos: int, chicos profesores: int, chicas alumnas: int, chicas profesoras: int, asignatura: char(50), observaciones: char(500), dni: char(10), codigo_centro: char(15))

CP: {codigo_grupo}

VNN: {/total, chicos alumnos, chicos profesores, chicas profesoras, asignatura, dni, codigo_centro}

CAj: {dni} -> Responsable{dni}

BORRADO CASCADA

RI3: Todo valor que aparezca en el atributo dni de responsable debe aparecer en el atributo dni de grupo.

CAj: {codigo_centro} -> Centro{codigo_centro}

BORRADO CASCADA

RI5: Todo valor que aparezca en el atributo codigo_centro de centro debe aparecer en el atributo codigo_centro de grupo.

Centro (codigo_centro: char(15), nombre: char(25), localidad: char(25), cp: int, provincia: char(25), país: char(15))

CP: {codigo_centro}

VNN: {nombre, localidad, cp, provincia}

Curso (codigo_grupo: char(25), curso: char(25))

CP: {codigo_grupo, curso}

CAj: {codigo_grupo} -> Grupo

BORRADO CASCADA

Dispone _equipa (codigo_clase: char(25), codigo Equipamiento: char(50), cantidad: int)

CP: {codigo_clase, codigo Equipamiento}

VNN: {cantidad}

CAj: {codigo_clase} -> Espacio{codigo_clase}

BORRADO CASCADA

CAj: {codigo Equipamiento} -> Equipamiento{codigo Equipamiento}

BORRADO CASCADA

Material (codigo_material: char(50), nombre: char(25), stock: int)

CP: {codigo_material}

VNN: {nombre, stock}

Equipamiento (codigo Equipamiento: char(50), nombre: char(50))

CP: {codigo Equipamiento}

VNN: {nombre}

Requiere equipa (codigo_actividad: char(50), codigo Equipamiento: char(50), cantidad: int)

CP: {codigo_actividad, codigo Equipamiento}

VNN: {cantidad}

CAj: {codigo_actividad} -> Actividad{codigo_actividad}

BORRADO CASCADA

CAj: {codigo Equipamiento} -> Equipamiento{codigo Equipamiento}

BORRADO CASCADA

Cuenta con (codigo_actividad: char(50), codigo Material: char(50), cantidad: int)

CP: {codigo_actividad, codigo Material}

VNN: {cantidad}

CAj: {codigo_actividad} -> Actividad{codigo_actividad}

BORRADO CASCADA

CAj: {codigo Material} -> Material{codigo Material}

BORRADO CASCADA

RI4: Todo valor que aparezca en el atributo codigo Material de Material debe aparecer en el atributo codigo Material de Cuenta con.

Entrada sostenible (codigo_reserva: char(50), \total)

CP: {codigo_reserva}

VNN: {\total}

CAj: {codigo_reserva} -> Reserva actividad{codigo_reserva}

BORRADO CASCADA

Contiene pieza (codigo_pieza: char(10), codigo_reserva: char(50))

CP: {codigo_pieza, codigo_reserva}

VNN: {cantidad}

CAj: {codigo_pieza} -> Pieza{codigo_pieza}

BORRADO CASCADA

CAj: {codigo_reserva} -> Entrada sostenible{codigo_reserva}

BORRADO CASCADA

Pieza (codigo_pieza: char(10), descripcion: char(25))

CP: {codigo_pieza}

VNN: {descripcion}

3.3.2. Diseño físico

En cuanto al diseño físico, las tablas serán creadas como ficheros desordenados debido a la alta cardinalidad de estas. Además, al ser tablas muy dinámicas y volátiles, las actualizaciones habrían sido demasiado caras en caso de tener un fichero ordenado o de otro tipo que no fuera el desordenado (Anónimo).

En cuanto a los índices, MySQL crea por defecto a las claves primarias un índice "PRIMARY" que es del tipo Árbol B+. Estos índices permiten una búsqueda óptima y efectiva por medio de la clave primaria de las tablas.

Las tablas Grupo y Reserva Actividad serán las que más consultas tendrán, por tanto, con estos índices sobre la clave primaria los resultados obtenidos en consultas sobre estas tablas serán proporcionados con velocidad y un coste bajo.

3.3.3. Diseño web

Finalmente, antes de proceder con la creación del sistema en cuestión, se realizará el diseño físico del mismo. El diseño físico permitirá evitar repeticiones innecesarias y mejorar el rendimiento del sistema de forma notable. Se identificarán los atributos y las tablas más accedidas a la hora de realizar consultas sobre la base de datos, y se definirán los índices y las tablas de la forma adecuada para acelerar el proceso de obtención de los mismos.

El segundo bloque será la creación de la web. Para ello se hará uso de Wordpress. Con este sistema la creación de la aplicación será sencilla ya que la interfaz es muy intuitiva. Se elegirá la plantilla que mejor se adapte al diseño de la web y se creará un menú en el cual se puedan acceder a los apartados más importantes.

El tema elegido será OceanWP. Además de ser uno de los mejor puntuados de Wordpress, es el que mejor se adapta a la aplicación, ya que dispone tanto de una gran versatilidad en cuanto a multitud de estilos de web, como de una compatibilidad con muchos plugins y complementos. Una característica importante del tema elegido es la posibilidad de utilizar Elementor para la edición de las páginas (Bhaskar, 2023).

Elementor es un plugin que permite una personalización total del diseño de las páginas de una web. Cuenta con infinidad de widgets y plantillas.

La conexión de la base de datos a la web se realizará con el plugin WP Data Access, el cual permitirá gestionar y visualizar las tablas sin problema.

Por tanto, con todas las herramientas mencionadas se creará la aplicación web a la que se enlazaré la base de datos.

3.4. Desarrollo de la solución

Para la creación del sistema de información, como se había definido anteriormente, serán necesarios una serie de pasos.

3.4.1. Creación de las tablas del sistema de información

Tras tener completo el diseño lógico y físico, se procedió a crear la base de datos en MySQL Workbench.

El host escogido fue 'localhost' con el puerto por defecto 3306.

Para la creación de las tablas se utilizó el lenguaje SQL. Para introducir los atributos y las características de cada uno de ellos en cada tabla se escribieron las sentencias SQL de cada una en el apartado de texto de MySQL.

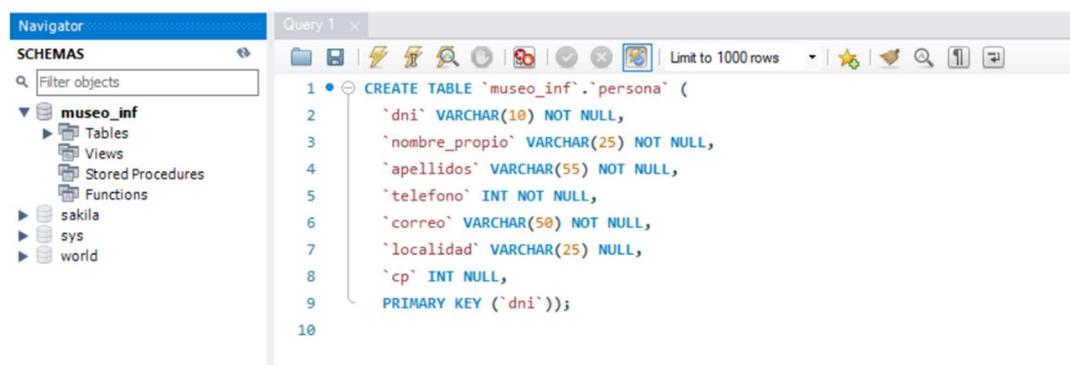


Imagen 13: En la imagen podemos observar la interfaz de MySQL para la creación de tablas. En la pestaña 'Query 1' se escribe el texto SQL para la creación de tabla y posteriormente, tras hacer clic en el símbolo del rayo con un cursor encima, se ejecuta este código.

El código necesario para la creación de todas las tablas es extenso. Para añadir las tablas sin tener problemas con las claves ajenas, se añadieron primero las que no tenían clave ajena asociada en la tabla. Posteriormente tras añadir todas las tablas sin clave ajena, se fueron añadiendo las restantes que dependían de las ya creadas. Otra opción habría sido haber utilizar la opción "Set Constraint Deferred" para garantizar la integridad referencial y poder añadir las tablas sin preocuparse por estas claves.

En este caso se prefirió añadirlas con cuidado sin utilizar la opción comentada anteriormente (Sqllearning, s.f.).

Un ejemplo de inserción SQL de tabla es:

Tabla 'reserva actividad':

```
CREATE TABLE `museo_inf`.`reserva_actividad` (  
  `codigo_reserva` VARCHAR(50) NOT NULL,
```

```
`codigo_actividad` VARCHAR(50) NOT NULL,  
`fecha` DATE NOT NULL,  
`hora_inicio` TIME NOT NULL,  
`duracion` INT NOT NULL,  
`num_participantes` INT NOT NULL,  
`codigo_espacio` VARCHAR(25) NOT NULL,  
`codigo_grupo` VARCHAR(25) NOT NULL,  
PRIMARY KEY (`codigo_reserva`),  
INDEX `reserva_actividad_espacio_idx` (`codigo_espacio` ASC) VISIBLE,  
INDEX `reserva_actividad_actividad_idx` (`codigo_actividad` ASC) VISIBLE,  
INDEX `reserva_actividad_grupo_idx` (`codigo_grupo` ASC) VISIBLE,  
CONSTRAINT `reserva_actividad_espacio`  
  FOREIGN KEY (`codigo_espacio`)  
  REFERENCES `museo_inf`.`espacio` (`codigo_espacio`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
CONSTRAINT `reserva_actividad_actividad`  
  FOREIGN KEY (`codigo_actividad`)  
  REFERENCES `museo_inf`.`actividad` (`codigo_actividad`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
CONSTRAINT `reserva_actividad_grupo`  
  FOREIGN KEY (`codigo_grupo`)  
  REFERENCES `museo_inf`.`grupo` (`codigo_grupo`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE);
```

En el Anexo 2 se añadirá el código de inserción de varias tablas.

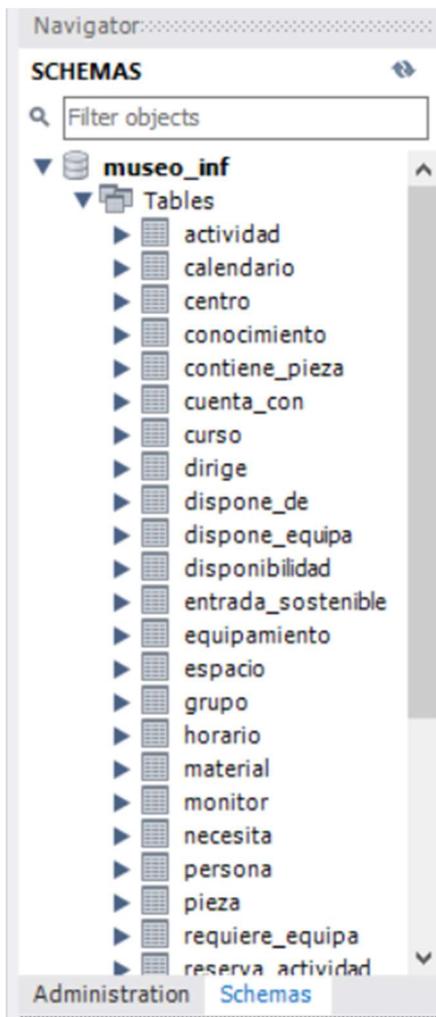


Imagen 14: En la imagen podemos observar como las tablas han quedado añadidas correctamente en el sistema MySQL. En el apartado 'Tables' se pueden observar añadidas cada una de las tablas, a falta de las que no se pueden ver dado que no hay una visualización completa de todas ellas.

3.4.2. Inserción de datos

Tras tener la estructura general de la base de datos terminada, fue necesario introducir datos en la misma. Para la introducción de valores se utilizó el lenguaje SQL. Se introdujeron datos suficientes para poder observar cualquier tipo de casuística en cuanto a las diferentes tablas de la base de datos.

En algún caso fue necesario utilizar la sentencia SQL "SET CONSTRAINTS DEFERRED", ya que para introducir ciertos datos que dependían de varias tablas se complicaba sin el uso del modo diferido de inserción.

En algunos casos, en vez de insertar con secuencias SQL, fue más sencillo y rápido realizar la inserción manualmente con la opción rápida de MySQL.

En anexos se mostrarán varias instrucciones de inserción de datos SQL sobre varias tablas.

En primer lugar, se creó la tabla persona con los datos de los monitores y responsables del museo de informática. Un ejemplo de persona es:

```
INSERT INTO `museo_inf`.`persona` (`dni`, `nombre_propio`, `apellidos`, `telefono`, `correo`, `localidad`, `cp`) VALUES ('23654268S', 'Carlos', 'Martínez García', '685201576', 'correo1@gmail.com', 'Valencia', '46003');
```

(Los datos son un ejemplo de cómo sería la base de datos llevada a la realidad. No son los nombres reales de los monitores y responsables de grupo del museo de informática.)

Posteriormente, se introdujeron los datos asociados a las tablas “responsable” y “monitor” por la herencia con la clase “persona”.

En las tablas “horario” y “calendario” se tuvieron que insertar muchísimas entradas, debido a que tenían que estar anotados todos los horarios de los monitores del museo, de las clases en las cuales se podrían realizar las actividades, y las fechas en los cuales estarían disponibles cada una de las actividades.

En el apartado “ocupado” de horario se indicaría si el horario escogido para el espacio, actividad o monitor es el adecuado o si por el contrario no hubiera disponibilidad y no pudiera realizar la reserva.

Dada la importancia de esta tabla se instaron el número de datos necesario para crear unos horarios realistas que permitiesen hacer pruebas acertadas sobre la base de datos.

	codigo_horario	fecha_inicio	fecha_fin	hora_inicio	hora_fin	dias_semana
▶	HM0012	2022-09-01	2023-09-01	13:00:00	15:00:00	Miércoles
	HM0011	2022-09-01	2023-09-01	11:00:00	13:00:00	Lunes
	HM0010	2022-09-01	2023-09-01	10:00:00	12:00:00	Jueves
	HM0009	2022-09-01	2023-09-01	08:30:00	10:00:00	Martes
	HM0008	2022-09-01	2023-09-01	12:30:00	14:00:00	Miercoles
	HM0007	2022-09-01	2023-09-01	09:30:00	11:00:00	Viernes
	HM0006	2022-09-01	2023-09-01	08:00:00	10:00:00	Lunes
	HM0005	2022-09-01	2023-09-01	10:00:00	12:00:00	Jueves
	HE0004	2022-09-01	2023-09-01	09:00:00	11:00:00	Martes
	HE0003	2022-09-01	2023-09-01	08:00:00	10:00:00	Miercoles
	HE0002	2022-08-01	2023-11-25	10:00:00	12:00:00	Martes
	HE0001	2022-05-12	2024-05-12	12:00:00	14:00:00	Lunes
	HA0021	2022-09-01	2023-09-01	08:00:00	15:00:00	Lunes
	HA0020	2022-09-01	2023-09-01	08:00:00	15:00:00	Lunes
	HA0019	2022-09-01	2023-09-01	08:00:00	15:00:00	Lunes
	HA0018	2022-09-01	2023-09-01	08:00:00	15:00:00	Lunes
	HA0017	2022-09-01	2023-09-01	08:00:00	15:00:00	Lunes
	HA0016	2022-09-01	2023-09-01	08:00:00	15:00:00	Lunes
	HA0015	2022-09-01	2023-09-01	08:00:00	15:00:00	Lunes

Imagen 15: En la imagen se pueden observar los distintos horarios de la clase “calendario” con sus respectivos días de la semana asociados a estos. En la tabla “horario” los códigos de horario dispondrán de un atributo “ocupado” que determinara su disponibilidad o no.

En el resto de las tablas, se insertaron los datos sin ningún problema. Las actividades reales del museo fueron introducidas en su totalidad y se han creado dos reservas a modo de ejemplo.

Para cada actividad se añadieron los conocimientos necesarios para ser impartidas por un monitor. Por tanto, se asociaron datos de conocimientos a cada monitor. Un ejemplo de conocimiento insertado es:

```
INSERT INTO `museo_inf`.`conocimiento` (`codigo_conocimiento`, `competencia`)
VALUES ('CM001', 'Programación en Scratch');
```

Al crearse estas reservas se han añadido centros, grupos y responsables asociados.

Una de las reservas creadas es:

```
INSERT INTO `museo_inf`.`reserva_actividad` (`codigo_reserva`,
`codigo_actividad`, `fecha`, `hora_inicio`, `duracion`, `num_participantes`,
`codigo_espacio`, `codigo_grupo`) VALUES ('R0001', 'A07', '2023-05-25', '12:00:00',
'120', '24', 'E0005', 'G0001');
```

En cuanto al material disponible, fueron introducidos todos los materiales necesarios para cada una de las actividades, y un número concreto de cuánto se dispone de cada uno de ellos. La cantidad de cada uno de estos materiales queda reflejada para que los administradores de la base de datos puedan actualizarlos y hacer cálculos aproximados de cuanto se necesitaría dependiendo del stock en el momento de la visualización del material.

El equipamiento mostrará datos sobre cuánto equipamiento dispone cada espacio y de cuánto es necesario para cada actividad. En el caso de actividades como la de “retro informática” el número máximo de alumnos es 20, y al estar previsto que los alumnos se colocarían por parejas en cada ordenador, los requisitos disminuyen en cuanto a cantidad. No obstante, hay que tener en cuenta el tipo de ordenador necesario para cada una y muchos otros aspectos que han sido introducidos en las tablas para ajustarse a la realidad de forma total,

Respecto a las entradas sostenibles y las piezas entregadas, se han introducido datos de las piezas más comúnmente entregadas en las actividades realizadas por el museo. Tales como monitores, ratones, teclados, disquetes, productos electrónicos y placas base. Al haber creado dos reservas, se han asociado dos entradas sostenibles a ellas, a pesar de que no hay obligación de requerir de una entrada asociada a una reserva, ya que como se indica en el diseño lógico, la cardinalidad de la relación no obliga a que cada grupo que acude a una reserva entregue una entrada sostenible.

Con esto queda terminada la inserción de datos.

Todas las tablas disponen de contenido y es posible realizar consultas sobre cualquiera de ellas para obtener información relevante al respecto.

Tras haber finalizado totalmente la base de datos, el siguiente paso fue crear la web. Esta web tras ser finalizada se conectaría con la base de datos para que, en modo administrador, se pudieran ver los detalles de todas las tablas del museo de informática.

3.4.3. Creación de la web

Para la creación de la web se utilizó Wordpress. Debido al nivel de personalización que permite, la sencillez, flexibilidad y la interfaz intuitiva que tiene, ha sido la opción escogida para esta tarea.

Se encontraron ciertos problemas en cuanto a la elección del dominio de la web, ya que los dominios gratuitos no eran compatibles con el programa que se utilizó en un primer momento para crear la base de datos (PostgreSQL).

Debido a los problemas de compatibilidad y a los dominios compatibles con el sistema usado para crear la base de datos en un primer momento, se decidió usar un dominio local para la web y evitar, de este modo, cualquier inconveniente. Además de ser gratis, la gestión de la página en local permite mucha más comodidad a la hora de trabajar. La mayoría de los dominios que fueron encontrados, tenían muchas limitaciones en sus versiones gratuitas, y prácticamente todos pedían un mínimo de 40 euros por alojar una web, por tanto, dado que la intención fue realizar este TFG de manera gratuita, se eligió el dominio local para crear la web.

El programa utilizado para permitir la creación de la web en local se llama 'Local'. Con este programa se puede crear una web mediante Wordpress usando el ordenador personal como dominio de esta.

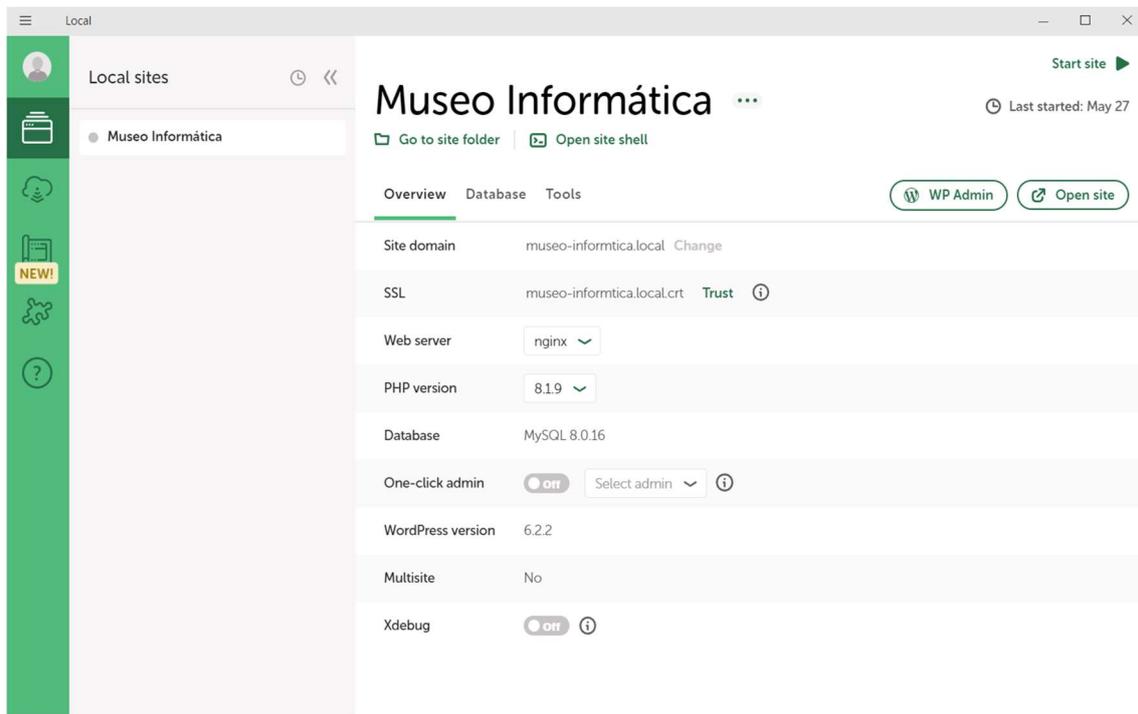


Imagen 16: En la imagen podemos observar la interfaz del programa. Tras crear una web, haciendo click en WP Admin, 'Local' nos dirige directamente al administrador de Wordpress de la página. Iniciando sesión ya es posible editar y personalizar la web. Con la opción "Open Site" el programa nos dirige a la visión web en modo usuario. En las opciones Database y Tools se puede acceder al SGBD de la web y a configuraciones adicionales del programa.

Una vez definido el dominio de la página como local, el siguiente paso a realizar fue la creación de la web como tal.

Para ello el plugin “Elementor” fue muy importante, ya que tiene muchas características favorables para la edición (Profesionalpress, 2023). Entre ellas destacan:

- Una amplia selección de widgets y plantillas
- Una personalización muy avanzada.
- Adaptabilidad a cualquier estilo de web.
- Plugins adicionales compatibles.
- Una interfaz muy intuitiva.

Las páginas de Wordpress pueden ser editadas con el editor por defecto o con Elementor. Al principio se usó el editor predeterminado y lo encontré demasiado básico. No permitía personalizar mucho el diseño, las estructuras creadas eran muy sencillas, y el resultado de horas de edición no era el deseado.

Por tanto, se cambió al editor de Elementor y se vio realmente mejorada la personalización general de la web.

El primer paso para empezar a diseñar la web fue encontrar el tema que mejor se ajustara al diseño deseado. Se encontraron varios temas interesantes que se adaptaron a las necesidades, pero a pesar de ello tenían un diseño muy especializado para ciertos tipos de web. Por ejemplo, uno que llamó la atención, fue Divi.

Divi es un tema muy flexible y que permite un diseño avanzado, pero cuenta con muchas plantillas dirigidas al comercio online. Al estar tan orientado a funcionalidades no deseadas en la web, se descartó del proyecto.

Por tanto, OceanWP fue el tema elegido. Entre sus características principales destaca un diseño adaptable a cualquier tipo de dispositivo, lo cual teniendo en cuenta que la web debía tener una gran accesibilidad, resultó muy útil. También dispone de una personalización avanzada en cuanto a los colores, menús y pies de página del tema. Por supuesto una compatibilidad con Elementor, y un gran rendimiento (Hostcupon, 2022).

Tras escoger OceanWP como tema, y Local como programa de dominio el siguiente paso fue crear la página y el menú principales de la web. En cuanto a la página principal, se trató de que fuera lo más sencilla, y al mismo tiempo gráfica posible. El objetivo de la web es poder visualizar tablas en modo administrador y como función secundaria, y no por tanto menos importante, que los usuarios puedan introducir reservas.

Se decidió crear una imagen principal de una parte del museo de informática como portada de la web. Para añadir dinamismo, se generó la opción de desplazamiento con “scroll” de la imagen para que no se tratase de una figura estática y simple. Desde la imagen se colocó un hipervínculo que llevase directamente al modo administrador para iniciar sesión.

En cuanto a la cabecera y el pie de página, se añadió un widget de redes sociales con vínculos directos a las redes del museo de informática, e información básica de contacto (teléfono, email y ubicación).

Posteriormente fue necesario crear un menú para acceder a las principales páginas de la web. En el menú se incluyeron cuatro entradas:

1. Inicio
2. Reservar
3. Administrador
4. Contacto

Con esto, las necesidades de la web quedaban totalmente cubiertas. Para continuar con la creación se procedió a crear la página de administrador.

Para crear el inicio de sesión tuvo que utilizarse un plugin totalmente gratuito llamado “Pro elements” (Salas, 2023). Este plugin permitió añadir opciones adicionales al editor por defecto de Elementor sin necesidad de pagar la versión premium. Con esto se pudo añadir un widget de inicio de sesión en el cual solo pudiesen iniciar la misma los perfiles de usuario de Wordpress asignados como administrador.

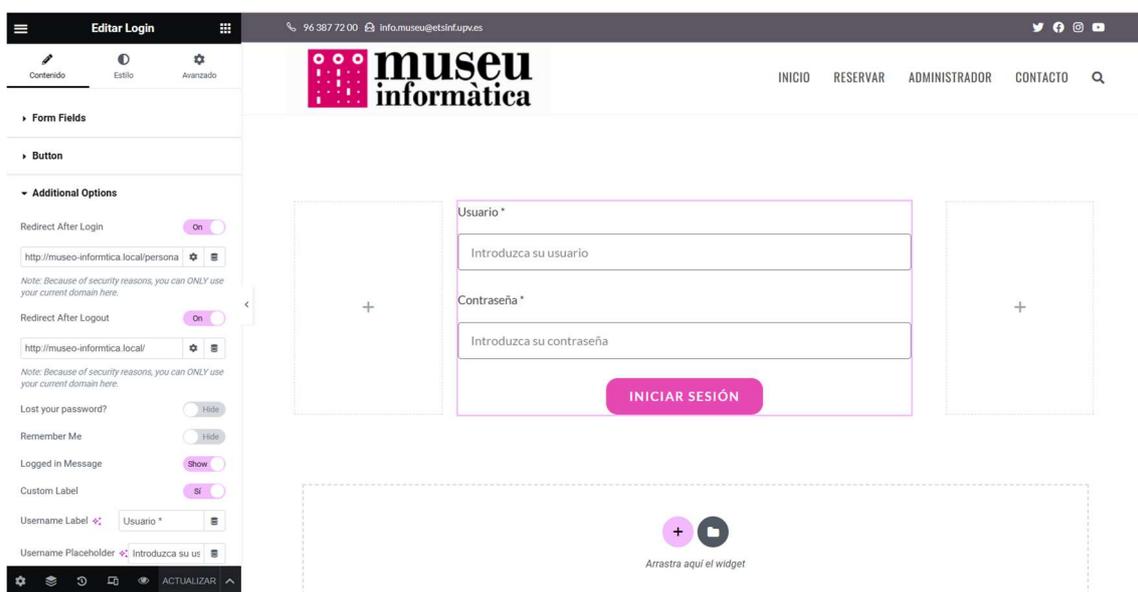


Imagen 17: En la imagen se puede observar el diseño del inicio de sesión de administrador, el menú y la cabecera de la página. A la izquierda se encuentra el editor del widget de inicio de sesión de elementor. En la pestaña “Form Fields” se pueden editar los textos de las entradas de inicio de sesión, y en la pestaña “Button” las características del botón asociado. En “Additional Options” se pueden configurar las acciones avanzadas relacionadas con las redirecciones y las contraseñas.

Se decidió no añadir un recordatorio de contraseña ya que al ser usuarios de Wordpress y teniendo en cuenta que la web solo dispone de inicio de sesión como administrador, las contraseñas serían sabidas por ellos y fácilmente visibles en el apartado “Usuarios” de Wordpress. En caso de haber también inicio de sesión de usuario, la opción recordar contraseña habría sido totalmente necesaria.

En cuanto a las redirecciones, tras cerrar sesión se hizo que la web dirigiera al administrador automáticamente a la página de inicio, y al iniciar sesión a las tablas de la web. En cuanto a estas tablas, posteriormente se explicará su creación y diseño.

Tras finalizar la página de inicio y la de administrador se hizo la de reservar. Para ello se tuvieron que instalar los plugin WPForms y WPMail SMTP. WPForms es un plugin que permite crear formularios de manera personalizada. Su versión gratuita es muy limitada,

pero permite realizar las configuraciones necesarias para el objetivo de la web (Cardozo, 2021).

En primer lugar, se creó un formulario llamado “Reservas museo”. Para elegir los campos a incluir, se replicó parte de la información que se pedía en el formulario original en el sistema actual del museo de informática. Se añadieron algunas entradas adicionales con información relevante y una pequeña descripción a modo de cabecera superior al formulario.

En el formulario se incluyen los siguientes campos:

- Nombre del centro
- Nombre de la persona responsable del grupo
- Móvil de contacto
- Email de contacto
- Localidad y código postal
- Titulación y curso del grupo
- Actividad
- Fecha deseada
- Idioma de la visita
- Número de alumnos
- Actividades realizadas previas a la visita

Todas ellas marcadas como obligatorias a rellenar, excepto el apartado de las actividades realizadas previamente a la visita. Con esto la información que será recibida tras el envío de un formulario permitirá añadir la información correspondiente a la base de datos y comprobar si el horario seleccionado es válido o no. En caso de no serlo se contactaría con el usuario para indicarle los horarios disponibles.

Tras tener finalizada la parte gráfica del formulario, el siguiente paso fue configurar la conexión SMTP para recibir un correo tras producirse un envío de formulario. Para ello se utilizó el plugin WPMail SMTP mencionado anteriormente.

En primer lugar, desde el plugin se accedió al menú general para seleccionar el servidor de correo electrónico. Se seleccionó Google/Gmail, ya que el correo que fue elegido para enviar los mensajes fue el Gmail personal (carlosolme05@gmail.com).

En segundo lugar, se accedió a Google Cloud para añadir un nuevo proyecto de conexión. Se seleccionó la API de Gmail y la opción “User data” en el proceso de creación de proyecto. Tras añadir información general, se seleccionó la opción “web application” como tipo de aplicación en el apartado OAuth Client ID y se copió el enlace del url de la web en la opción “Authorised Java origins. Posteriormente, se añadió el enlace del servidor SMTP del plugin WPMail SMTP. Con toda esta información añadida se creó el proyecto, y accediendo a él, se obtuvieron las credenciales necesarias para configurar el servidor SMTP desde el plugin (WPForms - WordPress Forms Plugin, 2022).

Con esto la configuración de la recepción de los formularios vía email quedó finalizada, y cualquier envío de formulario sería enviado automáticamente por el correo personal

(carlosolme05@gmail.com), al correo que se seleccionó como correo por defecto del modo administrador (carlos_oge@hotmail.com).

Finalmente se creó la página de contacto. En esta página se añadió toda la información referente a las distintas formas de contactar con el museo de informática, los horarios y su ubicación en Google Maps de forma gráfica.

Para la ubicación en Google Maps se creó un widget que permitiera visualizar de forma ampliada el lugar donde se encuentra el museo. La opción de zoom del widget fue muy útil, ya que se pueden ver con detalle las carreteras y las calles que rodean la ubicación, por tanto, es más fácil orientarse.

Tras tener finalizadas las páginas básicas de la web fue necesario conectarla con la base de datos para visualizar las tablas en el modo administrador.

3.4.4. Conexión de la web con la base de datos

Para realizar la conexión se utilizó un plugin de Wordpress llamado WP Data Access (ChatGPT, 2023).

En un principio se trató de conectar la base de datos con el plugin con la opción de “Añadir base de datos externa”. Tras introducir toda la información de la base de datos en MySQL el plugin no permitió realizar la conexión. Se probaron todas las alternativas posibles para arreglar este error e incluso se contactó con el servicio técnico del plugin para solucionarlo y no fue posible. Desde el servicio técnico se indicó que podía ser un problema de compatibilidad con el programa “Local” que se utilizaba para tener localhost como el host de la web. Dado que la web se realizó usando este programa y que ya estaba casi terminada en el momento de conectarla con la base de datos, no tuvo sentido crearla de nuevo con otro programa, principalmente porque el problema seguramente se tratara de un error del mismo plugin.

Tras muchos días sin avanzar, se consiguió enlazar la web y la base de datos de una forma distinta. La solución fue utilizar la opción de exportar en MySQL Workbench. Se Exportó todo el contenido de la base de datos “museo_inf”, es decir, tanto la estructura como los datos, usando la ruta Server->Data export. Tras finalizar, se obtuvo un documento SQL con toda la información necesaria.

El programa “Local”, el cual permitía realizar la web con localhost como dominio, dispone de un SGBD. En un principio no detectó esta opción, pero tras investigar dentro de la aplicación se descubrió esta información.

Desde el gestor de bases de datos de Local, se importó correctamente el documento SQL con todo el contenido de “museo_inf”. El sistema de gestión dispone de una interfaz muy sencilla e intuitiva, por lo tanto, no fue muy complicado encontrar las opciones necesarias para proceder con la importación del archivo.

Importar

279 sentencias SQL ejecutadas correctamente. (1.455 s)

Importar archivo

SQL[.gz] (< 1GB): No se ha seleccionado ningún archivo

Parar en caso de error Mostrar solamente errores

Imagen 18: Secuencia de inserción del documento sql

Teniendo las tablas y los datos en local, desde WP Data Tables, es posible visualizar toda la información local sin necesidades de realizar ninguna conexión. Por tanto, todas las tablas de mi base de datos fueron visibles en mi plugin tras realizar la importación.

El siguiente paso fue convertir las tablas al formato del plugin. Para esto, en la opción “tables” del plugin, se fueron creando las 26 tablas a partir de la información local de ellas. Estas tablas están conectadas a la información, por tanto, si desde el SGBD de “Local” son modificadas, los cambios serían visibles en tiempo real.

En cada una de las tablas se añadió la opción de poder buscar por medio de diferentes parámetros. Dado que los datos de las tablas eran variados, y la cantidad de atributos que compone cada una de ellas, la opción de búsqueda avanzada por parámetros fue necesaria para mejorar el rendimiento y ahorrar trabajo a los administradores.

The screenshot shows the configuration interface for a table in WP Data Tables. The interface is organized into several sections:

- Search Builder:** Add search builder (dni, nombre_propio, apellidos, telefono, correo, localidad, cp)
- Search Panes:** Add search panes
- Buttons:** Add buttons
- Add Buttons:** Excel CSV PDF Print Copy SQL Column list Column button - Button Caption:** Labels only Icons only Labels and icons
- Table Configuration:** Arrange

export buttons		
search builder		
page length		search box
processing...		
table		
summary		pagination

 - page length
 - search box
 - processing...
 - table
 - summary
 - pagination

Imagen 19: En esta imagen se pueden ver las opciones seleccionadas en cada una de las tablas. Los parámetros de búsqueda relativos a los atributos fueron configurados, y además se pueden exportar los datos como Excel, CSV y PDF.

Tras crear las tablas, fue necesario crear las páginas asociadas para insertarlas y visualizar su información en modo administrador. Para esto, se crearon las páginas como privadas, para que solo pudieran ser accesibles en el modo administrador y así garantizar

su seguridad. La visión de página que debe poder ver un usuario que no es administrador, es la web en su totalidad a excepción de las páginas que contienen las tablas del modo administrador. Estas tablas solo deben ser visibles al iniciar sesión correctamente, y no antes de ello.

Se creó un menú adicional que pudiese ser añadido mediante widgets de elementor. Este menú agrupa todas las páginas que contienen tablas relevantes.

El citado menú contiene los siguientes datos:

1. Personal: Con la información referente a los monitores y los responsables del museo de informática
2. Conocimientos: Con los conocimientos de cada monitor. En esta página son visibles los conocimientos asociados a cada actividad y los que presenta cada monitor, para así determinar que actividades puede realizar cada uno.
3. Grupos y centros: Con la información de todos los grupos y centros que han participado en una visita guiada en el museo.
4. Reservas: Con la información histórica de todas las reservas que han sido realizadas con éxito en el museo.
5. Equipamiento: En esta entrada encontramos dos páginas. Una relativa al equipamiento disponible en cada espacio, el cual muestra los recursos de mobiliario presentes en cada clase. Otra relativa al equipamiento necesario por cada actividad, el cual debe ser comparado al de los espacios para determinar cuál de ellos es óptimo para la realización de dichas actividades.
6. Material: En esta entrada encontramos la página “material de actividad”. La información de ella muestra el material necesario para realizar cada actividad.
7. Horarios: La entrada horarios cuenta con tres páginas. Una referente a los espacios, otra referente a las actividades, y otra referente a los monitores. En cada una de ellas se pueden observar los horarios y la disponibilidad general.
8. Entrada sostenible: Con las entregas voluntarias de material asociadas a cada reserva.

Con esto, el menú dispondría de todas estas entradas, las cuales serían visibles desde cualquiera de ellas. De esta forma el modo administrador dispondría de un gran dinamismo y un diseño ágil para poder visualizar el contenido de las tablas de forma rápida y ordenada.

Muchas de las consultas que pueden ser necesarias sobre las tablas disponen de la necesidad de visualizar varias páginas a la vez. Por tanto, la posibilidad de poder visualizarlas en ventanas adicionales mediante los accesos del menú es muy importante.

Monitores y responsables



Constructor de búsqueda

Añadir condición

Mostrar

10

registros

Buscar:

DNI	NOMBRE PROPIO	APELLIDOS	TELEFONO	CORREO	LOCALIDAD	CP
[redacted]	Carlos	Martínez García	[redacted]	correo1@gmail.com	Valencia	46003
[redacted]	Clara	Pérez Muñoz	[redacted]	correo2@gmail.com	Valencia	46001
[redacted]	Gerard	Marqués Pérez	[redacted]	correo5@gmail.com	Valencia	46008
[redacted]	Lucas	López Álvarez	[redacted]	correo3@gmail.com	Valencia	46002
[redacted]	Nuria	Torres Silva	[redacted]	correo5@gmail.com	Valencia	46003
[redacted]	Pepe	García Iborra	[redacted]	correo6@gmail.com	Mislata	46920

Imagen 20: En la imagen se puede observar la tabla con la información de los monitores y los responsables referente a la entrada del menú "Personal". Más abajo, bajando con la rueda del ratón, se pueden observar las tablas monitores y responsables con sus respectivos DNI asociados, ya que la tabla de la imagen es la tabla "Personas", la cual incluye la información de las dos mencionadas de forma conjunta. Se pixelan los DNI y Teléfonos por si alguno coincide con uno real, por privacidad.

Tal y como se ve en la imagen, las tablas disponen de un constructor de búsqueda, el cual permite generar condiciones para filtrar datos. Esta filtración de datos permite crear las búsquedas personalizadas deseadas, de manera que se puede seleccionar la información necesaria para visualizar de un modo sencillo. Las búsquedas permitidas son por igualdad, diferencia, valores intermedios, valor nulo e inicio o fin por una sentencia determinada. Además de las opciones avanzadas de búsqueda, también hay disponible un buscador rápido, con el cual es muy fácil buscar por igualdad cualquier valor.

También se incluyen los iconos para exportar información. Esta información puede ser exportada a PDF, Excel y CSV. Estas funciones pueden ser muy útiles a la hora de necesitar imprimir cierta información.

Un problema que se tuvo con la web fue que, tras volver al menú de inicio desde el modo administrador, no hubo forma de acceder de nuevo al menú de las tablas sin cerrar sesión y volver a abrirla. Esto se debía a que la redirección a las tablas era disponible solo tras iniciar sesión. Por tanto, en el menú principal de la página se añadió un método de

búsqueda para buscar las tablas por su nombre y así poder volver al modo administrador sin ningún problema.

De esta manera, tras finalizar las funcionalidades del modo administrador, la web quedó acabada. El mayor problema que se encontró en su realización fue el modo de conectarla con la base de datos. Todos los plugins eran de pago, a precios mayores de 50 euros y las opciones que ofrecían los gratuitos eran totalmente insuficientes.

Tras mirar en muchas páginas acerca de todos los plugins gratuitos y sus características, finalmente se encontró información de WP Data Access mediante chat GPT.

El plugin es muy recomendado, ya que además de permitir muchas funcionalidades que solo son disponibles en programas de pago, tiene una interfaz medianamente sencilla de entender y muchas opciones de búsqueda en las tablas.

La opción que no permitió añadir, que limitó en cierta manera el resultado final, fue el no poderse incluir un botón para añadir filas en las tablas directamente desde la web. La única forma de añadir información en las tablas es desde el modo administrador interno de Wordpress, entrando en el plugin en la configuración de tablas. Accediendo a la ventana “Tables” y posteriormente haciendo clic en “Explore” en una de las tablas se puede acceder a un editor en tiempo real.

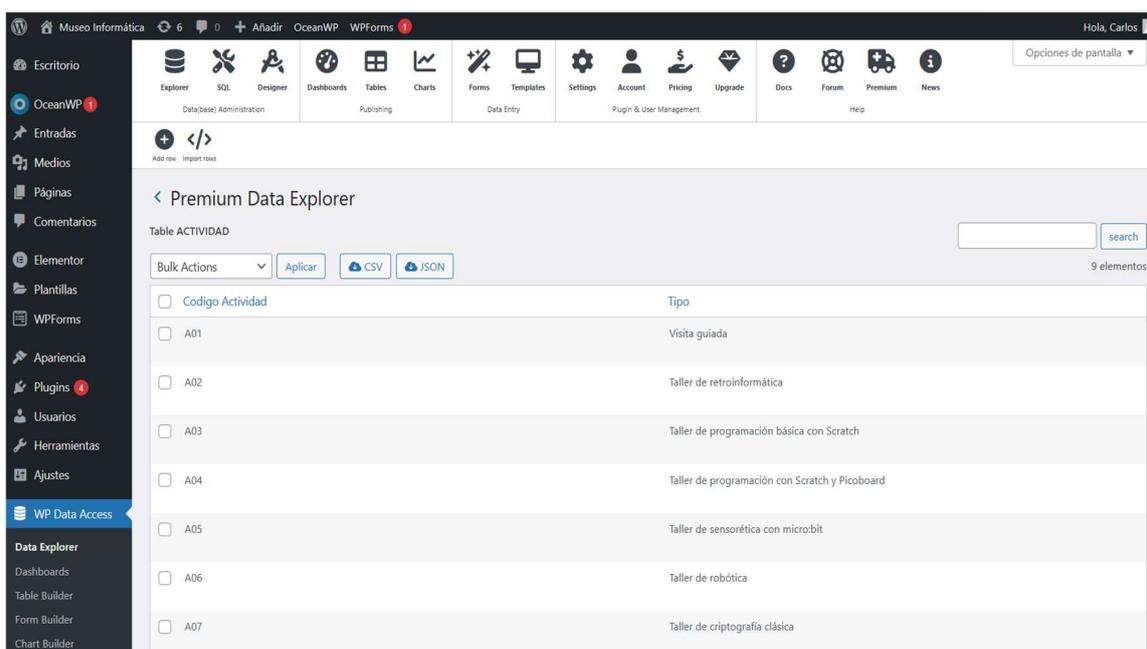


Imagen 21: En la imagen se puede observar la opción de “Add row” en la parte superior izquierda, justo encima de Premium Data Explorer. Desde esta opción se pueden añadir filas a la tabla que sea necesaria. Tras añadir las filas, estas se podrán visualizar de forma instantánea en las tablas de la web.

Por tanto, a pesar de no poder modificar los datos directamente desde la web, sí que se pueden modificar desde las opciones del plugin en el administrador de Wordpress.

Teniendo en cuenta que, en este caso, el administrador de la web soy yo, esto no supone ningún problema, ya que se puede acceder al menú de Wordpress con la misma facilidad que se realiza a las funciones de las tablas. En el caso de que fuese un administrador de

museo sin acceso a la información del modo administrador de la web, esto sí que supondría un problema, ya que no podría añadir la información y necesitaría la ayuda de un usuario con acceso al menú del plugin.

En este caso, al ser un usuario independiente que hace uso de una web con un plugin para visualizar datos, el escenario realista sería que formara parte de una empresa y que por tanto habría lógicamente presupuesto para poder gastar alrededor de unos 100 euros entre un plugin de pago y un sistema web con dominio no local.

4. Ejemplo de prueba

En este punto se procederá a explicar las diferentes pruebas realizadas para comprobar el correcto funcionamiento de la web y sus diferentes características.

4.1. Envío correcto de formulario

En primer lugar, será necesario comprobar si al rellenar un formulario conectado como usuario, este se envía correctamente sin dar errores.

Para ello se accedió a la web sin iniciar sesión, es decir, como lo haría un usuario externo al museo, y se rellenó el formulario con información inventada asociada a los campos. Se pudo comprobar como en el campo “Móvil de contacto” no dejaba escribir otra cosa que no fueran números, y en el campo “Correo de contacto” solo se aceptaba una dirección de correo como válida. Por tanto, el funcionamiento de estos es el correcto.

Al estar indicadas como obligatorias muchas entradas, en caso de querer enviar el formulario sin rellenar una de ellas se indicaba un error, tal y como debía suceder.

Finalmente, tras finalizar el formulario se clicó el botón enviar y el resultado fue correcto.

96 387 72 00 info.museu@etsinf.upv.es

museu
informàtica

INICIO RESERVAR ADMINISTRADOR CONTACTO

Reservar Actividad

Su historia es reciente, pero hoy sería impensable un mundo sin Informática. En la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València, la ETSINF – UPV, tenemos propuestas didácticas que pueden contribuir a que sus alumnos y alumnas conozcan mucho más de lo que imaginan sobre el pasado, presente y futuro de la Informática.

Envío realizado correctamente!

Contacto:
info.museu@etsinf.upv.es

Ubicación:
Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València
Camino de Vera, 46022 Valencia

Horario:
Lunes a Viernes de 9:00h a 20:00h
Sábados de 9:00h a 14:00h

Twitter Facebook Instagram YouTube

Imagen 22: En la imagen se puede observar el mensaje de envío correcto de formulario de la web.

4.2. Recepción correcta de formulario

Tras comprobar el envío del formulario, fue necesario comprobar la recepción adecuada del mismo. Para ello, tras realizar la prueba anterior, se comprobó si la información referente al envío había sido recibida en el correo asociado a la recepción.

Tras acceder a la bandeja de entrada, se encontró la información adecuadamente en un mail.

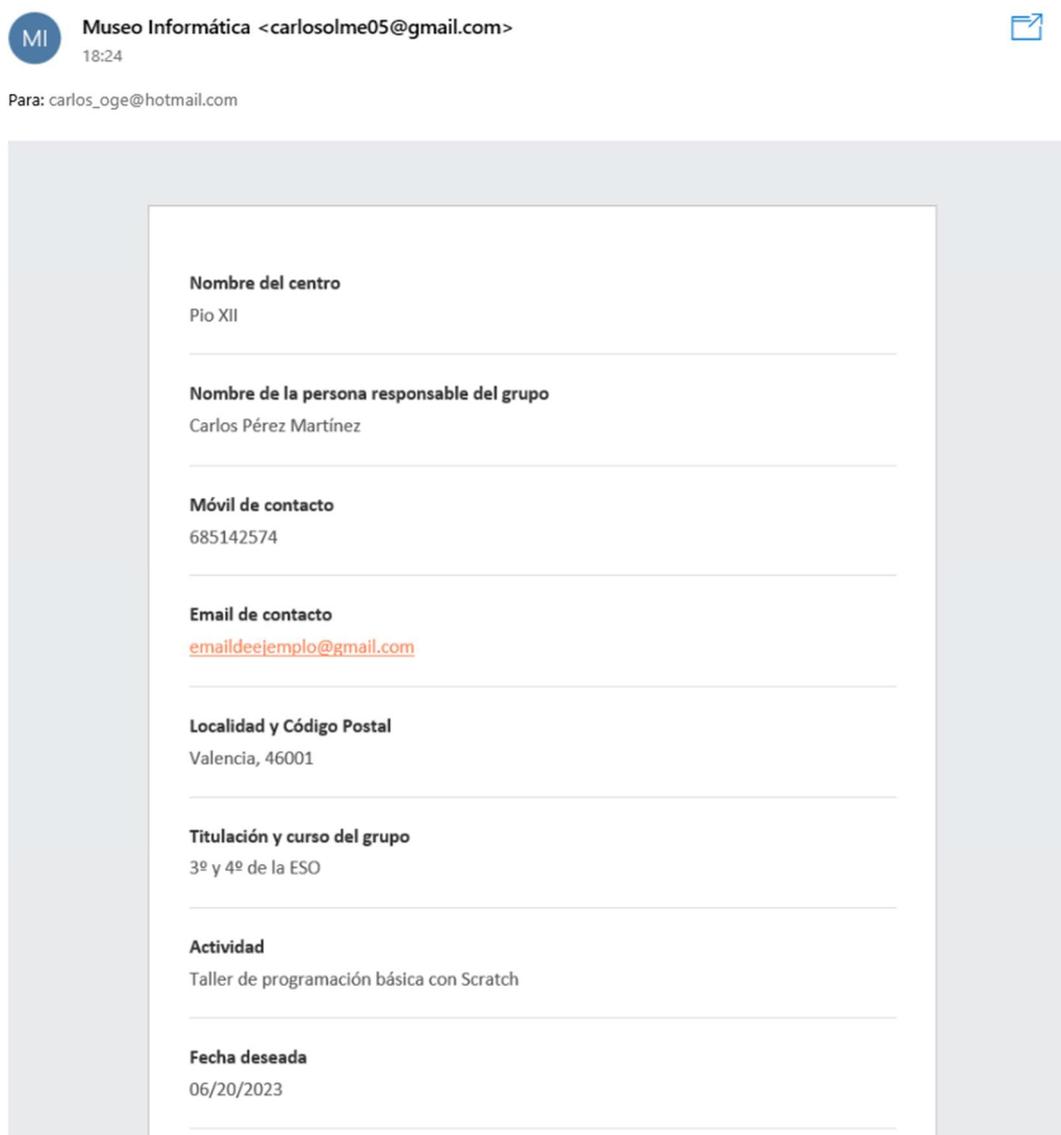


Imagen 23: En la imagen se puede observar el correo recibido con la información relativa al formulario rellenado a modo de ejemplo en el punto de "Envío correcto de formulario".

4.3. Visualización correcta de la información en modo administrador

Para realizar esta prueba, se comprobó si la información y la relación establecida en las tablas era correcta. Para ello, se estableció un ejemplo de consulta mínimamente avanzada para comprobar si era posible encontrar los datos necesarios adecuadamente.

La consulta fue “Obtener las actividades que necesitan más de 20 ordenadores modernos para que puedan ser realizadas”. Para visualizar la información necesaria se inició sesión como administrador y posteriormente se accedió a la tabla “Equipamiento de actividad”.

En esta página se pueden encontrar las tablas referentes a las diferentes actividades, la relación del código de actividad con el código de equipamiento, y el equipamiento general. Primero se buscó el código de equipamiento de los ordenadores modernos, posteriormente se filtró por código de equipamiento y cantidad mayor a 15, y finalmente tras obtener los códigos de actividad correspondientes, se buscaron a que actividades hacían referencia.

Constructor de búsqueda (1) Borrar todo

< Cantidad Mayor a 15 x

x Añadir condición

Mostrar 10 registros

Buscar: EQ001

CODIGO ACTIVIDAD	CODIGO EQUIPAMIENTO	CANTIDAD
A03	EQ001	20
A04	EQ001	20

Mostrando 1 a 2 de 2 registros (filtrado de un total de 24 registros) 0 filas seleccionadas Anterior 1 Siguiente

Constructor de búsqueda

Añadir condición

Mostrar 10 registros

Buscar: Ordenador moderno

CODIGO EQUIPAMIENTO	NOMBRE
EQ001	Ordenador moderno

Mostrando 1 a 1 de 1 registros (filtrado de un total de 4 registros) 0 filas seleccionadas Anterior 1 Siguiente

Imagen 24: En la imagen se puede ver la filtración de datos en las tablas. El resultado A03 y A04 fue comparado con la tabla actividades para obtener los nombres correspondientes.

4.4. Creación de nueva reserva

Con la información enviada en el formulario de prueba del punto 4.1 se realizó una creación de reserva a partir de los datos indicados.

En primer lugar, se observó la disponibilidad de la actividad seleccionada (Taller de programación básica con Scratch). La disponibilidad era correcta.

En segundo lugar, se comprobaron los espacios que cumplían con los requisitos de material de la actividad solicitada. Se obtuvieron los códigos de espacio E0001, E0002, E0005. En siguiente lugar se compararon los horarios de estos lugares con la fecha indicada por el usuario. En este caso la fecha es martes, por tanto, sería necesario comprobar la disponibilidad de las clases en esta fecha. Se observó que la clase 2 estaba ocupada parte de la mañana, por tanto, se eligió el espacio E0001.

Posteriormente se revisaron los conocimientos de los monitores para determinar los que podrían participar en la actividad. Al disponer de conocimientos de programación y Scratch y no tener ocupado el martes, se eligió a la monitora Clara Pérez Muñoz.

Con esto se seleccionaron los materiales necesarios para la actividad y se comprobó que había stock suficiente.

Con la actividad pendiente de realización se añadieron las filas necesarias a las tablas “dispone”, “reserva actividad”, “responsable” y “centro”. Las tablas “grupo”, “entrada sostenible” y “contiene_pieza” serían actualizadas tras la realización de la reserva, ya que hay datos que no son posibles de rellenar previamente a la realización de la misma.

Por tanto, es posible añadir una reserva y la finalidad de la web queda satisfecha.



5. Conclusiones

Para finalizar con la memoria del trabajo, se va a hacer un repaso del cumplimiento de los objetivos del TFG, y del impacto personal que ha tenido en mí su realización. Finalmente se comentarán las limitaciones que se han encontrado durante el proceso y que factores han impedido un resultado mejor.

5.1. Cumplimiento de objetivos

En cuanto a los objetivos propuestos, se han visto cumplidos en su totalidad:

1. Creación de una base de datos con la información relevante del museo de informática
2. Realización de una web para visualizar esta base de datos en modo administrador
3. Automatización del proceso de reserva en el museo de informática
4. Mayor facilidad para visualizar y organizar la información

Por tanto, se considera como un éxito el resultado final, ya que todo lo que se propuso realizar se consiguió de forma positiva.

Con el nuevo sistema, el museo de informática podría ver mejorada su gestión de las reservas y la información importante relativa a este. La automatización conseguida permite que los empleados del museo puedan realizar sus tareas de una forma más rápida y ordenada.

Al ser Wordpress el sistema usado para la realización de la web, es muy fácil de gestionar la administración de esta y crear nuevos usuarios como administrador.

Si el museo quisiera utilizar el sistema que he creado, tendrían que asignar un dominio no local a la web y generar un SSL correspondiente para mejorar su seguridad.

5.2. A nivel personal

A nivel personal ha sido una experiencia buena y mala a la vez. Buena en el sentido de lo que he aprendido. Ahora conozco muchas cosas relacionadas con las bases de datos y el diseño web que antes desconocía, además, las he podido llevar a la práctica con un resultado satisfactorio.

El llevar a la práctica conocimientos que he adquirido durante la carrera, y el ver que he podido realizarlo correctamente es una muy buena sensación.

La parte mala ha sido que me he visto muy limitado. Muchísimas cosas que hubiese querido implantar en la web eran de pago. Las alternativas gratuitas eran muy pobres y he tenido que buscar muchísimas opciones distintas hasta dar con las mejores sin tener que gastar dinero. El querer añadir muchas funcionalidades extra ha sido frustrante, pero aun sin poder añadir ciertos aspectos, el objetivo del trabajo queda cumplido totalmente.

5.3. Limitaciones

Las limitaciones que se han encontrado han sido mayormente económicas. En cuanto a la base de datos no ha habido ningún problema, pero en cuanto a la web sí.

Primero se encontraron limitaciones en el uso de un dominio no local. La mayoría de los dominios eran de pago, y no precisamente de bajo importe.

También se encontraron limitaciones en el uso de plugins para conectar la web y la base de datos. Excepto uno, que se encontró después de largas búsquedas, todos los demás eran de pago.

Existía la posibilidad de enlazar la web y la base de datos con código, pero a nivel gráfico las tablas eran muy poco estéticas y el diseño era realmente simple y pobre visualmente.

Finalmente, se encontraron limitaciones en la creación del formulario, ya que muchas opciones de cuadros de texto y estilo de formulario también eran de pago.

Por tanto, la limitación del trabajo fue el aspecto económico. Se pudo realizar el proyecto a la perfección sin gastar, pero a nivel estético y sobre todo en cuanto a funcionalidades extra de la conexión de la web y la base de datos, hubieron limitaciones.



6. Bibliografía

- Anónimo. (s.f.). *Adimen*. Obtenido de https://adimen.si.ehu.es/~rigau/teaching/EHU/ABD/Altres%20cursos/mjortin/bd_t10_df.pdf. Acceso(15/04/2023)
- Bhaskar, A. (12 de Marzo de 2023). *BforBloggers*. Obtenido de <https://bforbloggers.com/es/revisi%C3%B3n-del-tema-oceanwp-wordpress-pros-contras/>. Acceso(15/04/2023)
- Borges, S. (19 de Noviembre de 2019). *infranetworking*. Obtenido de <https://blog.infranetworking.com/servidor-postgresql/>. Acceso(15/04/2023)
- Cardozo, S. (17 de Agosto de 2021). *WPForms: Mejor Plugin De Formularios Para Wordpress (2022)*. Obtenido de <https://rankerstudio.com/wpforms/>. Acceso(16/04/2023)
- ChatGPT. (6 de Mayo de 2023). *Comunicación personal*. . Acceso(6/05/2023)
- Ciroapp. (1 de Mayo de 2023). *Ciroapp*. Obtenido de <https://ciroapp.com/es/wpdatatables-review/>. Acceso(7/05/2023)
- Enlinea.biz. (s.f.). *Enlinea.biz*. Obtenido de <https://www.enlinea.biz/ventajas-y-desventajas-de-squarespace/>. Acceso(7/05/2023)
- GPT, C. (26 de Mayo de 2023). *Comunicación personal*. . Acceso(26/05/2023)
- Hostcupon. (17 de Enero de 2022). *Hostcupon*. Obtenido de <https://www.hostcupon.com/oceanwp/>. Acceso(1/06/2023)
- KME. (1 de Diciembre de 2018). *Kolmite*. Obtenido de <https://kolmite.com/ninja-tables-complemento-de-tablas-de-datos-wp-para-wordpress/>. Acceso(1/06/2023)
- Profesionalpress. (18 de Marzo de 2023). *Profesionalpress*. Obtenido de <https://profesionalpress.com/elementor-la-herramienta-definitiva-para-crear-sitios-web-con-diseno-profesional-en-wordpress/>. Acceso(1/06/2023)
- Rootstack. (2022 de Junio de 8). *Rootstack*. Obtenido de <https://rootstack.com/es/blog/que-es-oracle-y-cuales-son-sus-caracteristicas>. Acceso(3/06/2023)
- Salas, F. (11 de Abril de 2023). *PRO Elements: la herramienta gratuita que añade funcionalidades PRO a Elementor*. Obtenido de <https://www.pluginsweb.es/pro-elements-wordpress-plugin/>. Acceso(12/06/2023)
- Sqllearning. (s.f.). *Sentencia INSERT INTO en SQL*. Obtenido de <https://sqllearning.com/es/creacion-gestion-bases-datos/insert/>. Acceso(11/06/2023)

UPV. (s.f.). *Bases de Datos y Sistemas de información: Diseño lógico*. Obtenido de https://poliformat.upv.es/access/content/attachment/GRA_11596_2022/Tareas/c330440a-1b31-4b38-a363-fb71f8718df8/Dise%C3%B1o%20L%C3%B3gico.pdf. Acceso(10/04/2023)

WPForms - WordPress Forms Plugin. (14 de Noviembre de 2022). *How to Set Up WP Mail SMTP with Gmail (Fix Failed Emails For Good!)*. Obtenido de <https://www.youtube.com/watch?v=m7wdo16ARSU>. Acceso(20/05/2023)

7. Anexo

7.1. Anexo 1

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.	X			
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Los objetivos que pueden ser relacionados con mi aplicación web son:

ODS 8. Trabajo decente y crecimiento económico: Se considera que la aplicación web realizada permite una situación de trabajo favorable para el empleado. Con las facilidades ofrecidas, las condiciones para desarrollar las tareas necesarias en el ámbito laboral son más que decentes. Además, el sistema implantado permite una mejoría respecto al anterior y, por tanto, un crecimiento económico garantizado.

ODS 12. Producción y consumo responsables: En cuanto a la producción y el consumo responsables este proyecto permite limitar casi en su totalidad la necesidad de utilizar materiales contaminantes. La producción se verá muy reducida, y por tanto el consumo de ciertos materiales se realizará de una forma responsable.

ODS 13. Acción por el clima: En cuanto a la acción por el clima, en este proyecto uno de los objetivos es ahorrar en el uso de material para anotar datos. La finalidad es no malgastar hojas y material para llevar la contabilidad de las reservas de actividades. Al ser un sistema informático en el cual se puede gestionar toda la administración sin necesidad de usar ningún material físico, se están dejando de utilizar recursos que contribuyen a la contaminación y, por tanto, se está contribuyendo a una acción por el clima.

Para concluir, estoy muy contento respecto a cómo mi trabajo ha contribuido con la consecución de los objetivos de desarrollo sostenibles de la UE.

Si todos intentásemos enfocar nuestro trabajo un poco a mejorar y contribuir con estos objetivos, el mundo sería un lugar mucho mejor. Estoy feliz de haber podido demostrar que las tecnologías informáticas pueden ayudar a crear un entorno responsable y sostenible.



7.2. Anexo 2

En este anexo se podrán ver las instrucciones de inserción sql de siete tablas del sistema de información del museo de informática a modo de ejemplo.

1- Tabla 'persona':

```
CREATE TABLE `museo_inf`.`persona` (  
  `dni` VARCHAR(10) NOT NULL,  
  `nombre_propio` VARCHAR(25) NOT NULL,  
  `apellidos` VARCHAR(55) NOT NULL,  
  `telefono` INT NOT NULL,  
  `correo` VARCHAR(50) NOT NULL,  
  `localidad` VARCHAR(25) NULL,  
  `cp` INT NULL,  
  PRIMARY KEY (`dni`));
```

2- Tabla 'responsable':

```
CREATE TABLE `museo_inf`.`responsable` (  
  `dni` VARCHAR(10) NOT NULL,  
  PRIMARY KEY (`dni`),  
  CONSTRAINT `responsable_persona`  
  FOREIGN KEY (`dni`)  
  REFERENCES `museo_inf`.`persona` (`dni`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE);
```

3- Tabla 'grupo':

```
CREATE TABLE `museo_inf`.`grupo` (  
  `codigo_grupo` VARCHAR(25) NOT NULL,  
  `chicos_alumnos` INT NOT NULL,  
  `chicos_profesores` INT NOT NULL,  
  `chicas_alumnas` INT NOT NULL,
```

```

`chicas_profesoras` INT NOT NULL,
`asignatura` VARCHAR(50) NOT NULL,
`observaciones` VARCHAR(500) NULL,
`dni` VARCHAR(10) NOT NULL,
`codigo_centro` VARCHAR(15) NOT NULL,
PRIMARY KEY (`codigo_grupo`),
INDEX `grupo_centro_idx` (`codigo_centro` ASC) VISIBLE,
INDEX `grupo_responsable_idx` (`dni` ASC) VISIBLE,
CONSTRAINT `grupo_responsable`
    FOREIGN KEY (`dni`)
    REFERENCES `museo_inf`.`responsable` (`dni`)
    ON DELETE CASCADE,
    ON UPDATE CASCADE,
CONSTRAINT `grupo_centro`
    FOREIGN KEY (`codigo_centro`)
    REFERENCES `museo_inf`.`centro` (`codigo_centro`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

```

4- Tabla 'reserva actividad':

```

CREATE TABLE `museo_inf`.`reserva_actividad` (
    `codigo_reserva` VARCHAR(50) NOT NULL,
    `codigo_actividad` VARCHAR(50) NOT NULL,
    `fecha` DATE NOT NULL,
    `hora_inicio` TIME NOT NULL,
    `duracion` INT NOT NULL,
    `num_participantes` INT NOT NULL,
    `codigo_espacio` VARCHAR(25) NOT NULL,
    `codigo_grupo` VARCHAR(25) NOT NULL,
    PRIMARY KEY (`codigo_reserva`),

```



```
INDEX `reserva_actividad_espacio_idx` (`codigo_espacio` ASC) VISIBLE,  
INDEX `reserva_actividad_actividad_idx` (`codigo_actividad` ASC) VISIBLE,  
INDEX `reserva_actividad_grupo_idx` (`codigo_grupo` ASC) VISIBLE,  
CONSTRAINT `reserva_actividad_espacio`  
  FOREIGN KEY (`codigo_espacio`)  
  REFERENCES `museo_inf`.`espacio` (`codigo_espacio`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
CONSTRAINT `reserva_actividad_actividad`  
  FOREIGN KEY (`codigo_actividad`)  
  REFERENCES `museo_inf`.`actividad` (`codigo_actividad`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
CONSTRAINT `reserva_actividad_grupo`  
  FOREIGN KEY (`codigo_grupo`)  
  REFERENCES `museo_inf`.`grupo` (`codigo_grupo`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE);
```

5- Tabla 'entrada sostenible':

```
CREATE TABLE `museo_inf`.`entrada_sostenible` (  
  `codigo_reserva` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`codigo_reserva`),  
  CONSTRAINT `entrada_sostenible_reserva_actividad`  
    FOREIGN KEY (`codigo_reserva`)  
    REFERENCES `museo_inf`.`reserva_actividad` (`codigo_reserva`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE);
```

6- Tabla 'pieza':

```
CREATE TABLE `museo_inf`.`pieza` (  

```

`codigo_pieza` VARCHAR(10) NOT NULL,

`descripcion` VARCHAR(25) NOT NULL,

7- Tabla 'contiene pieza':

CREATE TABLE `museo_inf`.`contiene_pieza` (

`codigo_pieza` VARCHAR(10) NOT NULL,

`codigo_reserva` VARCHAR(50) NOT NULL,

`cantidad` INT NULL,

PRIMARY KEY (`codigo_pieza`, `codigo_reserva`),

INDEX `contiene_pieza_entrada_sostenible_idx` (`codigo_reserva` ASC) VISIBLE,

CONSTRAINT `contiene_pieza_pieza`

FOREIGN KEY (`codigo_pieza`)

REFERENCES `museo_inf`.`pieza` (`codigo_pieza`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `contiene_pieza_entrada_sostenible`

FOREIGN KEY (`codigo_reserva`)

REFERENCES `museo_inf`.`entrada_sostenible` (`codigo_reserva`)

ON DELETE CASCADE

ON UPDATE CASCADE);



7.3. Anexo 3

En este anexo se mostrarán varios ejemplos extendidos de inserción sql de varias tablas del sistema de información del museo de informática.

```
INSERT INTO `museo_inf`.`persona` (`dni`, `nombre_propio`, `apellidos`,  
`telefono`, `correo`, `localidad`, `cp`) VALUES ('23654268S', 'Carlos', 'Martínez  
García', '685201576', 'correo1@gmail.com', 'Valencia', '46003');
```

```
INSERT INTO `museo_inf`.`persona` (`dni`, `nombre_propio`, `apellidos`,  
`telefono`, `correo`, `localidad`, `cp`) VALUES ('15246752G', 'Clara', 'Pérez Muñoz',  
'624815672', 'correo2@gmail.com', 'Valencia', '46001');
```

```
INSERT INTO `museo_inf`.`persona` (`dni`, `nombre_propio`, `apellidos`,  
`telefono`, `correo`, `localidad`, `cp`) VALUES ('26715648F', 'Lucas', 'López Álvarez',  
'634861245', 'correo3@gmail.com', 'Valencia', '46002');
```

```
INSERT INTO `museo_inf`.`persona` (`dni`, `nombre_propio`, `apellidos`,  
`telefono`, `correo`, `localidad`, `cp`) VALUES ('64123548V', 'Nuria', 'Torres Silva',  
'674256981', 'correo5@gmail.com', 'Valencia', '46003');
```

```
INSERT INTO `museo_inf`.`persona` (`dni`, `nombre_propio`, `apellidos`,  
`telefono`, `correo`, `localidad`, `cp`) VALUES ('23654125C', 'Gerard', 'Marqués  
Pérez', '632548521', 'correo5@gmail.com', 'Valencia', '46008');
```

```
INSERT INTO `museo_inf`.`conocimiento` (`codigo_conocimiento`, `competencia`)  
VALUES ('CM001', 'Programación en Scratch');
```

```
INSERT INTO `museo_inf`.`conocimiento` (`codigo_conocimiento`, `competencia`)  
VALUES ('CM002', 'Picoboard');
```

```
INSERT INTO `museo_inf`.`conocimiento` (`codigo_conocimiento`, `competencia`)  
VALUES ('CM003', 'Criptografía');
```

```
INSERT INTO `museo_inf`.`conocimiento` (`codigo_conocimiento`, `competencia`)  
VALUES ('CM004', 'Robótica');
```

```
INSERT INTO `museo_inf`.`conocimiento` (`codigo_conocimiento`, `competencia`)  
VALUES ('CM005', 'Historia');
```

```
INSERT INTO `museo_inf`.`reserva_actividad` (`codigo_reserva`,  
`codigo_actividad`, `fecha`, `hora_inicio`, `duracion`, `num_participantes`,  
`codigo_espacio`, `codigo_grupo`) VALUES ('R0001', 'A07', '2023-05-25', '12:00:00',  
'120', '24', 'E0005', 'G0001');
```