



INFORME FINAL DE RESIDENCIA PROFESIONAL

No. CONTROL
191K0284

NOMBRE DEL RESIDENTE
VICTOR MANUEL UICAB NAHUAT.

SEM. 10 GRUPO F

CARRERA: INGENIERÍA EN SISTEMAS COMPUTACIONALES

EMPRESA Ó DEPENDENCIA:
“GRAND VELAS RIVIERA MAYA”

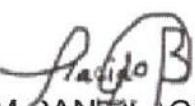
NOMBRE DEL PROYECTO

“Diseño e implementación de una plataforma e-learning para capacitación de lujo y elegancia para los colaboradores de Grand Velas Riviera Maya”

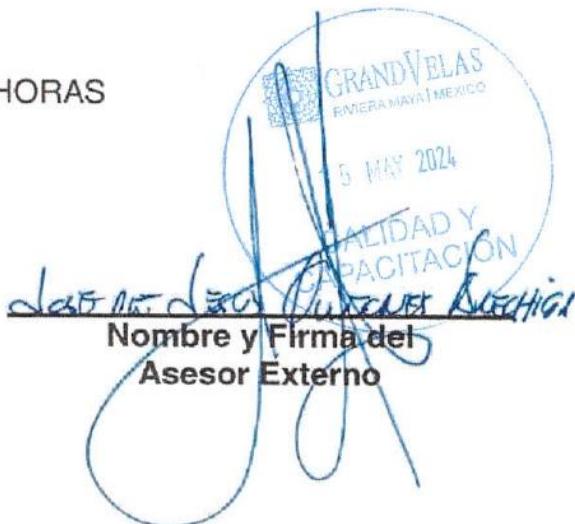
PERIODO DE LA RESIDENCIA PROFESIONAL
DEL 15 DE ENERO DE 2024 AL 15 DE MAYO DE 2024

TOTAL DE HORAS

500


BALAM CAN PLACIDO

Nombre y Firma del
Asesor Interno




Nombre y Firma del
Asesor Externo



INFORME FINAL DE RESIDENCIA PROFESIONAL

AGRADECIMIENTOS

Quisiera expresar mi más sincero agradecimiento a todas las personas e instituciones que han contribuido de manera significativa al desarrollo y realización de este proyecto de residencia profesional en Grand Velas Riviera Maya.

En primer lugar, deseo agradecer al equipo de Calidad y capacitación y de sistemas (IT) de Grand Velas Riviera Maya por su apoyo incondicional, orientación y confianza en este proyecto. Su compromiso con la excelencia, actitud y la innovación ha sido fundamental para el éxito de esta iniciativa.

Agradezco especialmente a Marco Antonio Martinez Ortega mi supervisor del departamento de sistemas, por su orientación experta, su valiosa retroalimentación y su dedicación para ayudarme a alcanzar mis metas académicas y profesionales.

También quiero reconocer el trabajo y la colaboración de mis compañeros y colegas, cuya contribución ha enriquecido este proyecto con sus ideas, sugerencias y esfuerzos colectivos.

Además, agradezco a mi familia y amigos por su constante apoyo, comprensión y ánimo durante todo este proceso de aprendizaje y desarrollo profesional.

Finalmente, extendemos nuestro agradecimiento a cualquier otra persona o entidad que haya brindado su apoyo, orientación o recursos para hacer posible la realización de este proyecto.

Sin la ayuda y el respaldo de todas estas personas e instituciones, este proyecto no habría sido posible. A todos ustedes, ¡muchas gracias!

INFORME FINAL DE RESIDENCIA PROFESIONAL

RESUMEN

El presente proyecto de residencia profesional tiene como objetivo principal el diseño e implementación de una plataforma E-Learning de lujo para la capacitación de los colaboradores de Grand Velas Riviera Maya. Esta plataforma se desarrollará con el propósito de mejorar la eficiencia y la calidad del servicio al cliente, proporcionando acceso a contenidos de capacitación personalizados y flexibles.

El proyecto se delimita a la creación de la plataforma E-Learning centrada específicamente en la capacitación de los colaboradores de Grand Velas Riviera Maya, abordando temas relacionados con el servicio al huésped de lujo, protocolos de seguridad, hospitalidad y otros aspectos relevantes para la industria hotelera. Se establece un cronograma de actividades desarrolladas en cada fase para llevar a cabo el proyecto en fases definidas.

Los objetivos del proyecto incluyen el desarrollo de contenido de capacitación personalizado, garantizar la accesibilidad de la plataforma para todos los colaboradores, implementar un sistema de seguimiento y evaluación, cumplir con un cronograma de implementación y mejorar la calidad del servicio al cliente. Se busca fomentar el desarrollo profesional de los colaboradores y elevar los estándares de la industria hotelera.

La justificación del proyecto se fundamenta en su conveniencia para proporcionar una solución moderna y efectiva para la capacitación de los colaboradores, su relevancia social al beneficiar a los colaboradores y huéspedes, y sus implicaciones prácticas al resolver el problema real de la capacitación en una industria competitiva. Además, el proyecto tiene valor teórico al llenar un vacío en el conocimiento sobre la implementación de plataformas E-Learning de lujo en la industria hotelera.

El cronograma preliminar de actividades incluye etapas como la reunión de planificación, investigación y análisis, diseño de la arquitectura, desarrollo del contenido, implementación de la plataforma, pruebas y depuración, capacitación del personal, lanzamiento y evaluación, y evaluación y ajuste.



INFORME FINAL DE RESIDENCIA PROFESIONAL

El proyecto se llevará a cabo en Grand Velas Riviera Maya, un resort reconocido por ofrecer experiencias excepcionales de viaje. Se cuenta con el apoyo y colaboración del equipo de calidad y capacitación de la empresa para garantizar la alineación con sus objetivos y estándares de servicio.

ÍNDICE

INTRODUCCIÓN	6
PROBLEMAS A RESOLVER, PRIORIZÁNDOLOS.	7
OBJETIVOS.....	9
JUSTIFICACIÓN.....	10
MARCO TEÓRICO (FUNDAMENTOS TEÓRICOS).....	11
PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.....	16
RESULTADOS, PLANOS, GRÁFICAS, PROTOTIPOS, MANUALES, PROGRAMAS, ANÁLISIS ESTADÍSTICOS, MODELOS MATEMÁTICOS, SIMULACIONES, NORMATIVAS, REGULACIONES Y RESTRICCIONES, ENTRE OTROS. SOLO PARA PROYECTOS QUE POR SU NATURALEZA LO REQUIERAN: ESTUDIO DE MERCADO, ESTUDIO TÉCNICO Y ESTUDIO ECONÓMICO.....	18
2. Diseño prototipo web:.....	19
3. Página web (Usuario):.....	22
4. Página web (Administrador):	30
5. Base de datos:	36
Ilustración 6. Diagrama Casos de Uso:.....	44
7. Manual de Usuario:	45
8. Manual de Administrador:	48
9. Explicación del código:.....	53
CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA PROFESIONAL ADQUIRIDA.....	103
COMPETENCIAS DESARROLLADAS Y/O APLICADAS.....	106
FUENTES DE INFORMACIÓN	107
ANEXOS.....	109

INTRODUCCIÓN

La industria hotelera se caracteriza por su constante evolución y búsqueda de la excelencia en la atención al cliente. En este contexto, la capacitación y el desarrollo del personal son aspectos cruciales para garantizar experiencias excepcionales para los huéspedes. En este sentido, Grand Velas Riviera Maya, un resort reconocido por ofrecer servicios de lujo y elegancia, reconoce la importancia de contar con un equipo altamente capacitado y comprometido.

El presente proyecto de residencia profesional surge de la necesidad de desarrollar e implementar una plataforma E-Learning de alta calidad para la capacitación de los colaboradores de Grand Velas Riviera Maya. Esta plataforma representa una innovadora herramienta que busca mejorar la eficiencia y la calidad del servicio al cliente, ofreciendo acceso a contenidos de capacitación personalizados y flexibles.

La capacitación de los colaboradores es un factor determinante en la satisfacción y fidelización de los huéspedes. Es por ello que este proyecto se enfoca en el diseño de una plataforma que permita ofrecer capacitación de lujo y elegancia, alineada con los estándares de excelencia de Grand Velas Riviera Maya. Los temas abordados incluirán aspectos relacionados con el servicio al huésped, protocolos de seguridad, hospitalidad y otros aspectos relevantes para la industria hotelera.

La implementación de esta plataforma E-Learning no solo beneficiará a los colaboradores al proporcionarles herramientas para mejorar sus habilidades y conocimientos, sino que también impactará de manera positiva en la experiencia de los huéspedes. Se espera que, a través de una capacitación efectiva y personalizada, se eleve la calidad del servicio ofrecido, garantizando estancias inolvidables para los visitantes de Grand Velas Riviera Maya.

En este documento se establece el contexto y la relevancia del proyecto, destacando la importancia de la capacitación del personal en la industria hotelera y la necesidad de contar con herramientas innovadoras para su desarrollo. A lo largo de este informe, se detallarán los objetivos, la metodología, los resultados esperados, las implicaciones prácticas teóricas de

este proyecto de residencia profesional en Grand Velas Riviera Maya así como también los requisitos, diseños, diagramas, funciones, entidades y restricciones.

PROBLEMAS A RESOLVER, PRIORIZÁNDOLOS.

El entorno dinámico y competitivo de la industria hotelera plantea una serie de desafíos que afectan directamente la calidad del servicio ofrecido por los colaboradores de Grand Velas Riviera Maya. Algunos de estos problemas identificados incluyen:

- 1. Falta de Acceso a Capacitación de Calidad:** La capacitación del personal es esencial para mantener altos estándares de servicio al cliente, sin embargo, la falta de acceso a programas de capacitación de calidad puede limitar el desarrollo profesional de los colaboradores. Es crucial abordar esta problemática mediante la implementación de una plataforma E-Learning que brinde contenido de lujo y elegancia, adaptado a las necesidades específicas del personal de Grand Velas Riviera Maya.
- 2. Desafíos en la Coordinación de Capacitación:** La coordinación de programas de capacitación tradicionales puede resultar compleja y costosa, especialmente en un entorno hotelero donde el personal trabaja en horarios variados y rotativos. La falta de flexibilidad en los horarios de capacitación puede dificultar la participación de los colaboradores. La implementación de una plataforma E-Learning permitirá superar estos desafíos al ofrecer acceso 24/7 a los contenidos de capacitación, permitiendo que los colaboradores puedan aprender a su propio ritmo y desde cualquier lugar.
- 3. Retención de Conocimiento y Actualización de Contenidos:** En un entorno donde el personal está constantemente expuesto a nuevas situaciones y desafíos, es fundamental asegurar que el conocimiento adquirido durante la capacitación se retenga y se actualice de manera continua. La falta de un sistema eficaz de seguimiento y evaluación puede dificultar este proceso, lo que resulta en una pérdida de eficiencia y calidad en el servicio. La implementación de una plataforma E-Learning con funcionalidades de seguimiento y

evaluación permitirá monitorear el progreso de los colaboradores, identificar áreas de mejora y actualizar los contenidos de manera oportuna.

4. Priorización:

Dada la naturaleza interrelacionada de estos problemas, se considera prioritaria la implementación de una plataforma E-Learning de calidad que aborde simultáneamente las siguientes áreas:

4.1 Acceso a Capacitación de Calidad: Garantizar que todos los colaboradores tengan acceso a contenido de capacitación de lujo y elegancia que se adapte a sus necesidades individuales.

4.2 Flexibilidad en la Coordinación de Capacitación: Proporcionar una plataforma que permita a los colaboradores acceder a los contenidos de capacitación en cualquier momento y desde cualquier lugar, facilitando así la participación y el aprendizaje continuo.

4.3 Sistema de Seguimiento y Evaluación: Implementar un sistema efectivo de seguimiento y evaluación que permita medir el progreso de los colaboradores, identificar áreas de mejora y garantizar la retención y actualización constante del conocimiento.

Al abordar estos problemas de manera integral, se espera mejorar significativamente la eficiencia y la calidad del servicio ofrecido por los colaboradores de Grand Velas Riviera Maya, contribuyendo así a una experiencia excepcional para los huéspedes del resort.

OBJETIVOS.

- 1. Desarrollar Contenido de Capacitación Personalizado:** Crear contenido de capacitación de alta calidad y personalizado que se adapte a las necesidades individuales de los colaboradores, incluyendo módulos sobre servicio al cliente de lujo, protocolos de seguridad y hospitalidad excepcional.
- 2. Garantizar Accesibilidad:** Asegurar que la plataforma E-Learning sea accesible para todos los colaboradores, independientemente de su nivel de competencia tecnológica, a través de una interfaz amigable y soporte técnico adecuado.
- 3. Fomentar el Desarrollo Profesional:** Ofrecer oportunidades de desarrollo y reconocimiento a los colaboradores que busquen avanzar en sus carreras en la industria hotelera.
- 4. Cumplir con un Cronograma de Implementación:** Llevar a cabo la implementación de la plataforma E-Learning en varias fases, siguiendo un cronograma definido que garantice una transición suave y eficiente.
- 5. Mejorar la Calidad del Servicio al Cliente:** Elevar la calidad del servicio al cliente brindado por los colaboradores, lo que se traducirá en experiencias excepcionales para los huéspedes de Grand Velas Riviera Maya.
- 6. Fomentar el Desarrollo Profesional:** Ofrecer oportunidades de desarrollo y certificación a los colaboradores que busquen avanzar en sus carreras en la industria hotelera.
- 7. Mejorar la Calidad del Servicio al Cliente:** Elevar la calidad del servicio al cliente brindado por los colaboradores para garantizar experiencias excepcionales para los huéspedes de Grand Velas Riviera Maya.

JUSTIFICACIÓN.

El diseño e implementación de una plataforma E-Learning de lujo para la capacitación de los colaboradores de Grand Velas Riviera Maya se justifica por su capacidad para abordar de manera efectiva los desafíos actuales en la industria hotelera, relacionados con la calidad del servicio al cliente y el desarrollo profesional del personal.

1. Necesidad de capacitación de calidad:

La industria hotelera de lujo exige un alto nivel de servicio al cliente y atención al detalle. Una plataforma E-Learning permite ofrecer capacitación personalizada y de alta calidad, adaptada a las necesidades específicas de los colaboradores de Grand Velas Riviera Maya. Esto asegura que el personal esté equipado con las habilidades y conocimientos necesarios para brindar experiencias excepcionales a los huéspedes.

2. Flexibilidad y accesibilidad:

La naturaleza dinámica del trabajo en un resort de lujo a menudo implica horarios variables y rotativos. La plataforma E-Learning supera las limitaciones de la capacitación tradicional al ofrecer acceso flexible al contenido, desde cualquier lugar y en cualquier momento. Esto facilita la participación de los colaboradores, independientemente de su horario de trabajo, y promueve el aprendizaje continuo.

3. Eficiencia y efectividad:

La plataforma E-Learning optimiza la inversión en capacitación al reducir costos asociados con la logística y los materiales de los programas tradicionales. Además, el seguimiento y la evaluación integrados en la plataforma permiten medir el progreso del personal de manera efectiva, identificar áreas de mejora y garantizar la retención del conocimiento.

4. Alineación con la visión de Grand Velas Riviera Maya:

El proyecto se alinea directamente con la visión de Grand Velas Riviera Maya de ofrecer experiencias excepcionales de viaje a través de un servicio de lujo y elegancia. Al invertir en la capacitación y el desarrollo del personal mediante una plataforma E-Learning de vanguardia, el resort demuestra su compromiso con la excelencia y la satisfacción del cliente.

Impacto social y económico:

El proyecto tiene un impacto social positivo al empoderar a los colaboradores con nuevas habilidades y conocimientos, lo que les permite avanzar en sus carreras y mejorar su calidad

de vida. Además, la mejora en la calidad del servicio al cliente genera un impacto económico positivo para el resort, al aumentar la satisfacción y la fidelización de los huéspedes.

MARCO TEÓRICO (FUNDAMENTOS TEÓRICOS)

1. Plataformas E-Learning similares:

1.1 Moodle

Descripción: Moodle es una plataforma de aprendizaje en línea de código abierto ampliamente utilizada en instituciones educativas y empresas. Ofrece una amplia gama de funciones para la creación y gestión de cursos, la comunicación y la colaboración entre estudiantes y docentes, y la evaluación del aprendizaje.

Similitudes: Moodle comparte con tu plataforma características como la flexibilidad y accesibilidad, la eficiencia y efectividad, y la alineación con la visión empresarial. Moodle es una plataforma personalizable que se puede adaptar a las necesidades específicas de cada organización. (Moodle. (n.d.))

1.2. Docebo

Descripción: Docebo es una plataforma E-Learning empresarial que ofrece una experiencia de aprendizaje personalizada y atractiva para los colaboradores. Cuenta con funciones como gamificación, aprendizaje social y análisis de aprendizaje para mejorar el engagement y la retención del conocimiento.

Similitudes: Docebo comparte con tu plataforma la capacidad de ofrecer capacitación de alta calidad y personalizada a los colaboradores. Docebo utiliza tecnologías innovadoras para crear experiencias de aprendizaje más efectivas y atractivas. (Docebo. (n.d.). Docebo.)

1.3. Cornerstone OnDemand

Descripción: Cornerstone OnDemand es una plataforma E-Learning completa que ofrece una amplia gama de soluciones para la gestión del talento, incluyendo la capacitación, el desarrollo, la evaluación del desempeño y la gestión de la sucesión.

Similitudes: Cornerstone OnDemand comparte con tu plataforma la visión de ofrecer una solución integral para la capacitación y el desarrollo de los colaboradores. Cornerstone OnDemand ofrece una plataforma robusta que puede ayudar a las organizaciones a alcanzar sus objetivos estratégicos. (Cornerstone OnDemand. (n.d.). Cornerstone OnDemand.)

1.4. Totara

Descripción: Totara es una plataforma E-Learning de código abierto diseñada específicamente para organizaciones que necesitan una solución flexible y escalable. Ofrece funciones para la creación de cursos, la gestión de usuarios, la evaluación del aprendizaje y la generación de informes.

Similitudes: Totara comparte con tu plataforma la flexibilidad y escalabilidad. Totara es una plataforma personalizable que se puede adaptar a las necesidades específicas de cada organización. (Totara. (n.d.). Totara)

1.5. SAP Learning Cloud

Descripción: SAP Learning Cloud es una plataforma E-Learning integrada con las soluciones de SAP para ofrecer una experiencia de aprendizaje personalizada y relevante para los usuarios de SAP. Ofrece funciones como gamificación, aprendizaje social y análisis de aprendizaje para mejorar el engagement y la retención del conocimiento.

Similitudes: SAP Learning Cloud comparte con tu plataforma la capacidad de ofrecer capacitación personalizada y relevante para los colaboradores. SAP Learning Cloud se integra con las soluciones de SAP para proporcionar una experiencia de aprendizaje más completa. (SAP. (n.d.). SAP Learning Cloud)

2. Teorías del Aprendizaje

Conductismo: Enfatiza el papel del refuerzo y el castigo en el aprendizaje. El E-Learning puede utilizar técnicas conductistas, como la retroalimentación inmediata y los puntos de recompensa, para fomentar el aprendizaje (Morra, 2020).

Cognitivismo: Se centra en los procesos mentales que intervienen en el aprendizaje. El E-Learning puede utilizar estrategias cognitivas, como la organización de contenidos, la presentación de ejemplos y la resolución de problemas, para facilitar la comprensión y el aprendizaje (Mayer, 2014).

Constructivismo: Sostiene que el aprendizaje es un proceso activo en el que los individuos construyen su propio conocimiento. El E-Learning puede promover el aprendizaje

constructivista mediante actividades que fomenten la reflexión, la colaboración y la resolución de problemas (Vygotsky, 1978).

3. Pedagogía

Los nuevos escenarios de aprendizaje, requieren la creación de estrategias para la acción educativa para unir nodos de conocimientos, mediados por tecnologías que propician la conectividad en cada uno de los participantes. Este se caracteriza por no tener limitaciones en la matriculación, interacción online en todo momento, es abierto y gratuito con un contenido fácil de utilizar. (Bartolomé y Steffens, 2015)

Diseño Instruccional: El E-Learning se basa en principios de diseño instruccional para crear experiencias de aprendizaje efectivas. Estos principios incluyen la definición de objetivos claros, la organización de contenidos en unidades coherentes, la implementación de actividades de aprendizaje variadas y la evaluación del progreso del estudiante (Reigeluth, 2013).

Estrategias de Enseñanza y Aprendizaje: El E-Learning utiliza diversas estrategias de enseñanza y aprendizaje para promover el aprendizaje activo y significativo. Estas estrategias incluyen la exposición, la explicación, la demostración, la práctica, la retroalimentación y la evaluación (Chickering & Reisser, 1993).

4. Psicología del Aprendizaje

Motivación: La motivación es un factor clave en el aprendizaje. El E-Learning puede utilizar diversas técnicas para motivar a los estudiantes, como establecer metas desafiantes, proporcionar retroalimentación positiva y crear un ambiente de aprendizaje positivo (Ryan & Deci, 2000).

Atención: La atención es un recurso limitado que debe ser enfocada para el aprendizaje efectivo. El E-Learning puede utilizar elementos multimedia, actividades interactivas y un diseño atractivo para captar y mantener la atención de los estudiantes (Pashler, 1998).

Memoria: La memoria es esencial para el almacenamiento y la recuperación de información. El E-Learning puede utilizar técnicas para mejorar la memoria, como la repetición, la elaboración y la organización de contenidos (Schacter, Gilbert, & Winocur, 2011).

Adicionalmente, Aguirre (2019) elaboró un estudio de enfoque cuantitativo en la especialidad de Electrónica Industrial que asistían al curso de Matemática I, cuyo objetivo estaba referido a determinar si la utilización de la plataforma Moodle estaba relacionado con el desempeño de los estudiantes. Con respecto a los aspectos metodológicos, el nivel alcanzado por la investigación fue de tipo correlacional, aplicándose un instrumento a una muestra de 46 estudiantes, logrando concluir que para los estudiantes el usar Moodle se relacionaba con un mejor desempeño académico, obteniéndose un índice de correlación positiva de Spearman de 0.609.

Adicionalmente se pudo comprobar la existencia de relación de las dimensiones gestión de contenidos, comunicación y evaluación con el desempeño académico del alumnado.

En el mismo sentido, Álvarez (2018) buscó determinar en estudiantes de educación a distancia la existencia de algún nivel de relación entre el uso de Moodle en la enseñanza y el aprendizaje. La investigación se elaboró tomando en cuenta los supuestos del enfoque cuantitativo, utilizando un diseño experimental. El estudio utilizó una muestra de 55 estudiantes, divididos en 2 grupos. concluyéndose que la plataforma Moodle facilitó el logro del aprendizaje de los estudiantes del curso de informática, en los distintos cursos llevados dentro del programa académico.

Sánchez (2009) señala que las entidades del Estado y Muchas instituciones privadas utilizan la plataforma Moodle para facilitar la administración y organización de la enseñanza. Esto, dado que permite administrar el acceso a múltiples usuarios, desde docentes, estudiantes y administradores, los cuales organizan y desarrollan el proceso de enseñanza aprendizaje. Estos sistemas gestionan de manera adecuada los cursos electrónicos, generando con ello los denominados campus virtuales. En tal sentido, es posible encajar a estos sistemas como administrador de contenidos y de servicio de mensajería, chats y foros de discusión, entre

otros servicios, que en conjunto mejoran notablemente las relaciones sociales entre los alumnos y el docente, implicados en el proceso de enseñanza aprendizaje.

Pruneda (2020) señala que Moodle contiene herramientas de comunicación, se tienen disponibles el acceso a Chats, Foros, Blogs, etc. Además, permite la creación de cuestionarios, lecciones, tareas, entre otros. También es posible configurar, gestionar y editar los cursos. Estas ventajas presentadas, permiten el seguimiento de las actividades de los alumnos y su calificación respectiva. Además, permite realizar mensajes por correo y también de manera interna con avisos por la plataforma.

5. Lista de las herramientas y tecnologías:

1.1 HTML: Para la estructura y el marcado de las páginas web, incluyendo la creación de formularios para la entrada de datos por parte de los usuarios. (World Wide Web Consortium. (n.d.). HTML 5.2.)

1.2 CSS: Para el diseño y la presentación visual de las páginas web, incluyendo estilos para hacer que el contenido sea atractivo y fácil de leer. (World Wide Web Consortium. (n.d.). Cascading Style Sheets (CSS))

1.3 JavaScript: Para la validación de formularios en el lado del cliente, lo que permite verificar que los datos ingresados por los usuarios cumplan con ciertos criterios antes de ser enviados al servidor. (Mozilla Developer Network. (n.d.). JavaScript)

1.4 AJAX: Para enviar datos de formularios al servidor de forma asíncrona, lo que permite actualizar partes específicas de una página web sin necesidad de recargar toda la página. (World Wide Web Consortium. (n.d.))

1.5 PHP: Para procesar los datos recibidos del cliente, realizar operaciones en el servidor (como la verificación de contraseñas utilizando la función hash sha), y generar respuestas dinámicas que se envían de vuelta al cliente. (The PHP Group. (n.d.). Manual de PHP)

1.5 SHA (Secure Hash Algorithm): Para la seguridad de las contraseñas, utilizando funciones de hash como SHA-256 para convertir las contraseñas en cadenas de caracteres irreversibles y seguras. (National Institute of Standards and Technology. (2019))

PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.

1. Reunión de Planificación:

Inicio formal del proyecto mediante una sesión estratégica con el equipo de dirección y recursos humanos de Grand Velas Riviera Maya.

Definición de objetivos, alcances y límites del proyecto.

Establecimiento de roles y responsabilidades clave.

2. Investigación y Análisis:

Conducción de investigaciones exhaustivas y análisis de mercado para identificar requisitos específicos y tendencias relevantes en la capacitación del personal hotelero.

Realización de un análisis competitivo en profundidad para evaluar plataformas E-Learning existentes en la industria.

3. Diseño de la Arquitectura:

Creación de una arquitectura de sistema sólida y escalable, detallando la estructura técnica y la integración de componentes clave de la plataforma E-Learning.

Selección de tecnologías adecuadas para el desarrollo y la implementación de la plataforma.

4. Tipo del Contenido:

Identificar qué tipo de materiales y contenido de capacitación de alta calidad se pueden subir, alineados con los estándares de lujo y elegancia de Grand Velas Riviera Maya.

Visualizar los tipos de módulos de capacitación sobre servicio al cliente de lujo, protocolos de seguridad, hospitalidad excepcional y otros temas relevantes para la industria hotelera.

5. Implementación de la Plataforma:

Desarrollo e implementación de la plataforma E-Learning de acuerdo con la arquitectura diseñada, utilizando las tecnologías seleccionadas.

Configuración de funcionalidades de usuario, como acceso personalizado, seguimiento de progreso y evaluación de desempeño.

6. Pruebas y Depuración:

Aplicación de pruebas exhaustivas para identificar y corregir cualquier defecto o error en la plataforma.

Foco en garantizar la estabilidad, confiabilidad y usabilidad de la plataforma para los colaboradores de Grand Velas Riviera Maya.

7. Capacitación del Personal:

Desarrollo e implementación de programas de formación detallados para el personal sobre el uso efectivo de la plataforma E-Learning.

Entrenamiento en la navegación de la plataforma, acceso al contenido de capacitación y uso de herramientas de seguimiento y evaluación.

8. Lanzamiento y Evaluación:

Despliegue oficial de la plataforma E-Learning para todos los colaboradores de Grand Velas Riviera Maya.

Evaluación inicial del rendimiento y la experiencia del usuario, recopilando retroalimentación para posibles mejoras adicionales.

9. Evaluación y Ajuste:

Ánálisis detallado del rendimiento y la retroalimentación del usuario para identificar áreas de mejora en la plataforma.

Realización de ajustes y mejoras en función de la retroalimentación recibida para garantizar la efectividad continua de la plataforma.

**RESULTADOS, PLANOS, GRÁFICAS, PROTOTIPOS, MANUALES,
PROGRAMAS, ANÁLISIS ESTADÍSTICOS, MODELOS
MATEMÁTICOS, SIMULACIONES, NORMATIVAS,
REGULACIONES Y RESTRICCIONES, ENTRE OTROS. SOLO
PARA PROYECTOS QUE POR SU NATURALEZA LO
REQUIERAN: ESTUDIO DE MERCADO, ESTUDIO TÉCNICO Y
ESTUDIO ECONÓMICO.**

Se logró el desarrollo e implementación exitosa de una plataforma E-Learning totalmente funcional y accesible para todos los colaboradores de Grand Velas Riviera Maya. La plataforma ofrece una interfaz intuitiva y amigable que permite a los usuarios acceder fácilmente al contenido de capacitación desde cualquier dispositivo y en cualquier momento. Se recopiló retroalimentación por parte de los colaboradores sobre la plataforma E-Learning, la cual fue en su mayoría positiva. Los usuarios valoraron especialmente la accesibilidad y la relevancia del contenido de capacitación, así como la flexibilidad que ofrece la plataforma en términos de horarios y ubicación.

2. Diseño prototipo web:

Ilustración 1. Crear cuenta:

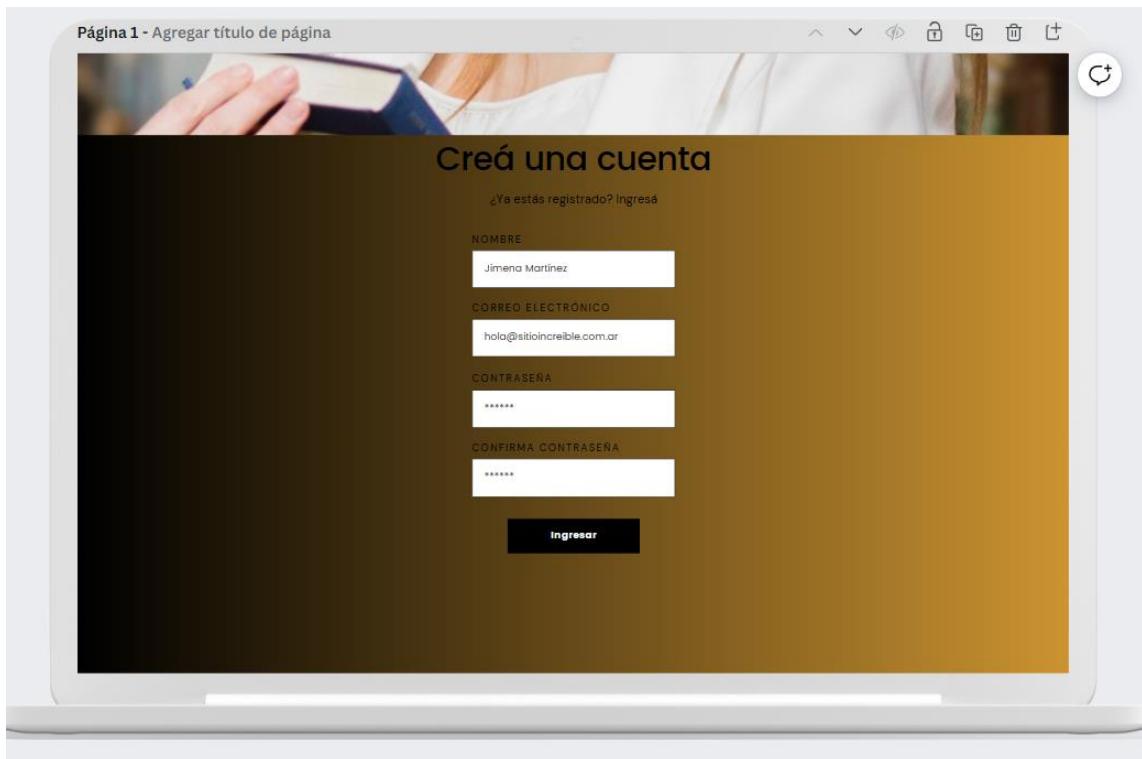


Ilustración 2. Login:

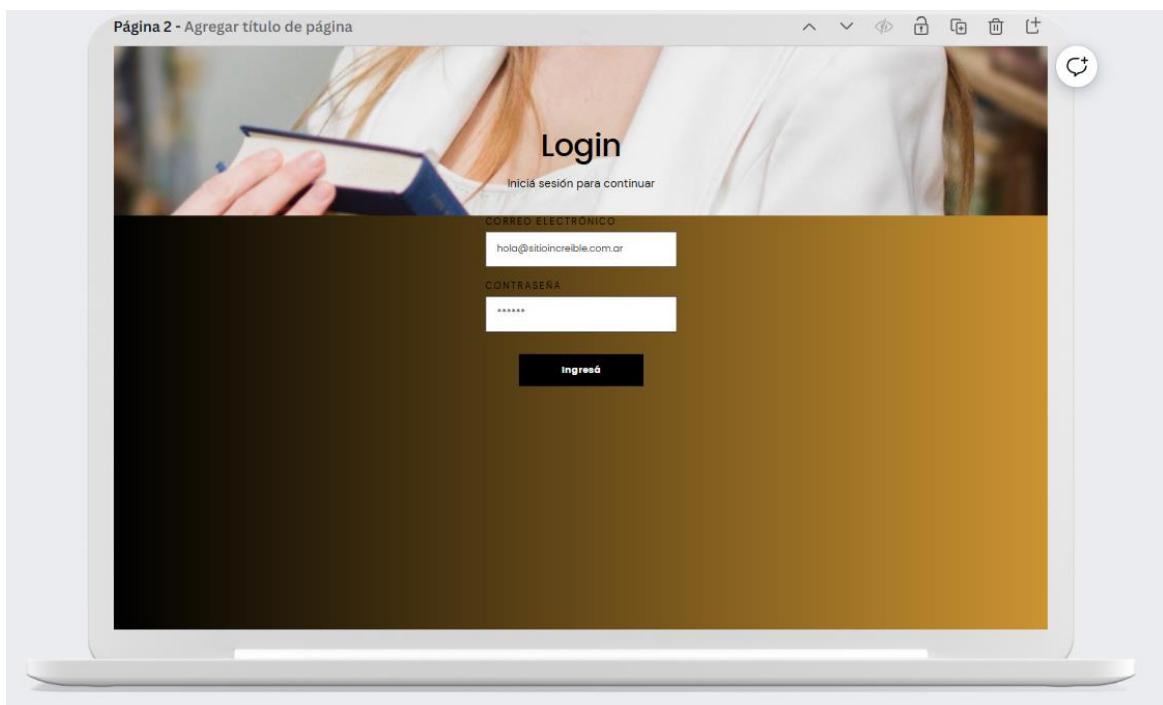


Ilustración 3. Index:

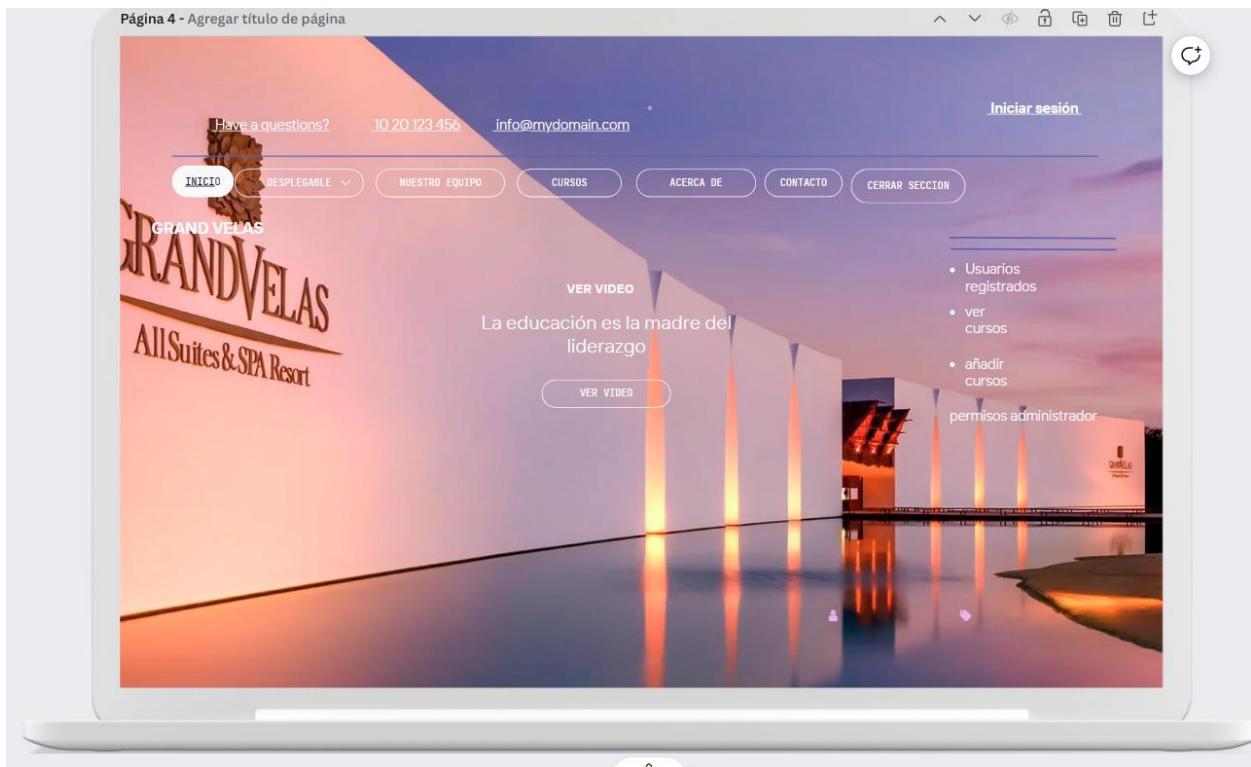


Ilustración 4. Vista cursos:

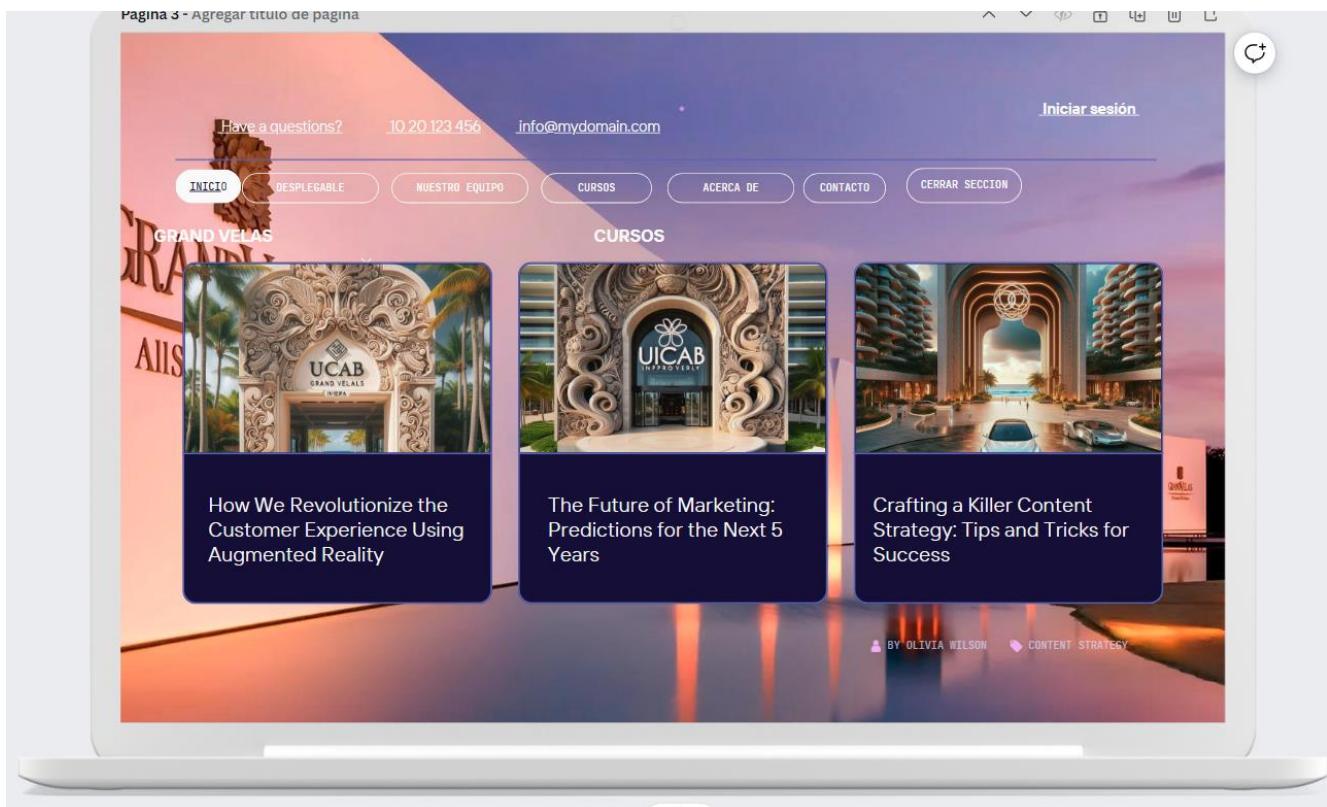


Ilustración 5. Descripción del curso:

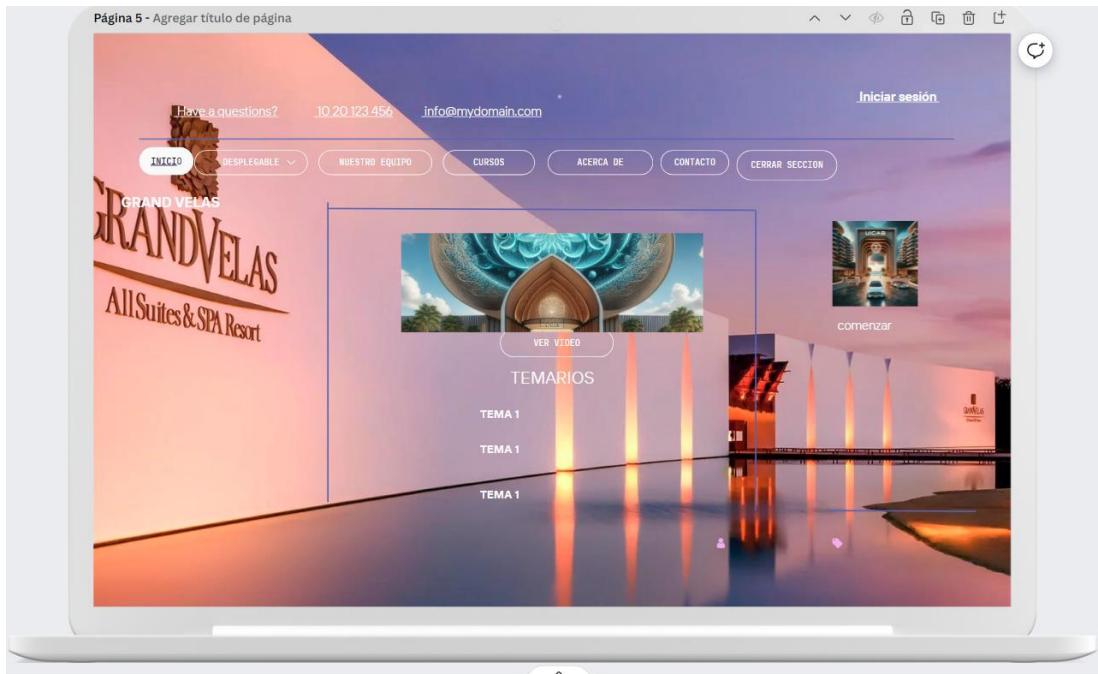
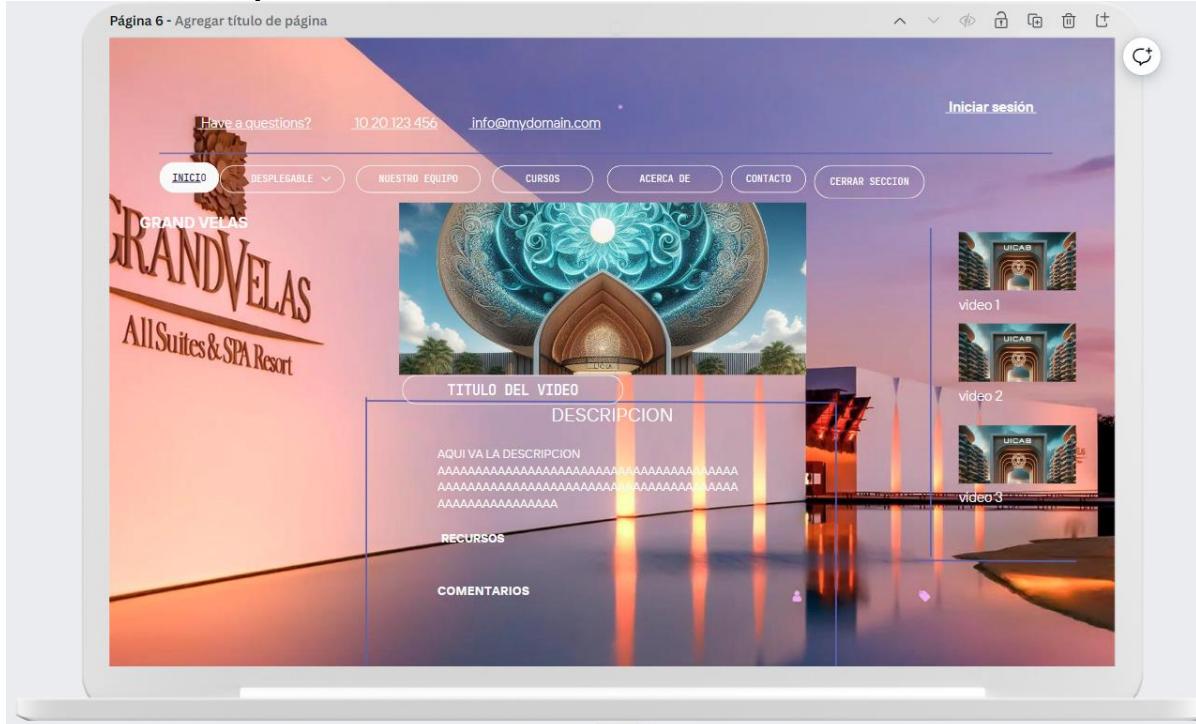


Ilustración 6. Reproductor de video:



3. Página web (Usuario):

Ilustración 1. Login:

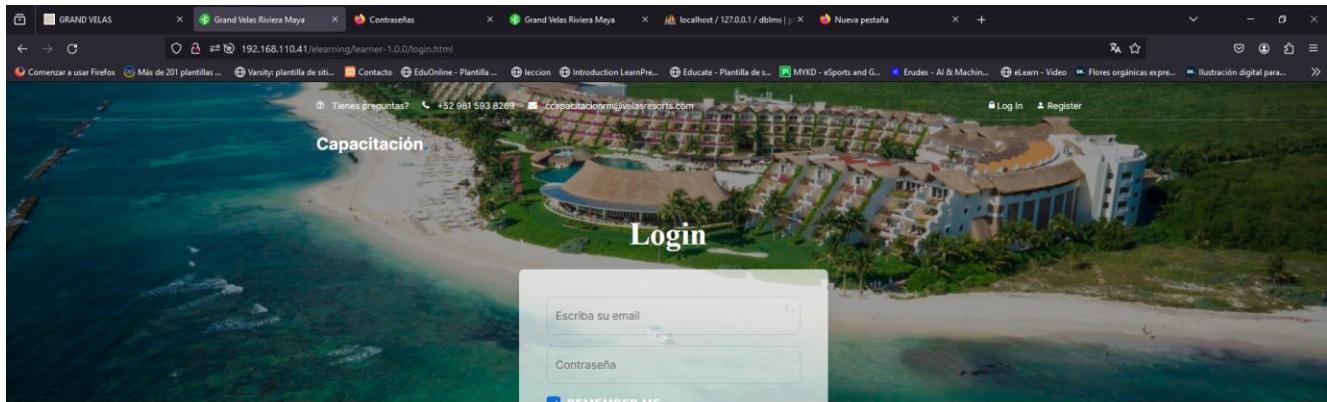


Ilustración 2. Registrar usuario:

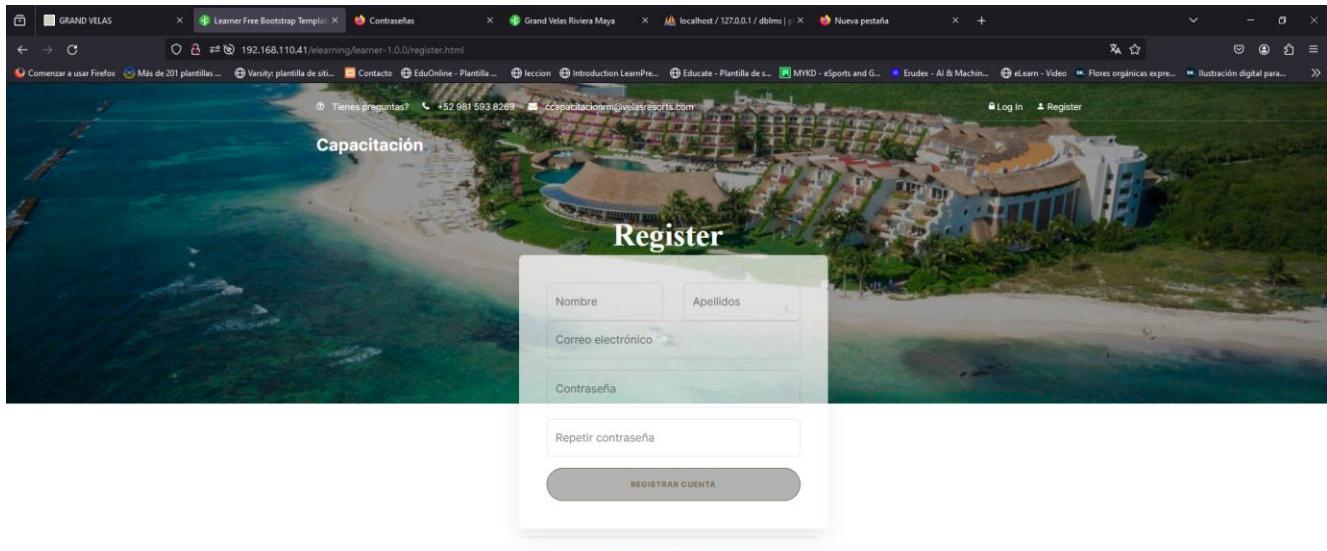


Ilustración 3. Index:



Ilustración 4. Vista cursos:

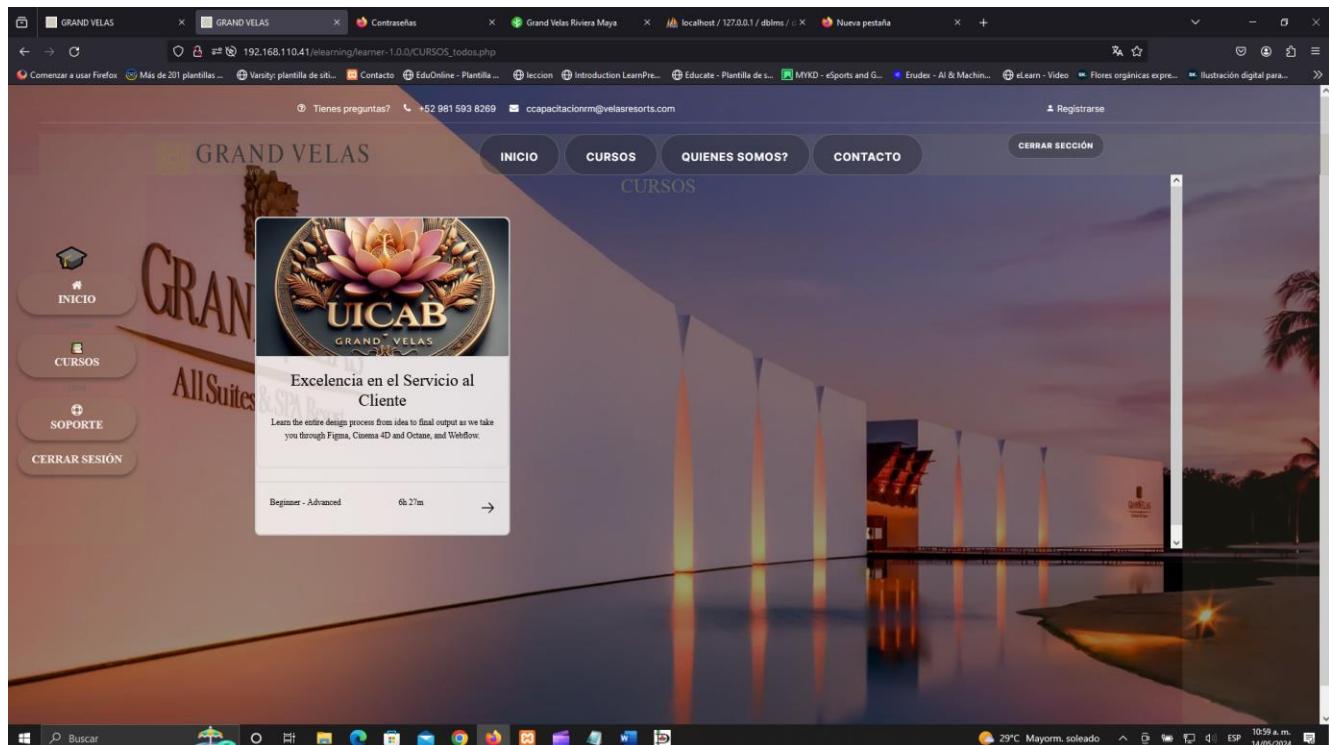


Ilustración 5. Descripción del curso:

The screenshot shows a course description page for 'Excelencia en el Servicio al Cliente'. The header features the Grand Velas logo and navigation links for INICIO, CURSOS, QUIENES SOMOS?, CONTACTO, and CERRAR SECCIÓN. On the left, there's a sidebar with icons for INICIO, CURSOS, SOPORTE, and CERRAR SESIÓN. The main content area displays the course title 'Excelencia en el Servicio al Cliente' and a brief introduction. Below this is a table titled 'TEMARIOS' (Table of Contents) with one item: 'Introducción a la importancia del servicio al cliente'. A large image of a beach chair with the Grand Velas logo is visible in the background. To the right, a sidebar shows a preview of another course titled 'UICAB GRAND VELAS'.

Ilustración 6. Reproductor de video:

The screenshot shows a video player interface for the course 'Excelencia en el Servicio al Cliente'. The video frame displays a large, ornate flower graphic. The sidebar on the right lists two topics: 'Tema 1' (Introducción a la importancia del servicio al cliente) and 'Tema 2' (Comunicar en diferentes escenarios). Each topic has a QR code and a brief description. The bottom of the screen shows the Grand Velas logo and 'Riviera Maya' text, along with a decorative banner featuring the resort's name in gold letters.

Descripción

¡Descubre los secretos de la excelencia en el servicio! Video introductorio al Módulo 1: Fundamentos de la Excelencia en el Servicio. En este video: Introducción a la importancia del servicio al cliente en Grand Velas, Definición de excelencia en el servicio, Los valores y estándares de lujo de Grand Velas. Acompáñanos en este viaje y descubre cómo convertirte en un experto en el arte de la excelencia en el servicio. ¡No te lo pierdas!

Tomar en cuenta

Más información relevante

Comentarios

Escribe tu comentario aquí...

UICABA4g
assas as sa a assa as a as as as a assa

UICABA4g
KLLLLLLLLL

Tema 1
Introducción a la importancia del servicio al cliente

Tema 2
Comunicar en diferentes escenarios
Comunicar en diferentes escenarios
Comunicar en diferentes escenarios
Como utilizar las habilidades para crear experiencias memorables

Ilustración 7. Editar perfil:

Editando Usuario

Nombre: UICABA4

Apellidos: Uicabdsd

Email: uicab.nahuatl.victor@cbta80.edu.mx

ACTUALIZAR INFORMACIÓN

Actualización correcta

Desplegable • **Usuario registrado**
• **Ver cursos**
• **Añadir cursos**
Permisos Administrador

Ilustración 8. Quienes somos:

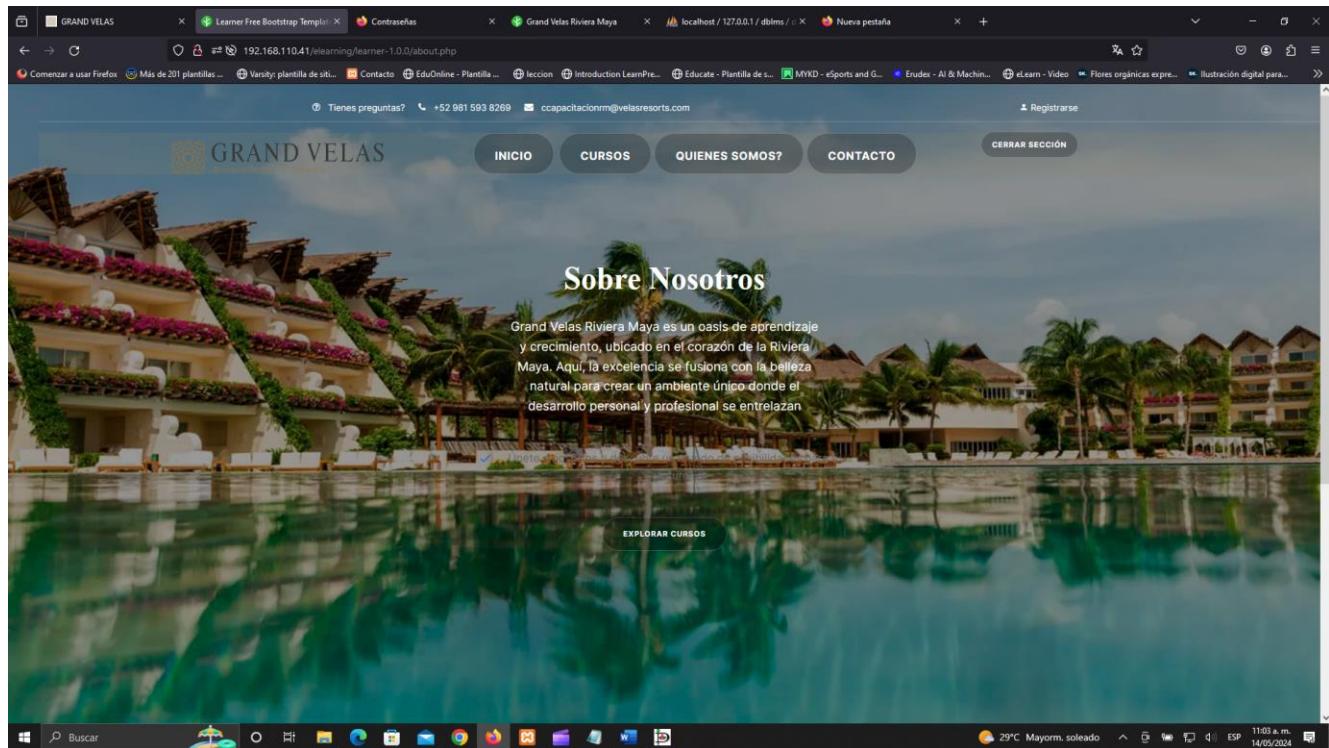


Ilustración 9. Contacto:

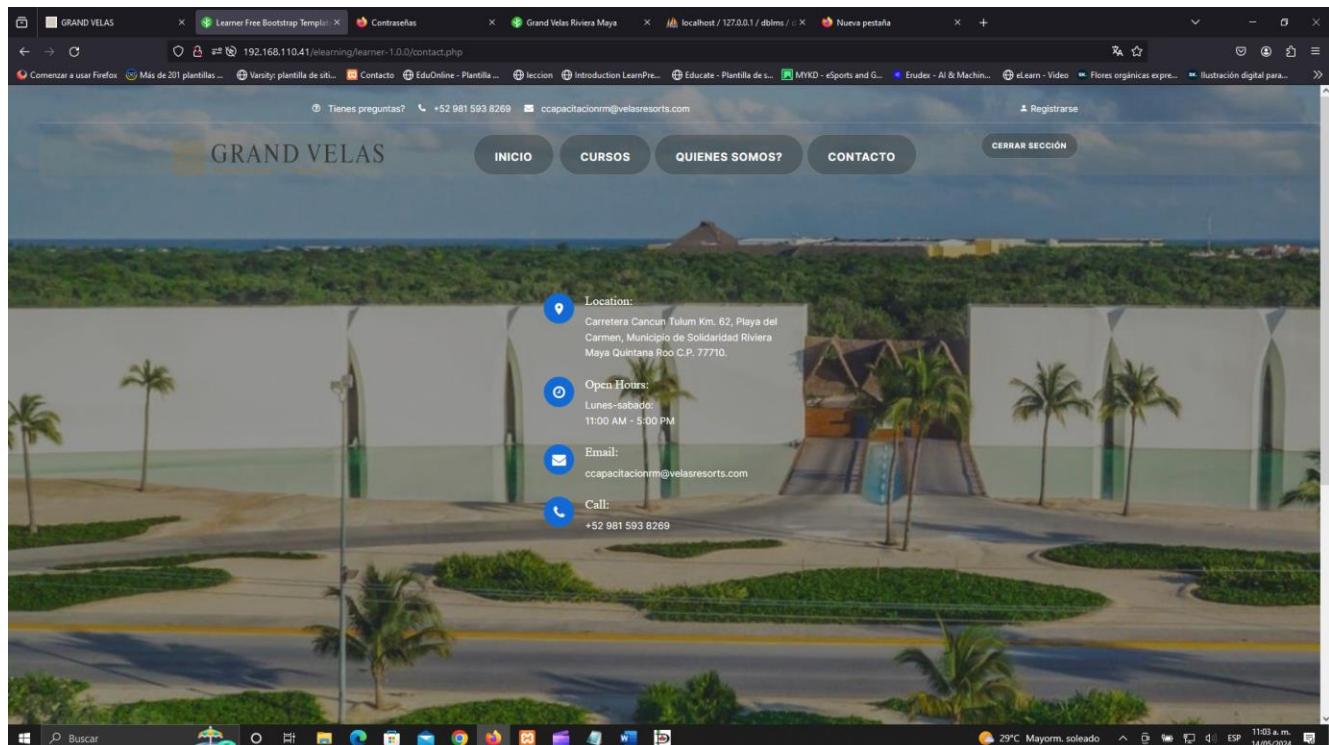


Ilustración 10. Categoría presentaciones y conviértete en instructor:

The screenshot shows a web browser window with multiple tabs open. The main content area displays a section titled "Categoría" with two items: "Presentaciones cursos" and "PLAYA LIMPIA". Below this, there is a section titled "Conviértete en Instructor" featuring a group photo of several people and some descriptive text.

Categoría

Presentaciones cursos

PLAYA LIMPIA

Conviértete en Instructor

¿Eres un experto en tu área y te apasiona enseñar? ¡Grand Velas Riviera Maya te invita a unirte a nuestro equipo de instructores! Comparte tus conocimientos y habilidades con otros miembros del equipo a través de nuestra plataforma de e-learning.

+52 981 593 8269

Ilustración 11. informativo:

The screenshot shows a web browser window displaying the Grand Velas Riviera Maya website. A large image of a resort building overlooking a pool and ocean serves as the background for an informative section titled "Educación para los líderes del mañana". The section contains descriptive text about the hotel's educational programs.

GRAND VELAS

INICIO CURSOS QUIENES SOMOS? CONTACTO CERRAR SECCIÓN

Educación para los líderes del mañana

En la paradisiaca Riviera Maya, donde el mar Caribe se encuentra con la exuberante selva, Grand Velas Riviera Maya se erige como un faro de conocimiento y desarrollo para los líderes del futuro. A través de programas de formación innovadores y un enfoque en la excelencia, cultivamos el talento de nuestros colaboradores, impulsándolos a alcanzar su máximo potencial y a liderar con pasión e integridad.

GRAND VELAS GRAND VELAS Contraseñas Grand Velas Riviera Maya localhost / 127.0.0.1 / dbms / Nueva pestaña

Comenzar a usar Firefox Más de 201 plantillas... Varsity: plantilla de siti... Contacto EduOnline - Plantilla ... lección Introduction LearnPre... Educate - Plantilla de ... MYKD - eSports and G... Erudex - AI & Machin... eLearn - Video Flores orgánicas expre... Ilustración digital para...

GRAND VELAS Riviera Maya | Mexico INICIO CURSOS QUIENES SOMOS? CONTACTO CERRAR SECCIÓN

Sobre Nosotros

Grand Velas Riviera Maya es un oasis de aprendizaje y crecimiento, ubicado en el corazón de la Riviera Maya. Aquí, la excelencia se fusiona con la belleza natural para crear un ambiente único donde el desarrollo personal y profesional se entrelazan.

✓ Únete a nosotros y descubre un mundo de posibilidades para tu futuro.



GRAND VELAS GRAND VELAS Contraseñas Grand Velas Riviera Maya localhost / 127.0.0.1 / dbms / Nueva pestaña

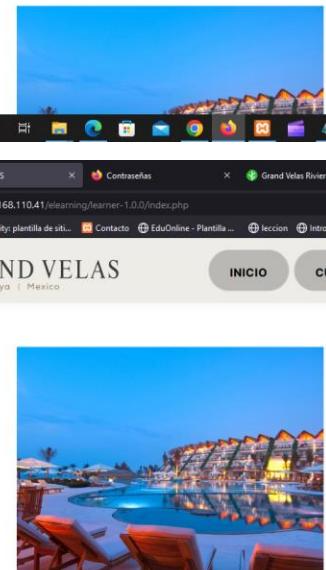
Comenzar a usar Firefox Más de 201 plantillas... Varsity: plantilla de siti... Contacto EduOnline - Plantilla ... lección Introduction LearnPre... Educate - Plantilla de ... MYKD - eSports and G... Erudex - AI & Machin... eLearn - Video Flores orgánicas expre... Ilustración digital para...

GRAND VELAS Riviera Maya | Mexico INICIO CURSOS QUIENES SOMOS? CONTACTO CERRAR SECCIÓN

¿Por qué elegirnos?

Grand Velas Riviera Maya se distingue por ofrecer una experiencia educativa excepcional, que va más allá de la formación tradicional.

Buenos profesores y Staffs



GRAND VELAS GRAND VELAS Contraseñas Grand Velas Riviera Maya localhost / 127.0.0.1 / dbms / Nueva pestaña

Comenzar a usar Firefox Más de 201 plantillas... Varsity: plantilla de siti... Contacto EduOnline - Plantilla ... lección Introduction LearnPre... Educate - Plantilla de ... MYKD - eSports and G... Erudex - AI & Machin... eLearn - Video Flores orgánicas expre... Ilustración digital para...

Ilustración 12. footer:

The screenshot shows a Firefox browser window with multiple tabs open. The main content area displays the footer of the Grand Velas Riviera Maya website. The footer features a navigation bar with links for INICIO, CURSOS, QUIENES SOMOS?, CONTACTO, and CERRAR SECCIÓN. Below this, a text block states: "profesores altamente calificados y programas de formación innovadores que te prepararán para enfrentar los retos del futuro." On the left, there's a section titled "Sobre Grand Velas Riviera Maya" with a brief description of the resort. In the center, there are links for Programas (Servicio al Cliente, Habilidades de Liderazgo, Desarrollo Personal, Idiomas, Cultura Local) and a Galería icon. On the right, there's a Contacto section with address details, a Google Maps link, phone number, and email. At the bottom, a copyright notice reads: "Copyright © 2024. Todos los derechos reservados. — Grand Velas Riviera Maya". The browser's taskbar at the bottom shows various pinned icons and the system tray on the right.

4. Página web (Administrador):

Ilustración 1. Usuarios registrados:

The screenshot shows the Grand Velas eLearning website interface. At the top, there is a navigation bar with links for Inicio, Desplegable, Nuestro Equipo, Noticias, Galería, Acerca de, and Contacto. A sidebar on the right contains a list of administrative tasks: 'Usuarios registrados', 'Ver cursos', 'Añadir cursos', and 'Permisos Administrador'. The main content area features a banner with the text 'La educación es la madre del liderazgo' and a button labeled 'EXPLORA CURSOS'. Below the banner, there are six user profiles displayed in a grid. Each profile includes a small thumbnail, the user's name, and their registration date. The profiles are: 3333sfesfeknjk (Fecha de registro: 2022-06-11), Administrador uicab (Fecha de registro: 2022-06-12), addss adadasdssa (Fecha de registro: 2022-08-22), manuel uicab nahsua (Fecha de registro: 2022-12-06), Victor Nahuat (Fecha de registro: 2023-02-16), and another unnamed user (Fecha de registro: 2022-06-11). The background of the page shows a scenic view of a resort at sunset.

Ilustración 2. Ver cursos:

The screenshot shows the Grand Velas eLearning website interface, specifically the course management section. The top navigation bar and sidebar are identical to Illustration 1. The main content area features a banner with the text 'La educación es la madre del liderazgo'. Below the banner, there is a summary of course statistics: 4 Cursos en curso, 3 Cursos pendientes, 5 Cursos resultados, and 10 Cursos completados. A table titled 'Lista de cursos' displays a list of courses. The columns include: #, Título, Categoría, Instructor, Precio, and Nombre de etiqueta. A filter button labeled 'FILTRAR' is located at the top of the table. An action menu is open over the first course entry, listing options: 'Ver Curso en el Frontend', 'Editar Este Curso', 'Sección y Lección', and 'Eliminar Curso'. The table data for the first course is: Nombre de la Categoría Ejemplo, Total de Secciones: 5, Total de Inscripc., Instructor: Nombre del Instructor Ejemplo. The bottom of the screen shows the Windows taskbar with various application icons.

Ilustración 3. Agregar videos:

The image consists of two side-by-side screenshots of a web-based course creation interface. Both screenshots show a header for 'GRAND VELAS' with a logo and a navigation bar with links like 'Inicio', 'Desplegable', 'Nuestro Equipo', 'Noticias', 'Galería', 'Acerca de', 'Contacto', 'CERRAR SECCIÓN', 'Edita perfil', and 'Registrarse'. A sidebar on the right contains links for 'Usuarios registrados', 'Ver cursos', 'Añadir cursos', and 'Permisos Administrador'.

The main form area has tabs for 'Básico', 'Requisitos', 'SEO', and 'Finalizar'. The 'Básico' tab is active, showing fields for 'Nombre' (name), 'Descripción del curso' (course description) containing the text 'La educación es la madre del liderazgo', 'Categoría' (category), 'Nivel' (level), and 'Selector de Temas' (topic selector). Below these fields is a checkbox labeled 'Compruebe si este curso es el mejor' (Check if this course is the best).

The second screenshot shows the same interface after a video has been added. A green notification bar at the top right says 'Congratulations! Welcom admin admin'. The 'Finalizar' (Finish) button is visible in the top right corner of the form area.

Ilustración 4. agregar categorías:

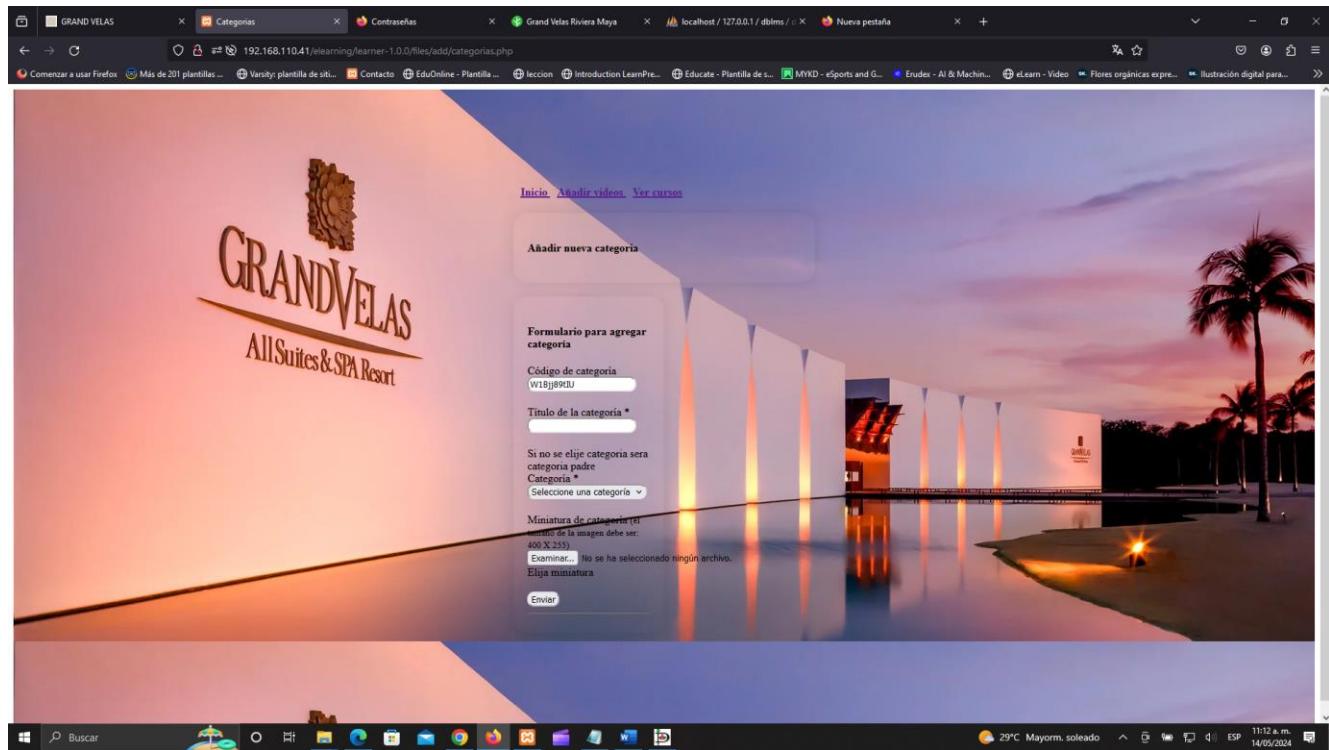


Ilustración 5. Agregar presentaciones

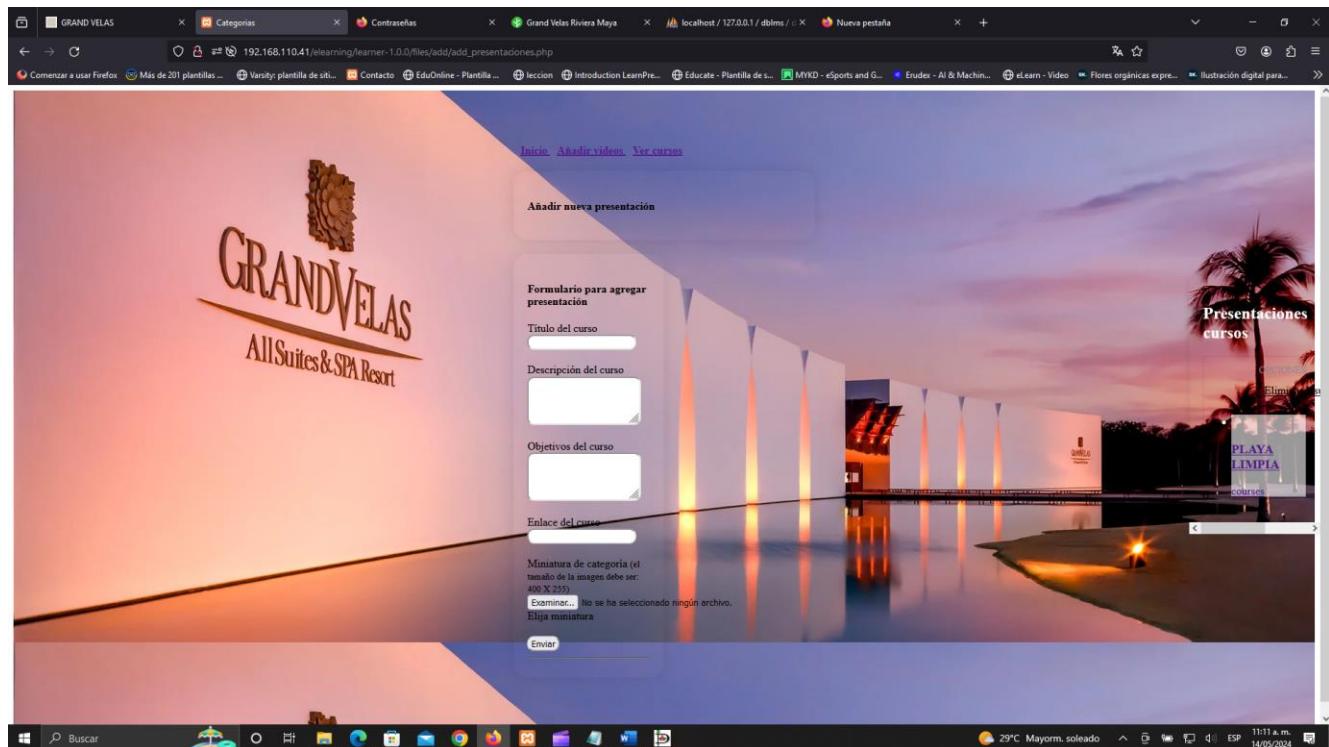


Ilustración 6 Editar el curso:

ID del curso: 73

TEMA 1

#	Título	Subcategoría	Descripción	Acciones
	Introducción a la importancia del servicio al cliente Instructor: Nombre del Instructor Ejemplo	Nombre Excelencia en el Servicio al Cliente		Ver Curso en el Frontend Editar Este Curso Eliminar Curso

TEMA 2

#	Título	Subcategoría	Descripción	Acciones
	Comunicación Efectiva Instructor: Nombre del Instructor Ejemplo	Nombre Excelencia en el Servicio al Cliente	descripcion	
	título vg 3 Instructor: Nombre del Instructor Ejemplo	Nombre Excelencia en el Servicio al Cliente	descripcion	

ID del curso: 73

TEMA 1

Formulario para agregar cursos

Básico Requisitos SEO Finalizar

Congratulations!
Welcom admin admin

Descripción Acciones

descripción

descripción

TEMA 2

Título

Comunicación Efectiva
Instructor: Nombre del Instructor Ejemplo

título vg 3
Instructor: Nombre del Instructor Ejemplo

Descripción Acciones

descripción

descripción



ID del curso: 73

localhost/learner-1.0.0 - copia (2)/files/add/update_courses2.php?id=73

Comenzar a usar Firefox Más de 201 plantillas ... Varsity: plantilla de sitio... Contacto EduOnline - Plantilla ... lección Introduction LearnPre... Educate - Plantilla de ... MYKD - esports and G... Erudex - AI & Machin... eLearn - Video Flores orgánicas expre... Ilustración digital para...

Document X 150% 100% 100%

Formulario para agregar cursos

Volver a la lista de cursos

Requisitos

Básico Requisitos SEO Finalizar

Conocimientos básicos de informática Acceso a internet Motivaci...

Descripción **Acciones**

descripción

Descripción **Acciones**

descripción

Descripción **Acciones**

descripción

TEMA 1

Título

Introducción a la importancia del Instructor: Nombre del Instructor Ejemplo

TEMA 2

Título

Comunicación Efectiva Instructor: Nombre del Instructor Ejemplo

titulo vg 3 Instructor: Nombre del Instructor Ejemplo

ID del curso: 73

localhost/learner-1.0.0 - copia (2)/files/add/update_courses2.php?id=32&requirements

localhost/learner-1.0.0 - copia (2)/files/add/update_courses2.php?id=73

Comenzar a usar Firefox Más de 201 plantillas ... Varsity: plantilla de sitio... Contacto EduOnline - Plantilla ... lección Introduction LearnPre... Educate - Plantilla de ... MYKD - esports and G... Erudex - AI & Machin... eLearn - Video Flores orgánicas expre... Ilustración digital para...

Document X 150% 100% 100%

Formulario para agregar cursos

Volver a la lista de cursos

SEO Básico Requisitos SEO Finalizar

Meta Keywords

Meta Description

Descripción **Acciones**

descripción

Descripción **Acciones**

descripción

Descripción **Acciones**

descripción

TEMA 1

Título

Introducción a la importancia del Instructor: Nombre del Instructor Ejemplo

TEMA 2

Título

Comunicación Efectiva Instructor: Nombre del Instructor Ejemplo

titulo vg 3 Instructor: Nombre del Instructor Ejemplo

GRAND VELAS Document Contraseñas Grand Velas Riviera Maya localhost / 127.0.0.1 / dblms / Nueva pestaña

Comenzar a usar Firefox Más de 201 plantillas ... Varsity: plantilla de sitio... Contactos EduOnline - Plantilla ... Lección Introduction LearnPre... Educate - Plantilla de ... MYKO - eSports and G... Erudex - AI & Machin... eLearn - Video Flores orgánicas expre... Ilustración digital para...

Formulario para agregar cursos

Básico Requisitos SEO Finalizar

Congratulations! Volver a la lista de cursos Welcome admin admin

Sube un video!

Estas a un click de subir un video

VIDEO ACTUAL: Video sin título - Hecho con Clipchamp (1).mp4

Examinar... No se ha seleccionado ningún archivo.

Miniatura de categoría (Solo admite formato imagen)

660d8d1f9225b.jpeg Browse

Guardar

Descripción Acciones

descripcion

descripcion

Descripción Acciones

descripcion

descripcion



5. Base de datos:

Ilustración 5.1 Diagrama base de datos:

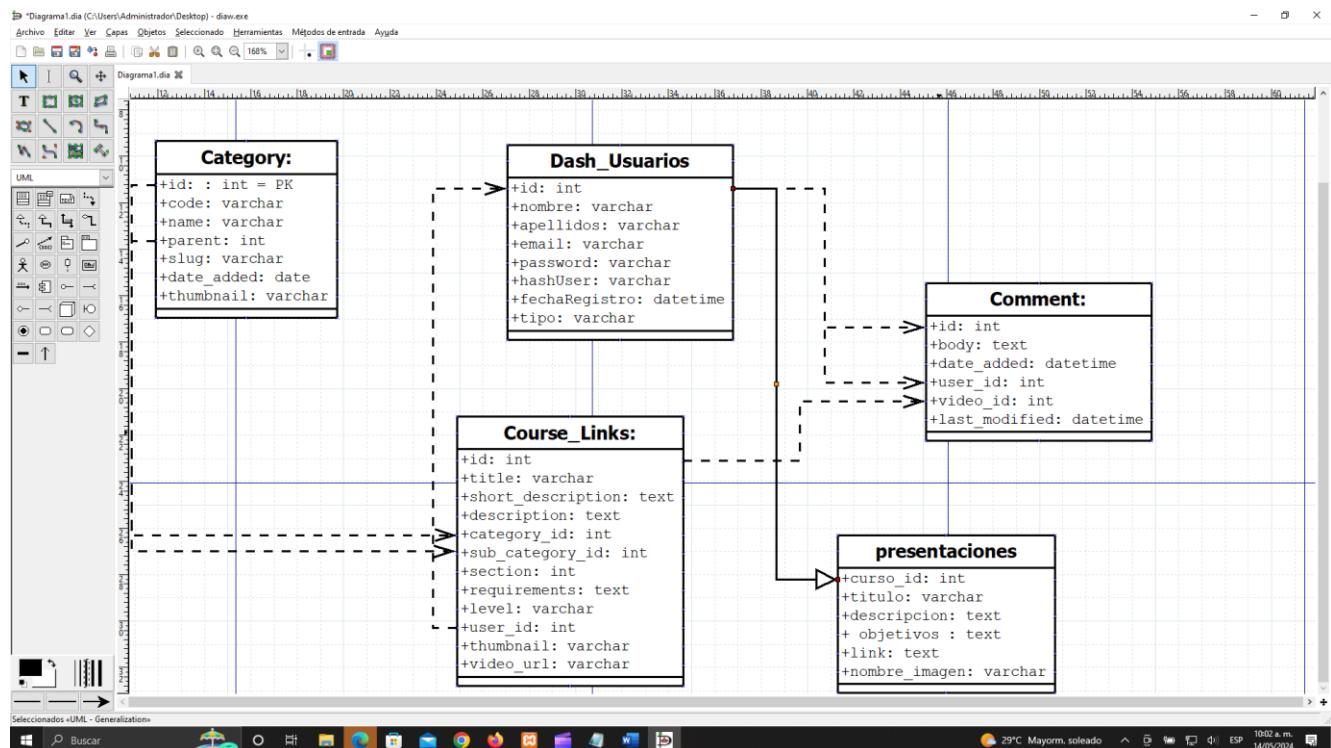
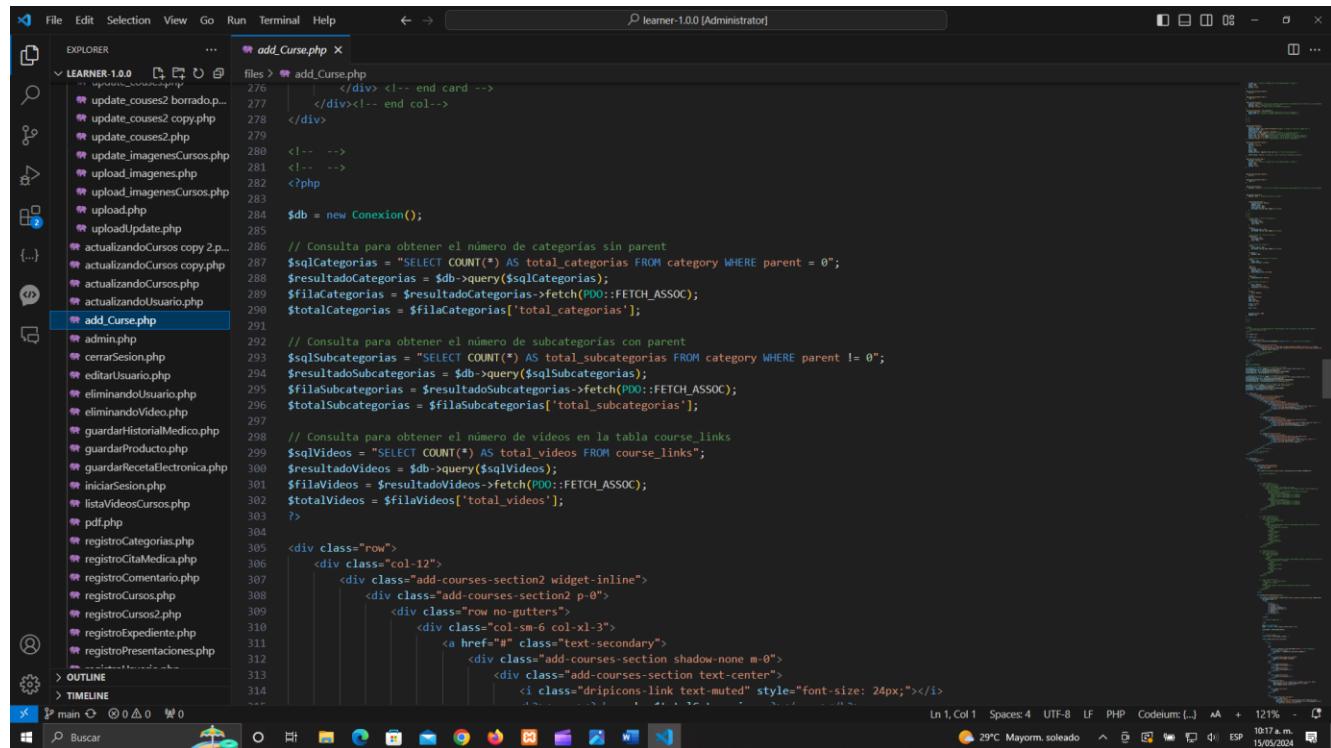


Ilustración 5.2 Base de datos:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
category	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_unicode_ci	16.0 KB	-
ci_sessions	Examinar Estructura Buscar Insertar Vaciar Eliminar	101	InnoDB	utf8_unicode_ci	48.0 KB	-
comment	Examinar Estructura Buscar Insertar Vaciar Eliminar	12	InnoDB	utf8_unicode_ci	16.0 KB	-
course	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_unicode_ci	16.0 KB	-
course_links	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_unicode_ci	16.0 KB	-
currency	Examinar Estructura Buscar Insertar Vaciar Eliminar	112	InnoDB	utf8_general_ci	16.0 KB	-
dash_usuarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	latin1_swedish_ci	16.0 KB	-
enrol	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_unicode_ci	16.0 KB	-
frontend_settings	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_unicode_ci	16.0 KB	-
language	Examinar Estructura Buscar Insertar Vaciar Eliminar	592	MyISAM	utf8_unicode_ci	24.8 KB	-
lesson	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8_unicode_ci	16.0 KB	-
message	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	MyISAM	latin1_swedish_ci	1.0 KB	-

Ilustración 5.3 añadir curso:

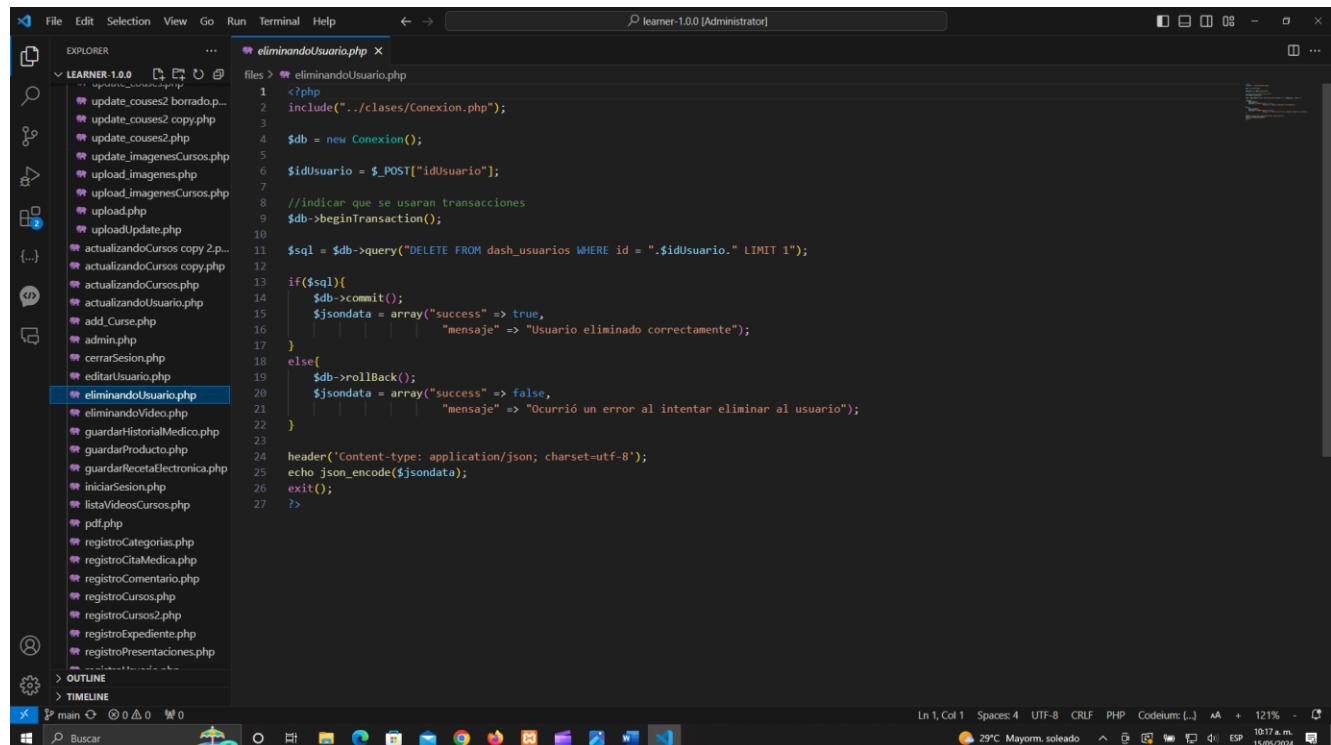


The screenshot shows a Windows desktop environment with a code editor window titled "learner-1.0.0 [Administrator]". The code editor displays the file "add_Curse.php". The code is written in PHP and includes SQL queries to count categories and subcategories, and to fetch video links. The code editor interface includes a sidebar for "EXPLORER" showing other files in the project, and a bottom status bar showing system information like temperature and date.

```
File Edit Selection View Go Run Terminal Help ← → learner-1.0.0 [Administrator]
EXPLORER ... add_Curse.php
LEARNER-1.0.0 ...
update_couses2_borrado.php
update_couses2_copy.php
update_couses2.php
update_imagenesCursos.php
upload_imagenes.php
upload_imagenesCursos.php
upload.php
uploadUpdate.php
actualizandoCursos copy 2.php
actualizandoCursos copy.php
actualizandoCursos.php
actualizandoUsuario.php
add_Curse.php
admin.php
cerrarSesion.php
editarUsuario.php
eliminandoUsuario.php
eliminandoVideo.php
guardarHistorialMedico.php
guardarProducto.php
guardarRecetaElectronica.php
iniciarSesion.php
listaVideosCursos.php
pdf.php
registroCategorias.php
registroCitaMedica.php
registroComentario.php
registroCursos.php
registroCursos2.php
registroExpediente.php
registroPresentaciones.php
...
OUTLINE
TIMELINE
main 0 0 0 0 0 0
Buscar
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP Codeium: (...) AA + 121% -
29°C Mayorm, soleado 10:17 a.m. 15/05/2024
```

```
files > add_Curse.php
276     </div> <!-- end card -->
277     </div><!-- end col -->
278   </div>
279   <!-- -->
280   <!-- -->
281 <?php
282
283 $db = new Conexion();
284
285 // Consulta para obtener el n mero de categorias sin parent
286 $sqlCategorias = "SELECT COUNT(*) AS total_categorias FROM category WHERE parent = 0";
287 $resultadoCategorias = $db->query($sqlCategorias);
288 $filaCategorias = $resultadoCategorias->fetch(PDO::FETCH_ASSOC);
289 $totalCategorias = $filaCategorias['total_categorias'];
290
291 // Consulta para obtener el n mero de subcategorias con parent
292 $sqlSubcategorias = "SELECT COUNT(*) AS total_subcategorias FROM category WHERE parent != 0";
293 $resultadoSubcategorias = $db->query($sqlSubcategorias);
294 $filaSubcategorias = $resultadoSubcategorias->fetch(PDO::FETCH_ASSOC);
295 $totalSubcategorias = $filaSubcategorias['total_subcategorias'];
296
297 // Consulta para obtener el n mero de videos en la tabla course_links
298 $sqlVideos = "SELECT COUNT(*) AS total_videos FROM course_links";
299 $resultadoVideos = $db->query($sqlVideos);
300 $filaVideos = $resultadoVideos->fetch(PDO::FETCH_ASSOC);
301 $totalVideos = $filaVideos['total_videos'];
302
303 >
304
305 <div class="row">
306     <div class="col-12">
307         <div class="add-courses-section2 widget-inline">
308             <div class="add-courses-section2 p-0">
309                 <div class="row no-gutters">
310                     <div class="col-sm-6 col-xl-3">
311                         <a href="#" class="text-secondary">
312                             <div class="add-courses-section shadow-none m-0">
313                                 <i class="dripicons-link text-muted" style="font-size: 24px;"></i>
```

Ilustraci n 5.4 eliminar usuario:



The screenshot shows a Windows desktop environment with a code editor window titled "learner-1.0.0 [Administrator]". The code editor displays the file "eliminandoUsuario.php". The code is written in PHP and uses MySQL queries to delete a user from a database table. It includes JSON encoding for the response. The code editor interface includes a sidebar for "EXPLORER" showing other files in the project, and a bottom status bar showing system information like temperature and date.

```
File Edit Selection View Go Run Terminal Help ← → learner-1.0.0 [Administrator]
EXPLORER ... eliminandoUsuario.php
LEARNER-1.0.0 ...
update_couses2_borrado.php
update_couses2_copy.php
update_couses2.php
update_imagenesCursos.php
upload_imagenes.php
upload_imagenesCursos.php
upload.php
uploadUpdate.php
actualizandoCursos copy 2.php
actualizandoCursos copy.php
actualizandoCursos.php
actualizandoUsuario.php
eliminandoUsuario.php
admin.php
cerrarSesion.php
editarUsuario.php
...
eliminandoVideo.php
guardarHistorialMedico.php
guardarProducto.php
guardarRecetaElectronica.php
iniciarSesion.php
listaVideosCursos.php
pdf.php
registroCategorias.php
registroCitaMedica.php
registroComentario.php
registroCursos.php
registroCursos2.php
registroExpediente.php
registroPresentaciones.php
...
OUTLINE
TIMELINE
main 0 0 0 0 0 0
Buscar
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP Codeium: (...) AA + 121% -
29°C Mayorm, soleado 10:17 a.m. 15/05/2024
```

```
files > eliminandoUsuario.php
1 <?php
2 include("../clases/Conexion.php");
3
4 $db = new Conexion();
5
6 $idUsuario = $_POST["idUsuario"];
7
8 //indicar que se usaran transacciones
9 $db->beginTransaction();
10
11 $sql = $db->query("DELETE FROM dash_usuarios WHERE id = ".$idUsuario." LIMIT 1");
12
13 if($sql){
14     $db->commit();
15     $jsondata = array("success" => true,
16                      "mensaje" => "Usuario eliminado correctamente");
17 }
18 else{
19     $db->rollBack();
20     $jsondata = array("success" => false,
21                      "mensaje" => "Ocurri  un error al intentar eliminar al usuario");
22 }
23
24 header('Content-type: application/json; charset=utf-8');
25 echo json_encode($jsondata);
26 exit();
27 >
```

Ilustración 5.5 eliminar video:

The screenshot shows a Windows desktop environment with a code editor window open. The title bar reads "learner-1.0.0 [Administrator]". The code editor displays a file named "eliminandoVideo.php" containing the following PHP code:

```
<?php
include("../clases/Conexion.php");
$db = new Conexion();
$idUsuario3 = $_POST["idUsuario3"];
//indicar que se usaran transacciones
$db->beginTransaction();
$sql = $db->query("DELETE FROM course_links WHERE id = ".$idUsuario3." LIMIT 1");
if($sql){
    $db->commit();
    $jsondata = array("success" => true,
                      "mensaje" => "Usuario eliminado correctamente");
} else{
    $db->rollBack();
    $jsondata = array("success" => false,
                      "mensaje" => "Ocurrió un error al intentar eliminar al usuario");
}
header('Content-type: application/json; charset=utf-8');
echo json_encode($jsondata);
exit();
?>
```

The code performs a DELETE query on the "course_links" table where the "id" column matches the value of the variable \$idUsuario3. It uses a transaction to either commit the change if successful or roll back if there is an error. The result is returned as a JSON object.

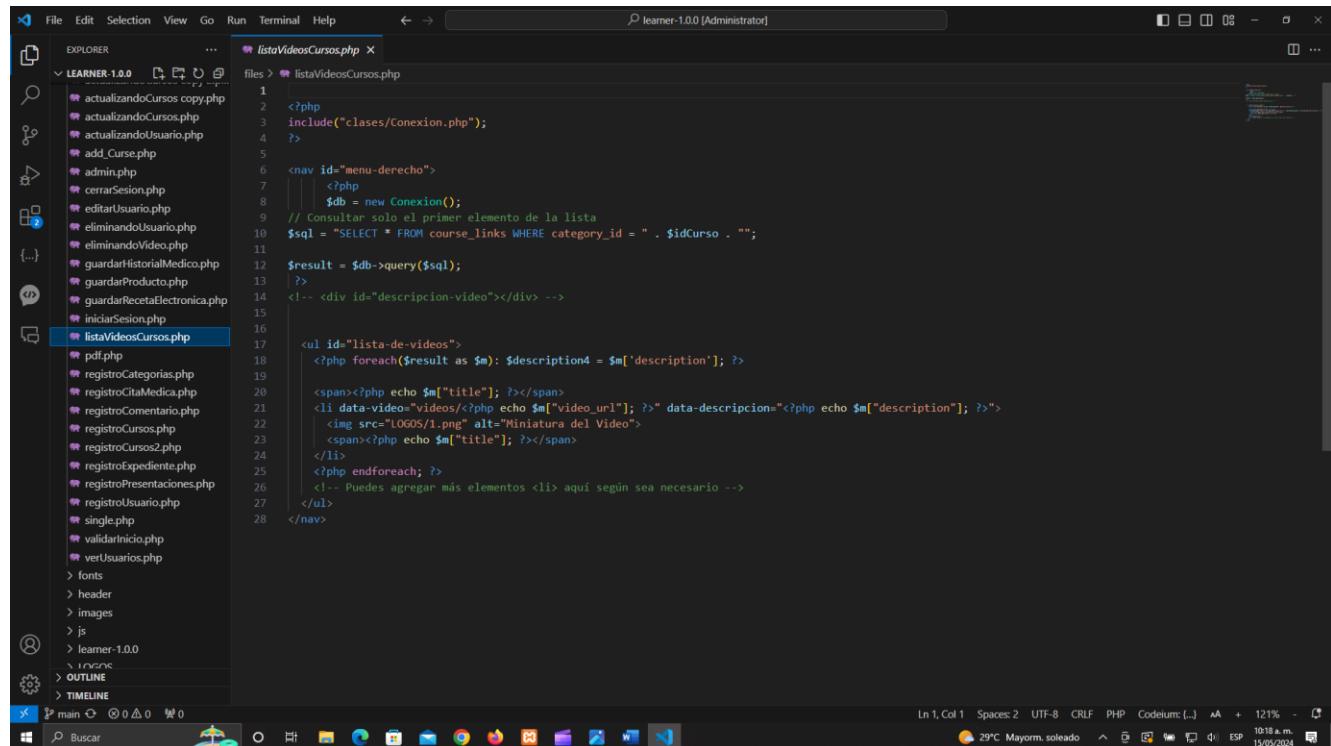
Ilustración 5.6 eliminando video:

The screenshot shows a Windows desktop environment with a code editor window open. The title bar reads "learner-1.0.0 [Administrator]". The code editor displays a file named "iniciarSesion.php" containing the following PHP code:

```
<?php
/include("clases/conexion.php");
$db = new Conexion();
$nombre_sesion = "inventario";
$regreso = "login.html";
if(isset($_COOKIE["cDash"])){
    $hashUser = $_COOKIE["cDash"];
    $sql = $db->prepare("SELECT *
                        FROM dash_usuarios
                        WHERE hashUser = :hashUser LIMIT 1");
    $sql->bindParam(":hashUser", $hashUser, PDO::PARAM_STR);
    $sql->execute();
    if(!$sql){
        setcookie("err", "1", time() + (60), "/");
        session_destroy();
        Header ("Location: ".$regreso);
        exit;
    } else{
        if ($sql->rowCount() > 0){
            $usuario_datos = $sql->fetch();
            unset($sql);
            if($hashUser != $usuario_datos["hashUser"]){
                setcookie("err", "2", time() + (60), "/");
                Header ("Location: ".$regreso);
                exit;
            }
            unset($hashUser);
            if(isset($_COOKIE["cDashExpiration"])){
?>
```

This script handles user login. It checks if a cookie named "cDash" exists. If it does, it retrieves the hashed user ID from the cookie and performs a database query to find the user record. If the user is found, it checks if the stored hash matches the one in the cookie. If they don't, it sets an error cookie and redirects the user. If the user is found and the hashes match, it removes the database connection object and sets a session variable. Finally, it checks if a cookie named "cDashExpiration" exists and if so, it sets another error cookie and redirects the user.

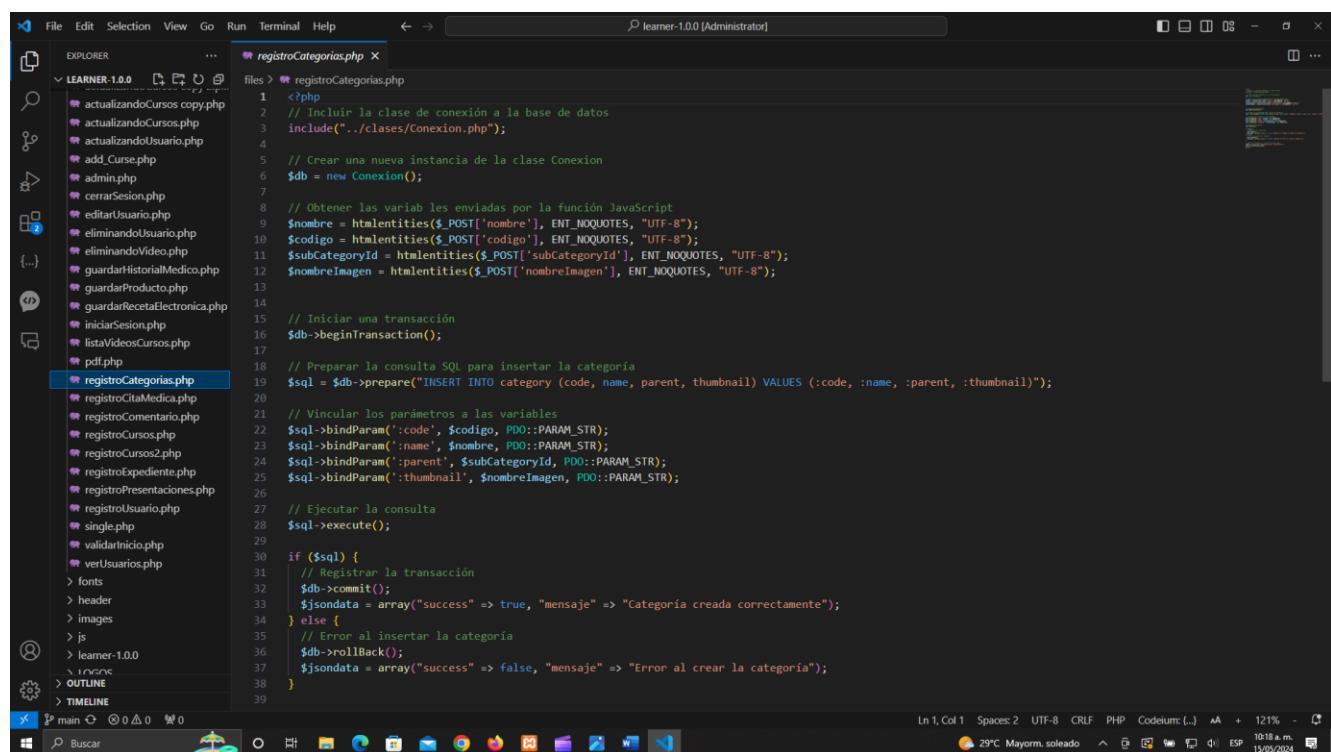
Ilustración 5.7 consulta lista de videos:



```
<?php
include("clases/conexion.php");
// Consultar solo el primer elemento de la lista
$sql = "SELECT * FROM course_links WHERE category_id = " . $idCurso . "";
$result = $db->query($sql);
<!-- <div id="descripcion-video"></div> -->
<ul id="lista-de-videos">
<php foreach($result as $m): $description4 = $m['description']; ?>
<span><php echo $m["title"]; ?></span>
<li data-video="videos/><php echo $m["video_url"]; ?>" data-descripcion=<php echo $m["description"]; ?>>

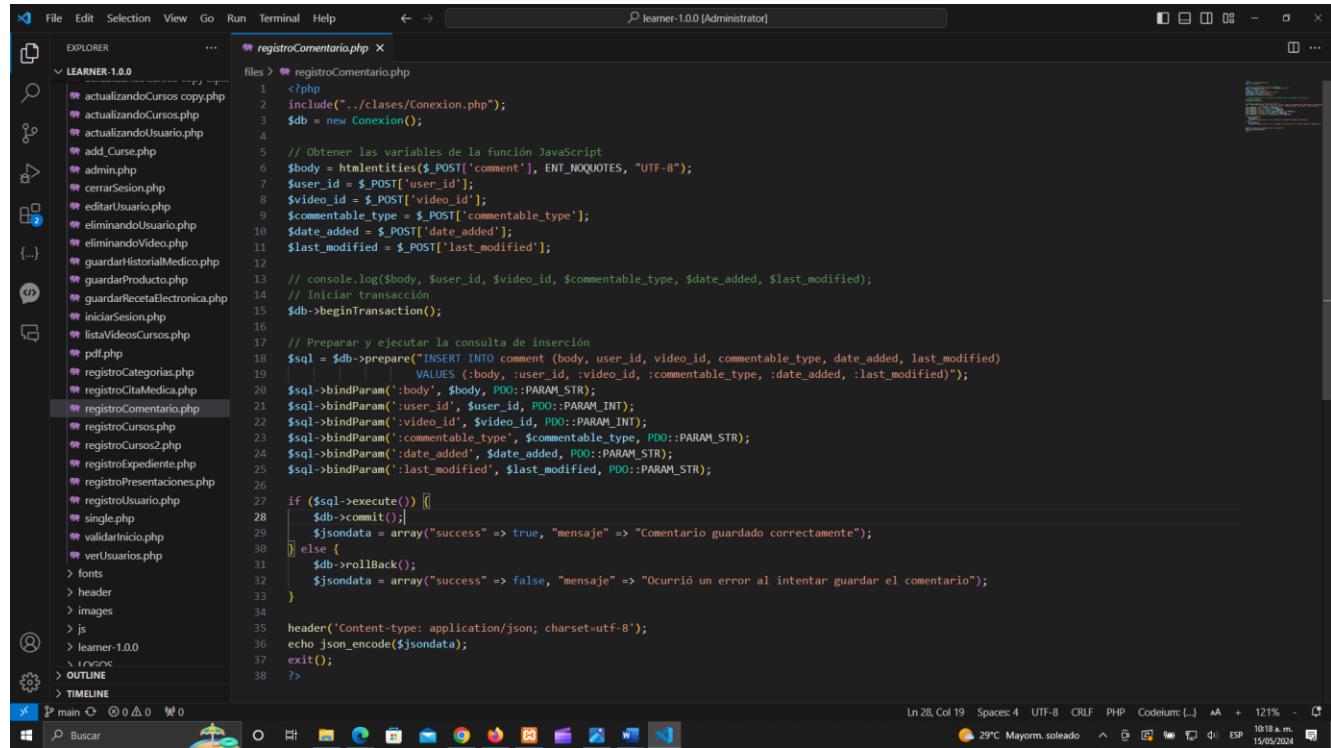
<span><php echo $m["title"]; ?></span>
</li>
<php endforeach; ?>
<!-- Puedes agregar más elementos <li> aquí según sea necesario -->
</ul>
</nav>
```

Ilustración 5.8 registrar una categoría:



```
<?php
// Incluir la clase de conexión a la base de datos
include("../clases/conexion.php");
// Crear una nueva instancia de la clase Conexion
$db = new Conexion();
// Obtener las variables enviadas por la función JavaScript
$nombre = htmlentities($_POST['nombre'], ENT_NOQUOTES, "UTF-8");
$codigo = htmlentities($_POST['codigo'], ENT_NOQUOTES, "UTF-8");
$subCategoryId = htmlentities($_POST['subCategoryId'], ENT_NOQUOTES, "UTF-8");
$nombreImagen = htmlentities($_POST['nombreImagen'], ENT_NOQUOTES, "UTF-8");
// Iniciar una transacción
$db->beginTransaction();
// Preparar la consulta SQL para insertar la categoría
$stmt = $db->prepare("INSERT INTO category (code, name, parent, thumbnail) VALUES (:code, :name, :parent, :thumbnail)");
// Vincular los parámetros a las variables
$stmt->bindParam(':code', $codigo, PDO::PARAM_STR);
$stmt->bindParam(':name', $nombre, PDO::PARAM_STR);
$stmt->bindParam(':parent', $subCategoryId, PDO::PARAM_STR);
$stmt->bindParam(':thumbnail', $nombreImagen, PDO::PARAM_STR);
// Ejecutar la consulta
$stmt->execute();
if ($stmt) {
    // Registrar la transacción
    $db->commit();
    $jsondata = array("success" => true, "mensaje" => "Categoría creada correctamente");
} else {
    // Error al insertar la categoría
    $db->rollBack();
    $jsondata = array("success" => false, "mensaje" => "Error al crear la categoría");
}
}
```

Ilustración 5.9 registrar comentario:



```
<?php
include("../clases/Conexion.php");
$db = new Conexion();

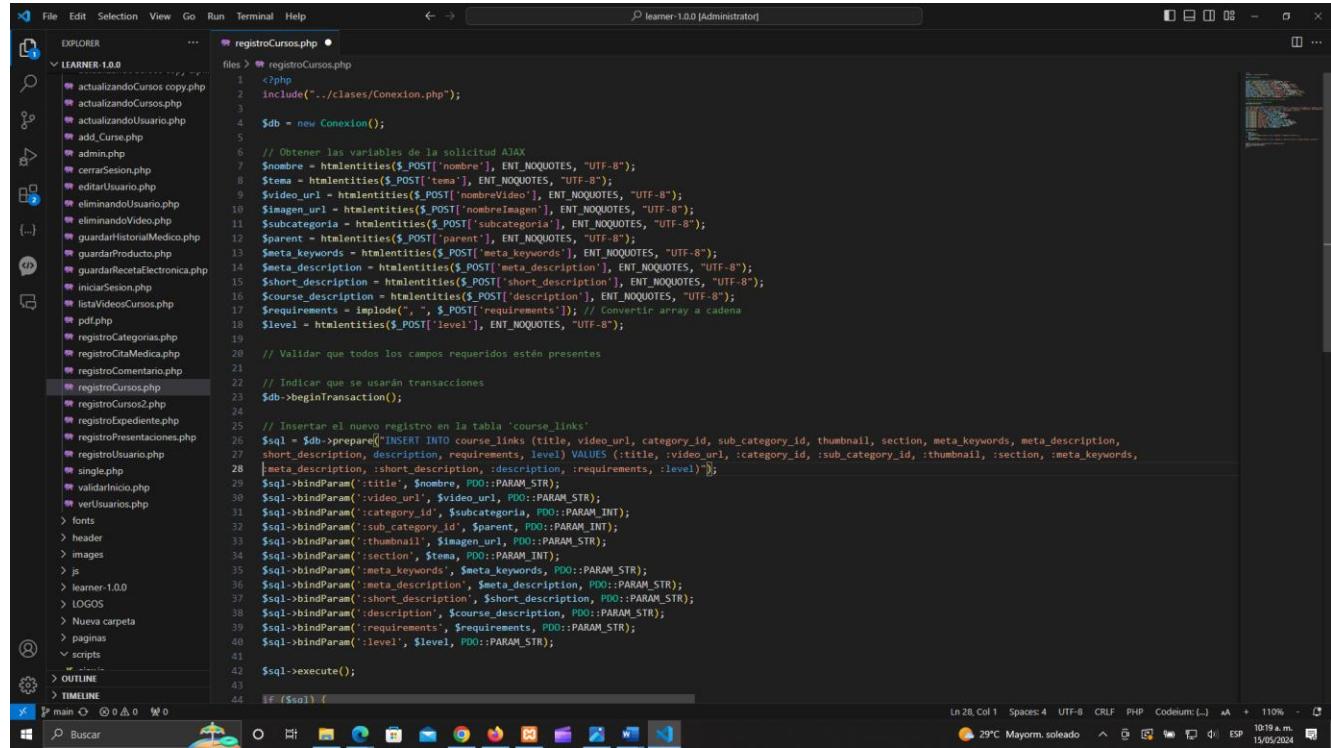
// Obtener las variables de la función JavaScript
$body = htmlentities($_POST['comment'], ENT_NOQUOTES, "UTF-8");
$user_id = $_POST['user_id'];
$video_id = $_POST['video_id'];
$commentable_type = $_POST['commentable_type'];
$date_added = $_POST['date_added'];
$last_modified = $_POST['last_modified'];

// Preparar y ejecutar la consulta de inserción
$sql = $db->prepare("INSERT INTO comment (body, user_id, video_id, commentable_type, date_added, last_modified)
VALUES (:body, :user_id, :video_id, :commentable_type, :date_added, :last_modified)");
$sql->bindParam(':body', $body, PDO::PARAM_STR);
$sql->bindParam(':user_id', $user_id, PDO::PARAM_INT);
$sql->bindParam(':video_id', $video_id, PDO::PARAM_INT);
$sql->bindParam(':commentable_type', $commentable_type, PDO::PARAM_STR);
$sql->bindParam(':date_added', $date_added, PDO::PARAM_STR);
$sql->bindParam(':last_modified', $last_modified, PDO::PARAM_STR);

if ($sql->execute()) {
    $db->commit();
    $jsondata = array("success" => true, "mensaje" => "Comentario guardado correctamente");
} else {
    $db->rollBack();
    $jsondata = array("success" => false, "mensaje" => "Ocurrió un error al intentar guardar el comentario");
}

header('Content-type: application/json; charset=utf-8');
echo json_encode($jsondata);
exit();
?>
```

Ilustración 5.10 registrar curso:



```
<?php
include("../clases/Conexion.php");
$db = new Conexion();

// Obtener las variables de la solicitud AJAX
$nombre = htmlentities($_POST['nombre'], ENT_NOQUOTES, "UTF-8");
$tema = htmlentities($_POST['tema'], ENT_NOQUOTES, "UTF-8");
$video_url = htmlentities($_POST['nombreVideo'], ENT_NOQUOTES, "UTF-8");
$imagen_url = htmlentities($_POST['nombreImagen'], ENT_NOQUOTES, "UTF-8");
$subcategoria = htmlentities($_POST['subcategoria'], ENT_NOQUOTES, "UTF-8");
$parent = htmlentities($_POST['parent'], ENT_NOQUOTES, "UTF-8");
$meta_keywords = htmlentities($_POST['meta_keywords'], ENT_NOQUOTES, "UTF-8");
$meta_description = htmlentities($_POST['meta_description'], ENT_NOQUOTES, "UTF-8");
$short_description = htmlentities($_POST['short_description'], ENT_NOQUOTES, "UTF-8");
$course_description = htmlentities($_POST['description'], ENT_NOQUOTES, "UTF-8");
$requirements = implode(", ", $_POST['requirements']); // Convertir array a cadena
$level = htmlentities($_POST['level'], ENT_NOQUOTES, "UTF-8");

// Validar que todos los campos requeridos estén presentes
// Indicar que se usarán transacciones
$db->beginTransaction();

// Insertar el nuevo registro en la tabla 'course_links'
$sql = $db->prepare("INSERT INTO course_links (title, video_url, category_id, sub_category_id, thumbnail, section, meta_keywords, meta_description,
short_description, description, requirements, level) VALUES (:title, :video_url, :category_id, :sub_category_id, :thumbnail, :section, :meta_keywords,
:meta_description, :short_description, :description, :requirements, :level)");
$sql->bindParam(':title', $nombre, PDO::PARAM_STR);
$sql->bindParam(':video_url', $video_url, PDO::PARAM_STR);
$sql->bindParam(':category_id', $parent, PDO::PARAM_INT);
$sql->bindParam(':sub_category_id', $subcategory, PDO::PARAM_INT);
$sql->bindParam(':thumbnail', $imagen_url, PDO::PARAM_STR);
$sql->bindParam(':section', $tema, PDO::PARAM_INT);
$sql->bindParam(':meta_keywords', $meta_keywords, PDO::PARAM_STR);
$sql->bindParam(':meta_description', $meta_description, PDO::PARAM_STR);
$sql->bindParam(':short_description', $short_description, PDO::PARAM_STR);
$sql->bindParam(':description', $course_description, PDO::PARAM_STR);
$sql->bindParam(':requirements', $requirements, PDO::PARAM_STR);
$sql->bindParam(':level', $level, PDO::PARAM_STR);

if ($sql->execute()) {
```

Ilustración 5.11 registrar usuario:

The screenshot shows a code editor with the file `registroUsuario.php` open. The code handles user registration by hashing the password, preparing an SQL insert statement, and committing the transaction if successful. It also checks if the email already exists in the database.

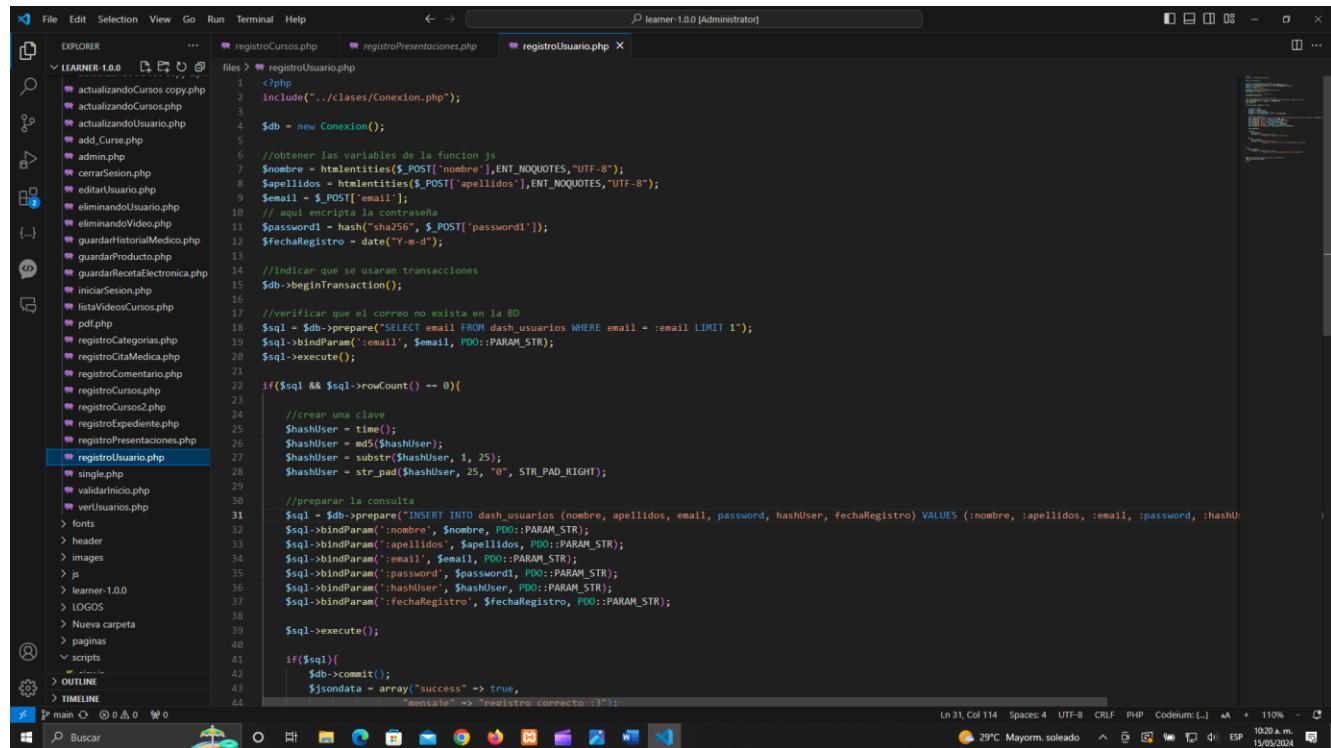
```
<?php
include("../clases/Conexion.php");
$db = new Conexion();
//obtener las variables de la funcion js
$nombre = htmlentities($_POST['nombre'], ENT_NOQUOTES, "UTF-8");
$apellidos = htmlentities($_POST['apellidos'], ENT_NOQUOTES, "UTF-8");
$email = $_POST['email'];
// aqui encripta la contraseña
$password1 = hash("sha256", $_POST['password1']);
$fechaRegistro = date("Y-m-d");
//indicar que se usaran transacciones
$db->beginTransaction();
//verificar que el correo no exista en la BD
$sql = $db->prepare("SELECT email FROM dash_usuarios WHERE email = :email LIMIT 1");
$sql->bindParam(':email', $email, PDO::PARAM_STR);
$sql->execute();
if($sql && $sql->rowCount() == 0){
    //crear una clave
    $hashUser = time();
    $hashUser = md5($hashUser);
    $hashUser = substr($hashUser, 1, 25);
    $hashUser = str_pad($hashUser, 25, "0", STR_PAD_RIGHT);
    //preparar la consulta
    $sql = $db->prepare("INSERT INTO dash_usuarios (nombre, apellidos, email, password, hashUser, fechaRegistro)
VALUES (:nombre, :apellidos, :email, :password, :hashUser, :fechaRegistro)");
    $sql->bindParam(":nombre", $nombre, PDO::PARAM_STR);
    $sql->bindParam(":apellidos", $apellidos, PDO::PARAM_STR);
    $sql->bindParam(":email", $email, PDO::PARAM_STR);
    $sql->bindParam(":password", $password1, PDO::PARAM_STR);
    $sql->bindParam(":hashUser", $hashUser, PDO::PARAM_STR);
    $sql->bindParam(":fechaRegistro", $fechaRegistro, PDO::PARAM_STR);
    $sql->execute();
    if($sql){
        $db->commit();
        $isodata = array("success" => true,
                        "mensaje" => "El usuario se ha registrado correctamente");
    }
}
else{
    $isodata = array("success" => false, "mensaje" => "Este correo ya existe");
}
echo json_encode($isodata);
```

Ilustración 5.12 registrar presentaciones:

The screenshot shows a code editor with the file `registroPresentaciones.php` open. The code inserts a new presentation into the database using prepared statements and transactions. It handles errors and provides feedback messages.

```
<?php
// Incluir la clase de conexión a la base de datos
include("../clases/Conexion.php");
$db = new Conexion();
// Crear una nueva instancia de la clase Conexion
// Obtener los valores enviados por la función JavaScript
$titulo = htmlentities($_POST['titulo'], ENT_QUOTES, "UTF-8");
$descripcion = htmlentities($_POST['descripcion'], ENT_QUOTES, "UTF-8");
$objetivos = htmlentities($_POST['objetivos'], ENT_QUOTES, "UTF-8");
$link = htmlentities($_POST['link'], ENT_QUOTES, "UTF-8");
$nombreImagen = htmlentities($_POST['nombreImagen'], ENT_QUOTES, "UTF-8");
// Iniciar una transacción
$db->beginTransaction();
try {
    // Preparar la consulta SQL para insertar el curso
    $sql = $db->prepare("INSERT INTO presentaciones (titulo, descripcion, objetivos, link, nombre_imagen) VALUES (:titulo, :descripcion, :objetivos, :link, :nombreImagen)");
    $sql->bindParam(":titulo", $titulo, PDO::PARAM_STR);
    $sql->bindParam(":descripcion", $descripcion, PDO::PARAM_STR);
    $sql->bindParam(":objetivos", $objetivos, PDO::PARAM_STR);
    $sql->bindParam(":link", $link, PDO::PARAM_STR);
    $sql->bindParam(":nombreImagen", $nombreImagen, PDO::PARAM_STR);
    // Ejecutar la consulta
    $sql->execute();
    // Verificar si se insertó correctamente
    if ($sql) {
        // Confirmar la transacción
        $db->commit();
        $isodata = array("success" => true, "mensaje" => "La presentación se ha agregado correctamente");
    } else {
        // Error al insertar el curso
        $db->rollBack();
        $isodata = array("success" => false, "mensaje" => "Error al agregar el curso");
    }
} catch (PDOException $e) {
    // Error en la transacción
    $db->rollBack();
}
echo json_encode($isodata);
```

Ilustración 5.13 registrar usuario:

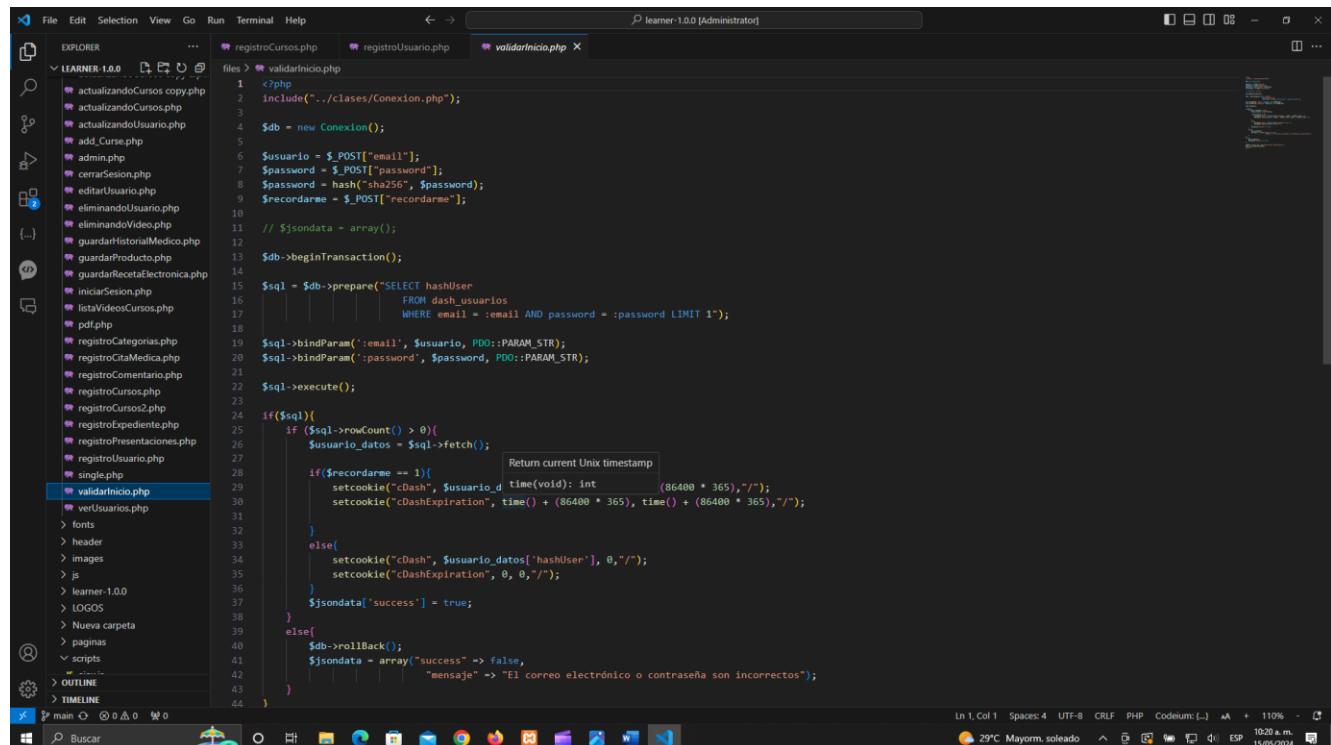


```
<?php
include("../clases/Conexion.php");
$db = new Conexion();
//obtener las variables de la funcion js
$nombre = htmlentities($_POST['nombre'],ENT_NOQUOTES,"UTF-8");
$apellidos = htmlentities($_POST['apellidos'],ENT_NOQUOTES,"UTF-8");
$email = $_POST['email'];
//se encripta la contraseña
$password1 = hash("sha256", $_POST['password1']);
$fechaRegistro = date("Y-m-d");
//indicar que se usaran transacciones
$db->beginTransaction();
//verifica que el correo no exista en la BD
$sql=>prepare("SELECT email FROM dash_usuarios WHERE email = :email LIMIT 1");
$sql->bindParam(':email', $email, PDO::PARAM_STR);
$sql->execute();
if($sql && $sql->rowCount() == 0){
    //crear una clave
    $hashUser = time();
    $hashUser = md5($hashUser);
    $hashUser = substr($hashUser, 1, 25);
    $hashUser = str_pad($hashUser, 25, "0", STR_PAD_RIGHT);

    //preparar la consulta
    $sql=>prepare("INSERT INTO dash_usuarios (nombre, apellidos, email, password, hashUser, fechaRegistro) VALUES (:nombre, :apellidos, :email, :password, :hashUser, :fechaRegistro)");
    $sql->bindParam(':nombre', $nombre, PDO::PARAM_STR);
    $sql->bindParam(':apellidos', $apellidos, PDO::PARAM_STR);
    $sql->bindParam(':email', $email, PDO::PARAM_STR);
    $sql->bindParam(':password', $password1, PDO::PARAM_STR);
    $sql->bindParam(':hashUser', $hashUser, PDO::PARAM_STR);
    $sql->bindParam(':fechaRegistro', $fechaRegistro, PDO::PARAM_STR);

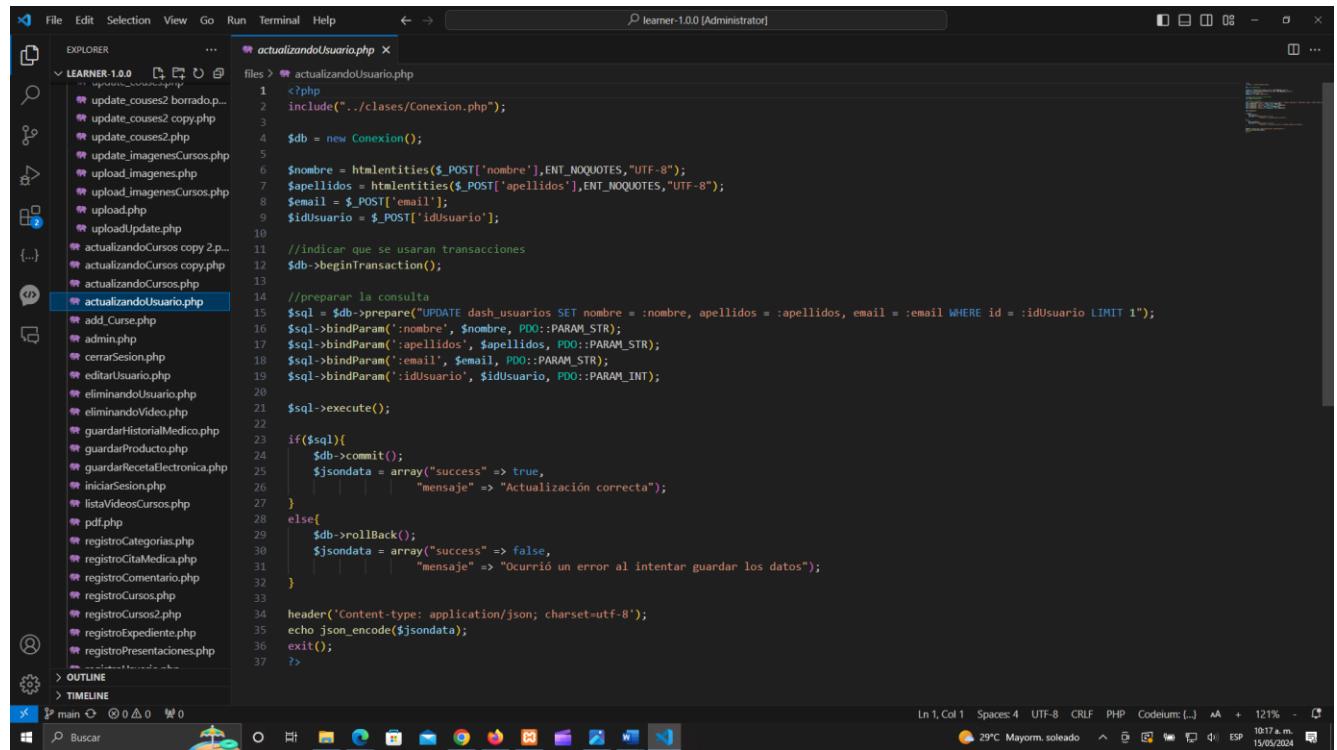
    $sql->execute();
    if($sql){
        $db->commit();
        $jsondata = array("success" => true,
                        "mensaje" => "registro correcto :3");
    }
}
else{
    $jsondata = array("success" => false,
                    "mensaje" => "correo electrónico o contraseña son incorrectos");
}
```

Ilustración 5.14 validar inicio sesión:



```
<?php
include("../clases/Conexion.php");
$db = new Conexion();
$usuario = $_POST['email'];
$password = $_POST['password'];
$recordarme = $_POST['recordarme'];
// $jsondata = array();
$db->beginTransaction();
$sql = $db->prepare("SELECT hashUser
                      FROM dash_usuarios
                      WHERE email = :email AND password = :password LIMIT 1");
$sql->bindParam(':email', $usuario, PDO::PARAM_STR);
$sql->bindParam(':password', $password, PDO::PARAM_STR);
$sql->execute();
if($sql){
    if ($sql->rowCount() > 0){
        $usuario_datos = $sql->fetch();
        if($recordarme == 1){
            setcookie("cDash", $usuario_datos['hashUser'], time(void): int -(86400 * 365),"/");
            setcookie("cDashExpiration", time() + (86400 * 365), time() + (86400 * 365),"/");
        }
        else{
            setcookie("cDash", $usuario_datos['hashUser'], 0, "/");
            setcookie("cDashExpiration", 0, 0, "/");
        }
        $jsondata['success'] = true;
    }
    else{
        $db->rollBack();
        $jsondata = array("success" => false,
                        "mensaje" => "El correo electrónico o contraseña son incorrectos");
    }
}
else{
    $db->rollBack();
    $jsondata = array("success" => false,
                    "mensaje" => "El correo electrónico o contraseña son incorrectos");
}
```

Ilustración 5.15 actualizar usuario:

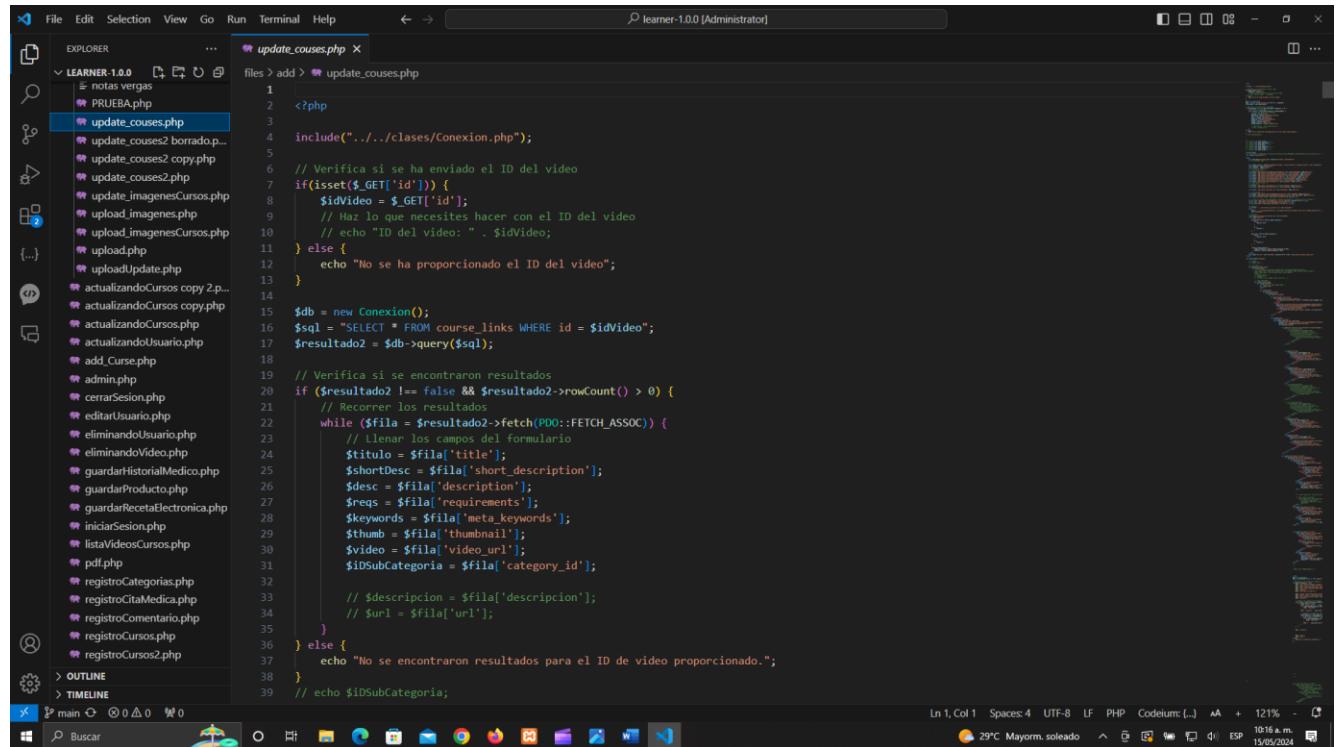


```

File Edit Selection View Go Run Terminal Help ↵ → learner-1.0.0 [Administrator]
EXPLORER ... actualizandoUsuario.php
files > actualizandoUsuario.php
1 <?php
2 include("../clases/Conexion.php");
3
4 $db = new Conexion();
5
6 $nombre = htmlentities($_POST['nombre'],ENT_NOQUOTES,"UTF-8");
7 $apellidos = htmlentities($_POST['apellidos'],ENT_NOQUOTES,"UTF-8");
8 $email = $_POST['email'];
9 $idUsuario = $_POST['idUsuario'];
10
11 //indicar que se usarán transacciones
12 $db->beginTransaction();
13
14 //Preparar la consulta
15 $sql = $db->prepare("UPDATE dash_usuarios SET nombre = :nombre, apellidos = :apellidos, email = :email WHERE id = :idUsuario LIMIT 1");
16 $sql->bindParam(':nombre', $nombre, PDO::PARAM_STR);
17 $sql->bindParam(':apellidos', $apellidos, PDO::PARAM_STR);
18 $sql->bindParam(':email', $email, PDO::PARAM_STR);
19 $sql->bindParam(':idUsuario', $idUsuario, PDO::PARAM_INT);
20
21 $sql->execute();
22
23 if($sql){
24     $db->commit();
25     $jsondata = array("success" => true,
26                       "mensaje" => "Actualización correcta");
27 }
28 else{
29     $db->rollBack();
30     $jsondata = array("success" => false,
31                       "mensaje" => "Ocurrió un error al intentar guardar los datos");
32 }
33
34 header('Content-type: application/json; charset=utf-8');
35 echo json_encode($jsondata);
36 exit();
37 >>
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP Codeium: (...) AA + 121% - 29°C Mayorm, soleado 10:17 a.m. 15/05/2024

```

Ilustración 5.16 actualizar curso:



```

File Edit Selection View Go Run Terminal Help ↵ → learner-1.0.0 [Administrator]
EXPLORER ... update_courses.php
files > add > update_courses.php
1 <?php
2
3 include("../clases/Conexion.php");
4
5 // Verifica si se ha enviado el ID del video
6 if(isset($_GET['id'])) {
7     $idVideo = $_GET['id'];
8     // Haz lo que necesites hacer con el ID del video
9     // echo "ID del video: " . $idVideo;
10 } else {
11     echo "No se ha proporcionado el ID del video";
12 }
13
14 $db = new Conexion();
15 $sql = "SELECT * FROM course_links WHERE id = $idVideo";
16 $resultado2 = $db->query($sql);
17
18 // Verifica si se encontraron resultados
19 if ($resultado2 !== false && $resultado2->rowCount() > 0) {
20     // Recorrer los resultados
21     while ($fila = $resultado2->fetch(PDO::FETCH_ASSOC)) {
22         // Llenar los campos del formulario
23         $titulo = $fila['title'];
24         $shortDesc = $fila['short_description'];
25         $desc = $fila['description'];
26         $reqs = $fila['requirements'];
27         $keywords = $fila['meta_keywords'];
28         $thumb = $fila['thumbnail'];
29         $video = $fila['video_url'];
30         $idSubCategoria = $fila['category_id'];
31
32         // $descripcion = $fila['descripcion'];
33         // $url = $fila['url'];
34     }
35 } else {
36     echo "No se encontraron resultados para el ID de video proporcionado.";
37 }
38
39 // echo $idSubCategoria;
Ln 1, Col 1 Spaces: 4 UTF-8 LF PHP Codeium: (...) AA + 121% - 29°C Mayorm, soleado 10:16 a.m. 15/05/2024

```

Ilustración 6. Diagrama Casos de Uso:

DIAGRAMA UML

E-learning GRAND VELAS



7. Manual de Usuario:

1. Introducción

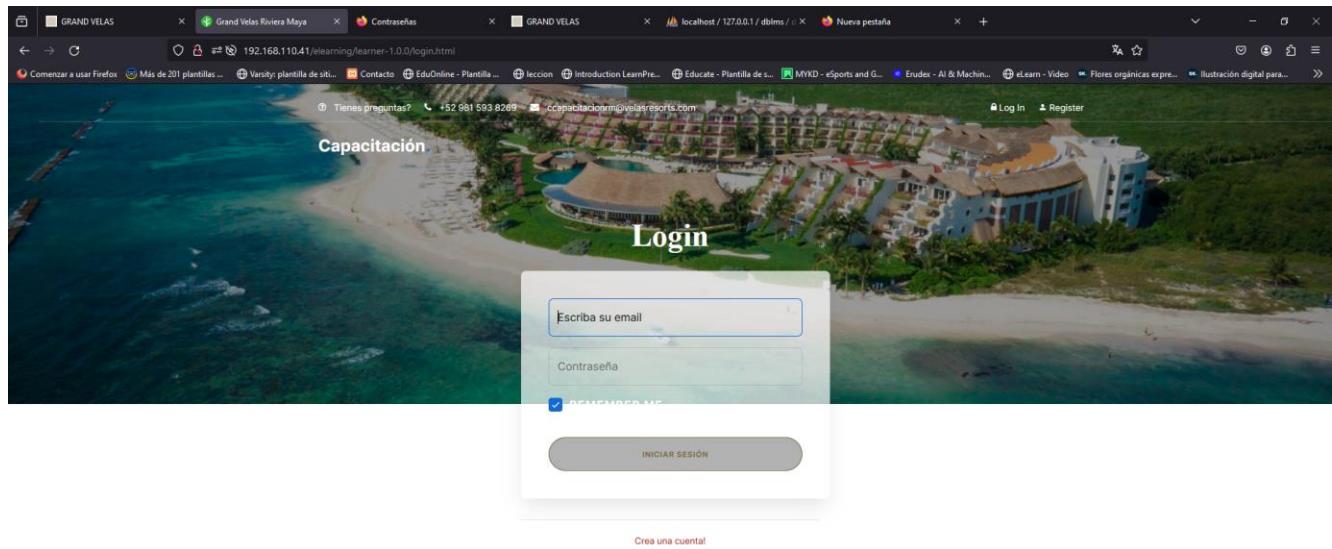
Bienvenido al Manual de Usuario de la Plataforma E-Learning de Grand Velas Riviera Maya. Este manual está diseñado para ayudarte a familiarizarte con las funcionalidades de la plataforma y sacar el máximo provecho de tu experiencia de aprendizaje.

2. Acceso a la Plataforma

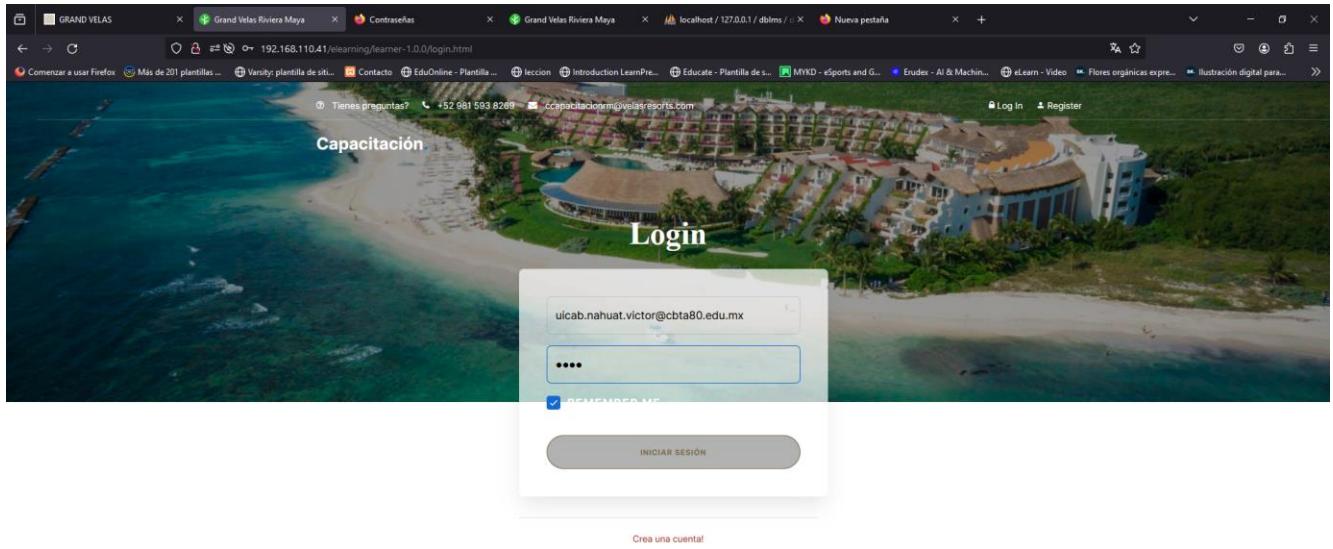
Para acceder a la plataforma, sigue estos pasos:

<http://192.168.110.41/elearning/learner-1.0.0/index.php>

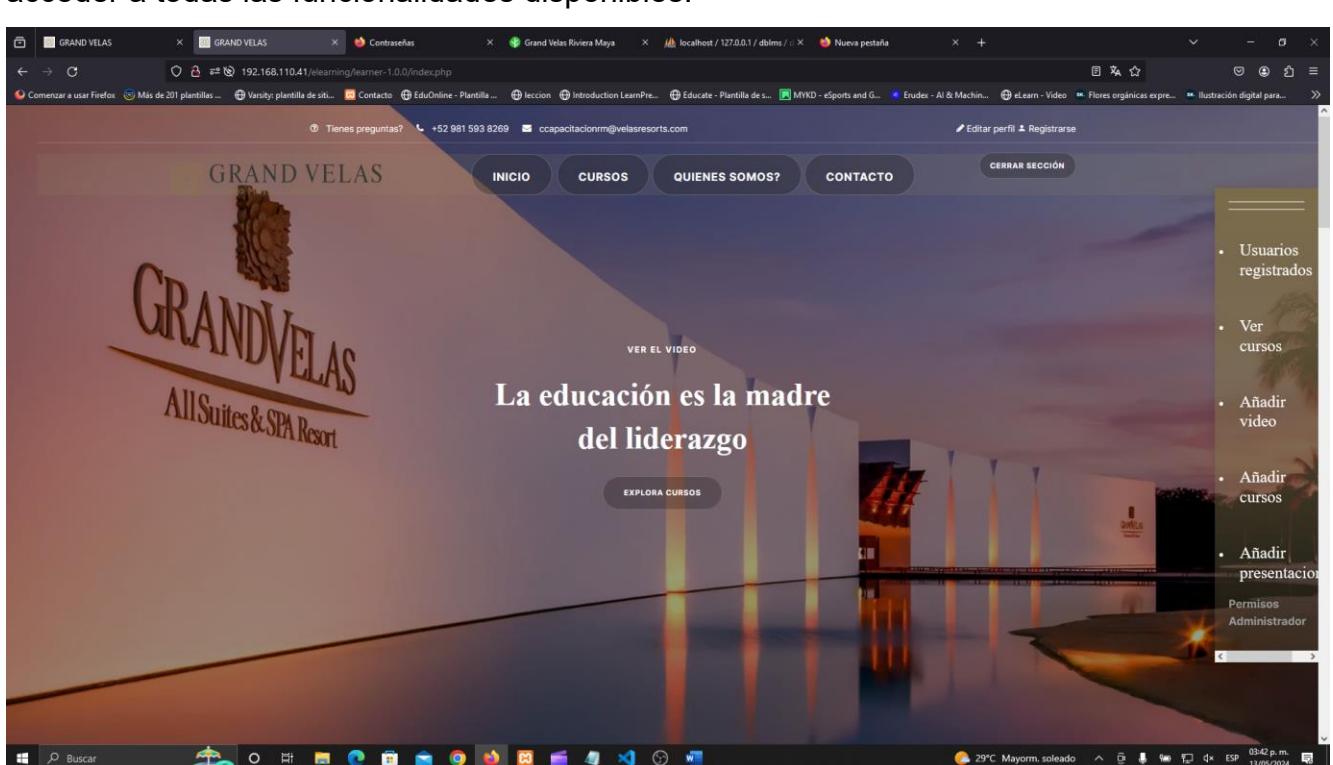
2.1. Abre tu navegador web e ingresa la URL proporcionada por tu administrador de la plataforma o te puedes registrar para uno.



2.2. Ingresa tus credenciales de inicio de sesión (nombre de usuario y contraseña) y haz clic en "Iniciar Sesión".



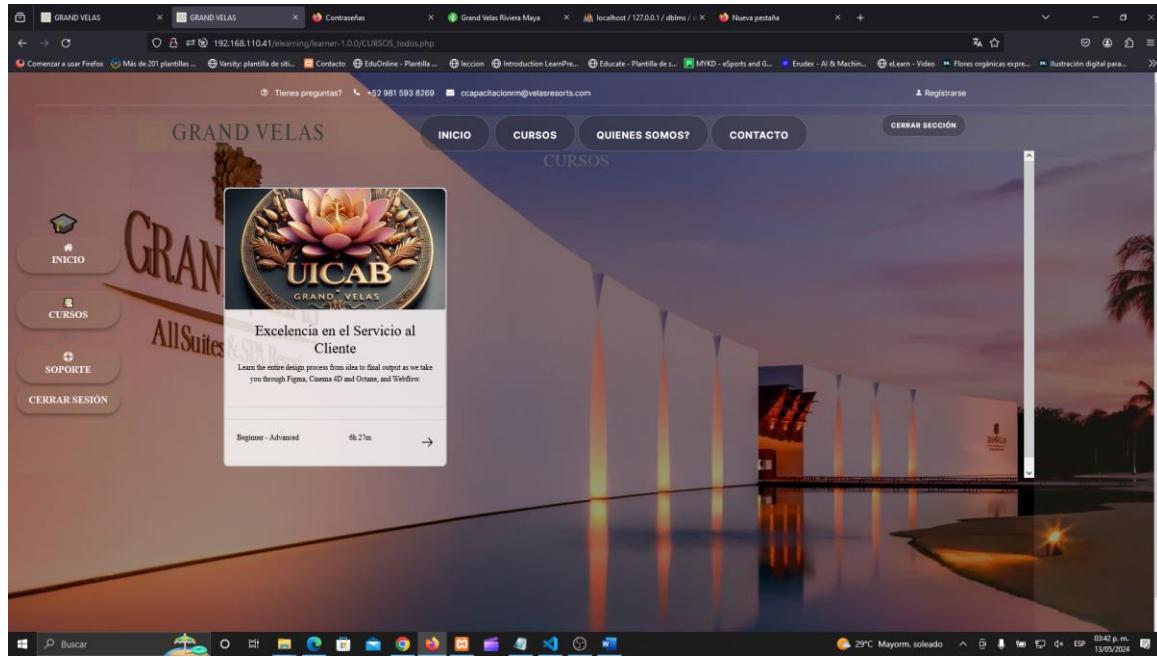
2.3. Una vez autenticado, serás dirigido al panel principal de la plataforma, donde podrás acceder a todas las funcionalidades disponibles.



3. Navegación y Funcionalidades

La plataforma E-Learning ofrece diversas funcionalidades para facilitar tu proceso de aprendizaje:

3.1 Visualización de Contenido: Explora los cursos y módulos disponibles y accede al contenido de capacitación de manera fácil y rápida dándole click a explorar cursos.



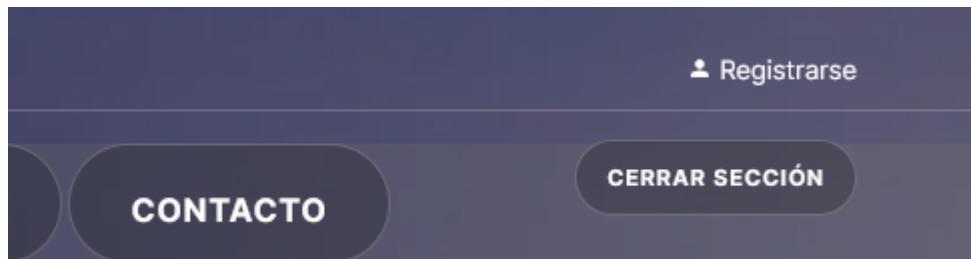
3.2. Completar Cursos y Módulos: Sigue el progreso de tu aprendizaje completando cursos y módulos interactivos.



3.4 Solicitar Asistencia Técnica: Si necesitas ayuda o encuentras algún problema técnico, puedes enviar una solicitud de asistencia técnica a nuestro equipo de soporte.

4. Cierre de Sesión

Recuerda cerrar sesión en la plataforma cuando hayas terminado de usarla, especialmente si estás accediendo desde un dispositivo compartido. Para cerrar sesión, haz clic en tu nombre de usuario y selecciona "Cerrar Sesión" en el menú.



5. Contacto y Soporte Técnico

Si tienes alguna pregunta o necesitas ayuda adicional, no dudes en ponerte en contacto con nuestro equipo de soporte técnico en ccapacitacionrm@velasresorts.com. Estamos aquí para ayudarte en cualquier momento.

8. Manual de Administrador:

1. Introducción

Bienvenido al Manual del Administrador de la Plataforma E-Learning de Grand Velas Riviera Maya. Este manual está diseñado para proporcionarte las instrucciones necesarias para administrar eficazmente la plataforma y garantizar una experiencia de aprendizaje óptima para los usuarios.

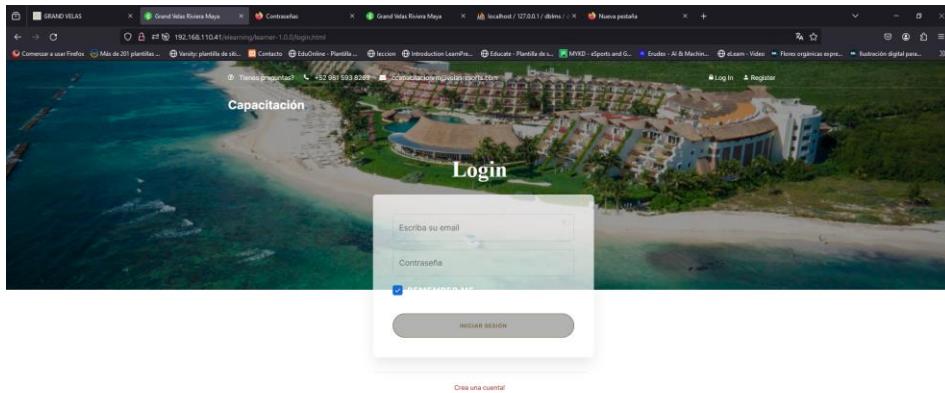
2. Acceso al Panel de Administración

Para acceder al panel de administración, sigue estos pasos:

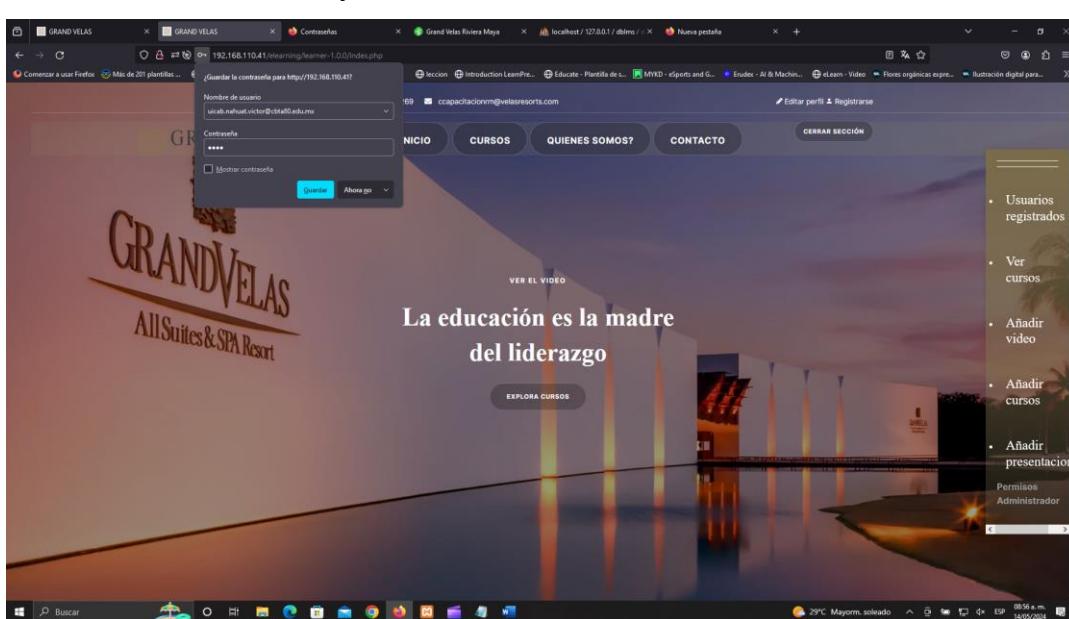
<http://192.168.110.41/elearning/learner-1.0.0/index.php>

2.1 Abre tu navegador web e ingresa la URL proporcionada por el proveedor de la plataforma.

Ingresá tus credenciales de administrador (nombre de usuario y contraseña) y haz clic en "Iniciar Sesión".



2.2 Una vez autenticado, serás dirigido al panel de administración, donde podrás acceder a todas las herramientas y funciones de administración.



3. Gestión de Usuarios

Desde el panel de administración, puedes realizar las siguientes acciones relacionadas con los usuarios:

The screenshot shows a user profile for 'Victor' with a dropdown menu open, containing 'Opciones' and 'Eliminar Usuario'. The sidebar on the right lists administrative actions related to users.

4. Administración de Contenido

4.1 Administra el contenido de capacitación de la plataforma:

The screenshot shows a table titled 'Lista de cursos' displaying course details. The table has columns for #, Título, Categoría, Lección y Sección, Estudiante, Inserto, Estado, Precio, and Acciones. The sidebar on the right lists administrative actions related to courses.

4.2 Agregar, editar y eliminar cursos y módulos de aprendizaje.

The screenshot shows a web browser window with multiple tabs open. The main content area displays two sections of course lists:

TEMA 1

#	Título	Subcategoría	Descripción	Acciones
1	Introducción a la importancia del servicio al cliente Instructor: Nombre del Instructor Ejemplo	Nombre Excelencia en el Servicio al Cliente	descripcion	
2	El arte de la comunicación efectiva Instructor: Nombre del Instructor Ejemplo	Nombre Excelencia en el Servicio al Cliente	descripcion	

TEMA 2

#	Título	Subcategoría	Descripción	Acciones
1	Comunicar en diferentes escenarios Instructor: Nombre del Instructor Ejemplo	Nombre Excelencia en el Servicio al Cliente	descripcion	
2	Como utilizar las habilidades para crear experiencias memorables Instructor: Nombre del Instructor Ejemplo	Nombre Excelencia en el Servicio al Cliente	descripcion	

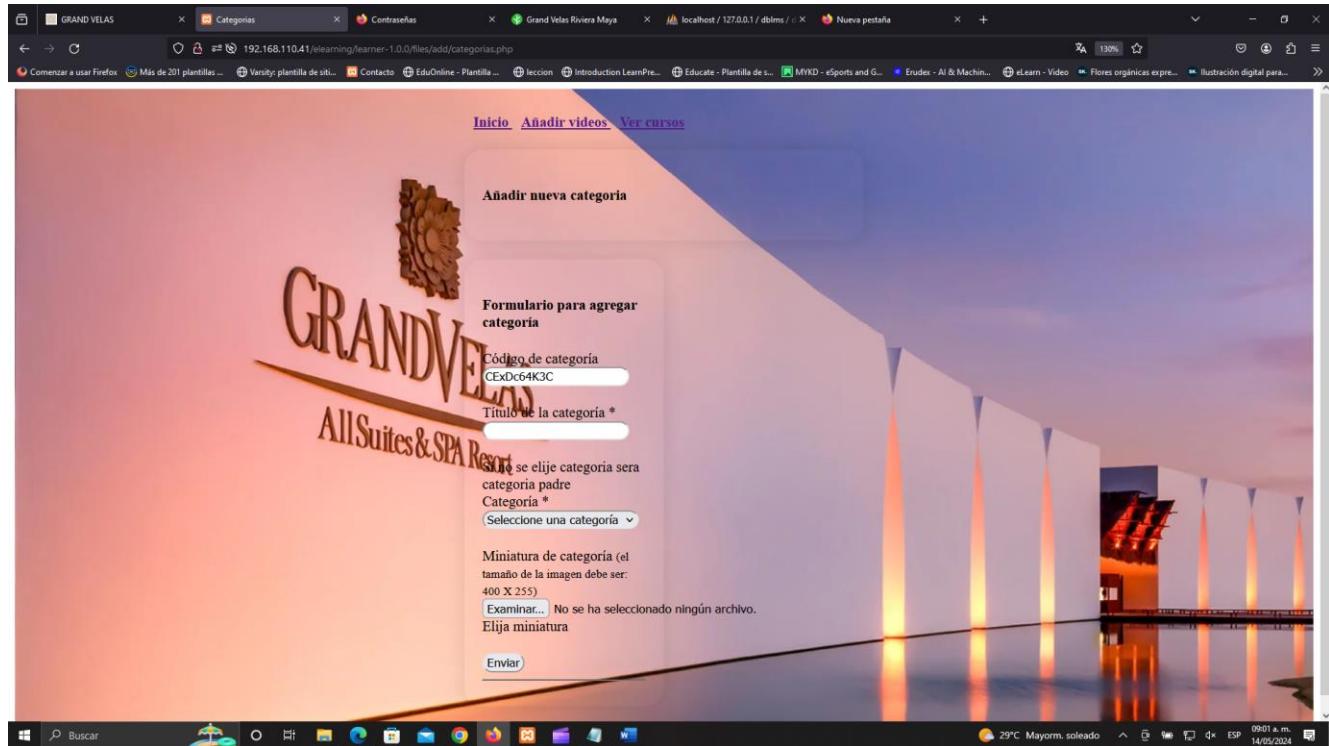
The screenshot shows a web browser window with multiple tabs open. The main content area displays a form titled "Formulario para agregar videos".

The form has several input fields and dropdown menus:

- Básico** tab selected.
- Requisitos** tab.
- SEO** tab.
- Finalizar** button.
- Nombre**: Input field containing "GRAND VELAS All Suites & SPA Resort".
- Descripción**: Input field containing "Este es un video de bienvenida a Grand Velas All Suites & SPA Resort".
- Categoría ***: Dropdown menu showing "Seleccionar una categoría".
- Nivel**: Dropdown menu showing "Principiante".
- Selector de Temas**: Dropdown menu showing "1".

The background of the form features a sunset over palm trees at a resort.

4.4 Agregar cursos (categorías y subcategorías)



Inicio [Añadir videos](#) [Ver cursos](#)

Añadir nueva categoría

Formulario para agregar categoría

Código de categoría
CExD64K3C

Titulo de la categoría *

Si se elige categoría sera categoría padre

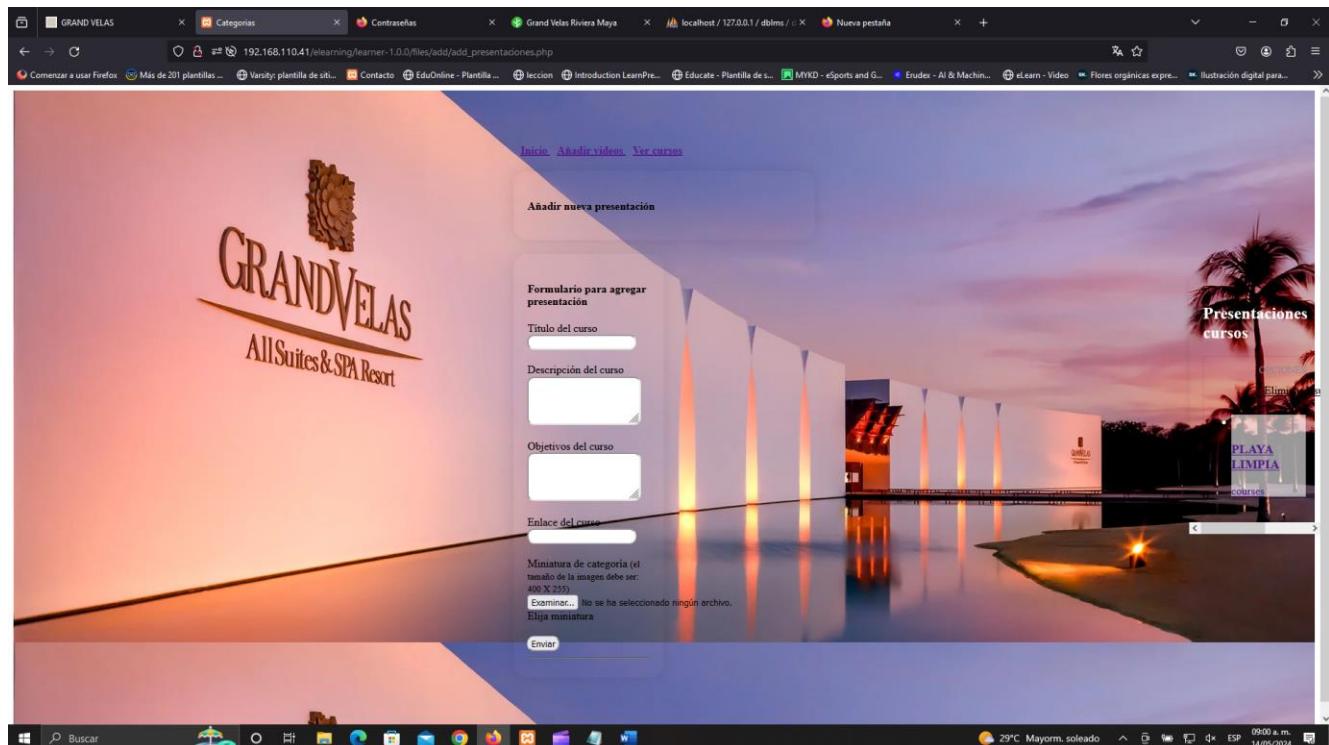
Categoría *

Seleccione una categoría ▾

Miniatrura de categoría (el tamaño de la imagen debe ser: 400 X 255)
Examinar... No se ha seleccionado ningún archivo.
Elija miniatura

Enviar

4.5 Gestionar archivos multimedia de recursos educativos.



Inicio [Añadir videos](#) [Ver cursos](#)

Añadir nueva presentación

Formulario para agregar presentación

Titulo del curso

Descripción del curso

Objetivos del curso

Palace del curso

Miniatrura de categoría (el tamaño de la imagen debe ser: 400X255)
Examinar... No se ha seleccionado ningún archivo.
Elija miniatura

Enviar

Presentaciones cursos

PLAYA LIMPIA

7. Soporte Técnico

Si necesitas ayuda o encuentras algún problema técnico, no dudes en ponerte en contacto con nuestro equipo de soporte técnico en ccapacitacionrm@velasresorts.com. Estamos aquí para ayudarte en cualquier momento.

9. Explicación del código:

9.1 Comenzaremos con la forma de manejar las solicitudes y la conexión a la base de datos:

class Conexion extends PDO: Define una nueva clase llamada Conexion que hereda todas las propiedades y métodos de la clase PDO.

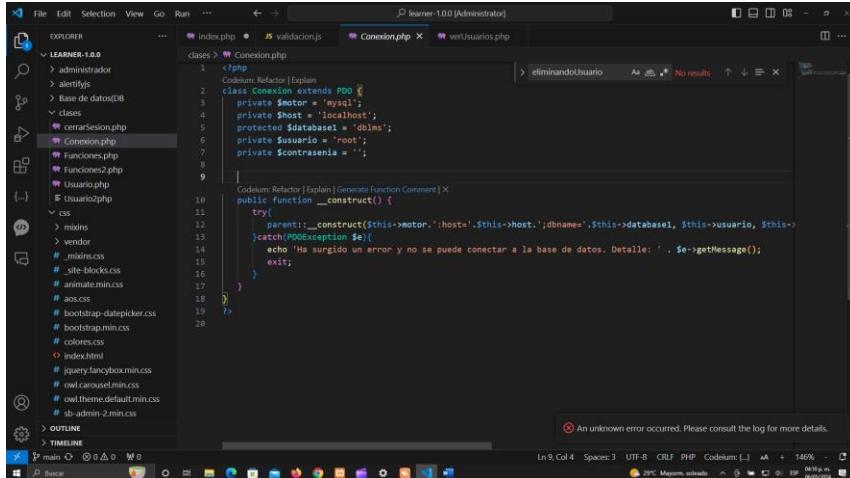
\$motor, \$host, \$database1, \$usuario, \$contrasenia: Estas son variables privadas que contienen la información necesaria para establecer la conexión con la base de datos.

__construct(): Este es el método constructor de la clase Conexion. Se ejecuta automáticamente cuando se crea un nuevo objeto Conexion. Dentro de este método, se intenta establecer una conexión con la base de datos utilizando los datos proporcionados.

parent::__construct(...): Llama al constructor de la clase base PDO con los parámetros necesarios para establecer la conexión.

catch(PDOException \$e): Captura cualquier excepción que ocurra durante el intento de conexión.

echo 'Ha surgido un error y no se puede conectar a la base de datos. Detalle: ' . \$e->getMessage();: En caso de error, imprime un mensaje indicando que ha ocurrido un error de conexión con la base de datos.



```
<?php
/*
    Código: Refactor | Explains
    Clase: Conexion.php
    */
class Conexion extends PDO {
    private $motor = 'mysql';
    private $host = 'localhost';
    protected $database1 = 'dolce';
    private $usuario = 'root';
    private $contrasenia = '';
    /**
     * Código: Refactor | Explain | Generate Function Comment | X
     */
    public function __construct() {
        try{
            parent::__construct($this->motor,'$this->host','$this->dbname',$this->database1,$this->usuario,$this->contrasenia);
        }catch(PDOException $e){
            echo 'Ha surgido un error y no se puede conectar a la base de datos. Detalle: ' . $e->getMessage();
            exit;
        }
    }
}
```

JavaScript: Función para Enviar Datos al Servidor

Este bloque de código JavaScript define una función llamada ajaxPostControl que se utiliza para enviar datos al servidor a través de AJAX. Aquí hay una explicación línea por línea:

const ajaxPostControl = (...)=> { ... }: Define una función llamada ajaxPostControl.

document.getElementById(divDestino).style.display = "block";: Muestra un elemento HTML con el ID especificado.

\$("#div#" + divDestino).html("");: Establece el contenido HTML de un elemento div con el ID especificado como una imagen de carga.

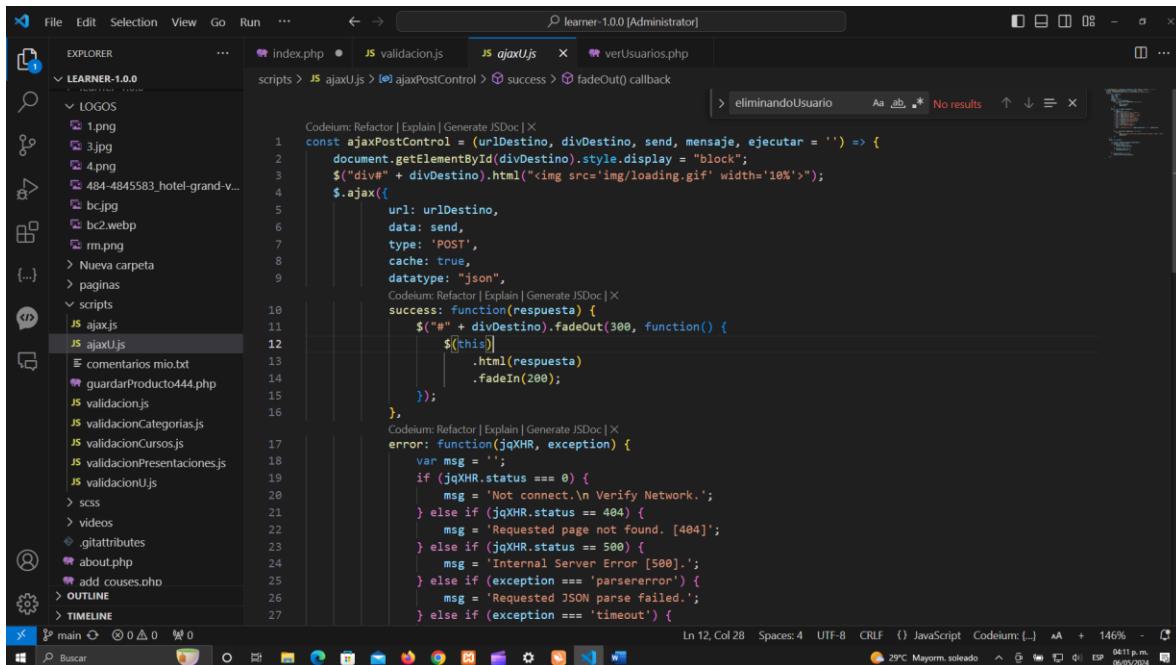
\$.ajax({ ... }): Realiza una petición AJAX al servidor.

url, data, type, cache, datatype: Parámetros de configuración de la petición AJAX.

success, error: Funciones de devolución de llamada para manejar respuestas exitosas o errores de la petición.

.done(function(data) { ... }): Función de devolución de llamada para manejar la respuesta del servidor después de que se complete la petición AJAX.

var content = JSON.parse(JSON.stringify(data));: Convierte los datos de la respuesta del servidor en un objeto JavaScript.



The screenshot shows a code editor interface with the following details:

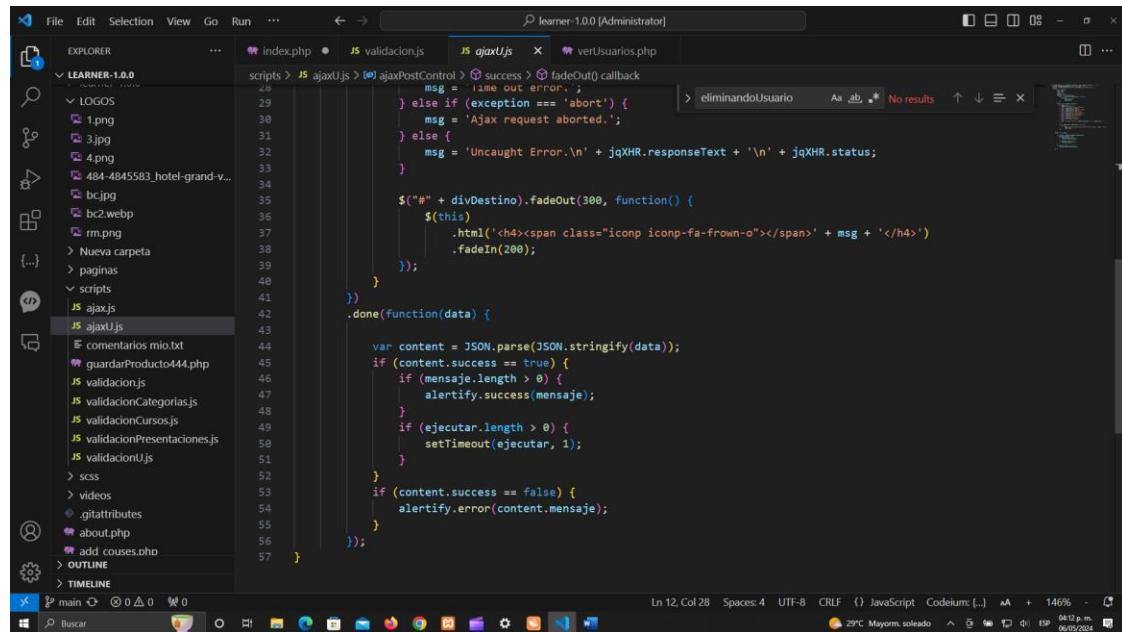
- File Explorer:** Shows a project structure for "LEARNER-1.0.0" with various files like index.php, JS validation.js, and JS ajaxUJs.
- Code Editor:** The active file is "JS ajaxUJs". It contains the following code:

```
const ajaxPostControl = (urlDestino, divDestino, send, mensaje, ejecutar = '') => {
    document.getElementById(divDestino).style.display = "block";
    $("#div#" + divDestino).html("<img src='img/loading.gif' width='10%'>");
    $.ajax({
        url: urlDestino,
        data: send,
        type: 'POST',
        cache: true,
        datatype: "json",
        success: function(respuesta) {
            $("#div#" + divDestino).fadeOut(300, function() {
                $(this)
                    .html(respuesta)
                    .fadeIn(200);
            });
        },
        error: function(jqXHR, exception) {
            var msg = '';
            if (jqXHR.status === 0) {
                msg = 'Not connect.\nVerify Network.';
            } else if (jqXHR.status == 404) {
                msg = 'Requested page not found. [404]';
            } else if (jqXHR.status == 500) {
                msg = 'Internal Server Error [500].';
            } else if (exception === 'parsererror') {
                msg = 'Requested JSON parse failed.';
            } else if (exception === 'timeout') {
                msg = 'Time-out error.';
            }
        }
    });
}
```

- Status Bar:** Shows line 12, column 28, and other standard status bar information.

if (content.success == true) { ... }: Verifica si la respuesta del servidor indica éxito.

`if (content.success == false) { ... };`: Verifica si la respuesta del servidor indica un error.



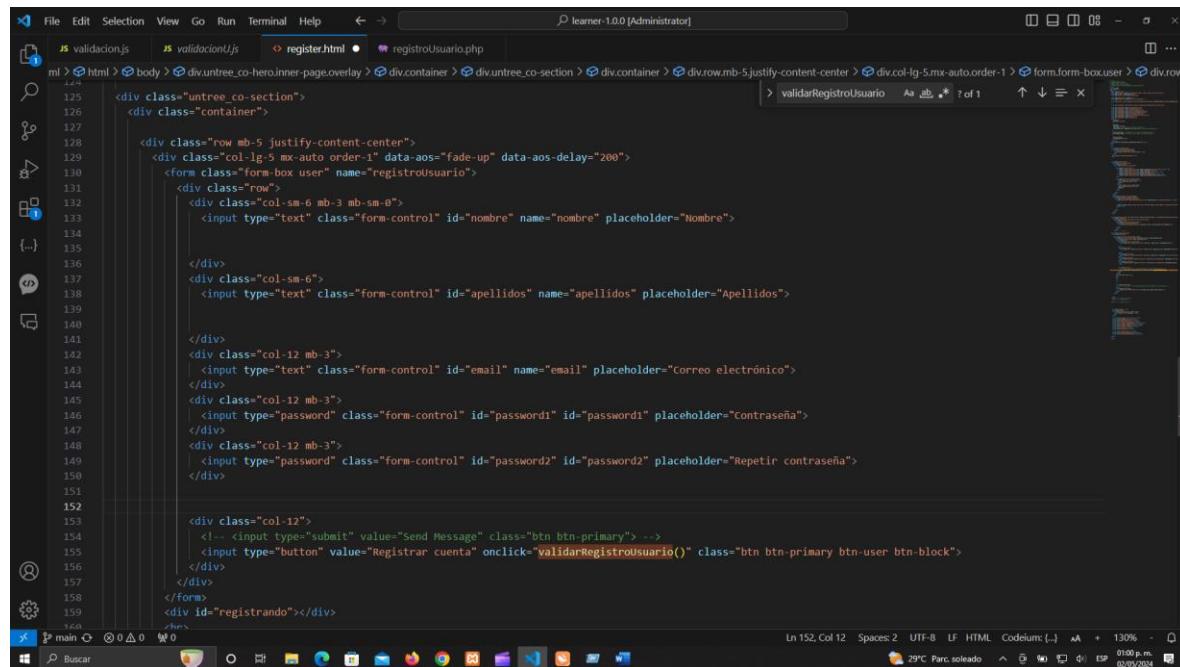
```
File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
EXPLORER ...
LEARNER-1.0.0 ...
LOGOS
1.jpg
3.jpg
4.png
484-4845583.hotel-grand-v...
bc.jpg
bc2.webp
rm.png
Nueva carpeta
paginas
scripts
JS ajax.js
JS ajaxUjs
comentarios mio.txt
guardarProducto444.php
JS validation.js
JS validationCategorias.js
JS validationCursos.js
JS validationPresentaciones.js
JS validationUjs
scss
videos
.gitattributes
about.php
add courses.php
OUTLINE
TIMELINE
index.php JS validar.js x JS ajaxUjs x verUsuarios.php
scripts > JS ajaxUjs > [o] ajaxPostControl > [o] success > [o] fadeOut() callback
26     msg = 'Time out error.';
29   } else if (exception === 'abort') {
30     msg = 'Ajax request aborted.';
31   } else {
32     msg = 'Uncaught Error.\n' + jqXHR.responseText + '\n' + jqXHR.status;
33   }
34
35   $("#" + divDestino).fadeOut(300, function() {
36     $(this)
37       .html('<h4><span class="iconp iconp-frown-o"></span>' + msg + '</h4>')
38       .fadeIn(200);
39   });
40
41   ).done(function(data) {
42
43     var content = JSON.parse(JSON.stringify(data));
44     if (content.success == true) {
45       if (mensaje.length > 0) {
46         alertify.success(mensaje);
47       }
48       if (ejecutar.length > 0) {
49         setTimeout(ejecutar, 1);
50       }
51     }
52     if (content.success == false) {
53       if (mensaje.length > 0) {
54         alertify.error(content.mensaje);
55       }
56     }
57   });
Ln 12, Col 28 Spaces: 4 UTF-8 CRLF {} JavaScript Codeium: [...] AA + 146% - 29°C Mayorm. soleado 08:12 p. m. 06/05/2024
```

9.2 Usuario:

Registro usuarios:

HTML (Registro de Usuarios):

He creado un formulario de registro de usuarios que recolecta información básica como nombre, apellidos, correo electrónico y contraseña. Cada campo del formulario tiene un identificador único (id) y un nombre (name) que utilizaremos para acceder a los valores desde JavaScript y PHP. Además, he incluido un botón "Registrar cuenta" que, al hacer clic, activa la función validarRegistroUsuario() en JavaScript para validar y enviar los datos del formulario.



The screenshot shows a code editor with several files open in tabs: validation.js, validationU.js, register.html, and registroUsuario.php. The register.html file is the active tab and contains the following HTML code:

```
125 <div class="untree_co-section">
126 <div class="container">
127
128 <div class="row mb-5 justify-content-center">
129 <div class="col-lg-5 mx-auto order-1" data-aos="fade-up" data-aos-delay="200">
130 <form class="form-box user" name="registroUsuario">
131 <div class="row">
132 <div class="col-sm-6 mb-3 mb-sm-0">
133 <input type="text" class="form-control" id="nombre" name="nombre" placeholder="Nombre">
134 </div>
135 <div class="col-sm-6">
136 <input type="text" class="form-control" id="apellidos" name="apellidos" placeholder="Apellidos">
137 </div>
138 <div class="col-12 mb-3">
139 <input type="text" class="form-control" id="email" name="email" placeholder="Correo electrónico">
140 </div>
141 <div class="col-12 mb-3">
142 <input type="password" class="form-control" id="password1" name="password1" placeholder="Contraseña">
143 </div>
144 <div class="col-12 mb-3">
145 <input type="password" class="form-control" id="password2" name="password2" placeholder="Repetir contraseña">
146 </div>
147
148 <div class="col-12">
149 <!-- <input type="submit" value="Send Message" class="btn btn-primary" -->
150 <input type="button" value="Registrar cuenta" onclick="validarRegistroUsuario()" class="btn btn-primary btn-user btn-block">
151 </div>
152 </div>
153 </div>
154 <!-- <div id="registrar" style="text-align: center;">-->
155 <div id="registrar" style="text-align: center; margin-top: 10px;">
156 <div id="registrando"></div>
157 </div>
158 </div>
159 </div>
```

JavaScript (Validación del Registro de Usuarios):

He definido la función validarRegistroUsuario() que se ejecuta al hacer clic en el botón de registro. En esta función, obtenemos los valores de los campos del formulario y realizamos una serie de validaciones para asegurarnos de que los datos proporcionados sean válidos. Si todas las validaciones son exitosas, creamos un objeto JavaScript con los datos del usuario y realizamos una solicitud AJAX para enviar estos datos al servidor. Además, he incluido una función verInicio() que se ejecuta después de que el registro sea exitoso, redirigiendo al usuario a la página de inicio de sesión.

```

File Edit Selection View Go Run Terminal Help
File > validation.js > validation.html > register.html > validarRegistroUsuario.php
validation.js validation.html register.html validarRegistroUsuario.php
  Codeium Refactor (Autosave: Enabled) (0/0)
  1 const validarRegistroUsuario = () => {
  2   let nombre = document.getElementById("nombre").value;
  3   let apellido = document.getElementById("apellidos").value;
  4   let email = document.getElementById("email").value;
  5   let password = document.getElementById("password").value;
  6   let confirmPassword = document.getElementById("confirmPassword").value;
  7
  8   if (nombre.length == 0) {
  9     alertify.error("Escriba el nombre del usuario");
 10    return false;
 11  }
 12  if (apellido.length == 0) {
 13    alertify.error("Escriba los apellidos del usuario");
 14    return false;
 15  }
 16  if (email.length == 0) {
 17    alertify.error("Escriba el email del usuario");
 18    return false;
 19  }
 20  if (password.length == 0) {
 21    alertify.error("Escriba la contraseña");
 22    return false;
 23  }
 24  if (password !== confirmPassword) {
 25    alertify.error("Las contraseñas no coinciden");
 26    return false;
 27  }
 28
 29  alert("Mensaje de prueba");
 30
 31  let send = {
 32   nombre,
 33   apellido,
 34   email,
 35   password
 36 };
 37  let url = "file://registroUsuario.php";
 38  let message = "Los datos se guardaron correctamente";
 39  let execute = "window.open(url)";
 40
 41  ejecutarControl(send, "registrarUsuario", send, message, ejecutar);
 42
 43
 44  Codeium Refactor (Autosave: Enabled) (0/0)
 45  const verificar = () => {
 46    location.href = login.html;
 47  }

```

PHP (Procesamiento del Registro de Usuarios):

He creado un archivo PHP que procesa la solicitud de registro de usuarios. En este archivo, primero establecemos una conexión con la base de datos y obtenemos los datos enviados desde el formulario HTML utilizando el método POST. Luego, verificamos si el correo electrónico del usuario ya está registrado en la base de datos. Si el correo electrónico no está registrado, generamos una clave única para el usuario y insertamos los datos del usuario en la tabla de usuarios. Además, hemos utilizado el método de encriptación SHA-256 para almacenar la contraseña de forma segura en la base de datos. Finalmente, devolvemos una respuesta JSON al cliente indicando si el registro fue exitoso o si hubo algún error.

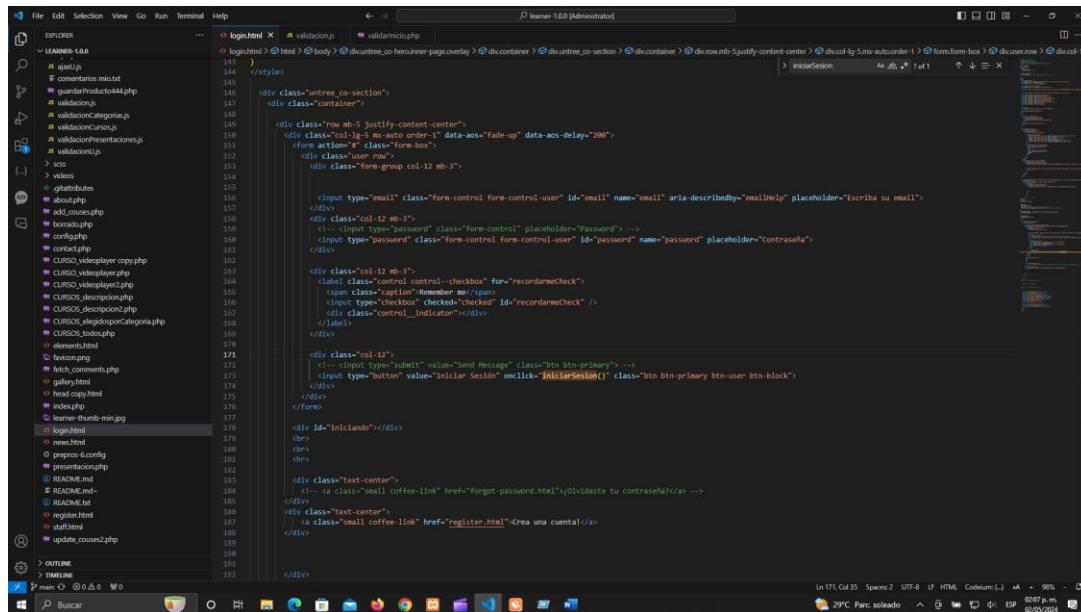
```

File Edit Selection View Go Run Terminal Help
File > registroUsuario.php > validation.js > validarRegistroUsuario.php
registroUsuario.php validation.js validarRegistroUsuario.php
  Codeium Refactor (Autosave: Enabled) (0/0)
  1 <?php
 2   include("../clases/conexion.php");
 3
 4   $db = new Conexion();
 5
 6   // obtener las variables de la función js
 7   $nombre = htmlspecialchars($_POST["nombre"], ENT_QUOTES, "UTF-8");
 8   $apellidos = htmlspecialchars($_POST["apellidos"], ENT_QUOTES, "UTF-8");
 9   $email = $_POST["email"];
10
11   // mi contraseña
12   $password = hash("sha256", $_POST["password"]);
13   $fechaRegistro = date("Y-m-d");
14
15   // indicar que se usaran transacciones
16   $db->beginTransaction();
17
18   // verificar que el correo no existe en la BD
19   $sql = $db->prepare("SELECT email FROM dash_usuarios WHERE email = :email LIMIT 1");
20   $sql->bindParam(":email", $email, PDO::PARAM_STR);
21   $sql->execute();
22
23   if($sql && $sql->rowCount() == 0){
24
25     // crear una clave
26     $hashUser = time();
27     $hashUser = md5($hashUser);
28     $hashUser = substr($hashUser, 1, 25);
29     $hashUser = str_pad($hashUser, 25, "0", STR_PAD_RIGHT);
30
31     // preparar la consulta
32     $sql = $db->prepare("INSERT INTO dash_usuarios (nombre, apellidos, email, password, hashUser, fechaRegistro) VALUES (:nombre, :apellidos, :email, :password, :hashUser, :fechaRegistro)");
33     $sql->bindParam(":nombre", $nombre, PDO::PARAM_STR);
34     $sql->bindParam(":apellidos", $apellidos, PDO::PARAM_STR);
35     $sql->bindParam(":email", $email, PDO::PARAM_STR);
36     $sql->bindParam(":password", $password, PDO::PARAM_STR);
37     $sql->bindParam(":hashUser", $hashUser, PDO::PARAM_STR);
38     $sql->bindParam(":fechaRegistro", $fechaRegistro, PDO::PARAM_STR);
39
40     $sql->execute();
41
42     if($sql){
43       $db->commit();
44       $jsondata = array("success" => true,
45                         "mensaje" => "registro correcto :3");
46     } else{
47       $db->rollBack();
48       $jsondata = array("success" => false,
49                         "mensaje" => "Ocurrió un error al intentar guardar los datos");
50     }
51
52   }

```

iniciar sesión:

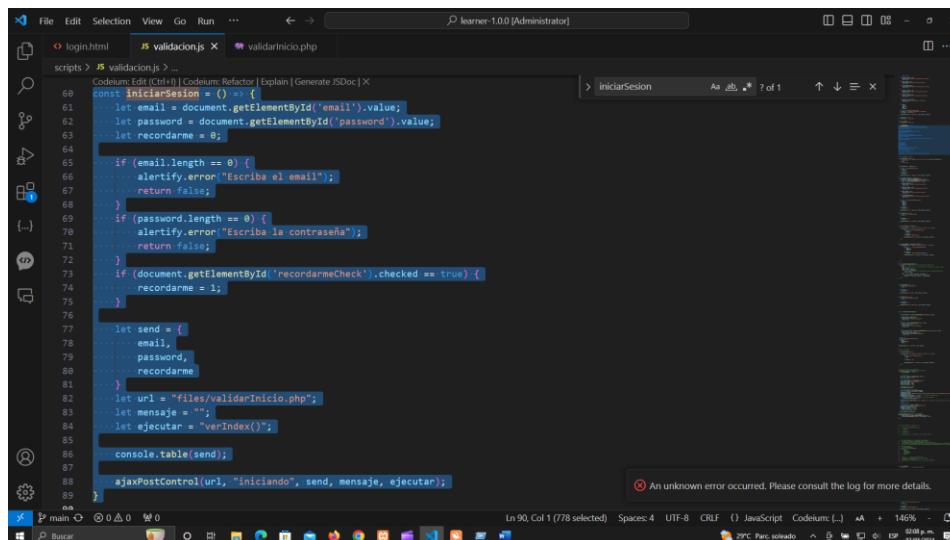
He creado un formulario de inicio de sesión que recopila la dirección de correo electrónico y la contraseña del usuario. Además, el formulario incluye una casilla de verificación para recordar al usuario en el dispositivo.



```
<div class="row mb-3 justify-content-center">
  <div class="col-12" data-aos="fade-up" data-aos-delay="200">
    <div class="row" data-aos="fade-up" data-aos-delay="200">
      <div class="col-12 mb-3" data-aos="fade-up" data-aos-delay="200">
        <div class="form-group col-12 mb-3">
          <label for="email" data-aos="fade-up" data-aos-delay="200">Email</label>
          <input type="email" class="form-control form-control-user" id="email" name="email" aria-describedby="emailHelp" placeholder="Escriba su email" data-aos="fade-up" data-aos-delay="200"/>
        </div>
        <div class="form-group col-12 mb-3" data-aos="fade-up" data-aos-delay="200">
          <label for="password" data-aos="fade-up" data-aos-delay="200">Contraseña</label>
          <input type="password" class="form-control form-control-user" id="password" name="password" placeholder="Contraseña" data-aos="fade-up" data-aos-delay="200"/>
        </div>
        <div class="col-12 control">
          <label class="control control--checkbox" for="recordarmeCheck">
            <input type="checkbox" checked="" id="recordarmeCheck" data-aos="fade-up" data-aos-delay="200"/>
            <span data-aos="fade-up" data-aos-delay="200" class="caption">Recordarme
          </label>
          <label class="control control--indicator" data-aos="fade-up" data-aos-delay="200"></label>
        </div>
      </div>
      <div class="col-12">
        <!-- <input type="submit" value="Send Message" class="btn btn-primary" -->
        <input type="button" value="Iniciar Sesión" onclick="iniciarSesion()" class="btn btn-primary btn-user btn-block" data-aos="fade-up" data-aos-delay="200"/>
      </div>
    </div>
  </div>
</div>
<div id="iniciando" data-aos="fade-up" data-aos-delay="200">
  <br>
  <br>
  <br>
  <div class="text-center">
    <!-- <a class="small coffee-link" href="forgot-password.html">Olvidaste tu contraseña?</a> -->
    <div class="text-center">
      <a class="small coffee-link" href="register.html">Crea una cuenta</a>
    </div>
  </div>
</div>
```

JavaScript (Validación del Inicio de Sesión):

Para validar el inicio de sesión, he definido la función iniciarSesion(), que se activa al hacer clic en el botón "Iniciar Sesión". Dentro de esta función, se obtienen los valores de los campos del formulario (correo electrónico y contraseña). Se asegura que ambos campos no estén vacíos y, si el usuario ha marcado la casilla de verificación "Recordarme", se asigna el valor 1 a la variable recordarme.



```
const iniciarSesion = () => {
  let email = document.getElementById('email').value;
  let password = document.getElementById('password').value;
  let recordarme = 0;

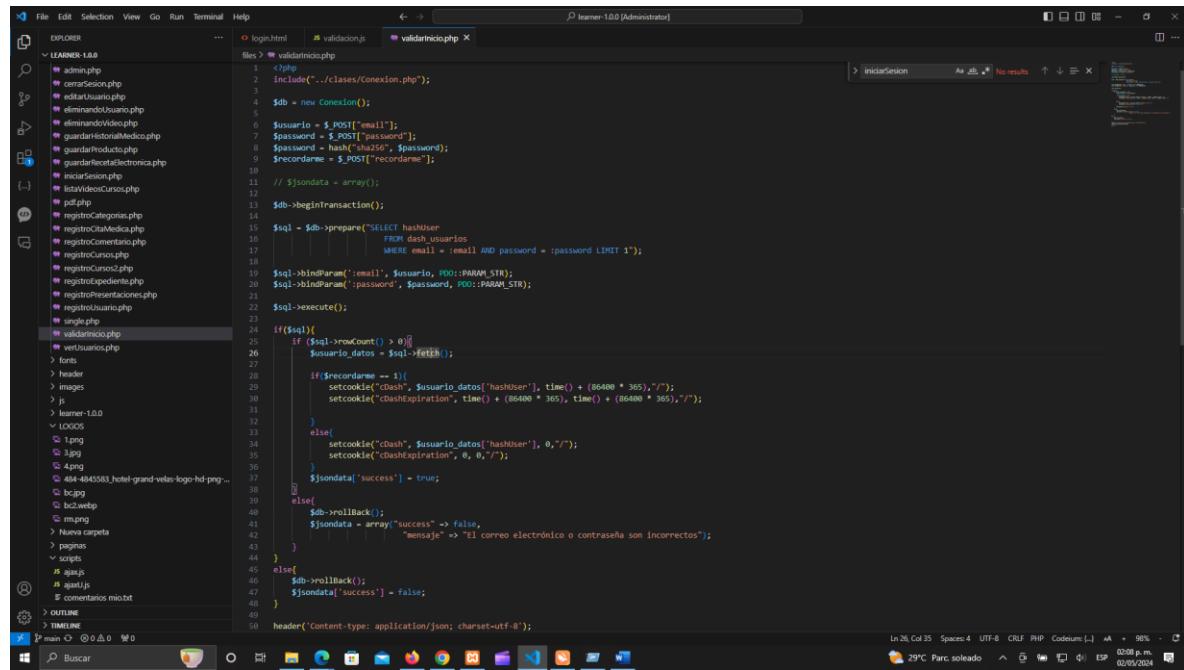
  if (email.length == 0) {
    alertify.error("Escriba el email");
    return false;
  }
  if (password.length == 0) {
    alertify.error("Escriba la contraseña");
    return false;
  }
  if (document.getElementById('recordarmeCheck').checked == true) {
    recordarme = 1;
  }

  let send = [
    email,
    password,
    recordarme
  ];
  let url = "files/validarInicio.php";
  let mensaje = "";
  let ejecutar = "verIndex()";
  console.table(send);
  ajaxPostControl(url, "iniciando", send, mensaje, ejecutar);
}

An unknown error occurred. Please consult the log for more details.
```

PHP (Procesamiento del Inicio de Sesión):

Para procesar la solicitud de inicio de sesión, he creado un archivo PHP. Este archivo establece una conexión con la base de datos y obtiene los datos enviados desde el formulario HTML utilizando el método POST. La contraseña proporcionada se encripta utilizando el algoritmo SHA-256 para compararla con la contraseña almacenada en la base de datos. Luego, se realiza una consulta SQL para verificar si existe un usuario con el correo electrónico y la contraseña proporcionados. Si las credenciales son correctas y el usuario ha marcado la casilla "Recordarme", se crean cookies para mantener la sesión del usuario durante un período prolongado. Finalmente, se devuelve una respuesta JSON al cliente indicando si el inicio de sesión fue exitoso o hubo algún error.



The screenshot shows a Windows desktop environment with a code editor window open. The editor displays a PHP script named 'validacion.php'. The script includes database connection logic, password hashing, and a query to check user existence. It also handles session management by setting cookies if the 'remember' checkbox is checked. The code concludes with a JSON response header. The desktop taskbar at the bottom shows various icons for system and application windows.

```
File Edit Selection View Go Run Terminal Help
  login.html validation.php validarInicio.php
  File Explorer
  LEARNER-1.0
  admin.php
  certificado.php
  eliminarUser.php
  eliminarVideo.php
  guardarHistorialMedico.php
  guardarProducto.php
  guardarRecetaElectronica.php
  iniciarSesion.php
  listarCursos.php
  pdf.php
  registrarCategorias.php
  registrarMedico.php
  registrarComentario.php
  registrarCursos.php
  registrarCursos.php
  registrarCuenta.php
  registrarPediente.php
  registrarReseñas.php
  simple.php
  validarInicio.php
  validaciones.php
  > fonts
  > header
  > images
  > js
  > learner-1.0.0
  > LOGOS
    1.png
    3.jpg
    4.png
    484-4845883_hotel-grand-velas-logo-hd-prg...
  > b2.js
  > b2c.js
  > b2cHelp.js
  > img.js
  > Nombre carpeta
  > paginas
  > scripts
    # ajax
    # ajax.js
    # comentarios mio.txt
  > Timeline
  main
  Buscar
  Timeline
  29°C Parc soleado
  02/05/2024
```

```
1 //login.php
2 include("../clases/conexion.php");
3
4 $db = new Conexion();
5
6 $usuario = $_POST["email"];
7 $password = $_POST["password"];
8 $password = hash("sha256", $password);
9 $recordarme = $_POST["recordarme"];
10
11 // $jsondata = array();
12
13 $db->beginTransaction();
14
15 $sql = $db->prepare("SELECT * FROM dash_usuarios
16 WHERE email = :email AND password = :password LIMIT 1");
17
18 $sql->bindParam(":email", $usuario, PDO::PARAM_STR);
19 $sql->bindParam(":password", $password, PDO::PARAM_STR);
20
21 $sql->execute();
22
23 if($sql->rowCount() > 0){
24     $usuario_datos = $sql->fetch();
25
26     if($recordarme == 1){
27         setcookie("cDash", $usuario_datos["hashUser"], time() + (86400 * 365), "/");
28         setcookie("cDashExpiration", time() + (86400 * 365), time() + (86400 * 365), "/");
29     }
30     else{
31         setcookie("cDash", $usuario_datos["hashUser"], 0, "/");
32         setcookie("cDashExpiration", 0, 0, "/");
33     }
34     $jsondata['success'] = true;
35
36 } else{
37     $db->rollback();
38     $jsondata = array("mensaje" => "El correo electrónico o contraseña son incorrectos");
39 }
40
41 else{
42     $db->rollback();
43     $jsondata['success'] = false;
44 }
45
46 header('Content-type: application/json; charset=utf-8');
```

Editar perfil:

HTML (Edición de Perfil):

He creado un enlace llamado "Editar perfil" que permite a los usuarios modificar su información personal. Al hacer clic en este enlace, se activa la función editarUsuario() en JavaScript y se muestra un formulario de edición.

```

101  Codeum: Refactor | Explain | Generate JSDoc | X
102  const editarUsuario = (idUsuario) => {
103
104      let url = "files/editarUsuario.php";
105      let send = {
106          idUsuario
107      }
108      let mensaje = '';
109      let ejecutar = '';
110
111      ajaxPostControl(url, "informacion", send, mensaje, ejecutar);
112

```

```

    class="d-none d-lg-inline-block">ccapacitacionrm@velasresorts.com</span></a>
</div>

<div class="col-6 col-lg-3 text-right">
<a href="javascript:editarUsuario(<?php echo $idUsuario; ?>)" class="small" onclick="mostrarEditarPerfil()">

Editar perfil
</a>

<a href="register.html" class="small">

Registrarse
</a>

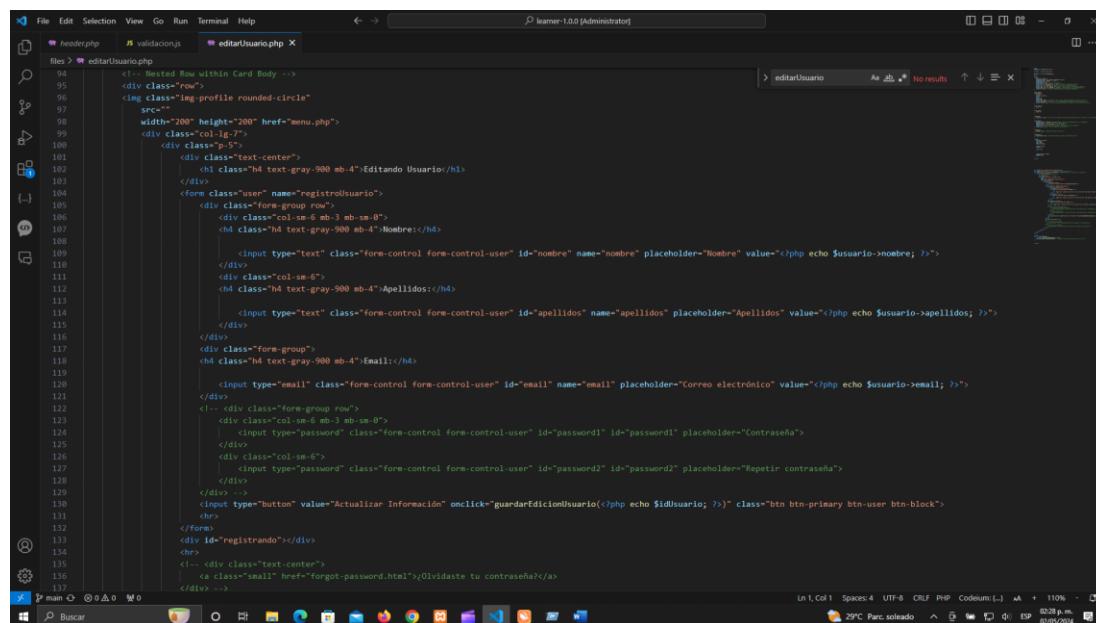
```

Al proporcionar a los usuarios la capacidad de editar su perfil, nos aseguramos de que puedan mantener actualizada su información personal según sea necesario al mismo tiempo consultar sus datos existentes. Para facilitar esta tarea, he creado un formulario que recopila los siguientes detalles:

Nombre: Un campo donde los usuarios pueden ingresar su nombre.

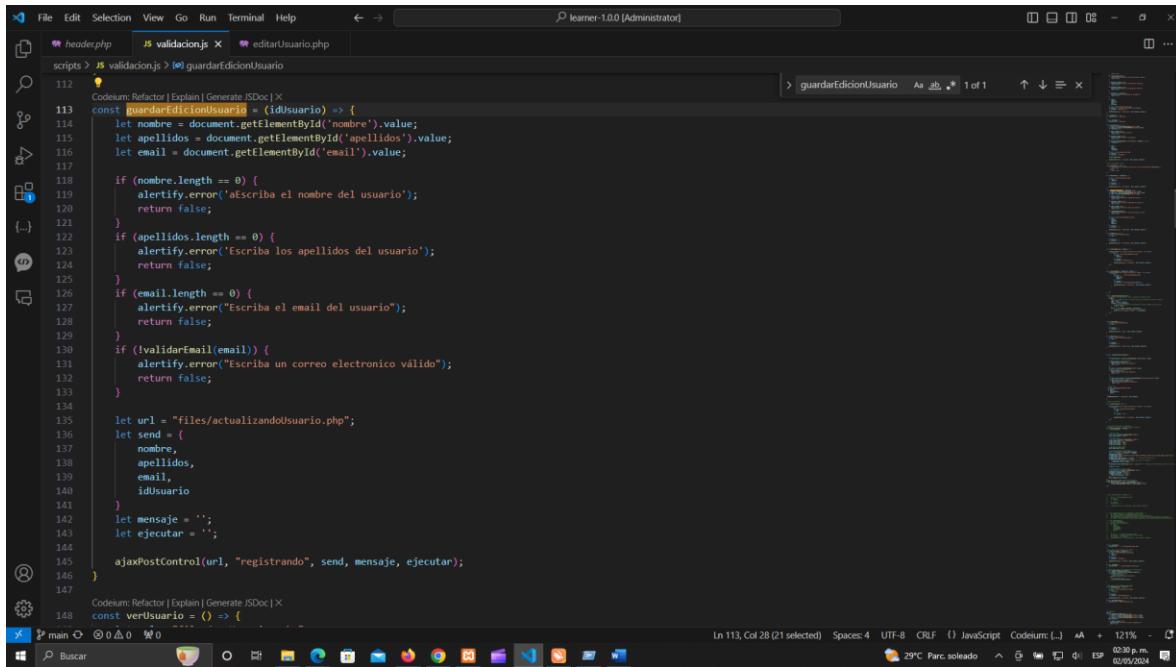
Apellidos: Otro campo que permite ingresar los apellidos del usuario.

Correo Electrónico: Un campo para ingresar la dirección de correo electrónico del usuario.



JavaScript (Edición de Perfil):

Cuando hago clic en el botón "Actualizar Información", la función guardarEdicionUsuario(idUsuario) se encarga de procesar los datos actualizados de mi perfil. Obtengo los nuevos valores de nombre, apellidos y correo electrónico desde los campos del formulario de edición. Además, recibo el parámetro idUsuario, que identifica de manera única mi perfil.



The screenshot shows a code editor window with several tabs at the top: 'header.php', 'JS validacion.js', and 'editarUsuario.php'. The 'JS validacion.js' tab is active, displaying the following JavaScript code:

```
112
113 const guardarEdicionUsuario = (idUsuario) => {
114   let nombre = document.getElementById('nombre').value;
115   let apellidos = document.getElementById('apellidos').value;
116   let email = document.getElementById('email').value;
117
118   if (nombre.length == 0) {
119     alertify.error('Añada el nombre del usuario');
120     return false;
121   }
122   if (apellidos.length == 0) {
123     alertify.error('Añada los apellidos del usuario');
124     return false;
125   }
126   if (email.length == 0) {
127     alertify.error('Añada el email del usuario');
128     return false;
129   }
130   if (!validarEmail(email)) {
131     alertify.error('Añada un correo electrónico válido');
132     return false;
133   }
134
135   let url = "files/actualizandoUsuario.php";
136   let send = {
137     nombre,
138     apellidos,
139     email,
140     idUsuario
141   }
142   let mensaje = '';
143   let ejecutar = '';
144
145   ajaxPostControl(url, "registrando", send, mensaje, ejecutar);
146 }
```

The code uses ES6 syntax like arrow functions and object destructuring. It includes validation for name, surname, and email fields, and a check for a valid email address using a helper function 'validarEmail'. It then constructs a POST request to 'actualizandoUsuario.php' with the user's information.

PHP (Procesamiento de la Edición de Perfil):

En el archivo actualizandoUsuario.php, proceso la solicitud de edición de perfil en el lado del servidor. Recibo los datos enviados desde el formulario HTML utilizando el método POST, incluyendo el idUsuario que me identifica. Luego, preparo una consulta SQL para actualizar mi información en la base de datos.

Después de ejecutar la consulta SQL, verifico si la actualización fue exitosa. Si todo va bien, confirmo la transacción y envío una respuesta JSON indicando que la actualización se realizó correctamente. En caso de algún error, cancelo la transacción y envío un mensaje de error al cliente.

```

1 <?php
2 include("../clases/Conexion.php");
3
4 $db = new Conexion();
5
6 $nombre = htmlentities($_POST['nombre'],ENT_NOQUOTES,"UTF-8");
7 $apellidos = htmlentities($_POST['apellidos'],ENT_NOQUOTES,"UTF-8");
8 $email = $_POST['email'];
9 $idUsuario = $_POST['idUsuario'];
10
11 //indicar que se usarán transacciones
12 $db->beginTransaction();
13
14 //preparar la consulta
15 $sql = $db->prepare("UPDATE dash_usuarios SET nombre = :nombre, apellidos = :apellidos, email = :email WHERE id = :idUsuario LIMIT 1");
16 $sql->bindParam(':nombre', $nombre, PDO::PARAM_STR);
17 $sql->bindParam(':apellidos', $apellidos, PDO::PARAM_STR);
18 $sql->bindParam(':email', $email, PDO::PARAM_STR);
19 $sql->bindParam(':idUsuario', $idUsuario, PDO::PARAM_INT);
20
21 $sql->execute();
22
23 if($sql){
24     $db->commit();
25     $jsondata = array("success" => true,
26                      "mensaje" => "Actualización correcta");
27 } else{
28     $db->rollback();
29     $jsondata = array("success" => false,
30                      "mensaje" => "Ocurrió un error al intentar guardar los datos");
31 }
32
33 header('Content-type: application/json; charset=utf-8');
34 echo json_encode($jsondata);
35 exit();
36
37 ?>

```

Apartado categorías de cursos y Presentaciones cursos:

Categorías de Cursos:

En esta sección, se muestran las diferentes categorías en las que se clasifican los cursos disponibles en el sitio web. Utilizando PHP y SQL, extraigo las categorías principales de la base de datos y las presento como enlaces en la página. Cada enlace lleva a una página dedicada que muestra los cursos específicos pertenecientes a esa categoría. Esto facilita la navegación de los usuarios al permitirles encontrar cursos relacionados agrupados bajo categorías específicas.

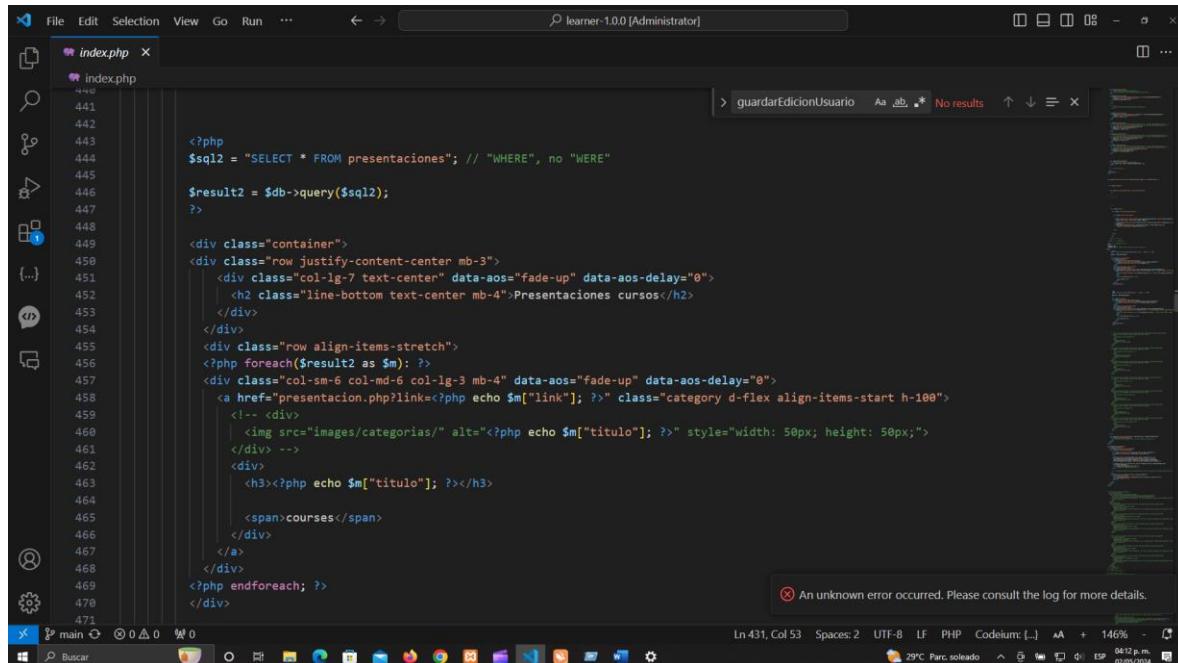
```

1 index.php x
2 index.php
3
404 ivv <-- ./entree_en_hero -->
405 yteo
406 tegeoy img (
407 adding: 3px; /* Puedes ajustar este valor según necesites */
408
409
410 </>php
411 $sql = "SELECT * FROM category WHERE parent = 0"; // "WHERE", no "WERE"
412
413 $result = $db->query($sql);
414
415
416 div class="untree_co-section">
417   div class="container">
418     div class="row justify-content-center mb-3">
419       div class="col-lg-7 text-center" data-aos="fade-up" data-aos-delay="0">
420         h2 class="line-bottom text-center mb-4">Categoría:
421       /div>
422     /div>
423     div class="row align-items-stretch">
424       <?php foreach($result as $e): ?>
425       div class="col-sm-6 col-md-6 col-lg-3 mb-4" data-aos="fade-up" data-aos-delay="0">
426         a href="CURSOS_elegidosporCategoria.php?id=$e[id]"; ?> . class="category d-flex align-items-start h-100">
427           /div>
428           img src="images/categorias/";?> echo $e["thumbnail"]; ?>" alt="<?php echo $e["thumbnail"]; ?>" alt="<?php echo $e["name"]; ?>" style="width: 50px; height: 50px;
429
430         /div>
431         ch3><?php echo $e["name"]; ?>/ch3>
432         span>courses</span>
433       /div>
434     </>div
435   </>div
436   </>div
437   </>div
438   </>div
439   </>div
440   </>div
441   </>div
442
443 ?>
444 <?php endforeach; ?>

```

Presentaciones de Cursos:

Aquí se muestran las presentaciones disponibles de los cursos. Utilizando PHP y SQL, recupero todas las presentaciones de cursos almacenadas en la base de datos y las muestro como enlaces en la página. Cada enlace conduce a la presentación específica de un curso. Esta sección proporciona a los usuarios acceso rápido a las presentaciones de cursos disponibles, lo que les permite explorar y acceder fácilmente a la información sobre los cursos ofrecidos.



The screenshot shows a code editor window with the file 'index.php' open. The code is written in PHP and uses Bootstrap CSS classes. It queries a database for all presentations and loops through the results to display them as cards. Each card contains a thumbnail image, the course title, and a brief description. A message at the bottom right indicates an unknown error occurred.

```
<?php
$sql2 = "SELECT * FROM presentaciones"; // "WHERE", no "WERE"
$result2 = $db->query($sql2);
?>

<div class="container">
<div class="row justify-content-center mb-3">
    <div class="col-lg-7 text-center" data-aos="fade-up" data-aos-delay="0">
        <h2 class="line-bottom text-center mb-4">Presentaciones cursos</h2>
    </div>
</div>
<div class="row align-items-stretch">
<?php foreach($result2 as $m): ?>
<div class="col-sm-6 col-md-6 col-lg-3 mb-4" data-aos="fade-up" data-aos-delay="0">
    <a href="presentacion.php?link=<?php echo $m["link"]; ?>" class="category d-flex align-items-start h-100">
        <!-- <div>
            " style="width: 50px; height: 50px;">
        </div> -->
        <div>
            <h3><?php echo $m["titulo"];></h3>
            <span>courses</span>
        </div>
        <a>
            <!-- <div>
                " style="width: 50px; height: 50px;">
            </div> -->
            <?php endforeach; ?>
        </a>
    </div>
</div>
```

Todos los cursos:

Esta sección muestra todos los cursos disponibles en el sitio web. Utilizando PHP y SQL, se recuperan los detalles de los cursos almacenados en la base de datos y se muestran como tarjetas de curso en la página. Cada tarjeta incluye una imagen representativa del curso, su título y una breve descripción. Además, se proporciona información sobre el nivel de habilidad requerido para el curso (de principiante a avanzado) y la duración estimada del curso en horas y minutos.

Los cursos se presentan en un diseño de cuadrícula de tres columnas para facilitar la visualización y la navegación. Cada tarjeta de curso es interactiva y se puede hacer clic para acceder a más detalles sobre el curso en una página dedicada.

The screenshot shows a code editor window titled "learner-1.0.0 [Administrator]". The file being edited is "CURSOS_todos.php". The code is a PHP script that generates a grid of course cards. Each card has a thumbnail image, a title, a description, and footer information including a skill level and duration. The code uses CSS classes like "cc_course-card" and "grid_thirds" to layout the cards. A search bar at the top right shows the query "guardarEdicionUsuario". The bottom status bar shows the line number (Ln 482), column (Col 67), and encoding (UTF-8). The taskbar at the bottom includes icons for File, Edit, Selection, View, Go, Run, and others.

```
File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
CURSOS_todos.php > guardarEdicionUsuario Aa ab * No results ↑ ↓ ⌂ ...
CURSOS_todos.php
466 iv class="course-description" s>
467 :div id="w-node-ad03d18d-b036-a9fd-f585-e0398c17c639-96b988bd">
468
469 <div class="w-dyn-list">
470 <!-- C:\xampp\htdocs\elearning\learner-1.0.0\CURSOS_descripcion.php -->
471 <div role="list" class="grid_thirds w-dyn-items" style="display: flex; flex-wrap: wrap;">
472 <?php foreach($result as $item); ?>
473 <div href="CURSOS_descripcion.php?id=<?php echo $item['id']; ?>" class="cc_course-card w-inline-block">
474 <div style="border: 2px solid #ccc; border-radius: 8px; overflow: hidden; color: #000; background-color: rgba(255, 255, 255, 0.8);>
475 <div style="background-image: url('images/categorias/<?php echo $item['thumbnail']; ?>'); height: 200px; background-size: cover; background-position: center; border-bottom: 1px solid #ccc; border-radius: 8px 8px 0 0; width: calc(33.33% - 20px); margin-right: 20px; ">
476 <div class="cc_course-card-inner" style="padding: 20px;>
477 <h3 class="h4 u-mb-0"><?php echo $item['name']; ?></h3>
478 <p class="cc_course-card-description u-fc-2 text-style-2lines">Learn the entire design process from idea to final output as we tal<?php endforeach; ?>
479 <div class="cc_course-card-footer" style="border-top: 1px solid #ccc; padding: 20px;>
480 <div class="cc_course-card-footer-item skill">
481 <p class="text-xs u-fc-2 u-mb-0">Beginner - Advanced</p>
482 </div>
483 <div class="cc_course-card-footer-item length">
484 <p class="text-xs u-fc-2 u-mb-0">6h 27m</p>
485 </div>
486 <div class="cc_course-card-footer-item arrow u-bgc-2">
487 <div class="icon-1-25 w-embed"><svg width="100%" height="100%" viewBox="0 0 14 12" fill="none" xmlns="#"><path d="M8.25 0.5L7.41 1.25L6.56 0.5L8.25 0z" /></svg></div>
488 </div>
489 </div>
490 </div>
491 </div>
492 </div>
493 </a>
494 </div>
495 <?php endforeach; ?>
496 <!-- Repite lo mismo para los otros cursos -->
```

El objetivo de esta sección es brindar a los usuarios una visión general de todos los cursos disponibles y permitirles explorar fácilmente los cursos que les interesen. Mediante el uso de tarjetas de curso visualmente atractivas y descriptivas, se pretende captar el interés del usuario y fomentar la participación en el aprendizaje.

Descripción de los cursos:

Vista previa del curso: Se muestra una imagen representativa del curso junto con un botón de reproducción que permite a los usuarios ver una vista previa del contenido del curso.

Precio: Indica si el curso es gratuito o tiene un costo. En este caso, se muestra "Gratis".

En este caso, se proporciona un botón "Comenzar" que redirige a los usuarios a la página de reproducción de videos del curso.

Incluye: Detalla los elementos incluidos en el curso, como videos bajo demanda, lecciones, acceso de por vida y acceso en dispositivos móviles y computadoras portátiles.

```
425 <nav id="menu-derecho">
426   <div class="course-sidebar natural">
427     <div class="preview-video-box">
428       <a data-toggle="modal" data-target="#CoursePreviewModal">
429         
430         <span class="preview-text">Vista previa del curso</span>
431         <span class="play-btn"></span>
432       </a>
433     </div>
434   </div>
435   <div class="course-sidebar-text-box">
436     <div class="price">
437       <span class="current-price">Gratis</span>
438       <span class="original-price"></span>
439       <input type="hidden" id="total_price_of_checking_out" value="$99.99">
440     </div>
441     <div class="buy-btns">
442       <!-- <a href="#" class="btn btn-buy-now" onclick="handleEnrolledButton()">Comenzar</a> -->
443       <a href="CURSO_videoplayer.php?id=<?php echo $idCurso; ?>" class="btn btn-buy-now" onclick="">Comenzar</a>
444     </div>
445   </div>
446   <div class="includes">
447     <div class="title"><b>Incluye:</b></div>
448     <ul>
449       <li><i class="far fa-file-video"></i> videos bajo demanda</li>
450       <li><i class="far fa-file"></i> Lecciones</li>
451       <li><i class="far fa-compass"></i> Acceso de por vida completo</li>
452       <li><i class="fas fa-mobile-alt"></i> Acceso en Pc y laptops</li>
453     </ul>
454   </div>
455 </div>
```

Temarios, Lo que obtienes, Requisitos, Descripción: Estas son pestañas que contienen información detallada sobre diferentes aspectos del curso:

Temarios: Proporciona una lista de temas cubiertos en el curso, organizados en secciones y subsecciones. Cada tema puede tener una descripción asociada.

Lo que obtienes: Enumera los materiales y beneficios que los estudiantes recibirán al inscribirse en el curso.

Requisitos: Enumera los requisitos previos que los estudiantes deben cumplir antes de inscribirse en el curso. En este caso, la información se extrae de la base de datos.

Descripción: Ofrece una descripción detallada del curso, que puede incluir información sobre los temas cubiertos, el enfoque del curso y los objetivos de aprendizaje.

A screenshot of a code editor window titled "learner-1.0.0 [Administrator]". The file being edited is "CURSOS_descripcion.php". The code includes a video player with PHP logic to embed a video and a script to handle video playback controls. The editor has a dark theme with syntax highlighting for PHP and HTML.

```
799 <div class="container">
800     <div class="video-player">
801         <video id="myVideo" controls autoplay muted playsinline>
802             <source src="videos\<?php echo $video_url; ?>" type="video/mp4">
803             Tu navegador no admite la etiqueta de video.
804         </video>
805     <div class="">
806         <div class="overlay-content">
807             <h1 style="color: black;"><?php echo $nameValue; ?></h1>
808             <p style="color: black;"><?php echo $short_description4; ?></p>
809         </div>
810     </div>
811     <script>
812         var video = document.getElementById('myVideo');
813
814         // Cuando el video comienza a reproducirse, quita el atributo 'muted'
815         video.addEventListener('play', function() {
816             video.removeAttribute('muted');
817         });
818     </script>
819     <!--
820     ...
821     -->
822     <!-- -->
823     <!-- -->
824
825
826
827
828
829
```

A screenshot of a code editor window titled "learner-1.0.0 [Administrator]". The file being edited is "CURSOS_descripcion.php". The code includes a tabbed navigation system using Bootstrap's tabs component. It defines tabs for "Temarios", "Lo que obtienes", "Requisitos", and "Descripción", each with its own href and tab content. The editor has a dark theme with syntax highlighting for PHP and HTML.

```
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670     role="tablist">
671
672     <a href="#Temarios" role="tab" aria-controls="Temarios" aria-selected="true">Temarios</a>
673
674     <a href="#lo-que-obtienes" role="tab" aria-controls="lo-que-obtienes" aria-selected="false">Lo que obtienes</a>
675
676     <a href="#requisitos" role="tab" aria-controls="requisitos" aria-selected="false">Requisitos</a>
677
678     <a href="#descripcion" role="tab" aria-controls="descripcion" aria-selected="false">Descripción</a>
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
```

```
<?php
$sql = "SELECT * FROM course_links WHERE category_id = " . $idCurso . " GROUP BY section ORDER BY section";
$result = $db->query($sql);
?>

<?php foreach ($result as $m) :
$tema = $m['section'];
?>

    // Consulta SQL para obtener los subtemas del tema actual
$subtemas_sql = "SELECT * FROM course_links WHERE category_id = " . $idCurso . " AND section = '$tema'";
$subtemas_result = $db->query($subtemas_sql);
?>

<div class="add-courses-section2">
<h4 class="mb-3 header-title">
| TEMA <a href="#" onclick="mostrarSubtemas('<?php echo $tema; ?>')><?php echo $tema; ?></a>
</h4>

<?php if ($subtemas_result && $subtemas_result->rowCount() > 0) : ?>
<div class="table-responsive-sm mt-4">
    <table id="course-datable" class="table table-striped dt-responsive nowrap" width="100%" data-page-length='25'>
        <thead>
            <tr>
                <th>#</th>
                <th>Título</th>
                <th>Subcategoría</th>
                <th>Descripción</th>
                <!-- <th>Acciones</th> -->
            </tr>
        </thead>
        <?php foreach ($subtemas_result as $subtemas) : ?>
<tbody class="add-courses-section3">
```

```
<!-- <th>Acciones</th> -->
</tr>
</thead>
<?php foreach ($subtemas_result as $subtemas) : ?>
<tbody class="add-courses-section3">
    <tr>
        <td></td>
        <td>
            <strong><a href="#"><?php echo $subtemas['title']; ?></a></strong><br>
            <small class="text-white">
                <!-- Instructor: <b>Nombre del Instructor Ejemplo</b> -->
            </small>
        </td>
        <?php
            // Consulta SQL para obtener la categoría del curso actual
            $subCategory = $subtemas['category_id'];
            $sqlc = "SELECT * FROM category WHERE id = $subCategory";
            $tiulocurso = $db->query($sqlc);

            // Mostrar la categoría del curso actual
            foreach ($tiulocurso as $cat) :
        ?>
        <td>
            <span class="badge badge-dark-lighten">
                <?php echo $cat['name']; ?>
            </span>
        </td>
        <td>
            <?php echo htmlspecialchars_decode($subtemas['description']); ?>
        </td>
    </tr>
</tbody>
```

A screenshot of a code editor window titled "learner-1.0.0 [Administrator]". The file being edited is "CURSOS_descripcion.php". The code is a PHP script with some CSS classes and HTML structures. The editor has a dark theme with syntax highlighting. The status bar at the bottom shows "Ln 446, Col 29" and "146%".

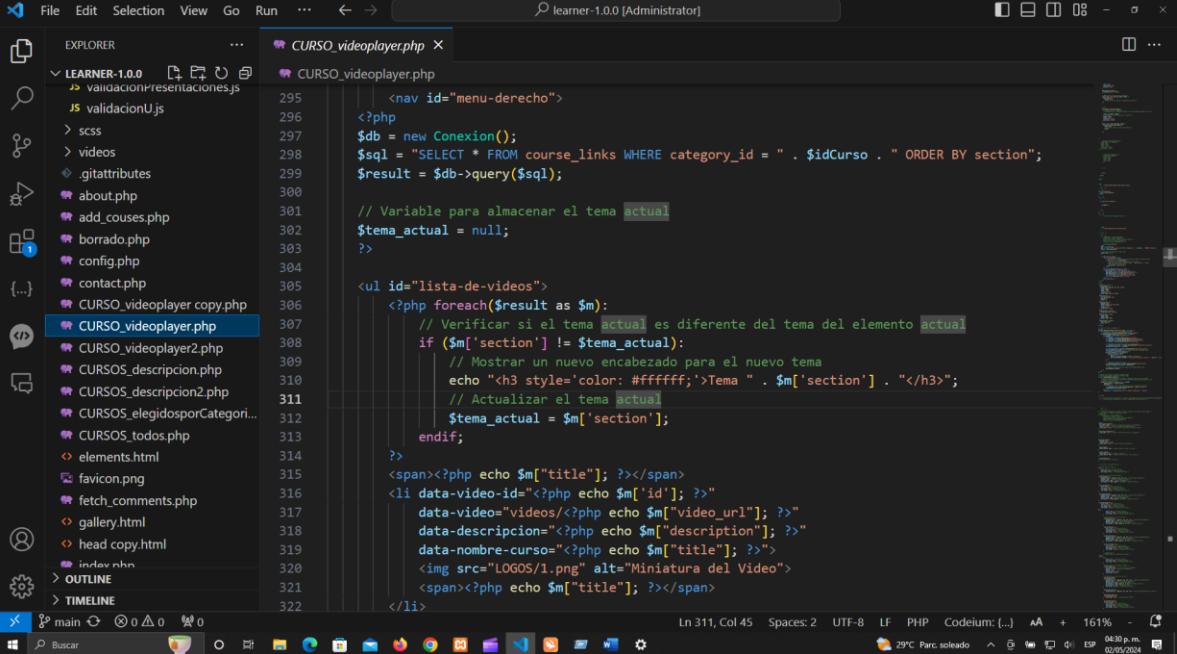
```
File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
CURSOS_descripcion.php
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979 <!-- AQUI -->
Ln 446, Col 29 Spaces: 2 UTF-8 LF PHP Codeium: [...] xA + 146% - 08 ...
main 0 0 0 % 0 Buscar 29°C Parc soleado 04:24 p.m. 02/05/2024
```

A screenshot of a code editor window titled "learner-1.0.0 [Administrator]". The file being edited is "CURSOS_descripcion.php". The code is a PHP script with some CSS classes and HTML structures. The editor has a dark theme with syntax highlighting. The status bar at the bottom shows "Ln 446, Col 29" and "146%".

```
File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
CURSOS_descripcion.php
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
Ln 446, Col 29 Spaces: 2 UTF-8 LF PHP Codeium: [...] xA + 146% - 08 ...
main 0 0 0 % 0 Buscar 29°C Parc soleado 04:24 p.m. 02/05/2024
```

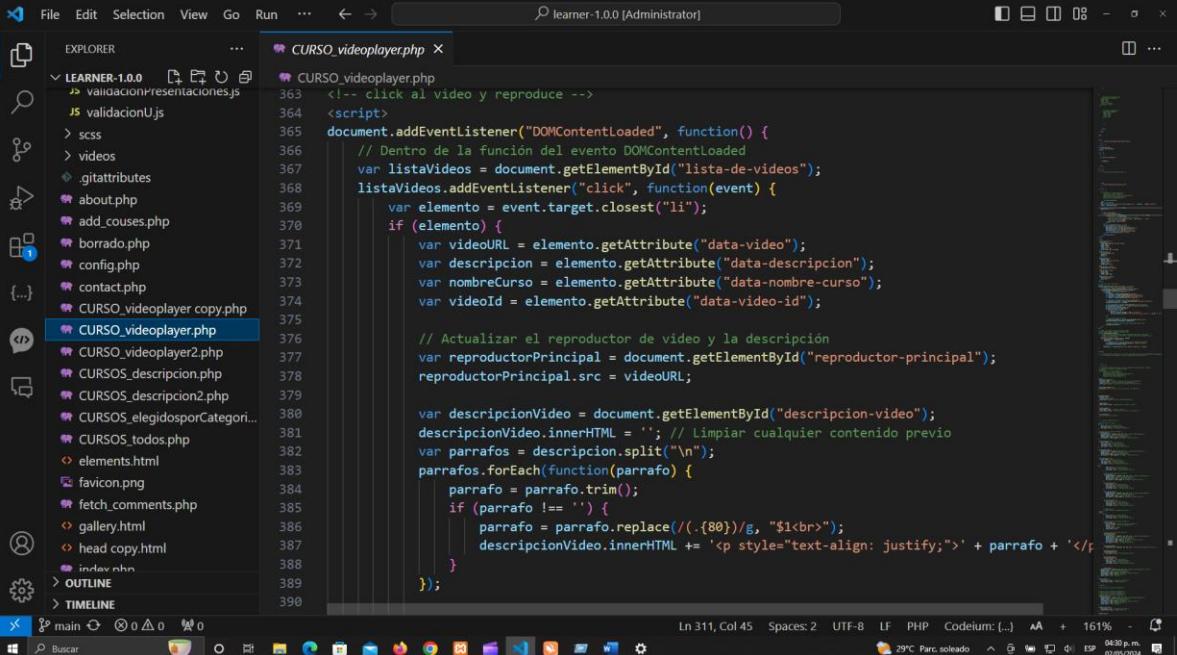
Video reproductor curso:

Consulta a la base de datos: La primera parte del código ejecuta una consulta SQL para obtener los enlaces de los videos del curso, ordenados por sección. Luego, itera sobre los resultados y muestra los videos en una lista.



```
File Edit Selection View Go Run ... 🔍 learner-1.0.0 [Administrator] ⚡ ...  
EXPLORER ...  
LEARNER-1.0.0  
↳ validacionPresentaciones.js  
↳ validationUjs.js  
> scss  
> videos  
↳ .gitattributes  
↳ about.php  
↳ add_couses.php  
↳ borrado.php  
↳ config.php  
↳ contact.php  
↳ CURSO_videoplayer copy.php  
↳ CURSO_videoplayer.php  
↳ CURSO_videoplayer2.php  
↳ CURSOS_descripcion.php  
↳ CURSOS_descripcion2.php  
↳ CURSOS_elegidosporCategoria.php  
↳ CURSOS.todos.php  
↳ elements.html  
↳ favicon.png  
↳ fetch_comments.php  
↳ gallery.html  
↳ head_copy.html  
↳ index.php  
> OUTLINE  
> TIMELINE  
CURSO_videoplayer.php x  
CURSO_videoplayer.php  
295     <nav id="menu-derecho">  
296     <?php  
297     $db = new Conexion();  
298     $sql = "SELECT * FROM course_links WHERE category_id = " . $idCurso . " ORDER BY section";  
299     $result = $db->query($sql);  
300  
301     // Variable para almacenar el tema actual  
302     $tema_actual = null;  
303     ?>  
304  
305     <ul id="lista-de-videos">  
306         <?php foreach($result as $m):  
307             // Verificar si el tema actual es diferente del tema del elemento actual  
308             if ($m['section'] != $tema_actual):  
309                 // Mostrar un nuevo encabezado para el nuevo tema  
310                 echo "<h3 style='color: #ffff00;'>Tema " . $m['section'] . "</h3>";  
311                 // Actualizar el tema actual  
312                 $tema_actual = $m['section'];  
313             endif;  
314         ?>  
315         <span><?php echo $m["title"];></span>  
316         <li data-video-id=<?php echo $m['id']; ?>"  
317             data-video="videos/<?php echo $m["video_url"];>"  
318             data-descripcion=<?php echo $m["description"];>"  
319             data-nombre-curso=<?php echo $m["title"];>"  
320               
321             <span><?php echo $m["title"];></span>  
322         </li>  
Ln 311, Col 45 Spaces: 2 UTF-8 LF PHP Codeium: (...) AA + 161% - 29°C Parc. soleado 04:30 p.m. ESP 02/05/2024
```

Reproductor de video: Cuando un usuario hace clic en un video de la lista, se actualiza el reproductor principal de video con la URL del video seleccionado. También se actualiza la descripción del video y el nombre del curso.



```
File Edit Selection View Go Run ... 🔍 learner-1.0.0 [Administrator] ⚡ ...  
EXPLORER ...  
LEARNER-1.0.0  
↳ validacionPresentaciones.js  
↳ validationUjs.js  
> scss  
> videos  
↳ .gitattributes  
↳ about.php  
↳ add_couses.php  
↳ borrado.php  
↳ config.php  
↳ contact.php  
↳ CURSO_videoplayer copy.php  
↳ CURSO_videoplayer.php  
↳ CURSO_videoplayer2.php  
↳ CURSOS_descripcion.php  
↳ CURSOS_descripcion2.php  
↳ CURSOS_elegidosporCategoria.php  
↳ CURSOS.todos.php  
↳ elements.html  
↳ favicon.png  
↳ fetch_comments.php  
↳ gallery.html  
↳ head_copy.html  
↳ index.php  
> OUTLINE  
> TIMELINE  
CURSO_videoplayer.php x  
CURSO_videoplayer.php  
363     <!-- click al video y reproduce -->  
364     <script>  
365     document.addEventListener("DOMContentLoaded", function() {  
366         // Dentro de la función del evento DOMContentLoaded  
367         var listaVideos = document.getElementById("lista-de-videos");  
368         listaVideos.addEventListener("click", function(event) {  
369             var elemento = event.target.closest("li");  
370             if (elemento) {  
371                 var videoURL = elemento.getAttribute("data-video");  
372                 var descripcion = elemento.getAttribute("data-descripcion");  
373                 var nombreCurso = elemento.getAttribute("data-nombre-curso");  
374                 var videoId = elemento.getAttribute("data-video-id");  
375  
376                 // Actualizar el reproductor de video y la descripción  
377                 var reproductorPrincipal = document.getElementById("reproductor-principal");  
378                 reproductorPrincipal.src = videoURL;  
379  
380                 var descripcionVideo = document.getElementById("descripcion-video");  
381                 descripcionVideo.innerHTML = ''; // Limpiar cualquier contenido previo  
382                 var parrafos = descripcion.split("\n");  
383                 parrafos.forEach(function(parrafo) {  
384                     parrafo = parrafo.trim();  
385                     if (parrafo !== '') {  
386                         parrafo = parrafo.replace(/(\.{80})\//g, "$1<br>");  
387                         descripcionVideo.innerHTML += '<p style="text-align: justify;">' + parrafo + '</p>';  
388                     }  
389                 });  
Ln 311, Col 45 Spaces: 2 UTF-8 LF PHP Codeium: (...) AA + 161% - 29°C Parc. soleado 04:30 p.m. ESP 02/05/2024
```

```

395     // Obtener comentarios para el video seleccionado mediante AJAX
396     fetchComments(videoId);
397   }
398 }
399 });
400
401 // Función para obtener comentarios mediante AJAX
402 // Función para obtener comentarios mediante AJAX
403 function fetchComments(videoId) {
404   // Realiza una solicitud AJAX para obtener comentarios del video seleccionado
405   // Puedes usar XMLHttpRequest, Fetch API o cualquier biblioteca como jQuery o Axios
406   // Ejemplo usando Fetch API
407   fetch('fetch_comments.php?video_id=' + videoId)
408     .then(response => response.text())
409     .then(data => {
410       // Actualiza la sección de comentarios con los comentarios obtenidos
411       var commentsSection = document.querySelector('.comments-section');
412       commentsSection.innerHTML = data;
413     })
414     .catch(error => console.error('Error al obtener comentarios:', error));
415 }
416
417 </script>
418
419 <!-- <iframe width="773" height="321" src="https://www.mediafire.com/file/llzpon69sy73441/videoplayer.html" -->
420 <!-- Consulta SQL para obtener la ruta del video desde la base de datos -->
421

```

Obtención de comentarios mediante AJAX: Se define una función fetchComments que realiza una solicitud AJAX para obtener los comentarios asociados al video seleccionado. Estos comentarios se muestran en la sección de comentarios después de obtenerse.

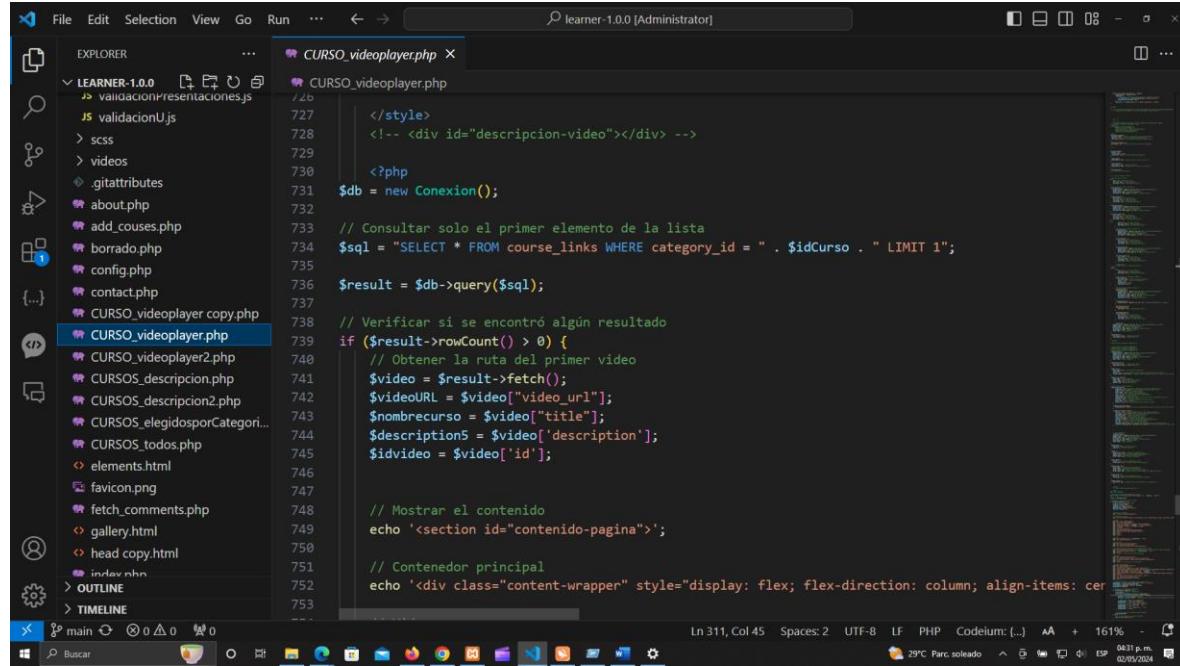
```

803
804     // Consulta para obtener los comentarios del primer video
805     $sqlComments = "SELECT * FROM comment WHERE video_id = :video_id";
806     $stmtComments = $db->prepare($sqlComments);
807     $stmtComments->bindParam(':video_id', $idvideo);
808     $stmtComments->execute();
809
810     // Verificar si se encontraron comentarios
811     if ($stmtComments->rowCount() > 0) {
812       // Construir el HTML de los comentarios
813       $commentsHTML = '';
814       while ($row = $stmtComments->fetch(PDO::FETCH_ASSOC)) {
815         // Obtener el nombre de usuario del comentario actual
816         $usuario = new Usuario($row['user_id']);
817         $username = $usuario->nombre; // Suponiendo que el nombre de usuario se almacena en la propiedad 'nombre'
818
819         // Agregar cada comentario a $commentsHTML
820         $commentsHTML .= '<div class="comment">';
821         $commentsHTML .= '<div class="comment-header">';
822         // Mostrar el nombre de usuario en el comentario
823         $commentsHTML .= '<p>' . $username . '</p>';
824         $commentsHTML .= '</div>';
825         $commentsHTML .= '<p>' . $row['body'] . '</p>';
826         $commentsHTML .= '</div>';
827       }
828
829       // Enviar el HTML de los comentarios como respuesta
830       echo $commentsHTML;
831     } else {
832       // Si no hay comentarios, mostrar un mensaje
833       echo 'No hay comentarios aún.';
834     }
835
836     echo '</div>'; // Cierre del div de .comments-section
837

```

Sección de comentarios: La sección de comentarios muestra los comentarios asociados al video seleccionado, así como un formulario para que los usuarios puedan agregar nuevos comentarios. Los comentarios se almacenan en la base de datos y se muestran dinámicamente.

Reproductor de video principal: Al inicio, se muestra el primer video del curso en el reproductor principal, junto con su descripción y comentarios asociados.



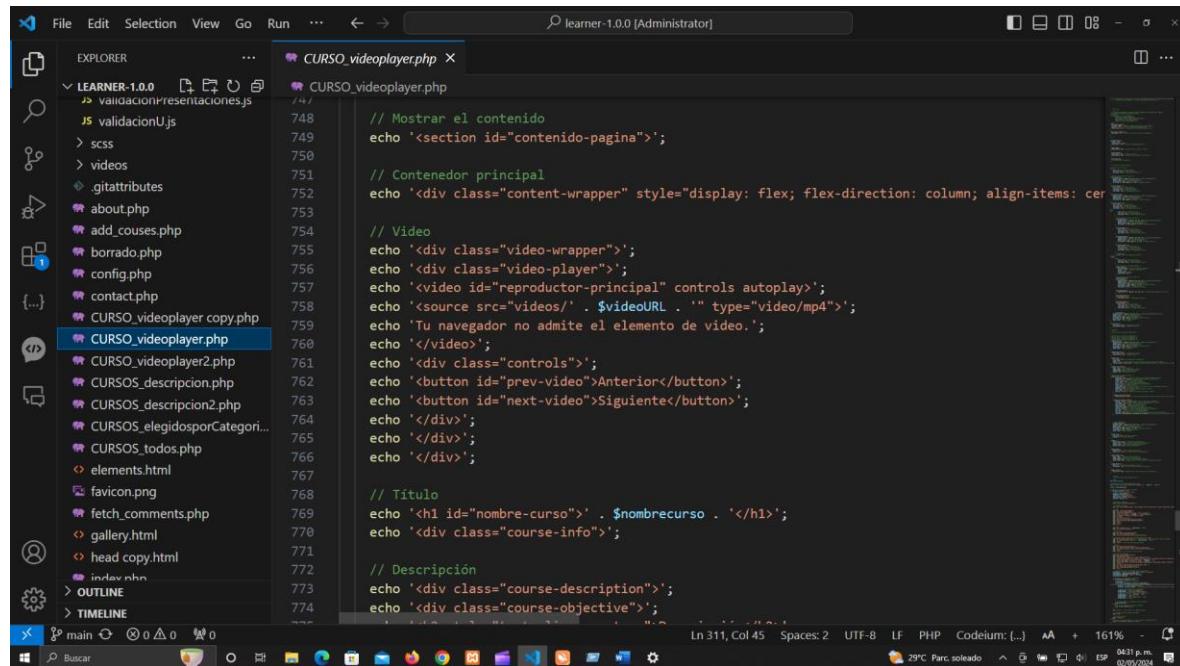
```
File Edit Selection View Go Run ... < > learner-1.0.0 [Administrator]
EXPLORER CURSO_videoplayer.php ...
LEARNER-1.0.0
> validarion/representaciones.js
> SCSS
> videos
> .gitattributes
> about.php
> add_couses.php
> borrado.php
> config.php
> contact.php
> CURSO_videoplayer copy.php
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
> OUTLINE
> TIMELINE
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
main 0 △ 0 0
Buscar
File Edit Selection View Go Run ... < > learner-1.0.0 [Administrator]
CURSO_videoplayer.php ...
LEARNER-1.0.0
> validarion/representaciones.js
> SCSS
> videos
> .gitattributes
> about.php
> add_couses.php
> borrado.php
> config.php
> contact.php
> CURSO_videoplayer copy.php
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
> OUTLINE
> TIMELINE
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
main 0 △ 0 0
Buscar
```

File Edit Selection View Go Run ... < > learner-1.0.0 [Administrator]

CURSO_videoplayer.php X

```
726 // Consultar solo el primer elemento de la lista
727 $sql = "SELECT * FROM course_links WHERE category_id = " . $idCurso . " LIMIT 1";
728
729 // Verificar si se encontró algún resultado
730 if ($result->rowCount() > 0) {
731     // Obtener la ruta del primer video
732     $video = $result->fetch();
733     $videoURL = $video["video_url"];
734     $nombrecurso = $video["title"];
735     $descripcion5 = $video['description'];
736     $idvideo = $video['id'];
737
738     // Mostrar el contenido
739     echo '<section id="contenido-pagina">';
740
741     // Contenedor principal
742     echo '<div class="content-wrapper" style="display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh;">';
743
744     // Video
745     echo '<div class="video-wrapper">';
746     echo '<div class="video-player">';
747     echo '<video id="reproductor-principal" controls autoplay>';
748     echo '<source src="videos/' . $videoURL . '" type="video/mp4">';
749     echo '</video>';
750     echo '<div class="controls">';
751     echo '<button id="prev-video">Anterior</button>';
752     echo '<button id="next-video">Siguiente</button>';
753     echo '</div>';
754     echo '</div>';
755
756     // Titulo
757     echo '<h1 id="nombre-curso">' . $nombrecurso . '</h1>';
758     echo '<div class="course-info">';
759
760     // Descripción
761     echo '<div class="course-description">';
762     echo '<div class="course-objective">';
```

Ln 311, Col 45 Spaces: 2 UTF-8 LF PHP Codeium: [...] AA + 161% - 04:31 p.m. 29°C Parc soleado 02/05/2024



```
File Edit Selection View Go Run ... < > learner-1.0.0 [Administrator]
EXPLORER CURSO_videoplayer.php ...
LEARNER-1.0.0
> validarion/representaciones.js
> SCSS
> videos
> .gitattributes
> about.php
> add_couses.php
> borrado.php
> config.php
> contact.php
> CURSO_videoplayer copy.php
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
> OUTLINE
> TIMELINE
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
main 0 △ 0 0
Buscar
File Edit Selection View Go Run ... < > learner-1.0.0 [Administrator]
CURSO_videoplayer.php ...
LEARNER-1.0.0
> validarion/representaciones.js
> SCSS
> videos
> .gitattributes
> about.php
> add_couses.php
> borrado.php
> config.php
> contact.php
> CURSO_videoplayer copy.php
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
> OUTLINE
> TIMELINE
CURSO_videoplayer.php
CURSO_videoplayer2.php
CURSOS_descripcion.php
CURSOS_descripcion2.php
CURSOS_elegidosporCategori...
CURSOS_todos.php
elements.html
favicon.png
fetch_comments.php
gallery.html
head copy.html
index.php
main 0 △ 0 0
Buscar
```

File Edit Selection View Go Run ... < > learner-1.0.0 [Administrator]

CURSO_videoplayer.php X

```
748 // Mostrar el contenido
749 echo '<section id="contenido-pagina">';
750
751 // Contenedor principal
752 echo '<div class="content-wrapper" style="display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh;">';
753
754 // Video
755 echo '<div class="video-wrapper">';
756 echo '<div class="video-player">';
757 echo '<video id="reproductor-principal" controls autoplay>';
758 echo '<source src="videos/' . $videoURL . '" type="video/mp4">';
759 echo '</video>';
760 echo '<div class="controls">';
761 echo '<button id="prev-video">Anterior</button>';
762 echo '<button id="next-video">Siguiente</button>';
763 echo '</div>';
764 echo '</div>';
765 echo '</div>';
766
767 // Titulo
768 echo '<h1 id="nombre-curso">' . $nombrecurso . '</h1>';
769 echo '<div class="course-info">';
770
771 // Descripción
772 echo '<div class="course-description">';
773 echo '<div class="course-objective">';
```

Ln 311, Col 45 Spaces: 2 UTF-8 LF PHP Codeium: [...] AA + 161% - 04:31 p.m. 29°C Parc soleado 02/05/2024

File Edit Selection View Go Run ...

CURSO_videoplayer.php

```

    echo '<div>';
    // Título
    echo '<h1 id="nombre-curso">' . $nombrecurso . '</h1>';
    echo '<div class="course-info">';

    // Descripción
    echo '<div class="course-description">';
    echo '<div class="course-objective">';
    echo '<h2 style="text-align: center;">Descripción</h2>';

    // Contenedor donde se mostrará la nueva descripción del video
    echo '<div id="descripcion-video">'; // Contenedor para la descripción del video seleccionado
    echo '<p style="text-align: justify;">' . $description5 . '</p>';
    echo '</div>'; // Cierre del div de la descripción

    // Recursos
    echo '<div class="course-objective">';
    echo '<h2 style="text-align: center;">Tomar en cuenta</h2>';
    echo '<p style="text-align: center;">Mas información relevante</p>';
    echo '</div>'; // Cierre del div del curso-info

    // Sección de comentarios
    echo '<div class="comments-section">';
    echo '<h2>Comentarios:</h2>';
    echo '<form action="" method="post" class="comments-form">';

```

Ln 311, Col 45 Spaces:2 UTF-8 LF PHP Codeium: (...) AA + 161% -

29°C Parc soleado 04:32 p.m. ESP 02/05/2024

File Edit Selection View Go Run ...

CURSO_videoplayer.php

```

    echo '</div>';
    echo '</div>'; // Cierre del div del curso-info

    // Sección de comentarios
    echo '<div class="comments-section">';
    echo '<h2>Comentarios:</h2>';
    echo '<form action="" method="post" class="comments-form">';
    echo '<textarea name="comment" id="comment" cols="30" rows="10" placeholder="Escribe tu comentario"' . $sqlComments;
    echo '<input type="hidden" id="user_id" value="' . $idUsuario . '"';
    echo '<input type="hidden" id="video_id" value="' . $idvideo . '"';
    echo '<button type="button" onclick="validarRegistroComentario()" name="submit-comment"></button>';
    echo '</form>';
    echo '<hr class="sidebar-divider" style="display: none;">';
    echo '<div class="row" id="anadirCursos">';
    echo '<div id="guardandoComentario"></div>';
    echo '</div>';

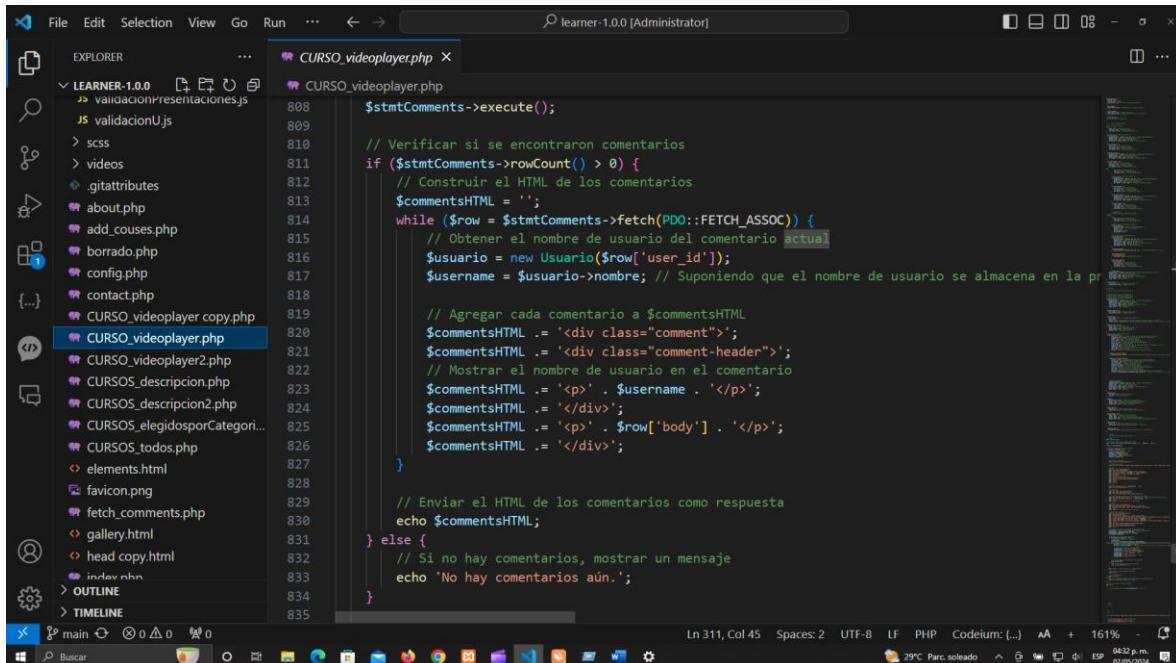
    // Consulta para obtener los comentarios del primer video
    $sqlComments = "SELECT * FROM comment WHERE video_id = :video_id";
    $stmtComments = $db->prepare($sqlComments);
    $stmtComments->bindParam(':video_id', $idvideo);
    $stmtComments->execute();

    // Verificar si se encontraron comentarios
    if ($stmtComments->rowCount() > 0) {
        // Construir el HTML de los comentarios
        $commentsHTML = '';
        while ($row = $stmtComments->fetch(PDO::FETCH_ASSOC)) {

```

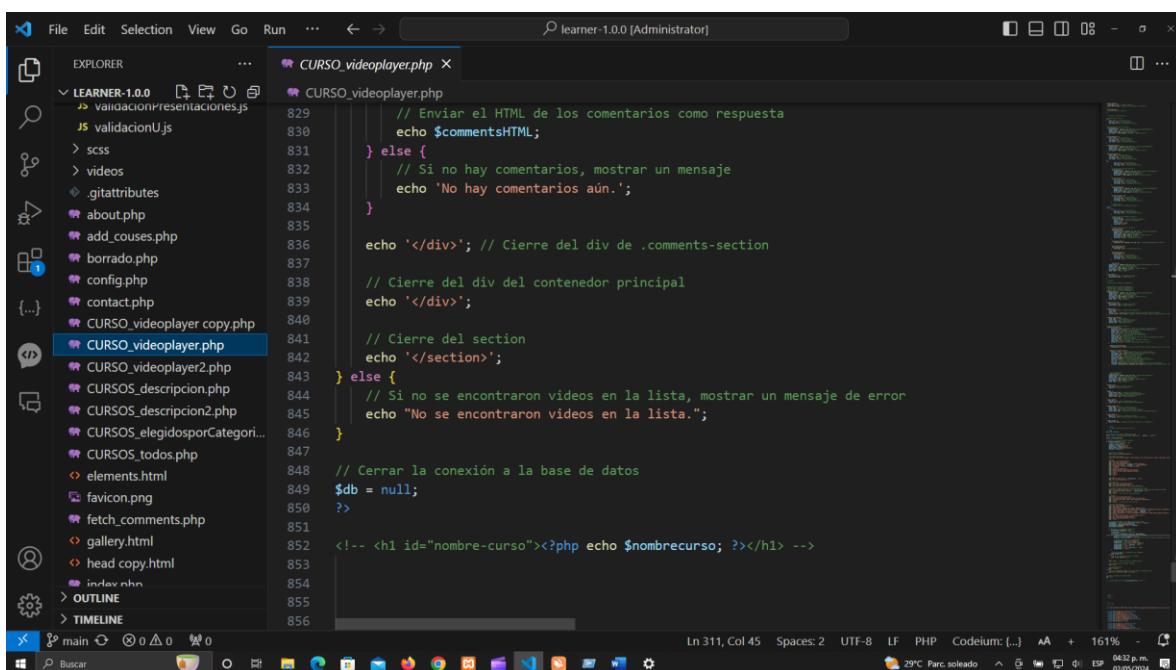
Ln 311, Col 45 Spaces:2 UTF-8 LF PHP Codeium: (...) AA + 161% -

29°C Parc soleado 04:32 p.m. ESP 02/05/2024



```
File Edit Selection View Go Run ... 🔍 learner-1.0.0 [Administrator] ⚡ ... EXPLORER CURSO_videoplayer.php ... CURSO_videoplayer.php 808     $stmtComments->execute(); 809 810     // Verificar si se encontraron comentarios 811     if ($stmtComments->rowCount() > 0) { 812         // Construir el HTML de los comentarios 813         $commentsHTML = ''; 814         while ($row = $stmtComments->fetch(PDO::FETCH_ASSOC)) { 815             // Obtener el nombre de usuario del comentario actual 816             $usuario = new Usuario($row['user_id']); 817             $username = $usuario->nombre; // Suponiendo que el nombre de usuario se almacena en la p 818 819             // Agregar cada comentario a $commentsHTML 820             $commentsHTML .= '<div class="comment">'; 821             $commentsHTML .= '<div class="comment-header">'; 822             // Mostrar el nombre de usuario en el comentario 823             $commentsHTML .= '<p>' . $username . '</p>'; 824             $commentsHTML .= '</div>'; 825             $commentsHTML .= '<p>' . $row['body'] . '</p>'; 826             $commentsHTML .= '</div>'; 827 828             // Enviar el HTML de los comentarios como respuesta 829             echo $commentsHTML; 830         } else { 831             // Si no hay comentarios, mostrar un mensaje 832             echo 'No hay comentarios aún.'; 833         } 834     } 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856
```

Ln 311, Col 45 Spaces: 2 UTF-8 LF PHP Codeium: (...) AA + 161% - 29°C Parc. soleado 04:32 p.m. ESP 02/05/2024

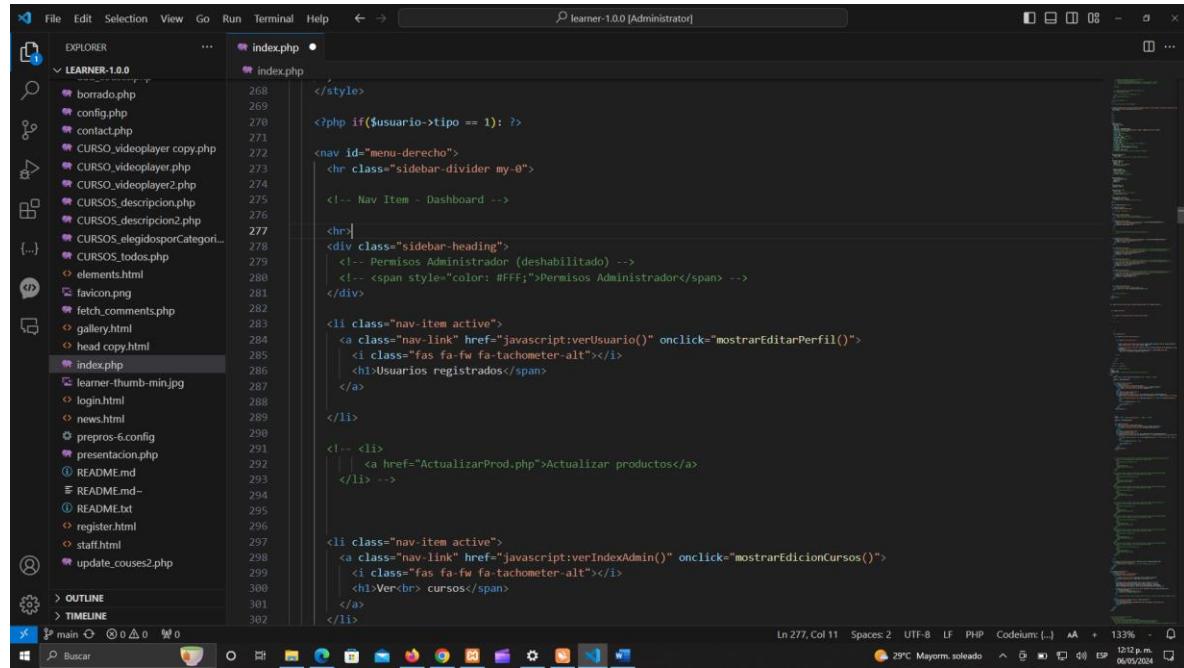


```
File Edit Selection View Go Run ... 🔍 learner-1.0.0 [Administrator] ⚡ ... EXPLORER CURSO_videoplayer.php ... CURSO_videoplayer.php 829     // Enviar el HTML de los comentarios como respuesta 830     echo $commentsHTML; 831 } else { 832     // Si no hay comentarios, mostrar un mensaje 833     echo 'No hay comentarios aún.'; 834 } 835 836 echo '</div>'; // Cierre del div de .comments-section 837 838 // Cierre del div del contenedor principal 839 echo '</div>'; 840 841 // Cierre del section 842 echo '</section>'; 843 } else { 844     // Si no se encontraron videos en la lista, mostrar un mensaje de error 845     echo "No se encontraron videos en la lista."; 846 } 847 848 // Cerrar la conexión a la base de datos 849 $db = null; 850 ?> 851 <!-- <h1 id="nombre-curso"><?php echo $nombrecurso; ?></h1> -->
```

Ln 311, Col 45 Spaces: 2 UTF-8 LF PHP Codeium: (...) AA + 161% - 04:32 p.m. 29°C Parc. soleado 02/05/2024

Recursos y toma en cuenta: Se muestra información adicional relevante para el curso, como recursos adicionales o instrucciones importantes.

9.3 Administrador:



```
<?php if($usuario->tipo == 1): ?>
<nav id="menu-derecho">
<hr class="sidebar-divider my-0">
<!-- Nav Item - Dashboard -->
<hr>
<div class="sidebar-heading">
| <!-- Permisos Administrador (desabilitado) -->
| <!-- <span style="color: #FFF;">Permisos Administrador</span> -->
</div>

<li class="nav-item active">
<a class="nav-link" href="javascript:verUsuario()" onclick="mostrarEditarPerfil()">
| <i class="fas fa-fw fa-tachometer-alt"></i>
| <h1>Usuarios registrados</h1>
</a>
</li>

<!-- <li>
| | <a href="ActualizarProd.php">Actualizar productos</a>
| </li> -->

<li class="nav-item active">
<a class="nav-link" href="javascript:verIndexAdmin()" onclick="mostrarEdicionCursos()">
| <i class="fas fa-fw fa-tachometer-alt"></i>
| <h1>Ver<br> cursos</h1>
</a>
</li>
```

<?php if(\$usuario->tipo == 1): ?>: Este bloque PHP verifica si el usuario actual tiene el tipo de usuario igual a 1, lo que generalmente significa que es un administrador. Si es así, se muestra el menú de administrador.

<nav id="menu-derecho">: Esto define un contenedor de navegación lateral para el menú del administrador.

<hr class="sidebar-divider my-0">: Esto agrega una línea horizontal como separador en el menú.

<div class="sidebar-heading">: Este div contiene el título "Permisos Administrador". Parece estar comentado actualmente.

<li class="nav-item active">: Este es un elemento de lista en el menú.

: Este es un enlace en el elemento de lista. Cuando se hace clic en él, se ejecuta la función mostrarEditarPerfil().

<i class="fas fa-fw fa-tachometer-alt"></i>: Esto agrega un ícono de "tachometer-alt" utilizando Font Awesome.

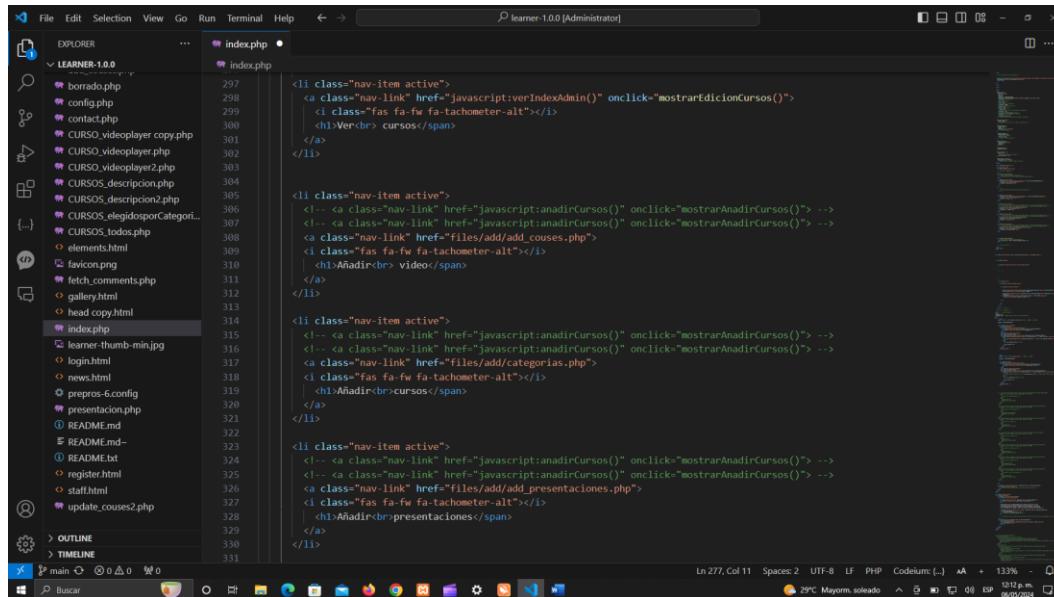
<h1>Usuarios registrados: Aquí se muestra el título "Usuarios registrados".

: Este es un enlace para agregar un nuevo video. Cuando se hace clic en él, lleva a la página add_courses.php.

: Este es un enlace para añadir nuevos cursos. Cuando se hace clic en él, lleva a la página categorias.php.

: Este es un enlace para añadir nuevas presentaciones. Cuando se hace clic en él, lleva a la página add_presentaciones.php.

<?php endif; ?>: Esto cierra el bloque if al principio del código, que verifica si el usuario es un administrador.



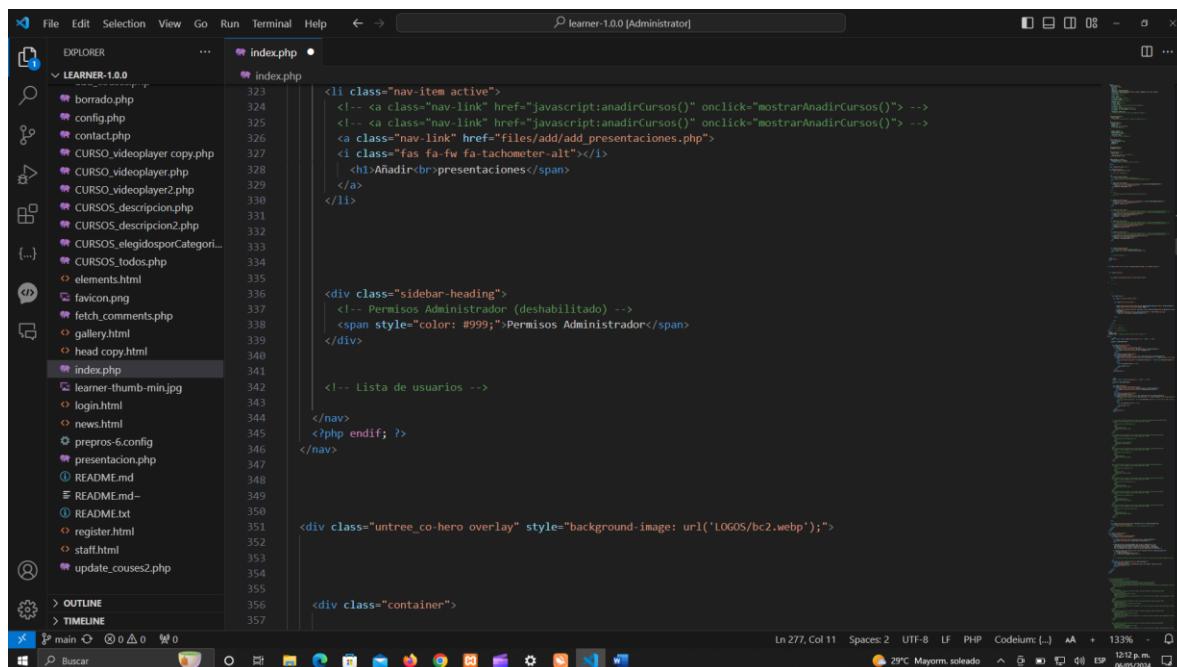
Code editor screenshot showing the navigation menu code for index.php. The code includes several active navigation items with links to add_courses.php, categorias.php, and add_presentaciones.php. The code is heavily commented with JavaScript onclick events and CSS classes like .nav-item-active and .fa-fw fa-tachometer-alt.

```
<li class="nav-item active">
    <a class="nav-link" href="javascript:verIndexAdmin()" onclick="mostrarEdicionCursos()">
        <i class="fas fa-fw fa-tachometer-alt"></i>
        <h1>Ver<br> cursos</h1>
    </a>
</li>

<li class="nav-item active">
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <a class="nav-link" href="files/add/add_courses.php">
        <i class="fas fa-fw fa-tachometer-alt"></i>
        <h1>Añadir<br> video</h1>
    </a>
</li>

<li class="nav-item active">
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <a class="nav-link" href="files/add/categorias.php">
        <i class="fas fa-fw fa-tachometer-alt"></i>
        <h1>Añadir<br>cursos</h1>
    </a>
</li>

<li class="nav-item active">
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <a class="nav-link" href="files/add/add_presentaciones.php">
        <i class="fas fa-fw fa-tachometer-alt"></i>
        <h1>Añadir<br>presentaciones</h1>
    </a>
</li>
```



Code editor screenshot showing the sidebar heading and user list code for index.php. It includes a sidebar heading for 'Permisos Administrador' and a list of users. The code uses PHP conditional logic to determine if the user has administrative permissions.

```
<li class="nav-item active">
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
    <a class="nav-link" href="files/add/add_presentaciones.php">
        <i class="fas fa-fw fa-tachometer-alt"></i>
        <h1>Añadir<br>presentaciones</h1>
    </a>
</li>

<div class="sidebar-heading">
    <!-- Permisos Administrador (deshabilitado) -->
    <span style="color: #999;">Permisos Administrador</span>
</div>

<!-- Lista de usuarios -->
</nav>
</php endif; ?>
</nav>

<div class="untree_co-hero overlay" style="background-image: url('LOGOS/bc.webp');">
    <div class="container">
```

Usuarios registrados:

Este código se encarga de mostrar una lista de usuarios registrados en forma de tarjetas.

```
</div>

<li class="nav-item active">
  <a class="nav-link" href="javascript:verUsuario()" onclick="mostrarEditarPerfil()">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <h1>Usuarios registrados</h1>
  </a>
</li>
```

En el primer bloque de código, se define la función verUsuario(). Esta función se ejecuta cuando se hace clic en el enlace "Usuarios registrados" en el menú de navegación lateral. Su objetivo es hacer una solicitud AJAX para obtener la información de los usuarios registrados y mostrarla en la página.

```
Codeium: Refactor | Explain | Generate JSDoc | X
const verUsuario = () => {
  let url = "files/verUsuarios.php";
  let send = {

  }
  let mensaje = '';
  let ejecutar = '';

  ajaxPostControl(url, "informacion", send, mensaje, ejecutar);
}
```

El segundo bloque de código es una sección HTML que contiene un contenedor para mostrar las tarjetas de los usuarios registrados. Utiliza un bucle foreach para iterar sobre el array \$usuarios, que parece ser una lista de usuarios obtenida previamente. Por cada usuario, se crea una tarjeta (`<div class="card shadow">`) que muestra el nombre completo del usuario, la fecha de registro y un menú desplegable de opciones. Este menú desplegable tiene una opción para eliminar el usuario, que está vinculada a la función eliminandoUsuario().

```

55     <?php foreach($usuarios as $usu): ?>
56     <div class="card shadow">
57       <!-- Card Header -->
58       <div class="card-header">
59         
60         <h6 class="m-0 font-weight-bold text-primary"><?php echo $usu["nombre"]." ".$usu["apellidos"]; ?></h6>
61       <div class="dropdown-item">
62         <a class="dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
63           | <i class="fas fa-ellipsis-v fa-sm fa-fw text-gray-400"></i>
64         </a>
65         <div class="dropdown-menu dropdown-menu-right shadow animated--fade-in" aria-labelledby="dropdownMenuLink">
66           <div class="dropdown-header">Opciones</div>
67           <a class="dropdown-item" href="javascript:eliminandoUsuario(<?php echo $usu["id"]; ?>)">Eliminar Usuario</a>
68         </div>
69       </div>
70     </div>
71   <!-- Card Body -->
72   <div class="card-body">
73     | Fecha de registro: <?php echo $usu["fechaRegistro"]; ?>
74   </div>
75
76   </div>
77 </php endforeach; ?>
78 </div>
79 <hr class="sidebar-divider" style="display: none;">
80   <div class="row" id="informacion">
81     <div id="informacion"></div>
82   </div>
83 </script>
84 document.querySelector('#closeButton').addEventListener('click', cerrarTodasLasPestanas);
85
86 function cerrarTodasLasPestanas() {
87   var cards = document.querySelectorAll('.card');
88   cards.forEach(function(card) {
89     card.parentNode.removeChild(card);
90   });
91   // Ocultar el botón de cierre
92   document.querySelector('#closebutton').style.display = 'none';
93 }

```

En el tercer bloque de código, se define la función cerrarTodasLasPestanas(). Esta función se ejecuta cuando se hace clic en el botón de cierre (<div class="close2" id="closeButton">×</div>) en la esquina superior derecha de cada tarjeta de usuario. Su propósito es cerrar todas las tarjetas de usuario mostradas en la página.

El último bloque de código define la función eliminandoUsuario(). Esta función se ejecuta cuando se confirma la eliminación de un usuario a través del cuadro de diálogo de confirmación de alertify. Realiza una solicitud AJAX para eliminar el usuario seleccionado y luego llama a la función verUsuarios() para actualizar la lista de usuarios mostrada en la página.

```

const eliminandoUsuario = idUsuario => [
  alertify.confirm("¿Está seguro de eliminar el usuario?", function(e) {
    if (e) {
      let url = "files/eliminandoUsuario.php";
      let send = {
        idUsuario
      }
      let mensaje = '';
      let ejecutar = 'verUsuarios()';

      ajaxPostControl(url, "informacion", send, mensaje, ejecutar);
    }
  });
]

```

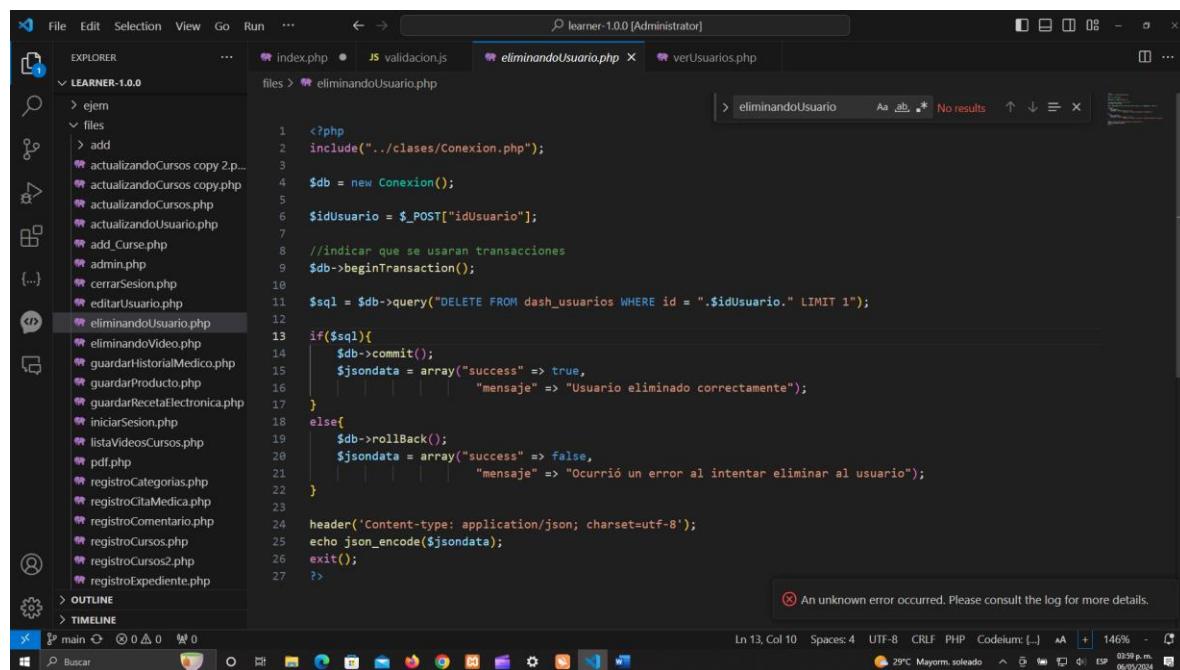
Conexión y recepción de datos: Se conecta a la base de datos y recibe el ID del usuario a eliminar.

Inicio de transacción: Se inicia una transacción para agrupar las operaciones de eliminación.

Eliminación del usuario: Se ejecuta una consulta SQL para eliminar el usuario de la base de datos.

Gestión de éxito o error: Se confirma la transacción si la eliminación tiene éxito, o se revierte si hay un error.

Respuesta JSON: Se devuelve una respuesta JSON indicando el éxito o fracaso de la operación.



The screenshot shows a code editor window titled "learner-1.0.0 [Administrator]". The left sidebar lists files in the "LEARNER-1.0.0" directory, including "index.php", "validacion.js", "eliminarUsuario.php", and "verUsuarios.php". The main editor area contains the following PHP code:

```
<?php
include("../clases/Conexion.php");
$db = new Conexion();
$idUsuario = $_POST["idUsuario"];
//indicar que se usaran transacciones
$db->beginTransaction();
$sql = $db->query("DELETE FROM dash_usuarios WHERE id = ".$idUsuario." LIMIT 1");
if($sql){
    $db->commit();
    $jsondata = array("success" => true,
                     "mensaje" => "Usuario eliminado correctamente");
} else{
    $db->rollBack();
    $jsondata = array("success" => false,
                     "mensaje" => "Ocurrió un error al intentar eliminar al usuario");
}
header('Content-type: application/json; charset=utf-8');
echo json_encode($jsondata);
exit();
?>
```

An unknown error occurred. Please consult the log for more details.

Ver cursos:

Enlace de navegación: Proporciona un enlace para que los administradores vean los cursos.

Al hacer clic en él, se llama a la función verIndexAdmin() y se muestra la sección de edición de

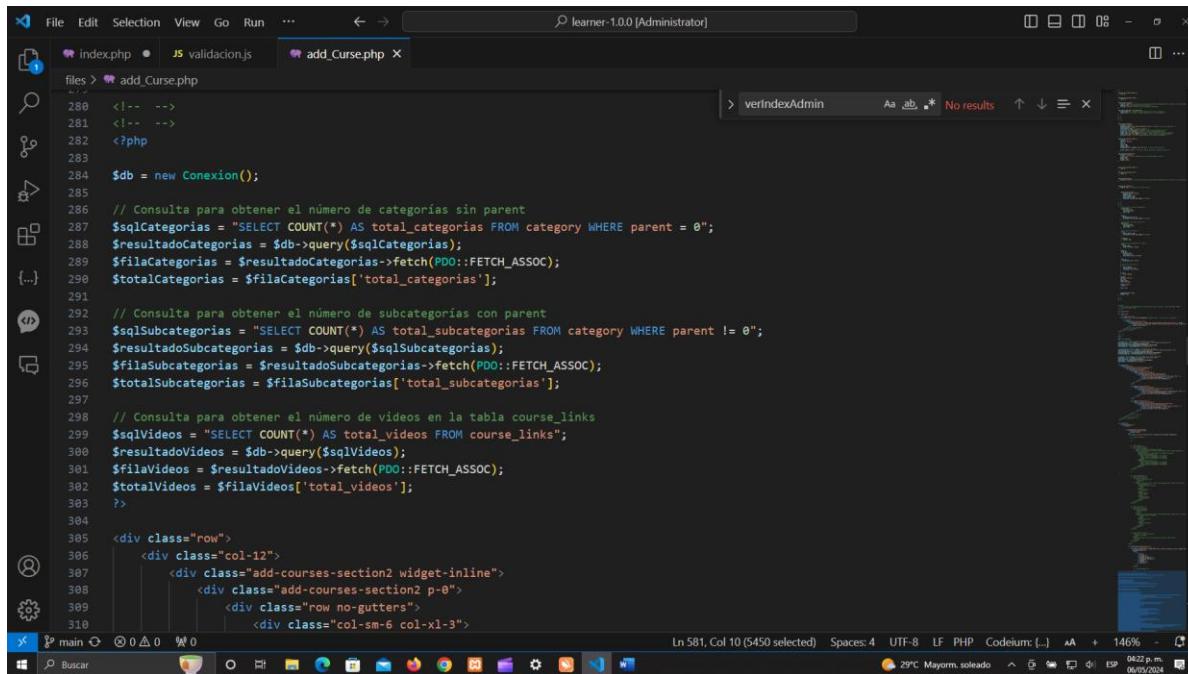
cursos.

```
<li class="nav-item active">
<a class="nav-link" href="javascript:verIndexAdmin()" onclick="mostrarEdicionCursos()">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <h1>Ver<br> cursos</h1>
</a>
</li>
```

Función verIndexAdmin(): Esta función realiza una solicitud para cargar la página files/add_Curse.php mediante AJAX. No envía ningún dato (send está vacío), y después de recibir la respuesta, no se ejecuta ninguna acción adicional (ejecutar está vacío).

```
Codeium: Refactor | Explain | Generate JSDoc | X
220 const verIndexAdmin = () => {
221     let url = "files/add_Curse.php";
222     let send = [
223
224     ]
225     let mensaje = '';
226     let ejecutar = '';
227
228     ajaxPostControl(url, "admin", send, mensaje, ejecutar);
229 }
230
231
```

Visualización de cursos: La sección visualiza estadísticas generales sobre categorías, subcategorías y videos. Además, muestra una lista de cursos disponibles con detalles como el nombre del curso, instructor, categoría, total de secciones, total de lecciones, total de inscritos, estado y precio. También proporciona enlaces para ver el curso en el frontend, editar el curso y eliminarlo.



The screenshot shows a code editor interface with several tabs open. The main tab contains PHP code for calculating course statistics and generating HTML output. The sidebar shows a file tree with files like index.php, add_Curse.php, and validation.js. A search bar at the top right shows the query 'verIndexAdmin' with 'No results' found. The bottom status bar displays file paths, line numbers, and other development information.

```
File Edit Selection View Go Run ... ↶ ↷
learner-1.0.0 [Administrator]
index.php • JS validation.js • add_Curse.php
files > add_Curse.php
280 <!-- -->
281 <!-- -->
282 <?php
283
284 $db = new Conexion();
285
286 // Consulta para obtener el número de categorías sin parent
287 $sqlCategorias = "SELECT COUNT(*) AS total_categorias FROM category WHERE parent = 0";
288 $resultadoCategorias = $db->query($sqlCategorias);
289 $filaCategorias = $resultadoCategorias->fetch(PDO::FETCH_ASSOC);
290 $totalCategorias = $filaCategorias['total_categorias'];
291
292 // Consulta para obtener el número de subcategorías con parent
293 $sqlSubcategorias = "SELECT COUNT(*) AS total_subcategorias FROM category WHERE parent != 0";
294 $resultadoSubcategorias = $db->query($sqlSubcategorias);
295 $filaSubcategorias = $resultadoSubcategorias->fetch(PDO::FETCH_ASSOC);
296 $totalSubcategorias = $filaSubcategorias['total_subcategorias'];
297
298 // Consulta para obtener el número de videos en la tabla course_links
299 $sqlVideos = "SELECT COUNT(*) AS total_videos FROM course_links";
300 $resultadoVideos = $db->query($sqlVideos);
301 $filaVideos = $resultadoVideos->fetch(PDO::FETCH_ASSOC);
302 $totalVideos = $filaVideos['total_videos'];
303 >
304
305 <div class="row">
306     <div class="col-12">
307         <div class="add-courses-section2 widget-inline">
308             <div class="add-courses-section2 p-0">
309                 <div class="row no-gutters">
310                     <div class="col-sm-6 col-xl-3">
```

```
306     <div class="col-12">
307         <div class="add-courses-section2 widget-inline">
308             <div class="add-courses-section p-0">
309                 <div class="row no-gutters">
310                     <div class="col-sm-6 col-xl-3">
311                         <a href="#" class="text-secondary">
312                             <div class="add-courses-section shadow-none m-0">
313                                 <div class="add-courses-section text-center">
314                                     <i class="dripicons-link text-muted" style="font-size: 24px;"></i>
315                                     <h3><?php echo $totalCategorias; ?></h3>
316                                     <p class="text-muted font-15 mb-0">Categorías</p>
317                             </div>
318                         </a>
319                     </div>
320
321                     <div class="col-sm-6 col-xl-3">
322                         <a href="#" class="text-secondary">
323                             <div class="add-courses-section shadow-none m-0 border-left">
324                                 <div class="card-body text-center">
325                                     <i class="dripicons-link-broken text-muted" style="font-size: 24px;"></i>
326                                     <h3><?php echo $totalSubcategorias; ?></h3>
327                                     <p class="text-muted font-15 mb-0">Subcategorías</p>
328                             </div>
329                         </a>
330                     </div>
331
332                     <div class="col-sm-6 col-xl-3">
333                         <a href="#" class="text-secondary">
334                             <div class="add-courses-section shadow-none m-0 border-left">
```

```
324     <div class="add-courses-section shadow-none m-0 border-left">
325         <div class="card-body text-center">
326             <i class="dripicons-link-broken text-muted" style="font-size: 24px;"></i>
327             <h3><?php echo $totalSubcategorias; ?></h3>
328             <p class="text-muted font-15 mb-0">Subcategorías</p>
329         </div>
330     </a>
331 </div>
332
333 <div class="col-sm-6 col-xl-3">
334     <a href="#" class="text-secondary">
335         <div class="add-courses-section shadow-none m-0 border-left">
336             <div class="add-courses-section text-center">
337                 <i class="dripicons-star text-muted" style="font-size: 24px;"></i>
338                 <h3><?php echo $totalVideos; ?></h3>
339                 <p class="text-muted font-15 mb-0">Videos</p>
340             </div>
341         </a>
342     </div>
343
344     </div> <!-- end row -->
345
346 </div> <!-- end card-box-->
347 </div> <!-- end col-->
348 </div> <!-- end row -->
349 </div>
350
351 <div class="row">
352     <div class="col-xl-12">
353         <div class="">
```

The screenshot shows a code editor window with the file `add_Curse.php` open. The code is a PHP script that queries a database to list courses. It includes sections for displaying course details and navigating to specific course pages. The interface is dark-themed.

```
<?php  
$db3 = new Conexion();  
$sql3 = "SELECT * FROM category WHERE parent != 0";  
  
$resultado3 = $db3->query($sql3);  
  
>  
<!-- LISTA DE CURSOS -->  
<?php foreach ($resultado3 as $dat) : ?>  
  
<tbody class="add-courses-section3">  
  
<tr>  
    <td>  
        <td>  
            <strong><a href="#"><?php echo $dat['name']; ?></a></strong><br>  
            <small class="text-white">  
                | Instructor: <b>Nombre del Instructor Ejemplo</b>  
            </small>  
        </td>  
        <td>  
            <span class="badge badge-dark-lighten">  
                Nombre de la Categoría Ejemplo  
            </span>  
        </td>  
        <td>  
            <small class="text-white">  
                <span class="text-white">  
                    $Gratis  
                </span>  
  
</td>  
</tr>
```

The screenshot shows a code editor window with the file `add_Curse.php` open. The code is a PHP script that generates a dropdown menu. It includes sections for displaying course details and navigating to specific course pages. The interface is dark-themed.

```
<td>  
    <span class="text-white">  
        $Gratis  
    </span>  
</td>  
  
<td>  
    <div class="dropdown dropdown-right">  
        <button type="button" class="btn btn-sm btn-outline-primary btn-rounded btn-icon" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">  
            <i class="mdi mdi-dots-vertical"></i>  
        </button>  
        <ul class="dropdown-menu">  
            <li><a class="dropdown-item" href="CURSOS_descripcion.php?id=<?php echo $dat["id"]; ?>">Ver Curso en el Frontend</a></li>  
            <li><a class="dropdown-item" href="files/add/update_couses2.php?id=<?php echo $dat["id"]; ?>">Editar Este Curso</a></li>  
            <!-- <li><a class="dropdown-item" href="javascript:ver_curso_temas(<?php echo $dat["id"]; ?>)">Editar Este Curso</a></li> -->  
  
            <!-- <li><a class="dropdown-item" href="#">Sección y Lección</a></li> -->  
            <!-- <li>  
                <a class="dropdown-item" href="#">  
                    Marcar como Pendiente  
                </a>  
            </li>  
            <li>  
                <a class="dropdown-item" href="#">  
                    Marcar como Activo  
                </a>  
            </li> -->  
            <!--
```

The screenshot shows a code editor window with several tabs open. The active tab is 'add_Curse.php'. The code contains PHP loops and conditionals, along with a script block containing a function named 'cerrarPestana()' which removes a DOM element when its close button is clicked. The interface includes a sidebar with file navigation, a search bar at the top, and a status bar at the bottom showing file paths, line numbers, and system information.

```

555     </td>
556     </tr>
557   </tbody>
558   <?php endforeach; ?>
559
560   </table>
561   <!-- <div class="img-fluid w-100 text-center">
562     <br>
563     No se encontraron datos
564   </div> -->
565   <!-- -->
566   </div>
567   </div>
568   </div>
569
570 </div>
571
572
573
574 <script>
575   function cerrarPestana() {
576     var row2 = document.querySelector('.row2');
577     row2.parentNode.removeChild(row2); // Elimina la tarjeta de la sección de cursos del DOM
578   }
579
580
581 </script>
582

```

Script de cierre de pestaña: Cuando se hace clic en la "X" para cerrar la pestaña, se ejecuta la función `cerrarPestana()`. Esta función elimina la tarjeta de la sección de cursos del DOM.

Añadir video:

Enlace de navegación: Proporciona un enlace para que los administradores añadan videos. Al hacer clic en él, se muestra la interfaz para agregar videos.

The screenshot shows a snippet of HTML code for a navigation menu. It includes a list item with a class 'nav-item active' and a sub-item with a class 'nav-link'. The sub-item has an href attribute pointing to 'files/add/add_couses.php' and an onclick attribute calling a JavaScript function 'mostrarAnadirCursos()'.

```

<li class="nav-item active">
  <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
  <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
  <a class="nav-link" href="files/add/add_couses.php">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <h1>Añadir<br> video</span>
  </a>
</li>

```

Consulta de categorías: Se consulta la base de datos para obtener las categorías disponibles y se muestran en un menú desplegable. Si se encuentran categorías, se muestran en el menú. Si no se encuentran categorías, se muestra un mensaje indicando que no se encontraron categorías.

```

files > add > add_couses.php
355     </label>
356     <div class="col-md-10">
357         <?php
358             $db = new Conexion();
359             $menu = $db->query("SELECT * FROM category");
360
361             // Verificar si se encontraron categorías
362             if ($menu->rowCount() > 0) {
363                 echo '<select class="form-control select2 select2-hidden-accessible" data-toggle="select2" name="sub_category_id" id="sub_category_id">';
364                 echo '<option value="" data-select2-id="2">';
365                 echo '<font style="vertical-align: inherit;">Seleccione una categoría</font>';
366                 echo '</font>';
367                 echo '</option>';
368
369                 // Imprimir las categorías principales
370                 while ($row = $menu->fetch(PDO::FETCH_ASSOC)) {
371                     if ($row["parent"] == 0) {
372                         echo '<option value="' . $row["id"] . '" disabled="" data-parent="" data-select2-id="2">' . $row["name"] . '</option>';
373
374                         // Consultar y mostrar las subcategorías
375                         $subcategories = $db->query("SELECT * FROM category WHERE parent = " . $row["id"]);
376                         while ($sub_row = $subcategories->fetch(PDO::FETCH_ASSOC)) {
377                             echo '<option value="' . $sub_row["id"] . '" data-parent="" data-select2-id="2">' . $sub_row["parent"] . '>' . $sub_row["name"] . '</option>';
378                         }
379                     }
380
381                     echo '</select>';
382                 } else {
383                     echo "No se encontraron categorías.";
384                 }
385             }
386
387             $menu->closeCursor();
388         }
389     
```

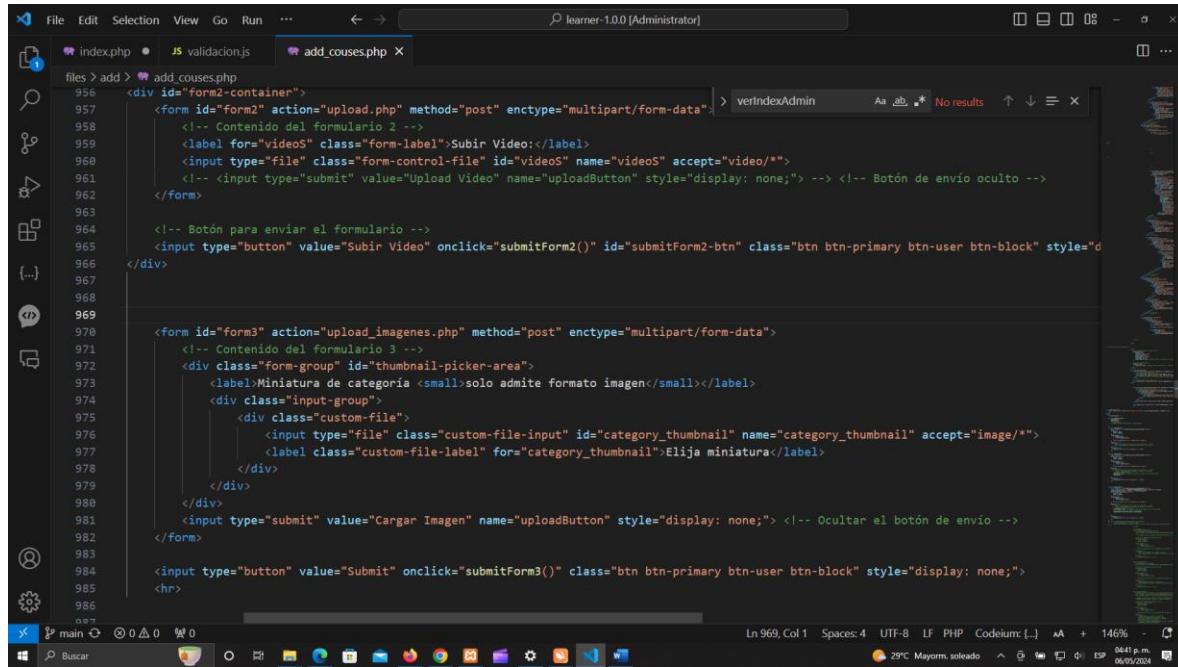
Script para llenar el desplegable: Cuando se carga la página, se llama a la función `populateSelect()` para llenar el menú desplegable con datos obtenidos de la función `fetchData()`. Esta función simula obtener datos y llenar el menú desplegable con números del 1 al 25.

```

files > add > add_couses.php
499
500     document.addEventListener("DOMContentLoaded", function() {
501         var selectElement = document.getElementById("theme_selector");
502
503         // Define la función fetchData() para obtener los datos
504         function fetchData(searchTerm, page) {
505             // Simular la obtención de datos (reemplaza esto con tu lógica de obtención de datos real)
506             var itemsPerPage = 25; // Número de elementos por página
507             var data = [];
508             for (var i = (page - 1) * itemsPerPage; i < page * itemsPerPage; i++) {
509                 var number = i + 1; // Empezar desde 1
510                 if (searchTerm & number.toString().indexOf(searchTerm) === -1) {
511                     continue; // Saltar si no coincide con el término de búsqueda
512                 }
513                 data.push(number);
514             }
515             return data;
516         }
517
518         // Llena el desplegable con datos obtenidos de fetchData()
519         function populateSelect() {
520             var data = fetchData('', 1); // Llama a fetchData() sin término de búsqueda y página 1
521             data.forEach(function(option) {
522                 var optionElement = document.createElement("option");
523                 optionElement.value = option;
524                 optionElement.textContent = option;
525                 selectElement.appendChild(optionElement);
526             });
527         }
528
529         // Llama a la función para llenar el desplegable cuando se carga la página
530     }
531 
```

Formularios:

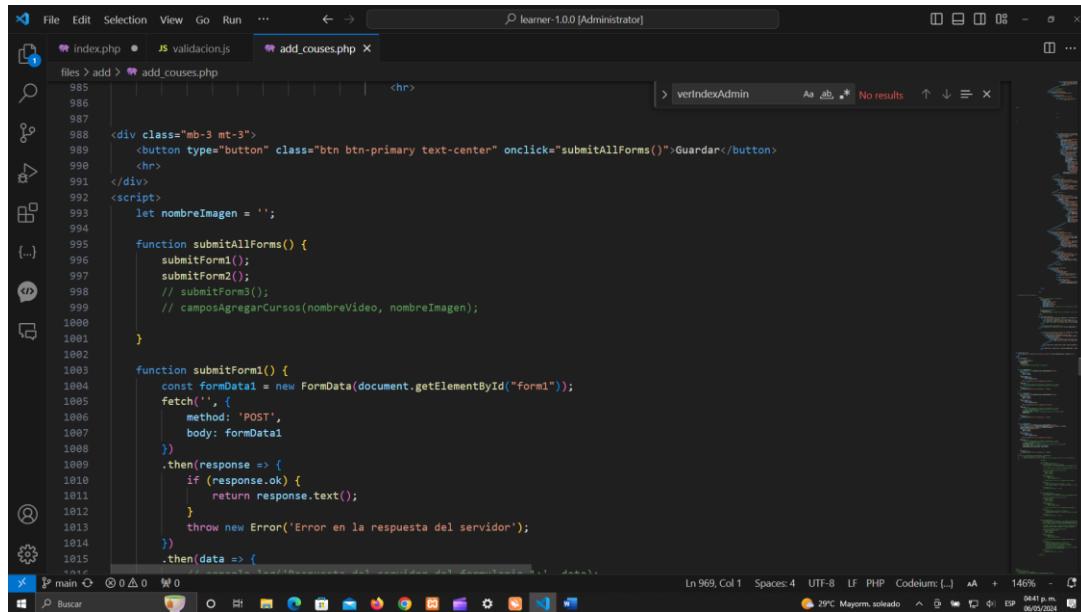
Formularios para cargar videos e imágenes: Se muestran dos formularios: uno para cargar videos y otro para cargar imágenes. Los campos de entrada permiten al usuario seleccionar archivos de video e imágenes, respectivamente.



The screenshot shows a code editor with two files open: index.php and add_couses.php. The add_couses.php file contains the following code:

```
files > add > add_couses.php
956     <div id="form2">
957         <!-- Contenido del formulario 2 -->
958         <label for="videoO" class="form-label">Subir Video:</label>
959         <input type="file" class="form-control-file" id="videoO" name="videoO" accept="video/*">
960         <!-- <input type="submit" value="Upload Video" name="uploadButton" style="display: none;"> --> <!-- Botón de envío oculto -->
961     </form>
962
963     <!-- Botón para enviar el formulario -->
964     <input type="button" value="Subir Video" onclick="submitForm2()" id="submitForm2-btn" class="btn btn-primary btn-user btn-block" style="display: none;">
965 </div>
966
967
968
969
970     <form id="form3" action="upload_imagenes.php" method="post" enctype="multipart/form-data">
971         <!-- Contenido del formulario 3 -->
972         <div class="form-group" id="thumbnail-picker-area">
973             <label>Minatura de categoría <small>solo admite formato imagen</small></label>
974             <div class="input-group">
975                 <div class="custom-file">
976                     <input type="file" class="custom-file-input" id="category_thumbnail" name="category_thumbnail" accept="image/*">
977                     <label class="custom-file-label" for="category_thumbnail">Elige miniatura</label>
978                 </div>
979             </div>
980             <input type="submit" value="Cargar Imagen" name="uploadButton" style="display: none;"> <!-- Ocultar el botón de envío -->
981         </form>
982
983         <input type="button" value="Submit" onclick="submitForm3()" class="btn btn-primary btn-user btn-block" style="display: none;">
984     <hr>
```

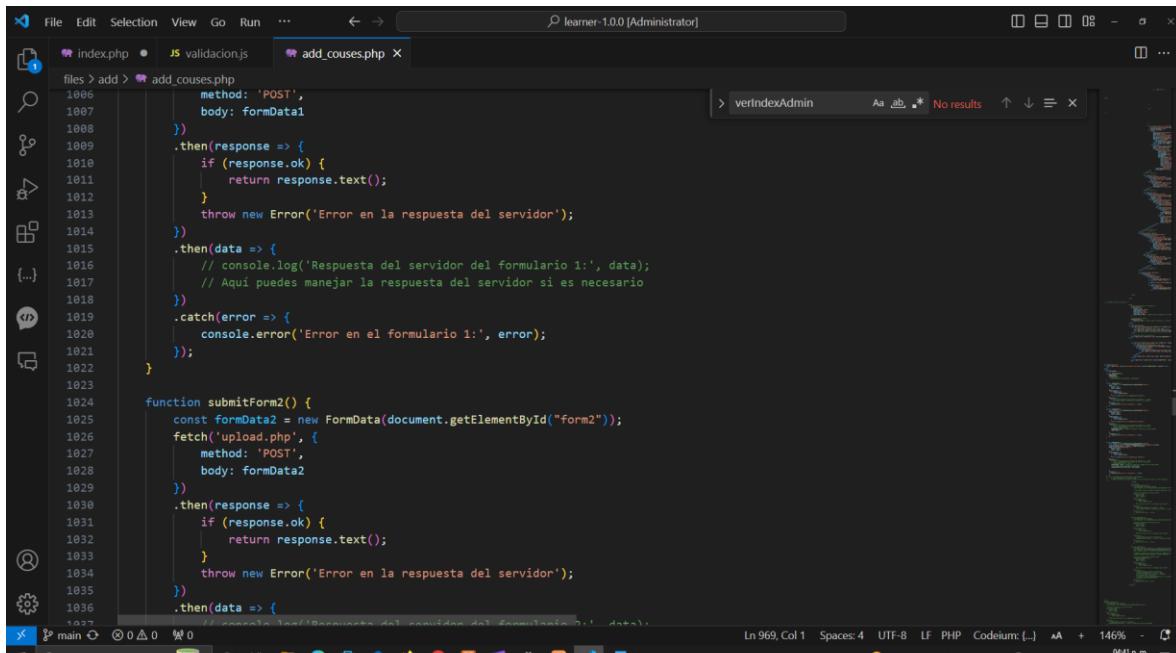
Botón para enviar el formulario: Se muestra un botón para enviar el formulario de carga de videos. Al hacer clic en este botón, se ejecuta la función submitForm2() para enviar el formulario de carga de videos mediante AJAX. Una vez que se carga el video, se llama a la función submitForm3() para enviar el formulario de carga de imágenes.



The screenshot shows a code editor with the add_couses.php file open. It contains the following JavaScript code:

```
955
956
957
958     <div class="mb-3 mt-3">
959         <button type="button" class="btn btn-primary text-center" onclick="submitAllForms()">Guardar</button>
960     </div>
961
962     <script>
963         let nombreImagen = '';
964
965         function submitAllForms() {
966             submitForm1();
967             submitForm2();
968             // submitForm3();
969             // camposAgregarCursos(nombreVideo, nombreImagen);
970         }
971
972         function submitForm1() {
973             const formData = new FormData(document.getElementById("form1"));
974             fetch('/', {
975                 method: 'POST',
976                 body: formData
977             })
978             .then(response => {
979                 if (response.ok) {
980                     return response.text();
981                 }
982                 throw new Error('Error en la respuesta del servidor');
983             })
984             .then(data => {
985                 // Procesar la respuesta del servidor
986             })
987         }
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
```

Script para enviar formularios: Se definen funciones para enviar los formularios de carga de videos e imágenes mediante AJAX. Estas funciones envían los datos del formulario al servidor y manejan la respuesta del servidor. Después de enviar el formulario de carga de videos, se llama a la función submitForm3() para enviar el formulario de carga de imágenes.



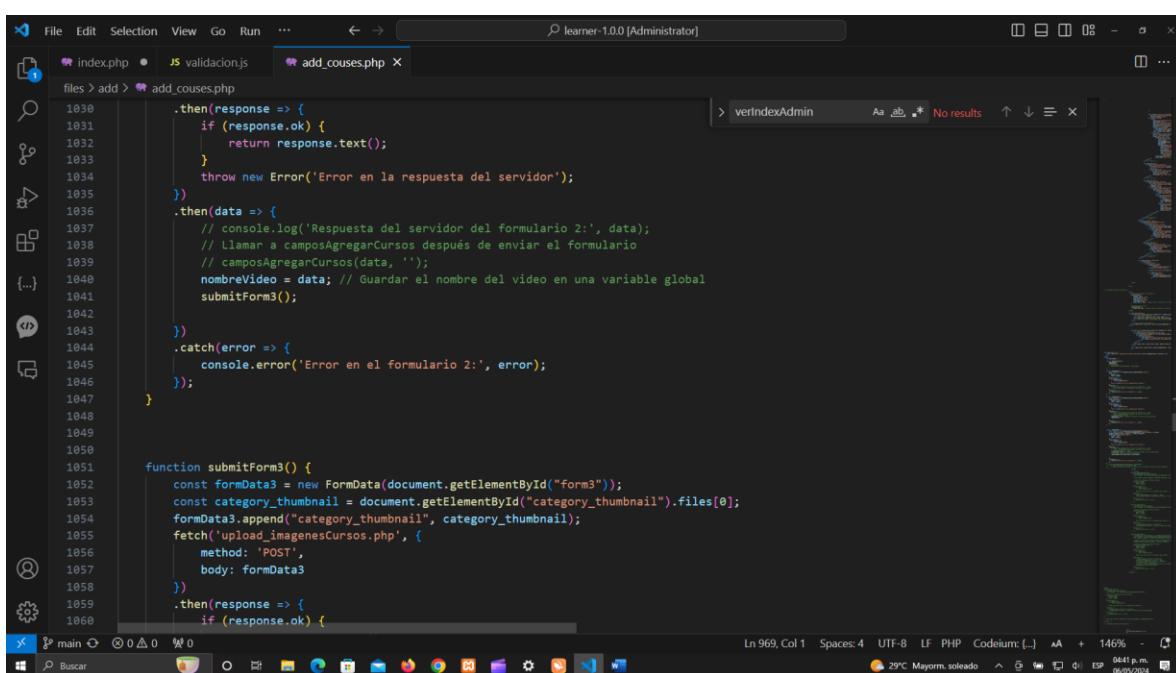
```

File Edit Selection View Go Run ... < > learner-1.0.0 [Administrator]
index.php JS validation.js add_courses.php
files > add > add_courses.php
1006     method: 'POST',
1007     body: formData1
1008   )
1009   .then(response => {
1010     if (response.ok) {
1011       return response.text();
1012     }
1013     throw new Error('Error en la respuesta del servidor');
1014   })
1015   .then(data => {
1016     // console.log('Respuesta del servidor del formulario 1:', data);
1017     // Aquí puedes manejar la respuesta del servidor si es necesario
1018   })
1019   .catch(error => {
1020     console.error('Error en el formulario 1:', error);
1021   });
1022 }

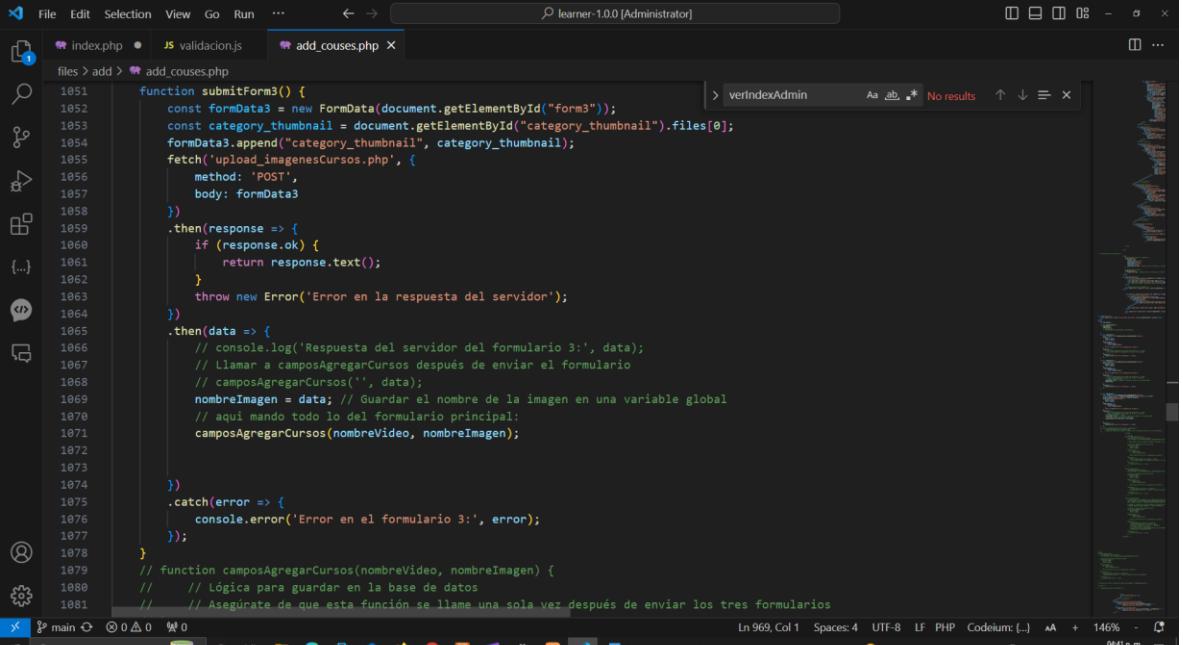
1023 function submitForm2() {
1024   const formData2 = new FormData(document.getElementById("form2"));
1025   fetch('upload.php', {
1026     method: 'POST',
1027     body: formData2
1028   })
1029   .then(response => {
1030     if (response.ok) {
1031       return response.text();
1032     }
1033     throw new Error('Error en la respuesta del servidor');
1034   })
1035   .then(data => {
1036     // ...
1037     // Llamar a camposAgregarCursos después de enviar el formulario
1038     // camposAgregarCursos(data, '');
1039     nombreVideo = data; // Guardar el nombre del video en una variable global
1040     submitForm3();
1041   })
1042 }
1043
1044 .catch(error => {
1045   console.error('Error en el formulario 2:', error);
1046 });
1047 }

1048
1049
1050
1051 function submitForm3() {
1052   const formData3 = new FormData(document.getElementById("form3"));
1053   const category_thumbnail = document.getElementById('category_thumbnail').files[0];
1054   formData3.append("category_thumbnail", category_thumbnail);
1055   fetch('upload_imagenesCursos.php', {
1056     method: 'POST',
1057     body: formData3
1058   })
1059   .then(response => {
1060     if (response.ok) {
1061       ...
1062     }
1063   })
1064 }

```



Función para enviar todos los formularios: La función submitAllForms() se utiliza para enviar todos los formularios juntos. Esta función llama a submitForm1() para enviar el formulario principal, luego a submitForm2() para enviar el formulario de carga de videos y finalmente a submitForm3() para enviar el formulario de carga de imágenes.



The screenshot shows a code editor window with several tabs at the top: 'index.php', 'JS validacion.js', and 'add_courses.php'. The 'add_courses.php' tab is active and displays the following JavaScript code:

```
function submitForm3() {
    const formData3 = new FormData(document.getElementById("form3"));
    const category_thumbnail = document.getElementById("category_thumbnail").files[0];
    formData3.append("category_thumbnail", category_thumbnail);
    fetch('upload_imagenesCursos.php', {
        method: 'POST',
        body: formData3
    })
    .then(response => {
        if (response.ok) {
            return response.text();
        }
        throw new Error('Error en la respuesta del servidor');
    })
    .then(data => {
        // console.log('Respuesta del servidor del formulario 3:', data);
        // llamar a camposAgregarCursos después de enviar el formulario
        // camposAgregarCursos('', data);
        nombreImagen = data; // Guardar el nombre de la imagen en una variable global
        // aquí mando todo lo del formulario principal:
        camposAgregarCursos(nombreVideo, nombreImagen);

    })
    .catch(error => {
        console.error('Error en el formulario 3:', error);
    });
}
// function camposAgregarCursos(nombreVideo, nombreImagen) {
//     // Lógica para guardar en la base de datos
//     // Asegúrate de que esta función se llame una sola vez después de enviar los tres formularios
```

The code is part of a larger application, likely a Laravel project, as indicated by the file structure and PHP code snippets. It handles file uploads for a third form ('form3') and then triggers other functions to handle video and image uploads.

Preparación de los datos con JavaScript

Obtención de los valores de los campos del formulario: Se obtienen los valores de varios campos del formulario, como el nombre del curso, el tema, la subcategoría, las palabras clave (meta_keywords), la descripción (meta_description), la descripción corta (short_description), la descripción completa (description), los requisitos, y el nivel del curso.

Verificación de la existencia de todos los campos requeridos: Se verifica si todos los campos requeridos tienen un valor. Si falta algún campo, se crea un array camposFaltantes que almacena los nombres de los campos que faltan.

```

File Edit Selection View Go Run Terminal Help learner-1.0.0 [Administrator]
index.php add_courses.php JS validacion.js JS validacionCursos.js
scripts > JS validacionCursos.js [ camposAgregarCursos ]
Codeium: Refactor | Explain | Generate JSDoc | ...
1 const camposAgregarCursos = (nombreVideo, nombreImagen) => {
2     let nombre = document.getElementById('nombre').value;
3     let tema = document.getElementById('theme_selector').value;
4
5     let subcategoriaSelect = document.getElementById('sub_category_id');
6     let subcategoria = subcategoriaSelect.value;
7     let parent = subcategoriaSelect.options[subcategoriaSelect.selectedIndex].getAttribute('data-parent');
8
9     let meta_keywords = document.getElementById('meta_keywords');
10    let meta_description = document.getElementsByName('meta_description')[0]; // Utiliza getElementsByName para obtener el primer elemento con ese nombre
11
12    let short_description = document.getElementById('short_description').value;
13    let description = document.getElementById('description').value;
14
15    // Obtener los valores de los requisitos
16    let requirementInputs = document.querySelectorAll('[name="requirements[]"]');
17    let requirements = Array.from(requirementInputs).map(input => input.value);
18
19    let level = document.getElementById('level').value;
20
21    // Verifica si los elementos existen antes de acceder a sus valores
22    if (nombre && tema && nombreVideo && nombreImagen && parent && meta_keywords && meta_description && short_description && description && requirements && level) {
23        let send = {
24            nombre,
25            tema,
26            nombreVideo,
27            nombreImagen,
28            subcategoria,
29            parent,
30            meta_keywords: meta_keywords.value,
31            meta_description: meta_description.value,
32            short_description,
33            description,
34            requirements,
35            level
36        };
37    }

```

The screenshot shows a Codeium IDE window with multiple tabs open. The active tab is 'validacionCursos.js'. The code in the editor is a function named 'camposAgregarCursos' that takes two parameters: 'nombreVideo' and 'nombreImagen'. It uses several variables to store values from HTML form fields: 'nombre', 'tema', 'parent', 'meta_keywords', 'meta_description', 'short_description', 'description', 'requirements', and 'level'. A check is performed to ensure all required fields have values. If they do, an object 'send' is created containing all these variables. The rest of the code is not visible in the screenshot.

Envío de los datos al servidor si todos los campos están completos: Si todos los campos requeridos tienen un valor, se construye un objeto send que contiene todos los datos recopilados. Luego, se especifica la URL del archivo PHP que procesará los datos, un mensaje de éxito y una función a ejecutar después de enviar los datos. Finalmente, se llama a la función ajaxPostControl para enviar los datos al servidor.

```

File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
index.php add_courses.php JS validacion.js JS validacionCursos.js
scripts > JS validacionCursos.js [ camposAgregarCursos ]
33         description,
34         requirements,
35         level
36     );
37
38     let url = "../../files/registroCursos.php";
39     let mensaje = "Los datos se guardaron correctamente";
40     let ejecutar = "verIndex2()";
41     ajaxPostControl(url, "registrandoCurso", send, mensaje, ejecutar);
42 } else {
43     // Encuentra los campos que faltan y muestra un mensaje de error
44     let camposFaltantes = [];
45     if (!nombre) camposFaltantes.push("Nombre");
46     if (!tema) camposFaltantes.push("Tema");
47     if (!nombreVideo) camposFaltantes.push("Nombre del Vídeo");
48     if (!nombreImagen) camposFaltantes.push("Nombre de la Imagen");
49     if (!subcategoria) camposFaltantes.push("Subcategoria");
50     if (!parent) camposFaltantes.push("Parent");
51     if (!meta_keywords) camposFaltantes.push("Meta Keywords");
52     if (!meta_description) camposFaltantes.push("Meta Description");
53     if (!short_description) camposFaltantes.push("Short Description");
54     if (!description) camposFaltantes.push("Description");
55     if (!requirements) camposFaltantes.push("Requirements");
56     if (!level) camposFaltantes.push("Level");
57
58     console.error('Error: No se encontraron todos los elementos necesarios');
59     console.error('Campos faltantes:', camposFaltantes.join(', '));
60 }
61
62
63

```

This screenshot shows the same Codeium IDE window as the previous one, but the code has been modified. It includes an 'else' block that handles cases where not all required fields are present. It creates an array 'camposFaltantes' and pushes various field names onto it if they are empty. It then logs an error message to the console indicating that not all required fields were found and lists the missing fields. The rest of the code remains the same as in the previous screenshot.

Manejo de campos faltantes: Si falta algún campo requerido, se muestra un mensaje de error en la consola del navegador indicando los campos que faltan.

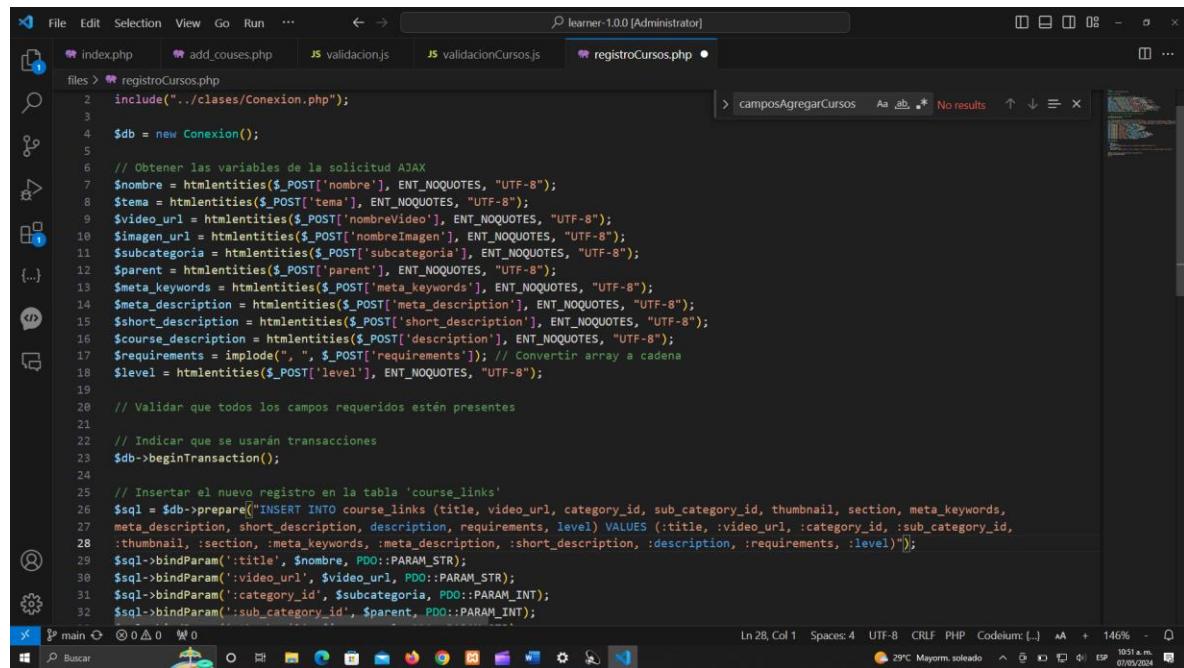
Validación a la base de datos

Inclusión de la clase de conexión a la base de datos: Se incluye el archivo que contiene la clase Conexion para establecer la conexión con la base de datos.

Creación de una instancia de la clase de conexión: Se crea una instancia de la clase Conexion para conectarse a la base de datos.

Obtención de variables de la solicitud AJAX: Se obtienen los datos enviados desde el formulario de registro de cursos a través de la solicitud AJAX. Las variables se filtran con htmlentities para evitar posibles ataques XSS.

Validación de campos requeridos: Aunque el código actualmente no incluye la validación de campos requeridos, generalmente se realizaría aquí para garantizar que todos los campos necesarios estén presentes antes de continuar con el procesamiento.



The screenshot shows a code editor with several tabs open. The main tab contains PHP code for registering a new course. The code includes variable filtering using htmlentities, transaction handling with PDO, and SQL query preparation to insert data into the 'course_links' table. Other tabs visible include 'index.php', 'add_courses.php', 'validacion.js', 'validacionCursos.js', and 'registroCursos.php'. The status bar at the bottom shows file statistics like 'In 28, Col 1', 'Spaces: 4', and 'UTF-8' encoding.

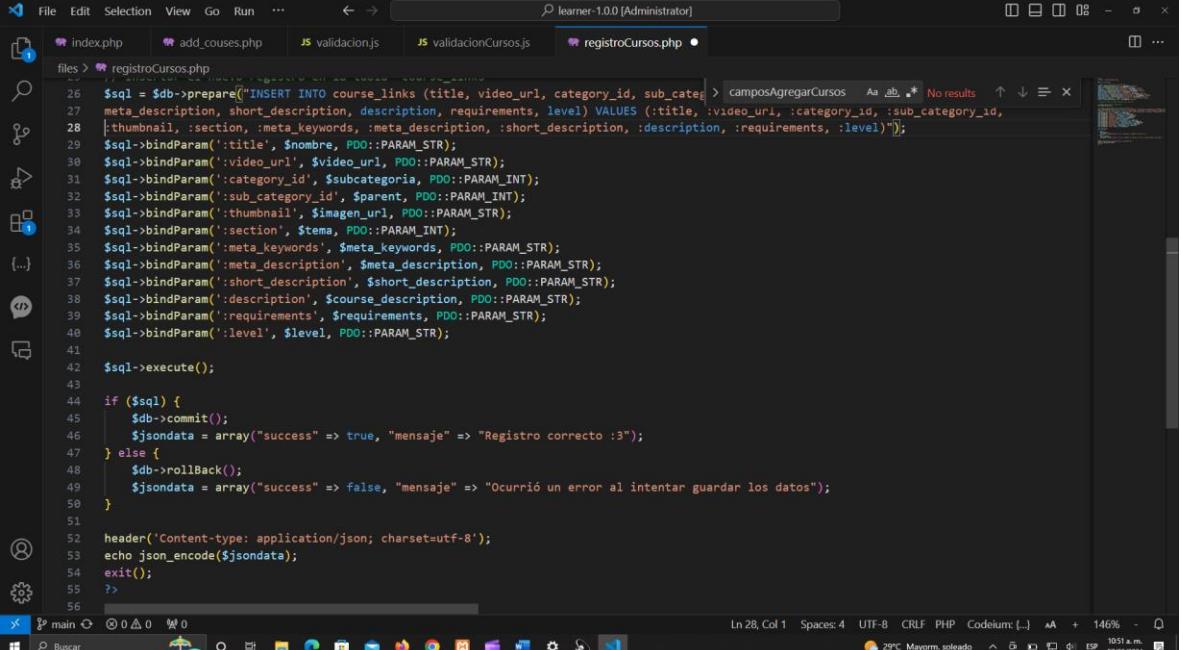
```
File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
files > registroCursos.php
1 index.php 2 add_courses.php 3 validacion.js 4 validacionCursos.js 5 registroCursos.php ...
2 include("../clases/Conexion.php");
3
4 $db = new Conexion();
5
6 // Obtener las variables de la solicitud AJAX
7 $nombre = htmlentities($_POST['nombre'], ENT_NOQUOTES, "UTF-8");
8 $tema = htmlentities($_POST['tema'], ENT_NOQUOTES, "UTF-8");
9 $video_url = htmlentities($_POST['nombreVideo'], ENT_NOQUOTES, "UTF-8");
10 $imagen_url = htmlentities($_POST['nombreImagen'], ENT_NOQUOTES, "UTF-8");
11 $subcategoria = htmlentities($_POST['subcategoria'], ENT_NOQUOTES, "UTF-8");
12 $parent = htmlentities($_POST['parent'], ENT_NOQUOTES, "UTF-8");
13 $meta_keywords = htmlentities($_POST['meta_keywords'], ENT_NOQUOTES, "UTF-8");
14 $meta_description = htmlentities($_POST['meta_description'], ENT_NOQUOTES, "UTF-8");
15 $short_description = htmlentities($_POST['short_description'], ENT_NOQUOTES, "UTF-8");
16 $course_description = htmlentities($_POST['description'], ENT_NOQUOTES, "UTF-8");
17 $requirements = implode(", ", $_POST['requirements']); // Convertir array a cadena
18 $level = htmlentities($_POST['level'], ENT_NOQUOTES, "UTF-8");
19
20 // Validar que todos los campos requeridos estén presentes
21
22 // Indicar que se usarán transacciones
23 $db->beginTransaction();
24
25 // Insertar el nuevo registro en la tabla 'course_links'
26 $sql = $db->prepare("INSERT INTO course_links (title, video_url, category_id, sub_category_id, thumbnail, section, meta_keywords,
27 meta_description, short_description, description, requirements, level) VALUES (:title, :video_url, :category_id, :sub_category_id,
28 :thumbnail, :section, :meta_keywords, :meta_description, :short_description, :description, :requirements, :level)");
29 $sql->bindParam(':title', $nombre, PDO::PARAM_STR);
30 $sql->bindParam(':video_url', $video_url, PDO::PARAM_STR);
31 $sql->bindParam(':category_id', $subcategoria, PDO::PARAM_INT);
32 $sql->bindParam(':sub_category_id', $parent, PDO::PARAM_INT);
```

Inicio de una transacción: Se inicia una transacción para asegurar la integridad de los datos en caso de que ocurra algún error durante el proceso de inserción.

Inserción de datos en la tabla course_links: Se prepara una consulta SQL para insertar un nuevo registro en la tabla course_links. Se vinculan los parámetros de la consulta con los valores obtenidos de la solicitud AJAX y se ejecuta la consulta.

Comprobación del resultado de la inserción: Se verifica si la consulta SQL se ejecutó correctamente. Si es así, se confirma la transacción y se devuelve un mensaje de éxito en formato JSON. En caso de error, se revierte la transacción y se devuelve un mensaje de error.

Respuesta JSON: Se envía la respuesta al cliente en formato JSON, que indica si el registro se realizó correctamente o si ocurrió algún error.



The screenshot shows a code editor window titled "learner-1.0.0 [Administrator]". The file "registroCursos.php" is open, displaying PHP code for inserting course data into a database. The code includes preparing an SQL statement, binding parameters for title, video URL, category ID, sub-category ID, thumbnail, section, meta keywords, meta description, short description, requirements, and level, and then executing the query. It also handles committing or rolling back the transaction based on the result and outputs a JSON response indicating success or an error message.

```
File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
index.php add_courses.php JS validacion.js JS validacionCursos.js registroCursos.php

26 $sql = $db->prepare("INSERT INTO course_links (title, video_url, category_id, sub_cate
27 meta_description, short_description, description, requirements, level) VALUES (:title, :video_url, :category_id, :sub_category_id,
28 :thumbnail, :section, :meta_keywords, :meta_description, :short_description, :description, :requirements, :level)");
29 $sql->bindParam(':title', $Nombre, PDO::PARAM_STR);
30 $sql->bindParam(':video_url', $video_url, PDO::PARAM_STR);
31 $sql->bindParam(':category_id', $subcategoria, PDO::PARAM_INT);
32 $sql->bindParam(':sub_category_id', $parent, PDO::PARAM_INT);
33 $sql->bindParam(':thumbnail', $imagen_url, PDO::PARAM_STR);
34 $sql->bindParam(':section', $tema, PDO::PARAM_INT);
35 $sql->bindParam(':meta_keywords', $meta_keywords, PDO::PARAM_STR);
36 $sql->bindParam(':meta_description', $meta_description, PDO::PARAM_STR);
37 $sql->bindParam(':short_description', $short_description, PDO::PARAM_STR);
38 $sql->bindParam(':description', $course_description, PDO::PARAM_STR);
39 $sql->bindParam(':requirements', $requirements, PDO::PARAM_STR);
40 $sql->bindParam(':level', $level, PDO::PARAM_STR);
41
42 $sql->execute();
43
44 if ($sql) {
45     $db->commit();
46     $jsondata = array("success" => true, "mensaje" => "Registro correcto :3");
47 } else {
48     $db->rollBack();
49     $jsondata = array("success" => false, "mensaje" => "Ocurrió un error al intentar guardar los datos");
50 }
51
52 header('Content-type: application/json; charset=utf-8');
53 echo json_encode($jsondata);
54 exit();
55 ?>
```

Añadir cursos:

Enlace de navegación: Proporciona un enlace para que los administradores añadan cursos.

Al hacer clic en él, se muestra la interfaz para agregar cursos.

```
<li class="nav-item active">
  <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
  <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
  <a class="nav-link" href="files/add/categorias.php">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <h1>Añadir<br>cursos</span>
  </a>
</li>
```

Generación de código de categoría aleatorio: Cuando se carga la página, se genera un código de categoría aleatorio mediante la función generateRandomCode(). Este código se asigna automáticamente al campo de entrada "Código de categoría".

```

122 </style>
123 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
124
125 <script>
126     window.onload = function() {
127         // Generar código de categoría aleatorio
128         var randomCode = generateRandomCode();
129
130         // Asignar código generado al campo de entrada
131         document.getElementById("code").value = randomCode;
132     }
133
134     function generateRandomCode() {
135         var characters = 'ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
136         var length = 10;
137         var randomCode = '';
138
139         for (var i = 0; i < length; i++) {
140             var randomIndex = Math.floor(Math.random() * characters.length);
141             randomCode += characters.charAt(randomIndex);
142         }
143
144         return randomCode;
145     }
146 </script>

```

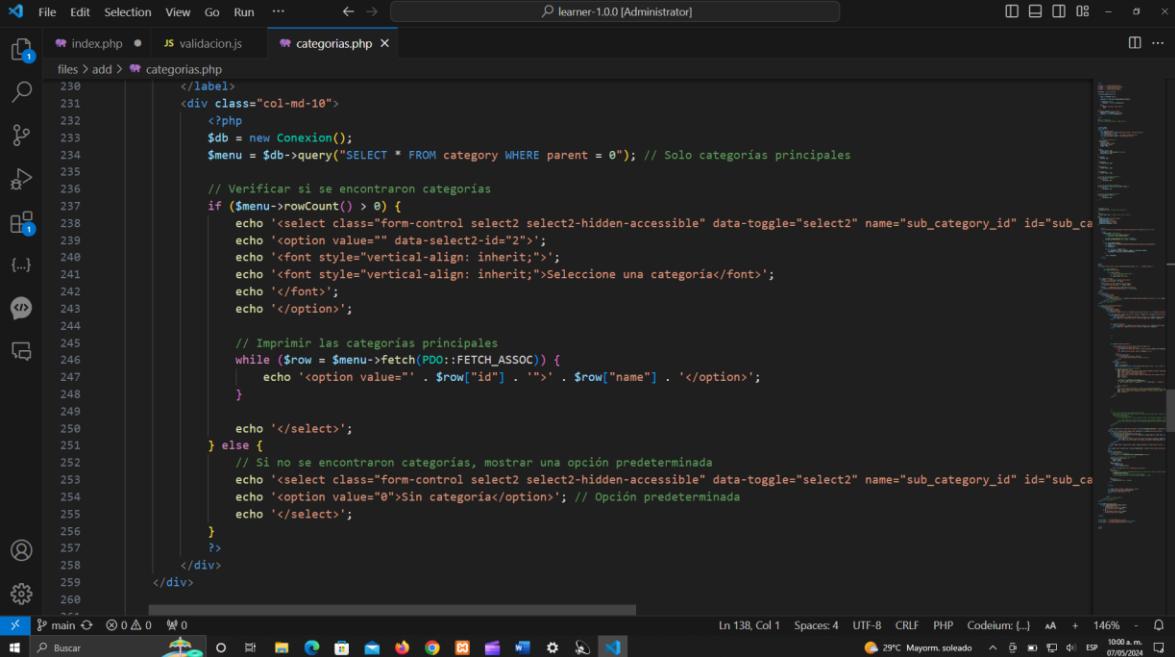
Formulario para agregar categoría: Se muestra un formulario que permite a los administradores agregar nuevas categorías. Los campos incluyen el código de categoría (generado automáticamente), el título de la categoría y la opción para seleccionar la categoría padre (si corresponde).

```

180 </div> <!-- end card body-->
181 > <!-- end card -->
182 - end col-->
183
184 <ow justify-content-center>
185 <div class="col-xl-7">
186 <div class="card">
187 <div class="card-body">
188 <div class="col-lg-12">
189 <h4 class="mb-3 header-title"><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">Formulario para agregar categoría</font></font></h4>
190 <form id="uploadForm" class="required-form" action="" method="post" enctype="multipart/form-data">
191 <div class="form-group">
192 <label for="code"><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">Código de categoría</font></font></label>
193 <input type="text" class="form-control" id="code" name="code" value="" readonly="">
194 </div>
195
196 <div class="form-group">
197 <label for="name"><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">Título de la categoría </font></font></label>
198 <input type="text" class="form-control" id="name" name="name" required="">
199 </div>
200
201 <!--
202
203
204
205
206
207
208
209
210

```

Selección de categoría padre: Se consulta la base de datos para obtener las categorías principales y se muestran en un menú desplegable. Si se encuentran categorías principales, se muestran en el menú. Si no se encuentran, se muestra una opción predeterminada para indicar que la categoría será una categoría padre.



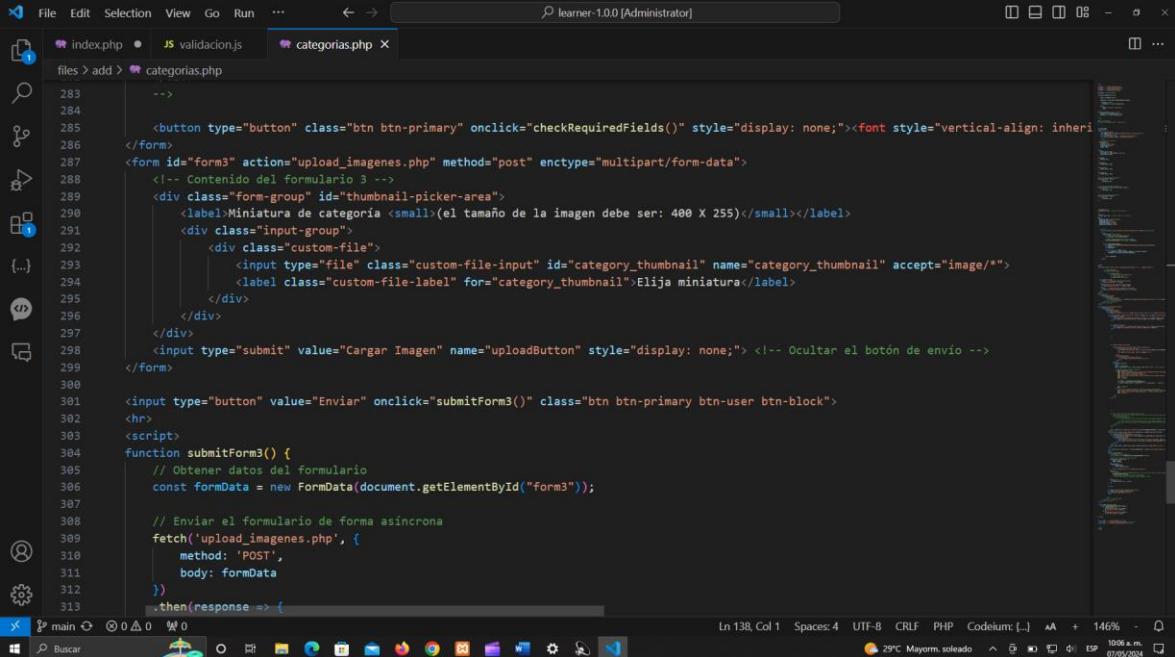
```

File Edit Selection View Go Run ... ↗ ⓘ learner-1.0.0 [Administrator]
index.php JS validacion.js categorias.php
230     </label>
231     <div class="col-md-10">
232         <?php
233             $db = new Conexion();
234             $menu = $db->query("SELECT * FROM category WHERE parent = 0"); // Solo categorias principales
235
236             // Verificar si se encontraron categorias
237             if ($menu->rowCount() > 0) {
238                 echo '<select class="form-control select2 select2-hidden-accessible" data-toggle="select2" name="sub_category_id" id="sub_cat_id">';
239                 echo '<option value="" data-select2-id="2">';
240                 echo '<font style="vertical-align: inherit;">Seleccione una categoria</font>';
241                 echo '</font>';
242                 echo '</option>';
243
244                 // Imprimir las categorias principales
245                 while ($row = $menu->fetch(PDO::FETCH_ASSOC)) {
246                     echo '<option value="' . $row['id'] . '">' . $row["name"] . '</option>';
247                 }
248
249                 echo '</select>';
250             } else {
251                 // Si no se encontraron categorias, mostrar una opcion predeterminada
252                 echo '<select class="form-control select2 select2-hidden-accessible" data-toggle="select2" name="sub_category_id" id="sub_cat_id">';
253                 echo '<option value="0">Sin categoria</option>'; // Opcion predeterminada
254                 echo '</select>';
255             }
256         ?>
257     </div>
258 </div>
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313

```

The screenshot shows a code editor with multiple tabs open: index.php, JS validacion.js, and categorias.php. The categorias.php tab is active and displays PHP code. The code uses a PDO database connection to query the 'category' table for rows where 'parent' is 0, which retrieves top-level categories. If no categories are found, it displays a default option 'Sin categoria'. The code uses the 'select2' plugin for dropdown menus. The interface includes a top bar with File, Edit, Selection, View, Go, Run, etc., and a bottom status bar showing file details like Ln 138, Col 1, and system information like temperature and weather.

Formulario para cargar miniatura de categoría: Se muestra un formulario para que los administradores carguen una miniatura de categoría. Los administradores pueden seleccionar un archivo de imagen para representar la categoría.



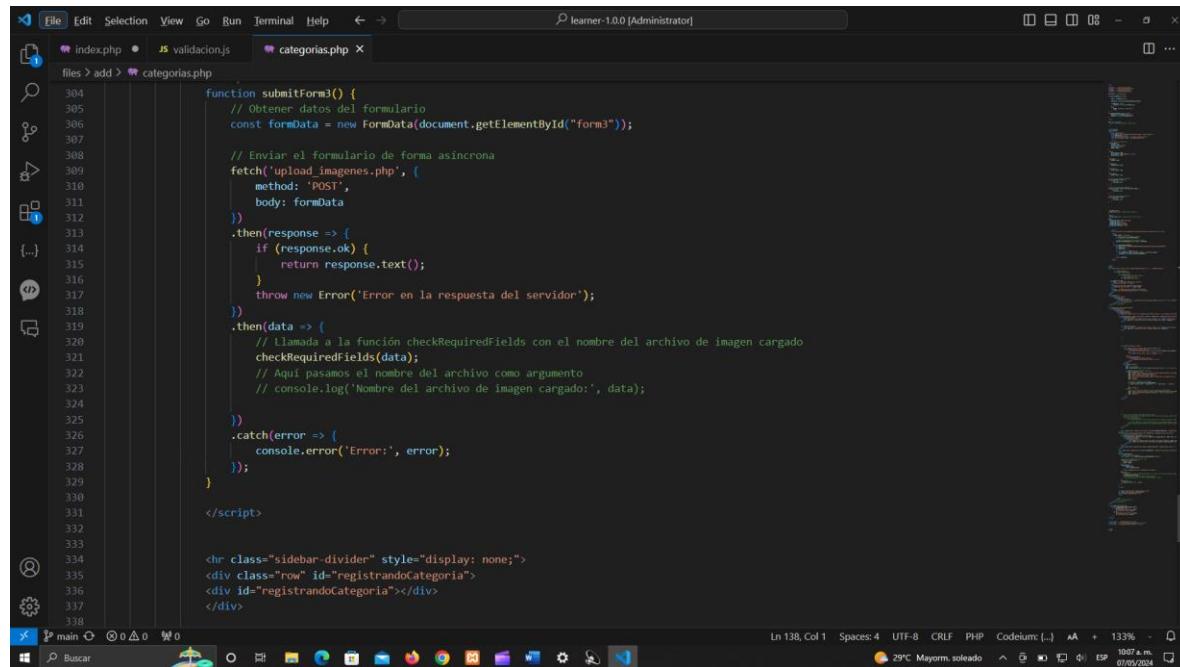
```

File Edit Selection View Go Run ... ↗ ⓘ learner-1.0.0 [Administrator]
index.php JS validacion.js categorias.php
283     -->
284
285     <button type="button" class="btn btn-primary" onclick="checkRequiredFields()" style="display: none;"><font style="vertical-align: inherit;">Cargar Imagen</font></button>
286
287     </form>
288     <!-- Contenido del formulario 3 -->
289     <div class="form-group" id="thumbnail-picker-area">
290         <label>Miniatura de categoria <small>(el tamaño de la imagen debe ser: 400 X 255)</small></label>
291         <div class="input-group">
292             <div class="custom-file">
293                 <input type="file" class="custom-file-input" id="category_thumbnail" name="category_thumbnail" accept="image/*">
294                 <label class="custom-file-label" for="category_thumbnail">Elija miniatura</label>
295             </div>
296         </div>
297     </div>
298     <input type="submit" value="Cargar Imagen" name="uploadButton" style="display: none;"> <!-- Ocultar el boton de envio -->
299
300
301     <input type="button" value="Enviar" onclick="submitForm3()" class="btn btn-primary btn-user btn-block">
302     <hr>
303     <script>
304         function submitForm3() {
305             // Obtener datos del formulario
306             const formData = new FormData(document.getElementById("form3"));
307
308             // Enviar el formulario de forma asincrona
309             fetch('upload_imagenes.php', {
310                 method: 'POST',
311                 body: formData
312             })
313             .then(response => {

```

The screenshot shows a code editor with multiple tabs open: index.php, JS validacion.js, and categorias.php. The categorias.php tab is active and displays PHP code. It includes a form for uploading a thumbnail image using a custom file input field and a button labeled 'Cargar Imagen'. Below the form, there's a script for the 'submitForm3()' function that sends a POST request to 'upload_imagenes.php' using a FormData object. The interface includes a top bar with File, Edit, Selection, View, Go, Run, etc., and a bottom status bar showing file details like Ln 138, Col 1, and system information like temperature and weather.

Envío del formulario de carga de miniatura: Cuando los administradores seleccionan una imagen para la miniatura de la categoría y hacen clic en el botón "Enviar", se llama a la función submitForm3() para enviar el formulario de forma asíncrona al servidor. La miniatura se carga en el servidor y se muestra un mensaje indicando que la categoría se ha registrado correctamente.

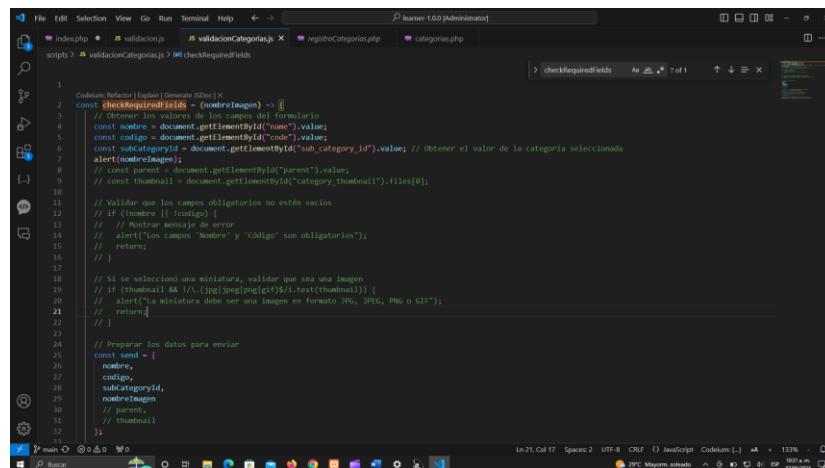


```

 304         function submitForm3() {
 305             // Obtener datos del formulario
 306             const formData = new FormData(document.getElementById("form3"));
 307
 308             // Envíar el formulario de forma asíncrona
 309             fetch("upload_imagenes.php", {
 310                 method: 'POST',
 311                 body: formData
 312             })
 313             .then(response => {
 314                 if (response.ok) {
 315                     return response.text();
 316                 }
 317                 throw new Error('Error en la respuesta del servidor');
 318             })
 319             .then(data => {
 320                 // Llamada a la función checkRequiredFields con el nombre del archivo de imagen cargado
 321                 checkRequiredFields(data);
 322                 // Aquí pasamos el nombre del archivo como argumento
 323                 // console.log("Nombre del archivo de imagen cargado:", data);
 324
 325             })
 326             .catch(error => {
 327                 console.error('Error:', error);
 328             });
 329
 330         }
 331
 332     </script>
 333
 334     <br class="sidebar-divider" style="display: none;">
 335     <div class="row" id="registrandoCategoria">
 336     <div id="registrandoCategoria"></div>
 337     </div>
 338
 339 
```

Preparación de los datos con javascript

Obtención de valores de los campos del formulario: La función obtiene los valores de los campos del formulario, como el nombre de la categoría, el código de la categoría y la categoría seleccionada.



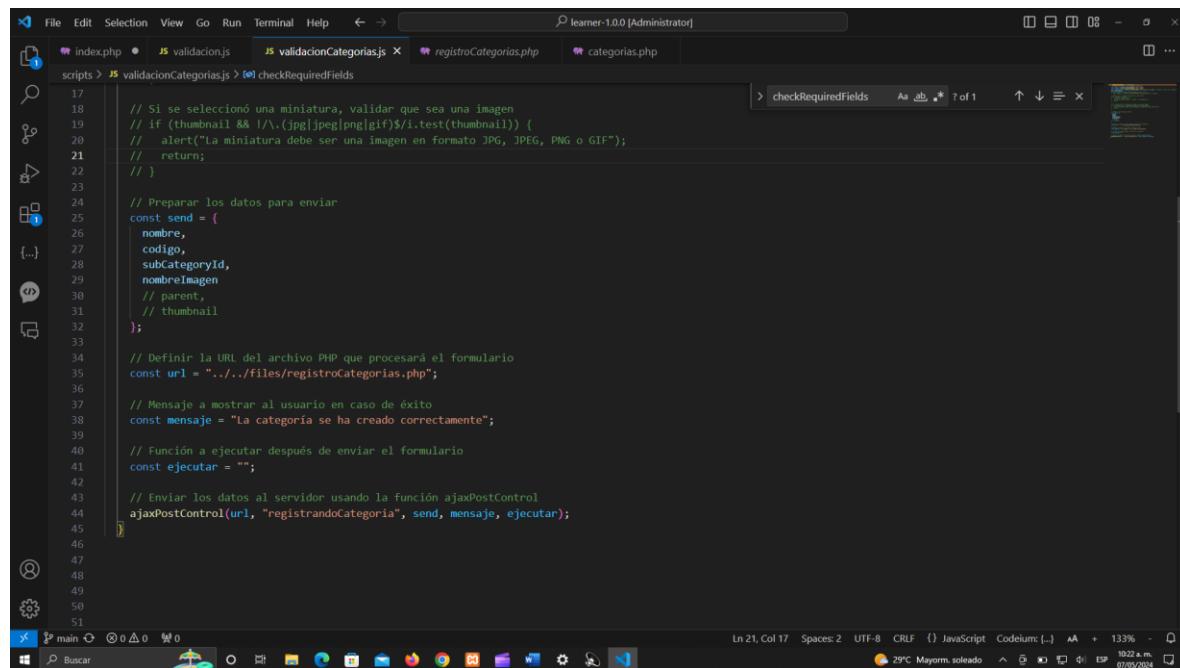
```

 1 // Definir función para validar campos
 2 const checkRequiredFields = (nombreImagen) => {
 3     // Obtener los valores de los campos del formulario
 4     const nombre = document.getElementById("name").value;
 5     const codigo = document.getElementById("code").value;
 6     const categoria = document.getElementById("sub_category_id").value; // Obtener el valor de la categoría seleccionada
 7     alert(nombre);
 8     const parent = document.getElementById("parent").value;
 9     const thumbnail = document.getElementById("category_thumbnail").files[0];
10
11     // Validar que los campos obligatorios no estén vacíos
12     // Si (nombre || !codigo) {
13     //     // Mostrar mensaje de error
14     //     alert("Los campos 'Nombre' y 'Código' son obligatorios");
15     //     return;
16     // }
17
18     // Si se seleccionó una miniatura, validar que sea una imagen
19     // Si thumbnail && (!img.jpeg||!img.png||!img.gif||!img.thumbnaill)
20     // alert("La miniatura debe ser una imagen en formato JPG, JPEG, PNG o GIF");
21     // return;
22 }
23
24 // Preparar los datos para enviar
25 const send = {
26     nombre,
27     codigo,
28     sub_category_id,
29     nombreImagen,
30     // parent,
31     // thumbnail
32 };
33 
```

Alerta de prueba: Se muestra una alerta con el nombre de la imagen que se pasa como argumento a la función. Esto sirve como una prueba para verificar que la variable nombreImagen se recibe correctamente en la función.

Validación de campos obligatorios: Se realiza una validación básica para verificar que los campos obligatorios, como el nombre y el código de la categoría, no estén vacíos. Sin embargo, esta validación está actualmente comentada en el código.

Preparación de datos para enviar: Se preparan los datos del formulario para enviar al servidor. Esto incluye el nombre de la categoría, el código de la categoría, la categoría seleccionada y el nombre de la imagen de la miniatura de la categoría.



The screenshot shows a code editor window with several tabs open. The active tab is 'JS validacionCategorias.js'. The code in the editor is as follows:

```
// Si se seleccionó una miniatura, validar que sea una imagen
// if (thumbnail && !/.(jpg|jpeg|png|gif)$/.test(thumbnail)) {
//   alert("La miniatura debe ser una imagen en formato JPG, JPEG, PNG o GIF");
//   return;
// }

// Preparar los datos para enviar
const send = {
  nombre,
  codigo,
  subCategoryId,
  nombreImagen
  // parent,
  // thumbnail
};

// Definir la URL del archivo PHP que procesará el formulario
const url = "../../files/registroCategorias.php";

// Mensaje a mostrar al usuario en caso de éxito
const mensaje = "La categoría se ha creado correctamente";

// Función a ejecutar después de enviar el formulario
const ejecutar = "";

// Enviar los datos al servidor usando la función ajaxPostControl
ajaxPostControl(url, "registrandoCategoria", send, mensaje, ejecutar);
```

The code is written in JavaScript and performs validation on a thumbnail image and prepares data for an AJAX POST request to a PHP file named 'registroCategorias.php'.

Envío de datos al servidor: Se define la URL del archivo PHP que procesará el formulario y se utiliza la función ajaxPostControl para enviar los datos al servidor.

Archivo PHP para procesar el formulario: En el archivo PHP registroCategorias.php, se reciben los datos enviados desde el cliente, se inserta la nueva categoría en la base de datos y se envía una respuesta JSON al cliente indicando si la operación fue exitosa o no.

Validación a la base de datos

Inclusión de la clase de conexión a la base de datos: Al principio del archivo, se incluye la clase de conexión a la base de datos (Conexion.php) para establecer una conexión con la base de datos.

Creación de una instancia de la clase de conexión: Se crea una nueva instancia de la clase Conexión para conectarse a la base de datos.

Obtención de variables enviadas por JavaScript: Se utilizan las funciones htmlentities y \$_POST de PHP para obtener las variables enviadas desde el cliente a través de JavaScript. Estas variables incluyen el nombre de la categoría, el código de la categoría, la categoría seleccionada y el nombre de la imagen de la miniatura de la categoría.

Inicio de una transacción: Se inicia una transacción de base de datos utilizando el método beginTransaction() de la instancia de conexión a la base de datos. Esto asegura que todas las operaciones dentro de la transacción se completen correctamente o se reviertan si ocurre algún error.

Preparación de la consulta SQL: Se prepara una consulta SQL utilizando el método prepare() de la instancia de conexión. La consulta SQL insertará los datos de la nueva categoría en la tabla category de la base de datos.

Vinculación de parámetros: Se vinculan los parámetros de la consulta SQL a las variables obtenidas anteriormente. Esto se hace utilizando el método bindParam() para evitar posibles ataques de inyección SQL y asegurar que los datos se inserten de manera segura en la base de datos.

Ejecución de la consulta SQL: Se ejecuta la consulta SQL utilizando el método execute() de la instancia de consulta preparada. Esto inserta los datos de la nueva categoría en la base de datos.

The screenshot shows a code editor with several tabs at the top: index.php, validacion.js, validacionCategorias.js, registroCategorias.php (which is the active tab), and categorias.php. Below the tabs, the file structure is shown as files > registroCategorias.php. The main content area contains the following PHP code:

```
1  <?php
2  // Incluir la clase de conexión a la base de datos
3  include("../clases/Conexion.php");
4
5  // Crear una nueva instancia de la clase Conexion
6  $db = new Conexion();
7
8  // Obtener las variables enviadas por la función JavaScript
9  $nombre = htmlentities($_POST['nombre'], ENT_NOQUOTES, "UTF-8");
10 $codigo = htmlentities($_POST['codigo'], ENT_NOQUOTES, "UTF-8");
11 $subCategoryId = htmlentities($_POST['subCategoryId'], ENT_NOQUOTES, "UTF-8");
12 $nombreImagen = htmlentities($_POST['nombreImagen'], ENT_NOQUOTES, "UTF-8");
13
14
15 // Iniciar una transacción
16 $db->beginTransaction();
17
18 // Preparar la consulta SQL para insertar la categoría
19 $sql = $db->prepare("INSERT INTO category (code, name, parent, thumbnail) VALUES (:code, :name, :parent, :thumbnail)");
20
21 // Vincular los parámetros a las variables
22 $sql->bindParam(':code', $codigo, PDO::PARAM_STR);
23 $sql->bindParam(':name', $nombre, PDO::PARAM_STR);
24 $sql->bindParam(':parent', $subCategoryId, PDO::PARAM_STR);
25 $sql->bindParam(':thumbnail', $nombreImagen, PDO::PARAM_STR);
26
27 // Ejecutar la consulta
28 $sql->execute();
29
30 if ($sql) {
31     // Registrar la transacción
32     $db->commit();
33     $jsondata = array("success" => true, "mensaje" => "Categoría creada correctamente");
34 } else {
35     // Error al insertar la categoría
36     $db->rollBack();
37     $jsondata = array("success" => false, "mensaje" => "Error al crear la categoría");
38 }
39
40 // Enviar la respuesta al cliente en formato JSON
41 header('Content-type: application/json; charset=utf-8');
42 echo json_encode($jsondata);
43 exit();
44 ?>
45
```

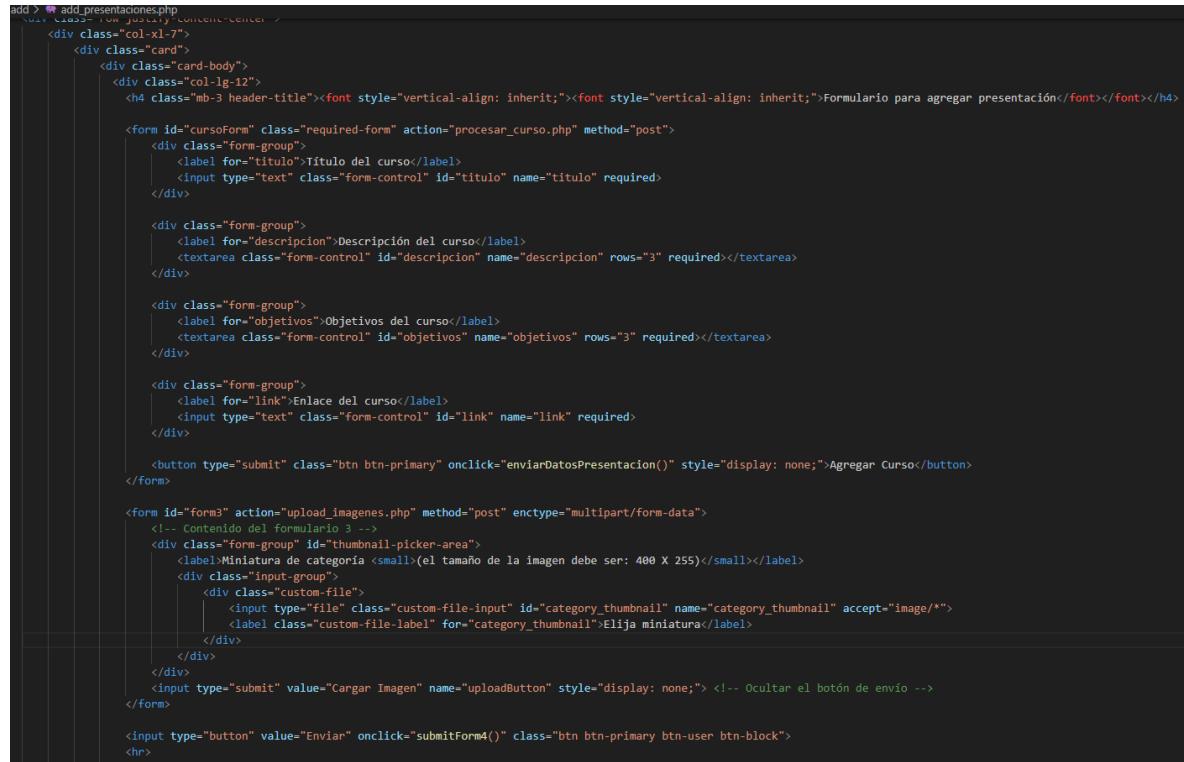
Comprobación del resultado de la operación: Se comprueba si la operación de inserción fue exitosa. Si la consulta se ejecuta correctamente, se confirma la transacción utilizando el método commit(). En caso de error, se revierte la transacción utilizando el método rollBack(). Respuesta al cliente: Se crea un arreglo \$jsondata que contiene un indicador de éxito (true o false) y un mensaje que describe el resultado de la operación. Este arreglo se convierte en formato JSON utilizando la función json_encode() y se envía al cliente utilizando el encabezado Content-type: application/json.

Añadir presentación:

Enlace para añadir presentaciones: Se ha añadido un enlace en el menú de navegación que redirige a add_presentaciones.php.

```
<li class="nav-item active">
  <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
  <!-- <a class="nav-link" href="javascript:anadirCursos()" onclick="mostrarAnadirCursos()"> -->
  <a class="nav-link" href="files/add/add_presentaciones.php">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <h1>Añadir<br>presentaciones</span>
  </a>
</li>
```

Formulario para agregar presentaciones: Se ha creado un formulario con campos para el título del curso, descripción, objetivos, enlace y miniatura de la presentación. También hay un botón para enviar el formulario. Este formulario está preparado para ser enviado de forma asíncrona a través de AJAX.



The screenshot shows a web page titled "Formulario para agregar presentación". It contains several input fields: "Título del curso" (text), "Descripción del curso" (textarea), "Objetivos del curso" (textarea), "Enlace del curso" (text), and a "Upload" section for "Imagen de categoría" (file input). A "Submit" button is present at the bottom of the form.

```
<add> ● add_presentaciones.php
<div class="row justify-content-center">
  <div class="col-xl-7">
    <div class="card">
      <div class="card-body">
        <div class="mb-3 header-title"><font style="vertical-align: inherit;"><font style="vertical-align: inherit;">Formulario para agregar presentación</font></font></div>
        <form id="cursoForm" class="required-form" action="procesar_curso.php" method="post">
          <div class="form-group">
            <label for="titulo">Título del curso</label>
            <input type="text" class="form-control" id="titulo" name="titulo" required>
          </div>

          <div class="form-group">
            <label for="descripcion">Descripción del curso</label>
            <textarea class="form-control" id="descripcion" name="descripcion" rows="3" required></textarea>
          </div>

          <div class="form-group">
            <label for="objetivos">Objetivos del curso</label>
            <textarea class="form-control" id="objetivos" name="objetivos" rows="3" required></textarea>
          </div>

          <div class="form-group">
            <label for="link">Enlace del curso</label>
            <input type="text" class="form-control" id="link" name="link" required>
          </div>

          <button type="submit" class="btn btn-primary" onclick="enviarDatosPresentacion()" style="display: none;">Agregar Curso</button>
        </form>

        <form id="form3" action="upload_imagenes.php" method="post" enctype="multipart/form-data">
          <!-- Contenido del formulario 3 -->
          <div class="form-group" id="thumbnail-picker-area">
            <label>Miniatura de categoría <small>(el tamaño de la imagen debe ser: 400 X 255)</small></label>
            <div class="input-group">
              <div class="custom-file">
                <input type="file" class="custom-file-input" id="category_thumbnail" name="category_thumbnail" accept="image/*">
                <label class="custom-file-label" for="category_thumbnail">Elige miniatura</label>
              </div>
            </div>
          </div>
          <input type="submit" value="Cargar Imagen" name="uploadButton" style="display: none;"> <!-- Ocultar el botón de envío -->
        </form>

        <input type="button" value="Enviar" onclick="submitForm4()" class="btn btn-primary btn-user btn-block">
        <hr>
      </div>
    </div>
  </div>
</div>
```

Lista de presentaciones cursos: Se muestra una lista de las presentaciones de cursos existentes en el panel derecho de la página. Cada presentación se muestra con su título y un enlace para ver más detalles.

The screenshot shows a code editor interface with two tabs: 'index.php' and 'add_presentaciones.php'. The 'add_presentaciones.php' tab is active, displaying PHP code. The code includes SQL queries to select data from a 'presentaciones' table, loops through the results to generate a list of presentations, and includes a dropdown menu with an 'eliminar-usuario' option. The code is well-structured with comments and CSS classes applied to the generated HTML.

```
283     $sql2 = "SELECT * FROM presentaciones"; // "WHERE", no "WERE"
284
285     $result2 = $db->query($sql2);
286
287
288     <div class="container-fluid">
289     <div class="row">
290         <div class="col-1g-9">
291             <!-- Contenido principal de tu página -->
292             iv>
293             <div class="col-1g-3">
294                 <!-- Lista de presentaciones cursos -->
295                 <?php
296                     $sql2 = "SELECT * FROM presentaciones"; // "WHERE", no "WERE"
297                     $result2 = $db->query($sql2);
298                 ?>
299                 <div class="card2 p-3">
300                     <h2 class="line-bottom text-center mb-4">Presentaciones cursos</h2>
301                     <ul class="list-group">
302                         <?php foreach($result2 as $m): ?>
303                         <div class="dropdown-item">
304                             <a class="dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" href="#"></a>
305                             <div class="dropdown-menu dropdown-menu-right shadow animated--fade-in" aria-labelledby="dropdownMenuLink">
306                                 <div class="dropdown-header">Opciones</div>
307                                 <!-- <a class="dropdown-item eliminar-usuario" href="javascript:eliminandoUsuario(<?php echo $usu["id"]; ?>)>Eliminar Usu</a> -->
308                                 </div>
309                         </div>
310                         <li class="list-group-item mb-2">
311                             <a href=".../presentacion.php?link=<?php echo $m["link"]; ?>">
312                                 <h3><?php echo $m["titulo"]; ?></h3>
313                                 <span>courses</span>
314                             </a>
315                         </li>
316                     <?php endforeach; ?>
317                     </ul>
318                 </div>
319             </div>
320         </div>
321     </div>
322
323     <script>
324         document.addEventListener("DOMContentLoaded", function() {
325             // Obtener referencias a todos los menús desplegables
326         });
327     </script>
328 
```

This screenshot shows the same code editor interface as the first one, but with some changes in the dropdown menu logic. The 'add_presentaciones.php' tab is active, displaying PHP code. The code is identical to the first screenshot up to line 313, but then it includes a script block that adds an event listener for 'DOMContentLoaded'. This listener calls a function that retrieves references to all dropdown menus. The rest of the code remains the same, including the 'eliminar-usuario' option in the dropdown menu.

```
299     <div class="card2 p-3">
300         <h2 class="line-bottom text-center mb-4">Presentaciones cursos</h2>
301         <ul class="list-group">
302             <?php foreach($result2 as $m): ?>
303             <div class="dropdown-item">
304                 <a class="dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" href="#"></a>
305                 <div class="dropdown-menu dropdown-menu-right shadow animated--fade-in" aria-labelledby="dropdownMenuLink">
306                     <div class="dropdown-header">Opciones</div>
307                     <!-- <a class="dropdown-item eliminar-usuario" href="javascript:eliminandoUsuario(<?php echo $usu["id"]; ?>)>Eliminar Usu</a> -->
308                     </div>
309                 </div>
310                 <li class="list-group-item mb-2">
311                     <a href=".../presentacion.php?link=<?php echo $m["link"]; ?>">
312                         <h3><?php echo $m["titulo"]; ?></h3>
313                         <span>courses</span>
314                     </a>
315                 </li>
316             <?php endforeach; ?>
317             </ul>
318         </div>
319     </div>
320
321     <script>
322         document.addEventListener("DOMContentLoaded", function() {
323             // Obtener referencias a todos los menús desplegables
324         });
325     </script>
326 
```

Preparación de los datos con JavaScript

Script para enviar el formulario de forma asíncrona: Se ha proporcionado un script JavaScript para enviar el formulario de forma asíncrona. Este script utiliza la función fetch para enviar los datos del formulario al archivo upload_imagenes.php que procesará la carga de la imagen de la miniatura. Una vez que se completa la carga de la imagen, se llama a la función enviarDatosPresentacion() con el nombre del archivo de imagen cargado.

```
File Edit Selection View Go Run ... learner-1.0.0 [Administrator]
EXPLORER ... index.php JS validacion.js add_presentaciones.php
LEARNER-1.0.0 ...
> administrador
> alertifyjs
> Base de datos(DB)
> clases
> css
> ejem
files ...
> add ...
> add_courses_files
...
> Categorías _Sistema de ge...
> SCRIPTS_validar_curso_en...
add_courses.php
add_presentaciones.php
...
Categorías _Sistema de ge...
categorias.php
deleteupdate_courses2 cop...
GUARDADO.PHP
index.php
js modal.js
notas verdes
PRUEBA.php
update_courses.php
update_courses2 borrado.p...
update_courses2 copy.php
update_courses2.php
...
OUTLINE
TIMELINE
main 0 0 0 0 Buscar
Ln 256, Col 35 Spaces 4 UTF-8 CRLF PHP Codeium: [...] xA + 146% - 29°C Mayorm. soleado 11:49 a.m. ESP 01/05/2024
```

```
function submitForm4() {
    // Obtener datos del formulario
    const formData = new FormData(document.getElementById("form3"));

    // Enviar el formulario de forma asíncrona
    fetch('../files/add/upload_imagenes.php', {
        method: 'POST',
        body: formData
    })
    .then(response => {
        if (response.ok) {
            return response.text();
        }
        throw new Error('Error en la respuesta del servidor');
    })
    .then(data => {
        // Llamada a la función checkRequiredFields con el nombre del archivo de imagen cargado
        enviarDatosPresentación(data);
        // Aquí pasamos el nombre del archivo como argumento
        // console.log('Nombre del archivo de imagen cargado:', data);

    })
    .catch(error => {
        console.error('Error:', error);
    });
}

</script>
</div>
```

Obtener los valores del formulario:

La función comienza obteniendo los valores de los campos del formulario de presentaciones. Estos valores incluyen el título del curso (titulo), la descripción del curso (descripcion), los objetivos del curso (objetivos), y el enlace del curso (link). También se recibe el nombre de la imagen de la miniatura (nombrelImagen), que presumiblemente se ha cargado previamente utilizando el formulario adjunto.

Procesar el enlace del curso:

Se realiza un procesamiento adicional en el enlace del curso (link). Se busca la posición de la cadena "/edit" en el enlace. Si se encuentra, se recorta el enlace para eliminar cualquier parte después de "/edit". Esto sugiere que el enlace puede contener información adicional que no es necesaria para el proceso de presentación.

Crear un objeto con los datos:

Se crea un objeto llamado datos que contiene todos los datos recopilados del formulario, así como el nombre de la imagen de la miniatura. Este objeto se utilizará para enviar los datos al servidor.

```

File Edit Selection View Go Run Terminal Help
... index.php JS validacion.js JS validacionPresentaciones.js ...
scripts > JS validacionPresentaciones.js > enviarDatosPresentacion
CodeLens: Refactor | Explain | Generate JSONDoc | X
1 const enviarDatosPresentacion = (nombreImagen) => {
2   // Obtener los valores de los campos del formulario
3   const titulo = document.getElementById("titulo").value;
4   const descripcion = document.getElementById("descripcion").value;
5   const objetivos = document.getElementById("objetivos").value;
6   let link = document.getElementById("link").value;
7   // alert("nombreImagen");
8   // alertify.success("El curso se ha agregado correctamente");
9   // Encontrar la posición de "/edit"
10  const indiceEdit = link.indexOf("/edit");
11  // Obtener el enlace sin la parte después de "/edit"
12  if (indiceEdit != -1) {
13    link = link.substring(0, indiceEdit);
14  }
15
16  // Crear un objeto con los datos a enviar
17  const datos = {
18    titulo,
19    descripcion,
20    objetivos,
21    link,
22    nombreImagen
23  };
24
25  // Definir la URL del archivo PHP que procesará los datos
26  const url = "../registroPresentaciones.php";
27
28  // Mensaje a mostrar al usuario en caso de éxito
29  const mensaje = alertify.success("El curso se ha agregado correctamente");
30
31  // Función a ejecutar después de enviar los datos
32  const ejecutar = "";
33
34  // Enviar los datos al servidor usando una función ajaxPostControl (debes definirla o usar una alternativa como fetch)
35  ajaxPostControl(url, "agregandoPresentaciones", datos, mensaje, ejecutar);
36
37

```

Definir la URL de destino:

Se define la URL del archivo PHP que procesará los datos del formulario. En este caso, la URL se establece como "../registroPresentaciones.php".

Mensaje de éxito:

Se define un mensaje de éxito que se mostrará al usuario si la presentación se agrega correctamente. Este mensaje utiliza una función `alertify.success()` para mostrar una notificación al usuario.

Enviar los datos al servidor:

Los datos recopilados se envían al servidor utilizando una función `ajaxPostControl()`. Esta función probablemente se encarga de realizar una solicitud HTTP POST al servidor y procesar la respuesta. La función espera la URL de destino, un identificador para el proceso (en este caso, "agregandoPresentaciones"), los datos a enviar, el mensaje de éxito y una función a ejecutar después de enviar los datos.

Validación a la base de datos

Incluir la clase de conexión a la base de datos:

Se incluye el archivo que contiene la clase de conexión a la base de datos (Conexion.php). Esto permite establecer una conexión con la base de datos utilizando los métodos y propiedades de la clase.

Crear una nueva instancia de la clase Conexion:

Se crea una nueva instancia de la clase Conexion, que establece la conexión con la base de datos.

Obtener los valores enviados por JavaScript:

Se recuperan los valores enviados por la función JavaScript mediante una solicitud POST. Estos valores incluyen el título del curso (titulo), la descripción del curso (descripcion), los objetivos del curso (objetivos), el enlace del curso (link), y el nombre de la imagen de la presentación (nombreImagen). Estos valores se obtienen utilizando el método \$_POST.

Iniciar una transacción:

Se inicia una transacción utilizando el método beginTransaction() de la instancia de la conexión a la base de datos. Esto garantiza que todas las consultas SQL realizadas dentro de esta transacción se completen correctamente o se deshagan en caso de error.

Preparar la consulta SQL:

Se prepara una consulta SQL utilizando el método prepare() de la instancia de la conexión. La consulta SQL es una instrucción de inserción que inserta los valores del formulario en la tabla de presentaciones.

```
<?php
// Incluir la clase de conexión a la base de datos
include("../clases/Conexion.php");
// Crear una nueva instancia de la clase Conexion
$db = new Conexion();
// Obtener los valores enviados por la función JavaScript
$titulo = htmlentities($_POST['titulo'], ENT_QUOTES, "UTF-8");
$descripcion = htmlentities($_POST['descripcion'], ENT_QUOTES, "UTF-8");
$objetivos = htmlentities($_POST['objetivos'], ENT_QUOTES, "UTF-8");
$link = htmlentities($_POST['link'], ENT_QUOTES, "UTF-8");
$nombreImagen = htmlentities($_POST['nombreImagen'], ENT_QUOTES, "UTF-8");
// Iniciar una transacción
$db->beginTransaction();
try {
    // Preparar la consulta SQL para insertar el curso
    $sql = $db->prepare("INSERT INTO presentaciones (titulo, descripcion, objetivos, link, nombre_imagen) VALUES (:titulo, :descripcion, :objetivos, :link, :nombreImagen)");
    // Vincular los parámetros a las variables
    $sql->bindParam(':titulo', $titulo, PDO::PARAM_STR);
    $sql->bindParam(':descripcion', $descripcion, PDO::PARAM_STR);
    $sql->bindParam(':objetivos', $objetivos, PDO::PARAM_STR);
    $sql->bindParam(':link', $link, PDO::PARAM_STR);
    $sql->bindParam(':nombreImagen', $nombreImagen, PDO::PARAM_STR);
    // Ejecutar la consulta
    $sql->execute();
    // Verificar si se insertó correctamente
    if ($sql) {
        // Confirmar la transacción
        $db->commit();
        $jsondata = array("success" => true, "mensaje" => "La presentación se ha agregado correctamente");
    }
}
catch (Exception $e) {
    // Deshacer la transacción en caso de error
    $db->rollBack();
    $jsondata = array("success" => false, "mensaje" => "Ocurrió un error al agregar la presentación");
}
```

Vincular los parámetros:

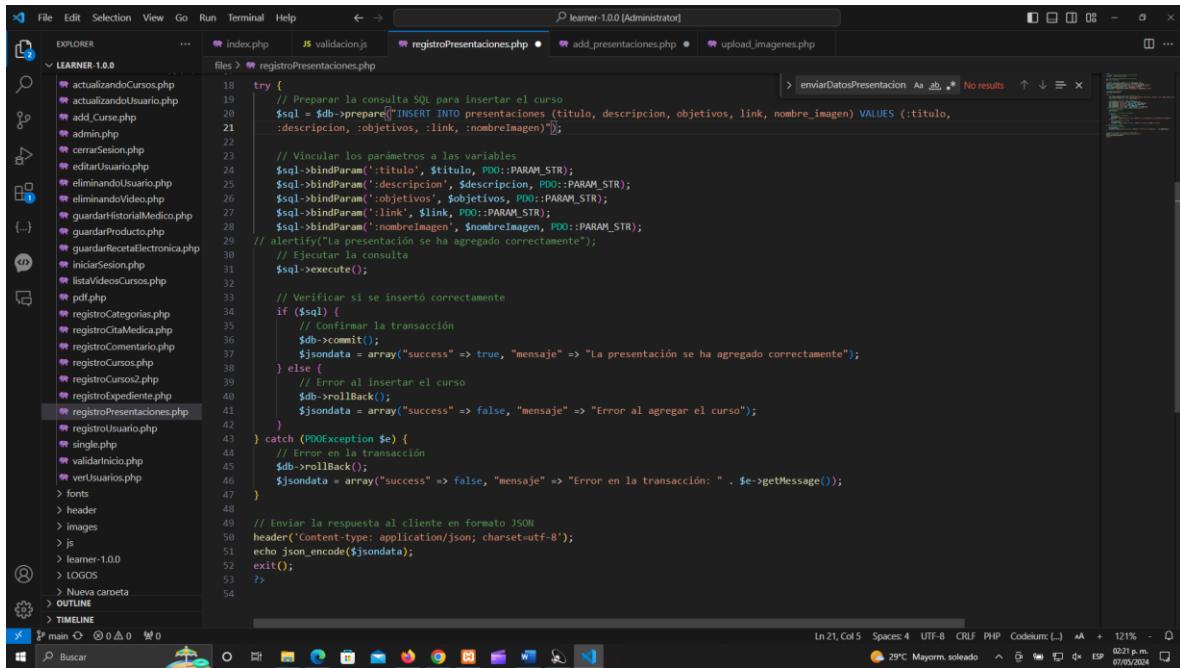
Se vinculan los parámetros de la consulta SQL a las variables obtenidas del formulario. Esto se hace utilizando el método `bindParam()` de la instancia de la consulta SQL.

Ejecutar la consulta SQL:

Se ejecuta la consulta SQL utilizando el método `execute()` de la instancia de la consulta SQL.

Verificar el resultado de la inserción:

Se verifica si la inserción fue exitosa. Si la inserción fue exitosa, se confirma la transacción utilizando el método `commit()` de la instancia de la conexión. En caso de error, se deshace la transacción utilizando el método `rollBack()` de la instancia de la conexión.



```
try {
    // Preparar la consulta SQL para insertar el curso
    $sql = $db->prepare("INSERT INTO presentaciones (titulo, descripcion, objetivos, link, nombre_imagen) VALUES (:titulo, :descripcion, :objetivos, :link, :nombreImagen)");
    $sql->bindParam(':titulo', $titulo, PDO::PARAM_STR);
    $sql->bindParam(':descripcion', $descripcion, PDO::PARAM_STR);
    $sql->bindParam(':objetivos', $objetivos, PDO::PARAM_STR);
    $sql->bindParam(':link', $link, PDO::PARAM_STR);
    $sql->bindParam(':nombreImagen', $nombreImagen, PDO::PARAM_STR);

    // Vincular los parámetros a las variables
    $sql->execute();
    // Ejecutar la consulta
    $sql->commit();

    // Verificar si se insertó correctamente
    if ($sql) {
        // Confirmar la transacción
        $sql->commit();
        $jsondata = array("success" => true, "mensaje" => "La presentación se ha agregado correctamente");
    } else {
        // Error al insertar el curso
        $sql->rollback();
        $jsondata = array("success" => false, "mensaje" => "Error al agregar el curso");
    }
} catch (PDOException $e) {
    // Error en la transacción
    $sql->rollback();
    $jsondata = array("success" => false, "mensaje" => "Error en la transacción: " . $e->getMessage());
}

// Enviar la respuesta al cliente en formato JSON
header('Content-type: application/json; charset=utf-8');
echo json_encode($jsondata);
exit();
}
```

Enviar la respuesta al cliente:

Se envía la respuesta al cliente en formato JSON, indicando si la operación fue exitosa o si ocurrió algún error. Esto se hace utilizando el método `json_encode()` para convertir un array asociativo en formato JSON y enviando los encabezados HTTP correspondientes.

CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA PROFESIONAL ADQUIRIDA.

A través de este proyecto, se logró diseñar, desarrollar e implementar una plataforma robusta que proporciona capacitación de alta calidad y personalizada para el personal del resort. Basándonos en la experiencia adquirida durante el desarrollo de este proyecto, puedo realizar las siguientes recomendaciones para optimizar la eficacia y el rendimiento continuo de la plataforma E-Learning:

- 1. Actualización Continua:** Se recomienda mantener el contenido de capacitación actualizado y relevante para garantizar su pertinencia y utilidad a lo largo del tiempo. Esto incluye la incorporación de nuevas tendencias, mejores prácticas y cambios en los estándares de la industria hotelera.
- 2. Retroalimentación Constante:** Es importante recopilar y analizar la retroalimentación de los usuarios de manera regular para identificar áreas de mejora y realizar ajustes en la plataforma según sea necesario. Se debe fomentar una cultura de retroalimentación abierta y constructiva entre los colaboradores y el equipo responsable de la plataforma.
- 3. Capacitación Continua del Personal:** Se recomienda ofrecer programas de capacitación y desarrollo continuo para el personal encargado de administrar y mantener la plataforma, así como para los usuarios finales. Esto garantizará que todos estén familiarizados con las funciones y características de la plataforma y puedan aprovechar al máximo sus capacidades.
- 4. Evaluación de Impacto:** Se sugiere realizar evaluaciones periódicas del impacto de la plataforma en la calidad del servicio ofrecido por Grand Velas Riviera Maya y en el desarrollo profesional del personal. Esto permitirá medir el retorno de la inversión del proyecto y realizar ajustes estratégicos según sea necesario mediante gráficas.

5. Añadiendo Reconocimientos al Terminar un Curso

Agregar reconocimientos por completar un curso puede otorgarle un significado considerable. Grand Velas reconoce así tu aprendizaje y enseñanza en el sector hotelero.

6. Subir archivos:

La implementación de funcionalidades que permitan a los usuarios subir y visualizar documentos directamente desde la página web. Esto proporcionará un valioso respaldo al temario, permitiendo a los usuarios acceder a materiales complementarios como notas de

clase, ejercicios adicionales, o incluso presentaciones y documentos relacionados. Además, la capacidad de visualizar estos documentos en línea brinda una experiencia de aprendizaje más completa y dinámica, facilitando la comprensión de los conceptos presentados en el temario principal. Esto no solo enriquecerá el contenido educativo ofrecido, sino que también mejorará la experiencia del usuario y promoverá una mayor participación e interacción con el material.

7. Integración con Google Gemini:

Esto implica conectar tu plataforma con la API de Gemini para aprovechar sus capacidades de IA conversacional. Podrías usar Gemini para:

Crear un chatbot: Para guiar a los usuarios, responder preguntas sobre la plataforma y el contenido, o incluso ofrecer tutorías personalizadas.

Generar contenido dinámico: Gemini podría crear resúmenes de cursos, quizzes, o incluso scripts para videos, adaptándose al nivel de cada usuario.

8. Apartado de comentarios con ChatGPT3_Turbo:

Dándole contesto del video del chatbot con el API de ChatGPT3_Turbo, los comentarios adquieren una relevancia crucial en el proceso educativo al facilitar la resolución de dudas y promover una comunicación fluida entre los usuarios. Este canal de retroalimentación ofrece un espacio propicio para plantear preguntas y recibir respuestas claras, enriqueciendo así la experiencia de aprendizaje al proporcionar claridad sobre los temas tratados. Además, los comentarios fomentan un ambiente colaborativo donde los usuarios pueden intercambiar ideas, compartir conocimientos y construir una comunidad de aprendizaje sólida y enriquecedora, todo ello con el respaldo y la eficacia del chatbot ChatGPT3_Turbo.
(actualmente lo uso en local en Python)

ver: https://github.com/Uicab04/Python/tree/main/GPT_turbo

9. Experiencia Profesional Adquirida:

Participar en este proyecto de residencia profesional ha sido una experiencia enriquecedora y altamente valiosa. He tenido la oportunidad de aplicar mis conocimientos teóricos en un entorno práctico y real, colaborando con un equipo multidisciplinario para alcanzar objetivos concretos y tangibles. Además, he adquirido habilidades clave en el diseño y desarrollo de plataformas E-Learning, gestión de proyectos, trabajo en equipo y comunicación efectiva. Esta experiencia ha fortalecido mi perfil profesional y me ha preparado para enfrentar nuevos

desafíos en mi carrera profesional en el campo de la tecnología educativa y la formación de recursos humanos además que tengo mucho entusiasmo por continuar con proyectos de educación con inteligencia artificial para la enseñanza más adaptada o chatbots que resuelven dudas al instante.

10. Analizar el progreso del usuario: Gemini podría identificar patrones de aprendizaje y sugerir contenido adicional o rutas de aprendizaje personalizadas.

Buscador de Cursos: Debes implementar una funcionalidad que permita a los usuarios buscar cursos de forma eficiente. Esto requiere:

Un campo de búsqueda visible: Que permita a los usuarios introducir palabras clave.

Un algoritmo de búsqueda: Que compare las palabras clave con la información de los cursos (título, descripción, etiquetas, etc.).

Una página de resultados: Que muestre los cursos relevantes en un formato claro y conciso.

11. Permisos de Administrador para Subir Videos: Necesitas modificar tu sistema de roles y permisos para que algunos usuarios puedan subir videos. Esto implica:

Crear un rol de "Editor" o "Creador de Contenido": Con permisos específicos para subir y gestionar videos.

Implementar un panel de administración de videos: Que permita a los editores subir, editar, eliminar y organizar los videos.

Asegurar la seguridad: Solo los usuarios con el rol adecuado deberían tener acceso a estas funciones.

COMPETENCIAS DESARROLLADAS Y/O APLICADAS.

Durante el desarrollo de este proyecto de residencia profesional en Grand Velas Riviera Maya, se han desarrollado y aplicado diversas competencias clave. Estas competencias abarcan tanto habilidades técnicas específicas como habilidades interpersonales y de gestión. A continuación, se detallan algunas de las competencias desarrolladas y/o aplicadas:

- **Gestión de Proyectos:** Se ha adquirido experiencia en la planificación, ejecución y control de proyectos, siguiendo un enfoque estructurado y metodológico para garantizar la entrega exitosa de resultados dentro de los plazos establecidos y el presupuesto asignado.
- **Tecnologías de la Información y la Comunicación (TIC):** Se ha adquirido experiencia en el uso y aplicación de diversas herramientas y tecnologías relacionadas con la creación y gestión de plataformas E-Learning, incluyendo sistemas de gestión del aprendizaje (LMS), software de autoría, y herramientas de comunicación en línea.
- **Trabajo en Equipo:** Se ha trabajado en colaboración con un equipo multidisciplinario, incluyendo ingenieros en sistemas, tecnología de la información, y gestión hotelera, demostrando habilidades para comunicarse de manera efectiva, resolver problemas y tomar decisiones en conjunto para alcanzar objetivos comunes.
- **Adaptabilidad y Resolución de Problemas:** Se ha demostrado capacidad para adaptarse a cambios en el entorno de trabajo y abordar de manera proactiva los desafíos y obstáculos que han surgido durante el desarrollo del proyecto, encontrando soluciones creativas y efectivas para superarlos.
- **Comunicación Efectiva:** Se han mejorado habilidades de comunicación oral y escrita, tanto en español como en inglés, para transmitir información de manera clara, concisa y persuasiva a diferentes audiencias, incluyendo colegas, supervisores y clientes.
- Este proyecto de residencia profesional ha proporcionado una oportunidad invaluable para el desarrollo y aplicación de competencias clave que son esenciales para el éxito en el campo de la tecnología educativa, la gestión de proyectos y el desarrollo de recursos humanos. Estas competencias son transferibles y pueden ser aplicadas en una amplia variedad de contextos profesionales en el futuro.

FUENTES DE INFORMACIÓN.

- Morra, D. (2020). Gamification for learning: A framework to design, develop, implement, and evaluate gamified learning experiences (2nd ed.). Hershey, PA: IGI Global.
- Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Cambridge, MA: Harvard University Press.
- Mayer, R. E. (2014). Multimedia learning (2nd ed.). New York, NY: Routledge.
- Reigeluth, C. M. (2013). Instructional design: Theories and models (3rd ed.). New York, NY: Routledge.
- Chickering, A. W., & Reisser, L. (1993). Education and identity (2nd ed.). San Francisco, CA: Jossey-Bass.
- Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social motivation, and well-being. American Psychologist, 55(1), 68.
- Pashler, H. E. (1998). Attention in human learning and cognitive behavior. Mahwah, NJ: Lawrence Erlbaum Associates.
- Schacter, D. L., Gilbert, D. T., & Winocur, G. (2011). Memory (3rd ed.). New York, NY: Norton & Company.
- Bartolomé-Pina, A., y Steffens, K. (2015). ¿ Son los MOOC una alternativa de aprendizaje?=are MOOCs promising learning environments?Comunicar, 22(44), 91-99.
- Aguirre, M.C. (2019) Uso de la plataforma Moodle y el desempeño académico de los estudiantes de la carrera de Electrónica Industrial en la asignatura de Matemática I, de la Escuela de Ingeniería en el Instituto Superior Tecnológico Privado CIBERTEC; 2017-II [tesis de maestría, Universidad Nacional Mayor de San Marcos]. <http://bit.ly/3u66Ypk>
- Álvarez, E. L. (2018) La plataforma MOODLE en el aprendizaje de la asignatura de informática en estudiantes de la escuela universitaria de educación a distancia de la Universidad Nacional Federico Villarreal - año 2015 [tesis de, Universidad Nacional Federico Villarreal]. <http://bit.ly/3f21Y0s>.
- Sánchez, I. (2009) Plataforma educativa MOODLE. Administración y gestión. España. Editorial y Publicaciones RA-MA S.A.
- Moodle. (n.d.). Moodle. Recuperado de <https://moodle.org/>
- Docebo. (n.d.). Docebo. Recuperado de <https://www.docebo.com/products/learn-lms/>

- Cornerstone OnDemand. (n.d.). Cornerstone OnDemand. Recuperado de <https://www.cornerstoneondemand.com/>
- Totara. (n.d.). Totara. Recuperado de <https://www.totara.com/>
- SAP. (n.d.). SAP Learning Cloud. Recuperado de <https://learning.sap.com/>
- World Wide Web Consortium. (n.d.). HTML 5.2. Recuperado de <https://html.spec.whatwg.org/>
- World Wide Web Consortium. (n.d.). Cascading Style Sheets (CSS) (3a ed.). Recuperado de <https://www.w3.org/TR/CSS/>
- Mozilla Developer Network. (n.d.). JavaScript. Recuperado de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- World Wide Web Consortium. (n.d.). Asynchronous JavaScript and XML (AJAX) (2a ed.). Recuperado de <https://www.w3.org/TR/XMLHttpRequest/>
- The PHP Group. (n.d.). Manual de PHP. Recuperado de <https://www.php.net/manual/es>
- National Institute of Standards and Technology. (2019). FIPS PUB 180-4: Secure Hash Standard (SHA-256) (5a ed.).

ANEXOS.



