|  |  |  |
| --- | --- | --- |
| **Objetivos**   1. Realizar un ejercicio que permita incorporar java web 2. Almacenar información en la memoria temporal 3. Serializar la información de un aplicativo 4. Demostrar la comunicación en la web | **Número de participantes**  participante.jpgparticipante.jpg  Jherson Daniel Revelo  Victor Manuel Pozo | **Materiales**   * Internet * Eclipse o Netbeans * Apache Tomcat * Maven |
| **Duración de la actividad**  **4:00** |
| **Proyecto:**  GESTOR DE VIDEO TUTORIALES.  **Descripción:**  Construir una sistema gestor de tutoriales, cada tutorial debe almacenar la información de una categoría de una categoría que puede ser (Lógica de programación, flutter, Node js……..) el acceso al tutorial se da a partir de una URL, de igual forma establecer una prioridad de lectura (1-10), estado (Revisar, Por Revisar)  1.- Un documento con requerimientos funcionales.  2.- Un diagrama de clases.  3.- Requerimientos técnicos.  4.- Diagrama relacional.  5.- Permitir que se ingrese un tutorial a la base de datos.  6.- Permitir que se liste la información de todos los tutoriales almacenados en la base de datos.  **Autoría (Brian Castro – Samir Sam boni )**   |  |  | | --- | --- | | Proyecto Curso: | Estructuras de datos II | | Ejercicio: | Aplicación web: Gestor de tutoriales. | | Autores: | 1. Permitir que se ingrese un tutorial a la base de datos. | | 2.Permitir que se liste la información de todos los tutoriales almacenados en la BD. | |  | | Fecha realización: | 17/04/2024 |   **Listado de Requerimientos:**   |  |  | | --- | --- | | Nombre | R1- INGRESO DE TUTORIAL. | | Resumen | El sistema debería permitir a los usuarios agregar nuevos cursos especificando la categoría, la URL, la publicación principal y el estado del curso. | | Entradas | * Categoría: se puede seleccionar de una lista predefinida (por ejemplo, Programación lógica, Flutter, Node.js, etc.).• URL: una cadena de texto que representa la URL del curso.• Nota clave: un número entero en el rango de 1 a 1. 10.• Estado: Puede elegir entre "Revisando" y "A revisar".. | |  |  | | Resultados | El curso se guarda en la base de datos con todos los datos proporcionados y el usuario confirma que la operación fue exitosa.. | |  | |  |  |  | | --- | --- | | Nombre | R2- LISTADO DE TUTORIALES. | | Resumen | El sistema debería permitir al usuario ver una lista de todos los cursos almacenados en la base de datos. | | Entradas | N/A | |  |  | | Resultados | * Se muestra una lista que incluye cada sección del curso, URL, prioridad de lectura y estado.• La lista se puede ver en la interfaz de usuario y generarse como un informe.. | |  | |  |  |  | | --- | --- | | Nombre | R2- ELIMINAR TUTORIALES | | Resumen | El sistema debe permitir eliminar tutoriales. | | Entradas | N/A | |  |  | | Resultados | * Se debe de eliminar el tutorial previamente ingresado. |  |  | | --- | |  |   **Modelo conceptual**    **DIAGRAMA RELACIONAL**    **Requisitos Técnicos del Sistema Gestor de Tutoriales.**  Para la construcción del sistema gestor de tutoriales, se especifican los siguientes requisitos técnicos, los cuales se han diseñado para asegurar una implementación efectiva y una estructura sostenible del sistema:  1. Uso de un framework de desarrollo web   * Requisito Técnico: Utilizar un framework de desarrollo web moderno como Django, Flask, o Spring Boot para la construcción de la aplicación. * Justificación: Facilitar la gestión de dependencias, rutas, y plantillas, así como también proveer una estructura robusta para la aplicación.   2. Implementación de una base de datos relacional   * Requisito Técnico: Utilizar una base de datos relacional como PostgreSQL, MySQL, o SQLite para almacenar la información de los tutoriales. * Justificación: Permitir una gestión eficiente y segura de los datos mediante el uso de tablas, que faciliten operaciones como el ingreso y listado de tutoriales.   3. Aplicar principios de programación orientada a objetos   * Requisito Técnico: Modelar las entidades del sistema (tutoriales, categorías) utilizando principios de programación orientada a objetos. * Justificación: Facilitar el mantenimiento y la expansión futura del código, además de mejorar la organización del mismo.   4. Crear una interfaz de usuario web amigable y responsiva   * Requisito Técnico: Desarrollar una interfaz de usuario web utilizando HTML, CSS, y frameworks como Bootstrap o Tailwind CSS. * Justificación: Proveer una experiencia de usuario óptima en dispositivos de diferentes tamaños y resoluciones.   5. Uso de ORM (Object-Relational Mapping)   * Requisito Técnico: Implementar un ORM como Hibernate (en el caso de Java con Spring Boot) o Django ORM para gestionar la interacción con la base de datos. * Justificación: Abstraer y facilitar las operaciones de la base de datos, mejorando la seguridad y reduciendo la cantidad de código SQL necesario.   6. Manejo de excepciones adecuado   * Requisito Técnico: Implementar un manejo de excepciones robusto para controlar y responder adecuadamente a situaciones de error. * Justificación: Asegurar que la aplicación maneje fallos de manera elegante y proporcione retroalimentación útil a los usuarios.   **Entregables Esperados:**  1. Debe crear un repositorio en GIT para el proyecto  2. Documentación que explique la estructura del proyecto, boceto de la interfaz de usuario  3. Capturas de pantalla que muestren la aplicación en funcionamiento  **Criterios de Evaluación:**  Los proyectos serán evaluados en función de la correcta implementación de los requisitos funcionales y técnicos, la aplicación adecuada de la programación orientada a objetos, así como la usabilidad y apariencia de la interfaz web. Los estudiantes también deben ser capaces de explicar y justificar las decisiones de diseño y tecnología que tomaron durante el desarrollo del proyecto. Con ello:   1. Todos los requisitos funcionales desarrollados 2. Utilización de la programación orientada a objetos 3. Funcionalidad, el programa debe funcionar en su totalidad sin ningún error 4. Usabilidad, el programa debe ser fácil de utilizar, no debe dejar dudas al usuario 5. El archivo en un repositorio GIT de forma adecuada 6. Código documentado en su totalidad 7. Puntualidad en la entrega 8. Esta documentación completa con el prototipo de la interfaz   **Prototipo de interfaz** | | |
|  | | |