|  |  |  |
| --- | --- | --- |
| **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.jpgparticipante.jpg | **Materiales**   * Internet * Eclipse o Netbeans * Apache Tomcat * Maven |
| **Duración de la actividad**  **4:00** |
| **Proyecto:**  Desarrollo de una aplicación web para una gestor de tutoriales.  **Descripción:**  Se requiere desarrollar un gestor de tutoriales que permita almacenar la siguiente información para cada tutorial:   1. **Categoría, que puede ser: lógica de programación, Flutter, Node.js.** 2. **URL de acceso al tutorial.** 3. **Prioridad de lectura, valorada en una escala del 1 al 10.** 4. **Estado del tutorial, que puede ser: leído, pendiente de revisión.**   **Listado de Requerimientos:**   |  |  | | --- | --- | | Nombre | R1. Ingrese un tutorial. | | Resumen | Se necesita permitir a los usuarios ingresar un nuevo tutorial en la base de datos del gestor. Al ingresar un tutorial, el usuario proporcionará la siguiente información: categoría del tutorial (que puede ser lógica de programación, Flutter o Node.js), la URL de acceso al tutorial, la prioridad de lectura en una escala del 1 al 10, y el estado del tutorial (ya sea leído o pendiente de revisión). | | Entradas | Categoría del tutorial (lógica de programación, Flutter, Node.js)  URL de acceso al tutorial  Prioridad de lectura (valor en una escala del 1 al 10)  Estado del tutorial (leído o pendiente de revisión) | |  |  | | Resultados | Tutorial registrado | |  | |  |  |  | | --- | --- | | Nombre | R2- Mostrar la información o listar los tutoriales existentes. | | Resumen | Se requiere ver un listado de todos los tutoriales almacenados en la base de datos del gestor. Cada tutorial listado incluirá la información siguiente: categoría del tutorial, URL de acceso, prioridad de lectura y estado del tutorial. Esto proporcionará a los usuarios una visión general de los tutoriales disponibles y su estado actual. | | Entradas | No aplica | |  |  | | Resultados | Se muestra un listado de todos los tutoriales almacenados en la base de datos del gestor | |  | |  |  | | --- | |  |  |  |  | | --- | --- | | Nombre | R3 Actualizar un tutorial. | | Resumen | Los usuarios deben poder actualizar la información de un tutorial existente en la base de datos del gestor | | Entradas | Categoría del tutorial (lógica de programación, Flutter, Node.js)  URL de acceso al tutorial  Prioridad de lectura (valor en una escala del 1 al 10)  Estado del tutorial (leído o pendiente de revisión) | |  |  | | Resultados | El tutorial se actualiza con la nueva información proporcionada por el usuario. | |  | |  |  |  | | --- | --- | | Nombre | R4 Eliminar un tutorial. | | Resumen | Se necesita la capacidad de eliminar un tutorial existente de la base de datos del gestor. Al eliminar un tutorial, se eliminará permanentemente de la base de datos y ya no estará disponible en el sistema. | | Entradas | identificador del tutorial a eliminar. | |  |  | | Resultados | El tutorial se elimina de la base de datos del gestor. | |  | |  |  |  | | --- | --- | | Nombre | R5 Buscar un tutorial por ID. | | Resumen | Los usuarios deben poder buscar tutoriales por id en la base de datos del gestor. Al buscar un tutorial por id, el sistema devolverá un modal con la información del tutoriales que pertenezcan al id especificado por el usuario | | Entradas | identificador del tutorial a buscar. | |  |  | | Resultados | Se muestra un modal con la información del tutorial | |  | |   **Modelo conceptual**    **Requisitos Técnicos:**  1. Utilizar el framework Maven para gestionar las dependencias del proyecto.  2. Implementar la una base de datos MySQL para realizar el guardado de información.  3. Aplicar conceptos de programación orientada a objetos para modelar los datos y la funcionalidad del gestor de tutoriales.  4. Crear una interfaz web amigable y responsiva basada en boostrap donde los usuarios puedan ver la exposición de perros  5. Utilizar un ArrayList para almacenar los objetos de manera temporal  **Diagrama ED**    **SCRIPT DE LA BASE DE DATOS** | | |