**Descripción del caso de estudio**

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.**

|  |  |  |  |
| --- | --- | --- | --- |
| ID: | RF1 | Nombre: | 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) |
| Resultado: | | | Tutorial registrado |

1. **Estado del tutorial, que puede ser: leído, pendiente de revisión.**

|  |  |  |  |
| --- | --- | --- | --- |
| ID: | RF2 | Nombre: | liste la información de los tutoriales en la base de datos. |
| Resumen: | | | Este requerimiento permite a los usuarios 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. |
| Resultado: | | | Se muestra un listado de todos los tutoriales almacenados en la base de datos del gestor |

**Diagrama de clases**

* **Tutorial**
* **Gestor tutorial**

**Requerimientos técnicos**

**Diagrama ED**