

PROYECTO

Desarrollar un aplicativo web asociado a la lógica de negocio encontrada en un contexto universitario para estudiantes, profesores y asignaturas, dónde se deberá llevar la gestión de:

- Estudiantes: consulta total e individual, registro, actualización y eliminación.
- Asignaturas: consulta total e individual, registro, actualización y eliminación.
- Profesores: consulta total e individual, registro, actualización y eliminación.
- Profesores que imparten asignaturas: donde estará registrado el id del profesor, el id de la asignatura, el grupo y el horario. Se debe poder consultar por profesor las asignaturas que imparte, así como los profesores que imparten una asignatura. También se debe poder actualizar el número de grupo así como el horario; debe permitirse eliminar.
- Estudiantes que inscriben asignaturas impartidas por profesores: dónde estará registrado el id del estudiante, el id de la asignatura, del profesor y el grupo, así como las notas n1, n2 y n3, que corresponden a los tres cortes académicos del semestre. Se debe poder consultar los estudiantes y sus notas por asignatura y grupo; listar las asignaturas que ve un estudiante en particular y sus notas; editar las notas de un estudiante para una asignatura en puntual. No debe permitirse la eliminación en este caso.

El proyecto debe ser una aplicación web, que deberá tener las vistas que satisfagan el enunciado (se solicita la creación de al menos 2 pantallas, bien sea para guardar información, mostrarla, etc. En el caso más elemental, deberá usar HTML+CSS+JS). En cuanto al back-end, se deben crear únicamente API REST que gestionarán la persistencia de datos a través de los métodos HTTP (GET, POST, PUT, DELETE) que deberán usarse para el manejo de peticiones y respuestas y garantizar la respectiva gestión sobre la base de datos.

En particular el proyecto deberá satisfacer las siguientes condiciones:

- Persistencia con base de datos (motor de preferencia del grupo de estudiantes). Puede ser relacional (MySQL, PostgreSQL, Oracle, etc.) o no relacional (MongoDB).
- Manejo de API REST para conectar con la base de datos (API-CRUD) y poder realizar las operaciones de:
 - Registro de estudiantes, profesores, asignaturas.
 - Actualización de estudiantes, profesores, asignaturas, asignaturas impartidas por profesor (grupo u horario), notas de un estudiante para una asignatura.
 - Consulta de estudiantes, profesores, asignaturas, asignaturas que son impartidas por un profesor y viceversa (profesores que imparten cierta asignatura), lista de estudiantes por asignatura y grupo con sus notas, asignaturas que cursa un estudiante en particular y sus notas.
 - Eliminación de estudiantes, asignaturas, profesores y asignaturas impartidas por profesores. Tenga en cuenta que la eliminación literal de un profesor, estudiante o asignatura podría acarrear errores inesperados en la base de datos (por integridad referencial si la base de datos es relacional).
- Debe usar buenas prácticas REST para la definición de endpoints y/o rutas.

- Use JSON como mecanismo para la transferencia de información.
- Como solo se solicitan algunas vistas para el front-end, el resto de los servicios deben probarse y demostrarse su funcionamiento desde un cliente tipo Postman.
- Claramente el uso de códigos de estado debe ser obligatorio y coherente según la respuesta.
- La tecnología para el back-end para la creación de las API, así como el framework, son de libre elección por parte del conjunto de estudiantes.
- Se deben encontrar las vistas de una página web requeridas para que cumplan con los requerimientos, pudiéndose gestionar al menos profesores, estudiantes y asignaturas.