# WEB 2 - TUDAI

# RECUPERATORIO 26/06/2021

La escuela de manejo **“Indy Drive”** desea desarrollar una plataforma web para gestionar a los alumnos y los autos que se utilizarán para la enseñanza de cada uno de ellos. Para esto mismo, el alumno al inscribirse se le asignará un vehículo con el cual realizará todas las pruebas de manejo durante las clases.

Para ello, desde la escuela de manejo nos facilitan algunas de las tablas de su base de datos interna:

**ALUMNO**

id: *int*;

nombre: *string*;

edad: *int*;

telefono: *string*;

aprobado: *boolean*;

recursante: *boolean*;

id\_vehiculo\_fk: *int*;

**VEHICULO**

id: *int*;

nombre: *string*;

profesor\_asignado: *boolean*;

funcionando: *boolean*;

En esta plataforma los **usuarios** podrán:

* Ver y buscar a los alumnos filtrando por **vehículo** o por alguno de sus campos **(nombre, edad, teléfono, aprobado, recursante).**
* Administrar los vehículos que tiene disponible la escuela, realizando consultas sobre los campos nombre, profesor asignado y funcionando para determinar el estado de cada uno. Esto mismo permitirá sacar estadísticas en un futuro.
* Administrar los alumnos de la escuela, listar utilizando diferentes filtros
* Visualizar pantalla de estadísticas de aprobados
* **Loguearse** para agregar un alumno a la plataforma.

Además los **usuarios administradores** podrán:

* Administrar vehículos (ABM)
* Modificar cualquier agregado de alumnos realizado por un usuario.

**1.** Piense y diseñe una **arquitectura MVC** para poder implementar esta plataforma. Dibuje el diagrama de componentes MVC desarrollado.

* **No es necesario implementar nada**, simplemente indique que controladores, modelos y vistas serían necesarios, y una explicación breve de la función de cada uno.
* Suponga las tablas de la base de datos que faltarían para la solución completa. No es necesario indicar su estructura (campos).

**2. Implemente** dentro de su arquitectura MVC dos (2) de los siguientes requerimientos. Ver tabla de asignación de temas.

**SECCIÓN HOME**

1. Listar los alumnos que sean recursantes y se encuentren aprobados.
2. Listar los que estén en funcionamiento y tengan profesores asignados.
3. Listar los alumnos diferenciados por los aprobados de los no aprobados. Ejemplo:
   1. *Aprobados*
      1. Alumno Diez
      2. Alumno Dicosimo
   2. *Desaprobados*
      1. Alumno Rall
      2. Alumno Tai
4. Listar los alumnos diferenciados por los recursantes de los no recursantes. Ejemplo:
   1. *Recursantes*
      1. Alumno Diez
      2. Alumno Tai
   2. *No recursantes*
      1. Alumno Rall
      2. Alumno Dicosimo

**SECCIÓN ADMIN** *\*\*no es necesario implementar el acceso de usuario\*\**

1. Agregar un alumno completoal sistema. Se debe verificar que el nombre del alumno no debe estar repetido.
2. Agregar un vehículo al sistema. Se debe verificar que no exista uno con el mismo nombre y que los campos agregados no estén vacíos.

NOTAS:

* Suponga que ya están logueados los usuarios administradores. No realice ningún control.
* Verifique los datos obligatorios en las altas.
* No es importante para nada la interfaz gráfica del sistema.

**3. Servicios Web**

¿Qué servicios web pensaría que pueden ser útiles en este sistema? De 2 ejemplos, defina los endpoints de cada uno y explique brevemente cual es su función y porqué pueden ser útiles, detalle todo lo que crea conveniente.

Para los endpoints se debe implementar los siguientes cambios en la base de datos:

Crear una tabla llamada **PROFESORES** con los siguientes campos:

**PROFESORES**

id\_profesor: *int*;

nombre: *string*;

edad: *int*;

2. Crear una columna en la tabla **ALUMNOS** yuna en la tabla **VEHICULOS:**

**ALUMNOS VEHICULOS**

**id\_profesor\_fk: int;** **id\_profesor\_fk: int;**

(Establecemos una relación sobre id\_profesor\_fk de la tabla **ALUMNOS** y **VEHICULOS** con id\_profesor de la tabla **PROFESORES**)

**1-“api/profesor/alumnos”:** **FUNCION:** Listar los alumnos por su profesor asignado.

**UTILIDAD:** Puede ser útil para buscar un profesor y conocer cuales alumnos tienen a ese profesor asignado.

**2-“api/profesores/vehículos”:** **FUNCION:** Listar los vehículos según el profesor que lo tenga asignado.

**UTILIDAD**:Seria útil para poder encontrar quien será el profesor encargado de dictar las clases cuando el usuario sepa que vehículo le corresponde para la clase.

**4. Nueva feature**

Describa brevemente qué cambios haría en el sistema si el cliente desea agregar un sistema de facturación para cobrar las clases que se dictan.

1. Crearia una tabla en la BBDD que se asi:

**Factuas**

id\_facturas: *int*;

pagado: *boolean*;

fecha: d*ate*;

1. Agregar una columna en la tabla usuario que se llame “id\_facturas\_fk” y establecer la relacion con id\_facturas de la tabla Facturas
2. Crear un formulario que registre si el usuario pagó las clases con esos datos se actualice la tabla para arrancar con las clases

**Ejemplo db\_indydrive.sql**

**(**[**https://drive.google.com/file/d/1yZOFM3ZVV1L\_hnl\_y44Uurb9wdQs19\_Q/view?usp=sharing**](https://drive.google.com/file/d/1yZOFM3ZVV1L_hnl_y44Uurb9wdQs19_Q/view?usp=sharing)**)**

TABLA **ALUMNO**



TABLA **VEHICULO**

