

Examen Final DDS

Condiciones generales

- Tiempo de resolución: **90 minutos**
 - Punto de partida: te brindamos los proyectos de backend y frontend iniciados para acortar el tiempo de resolución. Si preferís comenzar los proyectos desde cero y pensás que te va a alcanzar el tiempo podés hacerlo. Siempre utilizando las tecnologías que constan en el punto 2.
- Tecnología a utilizar**

1. Modelo de Datos

La base de datos ubicada en `/backend/datos/autos.db` ya contiene datos y obedece al siguiente modelo

1.1 autos

Campo	Tipo
id	Int Clave Primaria Autoincremental
anio	int
fabricante	varchar
modelo	varchar
categoria	varchar

1.2. categorias

Campo	Tipo
id	Int Clave Primaria Autoincremental
descripcion	int

Además hay un backup de la base de datos por si querés restaurar rápidamente los datos a su estado inicial. Ese backup está en: `/backend/datos/autos.db.bak`

2. Tecnología a utilizar

El ejercicio debe ser resuelto utilizando:

- Node.js
- Express
- DB: Sqlite 3
- ORM: Sequelize

- Frontend: React

3. Estructura/Arquitectura propuesta

El proyecto inicial tiene una estructura similar a la de varios ejemplos que hemos realizado durante el curso. No obstante, no es necesario que la sigas estrictamente.

Si, es un requisito que se utilice Express router para definir las rutas.

4. Backend: Requerimientos

Debés implementar una API con los siguientes endpoints:

4.1. GET /api/autos/ Recupera todos los autos

Este mismo endpoint debe permitir filtrar por fabricante y modelo

Por Ejemplo:

GET /api/autos?fabricante=toyo recupera todos los autos de la categoria Pickup cuyo fabricante contiene “toyo”

GET /api/autos?fabricante=toyo&modelo=taco recupera todos los autos de la categoria Pickup cuyo fabricante contiene “toyo” y el modelo contiene “taco”

4.2. POST /api/autos Crea un auto

4.3. GET /api/categorias/ Recupera todas las categorías

5. Frontend: Requerimientos

Debés construir una aplicación web con las siguientes características

5.1. Navegación: Debe constar de una barra o menú de navegación. El item “Listado” de la barra de navegación debe navegar hacia un componente que tenés que crear: Autos

5.2. Listado: El componente Autos debe mostrar inicialmente el listado de los autos. Debe mostrar solamente los campos siguientes:

- anio
- fabricante
- modelo
- categoria

5.3. Filtros: Se debe implementar un filtro en el componente Autos que permita filtrar por los siguientes campos:

- **fabricante:** mediante un campo de texto libre.
- **modelo:** mediante un campo de texto libre que permita ingresar solo números.

4. Alta: Se debe agregar un botón en este componente que permita navegar o exhibir otro componente: DetalleAuto. Este componente debe permitir agregar un nuevo Auto ingresando datos para los campos siguientes:

- año
- fabricante
- modelo
- categoría

Todos los campos son requeridos

Una vez ingresado exitosamente el registro del Auto, debe navegar hacia el listado, donde el registro recién ingresado estará visible (si concuerda con los filtros activos)

6. Criterios de Evaluación

Criterio	Puntos
Definición del modelo	10
Endpoints	10
Navegación	5
Listado	15
Filtros	20
Alta	20
Validación	10
Estructura de código y arquitectura	10

7. Escala de notas

Nota	Porcentaje	Calificación
1		No aprobado
2		No aprobado
3		No aprobado
4	55% a 57%	Aprobado
5	58% a 59%	Aprobado
6	60% a 68%	Aprobado
7	69% a 77%	Aprobado
8	78% a 86%	Aprobado
9	87% a 95%	Aprobado

Nota	Porcentaje	Calificación
10	96% a 100%	Aprobado

8. Entrega

Para entregar debés subir a la UV un archivo comprimido .zip conteniendo las carpetas

Antes de comprimir la carpeta, tenés que borrar las carpetas node_modules de ambos proyectos

El archivo tiene que nombrarse con el siguiente patrón: <apellido>_<nombre>_<legajo>.zip

Por ejemplo, si te llamás Juan Garcia Saravia y tu legajo es 45044, el archivo que debés subir será garcia_saravia_juan_45044.zip