



# Prova pràctica UF1844

Desenvolupament d'aplicacions web en entorn servidor

## Exercici 1 - 1 punt

1. Contesta les quatre preguntes al mateix fitxer app.js.
2. Fes-ho a la línia on posa "Resposta"
3. Cada pregunta correcte sumarà **0,25 punts**

```
/**
 * Contesta las preguntas en este mismo archivo
 */

/**
 * 1. Cuantos parámetros de entrada tiene el constructor de esta clase ?
 * Respuesta:
 *
 * 2. Cual es el valor de la propiedad del objeto "this.deposito" cuando instanciamos esta clase?
 * Respuesta:
 *
 * 3. Escribe un ejemplo de como crearías un objeto de la clase Coche, pasándole al constructor todos los
    parámetros
 * Respuesta:
 *
 * 4. Supón que la variable "coche" es una instancia de la clase Coche. Que mostrará por el terminal: console.
    log(coche.llenarDeposito())? Por qué?
 * Respuesta:
 *
 *
 */
```

## Exercici 2 - 1 punt

1. Defineix una funció "**acelerar**" a la classe cotxe, que gastí el doble de combustible que la funció "usar" **0,5 punts**

```
31 const coche = new Coche("Opel", "1234 AL", true);
32 console.log(coche.deposito);
33 coche.acelerar();
34 console.log(coche.deposito);
35
```

OUTPUT    DEBUG CONSOLE    PROBLEMS    TERMINAL

```
F:\Google Drive\Netmind\Temari Desenvolupat\Avaluacions\MF0492_3
100
98
```

2. Defineix una funció "estaVacio", que retorna *true* en cas que el dipòsit aquest buit, o *false* en cas contrari. **0,5 punts**

```
35 const coche = new Coche("Opel", "1234 AL", true);
36 console.log(coche.deposito);
37 coche.acelerar();
38 console.log(coche.deposito);
39 console.log(coche.estaVacio());
40
```

OUTPUT    DEBUG CONSOLE    PROBLEMS    TERMINAL

```
F:\Google Drive\Netmind\Temari Desenvolupat\Avaluacions\MF0492_3
100
98
false
```

### Exercici 3 - 1 punt

1. Implementa una funció **existeMatricula**, que rep com a paràmetre una matrícula de cotxe, i busca a la base de dades "baseDatosCoche"
2. Només has d'implementar el cos de la funció, perquè retorni *true* o *false*. No modifiquis altres parts del codi (llevat per a fer proves, si vols). Tampoc has de modificar els paràmetres de la funció.

Resultat per consola esperat:

```
true true  
false false  
false false
```

## Exercici 4 - 2,5 punts

Implementa un formulari HTML per introduir les dades d'un cotxe. Quan faci clic al botó d'Enviar, ha de fer una sèrie de validacions, que es detallen a continuació:

1. El formulari té un camp de nom **"Modelo"**. El camp és de tipus text. Com a màxim pot tenir 30 caràcters. És un camp obligatori. **0,5 punts**
2. El formulari té un camp de nom **"Matrícula"**. El camp és de tipus text. El camp ha de validar contra una expressió regular que sigui 4 números exactament, seguit d'un espai, i exactament dues lletres majúscules. Exemple: **"2211 BL"** és una matrícula vàlida. **"200AA"** no ho és. És un camp obligatori. **0,5 punts**
3. El formulari té un camp de nom **"GPS"**. És del tipus "checkbox". Indica si el cotxe té incorporat GPS o no. És obligatori. **0,5 punts**
4. El formulari ha d'enviar les dades que recull a un *endpoint* que té per URL: <https://postman-echo.com/post> **0,5 punts**
5. El formulari ha d'enviar les dades fent servir el mètode POST **0,5 punts**

Important: És un HTML simple. Aquest exercici no té res a veure amb NodeJS.

Modelo

Opel Astra

Matricula

1234 BB

GPS

☒

Submit

## Exercici 5 - 3,5 punts

Insta-la totes les dependències del projecte amb “npm install”. Recorda que pots iniciar l'aplicació amb “npm start”.

A continuació, accedeix a “<http://localhost:3000>”. L'aplicació en NodeJS pretén permetre a l'usuari afegir un número de DNI. En fer clic a “Enviar”; el script ha de respondre amb un text simple indicant si és un DNI vàlid o no. En aquest exercici, **només es pot modificar el fitxer “app.js”**

1. Aquesta aplicació té un error de sintaxi. Soluciona'l **0,75 punts**
2. Aquesta aplicació té un error d'execució. Soluciona'l. **0,75 punts**
3. Aquesta aplicació té tres errors lògic o de disseny, que estan evitant que es dugui a terme la validació del document d'identitat de forma correcta. Concretament, hi ha TRES línies errònies. Cada línia de codi, conté un error.
  - a. Troba el primer error lògic i soluciona'l **0,5 punts**
  - b. Troba el segon error lògic i soluciona'l **0,5 punts**
  - c. Troba el tercer error lògic i soluciona'l **0,5 punts**



**RECORDA: Només pots modificar l'arxiu app.js**

4. Fent servir el mòdul de tercers **spain-id**; modifica el script perquè la teva aplicació també validi document d'identitat d'estrangers (NIE). Per exemple, ‘**X8095495R**’ és un NIE vàlid. Important: l'aplicació ha de seguir validant els DNI! **0,5 punts**

Exemple de funcionament: <https://oscarm.tinytake.com/tt/NDM4MDMxN18xMzgyNzkyNg>

**Nota: Aquest projecte fa servir el mòdul: <https://www.npmjs.com/package/spain-id>**

## Exercici 6 - 1 punt

Insta-la totes les dependències del projecte amb "npm install". A continuació, accedeix a "<http://localhost:3000>".

1. Quan fem clic a Enviar, farem un POST al endpoint '/'. Es demanen les següents funcionalitats
  - a. Modifica formulari.html per tal de verificar que hem introduït un nombre enter, entre 1 i 100. **0,25 punts**
  - b. Els estils del fitxer ubicat a public/css/main.css no s'estan aplicant correctament al formulari. Soluciona-ho **0,25 punts**
  - c. La vista ha de renderitzar el número introduït per l'usuari **0,25 punts**
  - d. La vista ha d'informar si el número introduït per l'usuari és parell o imparell; tot seguint el format de l'exemple **0,25 punts**

**Exemple de funcionament:**

<https://oscarm.tinytake.com/tt/NDM4MDI0MI8xMzgyNzc4OA>