

## Práctica 1

### Requisitos de entrega

Se debe subir un único fichero .zip que contenga el proyecto Node. Comprueba que:

- El fichero principal que arranca el servidor se llama `app.js`. El proyecto ha de ser arrancable mediante la orden `npm start` o, en su defecto, `node app.js`.
- Se incluye un fichero `.sql` en el directorio raíz del proyecto con las sentencias que crean la BD, sin incluir la sentencia `CREATE DATABASE`. Este fichero puede ser generado automáticamente por *phpMyAdmin*.
- Existe un fichero `config.js` en el directorio raíz del proyecto con los datos de configuración de vuestra práctica. Puedes encontrar una plantilla de este fichero en el CV. Por ejemplo:

```
module.exports = {  
  // Nombre de la base de datos de vuestra  
  // práctica  
  dbName: "BDPractical",  
  
  // Nombre de usuario que utilizáis para acceder a la BD.  
  dbUser: "root",  
  
  // Contraseña que utilizáis para acceder a la BD  
  dbPassword: "",  
  
  // Host donde está la BD.  
  dbHost: "localhost",  
  
  // Puerto en el que escucha vuestra aplicación  
  port: 3000  
};
```

- Se hace uso de este fichero `config.js` en vuestra aplicación, importándolo donde sea necesario,

```
let config = require("./config");
```

y utilizando sus propiedades cuando se inicien las conexiones a la base de datos y se arranque el servidor web. Por ejemplo, en lugar de:

```
app.listen(3000, ...);
```

se debe poner

```
app.listen(config.port, ...);
```

- No hay ninguna referencia al nombre de la base de datos, o al nombre de usuario y contraseña de MySQL, excepto en el fichero `config.js`. Tened en cuenta que, cuando pruebe la práctica en mi sistema, voy a utilizar, para la base de datos, un nombre distinto al que habéis utilizado vosotros.
- Los `require(...)` se realizan prestando atención a mayúsculas y minúsculas (especialmente para aquellos que utilizáis Windows). Los nombres de módulos pasados como parámetros a `require` han de coincidir exactamente con el nombre del fichero que se quiere importar, teniendo en cuenta las letras mayúsculas y minúsculas. Por ejemplo, si tenéis un módulo llamado `daoUser.js` y hacéis lo siguiente:

```
let daouser = require("./daouser")
```

En Windows funcionará correctamente, pero en Linux o Mac no. El sistema de ficheros de Windows no hace distinción entre mayúsculas y minúsculas.