PROYECTO 1: HOLA USUARIO

TIEMPO ESTIMADO: 60 MINUTOS

DESCRIPCIÓN

- Realizar un programa que escriba en un fichero el texto siguiente:

```
Hola usuario, tienes 25 años
```

  - Usuario se obtendrá de la variable de entorno

```
$USERNAME
```

  - La edad se obtendrá de una variable

OBJETIVOS

- Entender el funcionamiento de los módulos
  - Utilizar módulos del núcleo (os, fs)
  - Crear un módulo sencillo
- Usar sintaxis de ES6 (*destructuring* y *template string*)

COMENZAR PROYECTO

```
mkdir holaUsuario
cd holaUsuario
touch app.js
code .
```

MÓDULOS DE SISTEMA

- Están "built-in"
  - Se cargan mediante un *require*
  - No es necesario instalarlos
- Podemos consultarlos en la web de node:
  - https://nodejs.org/es/docs/
    - Elegimos versión de la API
  - Simplemente ejecutando (zsh + plugin node):

```
node-docs
```

AÑADIR TEXTO A UN FICHERO

- Función asíncrona:

```
console.log('Iniciando app');
const fs = require('fs');
// fs es un objeto con muchas funciones, ver api
fs.appendFile('saludo.txt', 'Hola Usuario');
```

- Salida con warning:

```
(node:9493) [DEP0013] DeprecationWarning: Calling an asynchron
```

- ¡Recoger error o éxito con función de callback para evitar warning!

```
console.log('Iniciando app');
const fs = require('fs');
// fs es un objeto con muchas funciones, ver api
fs.appendFile('saludo.txt', 'Hola Usuario', (err)=>{
  err ? console.log('Ha habido un error') : console.log('Todo
});
```

- Podríamos utilizar también una función síncrona:

```
console.log('Iniciando app');
const fs = require('fs');
try {
  fs.appendFileSync('saludo.txt', 'Hola usuario');
  console.log('Todo ok!');
} catch (err) {
  console.log('Ha habido un error');
}
```

OBTENER EL NOMBRE DEL USUARIO

- Utilizaremos el módulo OS para averiguar el nombre del usuario

```
const os = require('os');
const user = os.userInfo();
console.log(user); // para ver que datos tiene userInfo()
```

IMPLEMENTACIÓN ES5

```
const fs = require('fs');
const os = require('os');
const user = os.userInfo();

console.log('Iniciando app');
const saludo = 'Hola ' + user.username;
fs.appendFile('saludo.txt', saludo, function (err) {
  err ? console.log('Ha habido un error') : console.log('Todo
});
```

ES6: DESTRUCTURING

- Mapeamos una o varias partes de un objeto a una o varias variables:

```
let { x, y, ...z } = { x: 1, y: 2, a: 3, b: 4 };
console.log(x); // 1
console.log(y); // 2
console.log(z); // { a: 3, b: 4 }
```

ES6: TEMPLATE STRINGS

```
var a = 5;
var b = 10;
console.log(`Fifteen is ${a + b} and not ${2 * a + b}.`);
// "Fifteen is 15 and not 20."
```

ES6: OBJECT LITERAL PROPERTY VALUE SHORTHAND

- Antes (ES5):

```
function createMonster(name, power) {
  return { type: 'Monster', name: name, power: power };
}
```

- Ahora (ES6):

```
function createMonster(name, power) {
  return { type: 'Monster', name, power };
}
```

IMPLEMENTACIÓN ES6

```
const fs = require('fs');
const os = require('os');
const { username } = os.userInfo();

console.log('Iniciando app');
const saludo = `Hola ${username}`;
fs.appendFile('saludo.txt', saludo, (err) => {
  err ? console.log('Ha habido un error') : console.log('Todo
});
```

REQUIRE

- *require* es un módulo que está en el objeto global
- Este código no es necesario:

```
require('require');
```

- Funciona de forma síncrona
  - Por eso se ponen al comienzo
  - Podríamos colocarlos más tarde y hacer *lazy loading*

USO DE MÓDULOS

- Vamos a crear un módulo que sea el encargado de proporcionarnos el usuario
- Creamos el fichero user.js con el siguiente texto:

```
console.log('Cargando módulo para el usuario');
```

- ¿Cómo lo cargamos dentro de nuestro app.js?

```
const user = require ('./user.js')
```

- Comprobamos la ejecución que muestra el texto del módulo requerido por consola.

MÓDULOS EN JAVASCRIPT

- En JavaScript no hay módulos ni namespaces.
- Todo va al objeto window y puede haber solapamiento de variables.
- Se implementan módulos de forma nativa con ES6

- Fichero index.js

```
var nombre = "juan";
```

- Fichero index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Document</title>
</head>

<body>
  <script>
    var nombre = "pepe";
    console.log(nombre);
  </script>
  <script src="./index.js"></script>
  <script>
    console.log(nombre);
  </script>
```

USO DE VARIABLES Y FUNCIONES DE OTRO MÓDULO EN NODE

- El objeto module tiene muchas propiedades, nos interesará **module.exports**

```
console.log(module)
```

- module.exports puede ser una función, un objeto, un string...
- Será ahí donde tendremos que crear un objeto con el nombre del usuario y la edad.

SOLUCIÓN PROYECTO

- Fichero app.js:

```
const fs = require('fs');
const os = require('os');
console.log('Iniciando app');

const { username, edad } = require('./user')

const saludo = `Hola ${username}, tienes ${edad} años`;
fs.appendFile('saludo.txt', saludo, (err) => {
  err ? console.log('Ha habido un error') : console.log('Todo
});
```

- Módulo user.js:

```
const { username } = require('os').userInfo()
const edad = 25;
module.exports = { username, edad }
```

EJERCICIO CARGA MÓDULOS

- ¿Qué mostraría el siguiente programa?
- ¿Y si comentamos la primera línea de app.js?

app.js:

```
require('./module1');
require('./module2');
console.log('Iniciando app');
```

module1.js:

```
console.log('Ejecutando módulo 1');
```

module2.js:

```
require('./module1')
console.log('Ejecutando módulo 2');
```

SALIDA EJERCICIO

- El texto *Ejecutando módulo 1* se muestra solo una vez
  - Ya está cargado previamente, se usa la caché y no se ejecuta
  - El texto *Inicializando app* sale después del console.log de los require (los require son síncronos).

EJERCICIO LEER FICHEROS

- Crea dos ficheros numero1.txt y numero2.txt y escribe un número en cada uno
- Crea un programa que:
  - Lea el contenido de los dos fichero y lo almacene en variables
  - Muestre por consola la suma de las variables

SOLUCIÓN LECTURA FICHEROS

```
const fs = require('fs')
const numero1 = fs.readFileSync('./numero1', 'utf-8')
const numero2 = fs.readFileSync('./numero2', 'utf-8')
console.log(`El resultado de la suma es  ${parseInt(numero1)+p
```

¿Y AHORA QUÉ?

- Proyecto apuntes en markdown