

## Módulo – operating system (os)



Fornece informações sobre o sistema operacional onde o Node.js está sendo executado.

Permitindo acessar informações como a arquitetura da CPU, o tempo de atividade do sistema e informações sobre a memória.

```
const os = require("os")

console.log("Nome da máquina:", os.hostname())
console.log("Sistema operacional:", os.platform())
console.log("Arquitetura da CPU (64 ou 32):", os.arch())
console.log("Informações da CPU:", os.cpus())
console.log("Memória RAM livre (bytes):", os.freemem())
console.log("Memória RAM Total (bytes):", os.totalmem())
```

## Módulo – path



É usado para trabalhar com caminhos de arquivos e pastas. Ele fornece métodos para manipular caminhos do sistema operacional.

```
const path = require("path")

const pasta = "aluno/Desktop/aula_2"
const arquivo = "script.js"

const caminhoCompleto = path.join(pasta, arquivo)
const nomeDoArquivo = path.basename(caminhoCompleto)
const extensao = path.extname(caminhoCompleto)
const caminhoAbsoluto = path.resolve(caminhoCompleto)
```

aluno\Desktop\aula\_2\script.js  
script.js  
.js  
C:\Users\...\script.js

## Módulo – process



Fornece informações e controle sobre o processo Node.js em execução. Ele inclui informações como argumentos de linha de comando, variáveis de ambiente e outros métodos relacionados ao processo.

```
console.log("Variáveis de ambiente:", process.env)
```

As variáveis de ambiente contêm informações sobre o ambiente no qual o processo está sendo executado.

Elas são definidas pelo sistema operacional e são frequentemente usadas para acessar informações sensíveis, como [senhas ou chaves de API](#). Essas informações podem ser [definidas como variáveis de ambiente](#) e acessadas no seu código Node.js. Isso ajuda a tornar o [código mais seguro](#) e portátil.

## Módulo – process



Fornece informações e controle sobre o processo Node.js em execução. Ele inclui informações como argumentos de linha de comando, variáveis de ambiente e outros métodos relacionados ao processo.

```
console.log("ID do processo:", process.pid)
```

Equivale ao comando *tasklist* do Windows ou *ps aux* do Linux, nos retorna o id do processo no sistema operacional

```
console.log("Comando executados para iniciar o processo:", process.argv)
```

O primeiro elemento é o caminho para o executável do Node.js, geralmente "node" ou o caminho completo para o executável e o segundo elemento é o caminho para o arquivo Node.js que está sendo executado

## Módulo – criação de variáveis de ambiente (dotenv)



A biblioteca dotenv é uma ferramenta popular para carregar variáveis de ambiente a partir de arquivos de configuração, onde podem ser incluídos senhas de acesso aos banco de dados, chaves de APIs e outras informações sensíveis.

Para iniciar, precisamos do pacote dotenv instalado no projeto via npm:

```
npm i dotenv
```

Depois, podemos criar um arquivo chamado .env para incluir as variáveis que precisamos.



```
gear .env
1  AULA_HOJE="modulos_node"
2  MINHA_SENHA="javascript123"
```

## Módulo – criação de variáveis de ambiente (dotenv)



Após realizar a criação, precisamos ativar a biblioteca no código para reconhecer as variáveis personalizadas.

```
const dotenv = require("dotenv");  
dotenv.config()
```

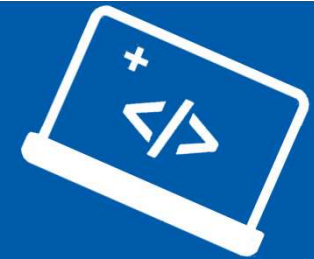
OU

```
require('dotenv').config();
```

Com isso, já conseguimos acessar os valores do arquivo .env pelo código, utilizando o pacote process.

```
const minhaSenha = process.env.MINHA_SENHA  
const aulaHoje = process.env.AULA_HOJE
```

## Atividades (node modules)



Crie um novo projeto node.js e realize:

- A criação de 3 variáveis de ambiente em um arquivo .env: DATA\_NASCIMENTO, EMAIL, NACIONALIDADE.
- A criação de 3 constantes em um arquivo js para receber essas 3 variáveis de ambiente e exiba as constantes no console. Exiba também alguma outra variável criada pelo sistema, como HOMEPATH ou LANG