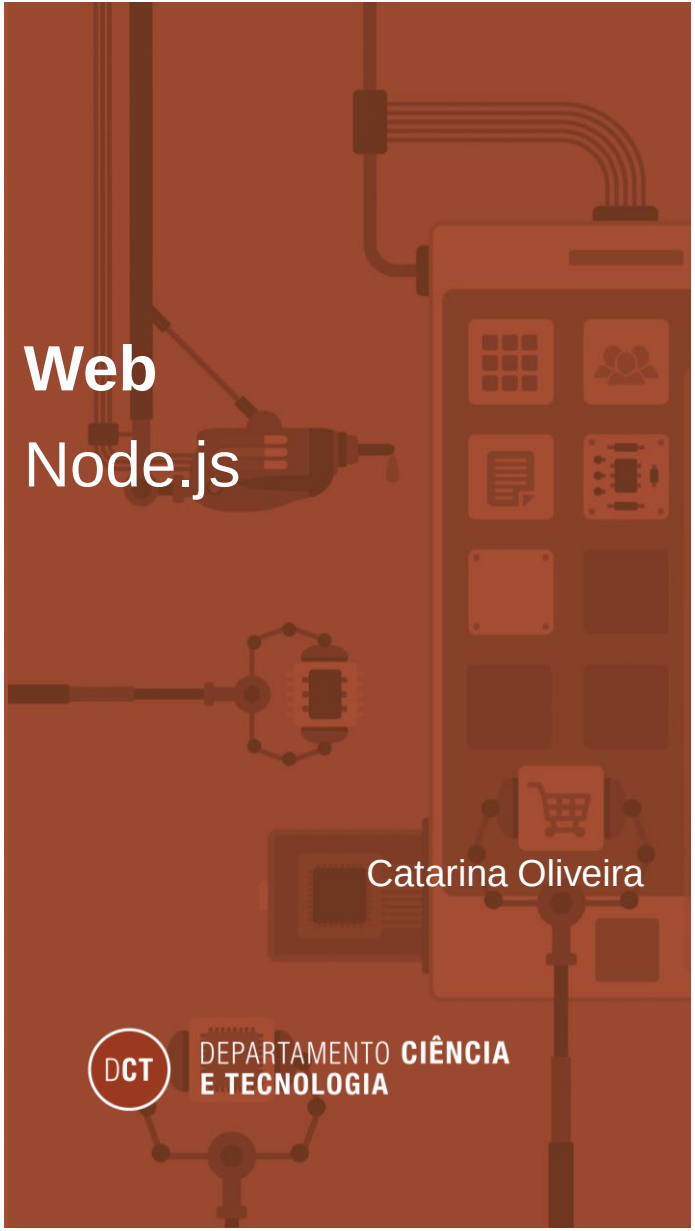

UPT UNIVERSIDADE PORTUCALENSE

CONTENT

1. Node.js
2. Node Package Manager (NPM)
3. NPM: main commands
4. package.json
5. Some remarks

Node.js

- Server side (back end) JavaScript interpreter

- Allows simple and fast development of real time, scalable network applications

- Characteristics:
  - Several network and internet protocols
  - Has libraries that use JavaScript to access the operating system's *input/output* resources
  - S*ingle-thread* and *event-loop*
    - *Single thread:* execution of a single task, until the end without interruptions
    - *Event-loop:* execution of a cycle waiting for events and process them as they arrive
  - Includes non-blocking input/output

- Processes treated sequentially to avoid waiting queues

- *Threads* work asynchronously without blocking the *input/output*

Node Package Manager (NPM)

- Node.js module (package) manager
  - List of available modules: https://www.npmjs.com

- Modules
  - Publicly available reusable components
  - Can be installed from an online repository or the *Command Line Interface* (**CLI**)

- Adding modules to Node.js: instruction `require(nome_modulo)`, that searches for the module:
  - On the native modules (https://nodejs.org/api/)
  - On the installed modules
    - Locally on folder node_modules

- It is possible to create modules by exporting functions: `module.exports`

```
// somar.js
const op = require('./operacoes');
op.somar(1,2);
```

```
// Module operacoes.js
function soma(x,y){
   return x+y;
}
function mult(x,y){
   return x*y;
}
module.exports = {
   somar: soma,
   multip: mult
}
```

NPM: main commands

Command	Description
`npm init`	Shows a questionnaire to help creating a project's package.json descriptor
`npm install nome_modulo`	Installs the module on the project
`npm install -g nome_modulo`	Installs the module globally (in all the folders)
`npm install nome_modulo --save`	Installs the module and adds it to the attribute `dependencies` on `package.json`
`npm install nome_modulo --save-dev`	Installs the module and adds it to the attribute `devdependencies` on `package.json`
`npm list`	Lists all the modules installed on the project
`npm list -g`	Lists all the modules globally installed on the project
`npm remove nome_modulo`	Uninstalls the module from the project
`npm remove -g nome_modulo`	Globally uninstalls the module from the project
`npm remove nome_modulo --save`	Uninstalls the module from the project and removes it from the attribute `dependencies`
`npm update nome_modulo`	Updates the module version
`npm update -g nome_modulo`	Updates the globally installed module version
`npm help`	Shows all the commands
`npm -v`	Shows the NPM installed version
`node -v`	Shows the Node.js installed version

package.json

- Module descriptor (project documentation)
- Created after executing `npm init`
- Modules installed with `npm nome_modulo –save` are added to the file
- Editing allows task automatization

Attribute	Description
name	Application's name
description	Application's description
author	Application's author
version	Application's version
main	Main execution file
keywords	Application's keywords
git	Application's git URL
license	Application's license information
scripts	Creation of predefined commands
dependencies	Application's dependencies (modules needed for execution)
devdependencies	Application's dependencies in development mode

Some remarks

- Node.js makes easier the task of creating servers
- It is not ready for the development of robust web applications
- Solution?
    - Frameworks
        - Example: ***Framework Express*** (next)

UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.