

Ocultar <<

Módulo: 8



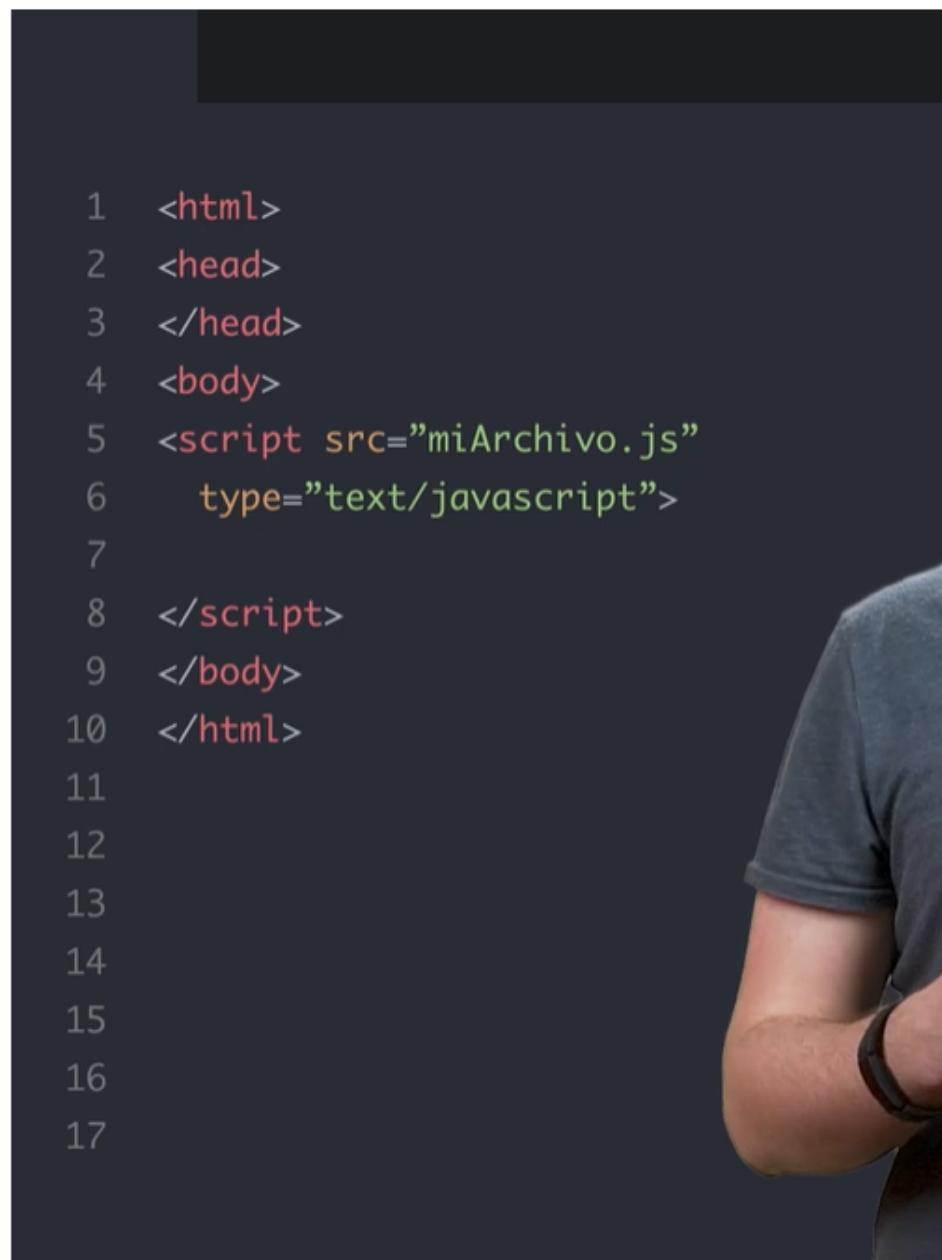
JavaScript para front-end

Clase 37 - Manipulando elementos con JavaScript



- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Vinculación
- Objeto window y document
- DOM Selectores
- Modificando el DOM
- Modificando estilos
- Modificando clases
- Hacia la clase en vivo

```
1  <html>
2  <head>
3  </head>
4  <body>
5  <script src="miArchivo.js"
6    type="text/javascript">
7
8  </script>
9  </body>
10 </html>
11
12
13
14
15
16
17
```



The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- Top Bar:** Shows tabs for `app.js`, `movies.js`, `moviesController.js` (highlighted in blue), `movie.js`, `actor.js`, and `genre.js`.
- Left Sidebar (Explorer):** Displays the project structure under `MOVIESAPI`. It includes `controllers`, `database`, `config`, `migrations`, `models` (with files `actor.js`, `genre.js`, `index.js`, `movie.js`), `seeders`, `node_modules`, `routes` (with files `movies.js`, `.gitignore`, `.sequelizerc`), `app.js`, `package.json`, and `yarn.lock`. There are 6 untracked files indicated by a blue square icon.
- Central Area (Code Editor):** Shows the content of `moviesController.js`. A tooltip message "Presiona Esc para salir de la pantalla completa" is displayed over the code area. The code handles a `list` endpoint for movies, returning a JSON response with total count, data, and status code 200.
- Bottom Navigation Bar:** Includes tabs for `PROBLEMS`, `OUTPUT`, `DEBUG CONSOLE`, and `TERMINAL`. The `TERMINAL` tab is active, showing logs from `nodemon` starting the server on port 3000.
- Bottom Status Bar:** Shows file paths like `fullApi*`, `Javier`, `Live Share`, `tabnine`, and file statistics: Ln 9, Col 34, Tab Size: 3, UTF-8, LF, JavaScript, 1h 8m, ESLint, Prettier.
- Bottom Right Corner:** Contains a "Blockframe" button.

38 (38) WhatsApp > Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "total": 26,
3   "data": [
4     {
5       "id": 1,
6       "title": "Avatar",
7       "rating": "7.9",
8       "awards": 3,
9       "release_date": "2010-10-04T00:00:00.000Z",
10      "length": 120,
11      "created_at": null,
12      "updated_at": null,
13      "genre_id": 5
14    },
15    {
16      "id": 2,
17      "title": "Titanic",
18      "rating": "7.7",
19      "awards": 11,
20      "release_date": "1997-09-04T00:00:00.000Z",
21      "length": 320,
22    }
  ]
```

Find and Replace Console

RESI

Creación de API propia

Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

¿Qué vamos a ver en esta clase?

Home Workspaces Reports Explore

Overview GET http://localhost:30... + ...

http://localhost:3000/movies

GET http://localhost:3000/movies

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY VALUE DESCRIPTION

Key Value Description

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "total": 26,
3   "data": [
4     ...
5   ],
6   "status": 200
7 }
```

Status: 200 OK Time: 291 ms Size: 4.73 KB Save Response

14:25

DigitalHouse >

Find and Replace Console

Escribe aquí para buscar

19°C Nublado 13:48 ESP 6/11/2021

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs < >

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

The screenshot shows a code editor interface with several tabs open at the top: 'JS app.js', 'JS movies.js', 'JS moviesController.js', 'JS movie.js', 'JS actor.js', and 'JS genre.js'. The 'moviesController.js' tab is active. The code in 'moviesController.js' is as follows:

```
const express = require('express');
const router = express.Router();
const controller = require('../controllers/moviesController');

router.get('/', controller.list);
router.get('/:id', controller.show);
module.exports = router;
```

The left sidebar shows a file tree for a 'MOVIESAPI' project, including 'controllers', 'database', 'config', 'models', 'seeders', and 'routes' folders, along with files like 'app.js', 'config.js', 'index.js', 'movie.js', 'actor.js', 'genre.js', '.gitignore', '.sequelizer', and 'package.json'. The 'routes' folder contains 'movies.js'.

The bottom right corner shows a terminal window with the following output:

```
Executing (default): SELECT `id`, `title`, `rating`, `awards`, `release_date`, `length`, `created_at`, `updated_at`, `genre_id` FROM `movies` AS `Movie`;
[nodemon] restarting due to changes...
[nodemon] starting 'node app.js'
Servidor andando en el puerto 3000
Executing (default): SELECT `id`, `title`, `rating`, `awards`, `release_date`, `length`, `created_at`, `updated_at`, `genre_id` FROM `movies` AS `Movie`;
```

At the bottom of the screen, there is a taskbar with icons for various applications like File Explorer, Google Chrome, Microsoft Word, and Microsoft Excel. The system tray shows the date and time as 14:56, 7/11/2021, and the weather as 19°C Nublado.

The screenshot shows a dark-themed interface of the Visual Studio Code (VS Code) code editor. At the top, there's a navigation bar with tabs for 'EXPLORER', '... JS app.js', 'JS movies.js', 'JS moviesController.js ●', 'JS movie.js', 'JS actor.js', and 'JS genre.js'. On the far right of the top bar are icons for file operations like 'Save' and 'Close'.

The left sidebar contains a tree view of the project structure under 'OPEN EDITORS 1 UNSAVED'. The structure includes:

- MOVIESAPI**:
 - controllers: **moviesController.js**
 - database
 - config: **config.js**
 - migrations
 - models:
 - actor.js**
 - genre.js**
 - index.js**
 - movie.js**
 - seeders
 - node_modules
 - routes:
 - movies.js**
 - .gitignore
 - .sequelizerc
 - app.js**
 - package.json
 - yarn.lock

The main code editor window displays **moviesController.js** with the following content:

```
4 module.exports = {
5   list: (req, res) => {
6     DB.Movie
7       .findAll()
8       .then(movies => {
9         return res.status(200).json({
10           total: movies.length,
11           data: movies,
12           status: 200
13         })
14       })
15     },
16
17   show: (req, res) => {
18     DB.Movie
19       .findById(req.params.id)
20       .then(movie => {
21         return res.status(200).json({
22           data: movie,
23           status: 200
24         })
25       })
26     }
27 }
```

A yellow lightbulb icon is positioned over the line 'DB.Movie' at line 17, indicating a potential issue. The bottom of the screen shows the terminal output:

```
(node:27092) UnhandledPromiseRejectionWarning: ReferenceError: movie is not defined
    at /Users/macbook/Desktop/moviesApi/controllers/moviesController.js:22:12
    at processTicksAndRejections (internal/process/task_queues.js:93:5)
(Use `node --trace-warnings ...` to show where the warning was created)
(node:27092) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an asy
nc function without a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the node process on un
handled promise rejection, use the CLI flag `--unhandled-rejections=strict` (see https://nodejs.org/api/cli.html#cli_unhandled_reject
```

The bottom status bar shows the current file is 'moviesController.js', the line number is '16:14', and the status is '1: node'.

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs ^

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

Postman Overview

http://localhost:3000/movies/1

GET http://localhost:3000/movies/1

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "data": [
3     {
4       "id": 1,
5       "title": "Avatar",
6       "rating": "7.9",
7       "awards": 3,
8       "release_date": "2010-10-04T00:00:00.000Z",
9       "length": 120,
10      "created_at": null,
11      "updated_at": null,
12      "genre_id": 5
13    },
14    "status": 200
15  }
16 }
```

DigitalHouse> Find and Replace Console

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar < Módulo: 7 APIs

Clase 35 - Creando nuestras propias APIs

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

EXPLORER JS app.js JS movies.js ● JS moviesController.js JS movie.js JS actor.js JS genre.js OPEN EDITORS UNSAVED routes > JS movies.js > ...
1 const express = require('express');
2 const router = express.Router();
3
4 const controller = require('../controllers/moviesController');
5
6 router.get('/', controller.list);
7 router.get('/:id', controller.show);
8 router.post('/:id', controller.store);
9
10 module.exports = router; (property) store: (req: any, re x s: any) => void
store
store
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Executing (default): SELECT `id`, `title`, `rating`, `awards`, `release_date`, `length`, `created_at`, `updated_at`, `genre_id` FROM `movies` AS 'Movie' WHERE `Movie`.`id` = '1';
Executing (default): SELECT `id`, `title`, `rating`, `awards`, `release_date`, `length`, `created_at`, `updated_at`, `genre_id` FROM `movies` AS 'Movie' WHERE `Movie`.`id` = '13';
[nodemon] restarting due to changes...
[nodemon] starting 'node app.js'
18:43
Ln 8, Col 34 Tab Size: 3 UTF-8 LF JavaScript ESLint Prettier

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs < >

Clase 35 - Creando nuestras propias APIs

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

The screenshot shows a browser window displaying a course interface. On the left, there's a sidebar with sections like 'Módulos', 'Contenido' (which is selected), 'Accesos a Zoom y Grabaciones', 'Proyecto integrador: sprints', and 'Info general'. Under 'Contenido', there are several items with checkmarks: 'Introducción al módulo', '¿Qué vamos a ver en esta clase?', 'Introducción', 'REST', 'Creación de API propia', and 'Hacia la clase en vivo'. Below this, there are two collapsed sections: 'Clase 35 - Clase en vivo: Creando nuestras propias APIs' and 'Clase 36 - Consumo de APIs propias y de terceros'. The main area of the browser contains a code editor for a Node.js application. The file 'moviesController.js' is open, showing code for handling movie requests. The code includes functions for 'show' and 'store' operations. The 'store' function uses 'DB.Movie.create(req.body)' to create a new movie in the database. The code editor has tabs for other files like 'app.js', 'movies.js', 'index.js', etc. At the bottom of the code editor, the terminal shows command-line output related to a MySQL database named 'movies'. The terminal output includes commands like 'nodemon' and 'node app.js', and it shows the server running on port 3000. The overall interface is a combination of a web-based course management system and a local development environment.

(40) WhatsApp | Ingreso a Blackboard - UVM Onli | Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs ^

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

Postman Overview

http://localhost:3000/movies/ POST http://localhost:3000/movies/

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body (Pretty, Raw, Preview, Visualize, JSON)

```
1 {  
2   "title": "Cars 2",  
3   "rating": 6.5,  
4   "length": 120,  
5   "release_date": "2018-05-11"  
6 }
```

Status: 200 OK Time: 115 ms Size: 307 B Save Response

DigitalHouse> Bootcamp Runner Trash

Escribe aquí para buscar

19°C Nublado 15:07 7/11/2021

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb... 🔑 ⭐ 🌐 Tp 🚧 🎫 ⏱

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

EXPLORER JS app.js JS movies.js JS moviesController.js X JS movie.js JS actor.js JS genre.js

controllers > JS moviesController.js > <unknown> > store > then() callback > created

```
show: (req, res) => {
  DB.Movie
    .findByPk(req.params.id)
    .then(movie => {
      return res.status(200).json({
        data: movie,
        status: 200
      })
    })
}

store: (req, res) => {
  DB.Movie
    .create(req.body)
    .then(movie => {
      return res.status(200).json({
        data: movie,
        status: 200,
        created: 'ok'
      })
    })
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting 'node app.js'
Servidor andando en el puerto 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node app.js'

23:04

Ln 36, Col 29 Tab Size: 3 UTF-8 LF JavaScript ESLint Prettier

DigitalHouse

(40) WhatsApp x | Bb Ingreso a Blackboard - UVM Onli x > Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs ^

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

Postman Overview http://localhost:3000/movies/ POST http://localhost:3000/movies/ Params Authorization Headers (9) Body Body (none) form-data x-www-form-urlencoded raw binary GraphQL JSON none POST http://localhost:3000/movies/ Status: 200 OK Time: 371 ms Size: 470 B Save Response Body Cookies Headers (7) Test Results Pretty Raw Preview Visualize JSON

```
1 {  
2   "title": "Cars 2",  
3   "rating": 6.5,  
4   "length": 120,  
5   "release_date": "2018-05-11"  
6 }  
  
1 {  
2   "data": {  
3     "id": 33,  
4     "title": "Cars 2",  
5     "rating": 6.5,  
6     "length": "1970-01-01T00:00:00.000Z",  
7     "release_date": "2018-05-11T00:00:00.000Z",  
8     "updated_at": "2021-03-03T11:50:24.644Z",  
9     "created_at": "2021-03-03T11:50:24.644Z"  
10   },  
11   "status": 200,  
12   "created": "ok"  
13 }  
  
Find and Replace Console

DigitalHouse> Bootcamp Runner Trash



Escribe aquí para buscar



19°C Nublado 15:12 ESP 7/11/2021


```

(40) WhatsApp | Ingreso a Blackboard - UVM Onli | Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar < Módulo: 7 APIs

Clase 35 - Creando nuestras propias APIs

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

movies_db_2020

id	created_at	updated_at	title	rating	awards	release_date	length	genre_id	deletedAt
1	NULL	NULL	Avatar	7.9	3	2010-10-04 00:00:00	120	5	NULL
2	NULL	NULL	Titanic	7.7	11	1997-09-04 00:00:00	320	3	NULL
3	NULL	NULL	La Guerra de las galaxias: Episodio VI	9.1	7	2004-07-04 00:00:00	NULL	5	NULL
4	NULL	NULL	La Guerra de las galaxias: Episodio VII	9.0	6	2003-11-04 00:00:00	180	5	NULL
5	NULL	NULL	Parque Jurásico	8.3	5	1999-01-04 00:00:00	270	5	NULL
6	NULL	NULL	Harry Potter y las Reliquias de la Mue...	9.0	2	2008-07-04 00:00:00	190	6	NULL
7	NULL	NULL	Transformers: el lado oscuro de la luna	0.9	1	2005-07-04 00:00:00	NULL	5	NULL
8	NULL	NULL	Harry Potter y la piedra filosofal	10.0	1	2008-04-04 00:00:00	120	8	NULL
9	NULL	NULL	Harry Potter y la cámara de los secretos	3.5	2	2009-08-04 00:00:00	200	8	NULL
10	NULL	NULL	El rey león	9.1	3	2000-02-04 00:00:00	NULL	10	NULL
11	NULL	NULL	Alicia en el país de las maravillas	5.7	2	2008-07-04 00:00:00	120	NULL	NULL
12	NULL	NULL	Buscando a Nemo	8.3	2	2000-07-04 00:00:00	110	7	NULL
13	NULL	NULL	Toy Story	6.1	0	2008-03-04 00:00:00	150	7	NULL
14	NULL	NULL	Toy Story 2	3.2	2	2003-04-04 00:00:00	120	7	NULL
15	NULL	NULL	La vida es bella	8.3	5	1994-10-04 00:00:00	NULL	3	NULL
16	NULL	NULL	Mi pobre angelito	3.2	1	1989-01-04 00:00:00	120	1	NULL
17	NULL	NULL	Intensamente	9.0	2	2008-07-04 00:00:00	120	7	NULL
18	NULL	NULL	Carrozas de fuego	9.9	7	1980-07-04 00:00:00	180	NULL	NULL
19	NULL	NULL	Big	7.3	2	1988-02-04 00:00:00	130	8	NULL
20	NULL	NULL	I am Sam	9.0	4	1999-03-04 00:00:00	130	3	NULL
21	NULL	NULL	Hotel Transylvania	7.1	1	2012-05-04 00:00:00	90	10	2020-05-24 00:14:43
22	NULL	NULL	La Monja	2.0	0	0000-00-00 00:00:00	120	3	NULL
29	2020-05-23 20:56:32	2020-05-23 21:34:18	Pulp fiction	0.0	0	2020-05-23 23:56:32	120	4	NULL
30	2020-05-23 21:08:18	2020-05-23 21:08:18	Up	8.0	0	2020-05-24 00:08:18	NULL	NULL	NULL
31	2020-05-23 21:26:53	2020-05-23 21:26:53	Rogue One	10.0	0	2020-05-24 00:26:53	NULL	NULL	NULL
32	2020-05-23 21:34:35	2020-05-23 21:42:55	Up vill	10.0	0	2020-05-24 00:34:35	250	5	NULL
33	2021-03-03 08:50:24	2021-03-03 08:50:24	Cars 2	6.5	0	2018-05-11 00:00:00	1970	NULL	NULL

DigitalHouse >

Escribe aquí para buscar

19°C Nublado 15:12 7/11/2021

The screenshot shows a Visual Studio Code interface with the following details:

- EXPLORER** sidebar on the left containing the project structure:
 - MOVIESAPI** folder
 - controllers** folder (empty)
 - database** folder (empty)
 - config** folder
 - JS config.js**
 - migrations** folder (empty)
 - models** folder
 - JS actor.js**
 - JS genre.js**
 - JS index.js**
 - JS movie.js**
 - seeders** folder (empty)
 - node_modules** folder (empty)
 - routes** folder
 - JS movies.js**
 - movies.js** file in the editor (marked as **UNSAVED**)

```
const express = require('express');
const router = express.Router();

const controller = require('../controllers/moviesController');

router.get('/', controller.list);
router.get('/:id', controller.show);
router.post('/', controller.store);
router.delete('/:id', controller.delete);

module.exports = router;
```
 - Other tabs in the editor bar: **app.js**, **moviesController.js**, **movie.js**, **actor.js**, **genre.js**.
 - Top right icons: **File**, **Edit**, **Search**, **Find**, **Replace**, **Copy**, **Paste**, **Find in files**.

(41) WhatsApp | Ingreso a Blackboard - UVM Onli | Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <> Módulo: 7 APIs

Clase 35 - Creando nuestras propias APIs

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

EXPLORER JS app.js JS movies.js JS moviesController.js JS movie.js JS actor.js JS genre.js

controllers > JS moviesController.js > <unknown> > delete > where > id

```
store: (req, res) => {
  DB.Movie
    .create(req.body)
    .then(movie => {
      return res.status(200).json({
        data: movie,
        status: 200,
        created: 'ok'
      })
    })
  },
  delete: (req, res) => {
    DB.Movie
      .destroy({
        where: [
          {
            id: req.params.id
          }
        ]
      })
      .then(response => {
        return res.json(response)
      })
  }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

(node:27641) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an asynchronous function without a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the node process on unhandled promise rejection, use the CLI flag '--unhandled-rejections=strict' (see https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode). (rejection id: 2)
(node:27641) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit code.
[nodemon] restarting due to changes...

DigitalHouse

Escribe aquí para buscar

19°C Nublado 15:18 7/11/2021

(41) WhatsApp | Ingreso a Blackboard - UVM Onli | Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs ^

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

Postman Overview

http://localhost:3000/movies/33

DELETE http://localhost:3000/movies/33

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body (JSON)

```
1 {
2   "title": "Cars 2",
3   "rating": 6.5,
4   "length": 120,
5   "release_date": "2018-05-11"
6 }
```

Status: 200 OK Time: 180 ms Size: 234 B Save Response

DigitalHouse> Bootcamp Runner Trash

Escribe aquí para buscar

19°C Nublado 15:18 ESP 7/11/2021

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs < >

Clase 35 - Creando nuestras propias APIs

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

The screenshot shows a browser-based course interface. On the left, there's a sidebar with sections like 'Módulos', 'Contenido' (which is selected), 'Accesos a Zoom y Grabaciones', 'Proyecto integrador: sprints', and 'Info general'. Under 'Contenido', there are several items with checkmarks: 'Introducción al módulo', '¿Qué vamos a ver en esta clase?', 'Introducción', 'REST', 'Creación de API propia', and 'Hacia la clase en vivo'. Below this, there are two collapsed sections: 'Clase 35 - Clase en vivo: Creando nuestras propias APIs' and 'Clase 36 - Consumo de APIs propias y de terceros'. The main area features a code editor with dark mode enabled, showing JavaScript code for a 'Movie' model. The code defines properties like title, rating, awards, release_date, length, and associations with 'Genre' and 'Actor'. Below the code editor is a terminal window showing Node.js startup logs. At the bottom, there's a taskbar with various icons and a system tray showing the date and time.

Ocultar <>

Módulo: 7

APIs

Clase 35 - Creando nuestras propias API

- Introducción al módulo
 - ¿Qué vamos a ver en esta clase
 - Introducción
 - REST
 - Creación de API propia
 - Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

The screenshot shows a VS Code interface with the following details:

- Explorer View:** Shows the project structure under "MOVIESAPI". The "routes" folder contains "movies.js". Other files include "app.js", "config.js", "database.js", "index.js", "migrations", "models", "movie.js", "actor.js", "genre.js", "seeders", and "package.json".
- Code Editor:** The "movies.js" file is open, showing routes for movies. A tooltip for the "search" method is displayed, indicating it takes a req and res object as parameters.
- Terminal:** The terminal shows the application running on port 3000. It includes the following logs:
 - Servidor andando en el puerto 3000
 - Executing (default): INSERT INTO `movies` (`id`, `title`, `rating`, `release_date`, `length`, `created_at`, `updated_at`) VALUES (DEFAULT, ?, ?, ?, ?, ?);
 - Executing (default): DELETE FROM `movies` WHERE `id` = '34'
 - [nodemon] restarting due to changes...
 - [nodemon] starting 'node app.js'
 - Servidor andando en el puerto 3000
- Bottom Status Bar:** Shows file status (fullSync), tabs (7 total, 40 active), tab size (3), LITE-R, LF, JavaScript, ESLint, and Prettier.

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb... 🔑 ⭐ 🌐 Tp 🚧 🎫 🎵 🎯

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

EXPLORER OPEN EDITORS UNSAVED MOVIESAPI controllers database config migrations models seeders routes app.js config.js movie.js actor.js genre.js index.js movieController.js moviesController.js movie.js actor.js genre.js

JS app.js JS movies.js JS moviesController.js JS movie.js JS actor.js JS genre.js

```
const Op = DB.Sequelize.Op;
```

module.exports = { list: (req, res) => { DB.Movie .findAll() .then(movies => { return res.status(200).json({ total: movies.length, data: movies, status: 200 }) }) }, show: (req, res) => { DB.Movie .findById(req.params.id) .then(movie => { return res.status(200).json({ data: movie, status: 200 }) }) } }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[nodemon] restarting due to changes... [nodemon] starting 'node app.js' Servidor andando en el puerto 3000 [nodemon] restarting due to changes... [nodemon] restarting due to changes... [nodemon] starting 'node app.js'

01:24 30:30

DigitalHouse

(41) WhatsApp | Ingreso a Blackboard - UVM Onli | Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <> Módulo: 7 APIs

Clase 35 - Creando nuestras propias APIs

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia**
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

EXPLORER JS app.js JS movies.js JS moviesController.js JS movie.js JS actor.js JS genre.js controllers > JS moviesController.js > <unknown> > search > then() callback

```
id: req.params.id
})
.then(response => {
  return res.json(response)
}),
search: (req, res) => {
  DB.Movie
    .findAll({
      where: {
        title: { [Op.like]: `%" + req.query.keyword + "%` }
      }
    })
    .then(movies => {
      return res.status(200).json(movies);
    })
}
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

at Object.<anonymous> (/Users/macbook/Desktop/moviesApi/routes/movies.js:4:20)
at Module._compile (internal/modules/cjs/loader.js:1063:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:1092:10)
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting 'node app.js'
Servidor andando en el puerto 3000

Ln 60, Col 47 (6 selected) Tab Size: 3 UTF-8 LF JavaScript ESLint Prettier

DigitalHouse

Escribe aquí para buscar

19°C Nublado 15:26 7/11/2021

(41) WhatsApp | Ingreso a Blackboard - UVM Onli | Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs

Clase 36 - Consumo de APIs propias y de terceros

Clase 36 - Clase en vivo: Consumo de

Postman Overview

http://localhost:3000/movies/search?keyword=a

GET http://localhost:3000/movies/search?keyword=a

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body (Pretty, Raw, Preview, Visualize, JSON)

Body (Pretty, Raw, Preview, Visualize, JSON)

Status: 200 OK Time: 242 ms Size: 3.28 KB Save Response

DigitalHouse> Bootcamp Runner Trash

Escribe aquí para buscar

19°C Nublado 15:26 7/11/2021

Windows Taskbar

Home Workspaces Reports Explore

Search Postman

Overview GET http://localhost:30... ● + ... Presiona Esc para salir de la pantalla completa No Environment

http://localhost:3000/movies/search?keyword=harry

GET http://localhost:3000/movies/search?keyword=harry Send

Params • Authorization Headers (9) Body • Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "title": "Una pelicula de prueba",
3   "rating": 5,
4   "length": 200,
5   "release_date": "2021-05-11"
6 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 18 ms Size: 815 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 6,
4     "title": "Harry Potter y las Reliquias de la Muerte - Parte 2",
5     "rating": "9.0",
6     "awards": 2,
7     "release_date": "2008-07-04T00:00:00.000Z",
8     "length": 190,
9     "created_at": null,
10    "updated_at": null,
11    "genre_id": 6
12  },
13  {
14    "id": 8,
15    "title": "Harry Potter y la piedra filosofal",
16    "rating": "10.0",
17    "awards": 1,
18    "release_date": "2008-04-04T00:00:00.000Z"
19  }
]
```

Blockframe

DigitalHouse >

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb... 🔑 ⭐ 🌐 Tp 🚧 🎫 🎵 🎯

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs ^

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

EXPLORER OPEN EDITORS UNSAVED controllers > JS moviesController.js > <unknown> > search > then() callback

```
id: req.params.id
}
.then(response => {
  return res.json(response)
})
,
search: (req, res) => {
  DB.Movie
    .findAll({
      where: {
        title: { [Op.like]: `%" + req.query.keyword + "%` }
      }
    })
    .then(movies => {
      if (movies.length > 0) {
        return res.status(200).json(movies);
      }
      return res.status(200).json(['No existen películas']);
    })
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1:node

```
'movies' AS 'Movie' WHERE 'Movie'.title LIKE '%a%';
Executing (default): SELECT `id`, `title`, `rating`, `awards`, `release_date`, `length`, `created_at`, `updated_at`, `genre_id` FROM
`movies` AS 'Movie' WHERE 'Movie'.title LIKE '%jurásico%';
Executing (default): SELECT `id`, `title`, `rating`, `awards`, `release_date`, `length`, `created_at`, `updated_at`, `genre_id` FROM
`movies` AS 'Movie' WHERE 'Movie'.title LIKE '%sharry%';
Executing (default): SELECT `id`, `title`, `rating`, `awards`, `release_date`, `length`, `created_at`, `updated_at`, `genre_id` FROM
`movies` AS 'Movie' WHERE 'Movie'.title LIKE '%sarasa%';

```

DigitalHouse

(41) WhatsApp x | Bb Ingreso a Blackboard - UVM Onli x > Playground Digital House

latam-playground.digitalhouse.com/course/2815/unit/0184334c-f8a7-11eb-876f-12eb947d8fb1/lesson/27d5f0b4-f8a9-11eb-876f-12eb947d8fb1/topic/d220ae82-f8a7-11eb-...

Módulos Contenido Accesos a Zoom y Grabaciones Proyecto integrador: sprints Info general

Ocultar <>

Módulo: 7

APIs <>

Clase 35 - Creando nuestras propias APIs ^

- Introducción al módulo
- ¿Qué vamos a ver en esta clase?
- Introducción
- REST
- Creación de API propia
- Hacia la clase en vivo

Clase 35 - Clase en vivo: Creando nuestras propias APIs ^

Clase 36 - Consumo de APIs propias y de terceros ^

Clase 36 - Clase en vivo: Consumo de

Postman Overview GET http://localhost:30... + ...

http://localhost:3000/movies/search?keyword=sarasa

GET http://localhost:3000/movies/search?keyword=sarasa

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "title": "Una pelicula de prueba",
3   "rating": 5,
4   "length": 200,
5   "release_date": "2021-05-11"
6 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 407 ms Size: 258 B Save Response

No Environment

DigitalHouse> Bootcamp Runner Trash

Escribe aquí para buscar

19°C Nublado 15:29 ESP 7/11/2021

Windows Taskbar