

Semana tec (Conectando el mundo web)

[Repositorio de GitHub](#)

Tabla de contenidos

- [Día 1](#)
- [Día 2](#)
- [Día 3](#)
- [Día 4](#)

Día 1

Act 1.1

Responde las siguientes preguntas:

- ¿Cuales son los servidores más comunes?
 - Apache HTTP Server
 - Nging
 - Litespeed
 - Microsoft IIS
 - OpenResty
- ¿Cuales son los browsers más usados?
 - Chrome
 - Safari
 - Edge
 - Firefox
 - Samsung Internet
 - Android
- ¿Cómo se comunican los browsers y los servidores web?
 - Using the HyperText Transfer Protocol (HTTP).

Entra al modo desarrollador e inspecciona

- ¿Qué elementos observas? ¿Para qué crees que sirvan?
 - Elements (Elementos HTML de la página)
 - Console (Consola para ejecutar JS)
 - Sources (Todos los elementos de HTML incluidos con la tag script)
 - Network (Grabar actividad de la red)

Act 1.2

Preguntas

- ¿Qué es un web framework?
 - Un conjunto de herramientas y recursos disponibles para los desarrolladores web.

- ¿Cuales son los frameworks de back-end más usados?
 - Laravel
 - Django
 - Express
 - Flask
 - Ruby on rails
- ¿Cuales son los frameworks de front-end más usados?
 - React
 - JQuery
 - Angular
 - Vue
 - Angular
 - Svelte

Explora el sitio [State of JS](#) ¿Qué te llama la atención?

- Que solo hay un framework de front y uno de backend en el cuadrante de have used y possitive opinions en la gráfica(React y express)

Act 1.3 Creando nuestro servidor web

Inicia tu servidor web en el puerto 3001

```
→ react-express npm start

> react-express@1.0.0 start
> node server/index.js

Example app listening on port 3001
```

Act 1.4

- ¿Qué es una API(Application programming iinterface)?
 - Sirve para comunicar varias aplicaciones para que puedan compartir datos.
- ¿Qué es una web API?
 - API que puede ser accedida mediante el protocolo http
- ¿A qué se refieren los endpoints?
 - Donde el API se comunica con otro sistema.


Día 2

Act 2.1


- Cómo le explicarías a tu abuelita/o SOAP y REST?
 - Son formas en las que los sistemas se comunican los sistemas, REST es solo una serie de reglas para guiarte que es muy libre y se puede memorizar cierta parte de los datos (cache). Mientras que SOAP es un protocolo que es más restrictivo y caro de implementar, pero es más seguro.
- ¿Dónde crees que se utiliza SOAP? ¿Dónde REST?
 - REST se utiliza en el mundo web en general y SOAP se utiliza para más para empresas grandes, bancos, hospitales, etc.

Act 2.2


Entra a Swagger y modifica el ejemplo de la tienda de mascotas

movie Everything about movies 



POST

/movie Add a new movie to the library 

GET

/movie/{movieId} Find movie by ID 

DELETE

/movie/{movieId} Deletes a movie  

Act 2.3 - 2.6

- **Creando el primer endpoint**



```
{"message": "Hello from the server side!"}
```

- **GET**

GETlocalhost:3001/movies

No Environment

localhost:3001/movies

GETlocalhost:3001/movies

Mock Servers

ParamsAuthorizationHeaders (5)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 54 msSize: 533 BSave Response

PrettyRawPreviewVisualizeHTML

```
1 {
2   "movie0": {
3     "name": "Hangover",
4     "year": 2009,
5     "genre": "Comedy"
6   },
7   "movie1": {
8     "name": "Hangover 2",
9     "year": 2011,
10    "genre": "Comedy"
11  },
12  "movie2": {
13    "name": "Hangover 3",
14    "year": 2013,
```

• POST

POSTlocalhost:3001/api/mx

No Environment

localhost:3001/api/movies

POSTlocalhost:3001/api/movies

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 44 msSize: 473 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "movie0": {
3     "name": "Hangover",
4     "year": 2009,
5     "genre": "Comedy"
6   },
7   "movie1": {
8     "name": "Hangover 2",
9     "year": 2011,
10    "genre": "Comedy"
11  },
12  "movie2": {
13    "name": "Hangover 3",
14    "year": 2013,
15    "genre": "Comedy"
16  },
17  "movie3": {
18    "name": "Due Date",
19    "year": 2010,
20    "genre": "Comedy"
21  }
22 }
```

• PUT

PUTlocalhost:3001/api/mov

No Environment

localhost:3001/api/movies?id=movie4&name=Are you here&year=2013&genre=Comedy

PUT

localhost:3001/api/movies?id=movie4&name=Are you here&year=2013&genre=Comedy

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	movie4	
<input checked="" type="checkbox"/> name	Are you here	
<input checked="" type="checkbox"/> year	2013	
<input checked="" type="checkbox"/> genre	Comedy	

Body

Cookies

Headers (7)

Test Results

Status: 200 OKTime: 32 msSize: 538 BSave Response

Pretty

Raw

Preview

Visualize

JSON

```
6  },
7  "movie1": {
8    "name": "Hangover 2",
9    "year": 2011,
10   "genre": "Comedy"
11  },
12  "movie2": {
13    "name": "Hangover 3",
14    "year": 2013,
15    "genre": "Comedy"
16  },
17  "movie3": {
18    "name": "Due Date",
19    "year": 2010,
20    "genre": "Comedy"
21  },
22  "movie4": {
23    "name": "Are you here",
24    "year": "2013",
25    "genre": "Comedy"
26  }
27 }
```

Find and Replace

Console

Cookies

Capture requests

Bootcamp

Runner

Trash

• DELETE

Home

Workspaces

API Network

Reports

Explore

Search Postman

Invite

Upgrade

DELlocalhost:3001/api/mov

No Environment

localhost:3001/api/movies?id=movie4

DELETE

localhost:3001/api/movies?id=movie4

Send

Params

Authorization

Headers (5)

Body

Pre-request Script

Tests

Settings

Cookies

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	movie4	
<input type="checkbox"/> name	Are you here	
<input type="checkbox"/> year	2013	
<input type="checkbox"/> genre	Comedy	

Body

Cookies

Headers (7)

Test Results

Status: 200 OKTime: 11 msSize: 473 BSave Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "movie0": {
3      "name": "Hangover",
4      "year": 2009,
5      "genre": "Comedy"
6    },
7    "movie1": {
8      "name": "Hangover 2",
9      "year": 2011,
10     "genre": "Comedy"
11   },
12   "movie2": {
13     "name": "Hangover 3",
14     "year": 2013,
15     "genre": "Comedy"
16   },
17   "movie3": {
18     "name": "Due Date",
19     "year": 2010,
20     "genre": "Comedy"
21   }
22 }
```

Find and Replace

Console

Cookies

Capture requests

Bootcamp

Runner

Trash

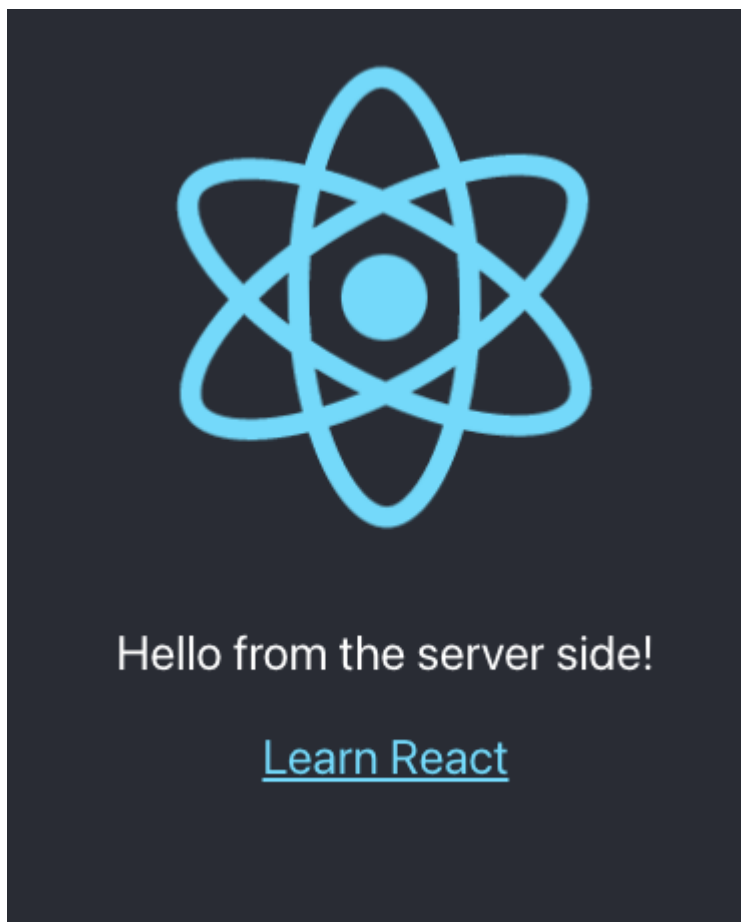
Día 3

Act 3.1

- Vuelve a entrar a State of JS y elige algún framework de front-end. Describe por qué sería útil o en qué caso debería usarse basado en la información presentada.
 - Elegí Svelte porque según los gráficos tiene muchas opiniones buenas aunque tiene pocos usuarios, cae en las tecnologías que vale la pena evaluar.
- ¿Qué es un proxy? ¿Para qué lo utilizamos para conectar nuestros servicios?
 - Un servidor proxy es básicamente una maquina en Internet con su propia dirección IP que tu ordenador conoce. Cuando envías un request, esta va primero al servidor proxy. Después el servidor proxy realiza el request en tu nombre, toma la respuesta del servidor y te reenvía los datos de la página web para que puedas verla en tu navegador.

Act 3.2-3.3

Crea un cliente de react con "create-react-app" y conéctalo con tu back-end.



Act 3.4

- Explora más sobre ReactJS en su documentación (Enlaces a un sitio externo.) y responde lo siguiente:
 - ¿Qué es JSX?
 - Es una extensión de sintaxis de JavaScript para describir como se tiene que ver el UI.
 - ¿Qué son los componentes y los props?
 - Los componentes son los bloques que conforman una aplicación de react y los props son los valores de entrada para modificar estos bloques.
 - ¿Qué es el estado y el ciclo de vida?

- El estado es un atributo local de un componente y el ciclo de vida es el proceso que sigue un componente de react desde durante su creación, actualización o destrucción.
 - ¿Cómo manejamos eventos?
 - Usando la propiedad de html on* y agregándole una función.
- Ahora investiga más a fondo la función "fetch" que utilizamos
 - ¿Para qué sirve fetch? Para hacer http requests.

Act 3.5

Revisa la documentación de algunas API de terceros

- ¿Cuáles te llaman la atención? ¿Por qué?
 - Me llamó la atención el api de the movie database porque funciona con el proyecto que armé en la semana.
- ¿Qué información podrías obtener de estas APIs?
 - Las plataformas en las que existen las películas en la lista de películas.
- ¿Cómo podrías conectar tu front-end con ellas?
 - Con http requests enviando un API key que ellos proporcionan como un header

Act 3.6

1. Realiza algún request al API del tercero y obtén información, utilicé The movie database

The screenshot shows the Postman interface with a GET request to the The Movie Database API. The URL is `https://api.themoviedb.org/3/search/movie?api_key=*****&query=Due date`. The response is a JSON object with the following structure:

```

{
  "results": [
    {
      "adult": false,
      "backdrop_path": "/st8hEUnz7LTwyGYTt70t9JAHuvb.jpg",
      "genre_ids": [
        35,
        18
      ],
      "id": 41733,
      "original_language": "en",
      "original_title": "Due Date",
      "overview": "Peter Highman must scramble across the US in five days to be present for the birth of his first child. He gets off to a bad start when his wallet and luggage are stolen, and put on the 'no-fly' list. Peter embarks on a terrifying journey when he accepts a ride from an actor.",
      "popularity": 30.841,
      "poster_path": "/4dzo4uWmkp28uwm0jpiduRtvju.jpg",
      "release_date": "2010-11-04",
      "title": "Due Date",
      "video": false,
      "vote_average": 6.3,
      "vote_count": 4597
    }
  ]
}
    
```

2. Crea 1 componente de React que despliegue información obtenida de la API externa.



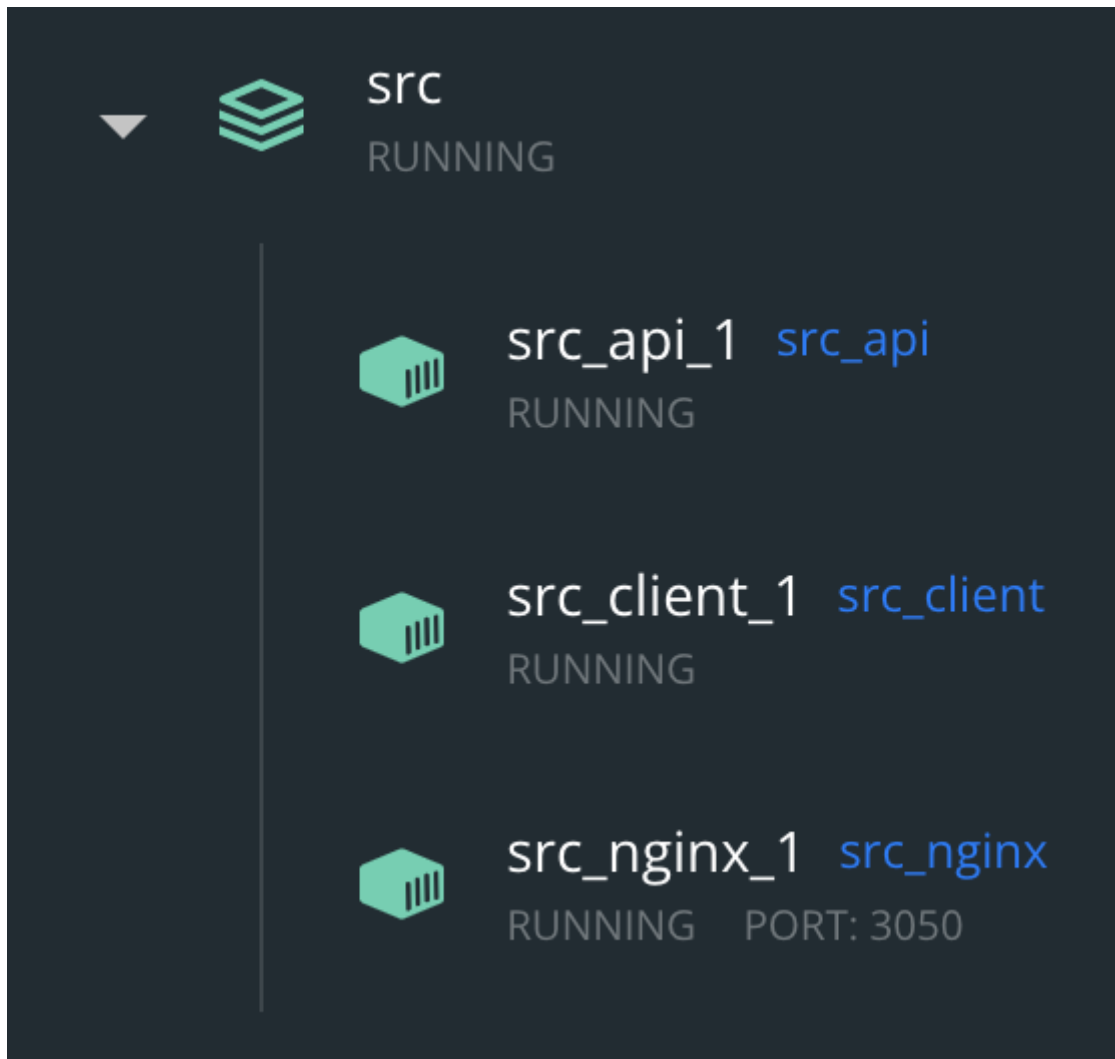
Día 4

Laboratorio ligero

Mi teclado USB tuvo una falla, por lo que no pude montar el servidor en la raspberry pi, lo que hice fue correrlo con docker compose en mi máquina, por lo que debería de funcionar en cualquier máquina que corra Docker.




Para el compose utilicé dos contenedores de node alpine para el front y el back end y uno nginx para el servidor




Anexo la arquitectura del compose:



Para comprobar que funcionara lo abrí en mi celular:


9:14



 33%

192.168.1.73:3050

Hangover



ID:movie0

2009

Comedy

The hangover part II




ID:movie1

2011

Comedy

The hangover part III




ID:movie2

2013

Comedy

Due Date



ID:movie3

2010

Comedy

