

Trabajo: Laboratorio: Computacion en Cliente Web

Peticiones con Ajax a una base de datos mediante estandares WEB

Objetivo

Utilizar las herramientas que posee VS Code para realizar peticiones a bases de datos cumpliendo con los estándares Web que se han creado para tal fin desde el año 2005. Trabajar con estándares web relacionados con conexiones AJAX, funciones asíncronas y componentes para realizar peticiones a una base de datos y representar la información obtenida en una página Web.

Estudiar cada una de las maneras de hacer peticiones AJAX desde la antigüedad, hasta nuestros días para la comunicación entre clientes web y servidores.

Descripcion

En este laboratorio vamos a realizar una sencilla petición mediante estándares web y vamos a representar los datos obtenidos en una página de manera limpia. Iremos complicando y actualizando esta petición de datos para que cada vez se acerque más a la manera actual de realizarse en la práctica. Es decir, repasaremos cada una de las maneras de hacer peticiones AJAX desde la antigüedad, hasta nuestros días.

Pre-requisitos

Que cosas necesitas para instalar el software y como instalarlas.

- **Bootstrap** // Es un framework CSS front-end para crear páginas web, dispone de una serie de recursos que simplifican el desarrollo de sitios web con HTML5, CSS3 y jQuery, de manera que facilita la labor de diseño y desarrollo de toda clase de proyectos Web.
- **HTML Boilerplate** // La extensión HTML5 Boilerplate (H5BP) agiliza el proceso de construcción de sitios web, permite añadir fácilmente toda una estructura de un sitio HTML5 a partir de una plantilla que sirve como base para que los desarrolladores web la adopten en sus diseños. Con H5BP podemos obtener código para la normalización de los navegadores, performance, optimizaciones para navegadores en dispositivos móviles, clases específicas para IE, clases Javascript, para estilos basados en capacidades del navegador carga de página más rápida entre otras funcionalidades que simplifican el desarrollo de software.
- **Element ChuckNorrisFact** Es un web component que muestra chistes de chuck norris de modo random

```
https://www.webcomponents.org/element/@npm/chuck-norris-fact-element
```

Instalación & Iniciación

Las Extenciones a usar las encontramos en la galeria de extensiones de VSCode.

<https://code.visualstudio.com/#hundreds-of-extensions>

Desarrollo

En este laboratorio se utiliza una base de datos llamada ICNDB, inspirada en el famoso actor y experto en artes marciales conocido como Chuck Norris, que se encuentra en la página Web: <http://www.icndb.com/>, aquí se almacenan cientos de chistes inspirados en su carrera. Esta base de datos cuenta con una API REST o protocolo de intercambio y manipulación de datos, funciona como una interfaz entre sistemas que usan HTTP para obtener datos en múltiples formatos, como XML y JSON.



Se usarán las siguientes tecnologías de comunicación entre clientes web y servidores para cumplir con los objetivos planteados:

- El objeto XMLHttpRequest.
- Las funciones AJAX del archiconocido framework jQuery.
- Un complemento específico para jQuery creado por el mantenedor del servicio al que nos vamos a conectar.
- Por último, implementación de un componente web moderno (Web component) desarrollado por un tercero que nos permitirá realizar peticiones parecidas, pero de manera increíblemente elegante.

Resolucion del ejercicio de la forma 2005

Mediante el objeto XMLHttpRequest es posible hacer peticiones AJAX a través de clientes Web hacia servidores de la siguiente manera:

```
<script type="text/javascript">
  window.onload = function() {
    xmlhttp = new XMLHttpRequest();
    xmlhttp.open('GET', 'http://api.icndb.com/jokes/random/', true);
    xmlhttp.onreadystatechange = function() {
      var textoChiste = JSON.parse(this.response).value.joke;
      console.log('chiste recibido: ' + textoChiste);
      var h1s = document.getElementsByTagName('h1');
      h1s[0].innerHTML = textoChiste;
    }
    xmlhttp.send();
  }
</script>
```

Jumbotron es un componente ligero y flexible usado para mostrar contenido estilizado, se puede expandir en toda la ventana gráfica para mostrar mensajes personalizados. Utiliza clases de utilidad para la tipografía y el espaciado del contenido dentro de contenedores más grandes. (JUMBOTRON, s.f.) Para aplicar este estilo y sustituir el texto del chiste en el título central de un elemento jumbotron del framework Bootstrap se hace de la siguiente manera:

```
<main role="main"> <!-- La etiqueta main define el contenido principal y
destacado del DOM -->
```

```

<div class="jumbotron">
  <div class="container">
    <h1 class="display-3">Aquí va el chiste =D</h1>
  </div>
</div>
</main>

```

La clase CSS **jumbotron** se implementa dentro de la etiqueta **div**, estas crean un contenedor en bloque que se define con una clase CSS a través del atributo **class**. Al utilizar la clase "jumbotron" se muestra un título que representa el contenido destacado de una página.

Para utilizar Bootstrap se debe asegurar que las páginas están configuradas con los últimos estándares de diseño y desarrollo usando un doctype HTML5 e incluir una etiqueta **meta** para configuración del **viewport** que a su vez se encuentra dentro de la etiqueta **head**, así se obtiene el comportamiento adecuado para las diferentes resoluciones de pantalla que disponen los dispositivos modernos.

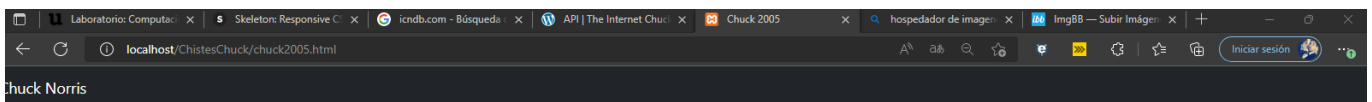
```

<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="Laboratorio Computación Cliente Web">

```



If you ask Chuck Norris what time it is, he always answers "Two seconds till". After you ask "Two seconds to what?", he roundhouse kicks you in the face.

Laboratorio Computación en el Cliente Web 2022

Página web con resultado 2005

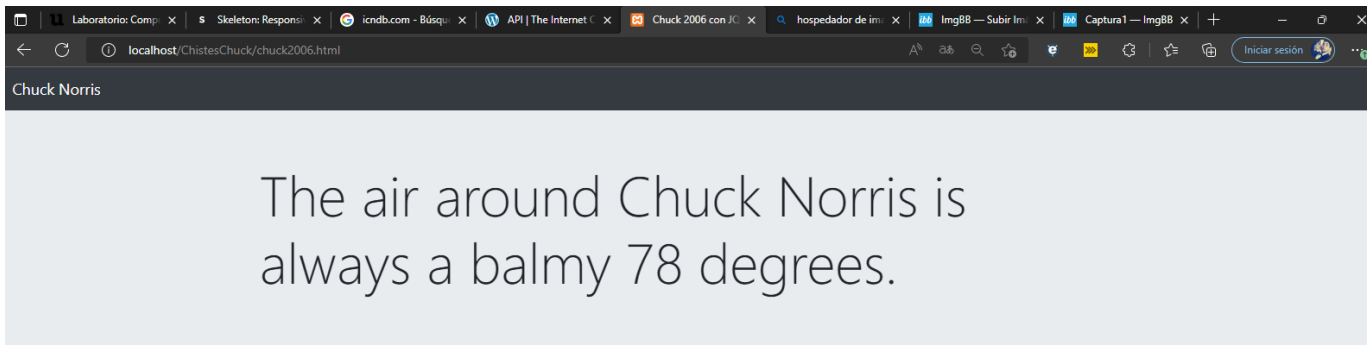
Resolución del Ejercicio de la forma 2006 🖥️

Al implementar la librería jQuery, se incorpora una nueva instrucción muy útil para realizar peticiones tal como se hacían en el 2006, para ello se debe enlazar el sitio Web con el recurso que provee dicha librería, por lo que se procede a crear otro fichero llamado "chuck2006.html" que utiliza el código JavaScript mostrado en el código anterior.

```
<script type="text/javascript">
  window.onload = function() {
    $.get("http://api.icndb.com/jokes/random", (response) => {
      var textoChiste = response.value.joke;
      $('h1').text(textoChiste);
    });
  }
</script>
```

Para enlazar con la biblioteca se utiliza la etiqueta "script" y mediante el atributo src se identifica la URL en donde se encuentra alojado jQuery. Cabe resaltar que el atributo integrity permite que el navegador verifique la fuente del archivo para asegurarse de que el código nunca se cargue si la fuente ha sido manipulada. Define un valor hash de un recurso (como una suma de comprobación) que debe coincidir para que el navegador lo ejecute. El hash asegura que el archivo no se modificó y contiene los datos esperados. De esta manera, el navegador no cargará recursos diferentes (por ejemplo, maliciosos). El atributo crossorigin permite configurar las peticiones CORS de los datos que se cargan. El Intercambio de Recursos de Origen Cruzado (CORS) es un mecanismo que utiliza cabeceras HTTP adicionales para permitir que un user agent obtenga permiso para acceder a recursos seleccionados desde un servidor, en un origen distinto (dominio) al que pertenece. Un agente crea una petición HTTP de origen cruzado cuando solicita un recurso desde un dominio distinto, un protocolo o un puerto diferente al del documento que lo generó.

```
<script src="https://code.jquery.com/jquery-3.5.0.min.js" integrity="sha256-
xNzN2a4ltkB44Mc/Jz3pT4iU1cmeR0FkXs4pru/JxaQ=" crossorigin="anonymous"></script>
```



Laboratorio Computación en el Cliente Web 2020

Resultado Pagina Solución 2006

¿Qué diferencias ves con respecto al ejercicio anterior?

Al usar el framework de JQuery se reduce la cantidad de código, de igual manera este mismo se presenta más limpio.

¿Cómo simplifica la vida jQuery?

jQuery nos permite de una manera mucho más fácil el desarrollo y la interacción con el DOM, de igual manera es intuitivo y fácil de usar.

¿Qué ocurre si tenemos varios tags h1?

Al tener varios tags h1, el código permite insertar el resultado obtenido (el chiste) e insertarlo en cada uno de los h1 que encuentre en el DOM.

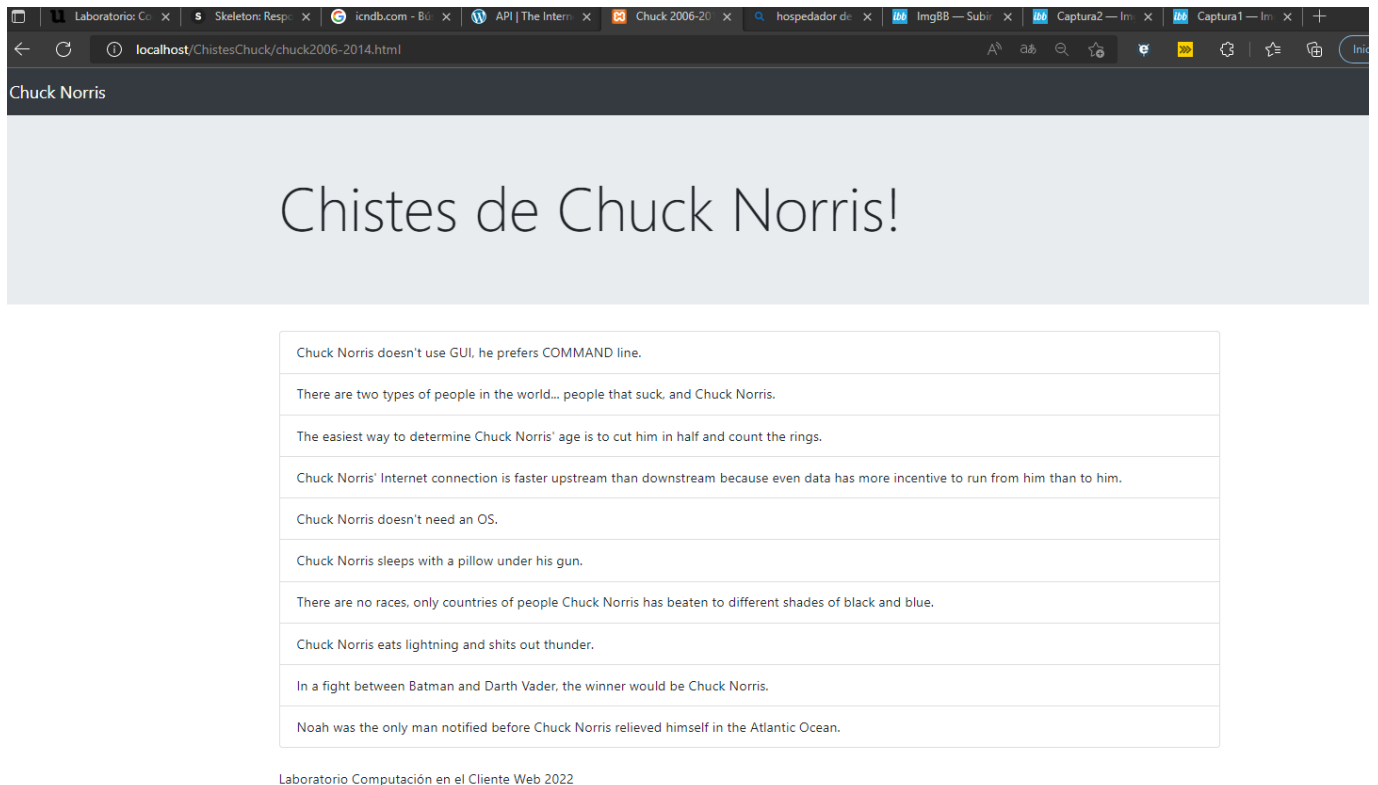
Resolución con el plugin de JQuery (2006-2014)

Este plugin que de manera general se puede definir como una función que se le agrega al objeto jQuery, se ejecuta la instrucción para extraer una lista de chistes, cuando el documento DOM termina de cargarse y activa **function()**. Mediante el método `$.icndb.getRandomJokes()` se obtiene una lista de chistes (10 en este caso) utilizando la propiedad `nu`

`src="Bootstrap/js/jquery.icndb.min.js" ></script>.` Cada elemento que se obtiene se va agregando a la lista mediante el atributo `append` sin que desaparezca el elemento anterior.

```
<script src="https://code.jquery.com/jquery-3.5.0.min.js" integrity="sha256-
xNzN2a4ltkB44Mc/Jz3pT4iU1cmeR0FkXs4pru/JxaQ=" crossorigin="anonymous"></script>
<script src="bootstrap/js/jquery.icndb.min.js"></script>
<script type="text/javascript">
    window.onload = function() {
        $.icndb.getRandomJokes({
            number: 10,
            success: (response) => {
                response.forEach(element => {
                    $("ul.list-group").append('<li class="list-group-item">' +
element.joke + '</li>');
                });
            }
        });
    }
</script>
```

Uso de plugin jQuery ICNDB para crear una lista de chistes



Resultado pagina 2006-2014

```
<div class="container">
  <div class="row">
    <div class="col-12">
      <ul class="list-group">
      </ul>
    </div>
  </div>
</div>
```

Clase container para dar características a la lista

Mediante la clase **container** se crea un contenedor de anchura fija y con la clase **row** se crean unas rejillas dentro del contenedor que en este caso se usan para separar cada chiste de la lista teniendo una división de un ancho definido con la clase **col-12**.

¿cómo se escribían las funciones en las versiones de ECMAScript previas a la versión 6? En las versiones de ECMAScript previas a la 6, solo se podían escribir las funciones de una forma tradicional con la palabra reservada **function**, dándole un nombre y permitiendo parámetros de entrada. En la versión 6 se agregaron una variedad de formas para escribir funciones, como por ejemplo las funciones flecha (del inglés, Arrow functions) que son fácilmente identificados al implementar la sintaxis **=>**.

Resolución 2014

Para desarrollar este ejercicio se utilizará el método **fetch**, el cual fue definido por la **WHATWG**, y es un nuevo estándar de interacción por **HTTP** basado en promesas el cual funciona desde **window** como desde **worker**.

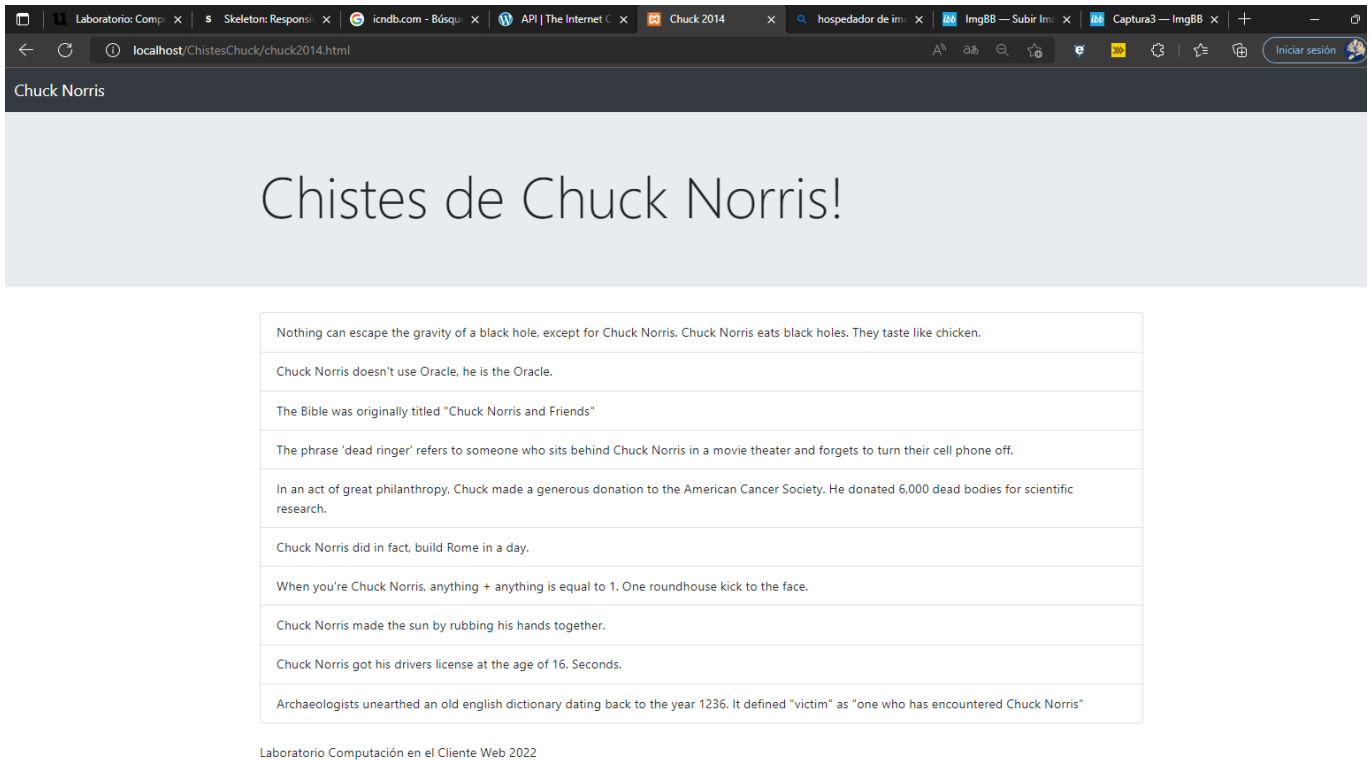
Para su implementación hacemos su llamado como una propiedad de window, se pasa como parámetro la URL del servicio y variables de configuración como el método (en este caso el método GET). Posteriormente, el resultado de la promesa es convertido a formato JSON y se devuelve como promesa para que a través del iterador forEach sea insertado cada ítem (elemento li) a la lista del DOM.

```
<script type="text/javascript">
  window.fetch('http://api.icndb.com/jokes/random/10', {
    method: 'get'
  })
  .then(function(response) {
    return response.json();
  })
  .then(function(data) {
    var ul = document.getElementById("list");
    jokes = data.value;
    jokes.forEach(element => {
      ul.innerHTML += '<li class="list-group-item">' + element.joke
+ '</li>';
    });
  });
</script>
```

Uso del Fetch

```
<div class="container">
  <div class="row">
    <div class="col-12">
      <ul class="list-group" id="list">
      </ul>
    </div>
  </div>
</div>
```

Clase container para aplicar los resultados



Resultado pagina 2014

Para realizar la prueba a través del fetch con NodeJS, debemos haber instalado node en nuestro equipo, el cual se puede descargar desde el siguiente enlace: <https://nodejs.org/es/download/>

Después creamos un archivo .js para este caso chuck-node.js el cual contendrá el siguiente código:

```
const fetch = require('node-fetch');

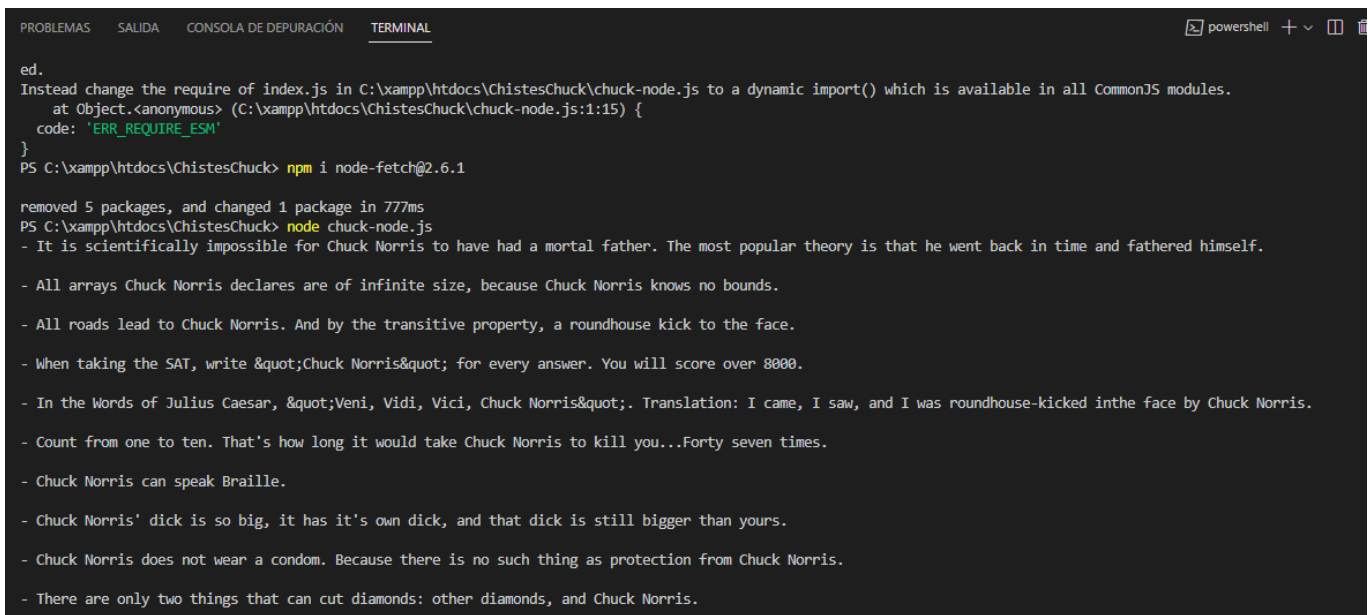
fetch('http://api.icndb.com/jokes/random/10',
  {
    method: 'get'
  })
  .then(function(response) {
    return response.json();
  })
  .then(function(data) {
    jokes = data.value;
    jokes.forEach(element => { console.log("- " + element.joke + "\n"); });
  });
```

El siguiente paso es abrir una consola e ir a la ruta de la carpeta donde se encuentra el archivo creado y ejecutamos el siguiente comando para agregar el módulo de node:

```
npm i node-fetch --save
```

Por último, ejecutamos el archivo creado con el comando node:


```
node chuck-node.js
```



```

ed.
Instead change the require of index.js in C:\xampp\htdocs\ChistesChuck\chuck-node.js to a dynamic import() which is available in all CommonJS modules.
    at Object.<anonymous> (C:\xampp\htdocs\ChistesChuck\chuck-node.js:1:15) {
  code: 'ERR_REQUIRE_ESM'
}
PS C:\xampp\htdocs\ChistesChuck> npm i node-fetch@2.6.1
removed 5 packages, and changed 1 package in 777ms
PS C:\xampp\htdocs\ChistesChuck> node chuck-node.js
- It is scientifically impossible for Chuck Norris to have had a mortal father. The most popular theory is that he went back in time and fathered himself.
- All arrays Chuck Norris declares are of infinite size, because Chuck Norris knows no bounds.
- All roads lead to Chuck Norris. And by the transitive property, a roundhouse kick to the face.
- When taking the SAT, write "Chuck Norris" for every answer. You will score over 8000.
- In the Words of Julius Caesar, "Veni, Vidi, Vici, Chuck Norris". Translation: I came, I saw, and I was roundhouse-kicked in the face by Chuck Norris.
- Count from one to ten. That's how long it would take Chuck Norris to kill you...Forty seven times.
- Chuck Norris can speak Braille.
- Chuck Norris' dick is so big, it has it's own dick, and that dick is still bigger than yours.
- Chuck Norris does not wear a condom. Because there is no such thing as protection from Chuck Norris.
- There are only two things that can cut diamonds: other diamonds, and Chuck Norris.
  
```

Resultado de la petició

¿Qué es el WHATWG (<https://whatwg.org/>)?

El WHATWG (Web Hypertext Application Technology Working Group) es una comunidad de personas y empresas interesadas en la evolución de HTML y tecnologías conexas.

Resolució



Mediante el uso de Web Components se puede obtener el contenido de la API de ICNDB. Esto se puede realizar de dos maneras:

- Haciendo una referencia mediante el tag

```
<link rel="import">
```

hacia el repositorio del sitio web del componente.

- O haciendo el import de manera local instalando el Web Component a través de un gestor de paquetes como Bower por ejemplo.

Link a URL

Bajo el primer método basta con insertar la siguiente línea, que permite importar del Web component

```
chuck-norris-fact
```

y poderlo utilizar dentro del DOM:

Para importar polify que es dependencia

```
<script src="node_modules/@webcomponents/custom-elements/custom-elements.min.js">
</script>
```

Para Importar el Custom Element Chuck-Norris-Fact

```
<script type="module" src="node_modules/chuck-norris-fact-element/chuck-norris-
fact.js"></script>
```

Así mismo, se puede referenciar el framework de maquetado Skeleton mediante las instrucciones:

```
<!-- linking Skeleton -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/skeleton.css">
```

Hecho esto, se puede generar el contenido de la tabla en la que se insertarán las bromas obtenidas haciendo uso de la etiqueta

```
<chuck-norris-fact>
```

No obstante, para hacerlo de una manera más dinámica se desarrolló una función que genere el contenido de la tabla. Antes de codificar las funciones se declaran dos variables, una para validación y otra que contendrá la etiqueta que hace referencia al Web Component importado previamente.

```
var bool = false; //Variable de Validacion
var joke = '<chuck-norris-fact>' //Variable que contiene el tag

//Declaracion de la Función
function generateJokeTable() {
//Inicializacion de ciclo para otorgar 10 bromas
  for (var i = 0; i <= 10; i++) {
    //Imprecion de Contenido
    $('table').append('<tr><td>' + joke + '</td></tr>')
    console.log(joke)
  }
  bool = true;
}

//Función para Aregar mas bromas al inicio
function addjokes() {
  //Ciclo para agregar 10 bromas mas
  for (var i = 0; i <= 10; i++) {
    //El metodo prepend agrega registros por encima
```

```

        $('table').prepend('<tr><td>' + joke + '</td></tr>')
        console.log('adding' + joke)
    }
}
//Funcion global que involucra a las otras funciones
function getJokes() {
    //Validación de registros
    if (bool === true) {
        addjokes();
    } else {
        generateJokeTable();
    }
}
//Genera Contenido al cargar la ventana.
window.onload = function() {
    generateJokeTable();
}

```

```

<style>
    body {
        font-size: 1.5em;
        line-height: 1.6;
        font-weight: 400;
        font-family: "Raleway", "HelveticaNeue", "Helvetica Neue", Helvetica,
        Arial, sans-serif;
        color: #222;
    }

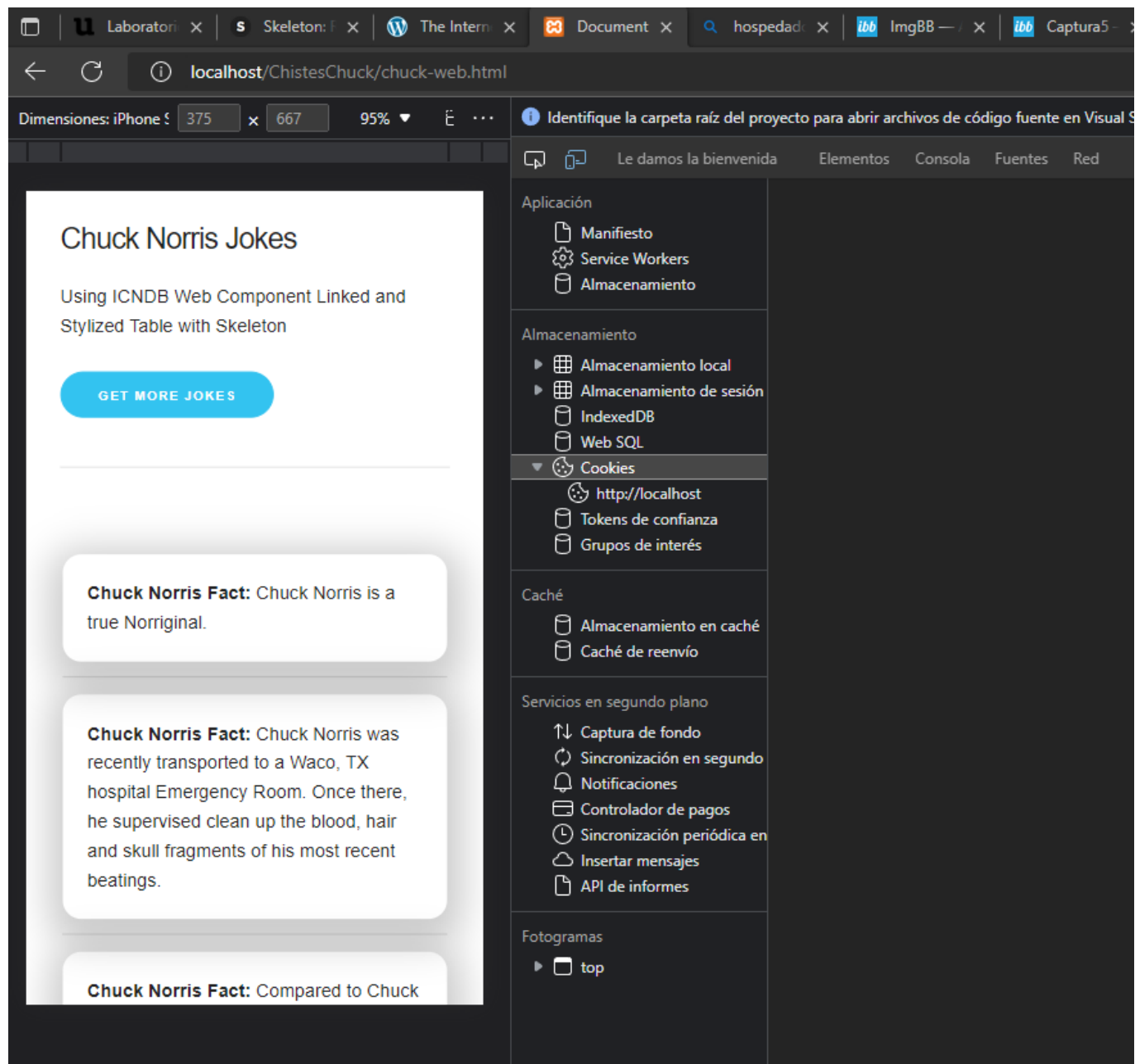
    .button {
        border-radius: 100px;
    }

    .section-heading,
    .section-description {
        margin-top: 1.5em;
        margin-bottom: 1.2em;
    }

```

Cambio de diseño a Skeleton

Para visualizar la petición con Web Components se pueden usar los navegadores con modo emulación de dispositivo.



Responsivo con Skeleton

Despliegue

En caso de no funcionar verificar las dependencias:

- Bootstrap 5
- Node_modules:
 - chuck-norris-fact-element
 - json3
 - node-fetch
 - polyfill
- Skeleton

Construido con

Menciona las herramientas que utilizaste para crear tu proyecto

- [Visual Studio Code](#) - El IDE Usado

Licencia

Este practica fue proporcionada por la Universidad de la Rioja para la materia de Cliente Servidor.

Bibliografia

- **API_Market, B.** (23 de MARZO de 2016). API REST. Obtenido de API REST:
<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- **CANINOS, N.** (06 de SEPTIEMBRE de 2015). bootstrap. Obtenido de bootstrap:
<https://www.negocioscaninos.com/que-es-bootstrap-bootstrap-framework-front-end/>
- **CARBALLO, R. G.** (07 de NOVIEMBRE de 2012). ETIQUETAS. Obtenido de TODO HTML:
<https://sites.google.com/site/dwebhtml/reference>
- **HTML5.** (2013). Que es HTML5 Boilerplate. Obtenido de Que es HTML5 Boilerplate:
<https://www.htmlcinco.com/html5-boilerplate/#comments>
- **JUMBOTRON, C.** (s.f.). jumbotron. Obtenido de jumbotron:
<https://www.tutorialesprogramacionya.com/cssya/bootstrapya/detalleconcepto.php?codigo=149>
- **MDN.** (06 de MAYO de 2019). MDN web docs . Obtenido de Control de acceso HTTP (CORS):
https://developer.mozilla.org/es/docs/Web/HTTP/Access_control_CORS

Conclusiones

VS Code es un potente editor de código usado en desarrollo, tiene incorporado multitud de funcionalidades que facilitan el trabajo del desarrollo Web, es así como se han utilizado en este laboratorio las extensiones HTML 5 Boilerplate y Bootstrap, que trabajan con los estándares usados en la mayoría de las aplicaciones Web.

Se ha desarrollado el código para realizar peticiones a la base de datos ICNDB y se han comprendido las características propias de cada uno de los estándares Web que han surgido desde el año 2005.

 con  por [Luisgr95](#) 