**PETICIONES CON AJAX A UNA BASE DE DATOS MEDIANTE ESTANDARES WEB**

**INTEGRANTES:**

**DUVAN ALEXANDER ANGARITA SANCHEZ**

**JASSON PINCAY PÁRRAGA**

**JULIÁN DAVID MORA RAMOS**

**GABRIEL RICARDO MORENO BOLÍVAR**

**PRESENTADO A:**

**DR. ISMAEL SAGREDO OLIVENZA**

**UNIVERSIDAD INTERNACIONAL DE LA RIOJA**

**MÁSTER EN INGENIERIA DE SOFTWARE Y SISTEMAS INFORMATICOS**

**COMPUTACIÓN EN EL CLIENTE WEB**

**AÑO 2020**

**CONTENIDO**

[TABLA DE ILUSTRACIONES 3](#_Toc39662812)

[OBJETIVOS 1](#_Toc39662813)

[INSTALACIÓN 2](#_Toc39662814)

[Bootstrap: 2](#_Toc39662815)

[HTML Boilerplate: 2](#_Toc39662816)

[DESARROLLO 3](#_Toc39662817)

[RESOLUCIÓN DEL EJERCICIO A LA MANERA DE 2005 4](#_Toc39662818)

[RESOLUCIÓN DEL EJERCICIO A LA MANERA DE 2006 6](#_Toc39662819)

[RESOLUCIÓN CON *PLUGIN* DE jQuery (HASTA 2014) 7](#_Toc39662820)

[RESOLUCIÓN EN 2014 9](#_Toc39662821)

[RESOLUCIÓN DEL EJERCICIO CON WEB COMPONENTS 12](#_Toc39662822)

[Link a URL 12](#_Toc39662823)

[Link al web Component Local 16](#_Toc39662824)

[BIBLIOGRAFIA 18](#_Toc39662825)

[CONCLUSIONES 19](#_Toc39662826)

# **TABLA DE ILUSTRACIONES**

[Ilustración 1. Instalación de Bootstrap en VS Code 2](#_Toc39662832)

[Ilustración 2. Instalación del HTML Boilerplate con VS Code. 3](#_Toc39662833)

[Ilustración 3. Logo de la base de datos ICNDb.com 3](#_Toc39662834)

[Ilustración 4. Petición con el objeto XMLHttpRequest. 4](#_Toc39662835)

[Ilustración 5. Elemento jumbotron para sustituir el texto del chiste. 4](#_Toc39662836)

[Ilustración 6. Estándares de diseño, desarrollo y meta etiqueta de ventana grafica para el uso de Bootstrap. 5](#_Toc39662837)

[Ilustración 7. Página Web con el resultado de la petición a la manera 2005. 5](#_Toc39662838)

[Ilustración 8. Instrucción $.get() del framework jQuery. 6](#_Toc39662839)

[Ilustración 9. Atributos src, integrity y crossorigin para enlazar con la biblioteca jQuery. 6](#_Toc39662840)

[Ilustración 10. Página Web con el resultado de la petición a la manera 2006. 7](#_Toc39662841)

[Ilustración 11. Uso del pluggin jQuery ICNDB para crear una lista de chistes. 8](#_Toc39662842)

[Ilustración 12. Página Web con el resultado de la petición usando el plugin de jQuery. 8](#_Toc39662843)

[Ilustración 13. Clase container para crear añadirle características a la lista. 8](#_Toc39662844)

[Ilustración 14. Uso de fetch. 9](#_Toc39662845)

[Ilustración 15. Lista a aplicar resultado. 10](#_Toc39662846)

[Ilustración 16. Página Web con el resultado de la petición a la manera 2014. 10](#_Toc39662847)

[Ilustración 17 Codigo fetch-node 11](#_Toc39662848)

[Ilustración 18 Ejecución node-fetch 11](#_Toc39662849)

[Ilustración 19. Referencia para importar del Web Component. 12](#_Toc39662850)

[Ilustración 20. Referencia del framework de maquetado Skeleton. 13](#_Toc39662851)

[Ilustración 21. Variables de validación y para contener el tag. 13](#_Toc39662852)

[Ilustración 22. Función para generar el contenido inicial. 13](#_Toc39662853)

[Ilustración 23. Función para agregar contenido después de una primera ejecución. 14](#_Toc39662854)

[Ilustración 24. Función que invocara a las funciones anteriores. 14](#_Toc39662855)

[Ilustración 25. Función para generar el contenido. 14](#_Toc39662856)

[Ilustración 26. Sobreescritura y personalización de los estilos de Skeleton. 15](#_Toc39662857)

[Ilustración 27. Página Web con el resultado de la petición con Web Components. 16](#_Toc39662858)

[Ilustración 28. Referencias hacia el Web Component y el framework Skeleton. 17](#_Toc39662859)

# **OBJETIVOS**

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.

# **INSTALACIÓN**

## **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. En VS Code está disponible una extensión para utilizar este framework, para ello se abre la pestaña extensiones como se muestra en la Ilustración 1. (CANINOS, 2015)

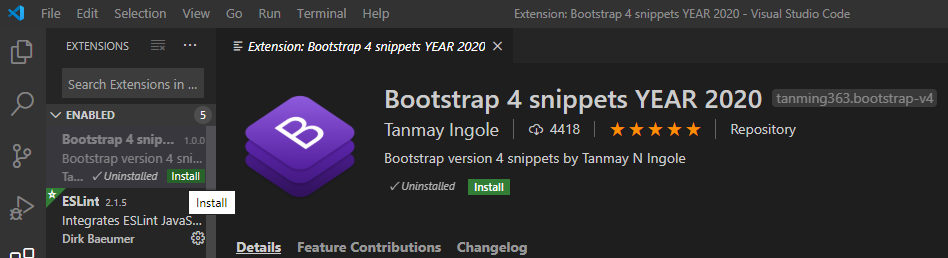


Ilustración 1. Instalación de Bootstrap en VS Code

## **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. (HTML5, 2013)

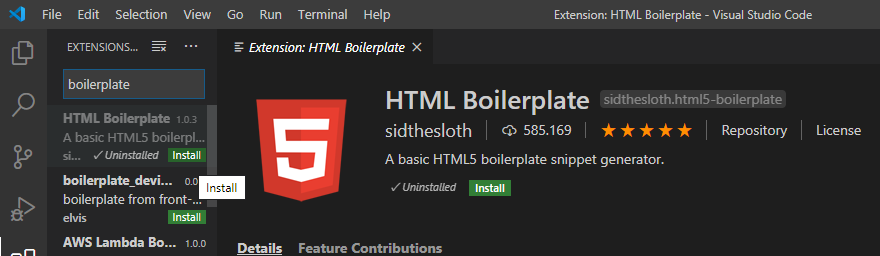


Ilustración 2. Instalación del HTML Boilerplate con VS Code.

# **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. (API\_Market, 2016)



Ilustración 3. Logo de la base de datos ICNDb.com

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.

# **RESOLUCIÓN DEL EJERCICIO A LA MANERA DE 2005**

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



Ilustración 4. Petición con el objeto XMLHttpRequest.

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:

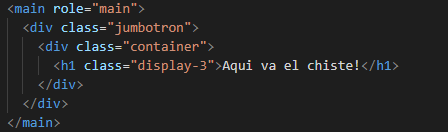


Ilustración 5. Elemento jumbotron para sustituir el texto del chiste.

Como se observa en la Ilustración 5, la etiqueta “<main>” sirve para definir el contenido principal y destacado del DOM. 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. (CARBALLO, 2012)

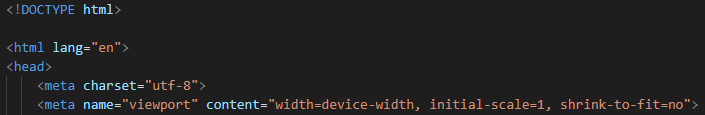


Ilustración 6. Estándares de diseño, desarrollo y meta etiqueta de ventana grafica para el uso de Bootstrap.

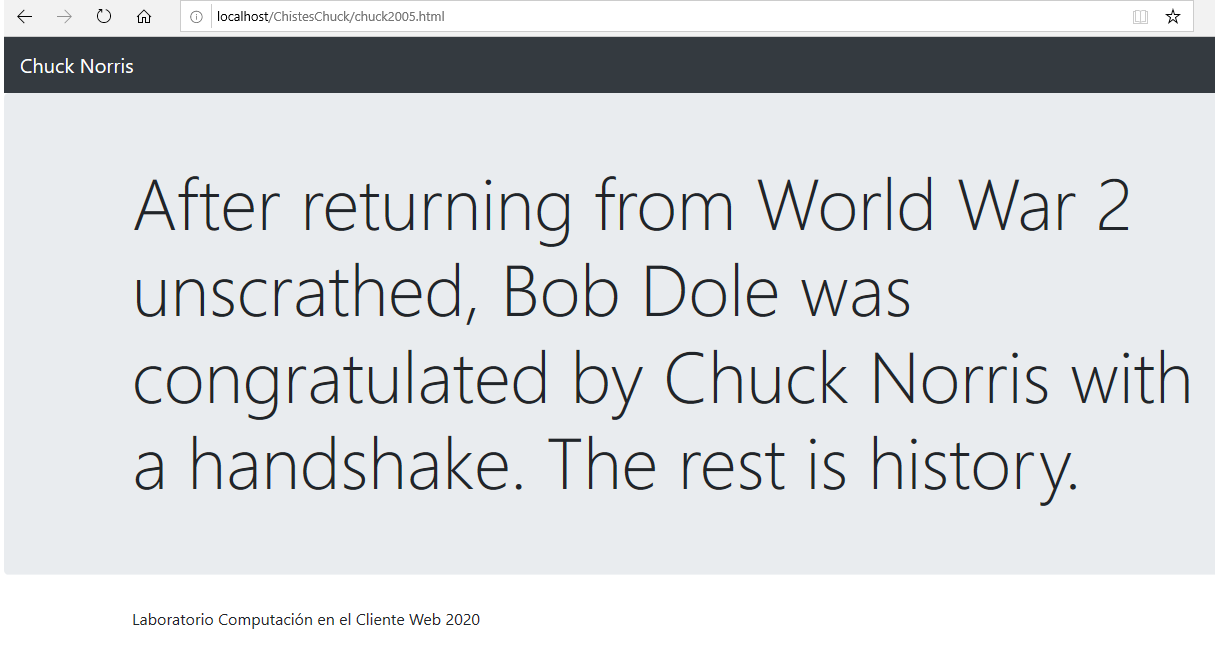
****

Ilustración 7. Página Web con el resultado de la petición a la manera 2005.

# **RESOLUCIÓN DEL EJERCICIO A LA MANERA DE 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 la Ilustración 8 para conseguir un funcionamiento parecido al del ejercicio anterior (Ilustración 4):

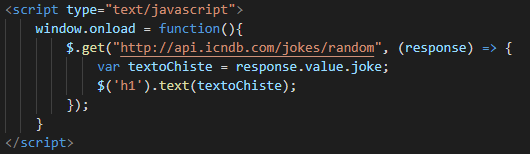


Ilustración 8. Instrucción $.get() del framework jQuery.

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](https://developer.mozilla.org/es/docs/Glossary/CORS)) es un mecanismo que utiliza cabeceras [HTTP](https://developer.mozilla.org/es/docs/Glossary/HTTP) adicionales para permitir que un [user agent](https://developer.mozilla.org/es/docs/Glossary/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. (MDN, 2019)



Ilustración 9. Atributos src, integrity y crossorigin para enlazar con la biblioteca jQuery.

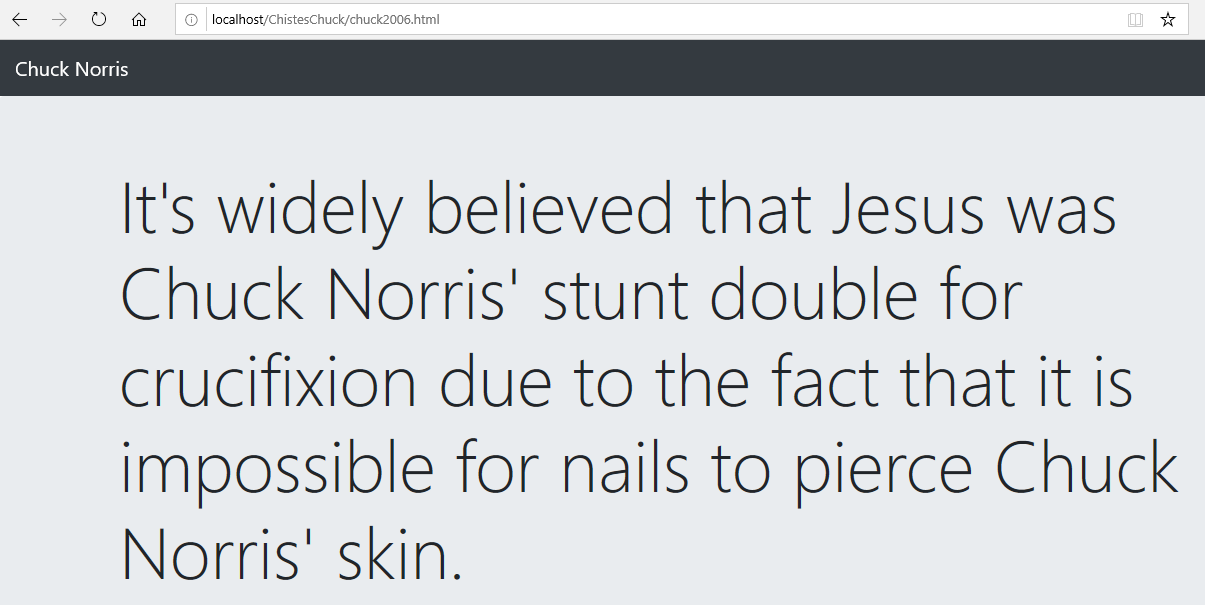


Ilustración 10. Página Web con el resultado de la petición a la manera 2006.

**¿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 *PLUGIN* DE jQuery (HASTA 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 $.icdn.getRandomJokes() se obtiene una lista de chistes (10 en este caso) utilizando la propiedad number de la base de datos gracias a la instrucción <script 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.

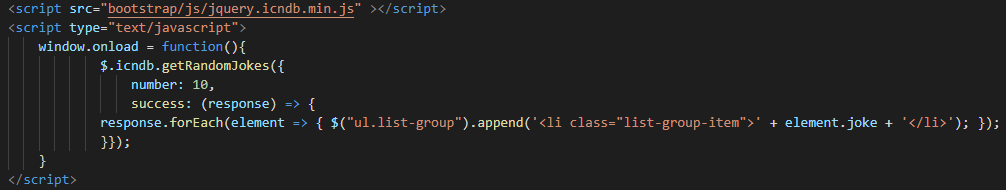


Ilustración 11. Uso del pluggin jQuery ICNDB para crear una lista de chistes.

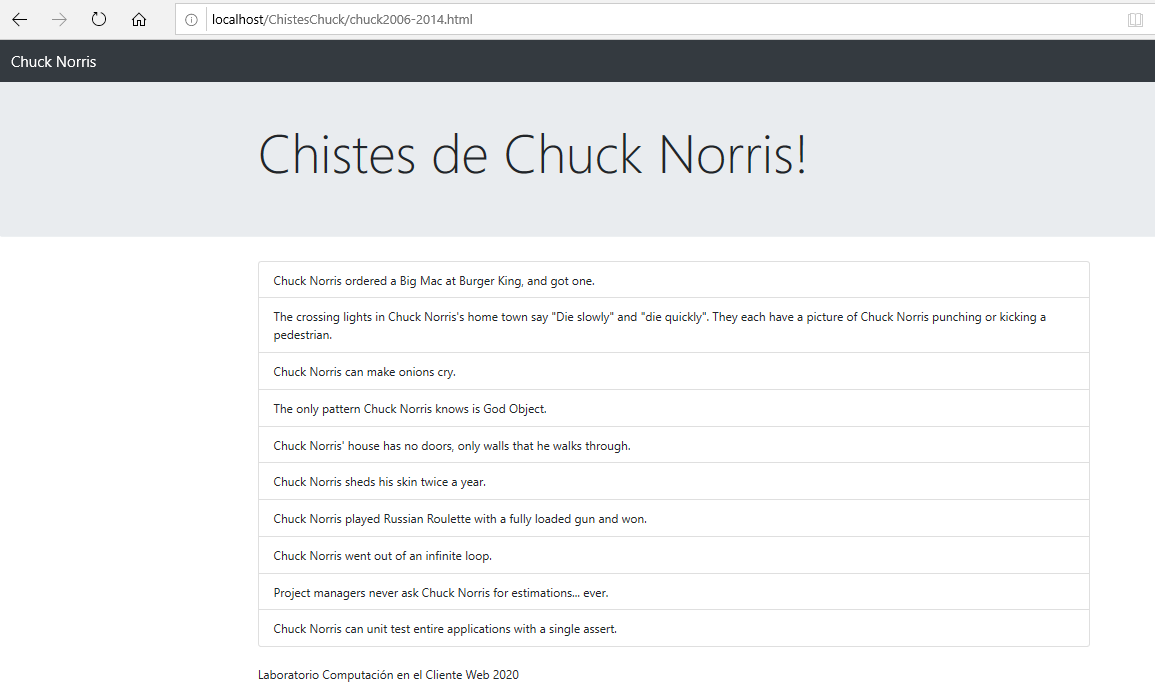


Ilustración 12. Página Web con el resultado de la petición usando el plugin de jQuery.

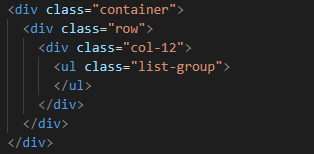


Ilustración 13. Clase container para crear añadirle características a la lista.

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 EN 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.

Captura de pantalla de un celular con letras

Descripción generada automáticamente

Ilustración 14. Uso de fetch.

Imagen que contiene reloj, negro, medidor, cuarto

Descripción generada automáticamente

Ilustración 15. Lista a aplicar resultado.

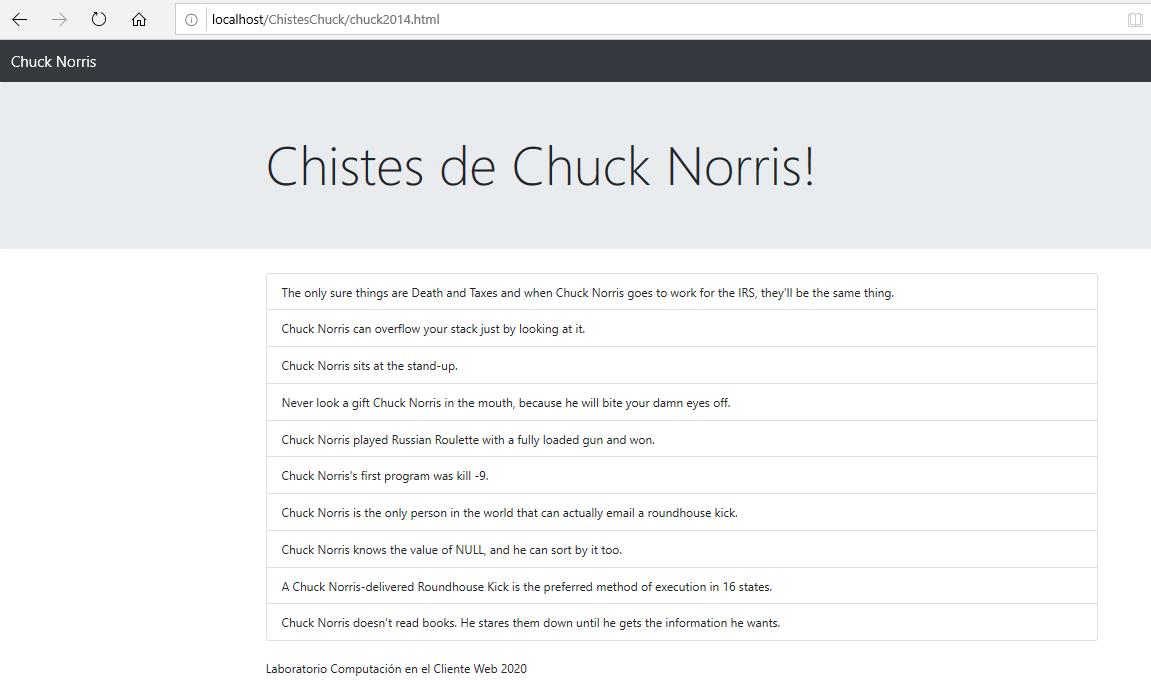


Ilustración 16. Página Web con el resultado de la petición a la manera 2014.

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:

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

Ilustración 17 Codigo fetch-node

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 chuck-node.js

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 18 Ejecución node-fetch

Podemos evidenciar como se nos muestra en consola el resultado de la petición.

**¿Qué es el WHATWG (**[**https://whatwg.org/**](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.

# **RESOLUCIÓN DEL EJERCICIO CON WEB COMPONENTS**

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-joke> y poderlo utilizar dentro del DOM:

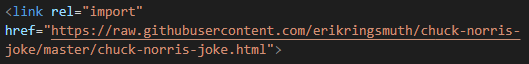


Ilustración 19. Referencia para importar del Web Component.

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



Ilustración 20. Referencia del framework de maquetado Skeleton.

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-joke>. 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.

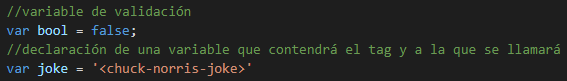


Ilustración 21. Variables de validación y para contener el tag.

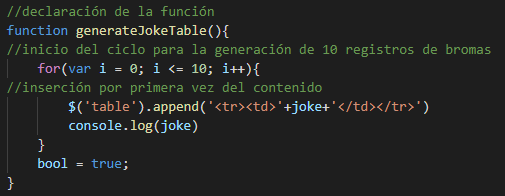


Ilustración 22. Función para generar el contenido inicial.

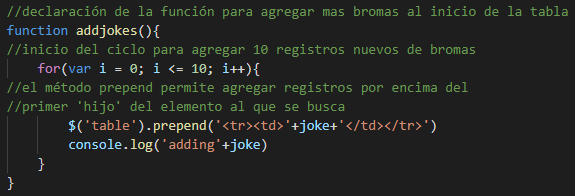


Ilustración 23. Función para agregar contenido después de una primera ejecución.

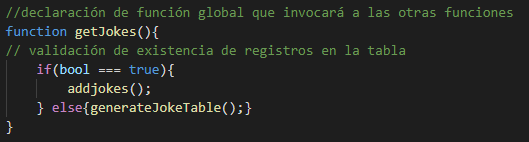


Ilustración 24. Función que invocara a las funciones anteriores.

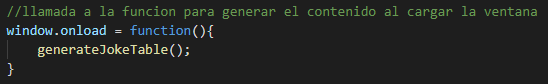


Ilustración 25. Función para generar el contenido.



Ilustración 26. Sobreescritura y personalización de los estilos de Skeleton.

Para visualizar la petición con Web Components se pueden usar los navegadores Opera, Blisk o Edge en su ultima versión.

Hecho esto, se obtiene el siguiente resultado:

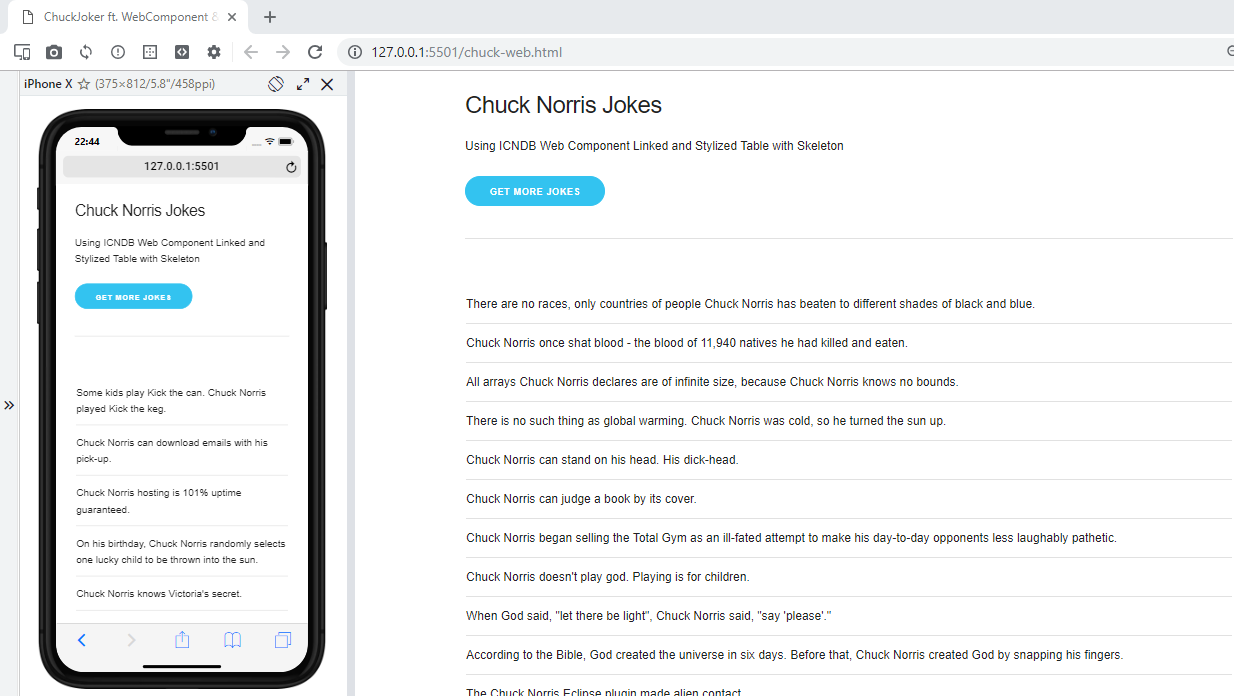


Ilustración 27. Página Web con el resultado de la petición con Web Components.

## **Link al web Component Local**

Bajo este método es necesario instalar el componente de manera local con la siguiente línea

**bower install chuck-norris-joke --save**

Esto genera un archivo: bower.json, en el que se almacenan todas las dependencias del proyecto para poder actualizarlas cuando sea necesario. Así mismo, genera una estructura de directorio:

**+bower\_components**

**L chuck-norris-jokes**

**L .bower.json (dependencias para el gestor de paquetes p.e. npm)**

**L bower.json (dependencias para bower)**

**L chuck-norris-joke.html (código del Tag requerido)**

**L demo.html**

**L LICENSE.md**

**L README.md**

Seguido de esto, basta con hacer la referencia tal cual el método anterior, pero hacia el directorio local en el atributo **[href].**

De la misma manera se puede instalar el framework Skeleton con la siguiente línea

**bower install skeleton-framework --save**

Finalmente, las referencias hacia el Web Component y el framework Skeleton se escriben así

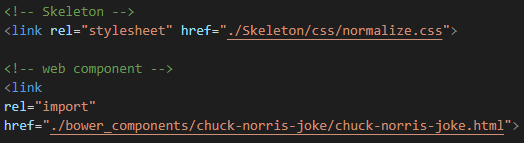


Ilustración 28. Referencias hacia el Web Component y el framework Skeleton.

# **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.