



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**ANÁLISIS DE CÓDIGO ESTÁTICO UTILIZANDO LA HERRAMIENTA GOOGLE CLOSURE  
PARA PROYECTOS QUE UTILIZAN EL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT EN  
EL SISTEMA DE INFORMACIÓN GERENCIAL DE SALUD DEL MINISTERIO DE SALUD  
PÚBLICA Y ASISTENCIA SOCIAL**

**César Humberto Carbonell Ordoñez**

Asesorado por el Ing. Esduardo Efraín Avelar Flores

Guatemala, noviembre de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ANÁLISIS DE CÓDIGO ESTÁTICO UTILIZANDO LA HERRAMIENTA GOOGLE CLOSURE  
PARA PROYECTOS QUE UTILIZAN EL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT EN  
EL SISTEMA DE INFORMACIÓN GERENCIAL DE SALUD DEL MINISTERIO DE SALUD  
PÚBLICA Y ASISTENCIA SOCIAL**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**CÉSAR HUMBERTO CARBONELL ORDOÑEZ**  
ASESORADO POR EL ING. ESDUARDO EFRAÍN AVELAR FLORES

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, NOVIEMBRE DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Miguel Ángel Cancinos Rendon
EXAMINADOR	Ing. Miguel Marín de León
SECRETARIA	Inga. Lesbia Magalí Herrera López

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**ANÁLISIS DE CÓDIGO ESTÁTICO UTILIZANDO LA HERRAMIENTA GOOGLE CLOSURE  
PARA PROYECTOS QUE UTILIZAN EL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT EN  
EL SISTEMA DE INFORMACIÓN GERENCIAL DE SALUD DEL MINISTERIO DE SALUD  
PÚBLICA Y ASISTENCIA SOCIAL**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha septiembre de 2016.

**César Humberto Carbonell Ordoñez**

Guatemala, 10 de Febrero del 2017

Ingeniero Carlos Azurdia  
Revisor de Trabajos de Graduación  
Escuela de Ciencias y Sistemas  
Facultad de Ingeniería, USAC.

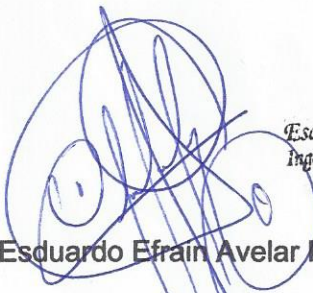
Respetable Ingeniero Azurdia.

Por medio de la presente hago de su conocimiento que como asesor del trabajo de graduación "ANÁLISIS DE CÓDIGO ESTÁTICO UTILIZANDO LA HERRAMIENTA GOOGLE CLOSURE PARA PROYECTOS QUE UTILIZAN EL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT EN EL SISTEMA DE INFORMACIÓN GERENCIAL DE SALUD DEL MINISTERIO DE SALUD PÚBLICA Y ASISTENCIA SOCIAL." desarrollado por el estudiante "CESAR HUMBERTO CARBONELL ORDOÑEZ", y luego de haberlo revisado lo estoy aprobando considerando que el mismo cumple con los objetivos planteados al inicio del trabajo.

Aprovecho para informarle que como asesor me estoy haciendo corresponsable del contenido del trabajo de graduación.

Sin otro particular,

Atentamente,

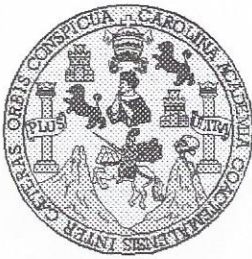


*Esduardo Efraim Avelar Flores  
Ingeniero en Sistemas de Información  
y Ciencias de la Computación  
Colegiado No. 13008*

Ing. Esduardo Efraim Avelar Flores

ASESOR

No. Colegiado 13008



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 01 de Marzo de 2017

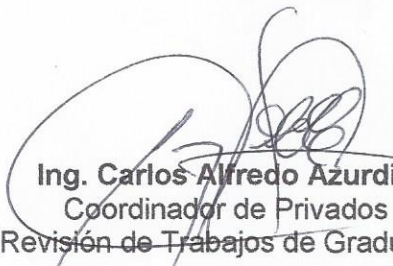
Ingeniero  
**Marlon Antonio Pérez Türk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **CÉSAR HUMBERTO CARBONELL ORDOÑEZ** con carné **200819179** y CUI **2296 18626 0101**, titulado "ANÁLISIS DE CÓDIGO ESTÁTICO UTILIZANDO LA HERRAMIENTA GOOGLE CLOSURE PARA PROYECTOS QUE UTILIZAN EL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT EN EL SISTEMA DE INFORMACIÓN GERENCIAL DE SALUD DEL MINISTERIO DE SALUD PÚBLICA Y ASISTENCIA SOCIAL" y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“ANÁLISIS DE CÓDIGO ESTÁTICO UTILIZANDO LA HERRAMIENTA GOOGLE CLOSURE PARA PROYECTOS QUE UTILIZAN EL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT EN EL SISTEMA DE INFORMACIÓN GERENCIAL DE SALUD DEL MINISTERIO DE SALUD PÚBLICA Y ASISTENCIA SOCIAL”**, realizado por el estudiante **CÉSAR HUMBERTO CARBONELL ORDOÑEZ**, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑADA A TODOS”**

  
Ing. Ma Non Antonio Pérez Türk  
Director



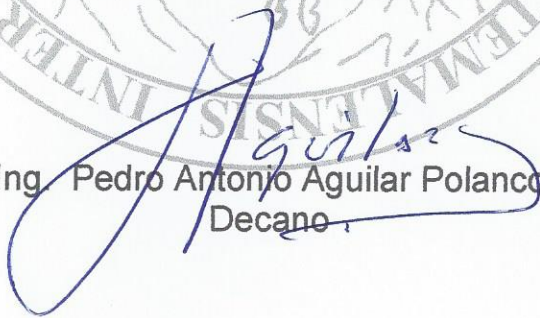
**Escuela de Ingeniería en Ciencias y Sistemas**

Guatemala, 09 de noviembre de 2017



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ANÁLISIS DE CÓDIGO ESTÁTICO UTILIZANDO LA HERRAMIENTA GOOGLE CLOSURE PARA PROYECTOS QUE UTILIZAN EL LENGUAJE DE PROGRAMACIÓN JAVASCRIPT EN EL SISTEMA DE INFORMACIÓN GERENCIAL DE SALUD DEL MINISTERIO DE SALUD PÚBLICA Y ASISTENCIA SOCIAL**, presentado por el estudiante universitario: **César Humberto Carbonell Ordoñez**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

  
Ing. Pedro Antonio Aguilar Polanco  
Decano

Guatemala, noviembre de 2017



/cc



## **ACTO QUE DEDICO A:**

- Dios** Por la vida, por brindarme las oportunidades y darme soporte en los momentos de necesidad.
- Mi madre** Rosa Maria Carbonell, por el apoyo incondicional, el amor y el soporte a pesar de todos los obstáculos.
- Mi hermano** Jose Alejandro Cabonell Ordoñez, por la ayuda y el apoyo ante las distintas adversidades; por ser un hermano, amigo y consejero siempre que se necesitó.
- Mi abuelo** Damian Carbonell Preti, por forjar mis principios, virtudes y valores; siempre buscando que sacara lo mejor de mi ante cualquier situación además por su confianza total por mi trayecto.
- Mi abuela** Rosalina Maribel Ordoñez de Carbonell, por apoyar todo mi trayecto profesional hasta el final y respetar mis decisiones.

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos se Guatemala**

Por la oportunidad de los estudios y la enseñanza de los principios éticos y profesionales.

**Facultad de Ingeniería**

Por brindarme la posibilidad de aprender de profesionales con excelencia y encaminar de la mejor manera mi carrera profesional.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN .....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN .....	XIX
1. ANÁLISIS DEL CÓDIGO DE PROGRAMACIÓN .....	1
1.1. Definición de análisis de código de programación.....	2
1.2. Tipos de análisis de código .....	4
1.2.1. Análisis de código estático .....	6
1.2.2. Análisis de código dinámico.....	7
1.3. Ventajas y desventajas de cada tipo de análisis .....	8
1.3.1. Definición de lenguaje de programación.....	11
1.3.2. Clases de lenguajes programación.....	13
1.3.3. Lenguaje de programación JavaScript .....	17
1.3.4. Función del lenguaje de programación JavaScript.....	20
1.4. Desarrollador Google .....	23
1.5. Desarrollador de herramientas Google Chrome .....	28
1.6. Compilador .....	33
2. HERRAMIENTAS DE ANÁLISIS DE CÓDIGO Y SU FUNCIONALIDAD .....	39
2.1. Herramientas de análisis de código.....	40

2.11.	Herramienta de código estático Google Closure .....	40
2.1.2.	Otros lenguajes de programación .....	44
2.1.2.1.	Java.....	45
2.1.2.2.	Visual Basic.....	47
2.1.2.3.	C# (C Sharp) .....	49
2.1.2.4.	PHP .....	51
2.1.2.5.	HTML.....	55
2.1.2.6.	SQL .....	57
2.1.2.7.	Oracle PL SQL .....	60
3.	DESCRIPCIÓN DE ESCENARIO INICIAL.....	65
3.1.	Contexto.....	65
3.2.	Estructura de desarrollo en JavaScript para futuros proyectos en el Ministerio de Salud, departamento de SIGSA .....	68
3.3.	Herramientas de apoyo para el desarrollo del estudio .....	69
3.3.1.	MySQL .....	69
3.3.2.	Oracle.....	70
3.3.3.	Maven.....	74
3.3.4.	IDE Eclipse Neon .....	76
3.4.	Descripción de proyectos para analizar .....	80
3.4.1.	Proyecto, atención al ciudadano .....	80
3.4.2.	Proyecto codificaciones y defunciones.....	80
3.4.3.	Proyecto, cooperación internacional .....	81
3.4.4.	Proyecto, DRACES .....	81
3.4.5.	Proyecto, encuestas servicios atención .....	81
3.4.6.	Proyecto, Epiweb .....	82
3.4.7.	Proyecto, Incan Cáncer .....	82
3.4.8.	Proyecto, InfraUPE.....	82

3.4.9.	Proyecto, mantenimiento red de servicios .....	82
3.4.10.	Proyecto, monitoreo de servicios.....	83
3.4.11.	Proyecto, UNIP .....	83
3.5.	Resultados previos en la utilización de herramienta.....	83
3.5.1.	Cantidad de código.....	83
3.5.2.	Calidad de código a optimizar.....	87
3.6.	Proceso descrito paso a paso del uso de la herramienta Google Closure Compiler .....	92
3.6.1.	Requisitos.....	92
3.6.2.	Proceso de Instalación .....	93
3.6.3.	Guía de uso .....	95
4.	RESULTADOS .....	97
4.1.	4.1. Resultados.....	97
4.1.1.	Mensajes de alerta .....	97
4.1.2.	Cantidad de código en su validación .....	98
4.1.3.	Calidad del código .....	102
4.2.	Comparaciones estadísticas.....	108
4.2.1.	Número de líneas de código .....	108
4.2.2.	Cantidad de código.....	111
4.2.3.	Optimización de código .....	115
4.3.	Conclusiones sobre la utilización de herramienta.....	120
	CONCLUSIONES .....	123
	RECOMENDACIONES .....	125
	BIBLIOGRAFÍA.....	127



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Instrucción mvn-Dskip Test. Administrador.Command Prompt.....	94
2.	Número de líneas código en cooperación internacional.....	108
3.	Número de líneas código en monitoreo de servicios.....	108
4.	Número de líneas código en Epiuario.....	109
5.	Número de líneas código en Epiweb.....	109
6.	Número de líneas de código en Epiweb, usuarios y fronteras .....	110
7.	Número de líneas código en Incan-Cáncer .....	110
8.	Número de líneas código en InfraUPE .....	111
9.	Cantidad de código en cooperación internacional.....	111
10.	Cantidad de código en monitoreo de servicios.....	112
11.	Cantidad de código en Epiuario .....	112
12.	Cantidad de código en Epiweb.....	113
13.	Cantidad de código en Epiweb, usuarios y fronteras .....	113
14.	Cantidad de código en Incan-Cáncer .....	114
15.	Cantidad de código en InfraUPE .....	114
16.	Calidad de código en cooperación internacional.....	115
17.	Calidad de Monitoreo de Servicios.....	116
18.	Calidad de Epiuario .....	116
19.	Calidad de Epiweb (antes) .....	117
20.	Calidad de Epiweb (después) .....	118
21.	Calidad de usuarios y fronteras.....	119
22.	Calidad de código Incan-Cáncer .....	119
23.	Calidad de código InfraUPE .....	120

## TABLAS

I.	Proyecto, Cooperación internacional .....	83
II.	Proyecto, monitoreo servicios .....	84
III.	Proyecto, epidario .....	84
IV.	Proyecto Epiweb .....	84
V.	Proyecto, usuarios .....	86
VI.	Proyecto, fronteras.....	86
VII.	Proyecto INCAN-Cáncer .....	86
VIII.	Proyecto InfraUPE .....	87
IX.	Proyecto, cooperación internacional .....	88
X.	Proyecto monitoreo de servicios .....	88
XI.	Proyecto, Epidario .....	89
XII.	Proyecto, Epiweb .....	89
XIII.	Proyecto, usuarios .....	90
XIV.	Proyecto, fonteras.....	91
XV.	Proyecto INCAN-Cáncer .....	91
XVI.	Proyecto InfraUPE .....	92
XVII.	Cooperación internacional .....	98
XVIII.	Monitoreo de servicios .....	98
XIX.	Epidario .....	99
XX.	Epiweb .....	99
XXI.	Usuarios.....	100
XXII.	Fronteras .....	100
XXIII.	INCAN-Cáncer .....	101
XXIV.	InfraUPE .....	101
XXV.	Cooperación internacional .....	102
XXVI.	Monitoreo servicios .....	102
XXVII.	Epidario .....	103



XXVIII.	Epiweb .....	103
XXIX.	Usuarios .....	105
XXX.	Fronteras.....	106
XXXI.	INCAN-Cáncer .....	106
XXXII.	InfraUPE.....	107



## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>DB</b>	Base de datos
<b>CSS</b>	<i>Cascading style sheets / hoja de estilo</i>
<b>CGI</b>	<i>Common gateway interface</i>
<b>CLR</b>	<i>Common language runtime</i>
<b>C#</b>	<i>C Sharp – C Sostenido-lenguaje</i>
<b>CDA</b>	<i>Coverty dinamic analysis</i>
<b>JDK</b>	<i>Java development kit</i>
<b>JIT</b>	<i>Just in time (technology)</i>
<b>mb</b>	Megabits
<b>MB</b>	Megabytes
<b>NET</b>	<i>Networking, red de Internet</i>



## GLOSARIO

<b>Applets</b>	Programas o componente de una aplicación que corre o se ejecuta, en el contexto de otro programa.
<b>BIT</b>	Unidad mínima de información que puede ser almacenada en una memoria o puede transmitirse, únicamente, presenta dos estados lógicos: cero o uno.
<b>Bytecodes</b>	Un código intermedio más abstracto que el código máquina. Habitualmente, es tratado como un archivo binario con un programa ejecutable similar a un módulo objeto, que es un archivo binario producido por el compilador cuyo contenido es el código objeto o código máquina.
<b>CGI</b>	<i>Common gateway interface</i> , forma de programación de web dinámica.
<b>Compilador</b>	Un compilador es un programa informático que se encarga de traducir el código fuente de una aplicación en desarrollo; es decir, convierte un programa hecho en lenguaje de programación de alto nivel a un lenguaje de máquina, el cual es conocido como de bajo nivel, de tal forma que sea

más entendible y mucho más fácil de procesar en el equipo en el que se está ejecutando.

## **CSS**

Conocido como hoja de estilo, es un lenguaje utilizado en la presentación de documentos HTML. Sirve para organizar la presentación y de aspecto de una página web. Es principalmente utilizado por navegadores web de internet y programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc. No es un lenguaje de programación.

## **Minificar**

Proceso a través del cual se acelera una descarga, el análisis y el tiempo de ejecución; en los lenguajes JavaScript y CSS, se reduce lo más posible el tamaño del archivo, al modificar el nombre de las variables, una vez que el código HTML, se encuentre actualizado de forma correcta, en función de garantizar que los selectores sigan funcionando.

## **NET**

Dominio de primer nivel, nivel superior, es conocido como Networking. Es el sistema de nombres de dominio de Internet que agrupa a todos los dominios inscritos en la red.

## **net / .com / .org.**

Sufijo o identificador de las llamadas extensiones de los dominios de Internet.

<b>Script</b>	Trabaja para el servidor, tiene la función de llevar a cabo la creación la página Web que se envía al navegador; por lo tanto, mostrará la página, seguidamente ejecutará las secuencias de los comandos del lado del cliente como JavaScript.
<b>Etiquetas o tags</b>	Conjunto de palabras claves que se asocian a una entrada, post o contenido que se lleve a cabo; ayudan a clasificar los temas de un blog ya que contienen la información requerida para identificar el tema que se trabaja y son presentadas siempre dentro de un espacio denominado campo.
<b>Tecnología JIT</b>	Tecnología <i>just in time</i> (justo a tiempo), se refiere a un sistema de producción al mínimo de unidades en las menores cantidades posibles y en el último momento posible. Es específicamente un subsistema de procesamiento de la información. Busca producir una mayor capacidad en cuanto a procesos alternativos.
<b>Traductor</b>	Es el programa informático que toma como entrada un texto escrito y da como salida otro texto en un lenguaje llamado objeto.





## RESUMEN

La presente investigación consistirá en la realización del análisis de código estático de los sistemas desplegados en el departamento de desarrollo del Ministerio de Salud Pública y Asistencia Social, departamento, con el lenguaje de programación de JavaScript, con el uso de la herramienta Google Closure para su posible optimización.

Esta optimización obedece a la falta de una metodología definida al momento de la codificación en el desarrollo de proyectos que manejan lenguaje JavaScript; por lo tanto, se hace necesaria la optimización de procesos en función de aumentar su velocidad mediante la definición de una plantilla práctica y funcional para ser utilizada durante las pruebas aplicadas a los diferentes sistemas.

Lo que se pretende es observar y verificar la factibilidad de la optimización del código de programación y si representa un proceso que brinde grandes beneficios a los sistemas de programación de la institución; entre dichas ventajas se pueden mencionar: menor cantidad de código y mejora de la eficiencia de desarrollo. Para demostrar tal procedimiento, se implementará la herramienta Google Closure, que tendrá, fundamentalmente, la función de compactar el código: eliminar el inservible y reescribir de la manera adecuada y correcta para optimizarlo.

Dicha herramienta será utilizada en diversos proyectos creados en el Ministerio de Salud; luego, se analizarán los cambios que realizó la herramienta

en referencia, para comprobar su utilidad y funcionalidad práctica en la optimización realizada por dicho recurso.

El presente estudio de investigación aplicará una metodología experimental cualitativa con la realización de pruebas y la comparación de resultados que puedan definir la factibilidad de la optimización del código referido, con el uso de la herramienta Google Closure; se implementará este proceso en los proyectos que se utilicen en el lenguaje de programación JavaScript para su análisis; se utilizará Internet y el equipo de computación disponible y adecuado para llevar a cabo las pruebas referidas.

# OBJETIVOS

## General

Demostrar que a través de la utilización de la herramienta Google Closure es factible lograr la optimización del código en el desarrollo de determinados proyectos.

## Específicos

1. Establecer una plantilla para ser utilizada de forma práctica y funcional en el desarrollo mejorado de nuevos proyectos.
2. Mejorar de manera significativa el rendimiento de los sistemas al optimizar su código.
3. Optimizar el código actual de los proyectos creados con anterioridad.



## INTRODUCCIÓN

El mundo de la tecnología ha evolucionado; en la actualidad, sus formas, representaciones, ordenadores, aplicaciones y dispositivos y casuísticas necesitan de la realización de verificaciones periódicas y de análisis sobre la funcionalidad de las actividades complejas, rendimiento y la seguridad de los sistemas y el software.

Durante el desarrollo de software, de forma frecuente, no se tiene un estándar para estructurar el código de un proyecto. En estos casos, la entidad encargada no define una metodología específica para el desarrollo. Esta falta de definición no es funcional. Además, cuando se desarrollan diferentes proyectos y no se tiene planteado la metodología de desarrollo, se tiene como resultado la creación de múltiples estructuras para los diferentes proyectos. Esto deriva en ciertas situaciones como el despliegue de gran cantidad de tiempo y esfuerzo en un proyecto desarrollado previamente por otra persona o en sistemas sin un código óptimo que muestren una latencia más alta de lo necesario, la presencia de código innecesario en el proyecto.

La optimización del código de programación brinda grandes beneficios: menor cantidad de código y mejoramiento de la eficiencia del desarrollo. Para demostrar dicho proceso, se utilizará la herramienta Google Closure, que se encargará de compactar el código de lenguaje de JavaScript, quitando el código innecesario y reescribiéndolo de la manera correcta para optimizarlo. Esta herramienta será utilizada en diversos proyectos creados en el Ministerio de Salud de Guatemala y luego se analizarán los cambios llevados a cabo por la herramienta, en la búsqueda de las mejoras realizadas.

Además, la herramienta Google Closure presenta su posible configuración para definir una estructura que podría ser utilizada para la estandarización y optimización de código antiguo y actual.

Por tal razón, se busca implementar la utilización de la referida herramienta con el propósito de definir y desarrollar una plantilla específica sobre las pruebas, con la intención de que dicha plantilla sea eficaz y funcional, así como de uso práctico, sencillo y efectivo para el usuario.

## **1. ANÁLISIS DEL CÓDIGO DE PROGRAMACIÓN**

En la actualidad, uno de los propósitos primordiales de la ingeniería en sistemas es la investigación, el desarrollo e implementación de herramientas tecnológicas, efectivas, factibles, prácticas y seguras. Se aplica en los diversos procesos, sistemas y software creados para la satisfacción de necesidades diversas de la sociedad.

El término análisis de código, dentro de la programación, se enfoca y refiere a la observación y exploración orientada a la búsqueda de errores e inconsistencias en sus procesos, los cuales, al surgir, puede afectar en mayor o menor grado de impacto, el desempeño de la aplicación desarrollada.

Dicho proceso consiste en la realización de pruebas constantes y periódicas a la aplicación para buscar un fallo, inconsistencia o error para, luego, modificar y solucionar el problema encontrado.

El análisis de código busca indagar acerca de situaciones y esquemas que se quieran o necesiten mejorar en lo que respecta al código, con base en reglas y con el uso de herramientas adecuadas, las cuales se aplican en ciertas estructuras, que consideran y sugieren posibles mejoras en el sistema; además, le otorgan facilidad en el mantenimiento y en el mejor desarrollo de proyectos diversos.

El análisis de código de programación es un recurso que ayuda específicamente a la detección de situaciones problema; a su vez, ofrece alternativas efectivas de solución.

El análisis de código debe realizarse con cierta periodicidad, esencialmente, al momento de diseñar una funcionalidad nueva en el software, con el propósito de que, al implementarla, el desarrollo de la nueva función se dé con facilidad y naturalidad.

Asimismo, se despliegan análisis de código con el objetivo de modificar las funciones que ya se tienen previamente programadas en el sistema. Por lo tanto, estos análisis se deben programar conforme las necesidades que se vayan presentando, así lo requieran.

Además, el análisis de código es un recurso de programación muy útil para los lenguajes dado que a través de estos es posible realizar innovaciones, transformaciones y mejorar, en general, el software y los sistemas.

### **1.1. Definición de análisis de código de programación**

El análisis de código de programación hace referencia de manera concreta, al estudio y observación dirigida a la indagación y búsqueda de situaciones visibles y que se representan en errores, así como, en inconsistencias que afectan y pueden hasta modificar el desempeño y propósito funcional de la aplicación desarrollada.

El proceso de análisis consiste específicamente en la realización de pruebas, intentos, ensayos y monitoreo sobre la aplicación respectiva, de forma persistente y habitual, con el fin de encontrar el fallo en su proceso para posteriormente transformar dicha falla, dándole la solución adecuada, mejorando su desarrollo y volviendo más eficiente su función.



“Una técnica aplicada de forma directa sobre el código fuente, sin mutación ni cambio previo, fundamentada en el código fuente, haciendo posible adquirir información que permita mejorar la base de código, conservando el significado inicial del mismo.”<sup>1</sup>

Dicho análisis se realiza con el propósito de encontrar las fallas y situaciones erráticas que se observan:

- Rendimiento reducido
- Errores en el software
- Complicación en el flujo de datos
- Problemas en la seguridad del sistema

Dichas fallas pueden localizarse al momento de ser detectadas mediante el uso de la herramienta adecuada la cual sirve al sistema al aportar la solución apropiada, en función de un mejor desarrollo en el mejoramiento de la base del código.

El análisis del código es, por lo tanto, un medio a través del cual se busca llevar a cabo mejoras en los procesos de desarrollo; es considerado, entonces, como una herramienta de soporte.

Este análisis de código facilita, con una mínima inversión de tiempo, la ubicación de los defectos, lo cual da acceso a un nivel alto de retorno de inversión y resulta ser un recurso de apoyo a la calidad con el uso de una metodología apropiada; aunado, da acceso a la validación de reglas metodológicas aplicadas en los proyectos.

---

<sup>1</sup> EXPÓSITO, Richard. *Análisis estático del código. Técnica de análisis. Proyecto, documentos y artículos.* p. 18.

“El análisis de código puede realizarse desde distintas perspectivas, siendo las de mayor relevancia, las siguientes:

- Calidad, con el fin de:
  - Asegurar el cumplimiento de los estándares del proyecto.
  - Asegurar el cumplimiento de los estándares de desarrollo, que permiten certificar el correcto cumplimiento de las métricas del proyecto.
  - Identificar áreas problemáticas.
  - Identificar problemas de gestión.
  
- Capacidad, en función de:
  - Dimensionamiento de sistemas
  - Rendimiento de sistemas bajo desarrollo
  - Rendimiento de capacidades humanas
  - Código de terceros
  
- Seguridad, dado que:
  - Es el método más adecuado de encontrar los defectos de seguridad, siendo el primer beneficio del análisis del código.
  - Permite identificar necesidades de análisis detallados, justificando las inversiones.
  - Debería realizarse como auditoria, de forma periódica.
  - Permite validar la adhesión a los estándares.”<sup>2</sup>

De lo anterior descrito, se puede mencionar que, para realizar el análisis de código, de manera adecuada y funcional, los aspectos de calidad, capacidad y seguridad son fundamentales de tomar en consideración para cumplir con las demandas que requiere un proyecto.

## **1.2. Tipos de análisis de código**

Indica, además, que el análisis de código ostenta una tipología con distintas características y desarrolla diversas funciones. La clasificación del análisis de código y la siguiente:

---

<sup>2</sup> CRUZ PÉREZ, Roberto. *Análisis de código. Asegurar la calidad del código, un primer paso hacia la mejora de la calidad global del software.* p. 19.

- Análisis de código estático: consiste en el proceso de realizar la evaluación del software sin ejecutarlo; se implementa de forma sencilla, sobre el código fuente como tal. Se realiza sin ejecución de los programas, contrario al análisis dinámico. Contiene, además, otros análisis dentro de su configuración:
  - Análisis de código automático: realiza un programa de ordenador sobre el código. Minimiza el proceso de complejidad que debe localizar problemas en la base del código; los detecta con las reglas definidas previamente.
  - Análisis de código manual: lo realiza un individuo y se concentra en espacios propios de la aplicación seleccionada en función de determinar si, quien utiliza el programa, lo utiliza adecuadamente o si el diseño del software elegido es el correcto.

Estos análisis automático y manual son de características complementarias; el primero se concentra en aspectos de nivel más bajo como la sintaxis y la semántica, es funcional en cualquiera clase de aplicación y en lo que respecta al análisis manual, se centra en aspectos de nivel más alto en lo que se refiere a la estructura de la aplicación o a su forma de trabajar en cuanto a elementos de tipo externo.

Por lo tanto, la unificación de estos análisis de código referidos permite la identificación de problemas potenciales en los distintos niveles del sistema, lo cual da acceso a realizar mejoras en la base del código fuente que incide directamente en el mejoramiento del desarrollo y del mantenimiento del software, inclusive antes de proceder a ejecutarlo.

- Análisis de código dinámico: es de mayor complejidad y cuyo proceso de realización de prueba se lleva a cabo en tiempo real.

Dada su complejidad, este análisis se lleva a cabo en las aplicaciones a través de programas en un procesador real o virtual. Para que este procedimiento sea real y eficazmente efectivo, es demandante que el mismo, sea ejecutado con un grupo de entradas de pruebas suficientes que puedan producir un comportamiento de interés. Este análisis permite obtener métricas entre las cuales se pueden mencionar las siguientes:

- cobertura de código
- niveles de ejecución

### **1.2.1. Análisis de código estático**

El análisis de código estático es de índole más sencillo por su naturaleza estática dado que lleva a cabo su indagación analítica sin necesidad de ejecutar el código de la aplicación; realiza su proceso sencillamente analizando la estructura inicial con la cual fue creado el código fuente. Por lo cual, la búsqueda de errores en esta fase es mucho más práctica y factible.

Dicho de otra manera, el análisis de código estático realiza su procedimiento cuando recibe el código fuente del programa; luego, lo procesa para encontrar y descifrar lo que se le requiere que lleve a cabo; después emite o formula sugerencias mediante las cuales es posible realizar mejoras en el código salvaguarda la semántica de su origen en función de que este sea más eficaz en su facilidad de manejo y más efectivo en su desarrollo.

En lo que respecta a este código, existe diversidad de herramientas que pueden realizar este tipo de análisis, las cuales generalmente se enfocan en uno o varios lenguajes de programación.

“Análisis de código estático: utiliza determinadas herramientas con el propósito de desarrollar sus funciones; entre las cuales es importante mencionar las que se indican a continuación:

- Data Flow
- Metrics
- Google Closure<sup>3</sup>

### **1.2.2. Análisis de código dinámico**

El análisis de código dinámico es, por su naturaleza, el de mayor complejidad entre los dos tipos de código (estático y dinámico); por su configuración, consiste en el análisis de código, ejecutando las pruebas en tiempo real.

Partiendo del análisis dinámico y su conjunto de entradas y/o reglas, conforme su función, para satisfacer necesidades establecidas, es posible determinar valores mínimos de cobertura que pueden indicar y ubicar áreas de riesgo en el sistema.

Este análisis, dado su estructura de características que ostentan un grado mayor de complejidad, utiliza las siguientes formas:

---

<sup>3</sup> EXPÓSITO, Richard. *Análisis estático del código. Técnica de análisis. Proyectos, documentos y artículos.* p. 18.

- de comparación de cambios absolutos
- de comparación de estructuras en lo relativo a:
  - estado de origen
  - estado de la aplicación en sus distintas formas

En lo que concierne a las herramientas desarrolladas para la realización de este tipo de análisis, representan básicamente comprobaciones funcionales; en cuanto a su disponibilidad son relativamente más escasas, dentro de las cuales se pueden mencionar las siguientes:

- *Test automation*
- *Test cases interaction and evaluation*
- *Coverity dynamic analysis -CDA- (Java)*

### **1.3. Ventajas y desventajas de cada tipo de análisis**

“El hecho de realizar análisis de código otorga grandes beneficios y ventajas de índole general a los sistemas y software:

- Brindar directrices de protección.
- Mejorar la calidad del código.
- Envían mensajes de advertencia.
- Detección de errores.
- Identificación de errores de programación y diseño.
- Es altamente recomendable realizar estos análisis en el desarrollo de sistemas de cualquier tipo.”<sup>4</sup>

---

<sup>4</sup> LLUNA, Eduardo. *Análisis estático de código en el ciclo de desarrollo de software de seguridad crítica*. p. 38.

En lo relativo al análisis de código estático, es importante mencionar que este presenta también ventajas y desventajas:

- Ventajas
  - Puede realizarse de forma manual y automática.
  - Es una técnica comparativamente sencilla y de fácil automatización.
  - Es una técnica muy recomendable.
  - Permite mejorar la calidad del software desde su origen.
  - Su aplicación y desarrollo es de menor costo.
  - Brinda sugerencias que pueden realizar mejoras en el código.
  - Se aplica directamente sobre el código fuente sin cambios previos de ninguna índole.
  - Su mayor ventaja consiste en ahorrar tiempo para ahorrar dinero.
- Desventajas
  - No garantiza el descubrir todos los problemas o fallas existentes.

- Al ser aplicada esta técnica, en su forma automática en un proyecto de determinado grado de especificaciones, es sumamente necesario tener en cuenta, lo siguiente:
  - Los parámetros del proyecto como el lenguaje de programación y el nivel de seguridad.
  - Las circunstancias en las cuales se encuentra el mismo.
  - Está técnica utilizada en su forma automática, para ser usada en proyectos específicos debe de estar certificada, lo cual implica un aumento considerable en su valor.
  - Es un medio de obtener mejoras y no un fin en sí mismo.
  - Es una técnica de apoyo únicamente, es decir, de carácter complementario.
  - Al momento de aumentar el grado de complejidad de los sistemas, existe la posibilidad de que no sea posible detectar todos los errores.

“Acerca del análisis de código dinámico, que este también presenta determinadas ventajas y desventajas, dentro de las cuales se pueden referir de forma textual, las siguientes:

- Ventajas
  - Funciona como un mecanismo que puede concebir varias versiones de un mismo código, con el fin de sobrepasar los procesos comunes de análisis de los sistemas, tal es el caso de los antivirus.



- Es una técnica, usada para evitar que el código fuente de determinada entidad pueda ser utilizado u obtenido por otra que no esté debidamente autorizada.
- Analiza y compara la complejidad del código para determinar que parte del sistema es innecesariamente complejo.
- Tiene la capacidad de detectar dependencias que no son posibles de encontrar en el análisis estático.
- Puede compilar información de forma temporal.
- Puede identificar y dar prioridad a las vulnerabilidades de seguridad que encuentre.
- Desventajas
  - No posee la capacidad de dar garantía a la cobertura completa del código fuente.
  - Necesita integrarse con algunas herramientas como Java para lograr un análisis dinámico de código más complementario.
  - Necesita ser ejecutado con un conjunto de entradas de prueba suficientes para general un comportamiento eficaz.
  - Tiene la capacidad de dejar ver la presencia de errores o fallas, no así de revelar la ausencia de las mismas.”<sup>5</sup>

### 1.3.1. Definición de lenguaje de programación

Las computadoras, por sus propiedades tecnológicas de disposición compleja, requieren de un lenguaje configurado de determinada forma, con el fin de que el ser humano pueda tener acceso a estas con cierta facilidad y control; de ahí se originan los lenguajes de programación que permiten al programador tener comunicación y vinculación con el microprocesador dicho proceso, el dispositivo lo realiza a través de símbolos y términos vinculados con

---

<sup>5</sup> GRACIA, Luis Miguel. *Análisis de código dinámico. Un poco de Java. Herramientas Java para el análisis de código dinámico.* p. 6.

el problema que se requiere que resuelva; hace uso de los distintos recursos y herramientas que ha creado la ciencia de la ingeniería en sistemas.

En ese contexto de lenguaje de programación, es primordialmente necesario identificar que, en un programa informático, el código fuente hace alusión a un grupo de líneas de texto con los pasos o directrices que debe realizar el ordenador en función de ejecutar un programa, lo cual significa que el funcionamiento de un programa se encuentra completamente escrito en código fuente.

Así mismo, el código fuente de un programa se encuentra escrito por un programador en determinado lenguaje de programación; no obstante, en esta primera fase o estado no es de forma directa ejecutable por el ordenador por lo que requiere de ser traducido a otro lenguaje o código binario; siendo de esta manera, más factible para la máquina el poder interpretarlo. En esta etapa para realizar la traducción en referencia se utilizan los denominados sistemas de traducción: compiladores, ensambladores e intérpretes.

Dicho término de código fuente, además, es utilizado para referirse a otros componentes o dispositivos del software; hacen alusión al código fuente de una página Web escrito en lenguaje de marcado HTML o Javascript, entre otros lenguajes de programación existentes; dicho código es desarrollado por el navegador Web en función de visualizar la página Web. Esta es el área de la ingeniería de software, o área informática, que se encarga del desarrollo de programas y, por lo tanto, de la creación del código fuente correspondiente.

De ahí, que el lenguaje de programación se desarrolla con el propósito de llevar a cabo procesos que manejan las máquinas denominadas ordenadores o computadoras, mediante las cuales se crean programas con el control del

accionar físico y lógico de una máquina. Dicho lenguaje está desarrollado con símbolos y reglas de sintáctica y semántica que especifican la organización estructuras y significado de sus elementos y expresiones.

“Un lenguaje de programación es un conjunto de normas que permiten vincular a cada programa correcto un cálculo que será realizado por un ordenador sin imprecisiones. Por lo tanto, hace referencia a una forma de interpretar el significado de los programas de determinado lenguaje”<sup>6</sup>.

“Un lenguaje de programación es un lenguaje que ha sido planteado para poder describir un conjunto de acciones secuenciales que el equipo que las soporta debe de llevar a cabo. Es, por lo tanto, una forma práctica para que las personas puedan ordenarle a un equipo que desarrolle determinadas instrucciones que este debe de ejecutar”<sup>7</sup>.

El lenguaje de programación también presenta ciertas ventajas:

- Tiene la posibilidad y propiedad de ser más fácil de comprender que un lenguaje de máquina.
- Da acceso a mayor portabilidad, lo cual significa que tiene mayor facilidad de adaptabilidad para ser ejecutado en diversos tipos de equipos y sus funciones.

### **1.3.2. Clases de lenguajes programación**

Dentro de los lenguajes de programación existe variedad y de forma general, se dividen en dos clases:

---

<sup>6</sup> UREÑA ALMAGRO, Carlos. *Concepto de lenguaje de programación*. p. 17.

<sup>7</sup> Benchmark Group, Revista electrónica CCM. *Lenguaje de programación*. p. 7.

Con base en el procesamiento de sus comandos, los lenguajes de programación se clasifican en:

- Lenguaje de programación imperativo

Este lenguaje programa a través de una serie de comandos agrupados en bloques y se componen de órdenes condicionales que admiten al programa regresar a un bloque de comandos si la condición dada ha sido ejecutada. Esta clase de lenguajes de programación fue de los primeros en ser utilizados; aún en la actualidad, varios lenguajes modernos utilizan este principio.

Sin embargo, los lenguajes de programación imperativos estructurados no poseen flexibilidad, dado el proceso de sucesión de instrucciones mediante la cual, estos trabajan.

- Lenguaje de programación funcional

Esta clase de lenguaje también es conocido como lenguaje procedimental, dado que el mismo crea programas a través de funciones; este restablece un estado nuevo de resultado y toma como entrada el resultado de otras funciones; es decir, que utiliza la función de recursividad se refiere a la función que se invoca a sí misma.

En lo que respecta a aspectos de lineamientos generales, los lenguajes de programación se clasifican de la siguiente manera:

- Lenguaje de interpretación o interpretado

Este es un lenguaje de programación de que debe de traducirse con el propósito que el procesador tenga la capacidad de entenderlo.

Un programa que se encuentra escrito en esta clase de lenguaje requiere del soporte de un programa auxiliar, denominado interprete, el cual realiza la traducción de los comandos de los programas conforme sea requerido.

- Lenguaje de compilación o compilado

Un programa que se encuentre escrito en un lenguaje compilado es traducido mediante un programa de soporte denominado compilador, el cual, a su vez, crea un archivo nuevo independiente que no requiere de ningún otro programa para poder ejecutarse a sí mismo; tal archivo es denominado ejecutable.

Este lenguaje, además, de la ventaja de que no requiere de un programa auxiliar para ejecutarse cuando ya ha sido compilado y como únicamente es necesaria una traducción; la ejecución de dicho lenguaje se torna mucho más veloz.

No obstante, este lenguaje no tiene la capacidad de ser tan flexible como lo es un programa escrito en un lenguaje interpretado dado que cada transformación del archivo fuente, que es el archivo entendible para las persona; el archivo a ser compilado necesita de la compilación del programa para poder realizar las modificaciones pertinentes.

Además, un programa compilado tiene la propiedad de que garantiza la seguridad del código fuente. Efectivamente, el lenguaje interpretado, al ser concisamente un lenguaje entendible, permite que cualquier usuario tenga acceso a los secretos de manufacturación de un programa; por lo tanto, tener la posibilidad de copiar su código o inclusive transformarlo, para permitir el peligro de que los derechos de autor sean violentados.

- Lenguaje intermediario

Dentro de las variedades de lenguajes, algunos pertenecen a las clases de los lenguajes mencionados: de interpretación, de compilación (LISP, Cobol, Python, Basic, C++, Fortran, PHP, Perl, Java, entre otros, común y ampliamente utilizados); al programa escrito en dichos lenguajes le es posible, en determinados casos, soportar y tolerar una fase o ciclo de compilación intermediaria en un archivo que se encuentra escrito en un lenguaje no entendible (en tanto, distinto al archivo fuente). Tal es el caso de los Applets Java, que consisten en pequeños programas que con frecuencia se cargan en páginas Web, archivos compilados que únicamente pueden ser ejecutados en un navegador Web (estos son archivos que se configuran con la extensión .class).

Por lo tanto, conforme el enfoque de las definiciones similar de que se propone en las anteriores definiciones referidas, se infiere que un lenguaje de programación se describe esencialmente como un sistema de comunicación bien estructurado, con mucha semejanza con el lenguaje de comunicación de los seres humanos; también, utiliza, dentro de su configuración lingüística, una serie de signos con el uso de palabras, sonidos y/o gestos.

### 1.3.3. Lenguaje de programación JavaScript

Este lenguaje de programación interpretado, generalmente, es ubicado y dirigido al cliente, regularmente, utilizado dentro del espacio de los navegadores web, desarrollado con el propósito de mejorar la interfaz y reflejar efectos mucho más dinámicos en su presentación al usuario.

“El lenguaje JavaScript, es un lenguaje de programación de scripts (secuencia de comandos), fue creado por Eich, Brendan (trabajador de Netscape Communication Corporation, que publicó el Netscape Navigator); este lenguaje fue reconocido en sus inicio como lenguaje Mocha, posteriormente, se le designó como LiveScript y, finalmente en el año 1995, fue denominado como se le conoce actualmente lenguaje Web, JavaScript, este es reconocido como un lenguaje de programación interpretado, que no requiere ser compilado, se origina del lenguaje Java y su utilización se aplica esencialmente en el diseño y creación de páginas Web. Es, además, dentro de su configuración calificado como una composición entre los lenguajes Java (creado por Sun Microsystems) y HTML.”<sup>8</sup>

No obstante, es necesario puntualizar que JavaScript es un lenguaje que difiere mucho del Java, dado que opera de manera muy diferente y es orientado a objetos sin sucesión o legado, caso contrario al Java que si tiene herencia o sucesión.

El lenguaje JavaScript tiene en común con Java buena parte de su sintaxis y estructuración; también el establecimiento de jerarquía de objetos para definir sus propiedades, entre otras funciones. Además, estos comparten elementos comunes con los lenguajes C y C++.

---

<sup>8</sup> La Revista Informática.com. *Lenguaje de programación JavaScript*. p. 1-2.

“Asimismo, JavaScript es un lenguaje reconocido como de disposición a eventos. Es, además, un lenguaje configurado dentro de la página Web para conformar o ser parte del código HTML, sin el cual no puede existir.”<sup>9</sup>.

Para manejar el lenguaje JavaScript, dado que se maneja en un entorno de navegador Web, es necesario tener conocimientos esenciales acerca de codificación HTML y CSS; representa una ventaja para el manejo del lenguaje en referencia. El lenguaje JavaScript posee algunas características propias:

- Es un lenguaje de programación que admite escribir código fuente que es analizado por un ordenador.
- Su código fuente, escrito por un desarrollador, es un conjunto de acciones o instrucciones que admiten dar comandos al ordenador para que pueda operar el programa. Dicho código fuente es el que rige el funcionamiento del programa.
- Da acceso a programar scripts, de ahí se deriva su nombre, lo cual realiza de las siguientes formas:
  - En lenguaje compilado; esto es cuando el código fuente se da a un programa denominado compilador, el cual lee el código fuente y lo transforma en lenguaje que pueda ser interpretado, es decir, entendible:
    - binario, de 0 y 1 (lenguaje compilado conocido)
    - lenguaje C o C++ (lenguaje compilado conocido)

---

<sup>9</sup> MENENDEZ-BARZANALLANA ASECIO, Rafael. *Desarrollo de aplicaciones Web*. p. 46.



- En lenguaje precompilado que se da cuando el código fuente se compila solo parcialmente; generalmente, un código de lectura aún más entendible para el ordenador, pero que aún no es un lenguaje binario; dicho código de carácter intermedio, es configurado para leerle por lo que se denomina máquina virtual, que se encarga de ejecutar el código:
  - Lenguaje C # (lenguaje precompilado)
  - Lenguaje Java (lenguaje precompilado)
  
- En lenguaje interpretado en el cual no existe compilación dado que el código fuente se mantiene sin sufrir transformaciones; en dado caso se requiera ejecutar el código, es necesario proporcionar un intérprete que lea y realice las acciones que se ordenen.
  
- La mayoría de los scripts son interpretados, por lo cual JavaScript es conocido como un lenguaje interpretado; esto significa que necesita contar con un intérprete para desarrollar el código JavaScript y regularmente, el intérprete que usa es incluido en el navegador de internet. Es usual que cada navegador tiene incluido un intérprete JavaScript el cual varía conforme las características del navegador, como en los casos siguientes:
  - En el navegador Internet Explorer, el intérprete se denomina JScript.
  - En el navegador Mozilla Firefox, el intérprete se denomina Spider Monkey.
  - En el navegador Google Chrome, es el motor V8.

- Con el lenguaje JavaScript, no se necesita tener instalada ninguna clase de framework.
  
- El lenguaje de JavaScript es orientado a objetos dado que sus elementos poseen distintas características las cuales a su vez son específicas y poseen, también, diferentes formas de uso. Tales objetos esenciales que ofrece este lenguaje son:
  - Imágenes
  - Fechas
  - Cadenas de caracteres

Por lo tanto, a través del lenguaje JavaScript es factible diseñar objetos propios, en función de lograr un código fuente que sea mucho más sencillo y fácil de entenderse y, a su vez, una manera de programar que sea mucho más lógica.

#### **1.3.4. Función del lenguaje de programación JavaScript**

El código JavaScript puede incluirse en cualquier tipo de página Web o de documento HTML, así como desde la visión del cliente, como servidor en documentos de tipo PHP, Asp, etc. Dicho código se encuentra contenido dentro de las etiquetas de HTML, contextualmente, de la siguiente manera: `<SCRIPT></ SCRIPT>`. El lenguaje JavaScript tiene la función de diferenciar entre mayúsculas y minúsculas, los espacios en blanco los tiene en cuenta.

“En la actualidad, JavaScript se utiliza en internet y se encuentra incluido de forma directa, en la página Web o en un archivo de tipo externo y su función esencial el mejorar una página HTML, añadiendo características de interacción del usuario, así como animación y soporte a la misma, en las cuales se cuentan:

- Mostrar texto
- Ocultar texto
- Crear presentaciones de diapositivas
- Crear burbujas de información<sup>10</sup>

JavaScript, actualmente, es conocido como un lenguaje que, en su función, trabaja del lado del cliente, lo cual significa que los scripts se desarrollan por parte del navegador del usuario o cliente, situación que difiere de los lenguajes de script, diseñados para funcionar del lado del servidor y los cuales se ejecutan por el servidor como se da con los lenguajes como PHP. Este procedimiento es muy importante dado que el objetivo de los scripts, que trabajan del lado del cliente y del lado del servidor, es distinto en lo siguiente:

- Un script que trabaja para el servidor tiene la función de llevar a cabo la creación la página Web, que se envía al navegador, por lo que mostrará la página; seguidamente, ejecutará las secuencias de los comandos del lado del cliente como JavaScript.
- Si JavaScript ha sido diseñado para utilizarse juntamente con HTML, el lenguaje ha evolucionado a otros destinos; JavaScript, es comúnmente usado para crear extensiones para distintos programas, tal es el caso de los scripts codificados en Lua o Python.
- JavaScript, además, puede ser utilizado para construir aplicaciones, Mozilla Firefox es el ejemplo más representativo, ya que la interfaz del navegador se crea con una especie de lenguaje HTML, que es

---

<sup>10</sup> MENENDEZ-BARZANALLANA ASECIO, Rafael. *Desarrollo de aplicaciones Web*. p. 46.

denominado XUL y la intervención de JavaScript, que se usa con la función de animar la interfaz.

- Otros programas que se basan en esta tecnología son: TomTom Home, usado para administrar el navegador GPS TomTom, mediante la PC.

Por lo cual, JavaScript es un lenguaje de programación que ha sido creado con la idea fundamental de crear páginas Web dinámicas, dado que desarrolla las siguientes funciones:

- Incorpora efectos diversos.
- Crear texto que aparece y desaparece.
- Crear animaciones.
- Crear acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.
- Interactuar con usuarios.

Existen otras utilidades de JavaScript, se basan en crear códigos para insertarlos en las páginas Web; dentro de tales utilidades se pueden referir las siguientes por su importancia:

- Crear relojes, contadores e intervalos de tiempo (utilizando los objetos).
- Crear calendarios y agendas.
- Crear tooltips (pequeños recuadros de información adicional sobre un elemento seleccionado, o mensaje de ayuda para el usuario).

- Menú desplegable (de línea horizontal y vertical, así como de niveles jerárquicos distintos).
- Crear galerías de imágenes (LightBox) (diseñadas para mostrar productos y servicios).
- Crear validadores de formularios.
- Crear detectores de navegadores e idiomas.

Además, dentro de los distintos servicios comunes, muy conocidos y utilizados que se encuentran desarrollados con JavaScript, se pueden mencionar, esencialmente y de forma textual, los siguientes:

- Correo (e-mail)
- Chat
- Buscadores de información

#### **1.4. Desarrollador Google**

“El desarrollador Google, representa a una compañía que es subsidiaria a nivel multinacional de la empresa Alphabet Inc. Fue creado por Page, Larry y Brin Serguei quienes lo iniciaron como un proyecto universitario de ciencias de la computación en la Universidad de Stanford en el año de 1996. Actualmente, este desarrollador, se especializa en la promoción de servicios y productos relacionados con diversidad de dispositivos tecnológicos tales como:

- La Internet
- Software
- Dispositivos electrónicos variados
- Otras múltiples tecnologías relacionadas

El producto principal de Google es el motor de búsqueda de contenido en Internet que ostenta la misma denominación y que a su vez brinda otros servicios como:

- Correo electrónico –Gmail-
- Google Maps
- Google Earth
- Google Calendar
- Google Talk
- Google Analytics
- YouTube
- Google Libros
- Google Noticias
- Google Apps
- Google Traductor (traductor de Google)
- Google Académico
- Google Chrome
- La red social Google+, entre otros<sup>11</sup>

Además, es un potente desarrollador, que lidera la creación del sistema operativo que se basa en Linux y Android, el cual se orienta a teléfonos inteligentes, tabletas, televisores y vehículos y en gafas de realidad aumentada, referidas como las Google Glass.

En lo que respecta a su estructura, está conformada por diversos centros de datos que funcionan con la distribución RedHat del sistema operativo GNU/Linux, por su flexibilidad. Las conexiones de Google se realizan mediante los servidores DNS, los cuales traducen la dirección a distintas IP, que dan acceso a la distribución de la carga entre varios clusters, a través del sistema de planificación Round-robin. Los servidores proxy squid de Google, aceptan peticiones y devuelven resultados desde el cache local. Los servidores web coordinan la ejecución de las consultas enviadas por los distintos usuarios y formatean el resultado, usando para ello el lenguaje HTML.

El potencial del desarrollador de Google es tan extenso que cuenta con más de un millón de servidores y centros de datos a nivel mundial y tiene la capacidad de procesar más de 1 000 millones de búsquedas diariamente y su

---

<sup>11</sup> GRIGORIK, Ilya. *Optimizar la eficacia del contenido*. p. 5.

motor de búsqueda, representa el sitio web que tiene mayor cantidad de visitas a nivel mundial.

En Google, los desarrolladores tienen la oportunidad de optimizar cualquier sitio Web mediante el buceo en los detalles en lo que respecta a escritorio y los navegadores que se implementan en las páginas.

En esta plataforma de Google se establece la ruta de seguimiento o el conjunto de pasos que los desarrolladores deben seguir en función de convertir los códigos de los lenguajes HTML, CSS y JavaScript en sitios Web. Es a partir de ahí, que se inicia la exploración y experimentación con las herramientas aptas para mejorar el rendimiento y desarrollar estrategias simples para los desarrolladores, siguiendo las recomendaciones de PageSpeed Insights y otras herramientas de Google Chrome y herramientas de desarrollo y optimización de las aplicaciones, de las funciones y del manejo de la tecnología e informática.

En el desarrollador Google, se encuentran disponibles multiplicidad de recursos que pueden combinarse para facilitar la implementación de funciones, debiendo siempre tener en consideración las reglas establecidas; es muy importante, además, el hecho de que se debe analizar y eliminar descargas innecesarias para la optimización y la codificación de cada recurso, a través de distintas técnicas de compresión y la utilización del almacenamiento en caché, siempre que sea factible prescindir de descargas superfluas.

El desarrollador de Google dispone potente infraestructura para el análisis de datos y facilita el aprendizaje del manejo del mismo como tal. Posee multiplicidad de herramientas y funciones que se caracterizan por lo siguiente:

- Mejora constantemente la tecnología, actualmente basada en la nube.

- Analiza opiniones y entidades con la API NL de Google Cloud.
- Almacena archivos y realiza copia de seguridad para los distintos programas que maneja y desarrolla.
- Es un desarrollador de alto rendimiento y de constante evolución.
- Tiene la disponibilidad de realizar análisis de datos de forma eficaz en lo relativo al Big Data, lo cual lleva a cabo con gran velocidad.
- Pasa del prototipo a la producción, en una escala globalizada, dado su gran capacidad en cuanto a las operaciones de código, lo cual realiza con fiabilidad y gran rendimiento.
- Tiene la capacidad de crear y escalar aplicaciones, en lo que se refiere a diversidad de recursos informáticos, lo cual realiza a partir las máquinas virtuales con un rendimiento muy favorable, desde la plataforma de desarrollo de aplicaciones administrada que posee.
- Su almacenamiento de objetos y bases de datos con escalabilidad, tiene propiedades de mucha resistencia y otorga alto rendimiento en todo lo relativo a las aplicaciones.
- En lo relativo a las redes, estas son definidas a través de software de última generación dado que desarrolla los procesos mediante la red de fibra privada que es una de las propiedades características de Google.
- El almacenamiento de datos de Google Big Data, es totalmente administrado a través del procesamiento por lotes, transmisiones y



exploración de la información Hadoop/Spark y mensajería de gran fiabilidad.

- Desarrolla un aprendizaje automático mediante servicios de machine learning de gran velocidad, escalabilidad y facilidad de uso, a partir de los datos.
- Posee una consola de administración Web de herramientas, mediante la cual es posible realizar supervisión y registro, así como análisis y diagnóstico, de fácil uso.
- Posee una interfaz de línea de comandos y otras herramientas para desarrolladores, para implementar y desarrollar aplicaciones a través de la misma.
- Posee la capacidad e integración y escalabilidad, que otorga gran comodidad, lo cual otorga beneficios incalculables para los usuarios.
- A través de la plataforma Google ha desarrollado un sistema muy funcional para las pruebas de integración el cual es muy rápido y efectivo.
- A través de SQL, tiene la propiedad de almacenamiento y manejo de datos, muy funcional para los desarrolladores, dado el manejo relacional y administración de base de datos que posee SQL.
- Tiene la capacidad de manejo total y masivo de servicio de base de datos que los desarrolladores pueden administrar con escalabilidad y mucha rapidez.

- Posee un stock de herramientas de manejo que otorga a los desarrolladores la capacidad de monitoreo y diagnóstico para las aplicaciones así como el reporte de la identificación de errores, además de la investigación del comportamiento del código en el proceso de desarrollo, permite el manejo de recursos a través de templates lo cual se realiza mediante la plataforma de consola integrada, que posee una infraestructura de línea de comandos, interfaces y métodos que se pueden manejar de forma programática, para distintos proyectos a desarrollar a través de la plataforma de Google.
- Tiene la propiedad de controlar el acceso y la visibilidad de los recursos a través de una plataforma de identidad protegida por el modelo de seguridad muy efectivo, dado las herramientas de supervisión de última generación con el que cuenta el desarrollador de Google.

### **1.5. Desarrollador de herramientas Google Chrome**

El sitio transcripción de herramientas para desarrolladores de Google Chrome indica que “el set de herramientas para desarrolladores Web de Google Chrome, funciona desde el propio servidor Google Chrome, y contiene recursos muy útiles para optimizar el rendimiento, desde el menú de la aplicación Web, estos son muy prácticos y funcionales para los desarrolladores y usuario en general”<sup>12</sup>.

Dentro de este set de herramientas, existen grupos fundamentales de recursos disponibles, entre las cuales se encuentran:

---

<sup>12</sup> Google Chrome Developer Tools. *Transcripción de Herramientas para desarrolladores de Google Chrome*. p. 7.

- Elements, que permite ver información estructurada sobre la página que está en uso y que a su vez despliega variedad de tareas, entre las cuales están:
  - Inspeccionar HTML y CSS de una página web
  - Probar distintos diseños
  - Editar CSS
  
- Los Dev Tools, que son un grupo de herramientas de depuración integradas en Google Chrome, estos a su vez:
  - Dan acceso a lo interior del navegador y su aplicación Web.
  - Rastrear eficientemente problemas de diseño.
  - Establecen puntos de interrupción de JavaScript.
  - Brindan ideas muy efectivas en función de optimizar el código que se maneja.
  
- Resources, da acceso a inspeccionar las fuentes de datos locales de la aplicación en que se trabaja, incluye IndexedDB o bases de datos SQL, almacenamiento local y la sesión, cookies y recursos de caché de la aplicación. Este a su vez:
  - Inspecciona de manera rápida los recursos visuales de la aplicación en uso, incluyendo imágenes, fuentes y hojas de estilo.
  
- TimeLine, que permite grabar y analizar toda la actividad de la aplicación conforma esta se ejecuta. Esta tiene tres secciones:
  - Sección de información general en la parte superior

- Vista de registro
- Barra de herramientas
- Panel de perfiles, admite perfilar el tiempo de ejecución y el uso de memoria de una aplicación Web o página. Contiene dos perfiladores:
  - Dan acceso a entender en donde son disipados los recursos, de esta manera favorece a la optimización del código.
- Consola, ofrece una visión generalizada de los usos comunes de la consola. Es posible navegar por la API de consola y la API en línea de comandos, que es el material de referencias que permite entender más funciones.
- Depuración con JavaScript, ayuda a descubrir la causa de un problema y solucionarlo eficientemente.
- Drag Design, ofrece a los desarrolladores la funcionalidad de alterar el diseño de una página Web, simplemente, con el mecanismo de arrastrar los elementos para moverlos de lugar.
- CSS remove and combine, permite analizar el CSS de la Web, eliminar secciones completas, combinar, confeccionar un CSS que se puede bajar y utilizar.
- ColorPick Eyecropper, permite usar la página de colores usada en la pag. Web.

- JavaScript Errors Notifier, verifica si la página, Web que se visita presenta errores de JavaScript, siempre está alerta y da un listado de fallos en caso que estos estén presentes.
- Speed Tracer, extensión elemental que permite medir el rendimiento de una página, Web para dar una idea del tiempo que tarda en cargar o si necesita ser optimizada.
- Network, otorga información sobre cada operación de la red en la aplicación, incluye datos de sincronización detallada, petición HTTP, cabeceras de respuesta, cookies, datos, WebSocket y otros, a la vez.
  - Resuelven dudas sobre el rendimiento de la red.

“La función del panel Network de las herramientas para desarrolladores de Google Chrome, es el de mayor utilidad, dado que revela información referente al intercambio de recursos, permitiendo observar el protocolo que se emplea, cuando ha sido descargado del cache una herramienta que se usa para comprobar el funcionamiento exacto del servidor, en lo relativo a los tiempos de descarga de cada recurso que producen los tiempos de toda la conexión e intercambios de información, que mediante la selección de uno de los recursos que han sido descargados, es posible observar el desglose de tiempos que conforman el tiempo de descarga de un recurso determinado.<sup>13</sup>”

Estos tiempos a los cuales se hace mención, producen el tiempo de descarga total de un recurso y se clasifican conforme su función, de la manera siguiente:

- Queueing: se refiere a la función del tiempo cuando la petición ha sido pospuesta o se encuentra en espera, debido al orden de prioridad con respecto de otras peticiones, liberación de un socket TCP, dado que el

---

<sup>13</sup> RAMIREZ HERNANZ, Eloy. *Desarrollo y aplicación de herramientas para el análisis del envío anticipado de video mediante HTTP/2 en servicios de streaming multimedia*. p. 17.

navegador admite exclusivamente seis conexiones TCP o bien sea mediante la introducción de recursos en cache.

- Stalled: es acerca del tiempo que espera la petición que se realiza, previamente a ser enviada, dado a las condiciones de Queueing, o a una negociación con un proxy.
- DNS Lookup: es la función sobre el tiempo que se lleva a cabo la búsqueda DNS.
- Initial connection + SSL: hace referencia al tiempo que se tarda en establecer la conexión TCP y realizar la negociación SSL.
- Request Sent: es la función específica que se refiere al tiempo que tarda en formular, manifestar y prescribir la petición del recurso.
- Waiting (TTFB): es la función sobre el tiempo desde que se lleva a cabo la petición del recurso hasta el momento cuando el navegador recibe el primer bit del mismo (RTT).
- Content Download, se refiere a la función sobre el tiempo que tarda en descargar el recurso que ha sido enviado por el servidor.
- Request Headers y Response Headers: hacen alusión a la información que se observa acerca de las cabeceras de petición y de respuesta de los recursos.

## 1.6. Compilador

Un compilador es un programa informático que se maneja para traducir las instrucciones que se encuentran escritas en un lenguaje de alto nivel, en función de que el compilador, traslade tales instrucciones a un lenguaje de máquina, que también podría ser un código intermedio (bytecode) o sencillamente texto.

Para efectuar este proceso, el compilador recibe como datos de entrada el código fuente de determinado programa, escrito por un programador y que genera un programa equivalente como salida a un conjunto de instrucciones que se encuentran escritas en el lenguaje binario del ordenador en el cual se va a ejecutar el programa.

De esta manera un programador puede diseñar un programa en un lenguaje determinado y luego compilarlo a un programa que sea más manejable y entendible por un ordenador.

“El proceso que realiza un compilador de traducción se conforma de distintas fases, que llevan a cabo operaciones diferentes y lógicas”<sup>14</sup>. Por lo tanto, es importante entender que tales fases son etapas separadas del traductor y pueden ser escritas como operaciones decodificadas de forma separada; no obstante, en la práctica, de manera frecuente, se integran en conjunto. Tales fases, que lleva a cabo el compilador, son las siguientes:

---

<sup>14</sup> FLORES GARCÍA, Jaime. *¿Qué es un compilador y cómo funciona?* p. 15.

- Fases de análisis
  - Análisis de léxico: es la primera de las fases que lee el programa fuente de izquierda a derecha y se agrupa en componentes léxicos (tokens), que hacen referencia secuencial de caracteres con un significado. En este, los espacios y líneas en blanco, comentarios y otras informaciones innecesarias son eliminadas del programa fuente. Además, se hace la comprobación de que los símbolos del lenguaje se han escrito de forma correcta. Para ello son necesarios los métodos de especificación y reconocimiento de patrones autómatas y finitos que admiten expresiones regulares. No obstante, un analizador léxico, además, es la parte del traductor que maneja la entrada del código fuente que utiliza bastante tiempo y en este el analizador léxico debe de funcionar de la forma más eficiente posible.
  - Análisis sintáctico: es la fase en la cual los caracteres léxicos se agrupan de forma jerárquica en frases gramaticales, las cuales utiliza el compilador en función de sintetizar la salida. Dicha estructura jerárquica de un programa, generalmente, es expresada utilizando reglas recursivas como: cualquier identificador es una expresión o cualquier número es una expresión. Es necesario y muy importante mencionar que la división que se da entre el análisis léxico y el sintáctico es de alguna manera parcial. Un elemento para determinar la división entre estos es si una construcción del lenguaje fuente es propiamente recursiva o no lo es.



- Análisis semántico: se da al momento de revisar el programa fuente para tratar de encontrar errores de tipo semántico y colecta la información acerca de los tipos para la frase posterior de generación de código. Se usa la estructura de tipo jerárquico determinado por la etapa de análisis sintáctico para identificar los operadores y operando de expresiones y proposiciones. Además, en este análisis se lleva a cabo la verificación de tipos y el compilador verifica si cada operador tiene operados validos por la especificación del lenguaje fuente.
- Fase de síntesis: es acerca de la generación del código objeto equivalente al programa fuente. Únicamente, se genera código objeto cuando el programa fuente se encuentra libre de errores de análisis. Generalmente el código objeto es un código de máquina de tipo re-localizable o código ensamblador. Las posiciones de memoria son seleccionadas para cada una de las variables utilizadas por el programa.

Un aspecto de relevancia es que posteriormente, cada una de las instrucciones intermedias, se traducen a una secuencia de instrucciones de máquina que realiza la misma tares. Además, es sumamente importante mencionar que se realiza la asignación de variables de registros.

- Generación de código intermedio: después de realizados los análisis sintáctico y semántico, algunos compiladores desarrollan una representación intermedia clara y concisa del programa fuente. Tal representación es considerada como un programa para una máquina abstracta que debe contener dos propiedades básicas:

- Ser de fácil producción
- Ser de fácil traducción al programa objeto

La representación intermedia puede ser de diferentes formas. Una forma intermedia se denomina código de tres direcciones que se considera como lenguaje ensamblador de una máquina en la cual cada posición de memoria puede funcionar como un registro.

Este código sigue una secuencia de instrucciones y cada una tiene tres operandos a lo sumo, la cual tiene las propiedades siguientes:

- Primera: cada instrucción de tres direcciones posee a lo sumo un operador y su asignación y al generarse las mismas, el traductor decide el orden en que se efectuarán las operaciones.
- Segunda: el traductor necesita generar un nombre transitorio en función de guardar los valores calculados por cada instrucción.
- Tercera: ciertas instrucciones de tres direcciones poseen menos de tres operandos.
- Optimización de código: es la fase que consiste en mejorar el código intermedio, de manera que de cómo resultado un código máquina que sea más veloz de ejecutar.

Esta etapa de síntesis, es posible, ante todo si el traductor es un compilador.

No obstante, existe mucha variabilidad en la cantidad de optimización de código que realizan los diferentes compiladores; entre estos, los conocidos como compiladores optimizadores que toman bastante tiempo del compilador.

A pesar de ello, existen optimizaciones sencillas que tienen la capacidad de mejorar de manera sensible el tiempo de ejecución del programa objeto sin que se retarde demasiado el proceso de compilación



## **2. HERRAMIENTAS DE ANÁLISIS DE CÓDIGO Y SU FUNCIONALIDAD**

Las herramientas de análisis de código hacen referencia a los recursos tecnológicos usados con el objetivo de observar, investigar, explorar, corregir y mantener el desarrollo de estándares altos de excelencia del código, así como de optimización en su aplicación y ejecución.

La función de dichas herramientas de análisis de código se basa esencialmente en lo siguiente:

- Examinar el desarrollo del código, en función de encontrar problemas de diseño.
- Detección de errores de uso.
- Detectar problemas de mantenimiento.
- Detectar problemas de estilo, entre otros.

Además, estos recursos también pueden analizar la calidad de la aplicación, la complejidad y mantenimiento del código administrado, localizar código duplicado y dar un seguimiento del proceso de desarrollo y de ejecución del código.

Estas herramientas de análisis de código tienen la capacidad de crear, administrar y ejecutar pruebas de índole manual y automatizada en función de validar y analizar con una mejor perspectiva el código.

El uso de las herramientas de análisis de código puede otorgar la facilidad de realizar los ajustes que sean necesarios en el sistema y software, en función de:

- Reducir riesgos
- Evitar sorpresas poco agradables
- Administrar y prever el costo de los proyectos

## **2.1. Herramientas de análisis de código**

### **2.1.1. Herramienta de código estático Google Closure**

La herramienta Google Closure tiene dentro de sus funciones la premisa de asistir a los desarrolladores en todo lo relativo a la optimización del código estático. El propósito de dicha herramienta consiste en mejorar la estructura en la cual se encuentra el código en lenguaje JavaScript y busca eliminar todo aquel código que sea innecesario que pueda, además de otras situaciones, volver más lenta la ejecución de la herramienta.

Esta herramienta, dentro de sus características propias y fundamentales, posee las funciones de:

- Ejecutar procesos de eliminación de códigos innecesarios.
- Realizar procesos de revisión de la sintaxis y estilo.
- Recopilar y visualizar métricas del código fuente.
- Realizar procesos de revisión de la referencia de variables.
- Llevar a cabo la revisión y detección de los errores comunes del lenguaje JavaScript.

“La herramienta Google Closure consiste en un optimizador de JavaScript que tiene la función de compilar aplicaciones Web que se encuentran bajo el código JavaScript compacto y que presenta un nivel alto de rendimiento. Este en su proceso descarta el código muerto y posteriormente, reescribe y disminuye lo que queda de este, con el propósito de que sea ejecutado de forma veloz en los motores de JavaScript de los navegadores”<sup>15</sup>.

Dicho compilador, además, hace una comprobación de la sintaxis y de las referencias a variables y de los tipos y emite una advertencia acerca de otras situaciones problema habitual que se manejan dentro de JavaScript.

Los controles y optimizaciones que realiza el compilador Closure coadyuvan a escribir aplicaciones depuradas y mejoradas y que tengan mayor facilidad de mantenimiento.

Dentro de los beneficios que otorga la herramienta Google Closure, se pueden mencionar textualmente las siguientes:

- Eficiencia: el compilador minimiza el tamaño de los archivos JavaScript y, a su vez, los torna más eficaces; logra así que las aplicaciones carguen con mayor rapidez y requieran de menor ancho de banda.
- Comprobación de código: dado que el compilador crea un aviso acerca de JavaScript y acerca de operaciones de riesgo potencial.

Este compilador Closure, además, tiene herramientas complementarias que funcionan en el navegador Firefox y que colaboran a su mejor funcionamiento y propósito:

---

<sup>15</sup> Google Closure. *El compilador closure, funciones y beneficios*. p. 3.

- Closure inspector: representa una extensión, un plugin del agregado Firebug; tiene la función de asistir a los programadores a comprender y depurar el código javascript compactado (más conocido como debugging de errores en javascript ofuscado), con mayor facilidad; es decir, permite que el código sea más legible para los operadores.
- Closure library: consiste en una galería de código JavaScript, previamente programado, que asiste a los desarrolladores a realizar gestiones tecnológicas que poseen cierta sofisticación. Esta consiste en una librería tipo Mootools o JQuery con acceso a un grupo de widgets de interfaz de usuario y a controles reutilizables que son de bajo consumo, entre otras opciones como estructuración de datos o animaciones.
- Closure Templates: es una función reciente para crear templates HTML (conocidas también como plantillas) de manera dinámica; tiene el propósito de ampliar el código para facilitar la creación de interfaz de usuarios de JavaScript y HTML que puedan ser reutilizables.
- Closure Linter: es la utilidad más nueva de Google Closure tiene la función de comprobar los archivos de JavaScript, en lo que respecta a estilo en todo lo referente a:
  - Colocación del operador.
  - Colocación de puntos y comas que hacen falta dentro del texto.
  - Verificación en lo que respecta a que el espaciado sea correcto y simétrico.
  - Revisión acerca de la presencia de anotaciones como JSDoc.



Google Closure, además, posee la función de minificar especialmente en lo que respecta al lenguaje JavaScript; a través del mismo es posible desarrollar un proceso de construcción con acceso a minificar y modificar el nombre de los archivos de desarrolla y almacenarlos en un directorio denominado de producción.

Minificar es un proceso a través del cual se hace posible acelerar una descarga, el análisis y también el tiempo de ejecución y en el caso específico, de los lenguajes JavaScript y CSS (*Cascading style sheets*-hoja de estilo) se da la posibilidad de reducir aún más el tamaño del archivo, en el momento de modificar el nombre de las variables, lo cual se realiza una vez que el código HTML se encuentre actualizado de forma correcta en función de garantizar que los selectores sigan funcionando.

La ventaja de la función de minificar con Google Closure es que se genera una versión mucho más optimizada del código, en especial de HTML, dado que esta herramienta Closure es un optimizador por excelencia y una de sus funciones estriba en compilar las aplicaciones Web para modificarlas en más ágiles y compactas, lo cual realiza de la siguiente manera:

- Elimina el código redundante.
- Reduce el tamaño del código lo más que sea posible.
- Efectúa tales acciones principalmente en la declaración de tipos y funciones.

Otras funciones de la herramienta Closure, en su propiedad de compilador, de mucha ayuda para los desarrolladores, específicamente a JavaScript, importantes de mencionar, son:

- Descargar y ejecutar con mayor rapidez JavaScript
- Eliminar el código inservible (no funcional)
- Minimizar lo que queda luego de la eliminación del código inservible
- Comprobar la sintaxis
- Verificar las referencias a variables
- Confrontar los tipos de datos
- Advertir sobre los peligros más comunes de JavaScript

Es importante, también, mencionar que en lo relativo al lenguaje Java, Google Closure puede ser utilizado como una aplicación Java que es ejecutable desde la línea de comandos, en función de un framework (jQuery, Mootools o como una API RESTful).

Google Closure es, por lo tanto, una herramienta muy efectiva que funciona desde su característica de compilador hasta un sistema con el cual se pueden crear templates html de forma muy dinámica.

La herramienta Google Closure es considerada como un proyecto liberador de herramientas y su uso representa una gran ventaja y otorga aún mayores beneficios a los desarrolladores y usuarios comunes, cuando es utilizada y aprovechada en su conjunto complementario como Google Closure Tools (Kit).

### **2.1.2. Otros lenguajes de programación**

Existen diversos lenguajes de programación que son herramientas muy versátiles, funcionales y útiles en todo lo referente al manejo de programas, sistemas y software, los cuales presentan diversas configuraciones y características que, por su funcionalidad, son importantes de conocer.

Tales herramientas permiten desarrollar, crear programas y software, parten de la premisa de que un ordenador desarrolla su funcionamiento bajo el control de un programa. Dicho programa requiere estar almacenado en la unidad de memoria, el disco duro, en la cual los lenguajes de programación son las herramientas conocidas como código de máquinas o lenguaje de máquina.

“Se indica que los lenguajes codificados en un ordenador específico son lenguajes desarrollados en código de máquina y, por lo tanto, son de difícil entendimiento para una persona o usuario, dado que no tiene el conocimiento o la preparación para entender tal o tales lenguajes, siendo que los mismos se encuentran configurados con códigos numéricos que no tienen sentido nemotécnico alguno”<sup>16</sup>.

De ahí se origina que los lenguajes de programación, como su nombre lo indica, tienen la función fundamental de programar, los cuales han sido configurados de formas apropiadas, de manera que den acceso y oportunidad a ser leídos y escritos por las personas. Dichos lenguajes, a su vez, tienen independencia del modelo de computador a usar, representan simbólicamente, en forma de texto, los códigos y es de esta manera que pueden ser leídos por un usuario.

#### **2.1.2.1. Java**

El lenguaje Java es configurado y desarrollado con el objetivo de crear aplicaciones autónomas, tal es el caso de las applets, que son programas que pueden ser incluidos en las páginas web.

---

<sup>16</sup> La revista informática. *Lenguaje de programación C#*. <http://www.larevistainformatica.com/C1.htm>. Consulta: 11 de octubre de 2016.

“El lenguaje Java, fue creado por Gosling, de (emacs) James y Joy, Bill de (Sun). Este se deriva de un lenguaje anterior conocido como Oak. De ahí se originan los criterios iniciales para el diseño y desarrollo de Java, los cuales textualmente fueron: independencia de la máquina, seguridad para trabajar en la red, potencia para sustituir código nativo.”<sup>17</sup>

Las características de Java se basan en tener la configuración de un lenguaje compilado y a su vez interpretado, dado que todo programa en lenguaje Java requiere de ser compilado y el código que se genera, denominado bytecodes, es interpretado por una máquina virtual; es de esta manera en la cual se obtiene la independencia de la máquina; el código compilado es ejecutado en máquinas virtuales, las cuales son de carácter independiente de la plataforma como tal.

El lenguaje Java se orienta a objetos que son de propósito general, no obstante, Java es ya conocido como un lenguaje de programación de applets, los cuales son ejecutados en el contexto de un navegador web, el cual puede ser usado para desarrollar cualquier clase de proyecto.

La sintaxis de Java tiene similitudes con la de los lenguajes C y C++, pero es necesario aclarar que Java no evoluciona ni procede específicamente de ninguno de esos lenguajes. Su seguridad va desde el ámbito del programador hasta el de la ejecución en la máquina virtual. Garantiza, también, la seguridad de la ejecución antes de ser ejecutado el código. Es también un conjunto de tecnologías que se basan en el mismo lenguaje, el cual permite desarrollar aplicaciones para: gráficos, multimedia, la web, programación distribuida, bases de datos, entre otras. Además, Java contiene un grupo de reglas de codificación

---

<sup>17</sup> BELMONTE FERNANDEZ, Oscar. *Introducción al lenguaje de programación Java: una guía básica*. p. 5.

que se utilizan de forma común y general por los programadores y las cuales se hace necesaria conocer al ser utilizado por el usuario.

### **2.1.2.2. Visual Basic**

Visual Basic es un lenguaje práctico y sencillo que tiene el propósito principal de crear aplicaciones y tiene la propiedad de utilizar una interfaz visual que da acceso a programar ambientes gráficos y llevar a cabo diversidad de tareas, simplemente realizando operaciones con el mouse, sobre la pantalla del ordenador sin necesidad de escribir código. Es considerado como uno de los lenguajes de programación que llama mayormente la atención entre los desarrolladores, programadores y usuarios, dado que permite la ejecución de tareas complejas en un lapso corto de tiempo.

“El lenguaje Visual Basic es un lenguaje de programación y a su vez una aplicación que fue creado en 1991 por Cooper, Alan para Microsoft. Este se deriva del clásico lenguaje conocido BASIC. El lenguaje Visual Basic utiliza una interfaz completamente visual tal como su nombre lo revela.”<sup>18</sup>

Los programas creados en Visual Basic funcionan únicamente en Windows y admiten la creación de ventanas, botones, menús, etc., de una manera práctica que consiste en arrastrar y soltar los elementos. Además, permite definir apariencias, posiciones y comportamientos, ya sea de manera visual o usando los códigos de programación pertinentes.

El lenguaje Visual Basic en su configuración adopta elementos de distintos modelos como el que se orienta a objetos y el que se orienta a eventos. Este se considerar un sistema RAD (*rapid application development*), dado que

---

<sup>18</sup> ALEGSA. *Diccionario de informática y tecnología*. p. 5.

desarrolla aplicaciones de forma rápida, básicamente, en lo que se refiere a los prototipos, valiéndose de Data Access Objects, Remote Data Objects o ActiveX Data Objects, así como el desarrollo de controles ActiveX y de objetos. Los programas de Visual Basic también pueden valerse del API de Windows.

La característica más importante de Visual Basic es que da acceso a la programación de contenidos informáticos gráficos de forma simple y muy accesible, es decir, es de fácil manejo y es dirigido por eventos que le dan mejor operatividad, lo que le otorga mejores resultados. Este lenguaje puede ser utilizado y manejado tanto por expertos desarrolladores y programadores, así como por usuarios comunes y principiantes. Se deriva y fundamenta en el dialecto BASIC, también, maneja componentes nuevos que les dan acceso a lenguajes informáticos modernos. Algunas de las propiedades de Visual Basic son:

- La programación gráfica se realiza directamente.
- No necesita de escritura de códigos de programación por parte de los usuarios.
- Trabaja a partir de lenguajes RAD en inglés (*rapid application development*), que se refiere al desarrollo rápido de aplicaciones específicas para cada necesidad y función.
- Por su lenguaje sencillo y simple es ampliamente adaptable a las distintas plataformas de los sistemas de Windows.
- Se acomoda y adapta con facilidad a otros lenguajes de mayor complejidad.

- Su última versión se ha desarrollado con una combinación de elementos simples como es propio de Visual Basic, así como de elementos perfeccionados que le otorgan al lenguaje mayor calidad y diversificación, además, mayor adaptabilidad.
- Es fácil de aprender y es muy recomendable en aplicaciones de escritorio.

### 2.1.2.3. C# (C Sharp)

El lenguaje C# (C Sharp) es un lenguaje de programación desarrollado para objetos, el cual a través sus propiedades tiene el propósito de mejorar los lenguajes C y C++. “El lenguaje C#, fue desarrollado por el danés, Hejlsberg Anders, quien, a su vez, creó los lenguajes Turbo Pascal y Delphi. Este lenguaje C#, se conoce como C Sharp o C sostenido”<sup>19</sup>. El referido lenguaje se creó con el objetivo de incorporar al mismo los beneficios del lenguaje Java, especialmente en todo lo que se refiere a sus propiedades de productividad, adaptándole además las ventajas de los lenguajes, en especial la potencia de C y C++; ventajas con las cuales se da un ahorro de tiempo en la programación, el mismo posee un archivo de clases muy bien diseñado y completo.

El lenguaje C# está conformado dentro de la plataforma .NET, una interfaz de programación que desde sus inicios se desarrolló con el propósito de crear programas sobre la plataforma .NET con la que es posible trabajar de manera agradable y visual con diversos lenguajes que la conforman. El objetivo de C#, al momento de su creación, fue mejorar al lenguaje Visual Basic. Es compilado como código administrado, es decir, tiene la ventaja de los servicios de CLR (*common language runtime*) que poseen dentro de su configuración,

---

<sup>19</sup> HEJLSBERG ANDERS, Danés. *Lenguaje de programación C#*. p. 2.

interoperabilidad entre lenguajes, recolección de elementos no utilizados, una mejor seguridad y mayor compatibilidad entre las distintas versiones desarrolladas. Algunas de las características del lenguaje C# son:

- Es un lenguaje de propósito general, orientado a objetos, elegante.
- Su herencia en cuanto a las ventajas que posee derivadas de los lenguajes C, C++ y Java, Visual Basic y Delphi.
- Los métodos virtuales que utiliza para su desarrollo.
- La sobrecarga de operadores que ha conseguido C#.
- La plataforma en la que se desarrolla está contenida en el conjunto de programas Visual Studio .NET.
- Utiliza plantillas de: proyecto, diseñadores, páginas de propiedades, asistentes de código, modelo de objetos, entre otras propias del entorno de desarrollo.
- Es un lenguaje de carácter potente, versátil y sencillo de manejar.
- Este lenguaje incluye especificaciones de sintaxis, semántica y de diseño.
- Posee una biblioteca para programar en C# y compilar en diversas funciones que se ejecutan en .NET Framework.



#### 2.1.2.4. PHP

El lenguaje PHP es un lenguaje de código abierto muy conocido y apropiado para el desarrollo Web, que puede ser alojado en HTML; muchas páginas y portales Web han sido creadas con PHP, un código libre y gratis para los desarrolladores. El hecho de encontrarse alojado en HTML significa que está incrustado en un mismo archivo en el cual se puede combinar el código PHP, conjuntamente con HTML, siguiendo ciertas reglas para ello. La sintaxis de PHP se origina en una buena parte de los lenguajes C, Java y Perl, su propósito principal es dar acceso de forma veloz a los desarrolladores, a generar páginas dinámicas.

“El lenguaje PHP (Hypertext Pre-procesor) fue desarrollado por el programador canadiense Lerdorf Rasmus, en 1994, proviene de la escritura de un grupo CGI binarios escritos en el lenguaje C. Es muy utilizado para el desarrollo de páginas Web dinámicas, programando scripts del lado del servidor.”<sup>20</sup>

Este lenguaje se procesa en servidores con la propiedad de ser potentes ordenadores, que poseen software y hardware especial, como es el caso de Internet Explorer, Firefox o Chrome, a los cuales se envían los datos de solicitud al servidor que los procesa; reúne los datos y el servidor lo que devuelve es una página HTML, pero de forma estática. Para ello, se utilizan ISS y Apache para desarrollar un contenido dinámico que sea posible interactuar con el usuario.

---

<sup>20</sup> GONZALEZ, Enrique. *¿Qué es PHP? ¿Para qué sirve? Un potente lenguaje de programación para crear páginas Web.* p. 5.

Cuando en el contenido de un archivo se incluye el código HTML y en dado momento se visualiza una etiqueta escrita así `<? php y?>`, quiere decir que es ahí en donde inicia el código PHP y donde se dice que está incluido en el código HTML.

Las etiquetas especiales de PHP admiten poder entrar y salir del modo PHP. Estas sirven al servidor Web, en función de conocer que se tiene que interpretar el código que se encuentra incluido entre dichas etiquetas como PHP.

Lo que diferencia al lenguaje PHP del lenguaje JavaScript, estriba en que el código es ejecutado en el servidor, generando lenguaje HTML y dirigiéndolo al cliente en forma de página Web en modo estático. Es ahí en donde el cliente obtendrá los resultados que el servidor retorna, posteriormente de interpretar el código PHP, no habiendo posibilidad de establecer que código es el que produjo el resultado que se recibió. Esto significa que, mediante el código que se tiene en el navegador en uso es posible ver el código HTML, más no se puede visualizar el código PHP que generó; sin embargo, no será posible visualizar el código PHP que originó el resultado PHP. Por lo tanto, el servidor Web, puede ser también configurado en función de que los usuarios ignoren si están usando PHP.

El propósito esencial de PHP consiste en lograr la integración de las páginas HTML, con aplicaciones que corran en el servidor como procesos integrados en el mismo y no como un proceso separado, como se dio con los Common Gateway Interface

Las características de PHP son:

- PHP es extremadamente veloz, estable, seguro y simple, así como expandible y modificable, para el principiante.
- Es un lenguaje gratuito y multiplataforma.
- Corre casi en cualquier plataforma, usando el mismo código fuente.
- La sintaxis de PHP es similar a la de C.
- Permite realizar con un Script CGI (*common gateway interface*), el procesamiento de información en:
  - Formularios
  - Foros de discusión
  - Manipulación de cookies y páginas dinámicas, entre otros
- Ofrece a su vez, muchas propiedades avanzadas para los programadores profesionales y más avanzados.
- Con este lenguaje de PHP es posible procesar la información de formularios, generar páginas con contenidos dinámicos o enviar y recibir cookies, entre muchas más cosas.
- PHP es utilizado desde páginas pequeñas hasta grandes empresas.
- Muchas aplicaciones web están construidas usando PHP entre estas se pueden citar:

- Joomla y Drupal (gestores de contenido de páginas web).
- osCommerce y Prestashop (tiendas on-line para comercio electrónico).
- phpBB y SMF (sistemas de foros para páginas web).
- Moodle (plataforma educativa para educación on-line), entre otros.
- Dentro de su configuración se incluye la creación de imágenes.
- Puede usar y presentar resultado en otros estándares de datos o lenguajes propios de los desarrolladores Web como HYML y cualquier otro tipo de ficheros XML.
- Puede autogenerar archivos y luego almacenarlos, en lugar de presentarlos en pantalla, utilizar tales ficheros para crear contenido dinámico.
- Es posible interactuar con otros servidores y motores de bases de datos utilizando cualquier protocolo. Entre estos están:
  - MySQL
  - MS SQL
  - Oracle
  - Informix
  - PostgreSQL, entre otros.

- PHP puede vincularse con otros lenguajes muy potentes como Java.
- PHP es un lenguaje potente, muy común de usarse y de mucho interés para los desarrollos web.

#### 2.1.2.5. HTML

El lenguaje HTML se utiliza de manera generalizada para desarrollar la estructura y el contenido de los sitios web en Internet en sus distintos elementos que incluyen el texto, los diversos objetos y las imágenes que se visualizan en las referidas páginas. Además, este lenguaje, precisa una estructura básica y un código, conocido y nombrado código HTML.

“El lenguaje HTML (*hyper text markup language* o lenguaje de marcas hipertextuales) fue desarrollado por Berners-Lee, Tim, en 1991, derivado de *standard generalized markup language* (SGML) o lenguaje de marcación generalizado. Lo cual significa que HTML fue creado originalmente como un lenguaje de marcas para desarrollar cualquier clase de documentos estructurados.”<sup>21</sup>

Este lenguaje HTML no tiene las características de lenguajes de programación como las de Delphi o C, sino que se refiere a un sistema de etiquetas. Dichas etiquetas o marcas son colocadas a un texto, con las cuales puede darse la forma en la que se requiere que este sea visualizado. Las etiquetas son el modo de empleo por el cual funciona HTML, las cuales interpreta el navegador y las traduce a las páginas en las cuales los servidores consultan información o visualizan diariamente en la Web.

---

<sup>21</sup> BERNERS LEE, Tim. *Lenguaje de programación HTML*. p. 12.

Dichas etiquetas son también denominadas tags, se contienen en corchetes o paréntesis angulares <>, a los cuales se les designa signos de mayor y de menor qué. Estas etiquetas dan acceso a la interconexión de la información escrita en el lenguaje HTML, entre las conceptualizaciones y los formatos.

El lenguaje HTML es un lenguaje de mucha simplicidad y practicidad de interpretación en lo que respecta a sus directrices o etiquetas y estas representan las instrucciones a las cuales obedece el navegador en función de determinar la manera en la cual debe aparecer en la página web.

HTML en sus orígenes, desarrollaba documentos de índole muy básico; no obstante, con el desarrollo de la Internet WWW (*world wide web*), dicho lenguaje fue mejorando por el ritmo en el cual avanza la red de redes.

Por medio del lenguaje HTML se hace accesible visualizar los archivos que se encuentran en la Internet. Además, a través del mismo se hace posible comprar, vender y ofertar cualquier clase de servicios en el mercado sin fronteras en la red. Debido a que todas las páginas que se encuentran en la Internet están desarrolladas con base en el lenguaje HTML.

Por sus propiedades, dado que es un lenguaje interpretado de alto nivel, es esencial conocer ciertas características que definen su funcionamiento y manejo:

- HTML es un lenguaje especializado en el desarrollo de páginas web para el cual se usan diversas etiquetas que tienen la función de crear estructuras estrictamente definidas.

- La estructuración de HTML se inicializa con la etiqueta <HTML>, la cual contiene a toda la página Web.
- Este lenguaje, en su estructura, contiene dos unidades básicas muy bien diferenciadas: el encabezado y el cuerpo, correspondientes respectivamente a las etiquetas principales, <head> y <body>.
- El encabezado puede contener información y, por lo tanto, siempre maneja el título del documento HTML, circunscrito por la etiqueta <title>.
- Es en el cuerpo del documento, que se localiza el contenido completo de la página Web, conteniendo así:
  - El texto.
  - Las imágenes.
  - El sonido.
  - Los hipervínculos.
  - El video y otros.
  - Tiene la propiedad de redirigirse a otras páginas, saltos de líneas, listas, tablas, etc.

#### **2.1.2.6. SQL**

SQL es un lenguaje de programación estándar que se caracteriza por su interactividad en lo que respecta a la obtención de datos desde una base de datos que posee una interfaz de programación.

“El lenguaje SQL es estándar con características ANSI/ISO de definición de manipulación y control de bases de datos de tipo relacional. Es por sus propiedades un lenguaje declarativo, dado que su manejo es instruccional. Este es muy similar al lenguaje natural y específicamente al idioma inglés dado que es muy expresivo y a través de este es posible tener acceso a los sistemas relacionales de tipo comercial.”<sup>22</sup>

El lenguaje SQL fue creado en sus inicios por IBM y era conocido como SEQUEL (*structured english query language*); en 1982, ANSI (American National Standards Institute), lo definió como un lenguaje de bases de datos relacionales, y por estas y otras razones, posteriormente se le denominó SQL (*structured query language*), tal como se conoce actualmente, siendo un lenguaje muy popular. En este lenguaje se distinguen tres etapas:

- Nivel introductorio (*entry*): que incluye a SQL89 y las definiciones denominadas de clave primaria y foránea en la creación de una tabla.
- Nivel intermedio (*intermediate*): que aparte del SQL89, integra algunas ampliaciones del SQL92.
- Nivel completo (*full*): que ya incluye todas las ampliaciones del SQL92.

Acerca del modelo relacional, este tiene como estructura básica el almacenamiento de datos relacionales, lo cual se deriva de una relación que consiste en la denominación que se da a la relación y a un conjunto de atributos. Tal extensión de relación se refiere a un conjunto de registros y al momento de trabajar con SQL, dicha nomenclatura se modifica de la siguiente manera:

---

<sup>22</sup> ESCOFET MARTÍN, Carmen. *El lenguaje SQL*. p. 62.



- Se habla de tablas cuando se hace referencia a las relaciones
- Se habla de columnas, cuando se hace referencia a los atributos
- Se habla de filas, cuando se hace referencia a registros

No obstante, a la nomenclatura referida, los conceptos iniciales continúan siendo los mismos.

Las operaciones que realiza se denominan sentencias, que se manejan a través de notaciones, las cuales están conformadas por cláusulas.

Cuando se requiere acceder a la base de datos, desde una aplicación de lenguaje de programación cualquiera, para poder utilizar SQL, se necesitan sentencias especiales que den acceso y se distingue entre las instrucciones del lenguaje de programación y las sentencias de SQL, que en casos como este trabaja como un lenguaje hospedado.

SQL posee sencillez y homogeneidad que permiten realizar las siguientes instrucciones:

- Crear bases de datos, tablas, dominios, aserciones y vistas, usando la sentencia create.
- Modificar tablas y dominios, utilizando la sentencia alter.
- Borrar bases de datos, tablas, dominios, aserciones y vistas, utilizando la sentencia drop.

La adecuación de las sentencias en SQL hace las diferencias que se requieren perfilar, al realizar las descripciones individuales de cada una.

Lo anterior indica que, con el uso de SQL es posible definir, manipular y controlar una base de datos relacional.

### **2.1.2.7. Oracle PL SQL**

Este lenguaje se encuentra estructurado en bloques de código que se almacenan como objetos y los mismos permiten a los desarrolladores, crear paquetes y scripts de código, de manera mucho más rápida y efectiva ya que es un lenguaje de programación, el cual, viene incrustado en la base de datos de Oracle.

“Oracle PL SQL (*procedural language structured query language*), es un lenguaje de procedimiento o de programación, desarrollado por Oracle con la función de una extensión de SQL. Permite la combinación de consultas de SQL y las instrucciones de procedimientos (condicional y procesamiento iterativo) en función de crear un tratamiento de cierta complejidad y ser almacenados en la base de datos.”<sup>23</sup>

El bloque de PL SQL consta de una estructura conformada por código que necesita cumplir una serie de premisas que deben de cumplirse para su desarrollo.

---

<sup>23</sup> VICENCIO, Roberto. *Descubre qué es PL SQL y cómo se debe utilizar en Oracle. Cómo crear software de calidad.* p. 5.

Las variables que se usan requieren declararse en la sección Declare. En función de declarar las variables y las constantes en la sección respectiva. Este lenguaje utiliza distintos tipos de datos estándar en Oracle SQL:

- Number: números en los que p es la precisión y es opcional y significa la escala de decimales.
- Varchar2, límite, sucesión de caracteres de longitud variable, límite que significa el número máximo de Caracteres.
- Date, fecha que puede contener día, mes, año, hora, minutos y segundos. Estos tipos de datos son soportados en: tipos de datos en Oracle 11g.

Además, posee las características de realizar declaraciones de variables, utiliza como origen la declaración de:

- Bloques PL SQL
- Tipos de bloques
- Tipos de datos Oracle PL SQL
- Identificadores
- Operadores
- Variables
- Constantes
- Tipos de datos de una tabla
- Tablas anidadas

El lenguaje de Oracle PL SQL, como tal, no admite cambiar la estructura de la base de datos tipo DML. No obstante, puede utilizar instrucciones DML. Además, permite realizar:

- Gestión de errores que se generen durante la ejecución del proceso.
- Crear procedimientos y funciones.
- Maneja sintaxis que puede visualizar posibles errores de otra índole.
- Compilación de forma automática.
- Maneja paquetes que son una colección integrada de procedimientos y funciones y otros objetos que se encuentran recopilados en la base de datos.
- Ejecuta uno o más procedimientos después de un evento, mediante triggers o disparadores, los cuales se aplican a una base de datos.
- Los objetos y tipos de datos especiales que maneja se han desarrollado para enfrentar los problemas que plantean las bases de datos relacionales, en función de almacenar datos de gran tamaño, como en el caso de ficheros, imágenes, videos, entre otros.
- Maneja tipos de datos LOB externos que crean objetos, que pueden utilizarse también en las tablas como types, entre otras funciones.

Este lenguaje es una estructura compuesta que requiere cumplir una serie de procedimientos, dado sus características relacionales de combinar consultas

SQL, aunados a las instrucciones de procedimientos para almacenar en bases de datos.

Este programa da acceso a los desarrolladores a crear bloques PL SQL, almacenarlos, mediante paquetes que son ejecutados en el servidor, lo cual permite el ahorro de recursos en el contexto y pueden tener acceso únicamente los usuarios que tienen la debida autorización.



### **3. DESCRIPCIÓN DE ESCENARIO INICIAL**

El presente estudio se desarrolla en el Ministerio de Salud Pública y Asistencia social de Guatemala, MSPAS, encargado de la prestación y cobertura de servicios de salud a la población guatemalteca en todo su territorio provee atención de calidad, bajo la premisa de la universalización de la salud, es decir, salud para todos.

El referido estudio se lleva a cabo específicamente en el Sistema de Información Gerencial de Salud, SIGSA, la unidad del Ministerio de Salud, encargada de recolectar y consolidar la información que maneja este ministerio acerca de sus proyectos, para analizar y desarrollar esquemas de información, a través de mecanismos tecnológicos informáticos, con el fin de procesar datos para la provisión continua de la información de calidad del Sistema de Salud Nacional de Guatemala y que la misma esté y disponible en el momento en que se necesite consultar.

#### **3.1. Contexto**

“El Ministerio de Salud Pública y Asistencia Social de Guatemala (MSPAS) tiene a su cargo la función de gestionar, fortalecer, mejorar y ampliar la cobertura de atención en salud; se fundamenta en la prestación de servicios, de forma integral e integrada, con el fin de promover salud de calidad, conforme las políticas establecidas de acuerdo con los lineamientos de la política económica social y de seguridad 2012-2020.

En la implementación de acciones tales como:

- Prevención de enfermedades.
- Fortalecimiento de la infraestructura existente.

- Promoción de cobertura de la red hospitalaria y su extensión en los municipios prioritarios, alrededor del país.
- Dotación de equipo e insumos para los diversos nosocomios y centros de atención hospitalaria y de salud.
- Fortalecimiento de los programas de vigilancia en salud.
- Promoción y fortalecimiento de acciones que garanticen la accesibilidad a medicamentos.
- Reconocimiento al uso práctico de medicina alternativa y tradicional cultura.
- Implementación de sistemas de información.
- Promoción de la creación de proyectos de investigación y desarrollo tecnológico en salud.
- Mejoramiento del financiamiento y de la calidad del gasto en salud pública.
- Promoción de proyectos y programas de vigilancia y control de la aplicación de las normativas vigentes en materia de salud, saneamiento ambiental e higiene, con el fin de mejorar la calidad de vida de toda la población del país y responder así a la demanda de servicios de salud, conforme los programas de solidaridad y equidad social”<sup>24</sup>.

“El desarrollo del proceso de planificación implementación y ejecución de programas y proyectos para el funcionamiento y gestión del Ministerio de Salud Pública y Asistencia Social de Guatemala en su accionar, con el fin de brindar atención de calidad en la prestación de servicios de salud a la población guatemalteca en general, en todo su contexto, a nivel nacional, en cuanto a la universalización de la salud; se instituye siete ejes alrededor de los cuales gira la ejecución del referido plan estratégico:

- Fortalecimiento de los niveles de atención e integración de redes.
- Reforma al marco regulatorio del sector salud.
- Formación, desarrollo de recursos humanos e investigación.
- Gobernanza.
- Implementación del proceso de regionalización de la gestión técnica y financiera.
- Fortalecimiento al acceso de agua sanitariamente segura.
- Garantía de calidad.<sup>25</sup>

<sup>24</sup> PINEDA, B. *Sector Salud en Guatemala. Información, metodología, procesos y procedimientos. Políticas públicas en salud.* p. 42.

<sup>25</sup> Ministerio de Salud y Asistencia Social de Guatemala. *Plan estratégico 2014-2019. Proyecto de políticas en salud y educación con el apoyo de la agencia de los Estados Unidos para el desarrollo internacional bajo el acuerdo cooperativo no. AID-OAA-A-10-00067.* p. 7.



Cada eje referido, es definido por objetivos estratégicos, operativos, indicadores y metas que deben desarrollarse y ejecutarse durante el período 2014 a 2019, en el cual toman parte las distintas entidades (ministerio y viceministerios), consejos, direcciones, coordinaciones, departamentos, unidades, áreas, oficinas, sistemas y programas de la Red Integrada de Servicios de Salud de Guatemala, entre la cual se encuentra el Sistema de Información Gerencial de Salud, SIGSA.

El Sistema de Información Gerencial de Salud, SIGSA, dentro de sus funciones, se encarga del suministro de la demanda de datos actualizados y oportunos y de la provisión de información de calidad, disponible, sobre la función rectora del Ministerio de Salud Pública y Asistencia Social de Guatemala, en lo que respecta a:

- Problemas existentes en el sistema de salud.
- Gestión y fortalecimiento de eventos endémicos y epidémicos del servicio de salud.
- Seguimiento de intervenciones.
- Evaluación de resultados de programación.
- Control y toma de decisiones sobre razonamientos epidemiológicos y administrativos.
- Producción y cobertura de servicios.
- Plan operativo anual.

- Normas de atención y fuentes de formación exterior, entre otros.

El Sistema de Información Gerencial de Salud, SIGSA, se encuentra conformado por módulos de estadística con acceso a datos acerca de los recursos de salud y de la producción y cobertura de servicios, así como de la ejecución de proyectos que se relacionan con mortalidad, morbilidad, elaboración de la captura de información y presentación de reportes y gráficas, su divulgación e interpretación respectiva y adecuada.

Además, el SIGSA colecta y consolida la información que es usada por el Ministerio de Salud Pública y Asistencia Social, MSPAS, la cual es presentada a través de formularios estadísticos y que se maneja, mediante el análisis, transmisión y desarrollo de esquemas de información; se establece para ello mecanismos de incorporación de tecnología informática y de procesamiento de datos para el mejoramiento continuo del proceso, provisión y disponibilidad de información del sistema de salud nacional de Guatemala.

### **3.2. Estructura de desarrollo en JavaScript para futuros proyectos en el Ministerio de Salud, departamento de SIGSA**

Con la implementación del presente estudio, se pretende la creación y diseño de una estructura basada en una plantilla, a través de la cual se pueda desplegar un esquema específico establecido, en los procesos y sistemas informáticos, para el desarrollo de próximos planes por parte del departamento del Sistema Gerencial de Salud SIGSA, al momento de utilizar nuevamente lenguaje JavaScript en lo que respecta a futuros proyectos.

Dicha estructura busca ser creada valiéndose de la herramienta denominada Google Closure, la cual tiene como función principal asistir a los

desarrolladores en la optimización de código estático, con el objetivo de mejorar la estructura en el código en lenguaje JavaScript.

Para la estructura de desarrollo en JavaScript, con el uso de la herramienta Google Closure, de los proyectos existentes y futuros en el Ministerio de Salud en el área de SIGSA, se implementó un ambiente sistematizado consistente en distintas fases a describir.

Además, fue necesaria la creación de una *Guía de instalación Closure Compiler*, la cual se desarrolló con la utilización de la herramienta Google Closure ya referida.

### **3.3. Herramientas de apoyo para el desarrollo del estudio**

#### **3.3.1. MySQL**

MySQL es conocido comúnmente como una herramienta de apoyo que se fundamenta en un sistema administrador de bases de datos de código abierto y que se basa en el lenguaje SQL; se ejecuta en casi todas las plataformas, especialmente, Linux, UNIX y Windows.

“MySQL es un sistema gestos de base de datos de índole relacional (RDMBS) que maneja código abierto y que se basa en el lenguaje estructurado SQL. Dicha herramienta se origina inicialmente en la compañía sueca MySQL, AB y posteriormente en 2008, la adquirió Oracle. Los derivados de MySQL son denominados FORKS que incluyen lo siguiente:

- Un sistema de gestión de base de datos ligero de código abierto en el desarrollo que se basa en MySQL 6,0 que es llamado Drizzle.
- Un reemplazo público tipificado como drop-in, desarrollado en la comunidad para MySQL, que es usa las API y los comandos de MySQL que es llamado MariaDB.

- Una versión mejorada de MySQL, reconocida por su escalabilidad horizontal, que es denominada Percona Server con XtraDB<sup>26</sup>.

Esta herramienta administradora es desarrollada en todas las plataformas y puede ser usada en variedad de aplicaciones, se relaciona con las aplicaciones afirmadas en la Web y lo que se refiere a la publicación en línea.

MySQL es una herramienta que, además, representa un elemento muy importante de código abierto denominado LAMP, que está conformado de forma complementaria de la siguiente manera:

- En una plataforma de desarrollo Web usada por LINUX como un sistema operativo.
- En un servidor Web denominado, Apache.
- En el sistema de gestión de base de datos relacional, MySQL.
- En el lenguaje de programación PHP, en el cual se orienta a objetos.

### **3.3.2. Oracle**

Oracle es una herramienta diseñada fundamentalmente para que funcione como gestor de base de datos de gran potencial, de elevado costo; que es utilizado comúnmente a nivel empresarial-internacional.

“Oracle es una herramienta desarrollada con la función de administrar la gestión de bases de datos, creada el 1977, bajo el sistema de gestores de base de datos de Koch George. Este potente desarrollador es usado por grandes

---

<sup>26</sup> ROUSE, Margaret. *¿Qué es MySQL? Search Data Center en español. Contribuidores. Eng. P y McCormack. Teach Target. S.A. de C.V. 2005-2016. p. 4.*

empresas, multinacionales, es, además, empleada en la creación de páginas web; considerado uno de los sistemas de bases de datos más completos<sup>27</sup>.

Es también, una herramienta de tipología objeto-relacional (ORDBMS) *object-relational-data-base-management-system*, desarrollado por Oracle Corporation. En esta, se destacan las siguientes características:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Soporte multiplataforma.

Esta herramienta es configurada con base en la tecnología cliente-servidor y necesita de la instalación del servidor Oracle y de la complementación de otras herramientas básicas de programación de desarrollo como:

- Oracle developer: consiste en una herramienta que admite la creación de formularios y, a la vez, su compilación y ejecución. Para ello, es necesario copiar el formulario de forma regular, en una carpeta compartida para los usuarios, de manera que cuando se requiere realizar una modificación, es necesario copiarlo de la carpeta mencionada, realizar el cambio y posteriormente, subirlo de nuevo a la misma carpeta. La característica principal del developer es que permite modificar el formulario, de la misma forma en que se haría en Visual Basic o en Visual Basic C, lo cual representa un beneficio. Esta herramienta se muestra más factible de utilizar a un inicio, sus propiedades son más automatizadas.

---

<sup>27</sup> MASIP, David. *¿Qué es Oracle? Definición de Oracle como herramienta de gestión de bases de datos.* p. 5.

- Oracle designer: consiste en una herramienta complementaria de Oracle developer; con Designer se realiza la conexión con la base de datos y con lo cual se crean sus formularios; de esa manera todos los usuarios quedan conectados a la aplicación que contiene dichos formularios, cuenta con el beneficio de que no existen versiones distintas, lo cual representa una ventaja para evitar desintegrar el trabajo de otros usuarios. La única desventaja que se presenta es la falta de un contexto visual para crear el formulario, se incrustan bloques o módulos que son las estructuras con los elementos del formulario, los cuales pueden aparecer por medio de tablas.
- Además, para el desarrollo de Oracle se utiliza el lenguaje de Quinta (5ta.) Generación, PL/SQL, el cual es muy potencial, en función de realizar la gestión de base de datos es importante mencionar que es una norma general la utilización de SQL, en función de la creación del formulario.
- La última versión de Oracle Data base, es la 12c, diseñada para Cloud Computing en 2013, desarrollada con el objetivo de facilitar, estandarizar, consolidar y automatizar los servicios de gestión de bases de datos en la nube.

En su propiedad de base de datos, Oracle soporta todas las clases de datos relacionales estandarizados, además, datos nativos como:

- XML
- Texto
- Imágenes
- Documentos

- Audio
- Datos espaciales

La accesibilidad a toda esa información referida, Oracle la desarrolla mediante interfaces estandarizadas entre las cuales se cuentan:

- SQL
- JDBC
- SQLJ
- ODBC.Net
- OLE.Net y
- ODP.Net
- SQL/XML
- XQuery y
- WebDay

En la herramienta de Oracle, los procedimientos archivados pueden estar escritos en lenguajes como Java y PL/SQL. Además, en su actualización va incorporando nuevas capacidades en el desarrollo de aplicaciones, dando acceso a mayor velocidad y mejor desempeño, con el fin de permitir que el software se mantenga en constante transformación con el objetivo de mejorar la recopilación, administración y seguridad de los datos, a través de una plataforma de alto nivel de desempeño que ofrece soporte de las tecnologías principales, que en la actualidad se utilizan por quienes se dedican al desarrollo de aplicaciones.

### 3.3.3. Maven

Maven es una herramienta muy utilizada y versátil en el desarrollo de software, dado sus características de simplicidad.

“La herramienta Maven ofrece una interfaz de índole común para desarrollar gestión de proyectos de software, lo cual se realiza de forma independiente de sus módulos, dependencias o librerías y lo cual ejecuta con el simple hecho de ejecutar el comando `mvn install`”<sup>28</sup>.

Maven tiene la capacidad de hacer más simple el proceso de build del código; dando acceso a la compilación de cualquier tipo de proyecto, de la misma manera, cualquier tipo de dificultad en el desarrollo de su proceso. Este ejecuta la gestión de un proyecto software de manera completa, lo cual hace desde la etapa en la cual hace la comprobación acerca de la certidumbre del código hasta la etapa en la cual se realiza el despliegue de la aplicación; realiza, en el ínterin, la ejecución de pruebas y la generación de los informes y de la documentación.

Esta herramienta realiza la construcción o build del software, se fundamenta en patrones y en estándares. Lo cual da acceso a los desarrolladores de moverse a través de los proyectos y no requiere de aprender previamente como compilar o empaquetar. Maven despliega su proceso a través de varias etapas:

- Validación: en función de validar que el proyecto sea correcto.
- Compilación: realizando la acción de compilar.

---

<sup>28</sup> GARCIA OTERINO, Ana M. del Carmen. *¿Por qué Maven?* p. 6.



- Pasar test: comprobación del código fuente, a través del uso de un framework de pruebas unitarias.
- Empaquetar: empaquetar el código que ya ha sido compilado y luego modificarlo en algún formato de tipo .jar o war.
- Pruebas de integración: procesamiento y despliegue del código en algún contexto en el cual puedan ejecutarse las pruebas de integración.
- Verificación: comprobar que el código empaquetado sea válido y que cumpla con los criterios de calidad establecidos.
- Instalación: instalar el código empaquetado en el repositorio local de Maven, para ser utilizado como dependencia de otros proyectos.
- Despliegue: desarrollar el código en un contexto.

Estas etapas en referencia se llevan a cabo en el código; es necesario únicamente, al ejecutar mvn y el nombre de la etapa. De esta forma y en cadena, Maven ejecuta etapa por etapa. En lo que respecta a la gestión de dependencias, entre módulos y otras versiones de librería, solo se indican los módulos del proyecto que utiliza el software que se está desplegando, en un fichero de configuración Maven del proyecto, denominado POM. (*project object model*).

Además, Maven tiene un repositorio remoto en el cual se ubican la mayoría de librerías, usadas en el desarrollo de software y que esta herramienta descarga conforme se requiere. Tal herramienta ofrece una semántica común al proceso de build y de desarrollo del software; establece,

además, una estructura común de directorios que es funcional para todos los proyectos que se desarrollan en esta herramienta.

El paquete Maven es muy conocido como una herramienta de JAVA, por lo cual se requiere de tener instalado el lenguaje Java, en función de proceder. Maven, necesita ser descargado del servidor denominado: MAVEN APACHE, (<http://maven.apache.org/download.html#installation>) para su instalación, siguiendo las instrucciones que se indican y luego de su descarga, debe ser comprimido en el ordenador respectivo.

Es necesario puntualizar que en lo respectivo a la creación de proyectos y arquetipos para Maven, una plantilla tiene la capacidad de concebir una estructura de directorios y de ficheros. A partir de ahí se puede empezar un proyecto que indica a Maven la clase de proyecto que se quiere trabajar y esta herramienta desarrollará las estructuras básicas, que frecuentemente se despliegan en o con proyectos Java. En el caso de que se use Windows, es necesario revisar los requisitos de Windows, en función de usar sin problema la herramienta Maven en Windows.

Por todo lo anterior, se infiere que Maven es una herramienta muy útil y funcional con la característica esencial de administrar, agilizar y estandarizar el trabajo en equipo.

#### **3.3.4. IDE Eclipse Neon**

IDE Eclipse es una herramienta de programación que se ocupa de desarrollar y compilar aplicaciones de Java, posee una interfaz que le permite un manejo fácil y agradable en su proceso. Esta herramienta de interfaz se ha

desarrollado completamente en Java, por lo cual se hace necesario tener instalado Java para poder ejecutarlo.

Los alcances de IDE Eclipse son muy amplios dado que, no obstante, el idioma que el usuario seleccione, el editor de esta plataforma, ofrece herramientas de índole universal y de beneficio gratuito.

Además, para su conocimiento y desarrollo, se encuentran disponibles diversos tutoriales sobre IDE Eclipse que son de fácil entendimiento y manejo, aún para principiantes.

“Es un interfaz de desarrollo integrado de Código abierto y multiplataforma, que, generalmente, se usa para desarrollar aplicaciones basados en navegadores. Es una plataforma muy potente de programación, de desarrollo y de compilación, de elementos y sitios web diversos, de programas en C++, subproyecto de Cobol, desarrollo de herramientas PHP, puede dar soporte a compiladores de lenguajes dinámicos TCL, Ruby, Javascript y Python; así como de aplicaciones en Java; en el cual se encuentran todas las herramientas y funciones que se requieren para trabajar en una interfaz muy interesante que le otorga un manejo factible y atractivo<sup>29</sup>.

Esta plataforma se desarrolló inicialmente por IBM, Canadá en 2001; en la actualidad, lo desarrolla la fundación Eclipse con base en un código abierto y es complementada con un grupo de elementos, servicios y capacidades. Ha sido muy utilizada para entornos de desarrollo integrados de donde vienen sus siglas IDE, esta se encuentra disponible en diversos idiomas.

Es conocida como plataforma de cliente enriquecido, dado que es conformada por diversos componentes:

---

<sup>29</sup> ECURED, La Revista. Conocimiento con todos y para todos. *Entorno de desarrollo integrado (IDE)*. p. 8.

- Plataforma principal: contiene; inicio de eclipse, ejecución de plugins, OSGI, plataforma para bundling estándar, estándar widget toolkit (SWT), widget toolkit portable, face, manejo de archivos, de texto y de editores de texto, workbench de eclipse, vistas, editores, perspectivas y asistentes.
- Posee características como las siguientes: editor de texto con resaltado de sintaxis para visualizar el contenido del fichero en el que se está trabajando, contiene una lista de tareas y otros módulos semejantes, compila en tiempo real, tiene pruebas unitarias con JUnit, posee integración con Ant, asistentes wizards para la creación de proyectos, clases, tests y refactorización.

Además, IDE Eclipse posee características generales de programa que pueden ser ampliadas y mejoradas con el uso de plug-ins, los cuales se encuentran disponibles en dicha herramienta y sus componentes. Aunado a ello, los widgets que utiliza se encuentran implementados por una herramienta de Java (SWT). La interfaz de usuario posee una capa BUI intermedia denominada JFace que simplifica la construcción de aplicaciones que se basan en SWT.

La definición que otorga el proyecto Eclipse sobre su software hace alusión a una herramienta de carácter universal, dado que se trata de un IDE abierto y extensible para todas sus funciones. Esta herramienta provee a su usuario textualmente lo siguiente:

- Provee frameworks: donde sobreabundan las aplicaciones gráficas, definición y manipulación de modelos de software y aplicaciones Web.

- El SDK: incluye herramientas de desarrollo Java que posee un IDE con compilador de Java interno y un modelo completo de los archivos fuente de Java, lo que permite técnicas avanzadas de refactorización y análisis de código.
- El ID: además utiliza un espacio de trabajo con grupo metadata en un espacio para archivos planos, lo cual da acceso a cambios de índole externo en los archivos, en tanto se libere el espacio de trabajo respectivo.

El proyecto IDE Eclipse permite a los desarrolladores construir herramientas de pruebas y de desempeño como depuradores, profilers y aplicaciones Benchmark. Extiende su plataforma para el desarrollo de aplicaciones Web en Java EE; se encuentra conformada de editores de fuente para: HTML, JavaScript, CSS, JSP, SQL, XML, DTD, XSD y WSDL y además cuenta con una plataforma de desarrollo de software para dispositivos.

Otra de sus funciones fundamentales es proveer las herramientas que implementan un IDE de Java, además, soporta el desarrollo de cualquier aplicación Java.

Actualmente, se han desarrollado varias versiones de esta herramienta que se encuentra disponible en varios idiomas, con el fin de mejorar su función y a su vez ofrece técnicas mejoradas y más avanzadas con el propósito de innovar su funcionamiento y manejo tecnológico y programático.

### **3.4. Descripción de proyectos para analizar**

En el Ministerio de Salud Pública y Asistencia Social de Guatemala, existen diversos proyectos desplegados dentro del departamento de SIGSA, los cuales son desarrollados principalmente en lenguaje PHP y que cuentan con contenido de JavaScript. Este código nunca ha sido analizado, utilizando una herramienta de análisis en la búsqueda de una optimización.

En consecuencia, se pretende realizar un breve análisis acerca de la funcionalidad de cada proyecto existente, utilizando la herramienta diseñada y desarrollada para ser implementada con el objetivo primordial de optimizar el código y, además, mejorar la estructura y la ejecución con la cual fueron creados los mismos.

#### **3.4.1. Proyecto, atención al ciudadano**

Hace referencia al sistema desarrollado y utilizado por el Ministerio de Salud Pública y Asistencia Social, para la captura y seguimiento de las distintas denuncias que son presentadas por el usuario.

#### **3.4.2. Proyecto codificaciones y defunciones**

Se relaciona con el sistema que automatiza el proceso de codificación de las causas de muerte según CIE 10.

El sistema establece una estructura de base de datos que permite el vaciado de los archivos con información sobre las defunciones que RENAP proporciona actualmente al SIGSA.

Tales registros se muestran en pantalla al usuario, quien procederá a analizar cada caso y a realizar la misma codificación de causas hacia códigos CIE 10 correspondientes a su distrito, ahora desde la aplicación.

### **3.4.3. Proyecto, cooperación internacional**

“La política de cooperación internacional es parte de la política exterior del Gobierno que fija los criterios y lineamientos para el apoyo que la comunidad internacional brinda para el cumplimiento de los Acuerdos de Paz, la estrategia de pobreza y las políticas de Gobierno 2004-2008. El objetivo de esta política es establecer lineamientos políticos y operativos que permitan orientar la acción de los distintos actores del proceso de la gestión de la cooperación internacional, así como alcanzar un trabajo coordinado.”<sup>30</sup>

### **3.4.4. Proyecto, DRACES**

Proyecto que aún no ha salido a producción, con el cual se tiene planificado llevar un control de las licencias sanitarias las cuales son brindadas por el departamento de regulación, acreditación y control de establecimientos de salud.

### **3.4.5. Proyecto, encuestas servicios atención**

Este proyecto consiste en un pequeño sistema a través del cual se captura información cuando es necesario llevar a cabo la realización de determinados tipos de encuestas. Está orientado a la posibilidad de desarrollarse de manera dinámica y práctica.

---

<sup>30</sup> Cooperación Internacional. *Acuerdos de Paz, la estrategia de pobreza y las políticas de Gobierno 2004-2008*. <http://www.mspas.gob.gt/index.php/en/institucional/cooperacion-internacional.html>. Consulta: 4 de abril de 2017.

#### **3.4.6. Proyecto, Epiweb**

Consiste en un sistema bastante amplio, el cual permite la captura de enorme cantidad de información de enfermedades epidemiológicas, para lo cual es necesario llevar un epidemiario con registros continuos de consultas y a su vez generar reportes de diversidad de fichas.

#### **3.4.7. Proyecto, Incan Cáncer**

Se trata de un sistema proporcionado al Instituto de Cancerología para que se pueda llevar y tener un mejor control y registro de cáncer. Además, crea un exportable utilizado por la misma institución en conjunto con otro sistema acerca de análisis de tumores.

#### **3.4.8. Proyecto, InfraUPE**

Proyecto para apoyar la estructura y funciones de la unidad de planificación estratégica.

#### **3.4.9. Proyecto, mantenimiento red de servicios**

Es un sistema creado con la función de administrar la red que se maneja Institucionalmente de manera interna en el Ministerio de Salud Pública y Asistencia Social. En el cual, por lo general hay un solo usuario, quien es el encargado de su uso.



### **3.4.10. Proyecto, monitoreo de servicios**

Consiste en un sistema desarrollado y utilizado para llevar un control de la ubicación georeferencial de todos los servicios de salud a nivel de todo el país.

### **3.4.11. Proyecto, UNIP**

Proyecto aun en progreso, el cual busca la creación de un software para brindar asistencia a la unidad de ventanilla del Ministerio de Salud.

## **3.5. Resultados previos en la utilización de herramienta**

A continuación, se presentan los resultados previos en la utilización de la herramienta utilizada.

Aquellas casillas sin ningún valor, es debido a que no se utilizan los referidos archivos, dentro del proyecto.

### **3.5.1. Cantidad de código**

Tabla I. **Proyecto, cooperación internacional**

<b>Cooperación Internacional</b>	<b>Núm. de líneas</b>	<b>Cantidad de Caracteres</b>
validar_Ingreso.js	41	883
validar_Localizacion.js	76	1925
validar_Proyecto.js	30	566

Fuente: elaboración propia.

Tabla II. **Proyecto, monitoreo servicios**

Monitoreo servicios	Núm. de líneas	Cantidad de caracteres
estructuras.js	313	8066
modificarUsuario.js	63	1469
recurso_humano.js	72	1667
servicio_salud_inf_general.js	183	3998
servicio_salud_mapas.js	127	3668
servicios_basicos.js	186	4122
validarCabecera.js	29	611
validarCuerpo.js	80	2005
validarFecha.js	19	373
validarUser.js	44	926

Fuente: elaboración propia.

Tabla III. **Proyecto, epiduario**

Epiduario	Núm. de líneas	Cantidad de caracteres
CatalogoLugar.js	49	878
validarAcciones.js	32	611
validarCabecera.js	29	611
validarCuerpo.js	80	2005
validarFecha.js	19	373
validarNuevoUsuario.js	70	1802
validarUser.js	68	1441

Fuente: elaboración propia.

Tabla IV. **Proyecto Epiweb**

Epiweb	Núm. de líneas	Cantidad de caracteres
aviso.js	6	69
CatalogoLugar.js	49	878
validarAcciones.js	186	5006
validarCabecera.js	39	826
validarCuerpo.js	124	3436
validarFecha.js	31	629
validarNuevoUsuario.js	70	1802
validarUser.js	68	1441
fecha.js	250	5740

Continuación de la tabla IV.

fecha_des.js	351	7742
fecha_mat.js	359	8178
fecha_mhd.js	239	5219
fecha_mhd.js	239	5219
fecha_normal.js	332	7291
fecha_pfa.js	354	8008
fecha_tos.js	351	7742
mostrar_institucion.js	67	1678
validarAcciones.js	331	8695
validarAcciones_dcz.js	913	29494
validarAcciones_des.js	300	7506
validarAcciones_inf.js	2674	98129
validarAcciones_mal.js	738	18556
validarAcciones_mat.js	1162	32869
validarAcciones_mda.js	466	10567
validarAcciones_mhd.js	365	8728
validarAcciones_pfa.js	1145	29808
validarAcciones_pla.js	498	12237
validarAcciones_rab.js	322	7165
validarAcciones_rot.js	907	27686
validarAcciones_sar.js	998	26380
validarAcciones_tos.js	956	23794
validarAcciones_vih.js	330	8939
validarCampos.js	398	12923
validarCampos_dcz.js	367	15691
validarCampos_des.js	306	8573
validarCampos_inf.js	867	31411
validarCampos_mal.js	627	18132
validarCampos_mat.js	1163	36246
validarCampos_mda.js	674	19887
validarCampos_mhd.js	482	13777
validarCampos_pfa.js	595	17062
validarCampos_pla.js	447	12839
validarCampos_rab.js	403	11145
validarCampos_rot.js	640	18001
validarCampos_sar.js	629	18139
validarCampos_tos.js	587	17107
validarCampos_vih.js	273	9060

Fuente: elaboración propia.

Tabla V. **Proyecto, usuarios**

<b>Usuarios</b>	<b>Núm. de líneas</b>	<b>Cantidad de caracteres</b>
validarEst.js	74	1551
validarUser.js	6	6733

Fuente: elaboración propia.

Tabla VI. **Proyecto, fronteras**

<b>Fronteras</b>	<b>Núm. de líneas</b>	<b>Cantidad de caracteres</b>
CatalogoLugar.js	49	878
validarAcciones.js	32	611
validarCabecera.js	29	611
validarCuerpo.js	80	2005
validarFecha.js	19	373
validarNuevoUsuario.js	70	1802
validarUser.js	68	1441

Fuente: elaboración propia.

Tabla VII. **Proyecto INCAN-Cáncer**

<b>Incan-Cáncer</b>	<b>Núm. de líneas</b>	<b>Cantidad de caracteres</b>
CambioClave.js	47	1263
dataTable.js	7	111
funciones.js	91	2763
inicio.js	183	7176
mantenimiento.js	182	7574
registroCaso_eventos.js	304	10498
registroEncabezado.js	23	386
registroEncabezado_eventos.js	161	4823
registroTumor_eventos	239	9264
usuarios.js	234	8632
verificaTumor.js	133	3977
_inserEncabezado.js	55	1991
_insertRegistroCaso.js	1350	6734
_insertTumor.js	81	3790
_insertVerificaTumor.js	77	3423

Fuente: elaboración propia.

Tabla VIII. **Proyecto InfraUPE**

<b>infraUPE</b>	<b>Núm. de líneas</b>	<b>Cantidad de caracteres</b>
estructuras.js	315	8070
modificarUsuario.js	64	1471
recurso_humano.js	74	1671
servicio_salud_inf_general.js	185	4002
servicio_salud_mapas.js	130	3681
servicios_basicos.js	186	4122
validarCabecera.js	31	615
validarCuerpo.js	82	2009
validarFecha.js	19	373
validarUser.js	44	926

Fuente: elaboración propia.

### **3.5.2. Calidad de código a optimizar**

Para realizar el análisis del código fue necesario detectar donde es utilizado el archivo JavaScript dentro del proyecto y utilizar las herramientas de desarrollo del navegador de Google Chrome para detectar sus tiempos de respuesta.

En función de esta, se tomaron tres parámetros en cuenta para reconocer su eficiencia:

- RequestSent: referente al tiempo dedicado a la emisión de la solicitud de red.
- Waiting: referente al tiempo de espera para la respuesta inicial, también conocido como el tiempo hasta el primer byte.
- En este caso realiza la captura de la latencia de ida y vuelta al servidor, además del tiempo de espera para el servidor para entregar la respuesta.

- Content Download: referente al tiempo dedicado a la recepción de los datos de respuesta.

Tabla IX. **Proyecto, cooperación internacional**

Cooperación Internacional	Página ingresada en aplicación para capturar el archivo	Request sent(ms)	Waiting (ms)	Content Download (ms)
validar_Ingreso.js	Cooperantes- NuevosCooperantes	0,2	3,47	0,9
validar_Localizacion.js	Proyectos-Lugares de Intervención	0,21	2,04	0,97
validar_Proyecto.js	Proyectos-Ingreso Proyectos	0,19	5,96	1,04

Fuente: elaboración propia.

Tabla X. **Proyecto, monitoreo de servicios**

Monitoreo Servicios	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
Estructuras.js				
modificarUsuario.js	modificarusuario.php	0,39	1,13	0,96
recurso_humano.js	Modificaciones-Estados Comentarios	0	3,7	1,64
servicio_salud_inf_general.js				
servicio_salud_mapas.js				
servicios_basicos.js				
validarCabecera.js				
validarCuerpo.js				
validarFecha.js				
validarUser.js	Usuario.php	0,2	36,72	1,21

Fuente: elaboración propia.

Tabla XI. Proyecto, Epi diario

Epidiario	Página ingresada en aplicación para capturar el archivo	Request sent(ms)	Waiting (ms)	Content Download (ms)
CatalogoLugar.js	catalogos.php	0,23	16,58	0,94
validarAcciones.js	guardar_ficha.php	0,13	47,54	1,22
validarCabecera.js	guardar_ficha.php	0,17	29,83	1,01
validarCuerpo.js	guardar_ficha.php	0,3	25,32	0,83
validarFecha.js	variables.php	0,14	84,49	1,08
validarNuevoUsuario.js		No se encontró lugar donde es utilizado		
validarUser.js		No se encontró lugar donde es utilizado		

Fuente: elaboración propia.

Tabla XII. Proyecto, Epiweb

Epiweb	Página ingresada en Aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
aviso.js	guardar_ficha.php	0,12	10,92	1,32
catalogoLugar.js	catalogos.php	0,23	16,58	0,94
validarAcciones.js	guardar_ficha.php	0,13	47,54	1,22
validarCabecera.js	guardar_ficha.php	0,17	29,83	1,01
validarCuerpo.js	guardar_ficha.php	0,3	25,32	0,83
validarFecha.js	variables.php	0,14	84,49	1,08
validarNuevoUsuario.js		No se encontró lugar donde es utilizado		
validarUser.js		No se encontró lugar donde es utilizado		
fecha.js	\\epiweb\fichas\paginas\des\desnutricion_ingresar.php	0,12	46,94	1,36
fecha_des.js				
fecha_mat.js	\\epiweb\fichas\paginas\mat\materna_ingresar.php	0,12	43,36	1,06
fecha_mda.js	\\epiweb\fichas\paginas\mda\MdesnutricionA_ingresar.php	0,14	5,98	1,29
fecha_mhd.js	\\epiweb\fichas\paginas\mhd\MhospitalariaD_ingresar.php	0,15	7,7	1,45
fecha_normal.js	\\epiweb\fichas\paginas\chik\chikungunya_ingresar.php	0,14	5,73	1,36
fecha_pfa.js	\\epiweb\fichas\paginas\pfa\paralisis_ingresar.php	0,14	8,1	1,5
fecha_tos.js	\\epiweb\fichas\paginas\sar\sarampion_ingresar.php	0,13	6,74	3,26
mostrar_institucion.js	\\epiweb\fichas\paginas\des\desnutricion_ingresar.php	0,19	47,54	1,71
validarAcciones.js	\\epiweb\fichas\paginas\chik\chikungunya_ingresar.php	0,18	5,23	0,92
validarAcciones_dcj.js	\\epiweb\fichas\paginas\dcz\dcz_ingresar.php	0,18	6,75	2,11

Continuación de la tabla XII.

validarAcciones_des.js	\\epiweb\\fichas\\paginas\\des\\desnutricion_ingresar.php	0,15	5,17	2,18
validarAcciones_inf.js	\\epiweb\\fichas\\paginas\\inf\\bk\\influenza_ingresar.php	0,19	6,9	9,12
validarAcciones_mal.js	\\epiweb\\fichas\\paginas\\mal\\malaria_ingresar.php	0,17	13,51	1,02
validarAcciones_mat.js	\\epiweb\\fichas\\paginas\\mat\\materna_ingresar.php	0,17	5,99	1,03
validarAcciones_mda.js	\\epiweb\\fichas\\paginas\\mda\\MdesnutricionA_ingresar.php	0,19	5,27	0,62
validarAcciones_mhd.js	\\epiweb\\fichas\\paginas\\mhd\\MhospitalariaD_ingresar.php	0,2	5,07	2,43
validarAcciones_pfa.js	\\epiweb\\fichas\\paginas\\pfa\\paralisis_ingresar.php	0,2	6,34	1,82
validarAcciones_pla.js	\\epiweb\\fichas\\paginas\\pla\\plaguicida_ingresar.php	0,19	5,01	0,95
validarAcciones_rab.js	\\epiweb\\fichas\\paginas\\rab\\rabia_ingresar.php	0,2	4,91	1,34
validarAcciones_rot.js	\\epiweb\\fichas\\paginas\\rot\\rotavirus_ingresar.php	0,19	4,99	1,37
validarAcciones_sar.js	\\epiweb\\fichas\\paginas\\sar\\sarampion_ingresar.php	0,18	6,04	1,54
validarAcciones_tos.js	\\epiweb\\fichas\\paginas\\tos\\tos_ingresar.php	0,16	7,37	1,07
validarAcciones_vih.js		No se encontró lugar donde es utilizado		
validarCampos.js	\\epiweb\\fichas\\paginas\\chik\\chikungunya_ingresar.php	0,19	5,35	1,89
validarCampos_dcj.js	\\epiweb\\fichas\\paginas\\dcj\\dcj_ingresar.php	0,16	7,99	55,91
validarCampos_des.js	\\epiweb\\fichas\\paginas\\des\\desnutricion_ingresar.php	0,14	5,62	1,79
validarCampos_inf.js	\\epiweb\\fichas\\paginas\\inf\\bk\\influenza_ingresar.php	0,13	6,06	2,85
validarCampos_mal.js	\\epiweb\\fichas\\paginas\\mal\\malaria_ingresar.php	0,18	21,68	1,86
validarCampos_mat.js	\\epiweb\\fichas\\paginas\\mat\\materna_ingresar.php	0,14	5,46	1,2
validarCampos_mda.js	\\epiweb\\fichas\\paginas\\mda\\MdesnutricionA_ingresar.php	0,17	5,04	0,64
validarCampos_mhd.js	\\epiweb\\fichas\\paginas\\mhd\\MhospitalariaD_ingresar.php	0,24	7,46	1,69
validarCampos_pfa.js	\\epiweb\\fichas\\paginas\\pfa\\paralisis_ingresar.php	0,16	7,1	3
validarCampos_pla.js	\\epiweb\\fichas\\paginas\\pla\\plaguicida_ingresar.php	0,16	24,74	1,24
validarCampos_rab.js	\\epiweb\\fichas\\paginas\\rab\\rabia_ingresar.php	0,15	5,47	49,37
validarCampos_rot.js	\\epiweb\\fichas\\paginas\\rot\\rotavirus_ingresar.php	0,17	5,65	2,25
validarCampos_sar.js	\\epiweb\\fichas\\paginas\\sar\\sarampion_ingresar.php	0,11	8,7	1,65
validarCampos_tos.js	\\epiweb\\fichas\\paginas\\tos\\tos_ingresar.php	0,17	5,46	1,29
validarCampos_vih.js		No se encontró lugar donde es utilizado		

Fuente: elaboración propia.

Tabla XIII. Proyecto, usuarios

Usuarios	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
validarEst.js	\\epiweb\\usuarios\\paginas\\establecimientos.php	0	0,4	1,37
validarUser.js	\\epiweb\\usuarios\\paginas\\existentes.php	0,22	5,47	0,95

Fuente: elaboración propia.



Tabla XIV. Proyecto, fronteras

Fronteras	Página ingresada en Aplicación para capturar el archivo	Request sent(ms)	Waiting (ms)	Content Download (ms)
CatalogoLugar.js	\\epiweb\epidiario\paginas\catalogos.php	0,13	5,28	1,4
validarAcciones.js	\\epiweb\epidiario\paginas\guardar_ficha.php	0,19	6,22	1,38
validarCabecera.js	\\epiweb\epidiario\paginas\guardar_ficha.php	0,16	6,58	1,63
validarCuerpo.js	\\epiweb\epidiario\paginas\guardar_ficha.php	0,21	4,7	1,57
validarFecha.js	\\epiweb\epidiario\paginas\variables.php	0,16	6,52	0,85
validarNuevoUsuario.js				
validarUser.js	\\epiweb\usuarios\paginas\existentes.php	0,18	9,87	0,66

Fuente: elaboración propia.

Tabla XV. Proyecto, INCAN-Cáncer

Incan-Cancer	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
CambioClave.js	\\incancancer\Incan-Cancer\paginas\cambioClave.php	0,2	1,37	0,52
dataTable.js	\\incancancer\Incan-Cancer\paginas\tablaUsuarios.php	0,21	1,43	0,83
funciones.js	\\incancancer\Incan-Cancer\paginas\header.php	0,16	4,13	2,81
inicio.js	\\incancancer\Incan-Cancer\paginas\header.php	0,2	3,88	2,48
mantenimiento.js	\\incancancer\Incan-Cancer\paginas\mantenimientoRegistro.php	0,16	1,46	0,78
registroCaso_eventos.js	\\incancancer\Incan-Cancer\paginas\registroCaso.php	0,15	1,48	0,92
registroEncabezado.js				
registroEncabezado_eventos.js	\\incancancer\Incan-Cancer\paginas\registroEncabezado.php	0,24	1,8	1,87
registroTumor_eventos	\\incancancer\Incan-Cancer\paginas\registroTumor.php	0,19	1,28	0,86
usuarios.js	\\incancancer\Incan-Cancer\paginas\usuarios.php	0,24	2,74	0,83
verificaTumor.js	\\incancancer\Incan-Cancer\paginas\verificaTumor.php	0,45	3,86	2,91
_insertEncabezado.js	\\incancancer\Incan-Cancer\paginas\registroEncabezado.php	0,19	1,46	1,06
_insertRegistroCaso.js	\\incancancer\Incan-Cancer\paginas\registroCaso.php	0,58	2,55	2,56
_insertTumor.js	\\incancancer\Incan-Cancer\paginas\registroTumor.php	0,13	0,9	0,76
_insertVerificaTumor.js	\\incancancer\Incan-Cancer\paginas\verificaTumor.php	0,4	6,42	21,13

Fuente: elaboración propia.

Tabla XVI. **Proyecto, InfraUPE**

InfraUPE	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
estructuras.js	\\infraUPE\infraupe\paginas\Estructuras.php	0,32	44,98	3,2
modificarUsuario.js	\\infraUPE\infraupe\paginas\modificarUsuario.php	0,19	1,31	2,28
recurso_humano.js	\\infraUPE\infraupe\paginas\Recurso_Humano.php	0,48	4,22	2,38
servicio_salud_inf_general.js	\\infraUPE\infraupe\paginas\Servicio_de_Salud.php	0,14	4,64	2,59
servicio_salud_mapas.js	\\infraUPE\infraupe\paginas\Servicio_de_Salud.php	0,23	4,48	1,94
servicios_basicos.js	\\infraUPE\infraupe\paginas\servicios_basicos.php	0,29	1,77	1,62
validarCabecera.js				
validarCuerpo.js				
validarFecha.js				
validarUser.js	\\infraUPE\infraupe\paginas\usuario.php	0,44	3,04	1,29

Fuente: elaboración propia.

### 3.6. Proceso descrito paso a paso del uso de la herramienta Google Closure Compiler

El proceso de utilización de la herramienta Google Closure Compiler, involucró el hecho de seleccionar cada uno de los archivos JavaScript y mandarlos a analizar con el compilador para generar los mismos archivos con código optimizado.

Seguidamente de la ejecución de la referida acción, fue necesario modificar los proyectos con los archivos optimizados y nuevamente realizar las pruebas en búsqueda de mejoras.

#### 3.6.1. Requisitos

- Tener instalado una versión de Java 7 como mínimo. Link de descarga: <http://www.java.com/>

- Tener instalado Maven. Link de descarga: (<http://maven.apache.org/>).

### 3.6.2. Proceso de instalación

- Paso 1

Colocar la carpeta closure-compiler-master brindada junto con esta guía en la ubicación donde se quiere tener instalada la aplicación.

- Paso 2

Dirigirse a dicha carpeta y modificar el archivo ubicado en `/.m2/settings.xml`. Colocándole la siguiente sección de profile dentro de las etiquetas de profiles.

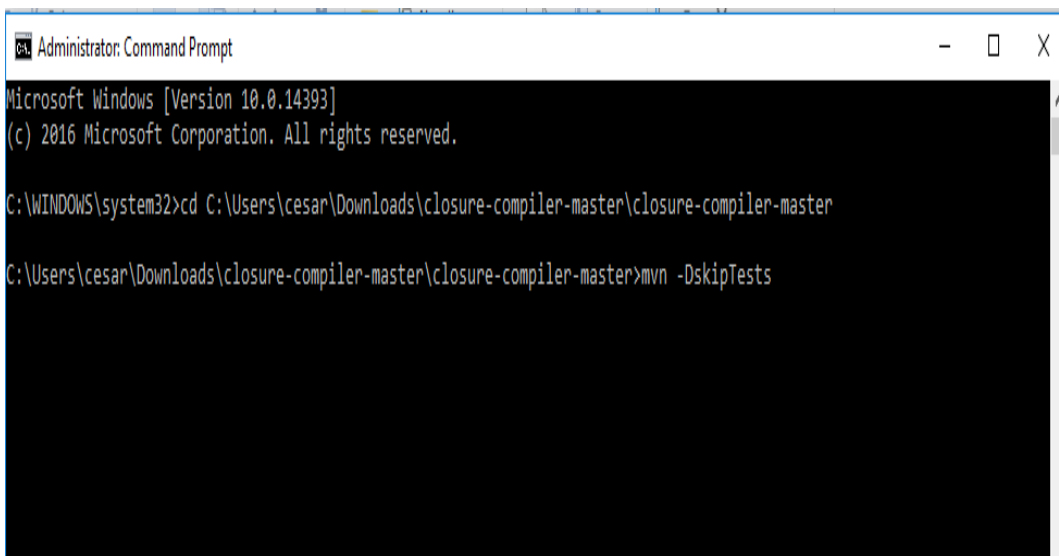
- `<profile>`
- `<id>allow-snapshots</id>`
- `<activation><activeByDefault>true</activeByDefault></activation>`
- `<repositories>`
- `repository>`
- `<id>snapshots-repo</id>`
- `<url>https://oss.sonatype.org/content/repositories/snapshots</url>`
- `<releases><enabled>>false</enabled></releases>`
- `<snapshots><enabled>true</enabled></snapshots>`
- `</repository>`
- `</repositories>`
- `</profile>`

- Paso 3

Dirigirse a la carpeta donde se colocó el proyecto desde consola y correr la siguiente instrucción:

- `mvn -DskipTests`

Figura 1. Instrucción mvn-Dskip Test. Administrador.Command Prompt



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\cesar\Downloads\closure-compiler-master\closure-compiler-master

C:\Users\cesar\Downloads\closure-compiler-master\closure-compiler-master>mvn -DskipTests
```

Fuente: elaboración propia.

Dicha instrucción puede tomar algún tiempo ya que empezará a realizar descargas para construir el proyecto.

- Paso 4

Para verificar que todo se encuentre funcionando bien, es necesario ubicarse en la carpeta target del proyecto y buscar que se encuentre un archivo llamado closure-compiler-1.0-SNAPSHOT.jar.

De encontrarse todo en orden, ejecutar el siguiente comando para java -jar target/closure-compiler-1.0-SNAPSHOT.jar para entrar al compilador.

- Paso 5

Ahora ya debe de estar funcionando todo bien favor ejecutar un ejercicio de prueba como, por ejemplo: var y= 5+5; Luego presionar enter, después Ctrl-Z en Windows o Ctrl-D en Mac o Linux y esperar el resultado.

Si el resultado da var y=10. Esto significa que todo está en orden. Con esto, estará terminada la guía de instalación.

### **3.6.3. Guía de uso**

Para utilizar la herramienta es posible tanto con el compilador de closure compiler, el cual está en la carpeta brindada de closure-compiler-master o desde el archivo jar brindado en la carpeta compiler latest.

Para el primero se debe ejecutar como se vio en la guía de instalación ubicando el archivo closure-compiler-1.0-SNAPSHOT.jar y corriendo los comandos mientras que con el archivo jar se debe ubicar y de la misma manera correr los comandos. Entre los comandos más importantes se encuentran:

- `java -jar compiler.jar -js archivo.js`

Este comando realizará la optimización de código del archivo especificado.

- `java -jar compiler.jar --js_output_file=out.js`

Esta instrucción, realizará el análisis y colocará el resultado en el archivo ya especificado.

- Múltiples archivos

Existen dos formas para analizar múltiples archivos.

- La primera se realiza al colocar archivo por archivo, así como los posibles archivos de resultado:

```
Java -jar compiler.jar --js_output_file=out.js in1.js in2.js in3.js
```

- La segunda se lleva a cabo, seleccionando recursivamente una carpeta con todos los archivos:

```
Java -jar compiler.jar --js_output_file=out.js 'src/**/*.js' '! **_test.js'.
```

- Finalmente, para utilizar más herramientas específicas, se puede correr el comando de ayuda, el cual desplegará todas las posibles acciones que pueden darse con dicha herramienta.

```
java -jar compiler.jar -help
```

## 4. RESULTADOS

### 4.1. Resultados

#### 4.1.1. Mensajes de alerta

- return =true; unreachablecode: mensaje que notifica la presencia de código al cual nunca se tiene acceso. Los archivos con dicho mensaje durante la optimización fueron:
  - Validar\_Localizacion.js del proyecto de cooperación internacional
  - servicio\_salud\_inf\_general.js del proyecto de infraUPE
- return =false; unreachablecode: mismo caso, son proyectos con código inaccesible; lo más recomendable sería realizar un análisis más detallado para quitar dicho código. Los archivos con dicho mensaje durante la optimización fueron:
  - validarCampos.js, validarCampos\_inf.js, validarCampos\_mal.js, validarCampos\_pfa.js, validarCampos\_pla.js, validarCampos\_sar.js, validarCampos\_tos.js y validarCampos\_vih.js del proyecto de epiweb
- {unreachablecode: Error del mismo tipo, se presentó en el siguiente archivo:
  - validarUser.js de la sección de usuarios del proyecto de epiweb.

- WarningSuspiciouscodeistheremissing '+' in case 1,3: es únicamente una advertencia, lastimosamente da a conocer que no tiene reconocido el código case con la coma y corrige advirtiendo que probablemente es un signo de suma, se presentó en los siguientes archivos:
  - \_insertRegistroCaso.js y inicio.js de proyecto de Incan-Cáncer

#### 4.1.2. Cantidad de código en su validación

Tabla XVII. Cooperación internacional

Cooperación internacional	Núm. de líneas	Cantidad de caracteres
validar_Ingreso.js	3	678
validar_Localizacion.js	4	1039
validar_Proyecto.js	2	421

Fuente: elaboración propia.

Tabla XVIII. Monitoreo de servicios

Monitoreo de servicios	Núm. de líneas	Cantidad de caracteres
estructuras.js	17	5616
modificarUsuario.js	3	885
recurso_humano.js	4	1106
servicio_salud_inf_general.js	9	2391
servicio_salud_mapas.js	7	2054
servicios_basicos.js	9	2816
validarCabecera.js	2	409
validarCuerpo.js	4	1381
validarFecha.js	2	292
validarUser.js	3	725

Fuente: elaboración propia.



Tabla XIX. **Epidiario**

<b>Epidiario</b>	<b>Núm. de líneas</b>	<b>Cantidad de caracteres</b>
CatalogoLugar.js	3	661
validarAcciones.js	2	424
validarCabecera.js	2	409
validarCuerpo.js	4	1381
validarFecha.js	2	292
validarNuevoUsuario.js	4	1473
validarUser.js	4	1121

Fuente: elaboración propia.

Tabla XX. **Epiweb**

<b>Epiweb</b>	<b>Número de líneas</b>	<b>Cantidad de caracteres</b>
aviso.js	2	29
CatalogoLugar.js	3	661
validarAcciones.js	8	2390
validarCabecera.js	3	561
validarCuerpo.js	6	2129
validarFecha.js	2	492
validarNuevoUsuario.js	4	1473
validarUser.js	4	1121
fecha.js	9	3362
fecha_des.js	11	3877
fecha_mat.js	11	4172
fecha_mda.js	8	2915
fecha_mhd.js	8	2915
fecha_normal.js	10	3574
fecha_pfa.js	10	3731
fecha_tos.js	11	3877
mostrar_institucion.js	4	1246
validarAcciones.js	14	5759
validarAcciones_dcز.js	38	16246
validarAcciones_des.js	12	4146
validarAcciones_inf.js	126	62024
validarAcciones_mal.js	31	13003
validarAcciones_mat.js	58	23917
validarAcciones_mda.js	17	6495
validarAcciones_mhd.js	13	4527
validarAcciones_pfa.js	42	18579
validarAcciones_pla.js	21	8759
validarAcciones_rab.js	12	7452
validarAcciones_rot.js	44	18327
validarAcciones_sar.js	41	17932
validarAcciones_tos.js	38	16273
validarAcciones_vih.js	16	6131
validarCampos.js	18	8035
validarCampos_dcز.js	19	8691

Continuación de la tabla XX.

validarCampos_des.js	15	6796
validarCampos_inf.js	39	17979
validarCampos_mal.js	30	14420
validarCampos_mat.js	57	28724
validarCampos_mda.js	27	12781
validarCampos_mhd.js	18	849
validarCampos_pfa.js	28	13702
validarCampos_pla.js	22	10350
validarCampos_rab.js	19	8742
validarCampos_rot.js	28	13609
validarCampos_sar.js	30	14420
validarCampos_tos.js	28	13998
validarCampos_vih.js	17	7590

Fuente: elaboración propia.

Tabla XXI. **Usuarios**

<b>Usuarios</b>	<b>Núm. de líneas</b>	<b>Cantidad de caracteres</b>
validarEst.js	4	1147
validarUser.js	11	4522

Fuente: elaboración propia.

Tabla XXII. **Fronteras**

<b>Fronteras</b>	<b>Num. de líneas</b>	<b>Cantidad de caracteres</b>
CatalogoLugar.js	3	661
validarAcciones.js	10	3492
validarCabecera.js	3	561
validarCuerpo.js	3	867
validarFecha.js	2	492
validarNuevoUsuario.js	4	1473
validarUser.js	4	1121

Fuente: elaboración propia.

Tabla XXIII. **INCAN-Cáncer**

<b>Incan-Cancer</b>	<b>Núm. de líneas</b>	<b>Cantidad de caracteres</b>
CambioClave.js	4	944
dataTable.js	2	74
funciones.js	4	1055
inicio.js	8	2786
mantenimiento.js	10	3121
registroCaso_eventos.js	18	6921
registroEncabezado.js	2	76
rregistroEncabezado_eventos.js	7	2319
registroTumor_eventos	12	5213
usuarios.js	12	5446
verificaTumor.js	10	2660
_insertEncabezado.js	3	773
_insertRegistroCaso.js	7	2557
_insertTumor.js	5	1629
_insertVerificaTumor.js	3	774

Fuente: elaboración propia.

Tabla XXIV. **InfraUPE**

<b>infraUPE</b>	<b>Núm de líneas</b>	<b>Cantidad de caracteres</b>
Estructuras.js	17	5616
modificarUsuario.js	3	885
tecurso_humano.js	4	1106
servicio_salud_inf_general.js	9	2391
servicio_salud_mapas.js	7	2054
servicios_basicos.js	9	2816
validarCabecera.js	2	409
validarCuerpo.js	4	1381
validarFecha.js	2	292
validarUser.js	3	725

Fuente: elaboración propia.

### 4.1.3. Calidad del código

Tabla XXV. Cooperación internacional

Cooperación Internacional	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
validar_Ingreso.js	Cooperantes-NuevosCooperantes	0,2	6,36	0,72
validar_Localizacion.js	Proyectos-Lugares de Intervención	0,24	2,99	0,87
validar_Proyecto.js	Proyectos-Ingreso Proyectos	0,18	4,73	0,78

Fuente: elaboración propia.

Tabla XXVI. Monitoreo servicios

Monitoreo servicios	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
Estructuras.js				
modificarUsuario.js	modificarusuario.php	0,18	1,86	0,99
recurso_humano.js	Modificaciones-Estados Comentarios	0,14	3,38	0,94
servicio_salud_inf_general.js				
servicio_salud_mapas.js				
servicios_basicos.js				
validarCabecera.js				
validarCuerpo.js				
validarFecha.js				
validarUser.js	Usuario.php	0,29	4,1	1,75

Fuente: elaboración propia.

Tabla XXVII. Epiidiario

Epiidiario	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
CatalogoLugar.js	catalogos.php	0,2	8	0,89
validarAcciones.js	guardar_ficha.php	0,2	6,07	1,18
validarCabecera.js	guardar_ficha.php	0,22	19,06	1,46
validarCuerpo.js	guardar_ficha.php	0,21	8,6	0,95
validarFecha.js	variables.php	0,22	9,78	0,99
validarNuevoUsuario.js		No se encontró lugar donde es utilizado		
validarUser.js		No se encontró lugar donde es utilizado		

Fuente: elaboración propia.

Tabla XXVIII. Epiweb

Epiweb	Página ingresada en aplicación para capturar el archivo	Request Sent (ms)	Waiting (ms)	Content Download (ms)
aviso.js	guardar_ficha.php	0	0,69	1,42
CatalogoLugar.js	catalogos.php	0,2	8	0,89
validarAcciones.js	guardar_ficha.php	0,2	6,07	1,18
validarCabecera.js	guardar_ficha.php	0,22	19,06	1,46
validarCuerpo.js	guardar_ficha.php	0,21	8,6	0,95
validarFecha.js	variables.php	0,22	9,78	0,99
validarNuevoUsuario.js		No se encontró lugar donde es utilizado		
validarUser.js		No se encontró lugar donde es utilizado		
fecha.js	\\epiweb\fichas\paginas\des\desnutricion_ingresar.php	0,15	21,6	0,95
fecha_des.js				
fecha_mat.js	\\epiweb\fichas\paginas\mat\materna_ingresar.php	0,14	18,94	1,2
fecha_mda.js	\\epiweb\fichas\paginas\mda\MdesnutricionA_ingresar.php	0,16	6,28	1,72

Continuación de la tabla XXVIII.

fecha_mhd.js	\epiweb\fichas\paginas\mhd\MhospitalariaD_ingresar.php	0,11	4,79	1,18
fecha_normal.js	\epiweb\fichas\paginas\chik\chikungunya_ingresar.php	0,18	8,37	0,91
fecha_pfa.js	\epiweb\fichas\paginas\pfa\paralisis_ingresar.php	0,13	18,78	1,19
fecha_tos.js	\epiweb\fichas\paginas\sar\sarampion_ingresar.php	0,13	34,39	1,16
mostrar_institucion.js	\epiweb\fichas\paginas\des\desnutricion_ingresar.php	0,15	37,6	0,87
validarAcciones.js	\epiweb\fichas\paginas\chik\chikungunya_ingresar.php	0,15	6,84	0,66
validarAcciones_dcz.js	\epiweb\fichas\paginas\dcz\dcz_ingresar.php	0,21	31,91	1,11
validarAcciones_des.js	\epiweb\fichas\paginas\des\desnutricion_ingresar.php	0,17	25	0,75
validarAcciones_inf.js	\epiweb\fichas\paginas\inf\bk\influenza_ingresar.php	0,19	22,43	1,76
validarAcciones_mal.js	\epiweb\fichas\paginas\mal\malaria_ingresar.php	0,2	25,98	3,08
validarAcciones_mat.js	\epiweb\fichas\paginas\mat\materna_ingresar.php	0,15	4,95	0,81
validarAcciones_mda.js	\epiweb\fichas\paginas\mda\MdesnutricionA_ingresar.php	0,2	4,98	0,87
validarAcciones_mhd.js	\epiweb\fichas\paginas\mhd\MhospitalariaD_ingresar.php	0,16	5,96	1
validarAcciones_pfa.js	\epiweb\fichas\paginas\pfa\paralisis_ingresar.php	0,16	6,15	1,32
validarAcciones_pla.js	\epiweb\fichas\paginas\pla\plaguicida_ingresar.php	0,3	32,56	0,97
validarAcciones_rab.js	\epiweb\fichas\paginas\rab\rabia_ingresar.php	0,21	30,93	0,99
validarAcciones_rot.js	\epiweb\fichas\paginas\rot\rotavirus_ingresar.php	0,2	64,97	1,43
validarAcciones_sar.js	\epiweb\fichas\paginas\sar\sarampion_ingresar.php	0	2,8	1,34
validarAcciones_tos.js	\epiweb\fichas\paginas\tos\tos_ingresar.php	0,12	36,94	1,41
validarAcciones_vih.js		No se encontró lugar donde es utilizado		
validarCampos.js	\epiweb\fichas\paginas\chik\chikungunya_ingresar.php	0	2,82	1,05
validarCampos_dcz.js	\epiweb\fichas\paginas\dcz\dcz_ingresar.php	0	2,82	0,76
validarCampos_des.js	\epiweb\fichas\paginas\des\desnutricion_ingresar.php	0	1,95	0,91
validarCampos_inf.js	\epiweb\fichas\paginas\inf\bk\influenza_ingresar.php	0	2,39	1,27
validarCampos_mal.js	\epiweb\fichas\paginas\mal\malaria_ingresar.php	0	0,37	0,9

Continuación de la tabla XXVIII.

validarCampos_mat.js	\epiweb\fichas\paginas\mat\materna_ingresar.php	0,16	4,81	0,68
validarCampos_mda.js	\epiweb\fichas\paginas\mda\Mde snutricionA_ingresar.php	0	1,25	1,66
validarCampos_mhd.js	\epiweb\fichas\paginas\mhd\MhospitalariaD_ingresar.php	0	1,82	1,95
validarCampos_pfa.js	\epiweb\fichas\paginas\pfa\paralisis_ingresar.php	0,21	5,14	1,25
validarCampos_pla.js	\epiweb\fichas\paginas\pla\plaguidada_ingresar.php	0,15	7	2,35
validarCampos_rab.js	\epiweb\fichas\paginas\rab\rabiaingresar.php	0,14	5,98	1,13
validarCampos_rot.js	\epiweb\fichas\paginas\rot\rotavirus_ingresar.php	0,46	4,56	1,04
validarCampos_sar.js	\epiweb\fichas\paginas\sar\sarampion_ingresar.php	0,15	4,89	0,56
validarCampos_tos.js	\epiweb\fichas\paginas\tos\tos_ingresar.php	0,16	5,8	0,7
validarCampos_vih.js		No se encontró lugar donde es utilizado		

Fuente: elaboración propia.

Tabla XXIX. **Usuarios**

Usuarios	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
validarEst.js	\epiweb\usuarios\paginas\establecimientos.php	0,41	19,66	1,45
validarUser.js	\epiweb\usuarios\paginas\existentes.php	0,23	35,33	1,06

Fuente: elaboración propia.

Tabla XXX. Fronteras

Fronteras	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
CatalogoLugar.js	\\epiweb\epidiario\paginas\catalogos.php	0,23	5,3	0,91
validarAcciones.js	\\epiweb\epidiario\paginas\guardar_ficha.php	0,12	13,51	1,48
validarCabecera.js	\\epiweb\epidiario\paginas\guardar_ficha.php	2,48	9,36	2,57
validarCuerpo.js	\\epiweb\epidiario\paginas\guardar_ficha.php	0,21	11,1	3,89
validarFecha.js	\\epiweb\epidiario\paginas\variables.php	0,21	4,97	3,02
validarNuevoUsuarios				
validarUser.js	\\epiweb\usuarios\paginas\existentes.php	0,17	10	0,95

Fuente: elaboración propia.

Tabla XXXI. INCAN-Cáncer

Incan-Cancer	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
CambioClave.js	\\incancancer\Incan-Cancer\paginas\cambioClave.php	0,21	1,78	0,94
dataTable.js	\\incancancer\Incan-Cancer\paginas\tablaUsuarios.php	0,38	21,67	1,36
funciones.js	\\incancancer\Incan-Cancer\paginas\header.php	0,2	13,62	2,15
inicio.js	\\incancancer\Incan-Cancer\paginas\header.php	0,18	13,16	1,64
mantenimiento.js	\\incancancer\Incan-Cancer\paginas\mantenimientoRegistro.php	0,2	3,96	1,13
registroCaso_eventos.js	\\incancancer\Incan-Cancer\paginas\registroCaso.php	0,2	5,04	1,72
RegistroEncabezado.js				
registroEncabezado_eventos.js	\\incancancer\Incan-Cancer\paginas\registroEncabezado.php	0,47	14,06	1,78
registroTumor_eventos	\\incancancer\Incan-Cancer\paginas\registroTumor.php	0,19	3,51	1,29
usuarios.js	\\incancancer\Incan-Cancer\paginas\usuarios.php	0,2	2,84	1,12



Continuación de la tabla XXXI.

VerificaTumor.js	\\incancer\Incan-Cancer\paginas\verificaTumor.php	0,39	4,01	1,82
_inserEncabezado.js	\\incancer\Incan-Cancer\paginas\registroEncabezado.php	0,44	6,94	1,54
_insertRegistroCaso.js	\\incancer\Incan-Cancer\paginas\registroCaso.php	0,15	132,94	3,16
_insertTumor.js	\\incancer\Incan-Cancer\paginas\registroTumor.php	0,2	10,92	0,95
_insertVerificaTumor.js	\\incancer\Incan-Cancer\paginas\verificaTumor.php	0,35	14,44	2,07

Fuente: elaboración propia.

Tabla XXXII. **InfraUPE**

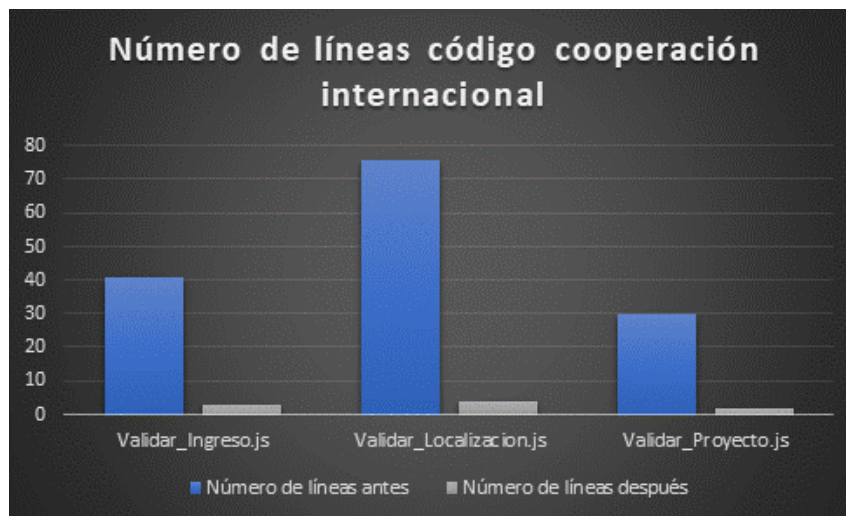
infraUPE	Página ingresada en aplicación para capturar el archivo	Request sent (ms)	Waiting (ms)	Content Download (ms)
Estructuras.js	\\infraUPE\infraupe\paginas\Estructuras.php	0,18	13,69	1,02
modificarUsuario.js	\\infraUPE\infraupe\paginas\modificarUsuario.php	0,22	2,87	1,42
recurso_humano.js	\\infraUPE\infraupe\paginas\Recurso_Humano.php	0,16	19,86	1,39
servicio_salud_inf_general.js	\\infraUPE\infraupe\paginas\Servicio_de_Salud.php	0,2	7,84	1,38
servicio_salud_mapas.js	\\infraUPE\infraupe\paginas\Servicio_de_Salud.php	0,13	9,74	1,49
servicios_basicos.js	\\infraUPE\infraupe\paginas\servicios_basicos.php	0,15	19,22	2,18
validarCabecera.js				
validarCuerpo.js				
validarFecha.js				
validarUser.js	\\infraUPE\infraupe\paginas\usuario.php	0,42	6,95	2,42

Fuente: elaboración propia.

## 4.2. Comparaciones estadísticas

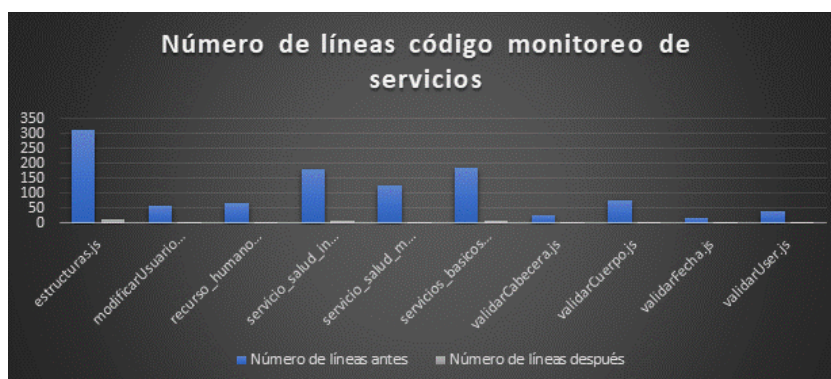
### 4.2.1. Número de líneas de código

Figura 2. Número de líneas, código en cooperación internacional



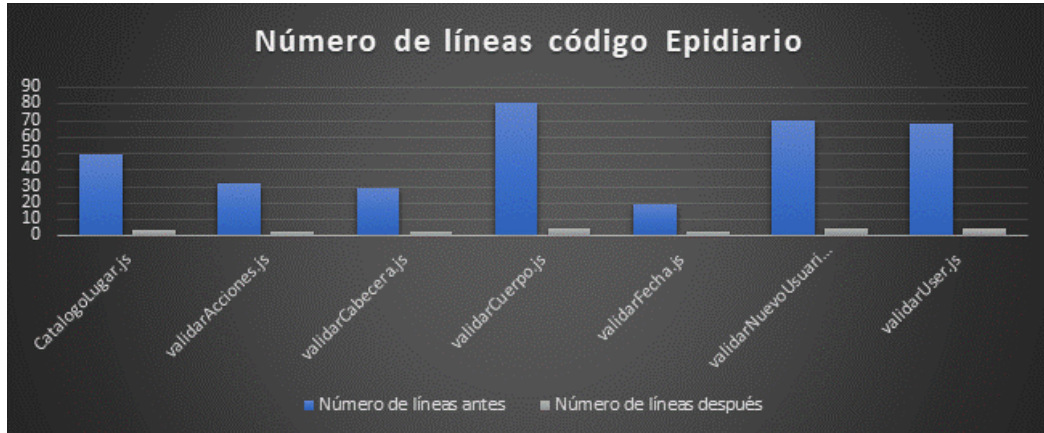
Fuente: elaboración propia.

Figura 3. Número de líneas, código en monitoreo de servicios



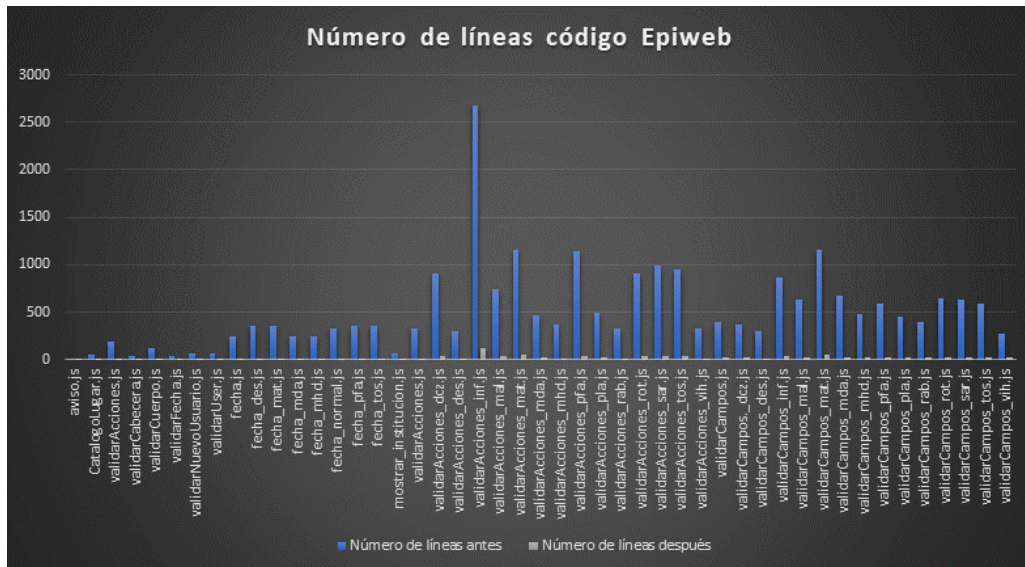
Fuente: elaboración propia.

Figura 4. Número de líneas, código en Epiidiario



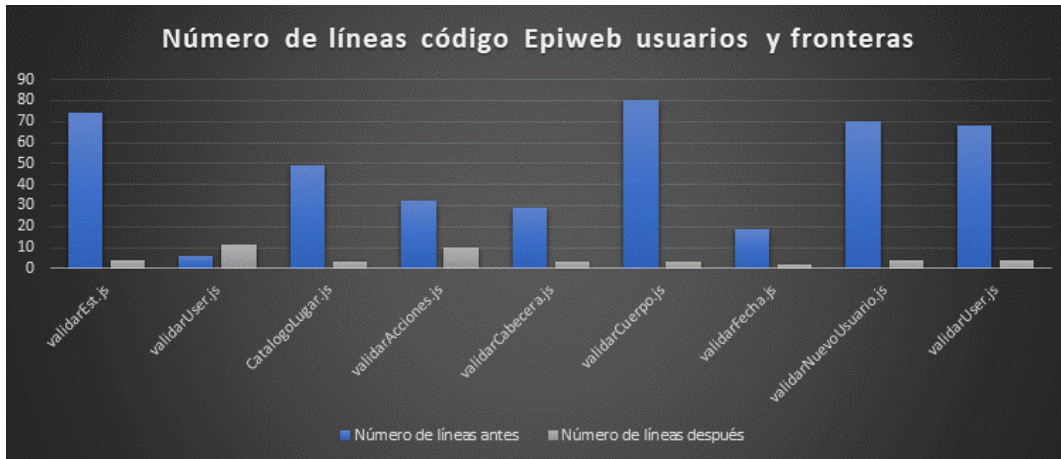
Fuente: elaboración propia.

Figura 5. Número de líneas, código en Epiweb



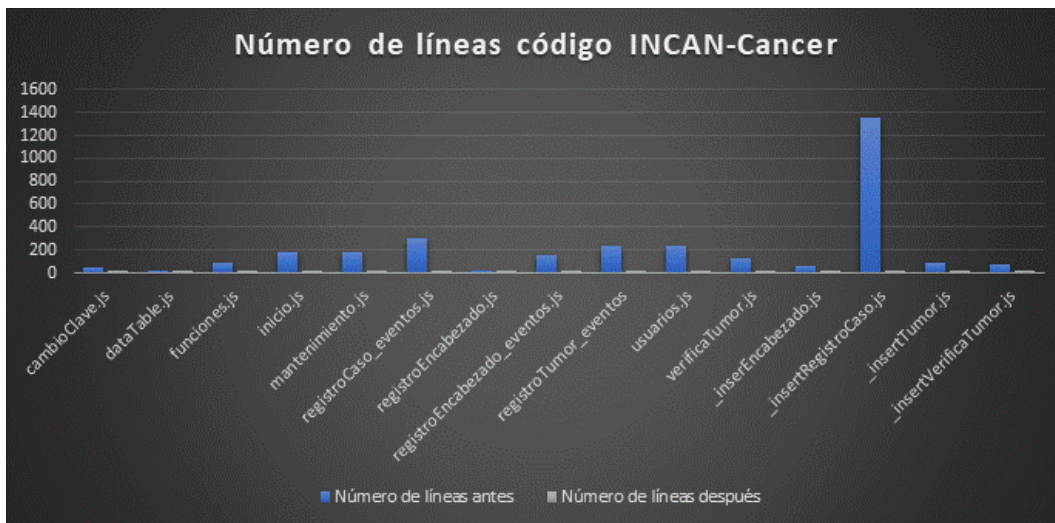
Fuente: elaboración propia.

Figura 6. **Número de líneas, código en Epiweb, usuarios y fronteras**



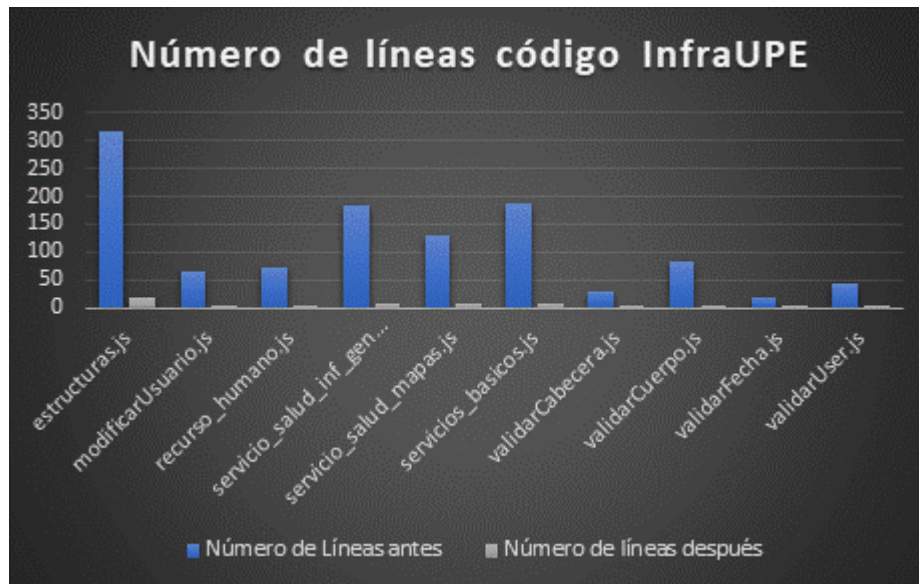
Fuente: elaboración propia.

Figura 7. **Número de líneas, código en Incan-Cáncer**



Fuente: elaboración propia.

Figura 8. **Número de líneas, código en InfraUPE**



Fuente: elaboración propia.

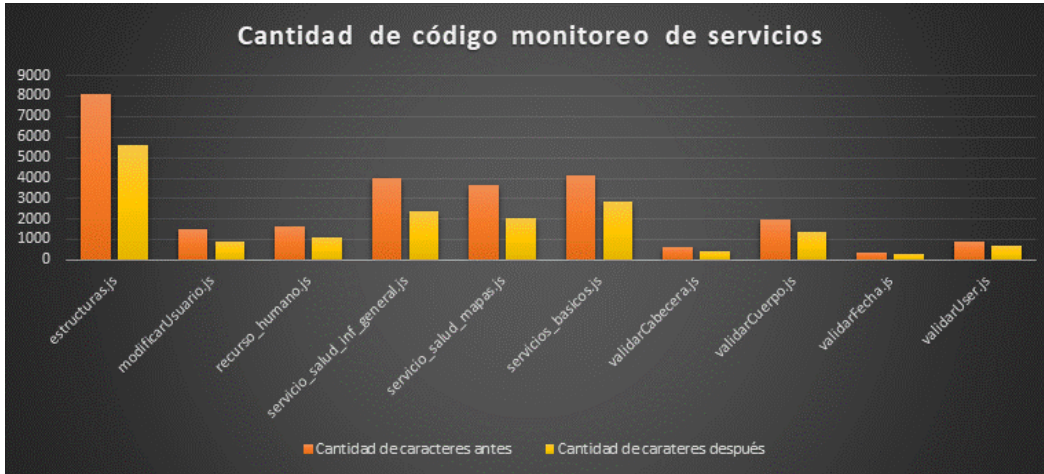
#### 4.2.2. Cantidad de código

Figura 9. **Cantidad de código en cooperación internacional**



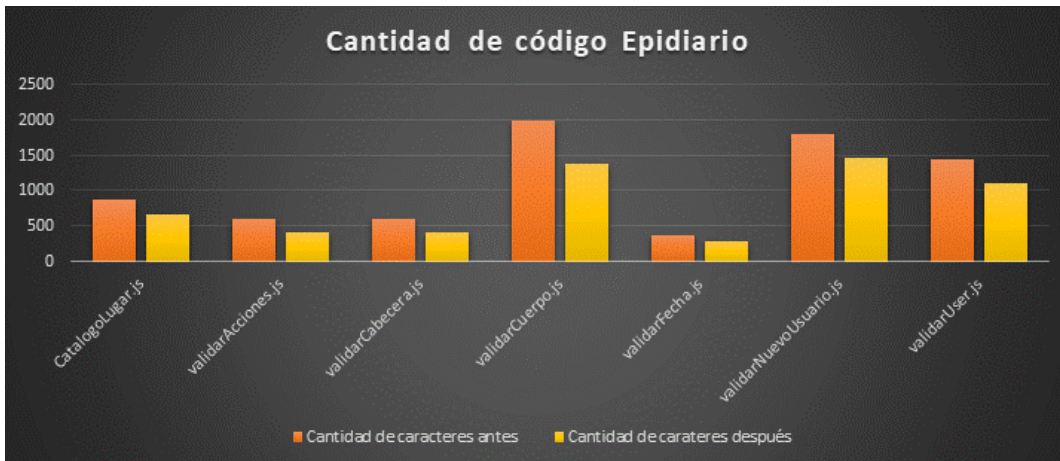
Fuente: elaboración propia.

Figura 10. Cantidad de código en monitoreo de servicios



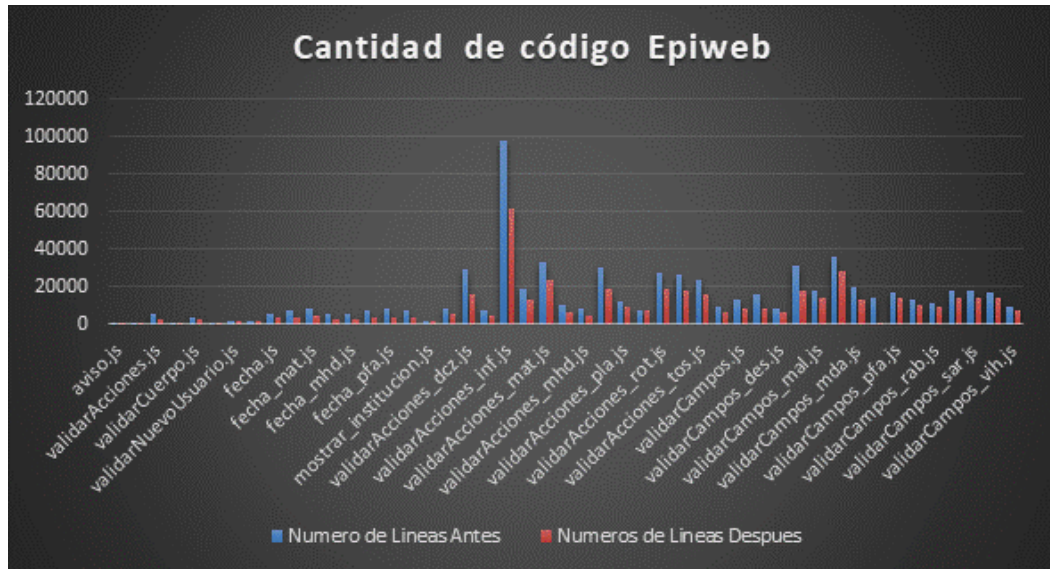
Fuente: elaboración propia.

Figura 11. Cantidad de código en Epiidiario



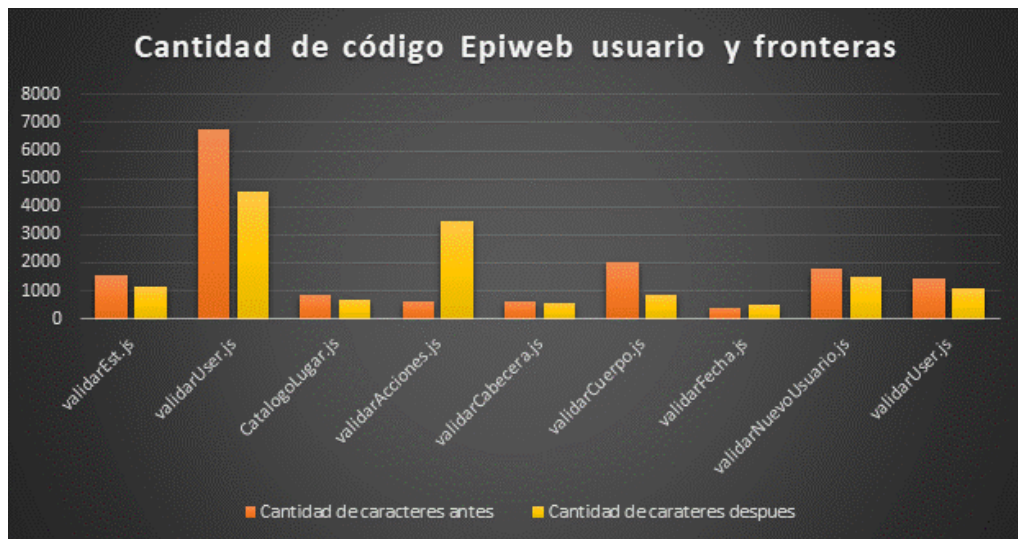
Fuente: elaboración propia.

Figura 12. Cantidad de código en Epiweb



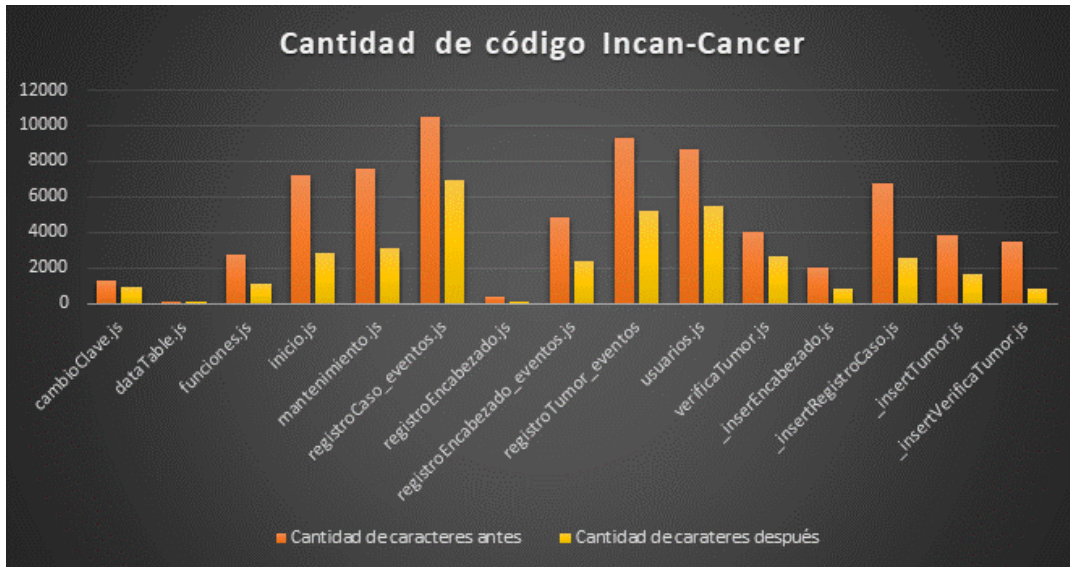
Fuente: elaboración propia.

Figura 13. Cantidad de código en Epiweb, usuarios y fronteras



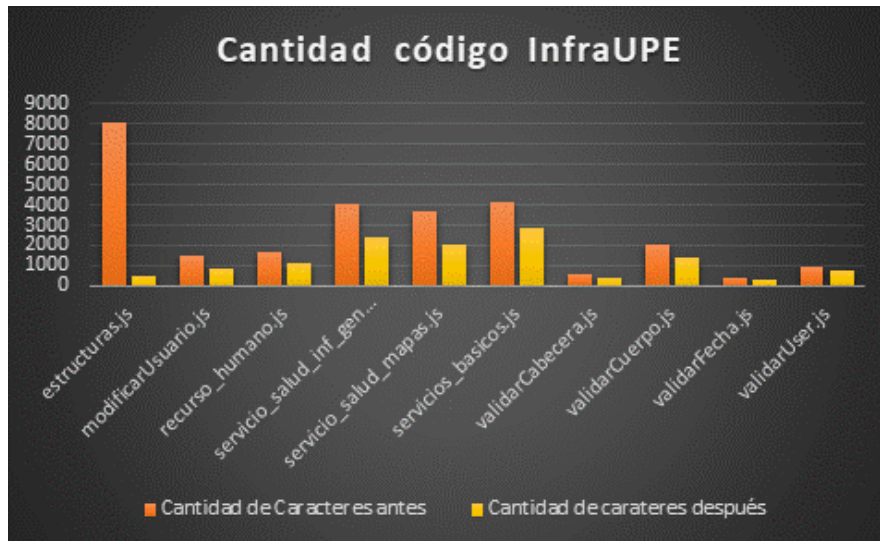
Fuente: elaboración propia.

Figura 14. Cantidad de código en Incan-Cáncer



Fuente: elaboración propia.

Figura 15. Cantidad de código en InfraUPE



Fuente: elaboración propia.



### 4.2.3. Optimización de código

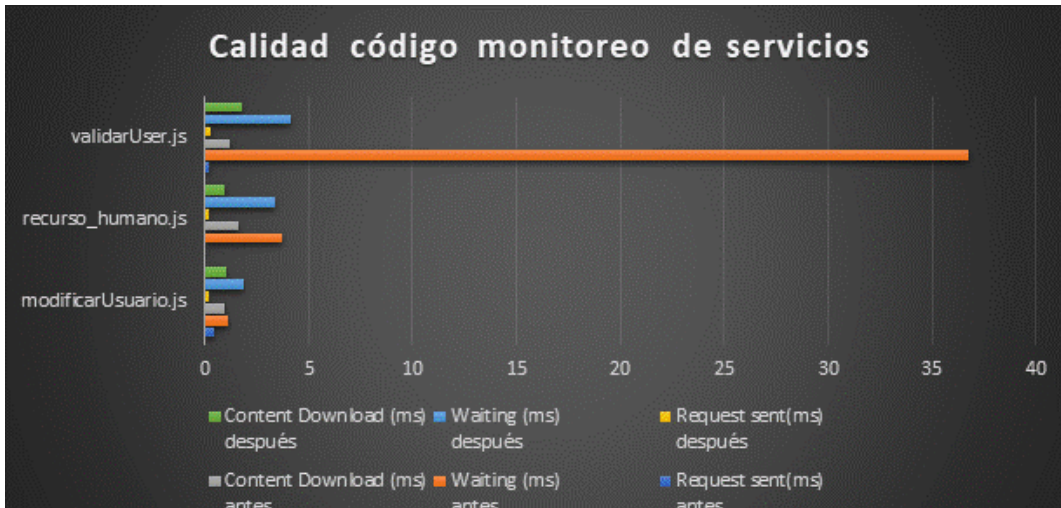
Las gráficas siguientes y sus resultados hacen referencia a la comparación entre la optimización de código realizada y el código anterior de los proyectos, motivo de estudio.

Figura 16. Calidad de código en cooperación internacional



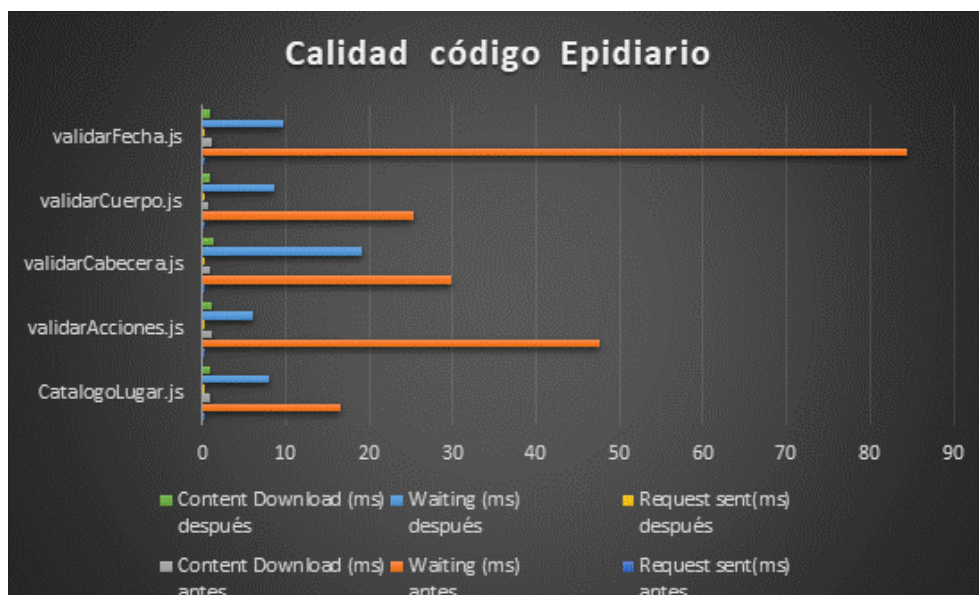
Fuente: elaboración propia.

Figura 17. **Calidad de monitoreo de servicios**



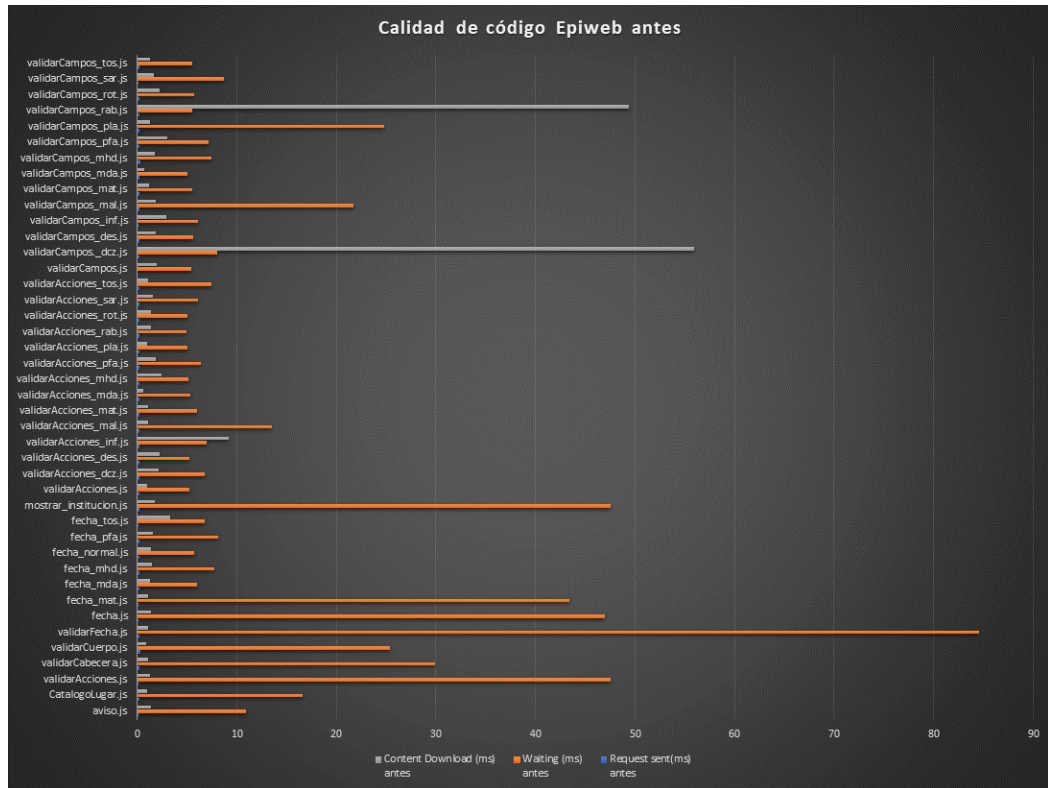
Fuente: elaboración propia.

Figura 18. **Calidad de Epiidiario**



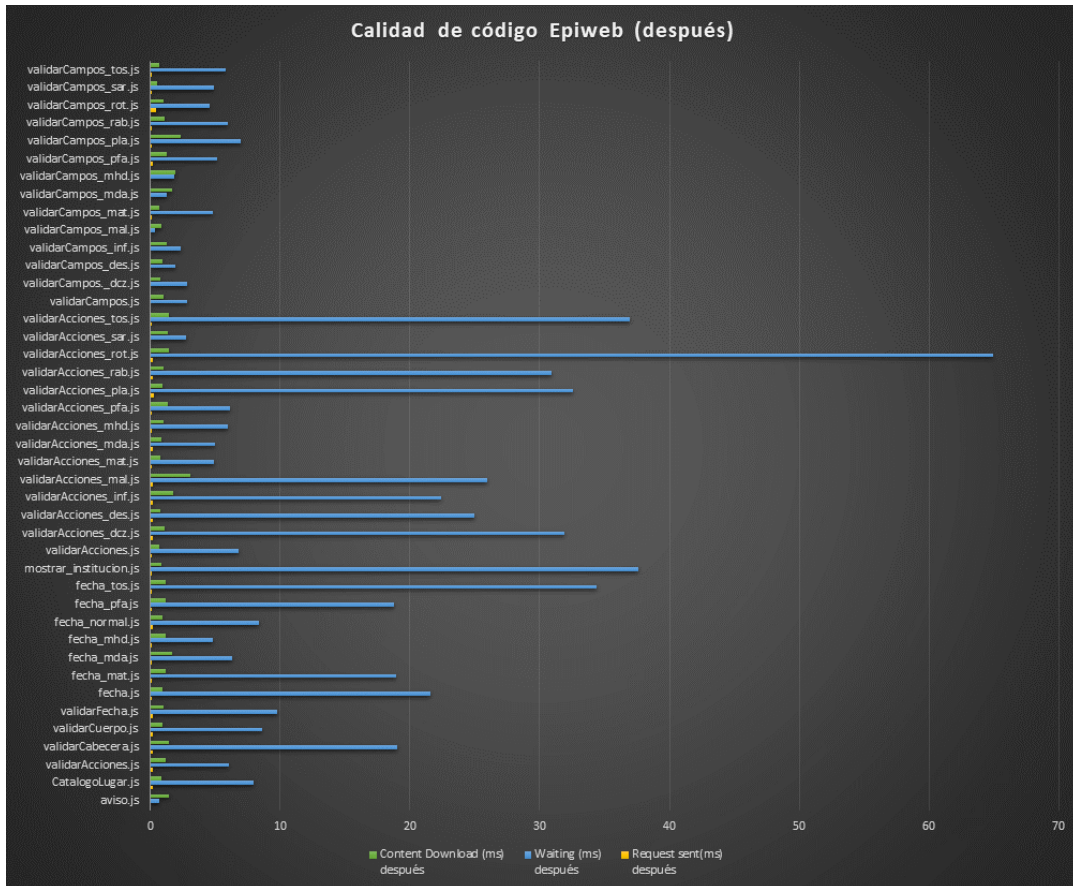
Fuente: elaboración propia.

Figura 19. Calidad de Epiweb (antes)



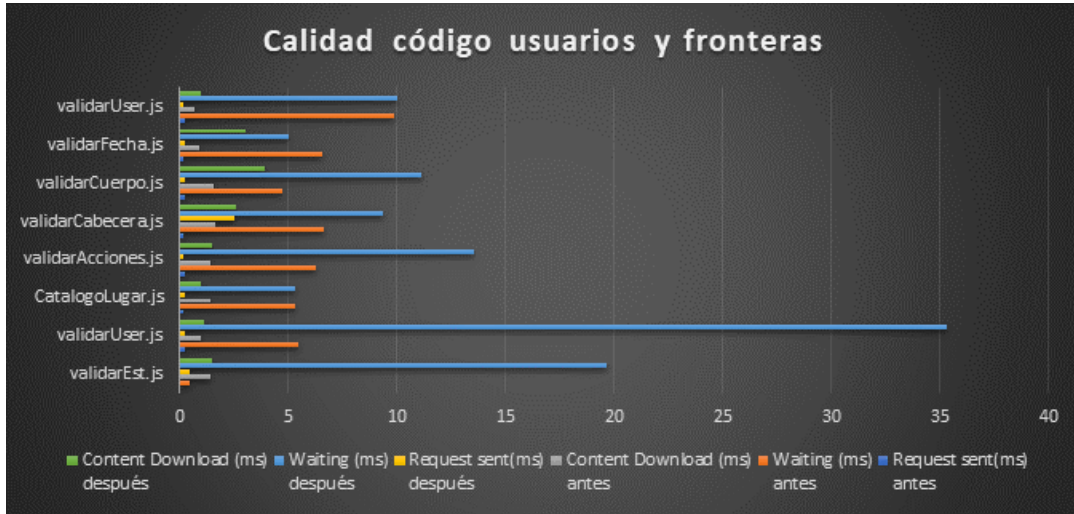
Fuente: elaboración propia.

Figura 20. Calidad de Epiweb (después)



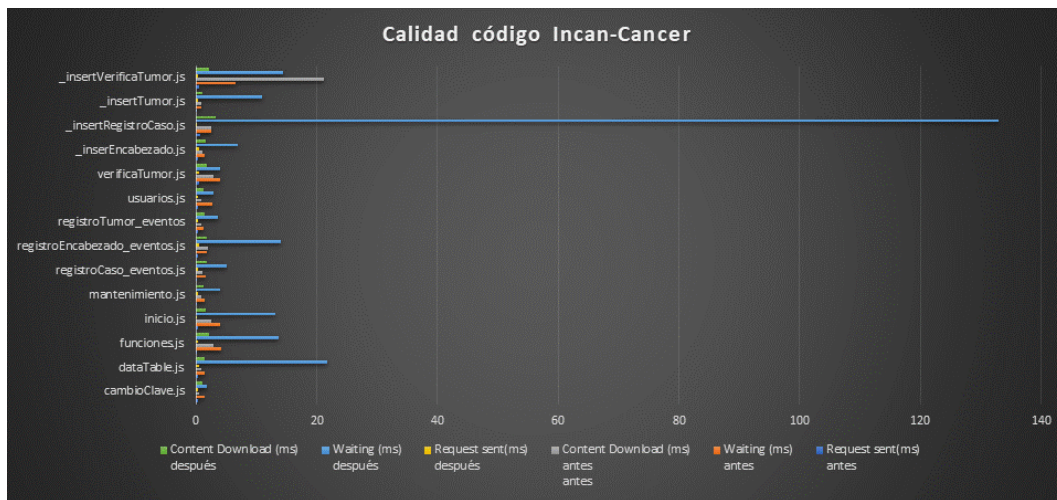
Fuente: elaboración propia.

Figura 21. Calidad de usuarios y fronteras



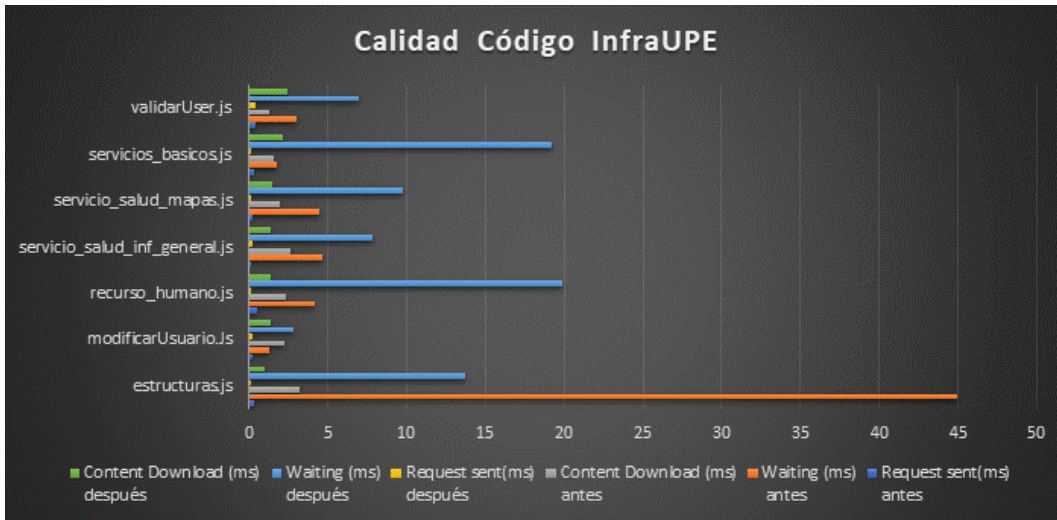
Fuente: elaboración propia.

Figura 22. Calidad de código Incan-Cáncer



Fuente: elaboración propia.

Figura 23. Calidad de código InfraUPE



Fuente: elaboración propia.

### 4.3. Conclusiones sobre la utilización de herramienta

La herramienta utilizada en el presente estudio realiza una optimización del código de JavaScript de una manera muy precisa para dar una respuesta satisfactoria conforme a la expectativa que se tenía de la misma. Optimiza la manera en que posteriormente es interpretado por el compilador para realizar la lectura más rápida, así como también llevando a cabo en el proceso, una compresión del código.

Es acertada la compresión del código. Sin embargo, para proyectos como los que maneja el Ministerio de Salud Pública y Asistencia Social, que en su mayoría no son demasiado robustos, la acción de compactar el tamaño del archivo, aunque bien es útil no necesariamente es indispensable.

Al momento de llevar a cabo el proceso de análisis de eficiencia de código, se encuentra una velocidad ligeramente aumentada la cual puede llegar a dar respuestas más veloces.

Se puede observar con facilidad las gráficas que Google Closure Compiler realiza, otorgando una mejora significativa al código, la cual podría brindar grandes beneficios, al optimizar cualquier proyecto previo a ser colocado en producción para que, de esta forma, consuma menos espacio y mejore la velocidad de respuesta del sistema.





## CONCLUSIONES

1. La optimización de código en los proyectos informáticos de una institución pública como lo es el Ministerio de Salud Pública puede llegar a reducir considerablemente el tamaño que llegan a cubrir estos proyectos en producción.
2. La optimización de código representa una mejora a nivel de velocidad facilitando al intérprete la lectura de información y por consiguiente acelerando la velocidad del software. Sin embargo, aunque los resultados muestren un decremento considerable en la carga del software. Esto no significa una mejora fija.
3. Es posible adaptar una herramienta como lo es Google Closure Compiler para optimizar código de lenguaje JavaScript de proyectos que previamente no se les realizó ningún tipo de análisis u optimización de software, y con esto mejorar su rendimiento.
4. Google Closure Compiler puede llegar a identificar todo aquel código innecesario dentro de los proyectos facilitando a los que utilizan la herramienta descartar este, optimizando así el código dentro del proyecto.
5. Los resultados muestran una optimización de tiempo al realizar cambios como lo son el renombramiento de variables y la simplificación de sintaxis.

6. La herramienta de Google Closure Compiler ofrece una manera bastante sencilla y útil de optimizar el código, brindando suficientes recursos y documentación para poder ser adaptado en una metodología de análisis de código estático.

## RECOMENDACIONES

1. En lo que concierne a la implementación de la herramienta en el sistema, es recomendable que al momento de su instalación se tenga en cuenta las especificaciones que necesita el software para que se adapte a los tipos de proyectos que se desarrollan en el Ministerio de Salud Pública y Asistencia Social.
2. Previamente a la implementación de la herramienta, se debe realizar una lectura detallada de la documentación de la herramienta haciendo énfasis en el alcance, ventajas y desventajas de esta, evitando así cualquier tipo de decisión errónea.
3. Si se planea utilizar la herramienta para optimizar algún proyecto con código JavaScript previo a su publicación a producción sea procesado por esta herramienta. De esta forma se tendrá una versión más rápida y compacta conservando todas sus validaciones y alcances.
4. Realizar *backups* previos a utilizar la herramienta, debido a los resultados obtenidos, casos como código innecesario o comentarios pueden llegar a ser eliminados por la herramienta.
5. No implementarla durante la fase de desarrollo, casos como una incomprensión por parte del desarrollador hacia el código o eliminación de comentarios pueden encontrarse al implementar la herramienta.

6. El Ministerio de Salud considere implementar herramientas que permitan la optimización y análisis de código estático y dinámico en los proyectos que utilizan lenguajes de programación no soportados por la herramienta Google Closure Compiler.

## BIBLIOGRAFÍA

1. ADAFRUIT. *High Temp Waterproof DS18B20 Digital Temperature Sensor* [en línea]. <<http://www.adafruit.com/products/642>> [Consulta: 11 de octubre de 2016].
2. *Análisis estático de software*. [en línea]. <[http://webarchive.nationalarchives.gov.uk/20080731135436/http://www.hse.gov.uk/foi/internalops/nsd/tech\\_asst\\_guides/tast046app1.htm](http://webarchive.nationalarchives.gov.uk/20080731135436/http://www.hse.gov.uk/foi/internalops/nsd/tech_asst_guides/tast046app1.htm)>. [Consulta: 7 de noviembre de 2016].
3. BELMONTE FERNANDEZ, Oscar. *Introducción al lenguaje de programación Java: Una guía Básica*. [en línea]. <<http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>>. [Consulta: 11 de octubre de 2016].
4. CCM Revista Electrónica. *Lenguaje de programación*. [en línea]. <<http://es.ccm.net/contents/304-lenguajes-de-programacion>>. [Consulta: 11 de octubre de 2016].
5. CEREZO LÓPEZ, Yolanda. *Iniciación a la programación en C#, un enfoque práctico*. Madrid: Delta Publicaciones, 2007. 301 p.
6. CRUZ PÉREZ, Roberto. *Asegurar la calidad del código, un primer paso hacia la mejora de la calidad global del software*. [en línea]. <<http://www.expoqa.com/pdf/session2010/AnalisisDeCodigo.pdf>> [Consulta: 11 de octubre de 2016].

7. CSS - *aprenderaprogramar.com*. [en línea]. <[http://www.aprenderaprogramar.com/index.php?option=com\\_content&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&Itemid=163](http://www.aprenderaprogramar.com/index.php?option=com_content&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&Itemid=163)>. [Consulta: 7 de noviembre de 2016].
8. Diccionario de informática y tecnología. definición de Visual Basic, ALEGSA. *¿Qué significa? Información sobre Visual Basic*. [en línea]. <<http://www.alegsa.com.ar/Dic/visual%20basic.php>>. [Consulta: 11 de octubre 2016].
9. DIGITTA.COM. *Herramientas de Google Closure*. [en línea]. <<http://digitta.com / 2009 / 11/herramientas-de-google-closure.html>>. [Consulta: 11 de octubre de 2016].
10. ECURED, La Revista. *Conocimiento con todos y para todos*. [en línea]. <[https://www.ecured.cu/Eclipse,\\_entorno\\_de\\_desarrollo\\_integrado](https://www.ecured.cu/Eclipse,_entorno_de_desarrollo_integrado)>. [Consulta: 11 de octubre de 2016].
11. ESCOFET MARTÍN, Carmen. *El lenguaje SQL. Introducción*. [en línea]. <[http://ocw.uoc.edu/computer-science-technology-and-multi-media/bases-de-datos/bases-dedatos/P06\\_M2109\\_02149.pdf](http://ocw.uoc.edu/computer-science-technology-and-multi-media/bases-de-datos/bases-dedatos/P06_M2109_02149.pdf)>. [Consulta: 11 de octubre de 2016].
12. EXPÓSITO, R. *Análisis estático del código*. [en línea]. <<http://raulexposito.com/documentos/AnalisisEstaticoCodigo.pdf>>. [Consulta: 11 de octubre de 2016].

13. FLORES GARCÍA, Jaime. *¿Qué es un compilador y cómo funciona?* [en línea]. <<http://michelletores.mx/que-es-un-compilador-y-como-funciona/>>. [Consulta: 7 de noviembre de 2016].
14. GRACIA, Luis Miguel. *Análisis de código dinámico*. [en línea]. <<https://unpocodejava.wordpress.com/2012/09/26/herramientas-java-para-el-analisis-dinamico-de-codigo/>>. [Consulta: 11 de octubre de 2016].
15. GARCIA OTERINO, Ana M. del Carmen. *Entiende qué es Maven*. [en línea]. <<http://www.javiergarzas.com/2014/06/maven-en-10-min.html>>. [Consulta: 11 de octubre de 2016].
16. GONZÁLEZ, Enrique. *¿Qué es PHP? ¿Para qué sirve? Un potente lenguaje de programación para crear páginas Web*. [en línea]. <[http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193)>. [Consulta: 11 de octubre de 2016].
17. Google Closure Tools. *Google te ofrece sus herramientas de programación en JavaScript*. [en línea]. <<http://gizmos.republica.com/programas-y-aplicaciones/google-closure-tools-google-te-ofrece-sus-herramientas-de-programacion-en-javascript.html>>. [Consulta: 7 de noviembre de 2016].
18. GOSLING, James. *The Java Language Specification*. California, EE.UU.: Sun Microsystems, 2000. 505 p.

19. GRIGORIK, Ilya. *Optimizar la eficacia del contenido*. [en línea]. <<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/?hl=es-419>>. [Consulta: 11 de octubre de 2016].
20. *Historia del lenguaje Java*. [en línea]. <[http://www.cad.com.mx/historia\\_del\\_lenguaje\\_java.htm](http://www.cad.com.mx/historia_del_lenguaje_java.htm)>. [Consulta: 7 de noviembre de 2016].
21. La revista informática. *Lenguaje de programación C#*. [en línea]. <<http://www.larevistainformatica.com/C1.htm>>. [Consulta: 11 de octubre de 2016].
22. \_\_\_\_\_. *Lenguaje de Programación HTML*. [en línea]. <<http://www.larevistainformatica.com/Lenguaje-programacion-HTML.htm>>. [Consulta: 11 de octubre de 2016].
23. La revista informática.com. *Principales lenguajes de programación*. [en línea]. <<http://www.larevistainformatica.com/JavaScript.htm>>. [Consulta: 11 de octubre de 2016].
24. Lenguajes de programación. *Programación Java. Herramientas de programación*. [en línea]. <<http://www.lenguajes-de-programacion.com/programacion-java.shtml>>. [Consulta: 7 de noviembre de 2016].
25. LLUNA, Eduardo. *Análisis estático de código en el ciclo de desarrollo de software de seguridad crítica*. Madrid, España: Revista Española REICIS, 2010. 55 p.



26. LOUDEN, Kennet C. *Lenguajes de programación: principios y práctica*. 2a ed. California, EE.UU.: Thomson. 2003. 625 p.
27. Manual – PHP .net. *PHP: ¿Qué es PHP?* [en línea]. <<http://php.net/manual/es/intro-whatism.php>>. [Consulta: 7 de noviembre de 2016].
28. MASIP, David. *¿Qué es Oracle? Definición de Oracle como herramienta de gestión de bases de datos*. [en línea]. <<http://www.desarrolloweb.com/articulos/840.php>>. [Consulta: 7 de noviembre de 2016].
29. *Medir los tiempos de carga en los recursos*. [en línea]. <<https://developers.google.com/web/tools/chrome-devtools/network-performance/resource-loading#view-network-timing-details-for-a-specific-resource>>. [Consulta: 7 de noviembre de 2016].
30. MENENDEZ-BARZANALLANA ASENCIO, Rafael. *Desarrollo de aplicaciones Web*. [en línea]. <<http://www.um.es/docencia/barzana/DAWEB/Lenguaje-deprogramacion-JavaScript-1.pdf>>. [Consulta: 7 de noviembre de 2016].
31. *Migración automatizada de scripts de compilación utilizando análisis dinámico y refactorización basada en la búsqueda*. [en línea]. <<http://research.microsoft.com/en-us/um/people/livshits/papers/pdf/thesis.pdf>>. [Consulta: 7 de noviembre de 2016].
32. Ministerio de Salud Pública y Asistencia Social de Guatemala, (MSPAS). *Proyecto de Políticas en Salud y Educación (HEPP) con el apoyo*

de la Agencia de los Estados Unidos para el Desarrollo Internacional (USAID). [en línea]. <<http://www.mspas.gob.gt/files/Descargas/LeyPresupuesto2014/17bis/Plan%20estrategico%20MSPAS%202014-2019%20version%20040414.pdf>>. [Consulta: 7 de noviembre de 2016].

33. *Organización Panamericana de la Salud. Información, metodología, procesos y procedimientos.* [en línea]. <<http://mejoremosguate.org/blog/wp-content/uploads/2012/02/Salud.pdf>>. [Consulta: 7 de noviembre de 2016].
34. PERRY, Greg; GARZA MARÍN, A. David y SÁNCHEZ LÓPEZ, Rebeca Alicia. *Aprendiendo Visual Basic 6 en 21 días.* México: Prentice Hall, 2003. 851 p.
35. *Programadores Web.* [en línea]. <<https://developers.google.com/web/tools/chrome-devtools/network-performance/understanding-resource-timing>>. [Consulta: 7 de noviembre de 2016].
36. RAMIREZ HERNANZ, Eloy. *Desarrollo y aplicación de herramientas para el análisis del envío anticipado de video mediante HTTP/2 en servicios de streaming multimedia.* [en línea]. <[15-17.http://oa.upm.es/42935/1/PFC\\_ELOY\\_RAMIREZ\\_HERNANZ\\_2016.pdf](http://oa.upm.es/42935/1/PFC_ELOY_RAMIREZ_HERNANZ_2016.pdf)>. [Consulta: 7 de noviembre de 2016].
37. Red Gráfica Latinoamericana, RG. *El lenguaje de programación PHP.* [en línea]. <<http://redgrafica.com/El-lenguaje-de-programacion-PHP>>. [Consulta: 7 de noviembre de 2016].

38. *Registro de establecimiento de dispositivo médico y listado de dispositivo.* [en línea]. <<http://www.fda.gov/MedicalDevices/ProductsandMedicalProcedures/GeneralHospitalDevicesandSupplies/InfusionPumps/ucm202511.htm>>. [Consulta: 7 de noviembre de 2016].
39. RODRÍGUEZ SALA, Jesús Javier y MARTÍNEZ BONASTRE, Oscar. *Introducción a la programación: teoría y práctica.* 2a ed. Madrid, España: ECU, 2006. 211 p.
40. ROUSE, Margaret. *My SQL, ¿Qué es MySQL?* [en línea]. <<http://search.datacenter.techtarget.com/es/definicion/MySQL>>. [Consulta: 7 de noviembre de 2016].
41. *Tecnología confiable de Google.* [en línea]. <<http://developers.google.com/closure/>>. [Consulta: 7 de noviembre de 2016].
42. Transcripción de herramientas para desarrolladores de Google Chrome. *Funciones que ofrece Google Chrome Developer Tools.* [en línea]. <<https://prezi.com/lk0eyydjnieuw/herramientas-para-desarrolladores-de-google-chrome/>>. [Consulta: 7 de noviembre de 2016].
43. UREÑA ALMAGRO, Carlos. *Lenguajes de programación.* [en línea]. <<http://lsi.ugr.es/~curena/doce/lp/tr-11-12/lp-c01-impr.pdf>>. [Consulta: 7 de noviembre de 2016].
44. VICENCIO, Roberto. *Como crear software de calidad.* [en línea]. <<http://dominatuspackagesenoracle.com/blog/descubre-que-es-plsql-y-como-se-debe-utilizar-en-oracle.html>>. [Consulta: 7 de noviembre de 2016].

