



**UNIVERSIDAD SALESIANA DE BOLIVIA  
CARRERA DE INGENIERÍA DE SISTEMAS**

**TESIS**



**Para obtener el Grado de Licenciatura en Ingeniería de  
Sistemas**

**ESTUDIO DE LA IMPLEMENTACIÓN DE UNA HERRAMIENTA  
INTERACTIVA DE GENERACIÓN DE CÓDIGO QUE DEFINE  
COLORES CON VARIABLES CSS EN UN SITIO WEB**

**Postulante: Fernando Javier Averanga Aruquipa**

**Docente Guía: Ing. Carla Silvana Escobar Olivares**

**LA PAZ - BOLIVIA  
2019**

## **DEDICATORIA**

A Dios, porque es Él quien permite que logre mis objetivos trazados. A mi madre por su amor incondicional y por apoyarme en todo momento. A mi Padre Enrique que en paz descanse, fue quien me apoyó y aconsejó a lo largo de mi carrera, y sé que Él estaría muy orgulloso de ver que lo haya culminado no solo con esfuerzo, sino también con mucha inspiración.

## **AGRADECIMIENTO**

A la Universidad Salesiana de Bolivia que me acogió durante todo el tiempo de carrera y desarrollo del proyecto. A la Ing. Carla Escobar por su apoyo como docente guía, A ReactBolivia quien me apoyó en este proyecto como una contribución a la comunidad Open Source y así culminar con éxito la presente tesis de grado.

## **RESUMEN**

La presente tesis de grado corresponde al análisis, diseño e implementación de una Herramienta interactiva de generación de código que define colores con variables CSS denominado CSS ColorVars, que mejora la simplicidad en las variables y el rendimiento de un sitio web.

Un sitio web implica las tecnologías de HTML CSS y Javascript nuevas características dinámicas van surgiendo, y la utilización de variables en el lenguaje CSS no es la excepción. Al definir una variable CSS en un color y obtener sus luminosidades y transparencias, es un trabajo manual demasiado tardado, y lamentablemente es imposible modificar estas variables utilizando funciones preprocesadas.

El objetivo principal del proyecto consiste en solucionar este tipo de dificultades y determinar su efecto.

El desarrollo de software se ha realizado bajo la metodología Open Unified Process (OpenUP) y consistió en la concepción, elaboración, construcción y transición de una herramienta interactiva utilizando diferentes tecnologías web de código abierto.

En el transcurso de la lectura, se procederá a explicar los beneficios obtenidos a partir de la implementación del software, así como también la comparación de procesos antes y después de su puesta en producción.

Por último, se le invita a proceder con la lectura del presente trabajo de investigación, esperando que alcance sus expectativas y permita aumentar su conocimiento sobre las tecnologías utilizadas.

## **ABSTRACT**

The present thesis corresponds to the analysis, design and implementation of an interactive code generation tool that defines colors with CSS variables called CSS ColorVars, which improves the simplicity of the variables and the performance of a website.

A website implies HTML CSS and Javascript technologies new dynamic features are emerging, and the use of variables in the CSS language is no exception. When defining a CSS variable in a color and obtaining its luminosities and transparencies, it is too late a manual job, and unfortunately it is impossible to modify these variables using preprocessed functions.

The main objective of the project is to solve this type of difficulty and determine its effect.

The software development was carried out under the Open Unified Process (OpenUP) methodology and consisted of the conception, elaboration, construction and transition of an interactive tool using different open source web technologies.

During the reading, the benefits obtained from the implementation of the software will be explained, as well as the comparison of processes before and after their production.

Finally, you are invited to proceed with the reading of this research work, hoping that it will reach your expectations and allow you to increase your knowledge about the technologies used.

<b>Índice de contenido</b>	<b>Nro. Pág.</b>
CAPÍTULO I .....	1
GENERALIDADES .....	1
Introducción .....	1
1.1. Antecedentes del tema .....	1
1.2. Antecedentes legales .....	4
1.3. Datos estadísticos.....	9
1.4. Trabajos afines .....	11
1.5. Descripción del problema.....	13
1.6. Pregunta de Investigación.....	15
1.7. Hipótesis .....	15
1.8. Variables .....	15
1.9. Indicadores .....	15
1.10. Objetivos de la investigación.....	16
1.11. Justificación .....	16
1.11.1. Justificación social .....	16
1.11.2. Justificación técnica.....	16
1.11.3. Justificación económica .....	17
1.12. Alcances .....	17
1.13. Límites .....	18
1.14. Definición de aportes .....	18
1.14.1. Aporte del proyecto.....	18
1.14.2. Aporte académico .....	18

	Nro. Pág.
CAPÍTULO II.....	19
MARCO TEÓRICO .....	19
2. Introducción .....	19
2.1. Fundamento teórico de la investigación.....	19
2.1.1. Herramienta interactiva.....	19
2.1.2. Generación de código.....	21
2.1.3. Colores Web .....	21
2.1.4. Variables CSS.....	21
2.1.5. Desarrollo web front-end.....	23
2.1.6. Simplicidad .....	25
2.1.7. Rendimiento.....	25
2.2. Marco metodológico.....	25
2.2.1. Tipo de investigación .....	25
2.2.2. Nivel de investigación .....	27
2.2.3. Método.....	28
2.2.4. Diseño.....	28
2.2.5. Población y muestra .....	28
2.2.6. Operacionalización de variables de la Hipótesis.....	29
2.3. Técnicas, instrumentos de la recolección de datos.....	31
2.3.1. Técnica de la recolección de datos.....	31
2.3.2. Instrumento de recolección de datos .....	31
2.4. Técnicas para la gestión del trabajo .....	32
2.4.1. Herramientas para la obtención de resultados esperados.....	32

	Nro. Pág.
2.4.2. Definición de conceptos técnicos.....	33
<b>CAPÍTULO III.....</b>	<b>35</b>
<b>DESARROLLO DE LA PROPUESTA DE INVESTIGACIÓN.....</b>	<b>35</b>
3.1. Método de desarrollo de la propuesta .....	35
3.1.1.Open unified process (open up).....	35
3.2. Diagrama de procesos.....	37
3.2.1. Diagrama de proceso general antes de la implementación .....	37
3.2.2. Diagrama de proceso con la herramienta interactiva.....	39
3.3. Arquitectura .....	40
3.3.1. Componentes principales.....	40
3.3.2. Patrón de diseño.....	41
3.3.3. Patrón arquitectónico .....	42
3.4. Prototipos de interfaz gráfica .....	44
3.5. Construcción.....	46
3.5.1. Framework.....	46
3.5.2. Herramientas, plataformas y tecnología.....	47
3.6. Implementación .....	48
3.6.1. Algoritmo de la implementación.....	48
3.6.2. Implementación basada en el patrón de diseño y arquitectónica.....	49
<b>CAPÍTULO IV .....</b>	<b>52</b>
<b>ANÁLISIS, SÍNTESIS Y EVALUACIÓN DE RESULTADOS.....</b>	<b>52</b>
4. Introducción .....	52

	Nro. Pág.
4.1. Diagnóstico .....	53
4.1.1. Análisis y síntesis para el efecto de simplicidad en las variables.....	53
4.1.2. Análisis y síntesis para el efecto de rendimiento web.....	59
4.2. Evaluación.....	62
4.2.1. Preguntas de la evaluación.....	62
 CAPÍTULO V .....	63
COMPROBACIÓN DE LA HIPÓTESIS .....	63
5.1. Comprobación de la hipótesis.....	63
 CAPÍTULO VI .....	74
CONCLUSIONES Y RECOMENDACIONES.....	74
6.1. Conclusiones .....	74
6.2. Recomendaciones .....	75
 Bibliografía:.....	76
Webgrafía: .....	78
 ANEXO (A) .....	
ANEXO (B) .....	.....
ANEXO (C) .....	.....
ANEXO (D) .....	.....

<b>Índice de Gráficos</b>	<b>Nro. Pág.</b>
Gráfico 1.1: Tabla de soporte de Variables CSS para navegadores .....	9
Gráfico 1.2: Tabla de soporte de funciones de Color CSS3 para navegadores ...	10
Gráfico 2.1: Paradigma de diseño .....	20
Gráfico 2.2: El Camino del Front-End .....	24
Gráfico 3.1: Las tres capas y flujo de trabajo del proceso OpenUP.....	35
Gráfico 3.2: Ciclo de vida Open Unified Process (Open UP) .....	37
Gráfico 3.3.1: Diagrama de proceso antes de la implementación de la herramienta interactiva.(General) .....	38
Gráfico 3.3.2: Diagrama de proceso antes de la implementación de la herramienta interactiva.(Parcial) .....	39
Gráfico 3.4: Diagrama de proceso con la herramienta interactiva .....	39
Gráfico 3.5: Patrón de diseño Singleton .....	42
Gráfico 3.6: Patrón de arquitectónico MVVM.....	43
Gráfico 3.7:. Interfaz principal de la Herramienta interactiva .....	44
Gráfico 3.8:.Cambiar color con Color picker de la Herramienta.....	45
Gráfico 3.9:.Crear nuevo color mediante la herramienta .....	45
Gráfico 3.10:.Código generado listo para copiar y utilizar en la hoja de estilos ....	46
Gráfico 3.11:. Almacén centralizado de la lógica de la aplicación .....	49
Gráfico 3.12: Componentes basados en el patrón de diseño Singleton global....	50
Gráfico 3.13: Componentes reusables para la interfaz de usuario .....	51
Gráfico 4.1: Sitios web reconocidos que utilizan Variables CSS .....	52
Gráfico 4.2: Definición de colores con variables CSS según la muestra poblacional .....	55
Gráfico 4.3: Salida en pantalla de variaciones de luminosidad.....	57

Gráfico 4.4: Salida en pantalla de variaciones de transparencia .....	58
Gráfico 4.5: Resultados de diagnósticos no soportados por la herramienta interactiva.....	60
Gráfico 4.6: Resultados de diagnósticos soportados por la herramienta interactiva .....	61

<b>Índice de Tablas</b>	<b>Nro. Pág.</b>
Tabla 1.1: Trabajos a fines .....	11
Tabla 1.2: Trabajos a fines .....	12
Tabla 1.3: Trabajos a fines .....	12
Tabla 1.4: Trabajos a fines .....	13
Tabla 2.1: Operacionalización de variables .....	30
Tabla 4.1: Cantidad de procesos para definir colores con variables CSS según la muestra poblacional.....	54
Tabla 5.1: Tabla de la distribución de Ji Cuadrado.....	64
Tabla 5.2: Tabla de la frecuencias observadas - Hipótesis A .....	66
Tabla 5.3: Tabla de la frecuencias teóricas - Hipótesis A .....	66
Tabla 5.4: Tabla de la frecuencias observadas - Hipótesis B .....	70
Tabla 5.5: Tabla de contingencias observadas - Hipótesis B .....	71
Tabla 5.6: Tabla de la frecuencias teóricas - Hipótesis B .....	71
.	
.	

# **CAPÍTULO I.**

**GENERALIDADES**

# CAPÍTULO I

## GENERALIDADES

### Introducción

En los últimos años, algunas características dinámicas han comenzado a aparecer como parte del lenguaje CSS en sí. Las variables nativas de CSS, o Propiedades personalizadas, como se les llama oficialmente, ya están disponibles y tienen un excelente soporte para el navegador.

Al definir una variable CSS para un color y tener la opción de aplicar sus luminosidades de claridad, oscuridad y transparencia, resulta un trabajo manual demasiado lento, y lamentablemente es imposible modificar estas variables utilizando funciones preprocesadas.

Es por ello que este trabajo dará a conocer una de las alternativas para que en dichas variables CSS de colores, sea posible modificarlas entre otros aspectos los cuales serán explicados de una manera clara y precisa en el desarrollo del mismo.

### 1.1. Antecedentes del tema

Todo aquel que haya usado un lenguaje de programación, ya está familiarizado con el concepto de variable. Las variables le permiten almacenar y actualizar los valores que su programa necesita para funcionar.

“La belleza de las variables es que le permiten almacenar su valor en un lugar y actualizarlo sobre la marcha para varios propósitos diferentes. No es necesario que agregue nuevas entidades con diferentes valores en todo su programa: todas las actualizaciones de valores se realizan utilizando el mismo lugar de almacenamiento, es decir, su variable.” (Perna, 2018)

CSS es principalmente un lenguaje declarativo, carece de capacidades dinámicas, se podría decir que tener variables en CSS sería casi una contradicción en términos.

Afortunadamente los idiomas de la Web son muy parecidos a los idiomas vivos, que evolucionan y se adaptan de acuerdo con el entorno y las necesidades de sus profesionales, CSS no es una excepción.

En resumen, las variables nativas de CSS ahora se han convertido en una realidad, algo emocionante en el mundo de CSS, esta nueva tecnología es bastante sencilla de aprender y usar, pero también es un gran cambio en el paradigma gracias a los nuevos estándares web ya que el uso de variables en CSS no es muy diferente a usar variables en un lenguajes de programación.

Una forma para aprovechar la flexibilidad de las variables al diseñar sitios web es mediante el uso de preprocesadores como Sass o Less.

“Los preprocesadores le permiten establecer variables y utilizarlas en funciones, bucles, operaciones matemáticas, etc. ¿Significa esto que las variables CSS son irrelevantes? No del todo, principalmente porque las variables CSS son algo diferentes de las variables del preprocesador.

Las diferencias surgen del hecho de que las variables CSS son propiedades CSS que están vivas y se ejecutan en el navegador, mientras que las variables del preprocesador se compilan en el código CSS normal, por lo tanto, el navegador no sabe nada sobre ellas. Lo que esto significa es que puede actualizar las variables de CSS en un documento de hoja de estilo, dentro de los atributos de estilo en línea y los atributos de presentación de SVG (dialecto de marcado XML para la generación de gráficos vectoriales), o puede elegir manipularlos sobre la marcha utilizando el lenguaje de programación Javascript. No puedes hacer nada de esto con las variables del preprocesador. Esto abre todo un mundo de posibilidades.”  
(Perna, 2018)

“La edad de los colores hexadecimales redundantes ha terminado. El inspector de Chrome inevitablemente te permitirá cambiar las definiciones de las variables de CSS en el navegador y verás los resultados en vivo en la página. En el caso de los colores, eso significa que se puede cambiar una variable y hacer que todo lo

que usa esa variable cambie de color a la vez. Si usamos nuestros métodos antiguos, esa característica será inútil.” (Albaugh, 2015)

“Las variables CSS se pueden usar en funciones nativas del navegador como calc(), rgb(), funciones de gradientes, etc.

Sin embargo como toda tecnología, tiene sus ventajas y desventajas. Por ejemplo, al fusionar funciones y variables preprocesadas junto a las variables nativas de CSS, posiblemente existan algunos errores de compilación.

Entonces ¿Cómo se podría aplicar opacidad a una variable de color CSS?. No se puede tomar un valor de color existente y aplicarle un canal alfa. Es decir, no puede tomar un valor hexadecimal existente, darle un componente alfa y usar el valor resultante con otra propiedad.

Lo que se podría hacer es que convertir manualmente el valor hexadecimal de la variables CSS a un triplete RGB (composición del color en términos de la intensidad de los colores primarios de la luz) para usar con rgba() nativo, almacenar ese valor en la propiedad personalizada (incluyendo las comas), Sustituir ese valor usando var() entro de la función rgba() con su valor alfa deseado.” (BoltClock, 2016)

Esto se podría convertir en una buena práctica para solucionar la redundancia de declaraciones de alpha (define la opacidad de un color como un número entre 0.0 a 1.0) de opacidad aunque realizarlo sería algo tedioso si no se cuenta con una paleta de colores en la hoja de estilos.

Actualmente existen sitios web reconocidos que utilizan Variables CSS de Color en todo su ecosistema, como por ejemplo: youtube.com, messenger.com, reddit.com, twitch.tv, entre otros.

## 1.2. Antecedentes legales

La W3C es el consorcio internacional que genera recomendaciones y estándares que garantizan el crecimiento de la web a largo plazo.

A continuación se dará a conocer el estándar y recomendación de las Variables Nativas de CSS según la W3C:

- [\*\*Propiedades personalizadas de CSS para variables en cascada Módulo Nivel 1. Recomendación del candidato del W3C, 03 de diciembre de 2015 \(W3C, 2015\)\*\*](#)

“Este módulo introduce variables en cascada como un nuevo tipo de valor primitivo que es aceptado por todas las propiedades de CSS y propiedades personalizadas para definirlas.

1. *Introducción.*- Esta sección no es normativa.

Los documentos o aplicaciones grandes (e incluso los pequeños) pueden contener bastante CSS. Muchos de los valores en el archivo CSS serán datos duplicados; por ejemplo, un sitio puede establecer un esquema de color y reutilizar tres o cuatro colores en todo el sitio. La modificación de estos datos puede ser difícil y propensa a errores, ya que está dispersa en todo el archivo CSS, y puede no ser susceptible de búsqueda y reemplazo.

Este módulo introduce una familia de propiedades personalizadas definidas por el autor conocidas colectivamente como propiedades personalizadas, que permiten a un autor asignar valores arbitrarios a una propiedad con un nombre elegido por el autor, y la función var() , que le permite a un autor usar esas Valores en otras propiedades en otro lugar en el documento. Esto facilita la lectura de archivos grandes, ya que los valores aparentemente arbitrarios ahora tienen nombres informativos y hacen que la edición de dichos archivos sea mucho más fácil y menos propensa a errores, ya que uno solo tiene que cambiar el valor una vez, en la propiedad personalizada, y el cambio Se propagara a todos los usos de esa variable automáticamente.

CSS es un lenguaje para de *propiedades* '\*'escribir la representación de documentos estructurados (como HTML y XML) en la pantalla, en el papel, en el discurso, etc.

## 2. *Definición de propiedades personalizadas:*

Esta especificación define un conjunto abierto de propiedades llamadas propiedades personalizadas, que, entre otras cosas, se utilizan para definir el valor de sustitución de las funciones var().

- A diferencia de otras propiedades de CSS, los nombres de propiedades personalizados distinguen entre mayúsculas y minúsculas.
- Todas las propiedades personalizadas no se restablecen. Podemos definir una propiedad en el futuro que restablece todas las variables. Las palabras clave de todo el CSS se pueden usar en propiedades personalizadas, con el mismo significado que en cualquier otra propiedad.

Es decir, se interpretan en el tiempo de valor en cascada como normal y no se conservan como el valor de la propiedad personalizada y, por lo tanto, no se sustituyen con la variable correspondiente.

## 3. *Usando variables en cascada: la notación ‘var()’*

El valor de una propiedad personalizada se puede sustituir en el valor de otra propiedad con la función var().

La función var() se puede usar en lugar de cualquier parte de un valor en cualquier propiedad de un elemento. La función var() no se puede utilizar como nombres de propiedad, selectores o cualquier otra cosa además de los valores de propiedad. (Al hacerlo, por lo general se produce una sintaxis no válida, o bien un valor cuyo significado no tiene conexión con la variable).

El primer argumento de la función es el nombre de la propiedad personalizada que se va a sustituir. El segundo argumento de la función, si se proporciona, es un valor de reserva, que se utiliza como valor de sustitución cuando la propiedad personalizada a la que se hace referencia no es válida.

Si una propiedad contiene una o más funciones var() , y esas funciones son sintácticamente válidas, se debe asumir que toda la gramática de la propiedad es válida en el momento del análisis. Solo se comprueba la sintaxis en el tiempo de valor calculado, después de que se hayan sustituido las funciones var().

Para sustituir un var() en el valor de una propiedad:

- a. Si la propiedad personalizada nombrada por el primer argumento de la función var () está marcada por la animación , y la función var() se está utilizando en la propiedad de animación o en una de sus longitudes largas, considere que la propiedad personalizada tiene su valor inicial para el resto de este algoritmo.
- b. Si el valor de la propiedad personalizada nombrada por el primer argumento de la función var() no es el valor inicial, reemplace la función var() por el valor de la propiedad personalizada correspondiente .
- c. De lo contrario, si la función var() tiene un valor de reserva como segundo argumento, reemplace la función var() por el valor de reserva. Si hay referencias var() en el respaldo, sustitúyase también.
- d. De lo contrario, la propiedad que contiene la función var() no es válida en el momento del valor calculado.

#### 4. APIs

Todas las declaraciones de propiedades personalizadas tienen un conjunto de indicadores que no distingue entre mayúsculas y minúsculas.

Las propiedades personalizadas no aparecen en un objeto CSSStyleDeclaration en forma de camellos, porque sus nombres pueden tener letras mayúsculas y minúsculas que indican propiedades personalizadas distintas. El tipo de transformación de texto que realiza la envoltura automática de camellos es incompatible con esto. Se puede acceder a ellos con su nombre propio a través de getPropertyValue() / etc. "(W3C, 2015)

- **Módulo de color CSS Nivel 3 - Recomendación del W3C 19 junio 2018**

#### Resumen

CSS (Cascading Style Sheets) es un lenguaje para describir la representación de documentos HTML y XML en pantalla, papel, voz, etc. Utiliza propiedades y valores relacionados con el color para colorear el texto, fondos, bordes y otras partes de elementos en un documento. Esta especificación describe valores de color y propiedades para el color de primer plano y la opacidad de grupo. Estos incluyen propiedades y valores del nivel 2 de CSS y nuevos valores.

#### Estado de este documento

Esta sección describe el estado de este documento en el momento de su publicación. Otros documentos pueden reemplazar este documento. Puede encontrar una lista de las publicaciones actuales del W3C y la última revisión de este informe técnico en el índice de informes técnicos del W3C en <https://www.w3.org/TR/>.

Este documento ha sido revisado por miembros del W3C, por desarrolladores de software y por otros grupos del W3C y partes interesadas, y el Director lo avala

como una recomendación del W3C. Es un documento estable y puede usarse como material de referencia o citado de otro documento.

El papel del W3C al hacer la Recomendación es llamar la atención sobre la especificación y promover su despliegue generalizado. Esto mejora la funcionalidad y la interoperabilidad de la Web.

Esta Recomendación editada es idéntica a la Recomendación propuesta el 15 de marzo de 2018, excepto por correcciones editoriales menores, como se señala en la sección Cambios.

Este documento fue producido por el Grupo de Trabajo de CSS como una recomendación propuesta. Este documento está destinado a convertirse en una recomendación editada del W3C.

Una recomendación del W3C es un documento que ha sido ampliamente revisado y está listo para su implementación. W3C anima a todos a implementar esta especificación y devolver comentarios a los problemas de GitHub.

Este documento fue producido por un grupo que opera bajo la Política de Patentes del W3C. La W3C mantiene una lista pública de cualquier divulgación de patentes realizada en relación con los entregables del grupo; esa página también incluye instrucciones para divulgar una patente. Una persona que tenga conocimiento real de una patente que la persona cree que contiene Reclamo (s) Esencial (es) debe divulgar la información de acuerdo con la sección 6 de la Política de Patentes del W3C.

Este documento se rige por el documento de proceso del W3C del 1 de febrero de 2018. Un informe de implementación separado muestra que cada prueba en el conjunto de pruebas fue aprobada por al menos dos implementaciones independientes.” (W3C, 2018)

### 1.3. Datos estadísticos

“Can I use”, es un sitio web que proporciona tablas actualizadas de compatibilidad de navegadores para la compatibilidad con tecnologías web front-end en navegadores web de escritorio y móviles.

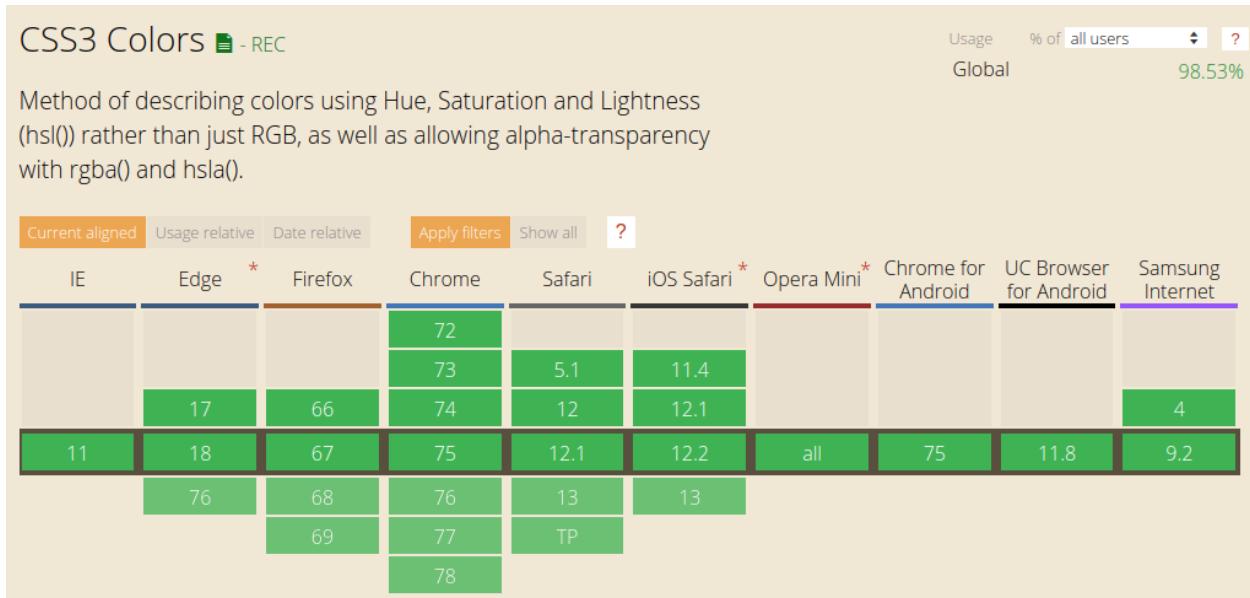


**Gráfico 1.1: Tabla de soporte de Variables CSS para navegadores.**

Fuente: Can I use (<https://caniuse.com>)

En el anterior gráfico se puede observar el soporte de las Variables y Funciones de color CSS para los navegadores actuales.

Como se puede observar en la tabla, el soporte para utilizar las variables CSS es totalmente compatible con los navegadores modernos, a excepción de Internet Explorer que en la actualidad dejó de tener soporte técnico en todas sus versiones anteriores.



**Gráfico 1.2: Tabla de soporte de funciones Color CSS3 para navegadores.**

Fuente: Can I use (<https://caniuse.com/>)

**“Colores CSS3:** Método para describir colores usando Hue, Saturación y Luminosidad (hsl ()) en lugar de solo RGB, así como permitir transparencia alfa con rgba () y hsla ().

Hue: Piensa en una rueda de colores. Alrededor de 0 o 360 o son rojos. 120 o es donde están los verdes y 240 o son azules. Use cualquier cosa entre 0-360. Los valores superiores e inferiores serán el módulo 360.

Saturación: 0% está completamente des-saturado (escala de grises). 100% está completamente saturado (a todo color).

Luminosidad: 0% es completamente oscuro (negro). 100% es completamente ligero (blanco). 50% es la ligereza promedio.

Alfa: valor de opacidad / transparencia. 0 es completamente transparente. 1 es completamente opaco. 0.5 es 50% transparente.

Puede manipular a mano cualquiera de esos cuatro valores y tener un sentido decente de lo que va a suceder. Cambia el tono para dar un paseo por la rueda de colores. Cambia la saturación para obtener colores más profundos o más apagados. Cambie la claridad para mezclar esencialmente en blanco o negro.

Es posible que tenga algunas habilidades mentales `rgb()`, sabiendo que `rgb(255, 0, 0)`es claramente rojo o `rgb(0, 0, 0)`negro, pero manipularlas para obtener un color púrpura claro o comenzar con un bosque verde y obtener un poco más claro no es exactamente matemática mental. Incluso podría ser el tipo inteligente que puede identificar el color por códigos hexadecimales.” (Can I Use, 2018)

#### **1.4. Trabajos afines**

A continuación se citan los trabajos a fines a la presente tesis de grado:

<b>Título:</b>	<b>Desarrollo e Implementación de un Sistema de Información para el control del proceso de capacitación de una empresa del rubro de las telecomunicaciones en el Perú</b>
<b>Autor:</b>	Carlos Moisés González López
<b>Año:</b>	2016
<b>Institución:</b>	Universidad Católica Sedes Sapientiae
<b>Descripción:</b>	El presente proyecto de investigación corresponde al análisis, diseño e implementación de un sistema de información en plataforma web denominado SIGIC (Sistema de Gestión Integrada y Control de Procesos), utilizado para la mejora continua de procesos de una entidad del rubro de las telecomunicaciones en el Perú.

**Tabla 1.1: Trabajos a fines**

**Fuente: Elaboración propia**

<b>Título:</b>	Aplicación web para tratar a personas con trastorno cognitivo leve (TCL) basada en html5
<b>Autor:</b>	Reynaldo Quispe Sánchez
<b>Año:</b>	2015
<b>Institución:</b>	Universidad Mayor de San Andrés
<b>Descripción:</b>	El presente trabajo tiene un método inductivo, Lanzaremos una hipótesis, el cual apunta a afirmar que estas tecnologías (tecnologías web), puede coadyuvar a mejorar el grado de deterioro cognitivo de los adulto mayores mediante ejercicios mentales.

**Tabla 1.2: Trabajos a fines**

**Fuente: Elaboración propia**

<b>Título:</b>	Análisis, diseño e implementación de la aplicación web para el manejo del distributivo de la facultad de ingeniería
<b>Autor:</b>	Eduardo Guillermo Aguilar Riera.
<b>Año:</b>	2013
<b>Institución:</b>	Universidad de Cuenca
<b>Descripción:</b>	El objetivo de esta tesis es analizar, desarrollar e implementar una aplicación web, abarcando el proceso que implica el distributivo de la Facultad de Ingeniería, utilizando para esto una base de datos libre.

**Tabla 1.3: Trabajos a fines**

**Fuente: Elaboración propia**

<b>Título:</b>	Herramienta web de hojas de estilo en cascada
<b>Autor:</b>	Kalthoum Y. Adam
<b>Año:</b>	2011
<b>Institución:</b>	Facultad de San Diego, Universidad del Estado
<b>Descripción:</b>	En esta tesis, desarrollamos una herramienta web CSS destinada a los desarrolladores web que codificaron manualmente su HTML y CSS para tener un control completo sobre el diseño y el estilo de la página web. La herramienta es un asistente de formulario que ayuda a los desarrolladores a través de una interfaz fácil de usar para crear una plantilla de sitio web con un código CSS y XHTML válido.

**Tabla 1.4: Trabajos a fines**

**Fuente: Elaboración propia**

A diferencia de trabajos ya mencionados, este trabajo se diferencia de los demás porque se desarrolló e implementó una herramienta interactiva de generación de código para luego evaluar su efecto en los sitios web entre otros aspectos que se explicará durante el desarrollo de este trabajo.

### **1.5. Descripción del problema**

La especificación de la variable nativas de CSS inevitablemente colisiona con las tendencias Sass.

Independientemente de si en algún proyecto web ya está utilizando las variables nativas de CSS, es probable que ya se use un preprocesador como Sass para administrar sus variables. Algunos desarrolladores web muy organizados, incluso pueden hacer uso de mapas de variables en Sass o una arquitectura donde se tengan sus variables de configuración.

Desafortunadamente, la sintaxis para acceder a los mapas Sass es un poco detallada, por lo que algunas personas prefieren usar funciones personalizadas para acceder a sus variables de mapa.

El hecho de usar la sintaxis en variables Sass más que la nueva especificación de variables CSS, no significa que deba abandonar esta nueva especificación.

Si se utiliza Sass para crear variables de colores globales compatibles con variables CSS, sería utilizando un mapa de variables Sass e iterar sobre él con un bucle. Este nivel de complejidad no es adecuado para todos, pero ciertamente es factible para escribir menos código.

Utilizar variables nativas de CSS en funciones preprocesadas de Sass como ser `lighten()`, `darken()`, y `transparency()`, es un procesos que no funcionara porque dichas funciones preprocesadas no son compatibles con las variables nativas. Para que Sass pueda pre-procesar estas funciones, tendría que reemplazar la variable CSS con un valor estático, en cuyo caso se pierde la utilidad de la variable nativa.

También se podría crear variables CSS de colores, sin la necesidad de la fusión, con ningún preprocesador. Pero esto tiene sus limitaciones ya que en el lenguaje CSS no existen funciones explícitas para manipular estas variables o crear bucles con sus diferentes luminosidades o transparencias.

Las luminosidades y transparencias de un color habitualmente utilizadas en la guía de estilos de un sitio web son posibles de crear con variables CSS, sin embargo esto realmente sería muy tardado y nada fácil de realizar.

En realidad crear variables CSS de color con sus luminosidades y transparencias resulta complejo de realizar, para quienes no tengan mucha experiencia con las variables CSS.

## **1.6. Pregunta de Investigación**

¿Qué mejoras se han identificado con la implementación de una herramienta interactiva de generación de código, que define colores con variables CSS en un sitio web?

## **1.7. Hipótesis**

“La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, mejora la simplicidad en las variables y el rendimiento en un sitio web”.

## **1.8. Variables**

### **VARIABLE INDEPENDIENTE:**

x = Implementación de una herramienta interactiva de generación de código, que define colores con variables CSS.

### **VARIABLE DEPENDIENTE:**

y = Simplicidad en las variables y rendimiento en un sitio web.

## **1.9. Indicadores**

- Cantidad de variables CSS declaradas para un color, con el fin utilizarlo para aplicar claridad, oscuridad, y transparencia del color.
- Cantidad de procesos al declarar variables CSS de un color, con el fin utilizarlo para aplicar claridad, oscuridad, y transparencia del color.
- Tiempo de descarga del recurso en un ejemplo real.
- Cantidad de Bytes de descarga del recurso en un ejemplo real.

La operacionalización de variables se encuentra en el capítulo 2 del presente documento.

## **1.10. Objetivos de la investigación**

Determinar las mejoras que se han identificado en la implementación de una herramienta interactiva de generación de código, que define colores con variables CSS en un sitio web.

## **1.11. Justificación**

### **1.11.1. Justificación social**

Es importante para una arquitectura CSS moderna obtener colores con variables CSS, ya que estas variables están vivas en el navegador y esto es un beneficio a la creatividad del desarrollador ya que podrá utilizar las últimas tendencias de los estándares de la web y a la vez obtener una buena optimización mediante en el desarrollo.

Bajo esta premisa el objetivo de este trabajo es crear colores web utilizando variables CSS con funciones programáticas en el navegador de una forma ágil, y utilizar estas variables en proyectos web actuales es un gran beneficio, ya que si se modifica una de las variables definidas, todos los selectores descendientes donde se utilizó esta variable se modifica en tiempo de ejecución. Si se utilizan métodos antiguos, esa característica será inútil.

Otro beneficio es que será una aplicación de código abierto con licencia MIT (licencia de software que se originó en el Instituto Tecnológico de Massachusetts), lo cual hace posible que cualquiera pueda clonar el repositorio, aportar contribuciones, reportar errores y/o sugerir mejoras libremente.

### **1.11.2. Justificación técnica**

El desarrollo de esta herramienta se justifica técnicamente, por el uso de alta tecnología y conceptos web.

Se implementó una herramienta interactiva de generación de código para el manejo de colores con variables CSS y lograr creará funciones basados en

funciones Sass que se encarguen de generar las variaciones claras, oscuras y con transparencia de los colores.

La herramienta interactiva irá acorde con las últimas tendencias del desarrollo de sitios web, para esto se utilizará un marco de trabajo progresivo para la creación de interfaces de usuario, con una guía de estilos Javascript estándar, junto a conceptos de centralización de estado.

En resumen se contará con una estructura basada en el patrón de diseño singleton global para los componentes, implementando programación reactiva junto a conceptos de persistencia de datos, que beneficiará a un mejor desarrollo flexible.

#### **1.11.3. Justificación económica**

El tiempo es dinero y por esa razón el presente trabajo tiene por objetivo aportar la implementación de una herramienta interactiva de generación de código que pretende mejorar el rendimiento de sitios web utilizando los últimos estándares de CSS y de la web en general.

La herramienta interactiva es de código abierto y estará bajo la licencia MIT, lo que por parte quiere decir es que es una herramienta sin fines de lucro.

#### **1.12. Alcances**

Para poder lograr el objetivo de este trabajo se tomará en cuenta los siguientes alcances:

- La interfaz de usuario que facilita y agiliza la creación de variables CSS de colores.
- El código generado de las variables CSS de colores se crean mediante un selector global.
- Capacidad de aplicar claridad, oscuridad, y transparencia de una variable de color, sin limitaciones en uso.

- Soporte de sintaxis para CSS, SCSS, STYLUS, SASS, LESS.

### **1.13. Límites**

- Para la elaboración del presente trabajo sólo se estudiará el código de las variables CSS enfocada a los colores.
- La herramienta sólo podrá ser ejecutada para usuarios que cuenten con un navegador predisposto para soportar Javascript.
- La herramienta preferentemente es para personas dedicadas al diseño y desarrollo de sitios web (HTML, CSS y Javascript).
- Tener acceso a internet mediante cualquier dispositivo que cuente con un navegador actual, preferiblemente sea computadoras de escritorio o portátiles.

### **1.14. Definición de aportes**

El aporte refiere a aquella contribución que alguien realiza a otro individuo o a una organización.

#### **1.14.1. Aporte del proyecto**

El trabajo aportará con una herramienta interactiva de generación de código que define colores con variables CSS, como una solución efectiva, aplicando conocimientos de desarrollo de sitios web front-end, con un estudio más interno de las nuevas especificaciones del lenguaje CSS.

#### **1.14.2. Aporte académico**

El aporte académico es reflejado en la combinación de conocimientos de programación web reactiva, funcional, basada en el patrón de diseño singleton, y con una arquitectura Modelo-Vista-Vista-Modelo.

# **CAPÍTULO II.**

MARCO TEÓRICO

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2. Introducción**

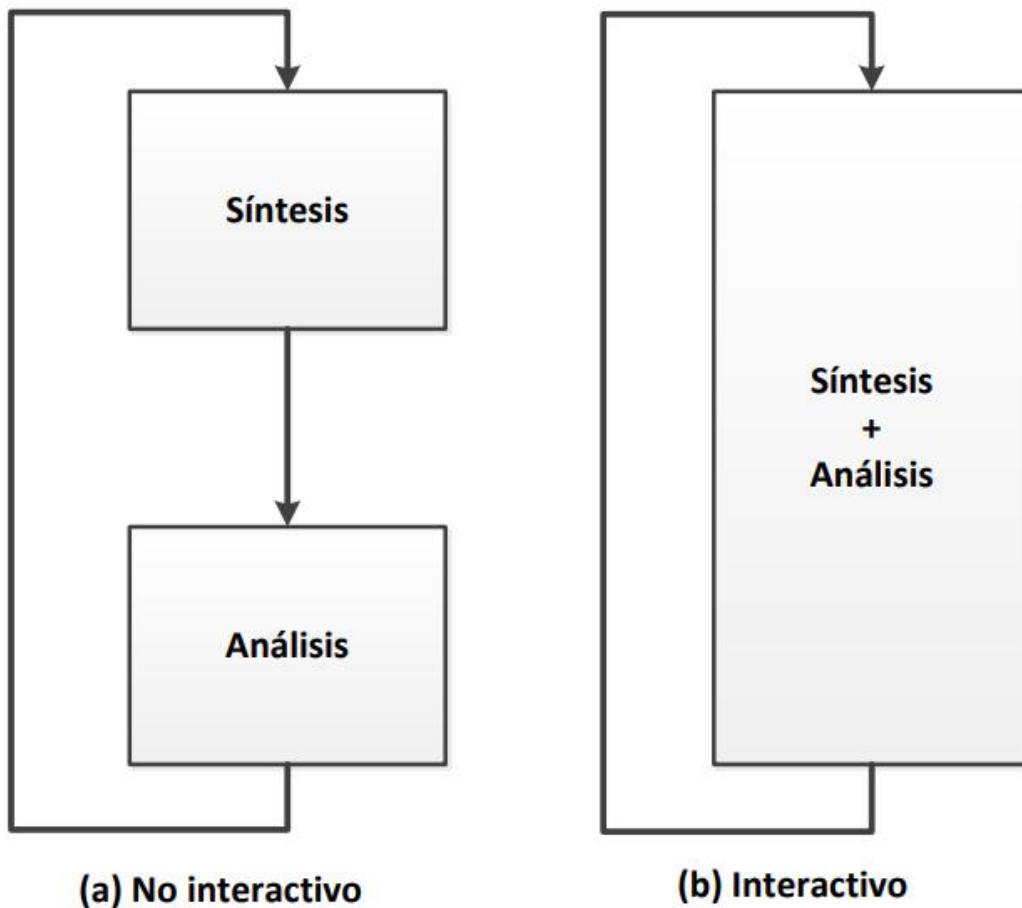
El marco teórico que fundamenta esta investigación proporcionará al lector una idea más clara acerca de este tema. Se encontrarán los conceptos muy básicos, los complementarios y específicos.

##### **2.1. Fundamento teórico de la investigación**

###### **2.1.1. Herramienta interactiva**

“Una herramienta interactiva es esencialmente una colección de ventanas gráficas que son manipuladas solamente usando el ratón. Si el estudiante cambia algún elemento activo en la ventana, se vuelve a re-calcular y a presentar de forma automática. De esta manera, se percibe cómo las modificaciones afectan al resultado. Las herramientas interactivas no son efectivas solamente en presentaciones de conceptos teóricos en clase, sino que también son beneficiosas para ampliar la experiencia del usuario en tareas de análisis y diseño.”  
(Rodríguez, 2012)

A la hora de llevar a cabo el diseño de un sistema es necesario seguir un proceso iterativo compuesto de dos fases:



**Gráfico 2.1: Paradigma de diseño.**

Fuente: (Rodríguez, 2012)

Gracias al tipo paradigma de diseño interactivo es posible realizar fases de síntesis y análisis de forma conjunta, de modo que cuando un parámetro es modificado su efecto se ve reflejado inmediatamente, de esta forma el proceso de diseño es realmente dinámico y se logra percibir el cambio del criterio de comportamiento con respecto a los elementos que manipula.

### **2.1.2. Generación de código**

“La generación de código en el mundo se ha hecho algo imprescindible, pues el ahorro de tiempo, la eficiencia en la programación y la estandarización de código son los pilares fundamentales para la construcción de un proyecto.

En términos más generales, la generación de código es usada para construir programas de una manera automática evitando que los programadores tengan que escribir el código a mano. Constituyendo un ahorro de tiempo en el desarrollo de proyectos y aplicaciones.” (Cepeda, 2002)

La simplicidad y el rendimiento en la web, es muy importante, por lo cual la generación de código ayuda a que un proceso específico en el desarrollo y ejecución sea menos repetitivo.

### **2.1.3. Colores Web**

Los colores web son aquellos colores que aparecen en un sitio web. Se pueden basar sobre los sistemas de color RGB o HSL. En el código CSS son especificados como valores numéricos.

“El concepto de colores en HTML es aquel que se utiliza para designar al conjunto de colores y tonalidades existentes o elegidas para decorar, pintar, colorear algún objeto o superficie.”(Bembibre, n.d.)

Cuando se diseña el prototipo de un sitio o una aplicación web es importante tener definida una paleta de color para crear una imagen coherente. Hay muchas formas de crear una paleta de color y guardarla para su reutilización.

### **2.1.4. Variables CSS**

#### **2.1.4.1. Variables dinámicas**

“En programación, una variable representa a un espacio de memoria en el cual se puede almacenar un dato.” (CarlosPes.com, 2017.)

“Existen una gran variedad de aplicaciones en las que no es posible conocer a priori la dimensión de los datos que serán necesarios, por lo que en dichos casos es necesario utilizar tipos de datos dinámicos. El inconveniente es que, muy posiblemente, se ocupará memoria que puede ser necesaria para otras variables. Las estructuras dinámicas se caracterizan porque la reserva de memoria se realiza durante la ejecución del programa pudiendo reservar y liberar posiciones de memoria según sea necesario durante la ejecución del mismo.”  
(Morales, 2016)

#### **2.1.4.2. CSS**

“El desafío en el CSS moderno tiene poco que ver con solucionar errores transitorios de navegador. El desafío ahora es utilizar las características de CSS que tenemos de una manera creativa, con el fin de crear soluciones mantenibles, flexibles, ligeras y, en la medida de lo posible, compatibles con los estándares.”(Verou, 2015)

“Una crítica popular de CSS es que "no se pueden construir componentes que se escalan de manera eficiente". Esto, mis amigos, es una mentira absoluta. Gracias a metodologías CSS, hay una gran cantidad de opciones sólidas y probadas para hacer exactamente eso. Incluso puede importar estos componentes CSS a sus componentes web si lo desea.”(Bell, 2018)

#### **2.1.4.3. Propiedades personalizadas CSS (Variables CSS)**

“Las propiedades personalizadas de CSS (tal vez más fácilmente entendidas como variables CSS) nos proporcionan maneras de hacer que el código sea más conciso, así como introducir nuevas formas de trabajar con CSS que antes no eran posibles. Pueden hacer lo que las variables del preprocesador realizan pero también con muchas ventajas más. Si ha sido un fanático de la naturaleza declarativa de CSS o prefiere manejar la mayor parte de la lógica de su estilo en Javascript, las Propiedades personalizadas aportan algo a la mesa para todos.”  
(Wilson, 2017)

“Aregar la capacidad de cambiar variables en el tiempo de ejecución no solo abre la puerta a cosas como la temática dinámica de la aplicación, sino que también tiene importantes ramificaciones para el diseño receptivo y el potencial para cumplir las futuras funciones de CSS.” (Coyier, 2016)

El leguaje de presentación CSS en la actualidad cuenta con variables nativas con un buen soporte para los navegadores web a excepción de en navegador Internet Explorer.

#### **2.1.5. Desarrollo web front-end**

“El desarrollo frontal o del lado del cliente es una disciplina de Internet relativamente oscura. Históricamente, este rol se ha conocido bajo varios alias, htmler, diseñador web, codificador, frontend, etc., pero sus funciones principales siguen siendo las mismas mientras se expanden con el progreso de Internet. Es un rol de bisagra que requiere sensibilidad estética y rigor programático.

Para muchas personas, el desarrollo del lado del cliente puede percibirse como "hacer las cosas bonitas" y, si bien es un comentario válido ya que las cosas se ven bien, ya que las cosas de buen aspecto se venden mejor, hay muchas otras tecnologías que entran en este campo que normalmente se puede pasar por alto.” (Codesido, 2009)

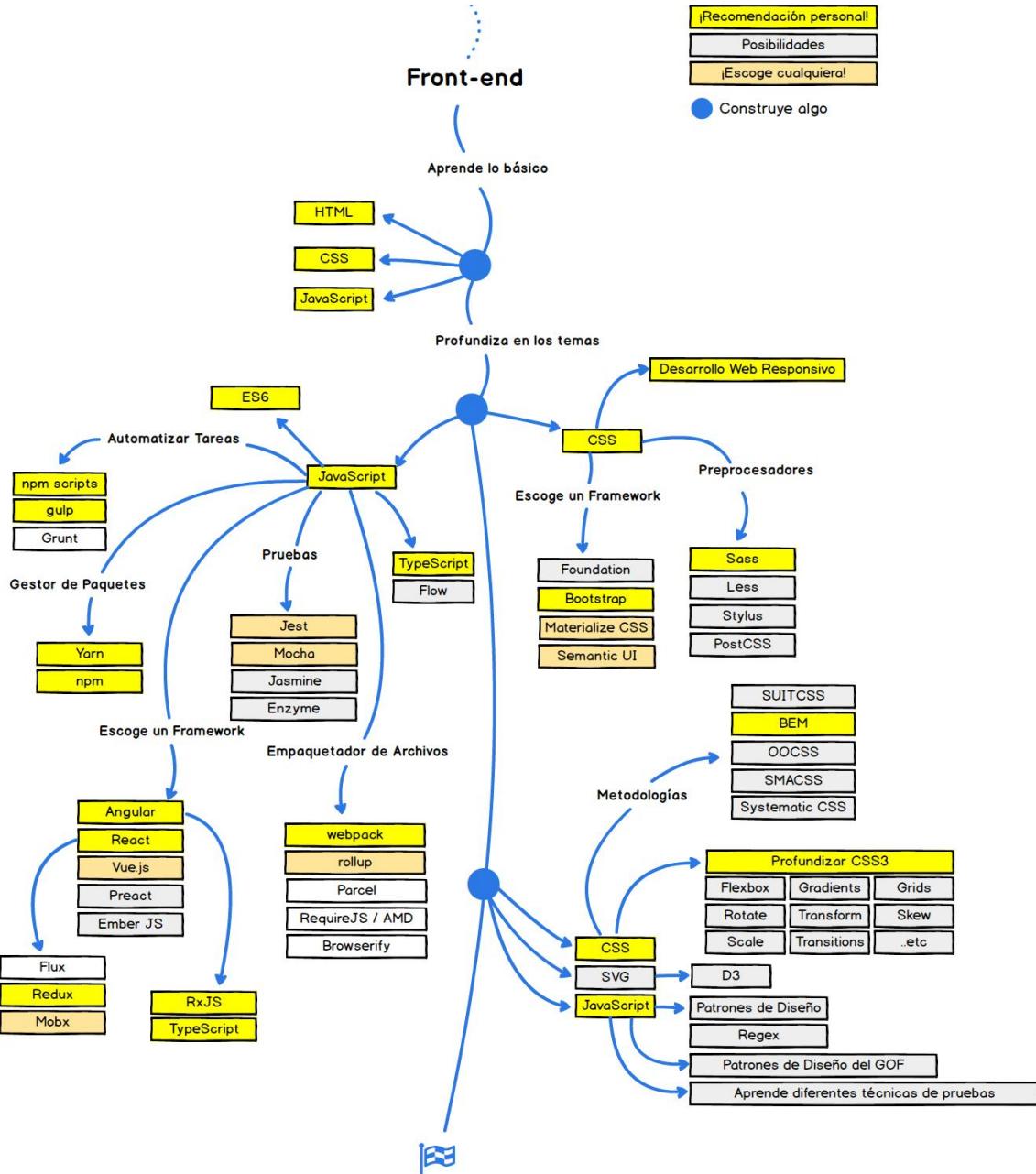


Gráfico 2.2: El Camino del Front-End.

Fuente: (Martin Mucito, 2018)

El objetivo de un desarrollador de front-end es crear páginas e interfaces claras, fáciles y rápidas que harán que la gente entienda y se preocupe por la información, al ponerla en contexto, exponer su legitimidad o falta de ella, y revelar su interconexión implícita o explícita.

El front-end no es solo la estructura y diseño visual, es la interfaz amigable y de futuro del desarrollo web.

#### **2.1.6. Simplicidad**

“Se puede considerar la simplicidad como la eliminación de elementos innecesarios, es decir, reducir algo a su mínima expresión –lo que muchos llaman esencia-. Por consiguiente, no es un estilo del diseño, de crear, de pintar, etc. sino una forma de abordarlo.” (Cadena, 2017)

“Después de tantos años revisando el código, todo lo que puedo decir es que por favor no compliques demasiado las cosas. Introducir la complejidad donde sea necesario. Apunta a la simplicidad, pero nunca a costa de la legibilidad.” (Drasner, 2019)

#### **2.1.7. Rendimiento**

El desarrollador web front-end tiene en mente, utilizar herramientas y técnicas disponibles para alcanzar una buena automatización de tareas y optimización en el rendimiento de un sitio web, manipulando HTML, CSS y JavaScript para asegurar que el sitio web sea más ligero y cargue rápidamente.

### **2.2. Marco metodológico**

#### **2.2.1. Tipo de investigación**

Teniendo en cuenta el punto de vista de Creswell (2003), la investigación puede ser de los siguientes tipos:

- a) Por la forma en que la investigación es usada: Básica o Aplicada.
- b) Por el propósito del estudio: Exploratoria, Descriptiva, Explicativa o Correlacional.
- c) Por la técnica de recolección de datos: Cualitativa, Cuantitativa o Mixta.

- d) Por el tiempo en evaluación de la investigación: Transversal, Longitudinal o Estudio de Caso.

Para el presente proyecto se ha determinado que el tipo de investigación es:

- a) Por la forma en que la investigación es usada: "La investigación Fundamental o Básica, es la que tiene como objetivo esencial, ampliar, intensificar y aclarar los campos de la ciencia. La investigación Aplicada o Técnica tiende a la resolución de problemas o al desarrollo de ideas, dirigidas a conseguir innovaciones, mejoras de procesos o productos, etc." (Sánchez, 2004).

Esta tesis de grado desea dar solución al problema de la investigación planteado en el presente documento, a su vez, mejorar la simplicidad de las variables y el rendimiento de un sitio web. Por lo tanto, el presente proyecto es considerado como una investigación **APLICADA**.

- b) Por el propósito del estudio: "Los estudios exploratorios se efectúan cuando es necesario examinar un tema o un problema de investigación poco estudiado, del cual se tienen muchas dudas o no se ha abordado antes. Los estudios descriptivos buscan especificar las propiedades, las características y los aspectos importantes del fenómeno que se somete a análisis. Los estudios explicativos están dirigidos a encontrar las causas de los eventos, sucesos y fenómenos físicos o sociales. Los estudios correlacionales tienen como objetivo evaluar la relación que existe entre dos o más conceptos, categorías o variables. Tales correlaciones se expresan en hipótesis sometidas a prueba." (Gómez, 2006)

El presente trabajo desea medir la relación entre las variables identificadas (Dependiente e Independiente). Para ello, se ha formulado la hipótesis que han sido contrastada y verificada (La contrastación de la hipótesis se encuentra más adelante en el documento). Por lo tanto, el presente trabajo es considerado como una investigación **CORRELACIONAL**.

- c) Por la técnica de recolección de datos: “Por un lado la investigación cuantitativa es un método científico que conlleva la formulación hipotética, reflexión objetiva, recopilación de datos, análisis de datos y aceptación o rechazo de las hipótesis. Sin embargo, la investigación cualitativa a la investigación que produce datos descriptivos: las propias palabras de las personas y la conducta observable. Las investigaciones mixtas representan la combinación entre los enfoques cualitativo y cuantitativo”. (Antonovica, 2012)

Se ha determinado que el presente trabajo corresponde a una investigación **MIXTA**, debido a que se ha realizado una recolección de datos y análisis cualitativo (observaciones de los procesos), como cuantitativo (elaboración de indicadores y números estadísticos para la contrastación de hipótesis).

- d) Por el tiempo en evaluación de la investigación: “Las investigaciones transversales o seccionales obtienen la información del objeto de estudio (población o muestra) una única vez en un momento dado. La investigación longitudinal compara datos obtenidos en diferentes oportunidades o momentos de una misma población, con el propósito de evaluar cambios. Por último, una investigación de Estudio de Casos analiza una unidad específica de un universo poblacional”. (Torres, 2006)

Debido a que la información expuesta en el presente trabajo se ha recogido únicamente después de haber implementado la solución informática (a través de la herramienta interactiva de generación de código), se determina que el presente trabajo es de tipo **TRANSVERSAL**.

### **2.2.2. Nivel de investigación**

El nivel de investigación está relacionado con el grado de profundidad y alcance que se pretende con la misma. Conforme a los propósitos de la presente tesis de grado, se ha determinado que el nivel de investigación es **CORRELACIONAL**

debido a que se pretende probar la Hipótesis formulada en el trabajo y medir así, la relación de las variables dependiente e independiente.

### **2.2.3. Método**

En la presente investigación se utilizó el método empírico; es decir, ha sido realizado a través de la observación, medición y experimento.

### **2.2.4. Diseño**

Para el diseño de la investigación, emplearemos el de una investigación por objetivos conforme al siguiente esquema:

Donde:  $O_I \Rightarrow C_F$

$O_I$  = Objetivo de la investigación

$C_F$  = Conclusión final

### **2.2.5. Población y muestra**

#### **POBLACIÓN:**

El universo de la investigación está constituido por los colores CSS definidas en variables CSS de un sitio web.

Los colores CSS según la especificación de la W3C, se rigen mediante el modelo RGB, que fusionan tres colores básicos (rojo, verde y azul). Cada color admite 256 niveles de intensidad. Lo que quiere decir que esta población está conformada por los  $256 \times 256 \times 256 = 16.777.216$  colores distintos posibles a utilizar en variables CSS. Para hacer referencia a un color concreto, basta con indicar las intensidades de cada uno de los tres colores básicos.

### **MUESTRA:**

Para calcular el tamaño de la muestra se utilizará el tipo de muestreo por conveniencia, debido a una extracción de población finita, y las sub-población es de un tamaño fijado de antemano, se seleccionará un color CSS y luego de ese color CSS obtener sus variaciones de luminosidad del 0% al 100% y de sus transparencia del 0 % al 100% mediante el modelo HSL y HSLA para un mejor control programático.

Por tanto el tamaño de la muestra es la suma del color escogido, 101 luminosidades y las 101 transparencias, la muestra son 203 colores CSS en total.

### **2.2.6. Operacionalización de variables de la Hipótesis**

#### **VARIABLE INDEPENDIENTE:**

x = Implementación de una herramienta interactiva de generación de código, que define colores con variables CSS.

#### **VARIABLE DEPENDIENTE:**

y = Simplicidad en las variables y rendimiento en un sitio web.

VARIABLES	INDICADORES
<b>VARIABLE INDEPENDIENTE:</b>  (x) = Implementación de una herramienta interactiva de generación de código, que define colores con variables CSS.	
	Cantidad de variables CSS declaradas para un color, con el fin utilizarlo para aplicar claridad, oscuridad, y transparencia del color.
<b>VARIABLE DEPENDIENTE:</b>  (y) = Simplicidad de variables y rendimiento en un sitio web.	Cantidad de procesos al declarar variables CSS de un color, con el fin utilizarlo para aplicar claridad, oscuridad, y transparencia del color.
	Tiempo de descarga del recurso en un ejemplo real.
	Cantidad de Bytes de descarga del recurso en un ejemplo real.

**Tabla 2.1: Operacionalización de variables**

**Fuente: Elaboración propia**

## **2.3. Técnicas, instrumentos de la recolección de datos**

### **2.3.1. Técnica de la recolección de datos**

La técnica utilizada en la investigación para recolectar la información fue:

- **Diagnóstico:** El diagnóstico se realizó 15 días después de que la herramienta interactiva de generación de código fue colocado en producción. El objetivo fue analizar las mejoras que se han identificado en la implementación de la herramienta interactiva, a fin de poder comprobar la simplicidad en las variables CSS y el rendimiento en un sitio web de las hipótesis formulada mediante los siguientes recolección de datos:

- Tamaño de la muestra de la población.
- Diagramas de proceso antes y después de la implementación de la herramienta.
- Ejemplo web de comparación: El objetivo del ejemplo con HTML y CSS es medir la cantidad de variables CSS declaradas con soporte de la herramienta y sin la herramienta.
- Análisis de rendimiento del ejemplo web de comparación: En este apartado se realizará el análisis de rendimiento en la ejecución de los ejemplos de la comparación.

### **2.3.2. Instrumento de recolección de datos**

El instrumento utilizado fue una Evaluación. La Evaluación es un instrumento de investigación que posee un alto grado científico y objetivo. Las preguntas de evaluación articulan los indicadores y fue la razón principal de realizar los diagnósticos, estas serán resueltas una vez se haya concluido el análisis y síntesis del trabajo de investigación.

Para obtener los resultados en el diagnóstico de la evaluación se utilizará las siguientes tecnologías:

- **Codepen:** es una comunidad en línea para probar y mostrar fragmentos de código HTML, CSS y Javascript creados por el usuario. Funciona como un editor de código en línea y un entorno de aprendizaje de código abierto, donde los desarrolladores pueden crear fragmentos de código, llamados creativamente "bolígrafos", y probarlos.
- **ChromeDevTools:** (Panel Elements) Usa el panel Elements para iterar la distribución y el diseño de tu sitio mediante la libre manipulación de DOM y CSS.
- **GithubPages:** Es un servicio de alojamiento de sitios estáticos diseñado para alojar sus páginas personales, de organización o de proyecto directamente desde un repositorio de GitHub.  
GithubPages se utilizar para alojar los ejemplos realizados para las pruebas de rendimiento.
- **PageSpeedInsights:**(PSI), informa sobre el rendimiento de una página en dispositivos móviles y de escritorio, y ofrece sugerencias sobre cómo se puede mejorar esa página. PSI proporciona datos de laboratorio y de campo sobre una página. Los **datos de laboratorio** son útiles para depurar problemas de rendimiento, ya que se recopilan en un entorno controlado. Los **datos de campo** son útiles para capturar la experiencia real del usuario en el mundo real, pero tienen un conjunto de métricas más limitado.

El análisis, síntesis y evaluación de resultados se encuentra en el capítulo 4 del presente documento.

## 2.4. Técnicas para la gestión del trabajo

### 2.4.1. Herramientas para la obtención de resultados esperados

- A. La metodología de desarrollo de software utilizada fue OPEN UP, debido a que se ha considerado la metodología ideal para obtener rápidos

resultados siguiendo el lineamiento de un entorno colaborativo en el desarrollo de software.

- B. El lenguaje de programación utilizado para el desarrollo del software es Javascript (EcmaScript) en conjunto con el framework VueJs.
- C. Se realizaron prototipos(maquetas) de la interfaz de la herramienta interactiva para tener una mejor visión, evaluación y demostración para una mejor la implementación del software y mejoras en el desarrollo.
- D. Los diagramas de procesos y los prototipos para la Herramienta interactiva, se realizaron mediante la plataforma web de Lucidchart ([lucidchart.com](http://lucidchart.com)), junto con la plataforma web MockFlow ([mockflow.com](http://mockflow.com)).
- E. Se verificó que los prototipos guarden relación con los procesos soportados por la alternativa de solución propuesta.

#### **2.4.2. Definición de conceptos técnicos**

##### **A. MVVM**

Es un patrón arquitectónico de software que facilita la separación del desarrollo de la interfaz gráfica de usuario, ya sea a través de un lenguaje de marcado o código GUI, a partir del desarrollo de la lógica de negocios (el modelo de datos). El modelo de vista de MVVM es un convertidor de valores, lo que significa que el modelo de vista es responsable de exponer (convertir) los objetos de datos del modelo de tal manera que los objetos se administran fácilmente. A este respecto, el modelo de vista es más un modelo que una vista y maneja la mayoría, si no toda la lógica de visualización de la vista.

##### **B. Editor de código fuente**

El editor de código fuente, tiene las características diseñadas exclusivamente para simplificar y acelerar la escritura de código fuente, como resultado de sintaxis, autocompletado y pareo de llaves.

También proveen un modo conveniente de ejecutar un compilador, un intérprete, un depurador, o cualquier otro programa que sea relevante en el proceso de desarrollo de software.

### **C. Vue.js**

Es una arquitectura de adopción incremental que se enfoca en la representación declarativa y la composición de componentes. Las funciones avanzadas requeridas para aplicaciones complejas como enrutamiento, administración de estado y herramientas de construcción se ofrecen a través de bibliotecas y paquetes compatibles que se mantienen oficialmente.

Por otro lado, Vue.js también es perfectamente capaz de soportar aplicaciones sofisticadas de una sola página (en inglés single-page-application o SPA) cuando se utiliza en combinación con herramientas modernas y librerías compatibles.

### **D. Patrón de diseño**

Es el patrón de administración de estado que hace que el componente se base en el singleton de la tienda global, y sirve como un almacén centralizado para todos los componentes de una aplicación, con reglas que garantizan que el estado solo se puede mutar de manera predecible. También pone en vigor algunas opiniones sobre cómo dividir la lógica de la administración del estado en diferentes lugares, pero aun así permite suficiente flexibilidad para la estructura del código actual.

### **E. Prototipo de la interfaz**

Es un modelo que representa una funcionalidad del sistema que interactúa con el usuario. Es utilizado para la demostración, evaluación del diseño y otros fines.

# **CAPÍTULO III.**

**DESARROLLO DE LA PROPUESTA DE  
INVESTIGACIÓN**

## CAPÍTULO III

### DESARROLLO DE LA PROPUESTA DE INVESTIGACIÓN

#### 3.1. Método de desarrollo de la propuesta

##### 3.1.1. Open unified process (open up)

Para la construcción del software se ha utilizado la metodología ágil OPEN UP. Esta metodología es idónea porque adopta un enfoque pragmático, con una filosofía ágil que se centra en la naturaleza colaborativa del desarrollo de software. Es un proceso que solo incluye el contenido fundamental pero que es completa en el sentido de que manifieste por completo el proceso de construir un sistema.

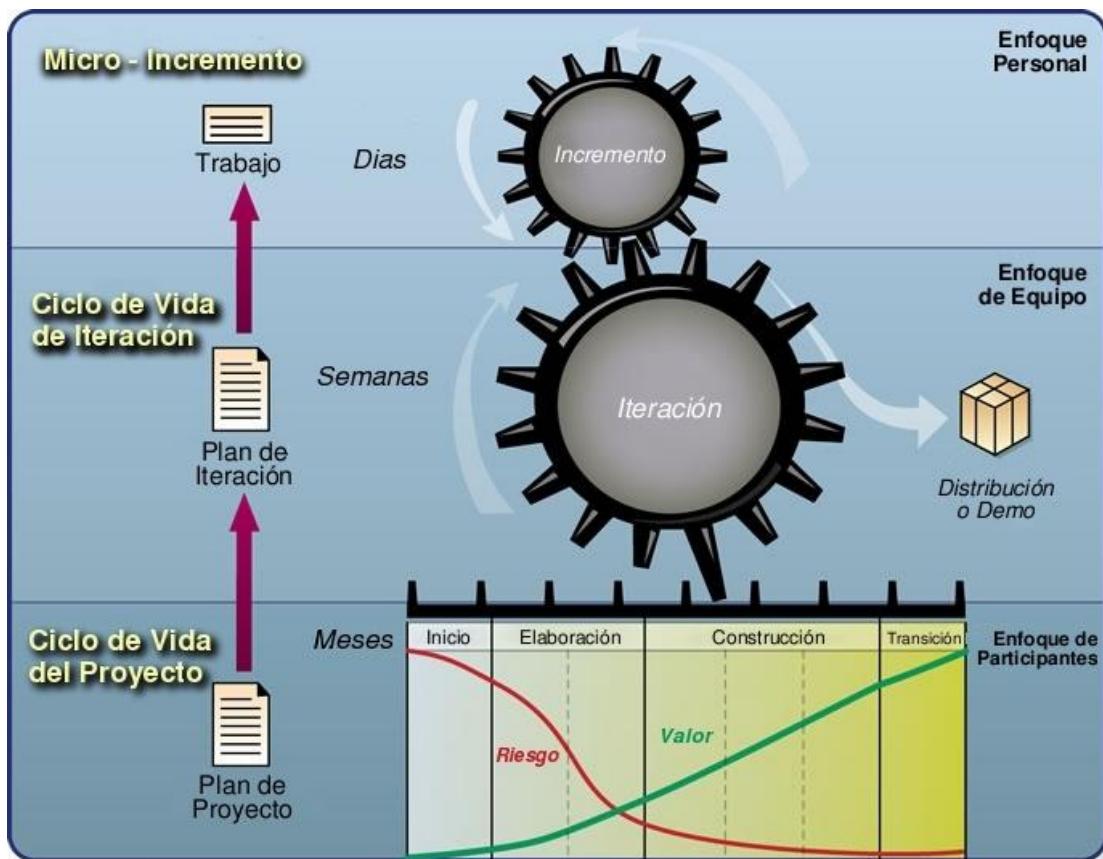


Gráfico 3.1: Las tres capas y flujo de trabajo del proceso OpenUP.

Fuente: Ricardo Balduino (2012).

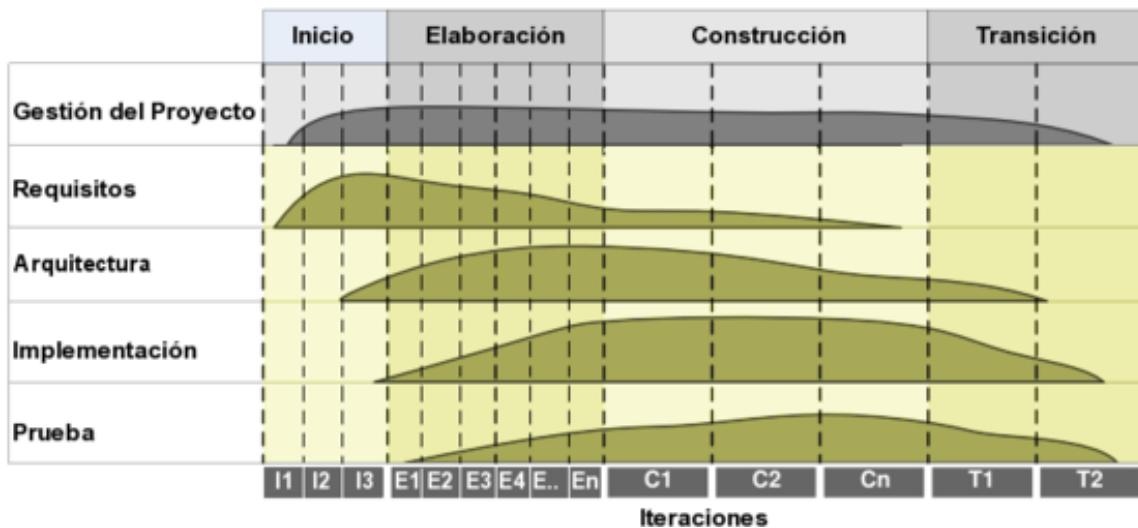
Open UP se basa en los siguientes principios:

- A. Colaborar para sincronizar intereses y promover prácticas que impulsan un ambiente de equipo saludable y desarrollan un conocimiento compartido del proyecto.
- B. Equilibrar las prioridades para maximizar el beneficio obtenido por los interesados del proyecto.
- C. Centrarse en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo.
- D. Desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo.

De esta manera, es posible ir probando el software desde el principio en acorde al cumplimiento del alcance del proyecto. A continuación se describirán las fases del ciclo de vida planteadas por la metodología y que se tuvieron en consideración para la realización del proyecto.

- **Interacción Inicial:** En esta fase se prepara el plan del proyecto, se definen los objetivos generales y específicos se recopilan los requerimientos y también determinamos la factibilidad de la arquitectura del sistema a través de la construcción de prototipos.
- **Iteración de Elaboración:** Es en esta fase donde se establecen los requisitos y la arquitectura estable del sistema.
- **Iteración de Construcción:** En esta fase el sistema es construido en base a las especificaciones realizadas en la fase anterior. Estas implementaciones se realizan en una serie de iteraciones cortas y limitadas en el tiempo, teniendo en cuenta que el resultado de cada iteración es una versión ejecutable del software.

- **Iteración de Transición:** Fase final donde se realiza el despliegue del sistema hacia los usuarios finales. La retroalimentación recibida permite afinar el sistema en las sucesivas iteraciones.



**Gráfico 3.2: Ciclo de vida Open Unified Process (Open UP).**

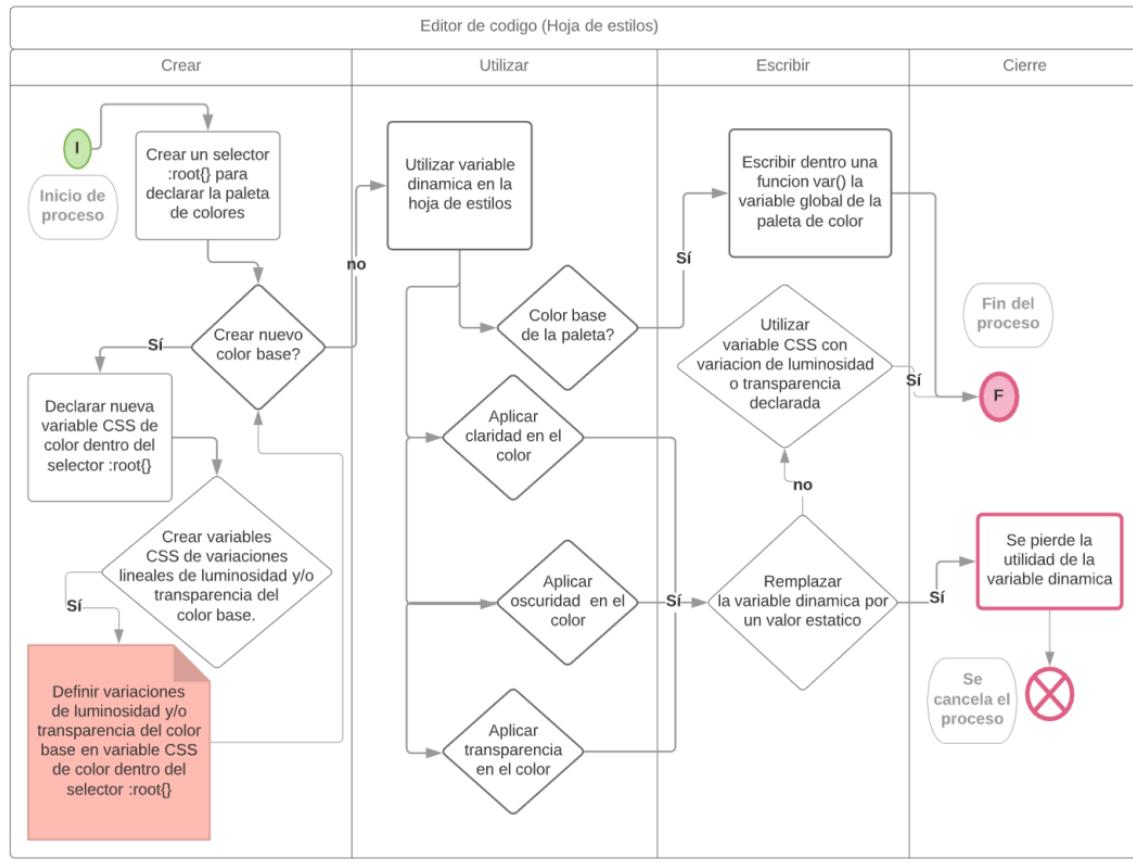
Fuente: Ricardo Balduino, 2012

### 3.2. Diagrama de procesos

#### 3.2.1. Diagrama de proceso general antes de la implementación

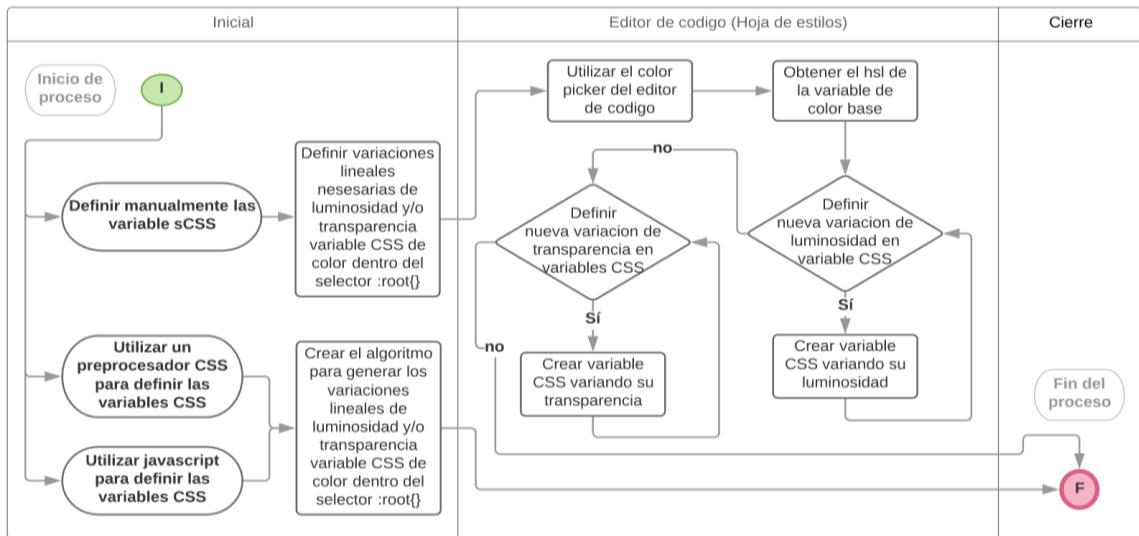
La información detallada a continuación fue recopilada utilizando distintos formatos y análisis de los procesos de rutinarios en el desarrollo web front-end, los cuales se consideran dentro del método de observación para analizar el comportamiento y flujo de los procesos a automatizar.

A continuación se procederá presentar el diagrama de procesos que engloba el proceso general de la definición y uso de variables CSS de color.



**Gráfico 3.3.1: Diagrama de procesos sin la herramienta interactiva. (General)**

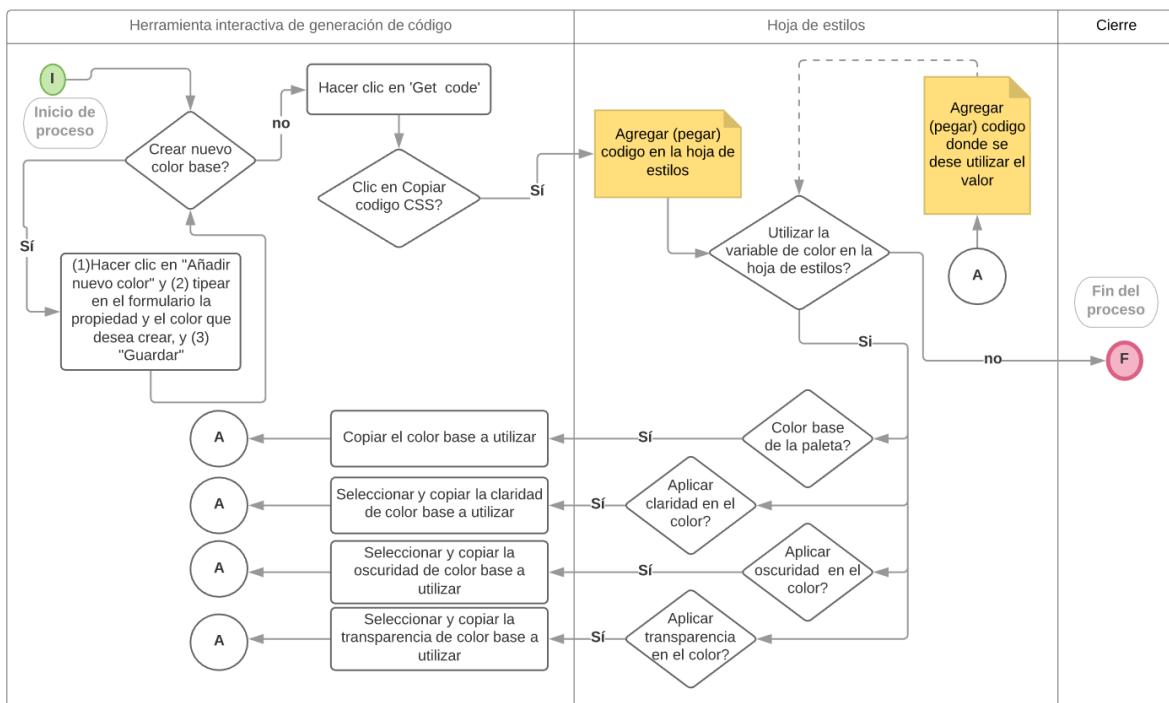
**Fuente:** Elaboración propia.



**Gráfico 3.3.2: Diagrama de proceso sin la herramienta interactiva.(Parcial)**

Fuente: Elaboración propia.

### 3.2.2. Diagrama de proceso con la herramienta interactiva.



**Gráfico 3.4: Diagrama de proceso con la herramienta interactiva.**

Fuente: Elaboración propia.

### **3.3. Arquitectura**

A continuación, se describirán los componentes que conforman la herramienta interactiva, la interacción entre ellos y el patrón de diseño que guía el desarrollo del proyecto. La arquitectura presente en el sistema se compone de tres capas, separando la capa de presentación de la capa de la lógica del negocio, y la capa de datos. La capa de presentación es la que visualizan los usuarios, también es denominada capa de usuario. Es la capa encargada de capturar la información y se comunica directamente con la capa del negocio.

La capa de negocio o lógica de negocio es donde se establecen todas las reglas que deben cumplirse. Se encarga de recibir las peticiones de usuario para su posterior presentación de resultados. La capa de datos es la encargada de almacenar la data y acceder a la misma. En esta capa se reciben las solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

#### **3.3.1. Componentes principales**

La herramienta interactiva está conformada por tres principales capas: cliente, servidor y datos. El cliente representa al navegador web, el servidor representa al servidor donde está publicado la herramienta interactiva y los datos representan a los datos locales.

- **Cliente:** El presente proyecto está conformado por un cliente web, ya sea el navegador de un ordenador o de un dispositivo móvil (La herramienta Interactiva CSSColorVars ha sido desarrollado para ejecutarse sin problemas desde cualquier tipo de dispositivo). Estas plataformas constituyen las formas de acceso a la herramienta interactiva que se comunican a través de un navegador web (Chrome, Firefox, Safari, Edge).
- **Servidor:** El servidor web almacena los componentes de la aplicación web, además de disponer de los servicios requeridos para el correcto funcionamiento de la aplicación.

Actualmente se utiliza un servidor web proporcionado por GitHubPages es un servicio de alojamiento de sitios diseñado para alojar sus páginas personales, de organización o de proyecto directamente desde un repositorio de GitHub.

Todos los sitios de GitHubPages, admiten el cumplimiento de HTTPS. HTTPS agrega una capa de encriptación que evita que otros puedan husmear o manipular el tráfico de su sitio.

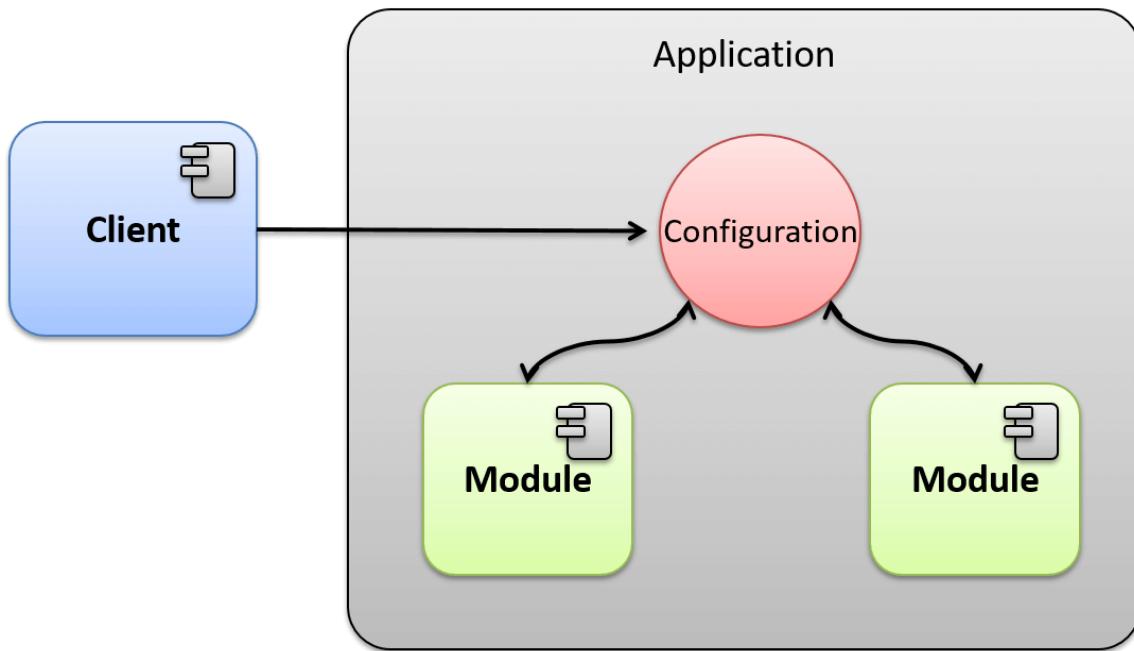
Los sitios de GitHubPages están sujetos a los siguientes límites de uso:

- a) Los repositorios de origen de las páginas GitHub tienen un límite recomendado de 1 GB.
  - b) Los sitios de GitHubPages publicados no pueden tener más de 1 GB.
  - c) Los sitios de GitHubPages tienen un límite de ancho de banda flexible de 100 GB por mes.
  - d) Los sitios de GitHubPages tienen un límite suave de 10 compilaciones por hora.
- **Datos:** Esta capa es la encargada de almacenar la información. El almacenamiento local es un tipo de la API de almacenamiento web, que es una API para almacenar pares de datos clave-valor dentro del navegador.

A diferencia de las cookies, el límite de almacenamiento es mucho mayor (hasta 10 MB) y la información nunca se transfiere al servidor.

### 3.3.2. Patrón de diseño

- **Singleton:** Gestiona la configuración global de la aplicación mediante un objeto único implementado con el patrón Singleton, dicho objeto podrá ser compartido y modificado por todos los módulos.



**Gráfico 3.5: Patrón de diseño Singleton.**

Fuente: ([reactiveprogramming.io](https://reactiveprogramming.io)).

Vuex es una biblioteca de patrones de administración de estado para aplicaciones Vue.js. Este patrón hace que el componente se base en el singleton de la tienda global. Cuando se usa un sistema de módulos, se requiere importar la tienda en cada componente que usa el estado de la tienda.

### 3.3.3. Patrón arquitectónico

- **Modelo vista vista modelo (MVVM):** Este patrón de arquitectura se separan los datos de la aplicación, la interfaz de usuario pero en vez de controlar manualmente los cambios en la vista o en los datos, estos se actualizan directamente cuando sucede un cambio en ellos, por ejemplo si la vista actualiza un dato que está presentando se actualiza el modelo automáticamente y viceversa.

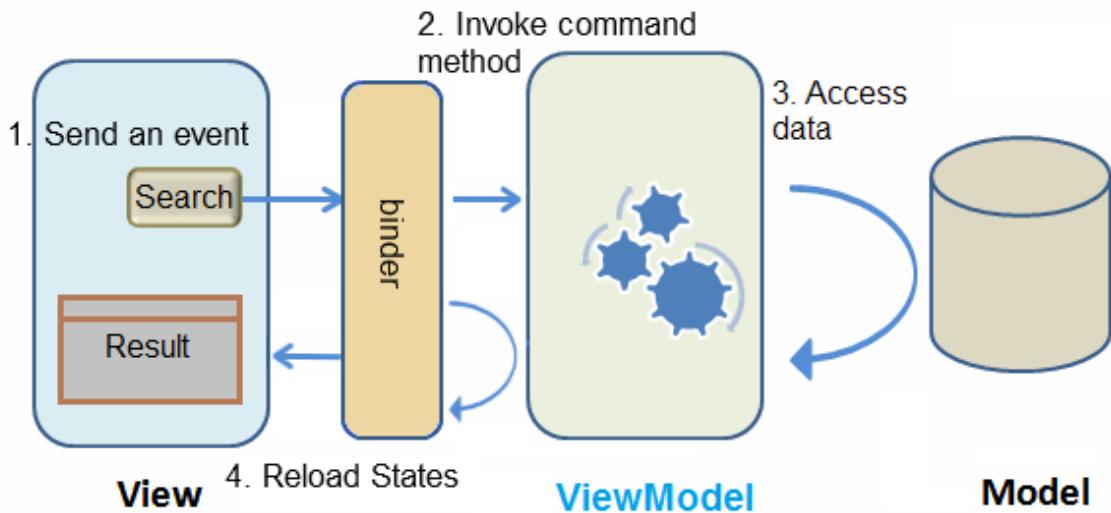


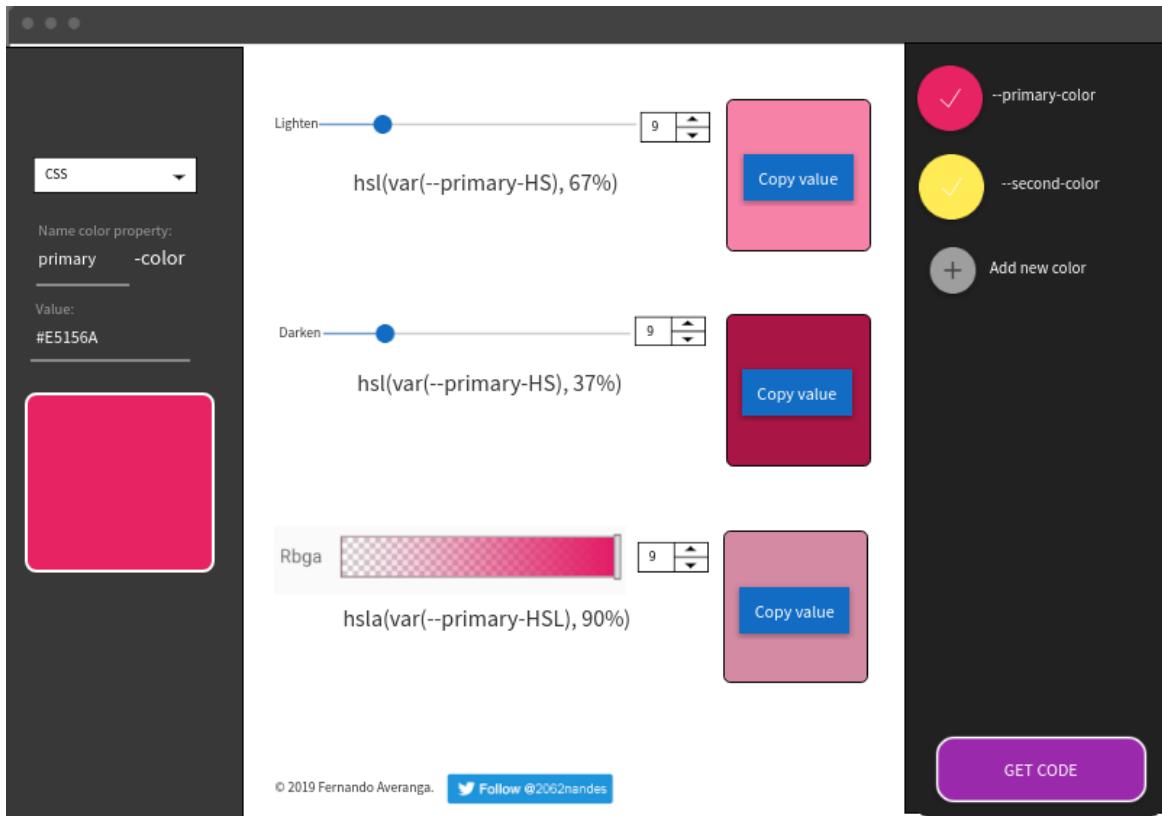
Gráfico 3.6: Patrón de arquitectónico MVVM.

Fuente: ([adictosaltrabajo.com](http://adictosaltrabajo.com)).

MVVM se ha puesto de moda en Javascript por su utilidad para pasar datos desde el código Javascript (modelo) al código HTML (vista), de una manera sencilla, sin tener que tratar directamente con las etiquetas HTML desde Javascript.

### 3.4. Prototipos de interfaz gráfica

En esta sección del proyecto se presentan los prototipos de interfaz gráfica utilizados para implementar el sistema. Para la elaboración de los prototipos se ha utilizado la herramienta online Mockflow ([mockflow.com](http://mockflow.com)).



**Gráfico 3.7: Interfaz principal de la Herramienta interactiva.**

Fuente: Elaboración propia

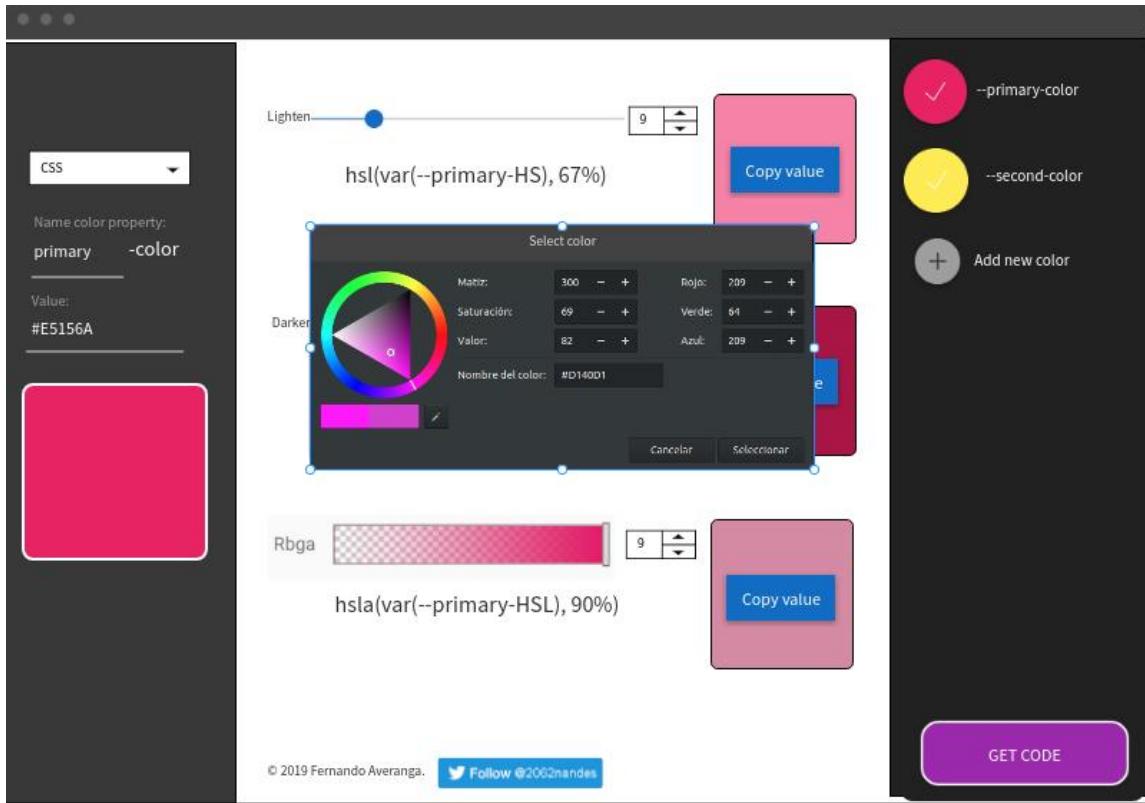


Gráfico 3.8: Cambiar color con Color picker de la Herramienta.

Fuente: Elaboración propia.

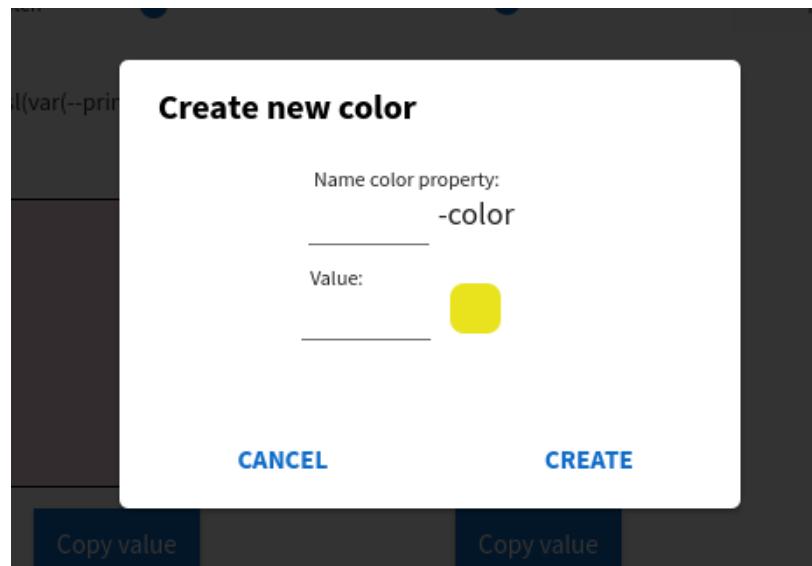


Gráfico 3.9: Crear nuevo color mediante la herramienta.

Fuente: Elaboración propia.

The screenshot shows a dark-themed CSS code editor interface. At the top, there are tabs labeled 'CSS' and 'SNIPPETS'. Below the tabs, the main area contains the following CSS code:

```
:root {  
    --first-color: #e91e63;  
    --first-HS: 340, 82%;  
    --first-RGB: 233, 30, 99;  
    --second-color: #ffeb3b;  
    --second-HS: 54, 100%;  
    --second-RGB: 255, 235, 59;  
}
```

In the bottom right corner of the code area, there is a small white button with the text 'COPY CODE' in black. To the right of the code area, there is a purple button with the text 'GET CO'.

**Gráfico 3.10: Código generado listo para copiar y utilizar en la hoja de estilos**

**Fuente: Elaboración propia.**

### **3.5. Construcción**

En esta sección, se mostrarán en detalle las diversas tecnologías utilizadas para la construcción del software así como también la implementación de la herramienta interactiva web a través de sectores de código más importantes.

#### **3.5.1. Framework**

El Framework utilizado para el desarrollo del presente proyecto es Vue.js. Vue.js es un marco de trabajo progresivo para crear interfaces de usuario. Está diseñado desde el principio para ser adoptable de forma incremental, y puede escalarse

fácilmente entre una biblioteca y un marco de trabajo en función de los diferentes casos de uso. Las grandes ventajas de este framework son:

- Accesible.
- Versátil: Su núcleo es bastante pequeño y se escala a través de plugins, con lo cual escucharás mucho que Vue es una librería muy parecida a React, una librería que cumple un propósito.
- Escalable por el mismo tema de la versatilidad.
- Reactivo.
- Optimizado: Su núcleo ocupa 74KB, como ves es bastante liviano.
- Comunidad: Va creciendo a un ritmo importante con más 130858 estrellas en GitHub y 266 personas contribuyendo al core cada día.
- Licencia MIT: se publicó bajo el amparo de esta licencia.

### **3.5.2. Herramientas, plataformas y tecnología**

A continuación se listaran todas aquellas herramientas de software, plataformas y tecnologías que se emplearon para el desarrollo de la Herramienta interactiva de solución web. Todas las herramientas usadas pertenecen al grupo de código abierto.

- Plataforma: web
- Lenguajes – scripting/web: EcmaScript, HTML5, Sass, Vue.
- Servidor de Aplicaciones: GitHubPages.
- Frameworks y Librerias: Vuejs, ColorJS, Vuetify.
- Modelo de Programación: Reactiva.
- Almacén de datos: LocalStorage, Vuex.
- Herramientas de Desarrollo: Visual Studio Code, NPM, VueCLI 3, VueUI, Git y GitHub.

### **3.6. Implementación**

Muchas implementaciones son dadas según a una especificación o un estándar. Por ejemplo, un navegador web respeta (o debe respetar) en su implementación, las especificaciones recomendadas según el World Wide Web Consortium.

La estructura de la Single Page Application (SPA) implementada con VueJs se creó progresivamente compuesto por un lado de ficheros \*.vue en el que se define la estructura con la lógica de componentes web (HTML, CSS, Javascript empaquetados por componente), y su intercomunicación declarativamente utilizando datos centralizados en el store de Vuex, con persistencia de datos al almacenamiento web.

Por otro lado en la carpeta /store en ficheros EcmaScript \*.js, en el que se implementó el código de la lógica de la aplicación. Posteriormente, el proyecto desarrollado se compila a producción el cual contendrá el código empaquetado, ofuscado mediante el paquete de módulos el cual genera un HTML dinámico emitido al ejecutar la aplicación.

#### **3.6.1. Algoritmo de la implementación.**

Al definir un color en una variable CSS, por ejemplo:

```
--primary-color: #ff00c22;
```

Se requiere obtener su valor equivalente en modelo HSL en un triplete desacoplado para así poder manipular alguna de sus propiedades, y convertir en constantes:

Modelo Hexadecimal	Modelo HSL
#ff00c22;	hs1(355, 100%, 52%);

Una vez obtenido el modelo HSL se deberá desacoplar el triplete en dos variables CSS complementarias de la siguiente manera:

```
--primary-HS: 355, 100%;
```

```
--primary-HSL: 355, 100%, 52%;
```

Con la definición de estas variables complementarias se pretende mejorar la simplicidad de las variables, para mayor detalle del análisis ver el capítulo 4.

### 3.6.2. Implementación basada en el patrón de diseño y arquitectónica.

En la siguiente gráfica se puede observar carpeta **store**. El store es un almacén centralizado para todos los componentes de la aplicación, con reglas que garantizan que el estado solo se puede mutar de manera predecible.

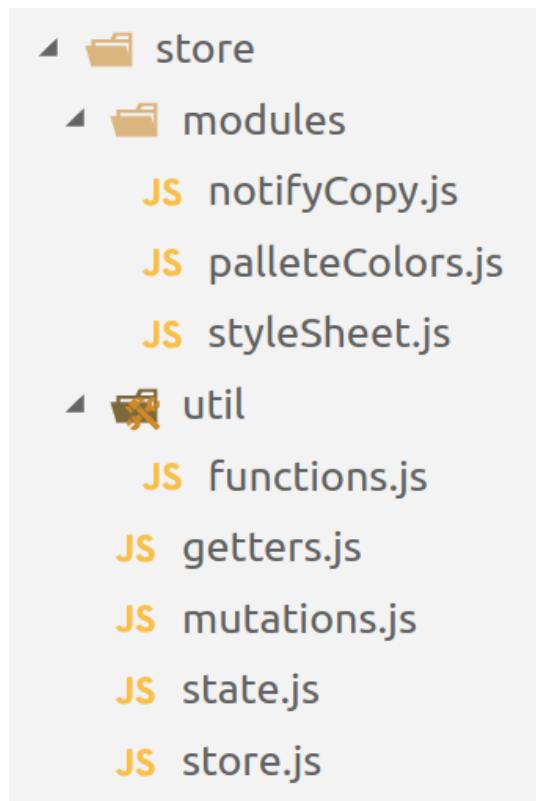


Gráfico 3.11: Almacén centralizado de la lógica de la aplicación

Fuente: Elaboración propia.

El almacén de datos centralizado hace que los componentes .vue se basen en el **patrón de diseño singleton** del store global.

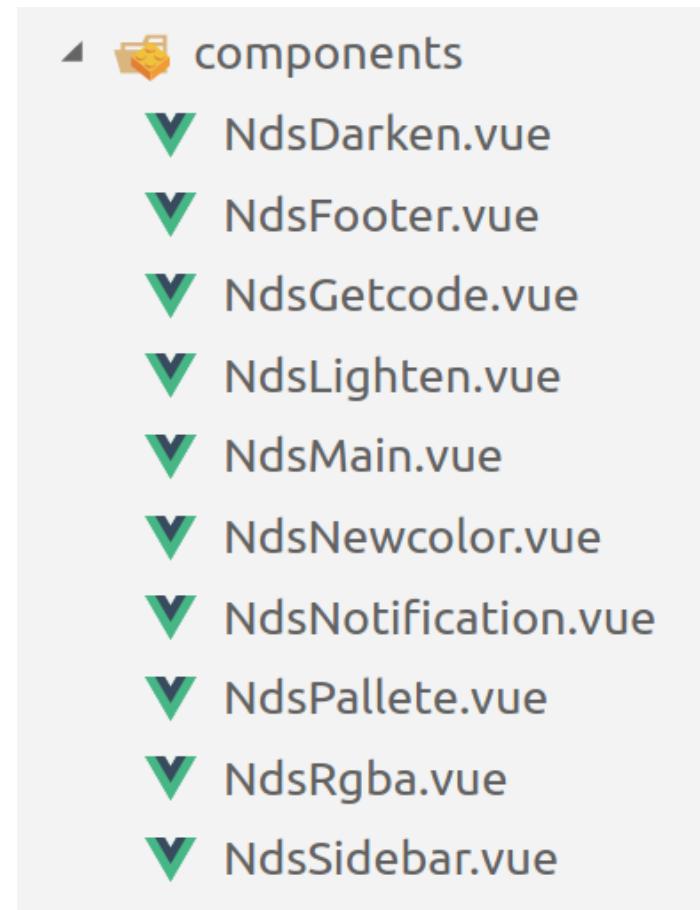
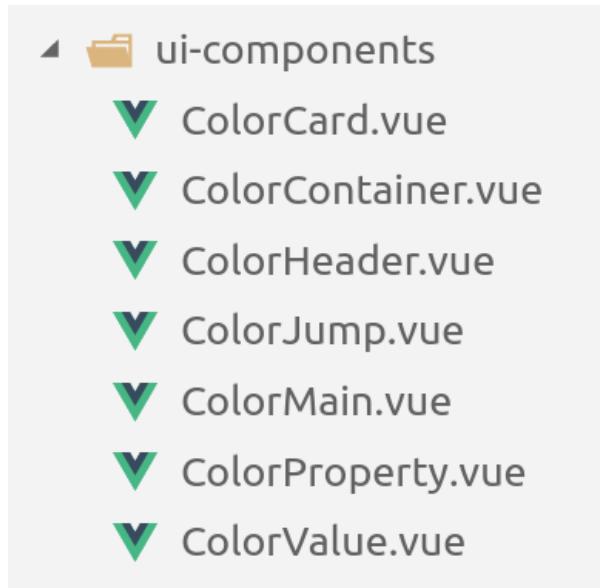


Gráfico 3.12: Componentes basados en el patrón de diseño Singleton global.

Fuente: Elaboración propia.

Cada componente se centra en el modelo de vista, la capa del **patrón arquitectónico Modelo-Vista-Vista-Modelo**. Conecta la Vista y el Modelo a través de enlaces de datos de dos vías. Las manipulaciones reales de DOM y el formato de salida se resumen en Directivas.



**Gráfico 3.13: Componentes reusables para la interfaz de usuario.**

Fuente: Elaboración propia.

Como se ve en el anterior gráfico, los componentes de Vue también está diseñado para ser una capa de vista simple y flexible. en los **ui-components** solo se utilizó para la creación de prototipos, o mezclar y combinar con otras bibliotecas para un apilamiento frontal personalizado.

# **CAPÍTULO IV.**

**ANÁLISIS, SÍNTESIS Y EVALUACIÓN DE  
RESULTADOS**

## CAPÍTULO IV

### ANÁLISIS, SÍNTESIS Y EVALUACIÓN DE RESULTADOS

#### 4. Introducción

En la actualidad existen sitios web reconocidos que utilizan colores con Variables CSS en todo su ecosistema web front-end, alguno de estos sitios web son los siguientes:

 YouTube	 Messenger
youtube.com	messenger.com
 reddit	 twitch
reddit.com	twitch.tv

Gráfico 4.1: Sitios web reconocidos que utilizan Variables CSS.

Fuente: Elaboración propia.

Se analizó el código CSS de estos sitios web y se observó que utilizan una gran cantidad de Variables CSS de color.

El diagrama de procesos antes de la implementación de la herramienta interactiva está basado en los sitios web descritos en la gráfica anterior.

Por lo tanto la técnica e instrumento de la recolección de datos son el diagnóstico y evaluación, que determinará el efecto de la implementación de la herramienta interactiva.

#### **4.1. Diagnóstico**

Se cumplió con la exigencia del diagnóstico mediante el siguiente procedimiento:

En un ejemplo real utilizando HTML y CSS se definió una variable del color aleatorio y se buscó la forma de obtener sus características para aplicar luminosidades y transparencias, finalmente visualizando los colores en pantalla.

##### **4.1.1. Análisis y síntesis para el efecto de simplicidad en las variables.**

###### **1) Tamaño del muestreo**

El tamaño de la muestra por conveniencia es la suma del color escogido, sus 101 luminosidades y 101 transparencias, y **la muestra es 203 colores en total.**

###### **2) Ejemplo sin la herramienta interactiva (antes de la implementación)**

Pasos a realizar en la hoja de estilos CSS:

1. Crear un selector :root{} para declarar variables CSS de color.
2. Crear la variable de color base (aleatoriamente).
3. Declarar la nueva variable CSS de color base dentro del selector :root{}
4. Crear variables CSS de variaciones lineales de luminosidad y/o transparencia del color base:

	<b>Cantidad de procesos para declarar una variable CSS</b>
<b>Color base</b>	1
<b>Luminosidades</b>	101
<b>Transparencias</b>	101
<b>TOTAL</b>	203

**Tabla 4.1: Cantidad de procesos para definir colores con variables CSS según la muestra poblacional.**

**Fuente:** Elaboración propia.

## 5. Utilizar variables CSS dinámicas.

Se creó una estructura en HTML para mostrar los colores en pantalla.

```
:root {  
    --primary-light-1: #9b16f0;  
    --primary-light-2: #d1bf1;  
    --primary-light-3: #f1ff1;  
    --primary-light-4: #a124f1;  
    --primary-light-5: #a329f2;  
    ...  
    ...  
    ...  
    --primary-light-45: #f6e9fe;  
    --primary-light-46: #f8eefe;  
    --primary-light-47: #faf2fe;  
    --primary-light-48: #fcf7ff;  
    --primary-light-49: #fefcf9;  
    --primary-light-50: white;  
    --primary-color: #9911f0;  
    --primary-dark-1: #960fed;  
    --primary-dark-2: #930fe8;  
    --primary-dark-3: #900ee3;  
    --primary-dark-4: #8d0edf;  
    --primary-dark-5: #8a0eda;  
    ...  
    ...  
    ...  
    --primary-dark-45: #10021a;  
    --primary-dark-46: #0d0115;  
    --primary-dark-47: #0a0110;  
    --primary-dark-48: #07010b;  
    --primary-dark-49: #040007;  
    --primary-dark-50: #010002;  
    --primary-transparency-1: rgba(153, 17, 240, 0.01);  
    --primary-transparency-2: rgba(153, 17, 240, 0.02);  
    --primary-transparency-3: rgba(153, 17, 240, 0.03);  
    --primary-transparency-4: rgba(153, 17, 240, 0.04);  
    --primary-transparency-5: rgba(153, 17, 240, 0.05);  
    --primary-transparency-6: rgba(153, 17, 240, 0.06);  
    ...  
    ...  
    ...  
    --primary-transparency-95: rgba(153, 17, 240, 0.95);  
    --primary-transparency-96: rgba(153, 17, 240, 0.96);  
    --primary-transparency-97: rgba(153, 17, 240, 0.97);  
    --primary-transparency-98: rgba(153, 17, 240, 0.98);  
    --primary-transparency-99: rgba(153, 17, 240, 0.99);  
}
```

**Gráfico 4.2: Definición de colores con variables CSS según la muestra poblacional.**

Fuente: Elaboración propia.

La cantidad de pasos del ejemplo real en total son **206**.

La cantidad de **variables CSS declaradas** son **203**.

### 3) Ejemplo utilizando la herramienta interactiva

Pasos a realizar:

1. Ingresar a la herramienta interactiva.
2. Crear nuevo color base
3. Hacer clic en "Añadir nuevo color"
4. Tipear en el formulario la propiedad y el color que desea crear
5. "Guardar"
6. Hacer clic en 'Getcode'
7. Hacer Clic en Copiar código CSS
8. Agregar (pegar) código en la hoja de estilos

```
:root{  
    --primary-color: #9911f0;  
    --primary-HS: 277, 88%;  
    --primary-HSL: 277, 88%, 50%;  
}
```

} 3 variables

9. Utilizar variable dinámica en la hoja de estilos

La cantidad de procesos del ejemplo real, en total son **9**.

La cantidad de **variables CSS declaradas** son **3**.

#### 4) Pantallas de muestra en el navegador

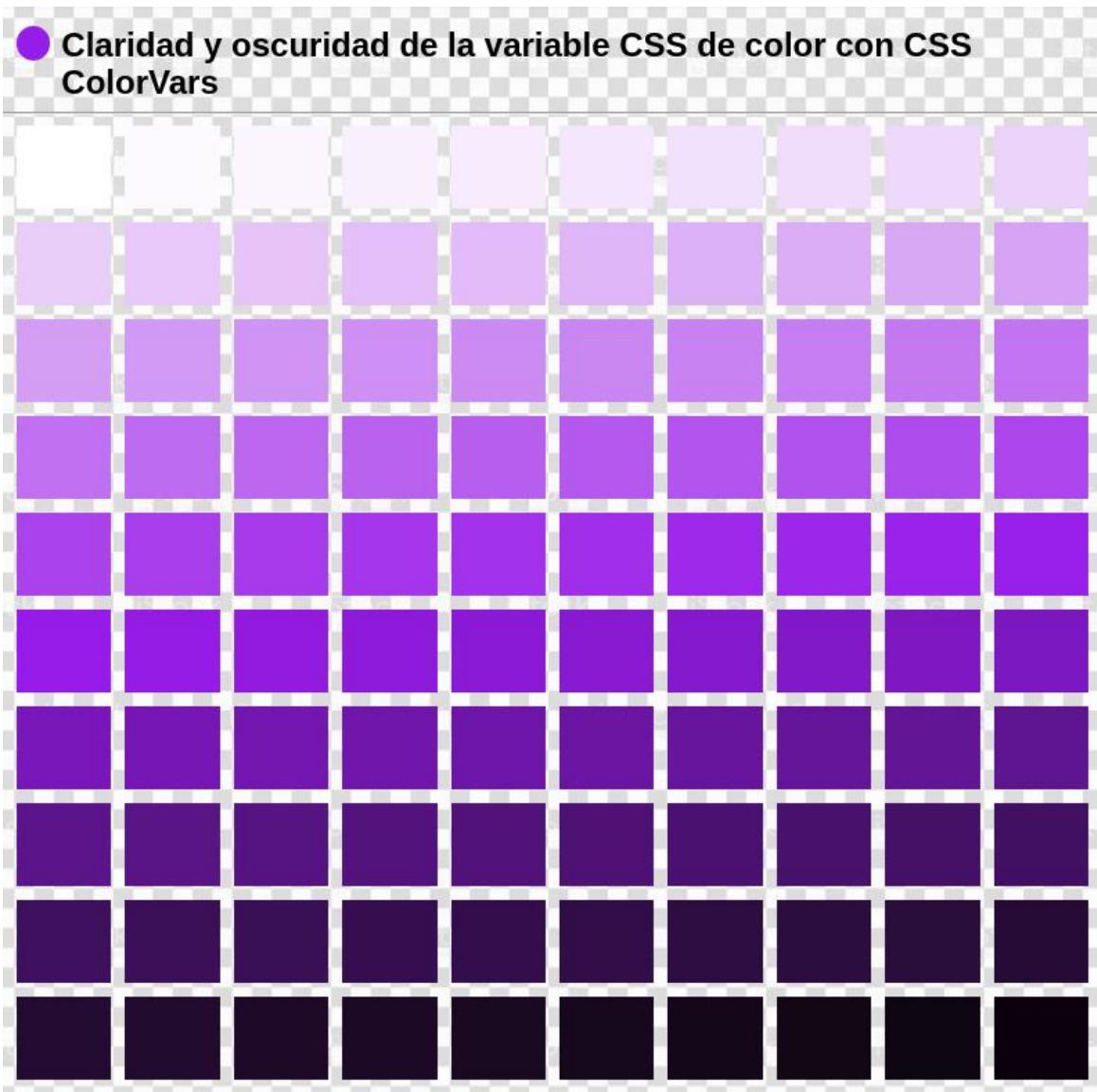
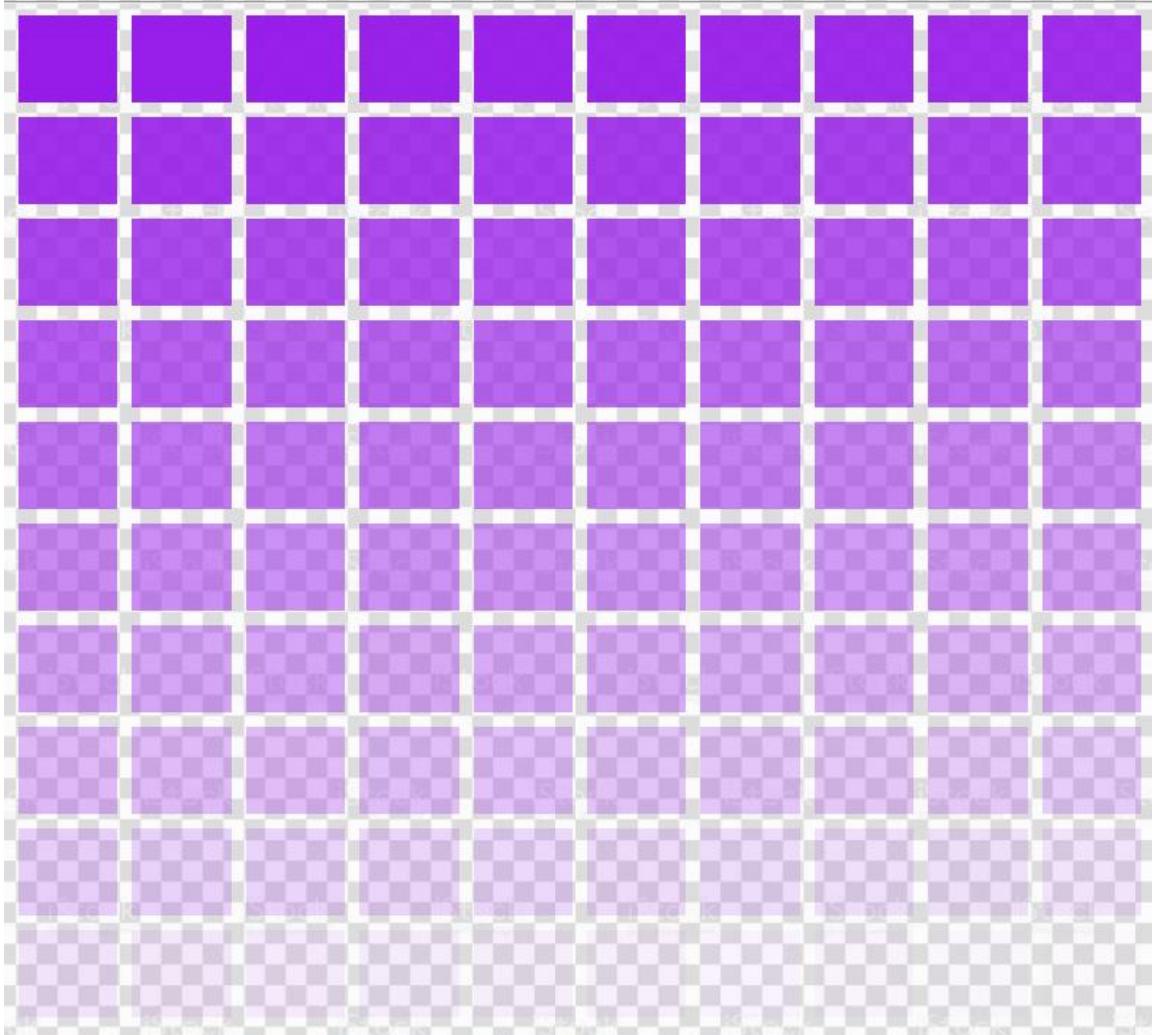


Gráfico 4.3: Salida en pantalla de variaciones de luminosidad.

Fuente: Elaboración Propia

● Trasparencia de la variable CSS de color con CSS ColorVars



**Gráfico 4.4: Salida en pantalla de variaciones de transparencia.**

Fuente: Elaboración Propia

#### **4.1.2. Análisis y síntesis para el efecto de rendimiento web.**

En este apartado se realizará el análisis de rendimiento del ejemplo real.

##### **1) Preparación y configuración para el análisis y síntesis**

Los ejemplos de comparación realizados en el anterior análisis, se subieron a un servidor y dominio de **GithubPages** para el análisis de rendimiento.

Se realizará el análisis de rendimiento en la ejecución de la hoja de estilos:

- **styles.css** (Archivo de desarrollo o sin minificar)
- **style.min.css** (Archivo de producción o minificado)

Los enlaces de las páginas web para el análisis de rendimiento son los siguientes:

- <https://nandes2062.github.io/solucion-sin-soporte>
- <https://nandes2062.github.io/solucion-sin-soporte/minificado.html>
- <https://nandes2062.github.io/solucion-con-soporte-de-csscolorvars/minificado.html>
- <https://nandes2062.github.io/solucion-con-soporte-de-csscolorvars>

##### **2) Page Speed Insights.**

Se realizó el diagnóstico utilizando **Page Speed Insights**, se utilizó esta tecnología porque sus **datos de laboratorio** son útiles para **depurar** problemas de **rendimiento**, ya que se recopilan **en un entorno controlado**.

En la siguiente grafica se puede observar los resultados del diagnóstico de una forma detallada:

 Diagnósticos

Consulta más información sobre el rendimiento de tu aplicación.

1 Minimiza la profundidad de las solicitudes críticas Se ha encontrado 1 cadena ^

Las cadenas de solicitud crítica que se muestran a continuación indican qué recursos son de alta prioridad. Te recomendamos que reduzcas la longitud de las cadenas, disminuyas el tamaño de los recursos o pospongas la descarga de recursos innecesarios para mejorar la carga de la página. [Más información](#).

Latencia de ruta crítica máxima: **60 ms**

*Navegación inicial*

```
└─ /solucion-sin-soporte/ (nandes2062.github.io)
    └─ /solucion-sin-soporte/styles.css (nandes2062.github.io) - 30 ms, 2,21 KB
```

 Diagnósticos

Consulta más información sobre el rendimiento de tu aplicación.

1 Minimiza la profundidad de las solicitudes críticas Se ha encontrado 1 cadena ^

Las cadenas de solicitud crítica que se muestran a continuación indican qué recursos son de alta prioridad. Te recomendamos que reduzcas la longitud de las cadenas, disminuyas el tamaño de los recursos o pospongas la descarga de recursos innecesarios para mejorar la carga de la página. [Más información](#).

Latencia de ruta crítica máxima: **140 ms**

*Navegación inicial*

```
└─ /solucion-sin-soporte/minificado.html (nandes2062.github.io)
    └─ /solucion-sin-soporte/styles.min.css (nandes2062.github.io) - 70 ms, 2,14 KB
```

#### Gráfico 4.5: Resultados del diagnóstico no soportados por la herramienta interactiva.

Fuente: GooglePage Speed Insights

## Diagnósticos

Consulta más información sobre el rendimiento de tu aplicación.

### 1 Minimiza la profundidad de las solicitudes críticas

Se ha encontrado 1 cadena

Las cadenas de solicitud crítica que se muestran a continuación indican qué recursos son de alta prioridad. Te recomendamos que reduzcas la longitud de las cadenas, disminuyas el tamaño de los recursos o pospongas la descarga de recursos innecesarios para mejorar la carga de la página. [Más información](#).

Latencia de ruta crítica máxima: **50 ms**

*Navegación inicial*



## Diagnósticos

Consulta más información sobre el rendimiento de tu aplicación.

### 1 Minimiza la profundidad de las solicitudes críticas

Se ha encontrado 1 cadena

Las cadenas de solicitud crítica que se muestran a continuación indican qué recursos son de alta prioridad. Te recomendamos que reduzcas la longitud de las cadenas, disminuyas el tamaño de los recursos o pospongas la descarga de recursos innecesarios para mejorar la carga de la página. [Más información](#).

Latencia de ruta crítica máxima: **50 ms**

*Navegación inicial*



**Gráfico 4.6: Resultados de diagnósticos soportados por la herramienta interactiva.**

Fuente: GooglePage Speed Insights

## **4.2. Evaluación.**

La evaluación articula los indicadores y el diagnóstico realizado. A continuación se observará las preguntas de la evaluación.

### **4.2.1. Preguntas de la evaluación.**

Las preguntas de evaluación se realizaron antes y después de la implementación de la herramienta interactiva:

- 1) ¿Cuál es la cantidad de variables CSS declaradas para un color, con el fin de utilizarlo para aplicar claridad, oscuridad, y transparencia?
- 2) ¿Cuáles la cantidad de procesos al declarar un color con variables CSS, con el fin de utilizarlo para aplicar claridad, oscuridad, y transparencia?
- 3) ¿Cuáles el tiempo de descarga del recurso en el ejemplo?
- 4) ¿Cuáles la cantidad de bytes de descarga del recurso en el ejemplo?

# **CAPÍTULO V.**

## **COMPROBACIÓN DE LA HIPÓTESIS**

## CAPÍTULO V

### COMPROBACIÓN DE LA HIPÓTESIS

#### 5.1. Comprobación de la hipótesis

Para las pruebas de Hipótesis Estadísticas, se ha utilizado el conocido procedimiento de los cinco pasos (método detallado en la decimotercera edición del libro Estadística Aplicada a los negocios y la economía por Lind, Marchal y Wathen).

Se utilizó la Prueba de Asociación de **Ji Cuadrado** porque que se requiere comprobar si una variable está asociada a otra variable. “Esta prueba se aplica en diseños de investigación en los que se estudia a un único muestreo poblacional y se realiza el análisis de la relación entre dos variables cualitativas.” (Dr. Jesús Reynaga Obregón, 2007)

Para lograr demostrar la hipótesis se dividió en dos:

- A) La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, mejora **la simplicidad en las variables** en un sitio web.
- B) La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, mejora **el rendimiento** en un sitio web.

Para evaluar en qué medida se ajusta la distribución de frecuencia obtenida se utilizará la siguiente tabla de Ji Cuadrado:

v/p	0,001	0,0025	0,005	0,01	0,025	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45
1	10,8274	9,1404	7,8794	6,6349	5,0239	3,8415	2,7055	2,0722	1,6424	1,3233	1,0742	0,8735	0,7083	0,5707
2	13,8150	11,9827	10,5965	9,2104	7,3778	5,9915	4,6052	3,7942	3,2189	2,7726	2,4079	2,0996	1,8326	1,5970
3	16,2660	14,3202	12,8381	11,3449	9,3484	7,8147	6,2514	5,3170	4,6416	4,1083	3,6649	3,2831	2,9462	2,6430
4	18,4662	16,4238	14,8602	13,2767	11,1433	9,4877	7,7794	6,7449	5,9886	5,3853	4,8784	4,4377	4,0446	3,6871
5	20,5147	18,3854	16,7496	15,0863	12,8325	11,0705	9,2363	8,1152	7,2893	6,6257	6,0644	5,5731	5,1319	4,7278
6	22,4575	20,2491	18,5475	16,8119	14,4494	12,5916	10,6446	9,4461	8,5581	7,8408	7,2311	6,6948	6,2108	5,7652
7	24,3213	22,0402	20,2777	18,4753	16,0128	14,0671	12,0170	10,7479	9,8032	9,0371	8,3834	7,8061	7,2832	6,8000
8	26,1239	23,7742	21,9549	20,0902	17,5345	15,5073	13,3616	12,0271	11,0301	10,2189	9,5245	8,9094	8,3505	7,8325
9	27,8767	25,4625	23,5893	21,6660	19,0228	16,9190	14,6837	13,2880	12,2421	11,3887	10,6564	10,0060	9,4136	8,8632
10	29,5879	27,1119	25,1881	23,2093	20,4832	18,3070	15,9872	14,5339	13,4420	12,5489	11,7807	11,0971	10,4732	9,8922
11	31,2635	28,7291	26,7569	24,7250	21,9200	19,6752	17,2750	15,7671	14,6314	13,7007	12,8987	12,1836	11,5298	10,9199
12	32,9092	30,3182	28,2997	26,2170	23,3367	21,0261	18,5493	16,9893	15,8120	14,8454	14,0111	13,2661	12,5838	11,9463
13	34,5274	31,8830	29,8193	27,6882	24,7356	22,3620	19,8119	18,2020	16,9848	15,9839	15,1187	14,3451	13,6356	12,9717
14	36,1239	33,4262	31,3194	29,1412	26,1189	23,6848	21,0641	19,4062	18,1508	17,1169	16,2221	15,4209	14,6853	13,9961
15	37,6978	34,9494	32,8015	30,5780	27,4884	24,9958	22,3071	20,6030	19,3107	18,2451	17,3217	16,4940	15,7332	15,0197
16	39,2518	36,4555	34,2671	31,9999	28,8453	26,2962	23,5418	21,7931	20,4651	19,3689	18,4179	17,5646	16,7795	16,0425
17	40,7911	37,9462	35,7184	33,4087	30,1910	27,5871	24,7690	22,9770	21,6146	20,4887	19,5110	18,6330	17,8244	17,0646
18	42,3119	39,4220	37,1564	34,8052	31,5264	28,8693	25,9894	24,1555	22,7595	21,6049	20,6014	19,6993	18,8679	18,0860
19	43,8194	40,8847	38,5821	36,1908	32,8523	30,1435	27,2036	25,3289	23,9004	22,7178	21,6891	20,7638	19,9102	19,1069
20	45,3142	42,3358	39,9969	37,5663	34,1696	31,4104	28,4120	26,4976	25,0375	23,8277	22,7745	21,8265	20,9514	20,1272
21	46,7963	43,7749	41,4009	38,9322	35,4789	32,6706	29,6151	27,6620	26,1711	24,9348	23,8578	22,8876	21,9915	21,1470
22	48,2676	45,2041	42,7957	40,2894	36,7807	33,9245	30,8133	28,8224	27,3015	26,0393	24,9390	23,9473	23,0307	22,1663
23	49,7276	46,6231	44,1814	41,6383	38,0756	35,1725	32,0069	29,9792	28,4288	27,1413	26,0184	25,0055	24,0689	23,1852
24	51,1790	48,0336	45,5584	42,9798	39,3641	36,4150	33,1962	31,1325	29,5533	28,2412	27,0960	26,0625	25,1064	24,2037
25	52,6187	49,4351	46,9280	44,3140	40,6465	37,6525	34,3816	32,2825	30,6752	29,3388	28,1719	27,1183	26,1430	25,2218
26	54,0511	50,8291	48,2898	45,6416	41,9231	38,8851	35,5632	33,4295	31,7946	30,4346	29,2463	28,1730	27,1789	26,2395
27	55,4751	52,2152	49,6450	46,9628	43,1945	40,1133	36,7412	34,5736	32,9117	31,5284	30,3193	29,2266	28,2141	27,2569
28	56,8918	53,5939	50,9936	48,2782	44,4608	41,3372	37,9159	35,7150	34,0266	32,6205	31,3909	30,2791	29,2486	28,2740
29	58,3006	54,9662	52,3355	49,5878	45,7223	42,5569	39,0875	36,8538	35,1394	33,7109	32,4612	31,3308	30,2825	29,2908

**Tabla 5.1: Tabla de la distribución de Ji Cuadrado.**

Fuente: Univ. Peruana Los Andes

Para la comprobación de hipótesis de Ji Cuadrado, si se desea obtener una mayor confianza en los datos se deberá establecer un **nivel de significancia** del 1%. Los niveles de significancia más bajos generalmente se usan en la manufactura al detectar defectos en productos

A continuación se procederá a realizar la prueba Ji Cuadrado para demostrar si existe asociación entre las variables.

**HIPÓTESIS A:** La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, **mejora la simplicidad en las variables** en un sitio web.

**Preguntas:**

- 1) ¿Cuál es la **cantidad de variables CSS declaradas para un color, con el fin de utilizarlo para aplicar claridad, oscuridad, y transparencia?**
- 2) ¿Cuál es la **cantidad de procesos al declarar un color con variables CSS, con el fin de utilizarlo para aplicar claridad, oscuridad, y transparencia?**

**Paso 1:**

$H_0$ : La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS **no mejora la simplicidad en las variables** en un sitio web.

$H_1$ : La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS **mejora la simplicidad en las variables** en un sitio web.

**Paso 2: Nivel** de significancia a utilizar será del 1%.

**Paso 3:** El estadístico de prueba a utilizar es “JI CUADRADO”.

**Paso 4:**

El análisis de los datos obtenidos se encuentra en el Capítulo 4 del presente documento.

A continuación, se muestra la tabla de frecuencias observadas.

Implementación de una herramienta interactiva	Simplicidad en las variables			TOTAL
	Cant. de variables declaradas	Cant. de procesos	Cant. de colores de la muestra	
Si	3	9	203	215
No	203	206	203	612
<b>TOTAL</b>	<b>206</b>	<b>215</b>	<b>406</b>	<b>827</b>

**Tabla 5.2: Tabla de la frecuencias observadas - Hipótesis A.**

Fuente: Elaboración propia.

A partir de la tabla de frecuencias observadas se elabora la tabla de **frecuencias teóricas**.

Implementación de una herramienta interactiva	Simplicidad en las variables			TOTAL
	Cant. de variables declaradas	Cant. de procesos	Cant. de colores de la muestra	
Si	53.56	55.89	105.55	215
No	152.44	159.11	300.45	612
<b>TOTAL</b>	<b>206</b>	<b>215</b>	<b>406</b>	<b>827</b>

**Tabla 5.3: Tabla de la frecuencias teóricas - Hipótesis A.**

Fuente: Elaboración propia.

Se procede a calcular el Ji cuadrado:

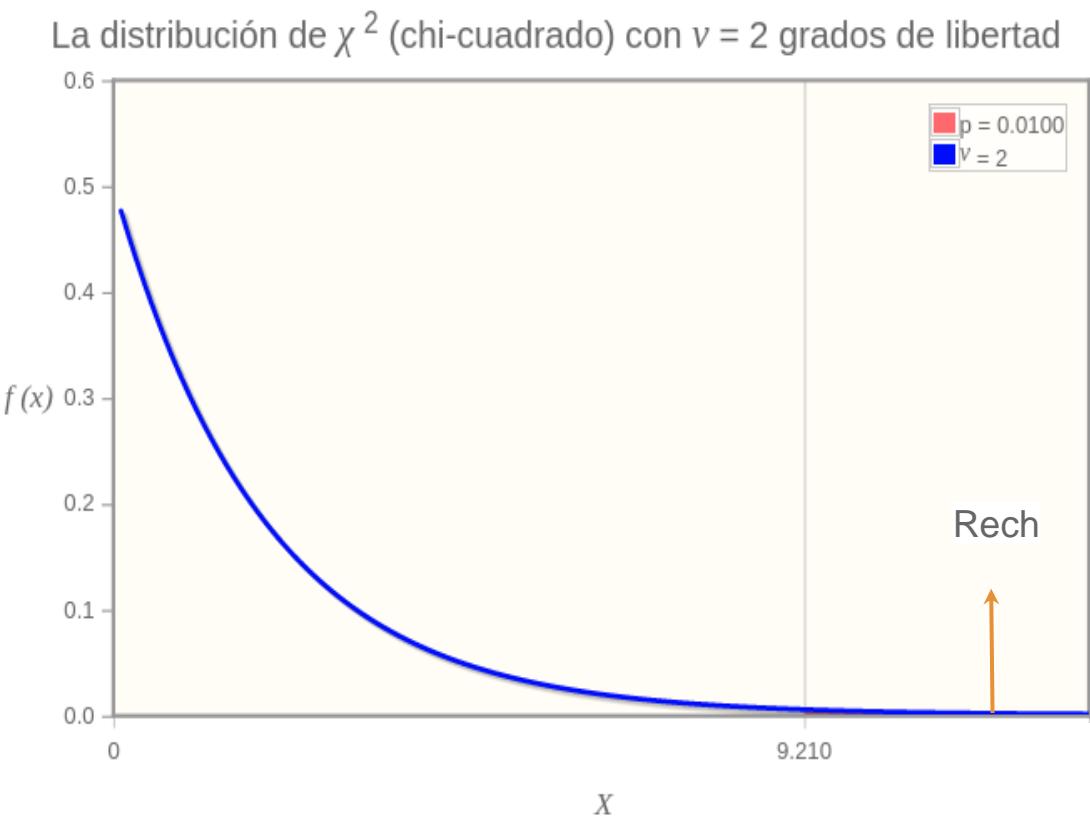
$$X^2 = \sum \frac{(f - f_t)^2}{f_t}$$
$$X^2 = \frac{(3 - 53.56)^2}{53.56} + \frac{(9 - 55.89)^2}{55.89} + \frac{(203 - 105.55)^2}{105.55} +$$
$$+ \frac{(203 - 152.44)^2}{152.44} + \frac{(206 - 159.11)^2}{159.11} + \frac{(203 - 300.45)^2}{300.45}$$
$$X^2 = 239.2329$$

Se procede a calcular los grados de libertad:

$$gl = (f - 1) \times (c - 1) = (2 - 1) \times (3 - 1) = 2$$

Como se trabajó con un nivel de confianza del 99%, tenemos que el valor del Ji Cuadrado con 2 grados de libertad es **239.2329**.

Por lo tanto, dado de  $239.2329 > 9.210$  se puede deducir que la Hipótesis nula es rechazada; es decir, el valor p es  $< 0,00001$ . El resultado es significativo en  $p < 0.01$ .



### Paso 5:

Comparando el valor del estadístico con el valor crítico, se concluye que la Hipótesis nula es rechazada, teniendo como conclusión que “La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, **mejora la simplicidad de las variables** de un sitio web”.

**HIPÓTESIS B:** La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, **mejora el rendimiento** en un sitio web.

**Preguntas:**

- 1) ¿Cuál es el **tiempo de descarga del recurso en el ejemplo?**
- 2) ¿Cuál es la **cantidad de bytes de descarga del recurso en el ejemplo?**

**Paso 1:**

$H_0$ : La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, **no mejora el rendimiento** en un sitio web.

$H_1$ : La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, **mejora el rendimiento** en un sitio web.

**Paso 2:**

Y el nivel de significancia a utilizar es del 1%.

**Paso 3:**

El estadístico de prueba a utilizar es “**JI CUADRADO**”.

**Paso 4:**

A continuación, se muestra la tabla de frecuencias observadas (tabla de datos obtenidos se obtiene mediante el ejemplo real).

Implementación de una herramienta interactiva		Rendimiento en un sitio web				TOTAL
		Tiempo de descarga del recurso (ms)	Latencia de ruta crítica máxima (ms)	Cantidad de bytes en descargar el recurso	Cantidad de colores de la muestra	
Si	Archivo sin minificar	30	110	930	203	<b>1273</b>
	Archivo minificado	20	60	900	203	<b>1183</b>
No	Archivo sin minificar	60	130	2140	203	<b>2533</b>
	Archivo minificado	30	70	2070	203	<b>2373</b>
<b>TOTAL</b>		<b>140</b>	<b>370</b>	<b>6040</b>	<b>812</b>	<b>7362</b>

**Tabla 5.4: Tabla de la frecuencias observadas - Hipótesis B.**

Fuente: Elaboración propia.

Re-categorizando a una matriz de 2x4, quedando la siguiente tabla de contingencias:

Implementación de una herramienta interactiva	Rendimiento en un sitio web				TOTAL
	Tiempo de descarga del recurso (ms)	Latencia de ruta crítica máxima (ms)	Cantidad de bytes en descargar el recurso	Cantidad de colores de la muestra	
Si	50	170	1830	406	2456
No	90	200	4210	406	4906
<b>TOTAL</b>	<b>140</b>	<b>370</b>	<b>6040</b>	<b>812</b>	<b>7362</b>

**Tabla 5.5: Tabla de contingencias observadas - Hipótesis B.**

Fuente: Elaboración propia.

A partir de la tabla de frecuencias observadas se elabora la tabla de **frecuencias teóricas**.

Implementación de una herramienta interactiva	Rendimiento en un sitio web				TOTAL
	Tiempo de descarga del recurso (ms)	Latencia de ruta crítica máxima (ms)	Cantidad de bytes en descargar el recurso	Cantidad de colores de la muestra	
Si	46.70	123.43	2014.97	270.89	2456
No	93.30	246.57	4025.03	541.11	4906
<b>TOTAL</b>	<b>140</b>	<b>370</b>	<b>6040</b>	<b>812</b>	<b>7362</b>

**Tabla 5.6: Tabla de la frecuencias teóricas - Hipótesis B.**

Fuente: Elaboración propia.

Se procede a calcular el Ji cuadrado:

$$X^2 = \sum \frac{(f - f_t)^2}{f_t}$$

$$X^2 = \frac{(50 - 46.70)^2}{46.70} + \frac{(170 - 123.43)^2}{123.43} + \frac{(1830 - 2014.97)^2}{2014.97} + \frac{(406 - 270.89)^2}{270.89} + \\ + \frac{(90 - 93.30)^2}{93.30} + \frac{(200 - 246.57)^2}{246.57} + \frac{(4210 - 4025.03)^2}{4025.03} + \frac{(406 - 541.11)^2}{541.11}$$

$$X^2 = 153.3201$$

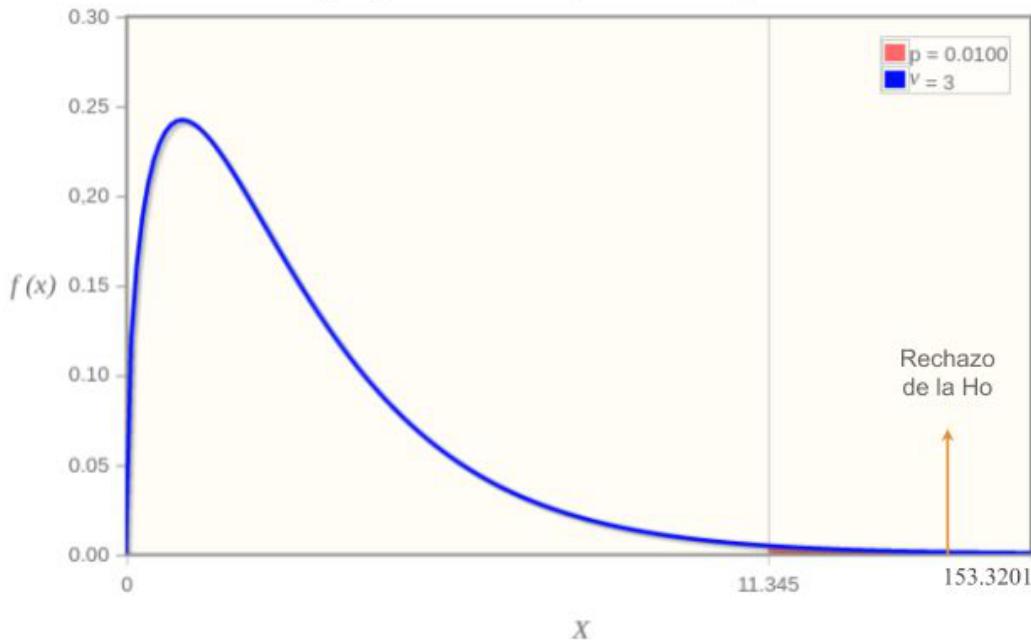
Se procede a calcular los grados de libertad:

$$gl = (f - 1) \times (c - 1) = (2 - 1) \times (4 - 1) = 3$$

Como se trabajó con un nivel de confianza del 99%, el valor del Ji Cuadrado con 3 grados de libertad es **153.3201**.

Por lo tanto, dado de  $153.3201 > 11.3449$  se deduce que la Hipótesis nula es rechazada; es decir, el valor p es  $< 0,00001$ . El resultado es significativo en  $p < 0.01$ .

La distribución de  $\chi^2$  (chi-cuadrado) con  $v = 3$  grados de libertad



**Paso 5:**

Comparando el valor estadístico con el valor crítico, se concluye que la Hipótesis nula es rechazada, teniendo como conclusión que “La implementación de una herramienta interactiva de generación de código, que define colores con variables CSS, **mejora el rendimiento** en un sitio web.

# **CAPÍTULO VI.**

**CONCLUSIONES Y RECOMENDACIONES**

## **CAPÍTULO VI**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **6.1. Conclusiones**

Luego de desarrollar y culminar el presente proyecto de investigación se obtuvieron las siguientes conclusiones:

- a) Se ha logrado implementar de forma satisfactoria una solución informática acorde al tipo de investigación de la presente tesis de grado, el cual sostiene que la implementación de una herramienta interactiva de generación de código que define colores con variables CSS, mejora la simplicidad en las variables y el rendimiento en un sitio web; esto se logró a través de determinar el efecto del objetivo establecido, el cual garantizó el éxito del mismo.
- b) Asimismo, se ha logrado con éxito documentar en los anexo las capturas de pantalla del software aplicado.
- c) Se logró implementar una herramienta interactiva de generación de código, a través de tecnologías web siguiendo un patrón arquitectónico y un patrón de diseño, con el fin de que la arquitectura de software sea escalable, lo cual es muy importante para el software que debe adaptarse constantemente a los cambios.
- d) Con la implementación de la herramienta interactiva de generación de código, se ha logrado establecer un medio para reducir la cantidad de procesos y la cantidad de variables CSS de color. (Ver capítulo 4 en el apartado de análisis para el efecto de simplicidad de variables) Como consecuencia de ello, se ha mejorado la simplicidad en las variables en un sitio web.

- e) Por último, con la implementación de la herramienta interactiva de generación de código, se ha logrado reducir el tiempo y la cantidad de bytes de descarga utilizadas en la ejecución del código generado (Ver capítulo 4 en el apartado de análisis para el efecto de rendimiento web). Como consecuencia de ello, se ha mejorado el rendimiento en un sitio web.

## 6.2. Recomendaciones

- a) Es recomendable, implementar el soporte para los preprocesadores faltantes como LESS, STYLUS y SASS.
- b) Es recomendable la integración de un sistema de respaldo, para recuperar variables de color.
- c) Es recomendable el soporte de internacionalización lingüística del proyecto web con preferencia al idioma español.
- d) Por último, en un futuro es recomendable implementar soporte con para los marcos de trabajo de CSS moderno.

## **Bibliografía:**

- Bell, A. (2018). CSS doesn't suck - Andy Bell.
- Albaugh, J. (2015). CSS4 Variables and Sass.
- Bembibre, C. (n.d.). Definición de Paleta de Colores.
- BoltClock. (2016). How do I apply opacity to a CSS color variable? | Stackoverflow
- CarlosPes.com. (n.d.). Definición de Variable (en programación).
- Cepeda, L. A. (2002). Generación de código en la programación Web avanzada.
- Codesido, I. (2009). What is front-end development?.
- Coyier, C. (2016). What is the difference between CSS variables and preprocessorvariables? | CSS-Tricks.
- Gómez, M. M. (2006). Introducción a la metodología de la investigación científica.
- Morales, J. I. S. (2016). Introducción: Variables dinámicas.
- Perna, M. A. (2018). A Practical Guide to CSS Variables (Custom Properties).
- Rodríguez, J. C. G. (2012). Herramientas interactivas basadas en tecnología web: aplicación al campo de la automática como caso de estudio.
- Sánchez, J. C. (2004). Metodología de la investigación científica y tecnológica.
- Torres, C. A. B. (2006). Metodología de la investigación.
- Verou, L. (2015). CSS Secrets: Better Solutions to Everyday Web Design Problems. "O'Reilly Media, Inc."
- Adrià Cadena (2017) ¿Que es la simplicidad?

- Drasner, Sarah. (2019) Desarrollador Senior Advocate en Microsoft
- W3C. (2015). Propiedades personalizadas de CSS para variables en cascada  
Módulo Nivel 1. Por la W3C.
- Wilson, D. (2017). Making Custom Properties (CSS Variables) More Dynamic |  
CSS-Tricks.
- W3C. (2018). Módulo de color CSS Nivel 3 por la W3C.
- Cegarra Sánchez, J. (2011). Metodología de la Investigación Científica y  
Tecnológica.
- Gómez, M. (2006). Introducción a la metodología de la investigación científica.  
Córdoba.
- Arta Antonovica, Comunicación e Imagen de los países Bálticos en España a  
través de la Técnica del discurso periodístico, pg.57.
- César Bernal Torres (2006). Metodología de la Investigación para administración,  
economía, humanidades y ciencias sociales, pg. 121.
- Ricardo Balduino . Introduction to OpenUP (Open Unified Process)
- Dr. Jesús Reynaga Obregón, Lecturas de apoyo sobre Estadística analítica.

## **Webgrafía:**

css-tricks.com

www.sitepoint.com

www.w3.org

www.carlospes.com

sarahdrasnerdesign.com

codepen.io/jakealbaugh

lea.verou.me

andy-bell.design

stackoverflow.com

www.dykinson.com/libros

inusual.com

www.academia.edu

www.theguardian.com

www.eclipse.org

# **ANEXO (A)**

Pantallas de la contribución y licencia del proyecto de código abierto.

1. Readme del proyecto de código abierto.



The screenshot shows a README.md file with the following content:

**CSSColorVars** (herramienta interactiva de generación de código), que define los colores con variables CSS, mejora la simplicidad de las variables y el rendimiento en un sitio web.

Ya no depende de los preprocesadores CSS para crear variables de color, use variables dinámicas CSS. La herramienta interactiva CSSColorVars te ayuda a crear tus colores con variables CSS y a usar funciones dinámicas CSS para aplicar claridad, oscuridad y transparencia en la variable de color.

## Configuración del proyecto

```
npm install
```

### Compila y recarga en caliente para el desarrollo

```
npm run serve
```

### Compila y minimiza para la producción.

```
npm run build
```

### Limpia y corrige archivos

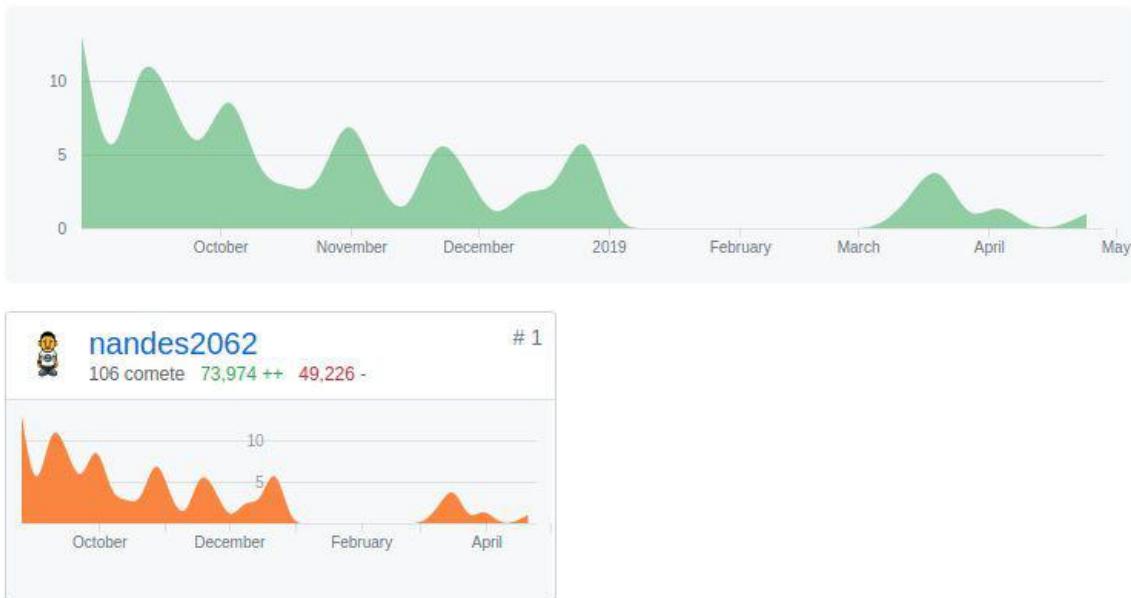
```
npm run lint
```

## 2. Gráfica de contribuciones al proyecto de código abierto

2 de septiembre de 2018 - 2 de mayo de 2019

Contribuciones: Compromisos ▾

Aportaciones al master, excluyendo las confirmaciones de fusión.



Fuente: (<https://github.com/CSSColorVars/csscolorvars/graphs/contributors>)

## 3. Licencia del software de código abierto.

 <b>Licencia MIT</b> CSSColorVars / csscolorvars está licenciado bajo la <a href="#">Licencia MIT</a>	<b>Permisos</b> ✓ Uso comercial ✓ Modificación ✓ Distribución ✓ Uso privado	<b>Limitaciones</b> ✗ Responsabilidad ✗ Garantía	<b>Condiciones</b> ⓘ Aviso de licencia y copyright
Esto no es un consejo legal. <a href="#">Obtenga más información sobre las licencias de repositorio</a> .			

## Licencia MIT

Copyright (c) 2018-presente, de Fernando Javier Averanga Aruquipa.

Por la presente, se otorga el permiso de forma gratuita a cualquier persona que obtenga una copia de este software y los archivos de documentación asociados (el "Software"), para en el Software sin restricción, incluyendo sin limitación los derechos para usar, copiar, modificar, fusionar, publicar, distribuir, sub-licenciar y/o vender copias del Software, y para permitir que las personas a las que el Software está Amueblado para ello, sujeto a las siguientes condiciones:

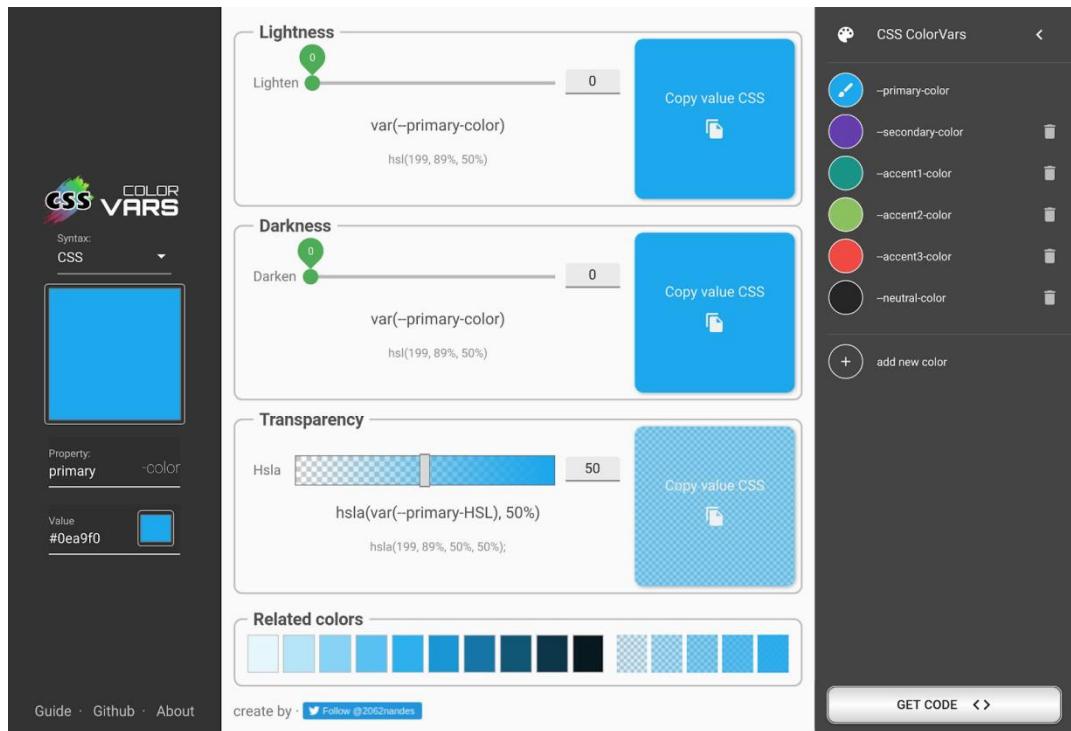
El aviso de copyright anterior y este aviso de permiso se incluirán en todas las Copias o partes sustanciales del Software.

EL SOFTWARE SE PROPORCIONA "TAL CUAL", SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUIDAS PERO NO LIMITADAS A LAS GARANTÍAS DE COMERCIABILIDAD, APTITUD PARA UN PROPÓSITO PARTICULAR Y NO INCUMPLIMIENTO. EN NINGÚN CASO DEBE LOS AUTORES O LOS TITULARES DEL DERECHO DE AUTOR SE HACEN RESPONSABLES POR CUALQUIER RECLAMACIÓN, DAÑO O OTRO LA RESPONSABILIDAD, YA SEA DE ACCIÓN DE CONTRATO, POR CORTE O DE OTRA MANERA, DERIVADA DE, FUERA DE O EN CONEXIÓN CON EL SOFTWARE O EL USO O OTRAS REPARACIONES EN EL SOFTWARE.

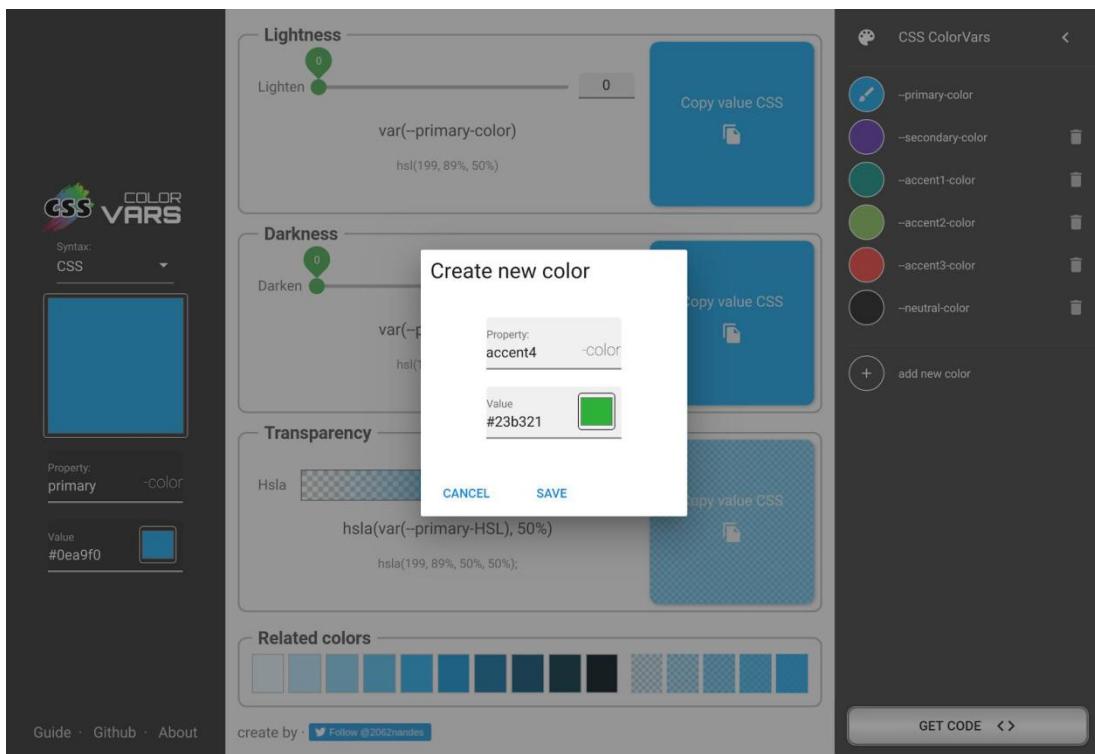
# **ANEXO (B)**

## Pantallas del software aplicado.

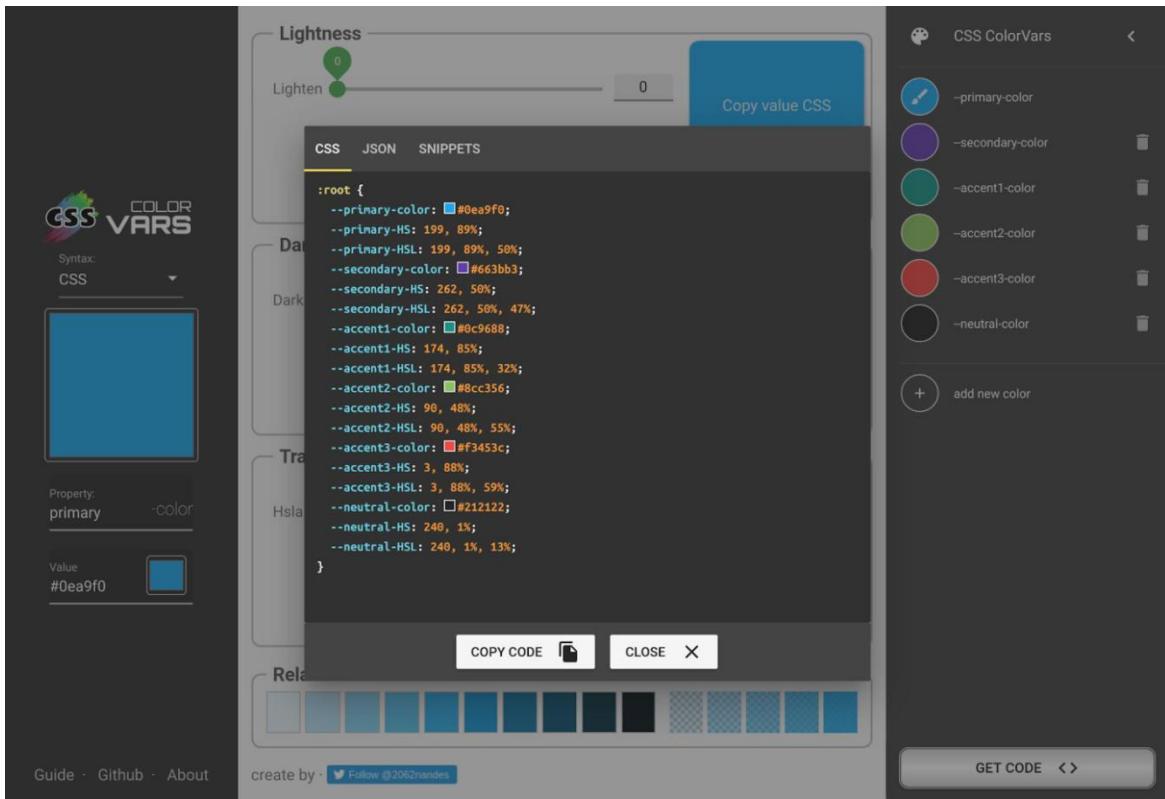
### 1. Pantalla principal de la herramienta interactiva.



### 2. Formulario para crear una nueva variable CSS de color.



3. Click en “Getcode” para copiar el Código CSS generado.



# **ANEXO (C)**

## Contribución de CSSColorVars en la Comunidad Awesome-Vue.js



### Interactive Experiences

- [Jean-Pierre Morin | 1700 LAPOSTE](#)
- [Facebook NewsFeed](#)
- [YouTube AdBlitz 2016](#)
- [Omnisense Experience](#)
- [Louis Ansa Website \(portfolio\)](#)
- [Djeco.com](#)
- [Tolks.io](#)
- [Meet Graham](#)
- [NOIZE original](#)
- [TR-101 Synth Drum Machine](#)
- [Bootstrap 4 Editor](#)
- [Subletab - Browser Extension](#)
- [web-riimote - Turn your smartphone into a 3D controller \(\[source code\]\(#\)\)](#)
- [CSS ColorVars - Interactive tool code generation \(\[source code\]\(#\)\)](#) ←



*Awesome Vue.js es una lista curada de proyectos impresionantes usando Vue.js*

Fuente: <https://github.com/vuejs/awesome-vue#interactive-experiences>

La herramienta interactiva “CSS ColorVars” cumplió con los requerimientos para ser aceptado en “Awesome-Vue.js” en la categoría “Interactive Experiences”, ya que cumple con las siguientes condiciones:

- La aplicación web está disponible sin errores ni problemas con el certificado SSL, y se carga en un tiempo razonable.
- La aplicación web está utilizando Vue.js intensivamente.
- La aplicación web es original y no es demasiado simple.
- La aplicación web está activa y mantenida.
- La aplicación web acepta aportes.

Existen dos enlaces provistos: el primero es la demostración ([csscolorvars.github.io](https://csscolorvars.github.io)), y el otro es el enlace del repositorio de github ([github.com/CSSColorVars/csscolorvars](https://github.com/CSSColorVars/csscolorvars)).

# **ANEXO (D)**

## Respuesta pública con reputación de experto en StackOverFlow.

<https://stackoverflow.com/questions/45496360/using-css-variables-as-sass-function-arguments>

stack overflow    Productos    Clientes    Casos de uso    Search...    Iniciar sesión    Registrarse

Casa    Usar variables CSS como argumentos de la función Sass    Pregunta

PÚBLICO    Desbordamiento de pila    Etiquetas    Los usuarios    Trabajos    ¿Qué es esto?

EQUIPOS    Preguntas y respuestas para el trabajo

13    :root {  
 --color: #ff0eff;  
}  
4 4    .div {  
 background-color: lighten(var(--color), 32%);  
}

¿Hay alguna manera de usar variables CSS con funciones Sass, por ejemplo, aclarar? Algo como esto:

Recibo un mensaje de error que dice "El argumento \$color de lighten(\$color, \$amount) debe ser un color".

Estoy usando variables CSS (no Sass), porque necesito usarlas en js.

css    css3    variables    sass    css-variables

compartir    mejorar esta pregunta    editado el 12 de diciembre de 17 a las 17:00    preguntó el 4 de agosto de 17 a las 0:00  
Jordan Whitfield    238 ▪ 3 ▪ 11    Majka    91 91 ▪ 1 ▪ 5

añadir un comentario

3 respuestas    votos    más antiguos    activos

1    Puede usar [CSS Colorvars \(csscolorvars.github.io\)](https://csscolorvars.csscolorvars.github.io/), y obtendrá una función de luminosidad y transparencia para poder aplicar claridad, oscuridad y transparencia en su variable de color.

:root {  
 --primary-color: #ff0c22;  
 --primary-HS: 355, 100%;  
 --primary-HSL: 355, 100%, 52%;  
 --secondary-color: #663399;  
 --secondary-HS: 270, 50%;  
 --secondary-HSL: 270, 50%, 40%;  
}  
.block{  
 background-color: hsl(var(--primary-HS), 60%);  
 /\*  
 Is equivalent to what is desired:  
 background: lighten(var(--primary-color), 8%);  
 \*/  
}

Enlace de CSS Colorvars: <https://csscolorvars.csscolorvars.github.io/>

Cualquier problema comenta en: <https://github.com/CSSColorVars/csscolorvars/issues>

compartir    mejorar esta respuesta    editado el 12 de mayo a las 4:20    respondió el 12 de mayo a las 4:09  
Fernando Javier Averanga    11 ▪ 2

añadir un comentario

R-0408

La Paz, 5 de septiembre de 2019  
CITE: USB/IDS/C-INT/237/19

Señora  
Abog. Susan Mabel Vargas Salazar  
**VICERRECTORA ACADÉMICA Y DE INVESTIGACIÓN**  
**UNIVERSIDAD SALESIANA DE BOLIVIA**  
Presente.-

Ref.: Aval de conformidad del Trabajo Final de Grado

De mi consideración:

Mediante la presente, tengo a bien poner en su conocimiento que, habiendo revisado el **Trabajo Final de Grado** del Universitario **AVERANGA ARUQUIPA FERNANDO JAVIER**, cuyo tema es: **ESTUDIO DE LA IMPLEMENTACIÓN DE UNA HERRAMIENTA INTERACTIVA DE GENERACIÓN DE CÓDIGO QUE DEFINE COLORES CON VARIABLES CSS EN UN SITIO WEB**, como Director de la Carrera de **Ingeniería de Sistemas** doy mi Aval de Conformidad para efectuar los empastados y la programación de fecha y hora para su defensa final ante el tribunal respectivo.

Sin otro particular, saludo a usted con las consideraciones más distinguidas.

Lic. Jesús Juan Rocha Vera  
**DIRECTOR DE LA CARRERA**  
**INGENIERÍA DE SISTEMAS**  
**UNIVERSIDAD SALESIANA DE BOLIVIA**



JR/dv  
c.c. Arch.  
Adj. lo indicado

La Paz, Septiembre del 2019.

Señor:  
Lic. Jesús Rocha Vera

DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS  
UNIVERSIDAD SALESIANA DE BOLIVIA

Presente.

Distinguido licenciado:

Tengo a bien dirigirme a su persona para darle a conocer la estructura y contenido de la tesis de grado “ESTUDIO DE LA IMPLEMENTACIÓN DE UNA HERRAMIENTA INTERACTIVA DE GENERACIÓN DE CÓDIGO QUE DEFINE COLORES CON VARIABLES CSS EN UN SITIO WEB”, doy fe que presente tesis de grado fue de elaboración propia y no fue plagiada, no existiendo otra observación se solicita poder acceder a la defensa final de tesis de grado.

Sin otro particular me despido de usted muy atentamente.



Univ. Fernando Javier Averanga Aruquipa  
RU. 25266  
Cel. 65636603

La Paz, Septiembre del 2019.

Señor:  
Lic. Jesús Rocha Vera

DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS  
UNIVERSIDAD SALESIANA DE BOLIVIA

Presente.

**REE.- AVAL DE CONFORMIDAD DE TESIS DE GRADO.**

Distinguido licenciado:

Tengo a bien dirigirme a su persona para darle a conocer que luego del seguimiento a la estructura y contenido de la tesis de grado "ESTUDIO DE LA IMPLEMENTACIÓN DE UNA HERRAMIENTA INTERACTIVA DE GENERACIÓN DE CÓDIGO QUE DEFINE COLORES CON VARIABLES CSS EN UN SITIO WEB", elaborado por el universitario Fernando Javier Averanga Aruquipa, no existiendo otra observación doy mi conformidad para que el mencionado postulante pueda acceder a su defensa final de tesis de grado.

Sin otro particular me despido de usted muy atentamente.



Ing. Carla Escobar  
Docente Guía

La Paz, Septiembre del 2019.

Señor:  
Lic. Jesús Rocha Vera

DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS  
UNIVERSIDAD SALESIANA DE BOLIVIA

Presente.

**REE- AVAL DE CONFORMIDAD DE TESIS DE GRADO.**

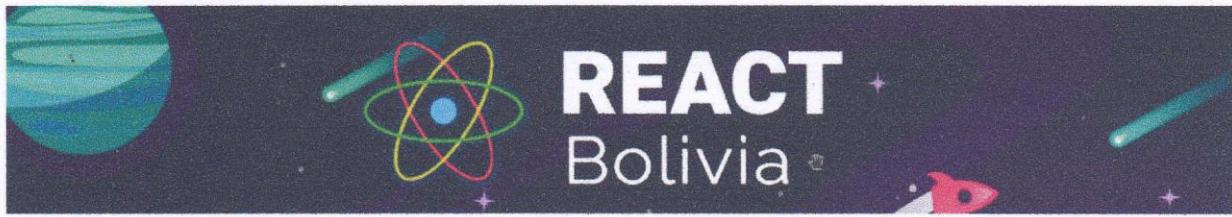
Distinguido licenciado:

Tengo a bien dirigirme a su persona para darle a conocer que luego del seguimiento a la estructura y contenido de la tesis de grado “ESTUDIO DE LA IMPLEMENTACIÓN DE UNA HERRAMIENTA INTERACTIVA DE GENERACIÓN DE CÓDIGO QUE DEFINE COLORES CON VARIABLES CSS EN UN SITIO WEB”, elaborado por el universitario Fernando Javier Averanga Aruquipa, no existiendo otra observación doy mi conformidad para que el mencionado postulante pueda acceder a su defensa final de tesis de grado.

Sin otro particular me despido de usted muy atentamente.



Ing. Rolando Sacaca  
Docente Revisor



La paz, Septiembre del 2019

## COMUNIDAD REACT BOLIVIA

De nuestra consideración:

Por la presente notificamos a ustedes, que nuestro experto Guillermo David Paredes Torrez de la organización/institución “React Bolivia”, apoya la realización del proyecto web de código abierto “CSS ColorVars - Herramienta interactiva de generación de código, que define colores con variables CSS, que mejora la simplicidad en las variables y el rendimiento en un sitio web”, llevado a cabo por el **Univ. Fernando Javier Averanga Aruquipa**, desde el 2 de febrero al 25 de abril del 2019.

Nuestro experto declaró conocer y aceptar los términos y condiciones del Proyecto, estando conforme con el aporte en el desarrollo web.

Sin otro particular, saludamos a ustedes atentamente.

Lic. Guillermo David Paredes Torrez

Organizador de React Bolivia

### Contactos

Puedes contactarte directamente con el Team Organizer.

Guillermo David Paredes Torrez - (+591) 60684585

Pablo M. Jordan - (+591) 70162630

Arnol Robles Tintaya - (+591) 67341446

community.react.bolivia@gmail.com