**UNIVERSIDAD MAYOR REAL Y PONTIFICIA DE SAN FRANCISCO XAVIER DE CHUQUISACA**

FACULTAD DE CIENCIAS Y TECNOLOGIA

**CARRERA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACION**

PERFIL DE PROYECTO DE GRADO



**“MODELO GENERADOR DE ESTILOS CSS EN BASE A ETIQUETAS HTML PARA DESARROLLADORES WEB”**

TUMIRI HUANCA ALEX

Sucre, 17 de Marzo del 2023

1. **INTRODUCCIÓN Y/O ANTECEDENTES**

En la actualidad el lenguaje de diseño y programación CSS tiene muchas ventajas, características, librerías, API’s y es utilizado en todos los navegadores y plataformas web del mundo, siendo así un estándar en el desarrollo de interfaces de usuario y en el desarrollo de aplicaciones web, el problema no reside en que la tecnología CSS sea una tecnología ineficiente para el desarrollo de software, o que tenga problemas de compatibilidad, o una ruta de aprendizaje compleja, todo lo contrario.

Los desarrolladores de software sobre todo especializados en la implementación de diseños e interfaces graficas durante mucho tiempo han intentado hacer su trabajo más eficiente y productivo en esta exhaustiva búsqueda de eficiencia y productividad los mismos programadores he ingenieros de software, han desarrollado diversas técnicas y herramientas las cuales facilitan su trabajo a la hora de crear software.

Estas herramientas aumentan la productividad y eficiencia sin embargo también aumentan la curva de aprendizaje y el tiempo para aprender las diferentes herramientas que constantemente se van actualizando día a día, sin mencionar que el proceso de desarrollo de estilos CSS no solo depende, que el desarrollador de software conozca muy bien la tecnología y los conceptos de la misma si no que tenga como vulgarmente se dice “un muy buen gusto”, el cual es importante para tener éxito con el desarrollo de la Interfaz de Usuario (UI) y Experiencia de usuario (UX), mientras que uno hace referencia a la experiencia y sensación del usuario, el otro está dirigido hacia un lado más racional de la navegación, también se deben considerar otros desafíos mayores como el buen diseño.

Las diversas herramientas también brindan un diseño estándar el cual es fácilmente reconocible y no es de muy buen gusto por lo cual no agrega valor al diseño de la interfaz.

1. **SITUACION PROBLEMÁTICA**

El problema, se halla en la poca productividad del ingeniero de software a la hora de desarrollar interfaces de usuario, diseño y estilos con el lenguaje de diseño gráfico CSS dado que este proceso consiste en escribir código repetitivo constantemente y a pesar de la gran cantidad de librerías disponibles para automatizar he incluso reducir la escritura de código, aumentando así la productividad del desarrollador. Estas librerías, frameworks y APIs comúnmente carecen de flexibilidad a la hora de desarrollar ofreciendo muchas veces estándares de diseños con colores predeterminados que no son de buen gusto y que para lograr el resultado deseado por el desarrollador se debe escribir más código he incluso tener más conocimientos sobre la librería que se este utilizando, muchas veces el trabajo solo se incrementa tanto en líneas de código como en tiempo de desarrollo, reduciendo la productividad y aumentando la complejidad del desarrollo y obviamente aumentando también el tiempo de desarrollo.

En algunas situaciones el desarrollador de software puede tener la capacidad y conocimiento para desarrollar interfaces de muy buena calidad con CSS, pero resulta que carece del buen gusto dando como resultado mucho tiempo perdido, por la indecisión a la hora de implementar estilos con CSS esto provoca que las interfaces desarrolladas no tengan un buen UI/UX, elementos que son muy importantes a la hora de desarrollar interfaces de usuario

Por lo general las diferentes herramientas para CSS que existen en la actualidad son muy diversas y muchas veces obtienes con estas el mismo resultado, también tienen una curva de aprendizaje compleja sin mencionar que debes conocer muy bien los conceptos y sintaxis del lenguaje CSS, por esta razón no son alternativas factibles para desarrolladores sin experiencia con CSS o que se estén iniciando en el desarrollo de interfaces de usuario.

1. **PROBLEMA CENTRAL**

El problema reside en la ineficiencia de los desarrolladores de software debido a que desde sus inicios el software es desarrollado por y para humanos, es imposible por no decir improbable erradicar el error e ineficiencia de los ingenieros de software dado que el error es una característica humana, pero esto no significa que no podamos mitigar el error y reducir la ineficiencia en el desarrollo de software, aumentando así la productividad, buenas prácticas y desarrollando de código limpio y escalable.

El desarrollo de software con la tecnología CSS como con otras tecnologías tiende a ser repetitivo, poco intuitivo, ineficiente y en su mayoría reduce la productividad del ingeniero de software a causa de las malas prácticas, desconocimiento de la tecnología CSS y otros factores que hacen del ingeniero de software sea improductivo a la hora de desarrollar software de buena calidad.

También la cantidad y diversidad de las librerías existentes en la actualidad solo incrementan la curva de aprendizaje y muchas veces no brindan la ayuda o solución al desarrollador de software incrementando así el tiempo de desarrollo y aumentando la complejidad para obtener el resultado deseado sin mencionar que muchas veces los diseños y código que ofrecen estas librerías, frameworks y APIs son estándares los cuales carecen de buen gusto esto hace que el desarrollador tenga que escribir mucho más código para lograr su objetivo.

1. **ABORDAJE DE SOLUCION**

Como solución se tiene automatizar y generar estilos CSS de manera eficiente en base a código HTML con un modelo de inteligencia artificial generador de estilos CSS será de gran utilidad e incrementará la productividad y eficiencia en el ámbito de desarrollo de software, optimizando así la calidad de código, el tiempo de desarrollo en esta tecnología.

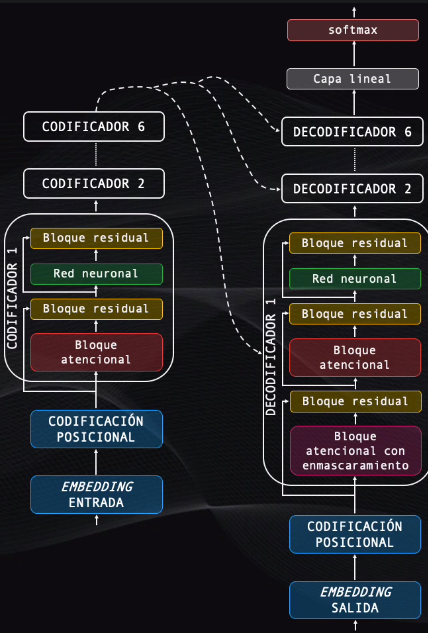
El desarrollo de un modelo de inteligencia artificial generará estilos CSS predefinidos y con características específicas, reconociendo el código HTML introducido por el desarrollador de software en una aplicación web intuitiva y fácil de usar que tendrá como motor de generación de código el modelo anteriormente mencionado, para reducir de manera notable el tiempo de desarrollo, aumentando la productividad y eficiencia del ingeniero de software, evitando que el ingeniero en cuestión escriba código repetitivo reduciendo así el tiempo de desarrollo con el cual el desarrollador de software podrá gestionar de manera más productiva su tiempo para optimizar y desarrollar un código de buena calidad y por consiguiente un mejor producto.

Se recopilará un gran volumen de datos de estilos CSS para el preprocesamiento de las diferentes características de la sintaxis del lenguaje CSS y clasificación de datos obtenidos para crear dos diferentes sets de datos que serán utilizados para el entrenamiento y pruebas de nuestro modelo generador de código CSS.

Como alternativa tecnológica para el desarrollo del modelo generador de código CSS se considera, luego de un análisis profundo e investigación sobre el estado actual del desarrollo de modelos de inteligencia artificial, es difícil ignorar el avance que se realizó en los últimos años en el área del Procesamiento del lenguaje natural (NLP) es una tecnología que brinda a las computadoras la capacidad interpretar, manipular, y comprender el lenguaje humano esta tecnología será de vital utilidad para reconocer, clasificar y extraer texto para que el modelo pueda aprender la sintaxis del lenguaje de programación CSS.

La arquitectura Transformer de redes neuronales será utilizada para el entrenamiento y optimización del modelo, esta arquitectura brinda la innovadora técnica del procesamiento en paralelo de la secuencia de parámetros introducidos a diferencia de las redes neuronales recurrentes donde lo parámetros se procesan de uno a uno de forma serial, a continuación, un gráfico de la arquitectura Transformer.

**Diagrama de la Arquitectura Transformers:**



**Fuente:** CODIFICANDOBITS Redes Transformer (... o el fin de las Redes Recurrentes). [En línea]. Disponible en Internet: <https://www.codificandobits.com/blog/redes-transformer/ > [Consulta: 03 de marzo 2023].

En el anterior grafico vemos todos los módulos de codificación y decodificación, también se utiliza las técnicas de embedding y los bloques atencionales toda esta arquitectura es muy eficiente e innovadora en el desarrollo de modelos de procesamiento de lenguaje natural y generación de texto específicamente, optimizando y ajustando la arquitectura Transformer a nuestro caso en específico obtendremos como resultado un eficiente y potente modelo de generación de código CSS

1. **OBJETIVO GENERAL**

Desarrollar una aplicación web para introducir como parámetro código HTML y generar código CSS con estilos predefinidos, servirá como herramienta de desarrollo de software para aumentar la productividad y reducir el tiempo de desarrollo del ingeniero de software para que el mismo pueda centrar sus actividades en otros aspectos del desarrollo de código igual de importantes como optimizar, refactorizar, y también aplicar mejores prácticas en el proceso de desarrollo de software.

1. **OBJETIVOS ESPECIFICOS**

* Realizar un modelo de inteligencia artificial que genere código CSS con un 90% de efectividad.
* Desarrollar una aplicación web para generar código CSS en base a etiquetas HTML.
* Reducir el tiempo de desarrollo del ingeniero de software.
* Reducir el desarrollo de código repetitivo.
* Creación de una Herramienta para el desarrollo de software ágil.

1. **JUSTIFICACIÓN**
   1. **Justificación socioeconómica**
   2. **Justificación tecnológica**

**5.3. Justificación medioambiental**

1. **MARCO TEÓRICO**
2. **ÍNDICE TENTATIVO**

**CAPÍTULO I: INTRODUCCIÓN**

* 1. Introducción
  2. Situación Problemática
  3. Objetivos
     1. Objetivo general
     2. Objetivo especifico
  4. Planteamiento del Problema
  5. Justificación
     1. Justificación socioeconómica
     2. Justificación tecnológica
     3. Justificación medioambiental

**CAPITULO II: MARCO TEÓRICO**

**CAPITULO III: INGENIERÍA DEL PROYECTO**

**CONCLUSIONES**

**RECOMENDACIONES**

**BIBLIOGRAFÍA**

**ANEXOS**

**PLANOS**

1. **CRONOGRAMA**
2. **COSTO DE ELABORACION DEL PROYECTO**
3. **BIBLIOGRAFIA**