

**SAIM BY MEDImages**

**Juan Jose Garcia Carabali**

**Karol Tatiana Guacas Piamba**

**Juan David Lopez Ramirez**

**Jessica Sthefanya Robayo Solarte**

**Introducción:**

Un simulador de artefactos de imágenes médicas es una herramienta diseñada para reproducir digitalmente los diferentes artefactos que pueden ocurrir en las imágenes generadas. Su propósito principal es ayudar a los profesionales y estudiantes relacionados con la salud, como radiólogos, técnicos de radiología y científicos médicos, a comprender y familiarizarse con los artefactos que pueden aparecer en las imágenes médicas y cómo pueden afectar la interpretación diagnóstica.

El simulador de artefactos de imágenes médicas se utiliza con fines educativos y de entrenamiento, así como para la investigación y el desarrollo de técnicas de corrección de artefactos. Al crear una representación digital precisa de los artefactos comunes, los usuarios pueden explorar y aprender cómo se ven y cómo se originan estos artefactos en las imágenes. Esto les permite desarrollar habilidades de detección y corrección de artefactos, lo que es esencial para una interpretación precisa y confiable.

En el simulador que se va a implementar se van a simular dos tipo de artefactos que se presentan de manera frecuente en estudios de imágenes médicas, uno es el artefacto de píxeles que se producen principalmente al daño en la unidad del equipo y el otro es el artefacto de ruidos que en general es por el estado y la configuración del equipo. El simulador estará cargado en una página web, la cual contará con una interfaz y un entorno que permita subir una imagen en formato JPG del estudio a analizar de rayos x. Posteriormente de subir el estudio el usuario podrá seleccionar entre estos dos artefactos y aplicarlos, el simulador le proporcionará en formato JPG cómo se vería el artefacto en el estudio y el usuario podrá exportarlo para que pueda guardarlo donde desee

Objetivos Generales:

* Desarrollar un software simulador de artefactos de imágenes médicas que sea capaz de generar artefactos realistas y controlables en diferentes modalidades de imágenes médicas.
* Mejorar la comprensión y el conocimiento de los profesionales de la salud sobre los artefactos de imágenes médicas y sus implicaciones en el diagnóstico y tratamiento de los pacientes.
* Proporcionar una herramienta de entrenamiento y educación que permita a los profesionales de la salud aprender a identificar y corregir artefactos en imágenes médicas de manera eficiente y precisa.

objetivos específicos:

* Implementar algoritmos de simulación de artefactos para diferentes estudios de rayos x.
* Ofrecer una interfaz de usuario intuitiva que permita a los usuarios especificar tipo de artefacto a simular.
* Proporcionar herramientas de visualización para comparar imágenes originales con imágenes simuladas, lo que permite una evaluación visual de la efectividad de la simulación de artefactos.
* Integrar opciones para guardar las imágenes simuladas.

**Audiencia objetivo:**

Usuarios: Los usuarios finales del simulador de artefactos de imágenes de RX son los profesionales de la salud, como radiólogos, técnicos de radiología, científicos médicos y estudiantes relacionados con el área de salud. Estos usuarios utilizan el simulador como una herramienta de aprendizaje y entrenamiento para comprender y reconocer los artefactos en las imágenes de RX, mejorar sus habilidades de interpretación y tomar decisiones diagnósticas más precisas.

Desarrolladores:Ingenieros biomédicos, estos profesionales están involucrados en la creación y el desarrollo del simulador de artefactos de imágenes de RX, trabajando en la programación, la implementación de algoritmos, la validación y la mejora continua del software. También serán los usuarios del simulador al probar y evaluar nuevas funcionalidades.

Administradores: ingenieros biomédicos, estos profesionales son responsables de gestionar y mantener el simulador de artefactos de imágenes de RX, asegurando su disponibilidad, actualización y adecuado funcionamiento, así como la planificación y asignación de recursos.

**Requisitos del sistema:**

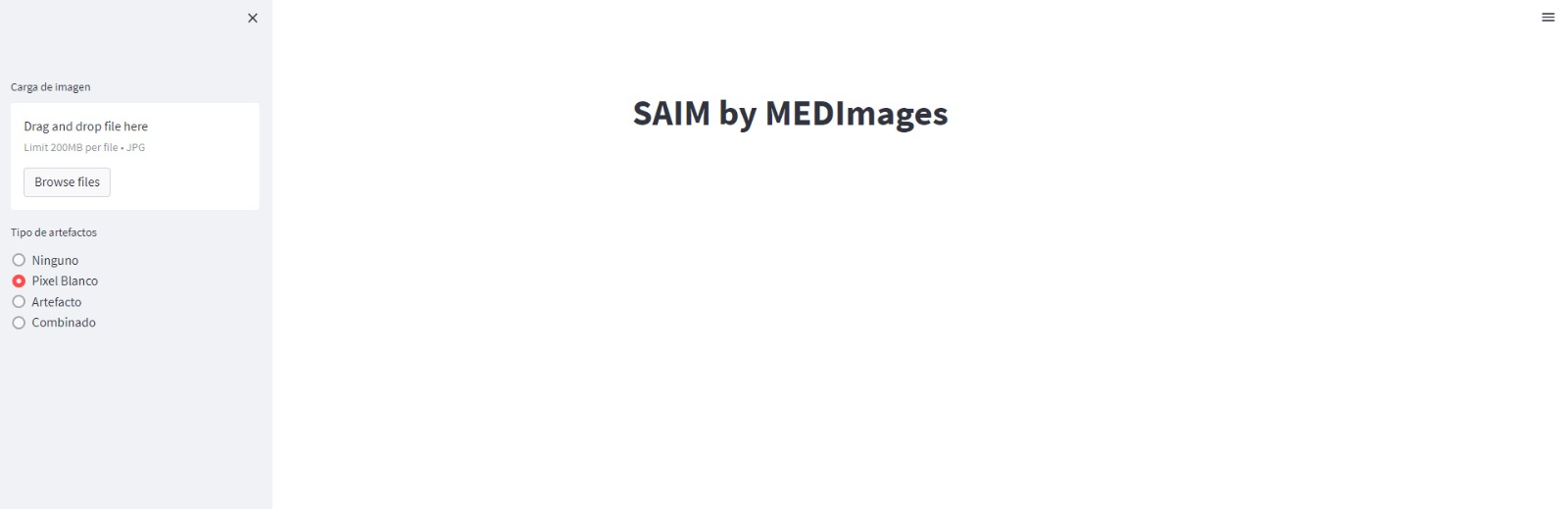
* Ordenador con capacidad de procesamiento de 2.66 GHz o más
* Memoria RAM: 2 GB.
* Tarjeta gráfica compatible que admita gráficos en 2D para visualizar las imágenes y los Artefactos simulados como Intel HD Graphics y NVIDIA GeForce GT 710
* Velocidad de conexión a internet: 3 Mb/s o superior

**Instrucciones para uso del software.**

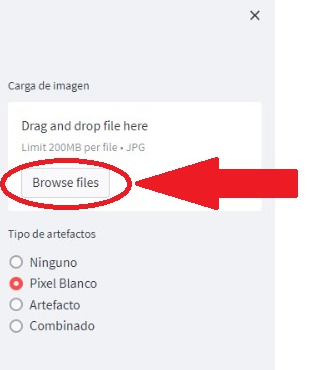
El usuario debe acceder a la página web del simulador a través de un navegador web compatible, los cuales son google chrome y edge.

Navegación por la interfaz

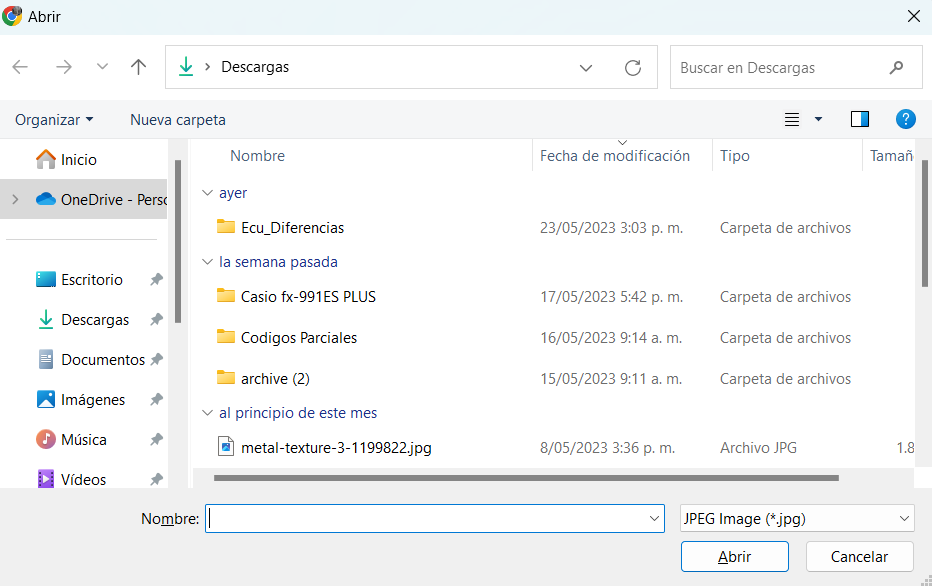
A continuación se muestra la interfaz de usuario del software:



Para cargar una imagen en la cual el usuario desea simular el artefacto, debe dar click en el botón de “Browse files” como se muestra en la siguiente imagen este apartado se encuentra en la parte superior izquierda de la interfaz:



Luego de presionar el botón “Browse files” posteriormente se le abrirá el selector de archivos en el cual el usuario podrá cargar su imagen médica a estudiar en formato JPG:



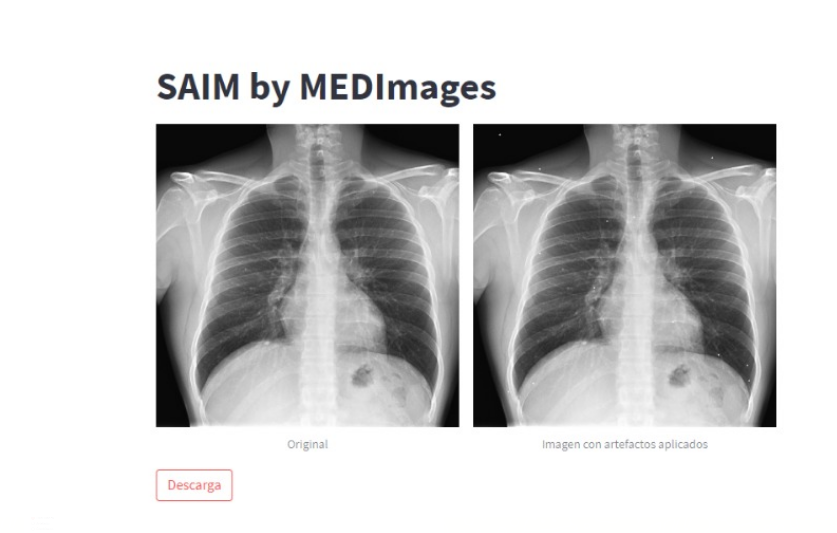
Si el usuario se equivoca de archivo o ya no quiere simular el artefacto en la imagen cargada podrá eliminar esta imagen dando click en la parte que ilustra la siguiente imagen:



Posteriormente de subir la imagen (peso máximo de la imagen 200MB ) el usuario podrá elegir entre ningún artefacto, artefacto de pixel blanco, artefacto por ruido y combinado, esto lo puede realizar en la parte de la interfaz que dice “tipo de artefactos” como se muestra en la siguiente imagen:

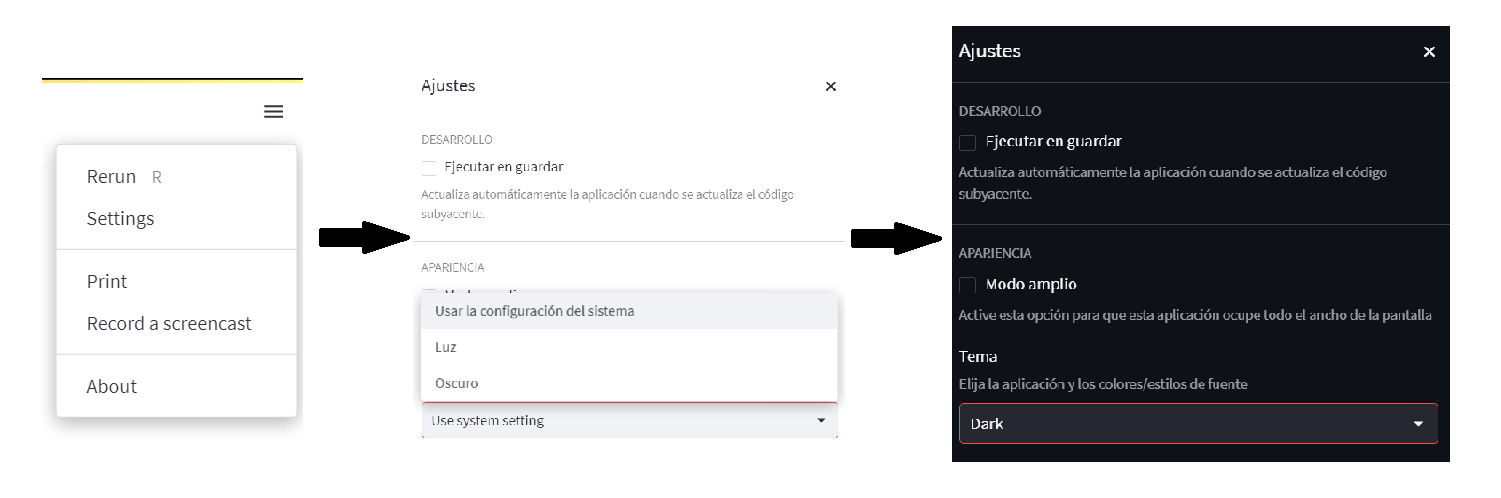


**Visualización de imágenes simuladas:** en la visualización el usuario podrá observar en la interfaz la imagen original y la imagen con los artefactos aplicados, además el usuario podrá descargar la imagen simulada

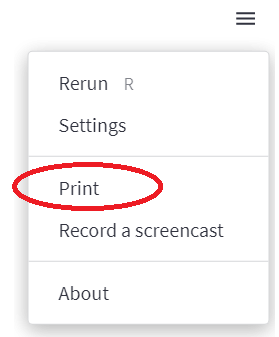


El simulador cuenta también con unas herramientas básicas que permite un agradable uso a continuación se definen estas herramientas y se ilustra cómo acceder a ellas:

* Modo oscuro: esta herramienta te proporciona un descanso visual mientras usas el simulador. Para aplicar esta herramienta te dijeres en la parte superior derecha de la interfaces en las tres líneas horizontales, ajustes y “use system setting”. Para una mejor compresión ilustramos los pasos en la siguiente imagen



* Imprimir: este herramienta te permite imprimir las imágenes si deseas tener una evidencia física de tu simulación. para acceder a esta herramienta lo ilustramos en la siguiente imagen:

****

**Documentación técnica:**

El desarrollo de la página web que simula artefactos de imágenes médicas se realizó en el lenguaje de python a continuación se muestra el código implementado con su respectiva explicación:

streamlit es una biblioteca de Python utilizada para crear aplicaciones web interactivas y de visualización de datos. Proporciona una forma sencilla de convertir scripts de Python en aplicaciones web rápidas y personalizables.

!pip install streamlit

para el siguiente fragmento de código es importante conocer la siguiente información:

* numpy: se utiliza para el manejo de matrices y números aleatorios
* cv2: se utiliza para el procesamiento de imágenes
* random: se utiliza para generar valores aleatorios
* PIL: se utiliza para manipular imágenes
* para la manipulación de bytes.

Funciones utilizadas en el código:

El comando %%writefile app.py te permite guardar el código de tu aplicación en un archivo separado, lo cual es útil para estructurar tu proyecto y ejecutarlo de manera independiente con Streamlit.

Función crear\_art\_PB: Esta función recibe una imagen (img) y un número de círculos (num\_circulos). Genera de manera aleatoria radios y posiciones para los círculos, crea una imagen en blanco con los círculos dibujados y luego la superpone a la imagen original. Finalmente, devuelve la imagen resultante con los artefactos aplicados.

Función add\_noise: Esta función recibe una imagen (image) y un valor de desviación estándar (sigma). Genera ruido gaussiano con la desviación estándar especificada y lo añade a la imagen original. Luego normaliza la imagen resultante en el rango de 0 a 255 y la devuelve.

función aplic: Esta función recibe una imagen (imag) y un tipo de artefacto (aa). Dependiendo del tipo de artefacto seleccionado, la función aplica diferentes transformaciones a la imagen. Si el tipo es "Ninguno", no se realiza ninguna modificación. Si el tipo es "Pixel Blanco", se agregan círculos blancos a la imagen. Si el tipo es "Artefacto", se aplica un desenfoque gaussiano y se añade ruido gaussiano a la imagen. Si el tipo es "Combinado", se aplican todas las transformaciones anteriores. La función devuelve la imagen resultante.

función main.

Verificación de la existencia de una imagen: La línea if imag1 is not None: verifica si la variable imag1, que representa la imagen cargada, no es None. En otras palabras, comprueba si se ha cargado una imagen.

Decodificación de la imagen: Si se ha cargado una imagen, decodifica los bytes de la imagen utilizando cv2.imdecode(). Se utiliza np.fromstring() para convertir los bytes de la imagen en un arreglo NumPy, y se especifica el tipo de datos (np.uint8) y el flag (1) para cargar la imagen a color.

Creación de columnas: Se utiliza st.columns(2) para crear dos columnas en la interfaz. Las columnas se asignan a las variables c1 y c2.

Visualización de la imagen original: Se muestra la imagen original en la primera columna utilizando c1.image(). El parámetro caption se utiliza para proporcionar un título o descripción de la imagen. use\_column\_width=True ajusta automáticamente el tamaño de la imagen al ancho de la columna.

La función aplic() aplica los artefactos correspondientes a la imagen y devuelve la imagen resultante, que se asigna a la variable imr.

Visualización de la imagen con artefactos aplicados: Se muestra la imagen resultante en la segunda columna utilizando c2.image(). Al igual que antes, caption="Imagen con artefactos aplicados" proporciona un título o descripción de la imagen, y use\_column\_width=True ajusta el tamaño de la imagen al ancho de la columna.

Creación del botón de descarga: A continuación, se crea un botón de descarga utilizando st.download\_button(). El parámetro label especifica el texto que se mostrará en el botón, data contiene los bytes de la imagen resultante, file\_name establece el nombre del archivo de descarga.

%%writefile app.py

import streamlit as st

import numpy as np

import cv2

import random

from PIL import Image

import io

#@st.cache(allow\_output\_mutation=True) #Guarda datos mientras se avanza

@st.cache\_resource

def crear\_art\_PB(img,num\_circulos):

radios = np.random.randint(1, 2, num\_circulos)

posiciones = np.random.randint(0, img.shape[0], (num\_circulos, 2))

artefactos = np.zeros\_like(img)

for i in range(num\_circulos):

cv2.circle(artefactos, tuple(posiciones[i]), radios[i], (255, 255, 255), -1)

img\_con\_artefactos = cv2.add(img, artefactos)

return img\_con\_artefactos

def add\_noise(image, sigma):

noise = np.random.normal(scale=sigma, size=image.shape)

noisy\_image = image + noise

noisy\_image = cv2.normalize(noisy\_image, None, alpha=0, beta=255, norm\_type=cv2.NORM\_MINMAX, dtype=cv2.CV\_8U)

return noisy\_image

def aplic(imag, aa):

if aa=='Ninguno':

imag=imag

elif aa=='Pixel Blanco':

imag=crear\_art\_PB(imag, 15)

elif aa=='Artefacto':

imag = cv2.GaussianBlur(imag, (15,15), 0)

imag=add\_noise(imag, 25)

elif aa== 'Combinado':

imag = cv2.GaussianBlur(imag, (15,15), 0)

imag=add\_noise(imag, 25)

imag=crear\_art\_PB(imag, 15)

return imag

def main():

st.title("SAIM by MEDImages")

imag1 =st.sidebar.file\_uploader("Carga de imagen", type=["jpg"])

sa=st.sidebar.radio("Tipo de artefactos", ["Ninguno", "Pixel Blanco", "Artefacto", "Combinado"])

if imag1 is not None:

imag2= cv2.imdecode(np.fromstring(imag1.read(),np.uint8),1)

c1,c2=st.columns(2)

c1.image(imag2,caption="Original",use\_column\_width=True)

imr=aplic(imag2, sa)

c2.image(imr,caption="Imagen con artefactos aplicados",use\_column\_width=True)

with io.BytesIO() as output:

img\_pil = Image.fromarray(imr)

img\_pil.save(output, format='JPEG')

img\_bytes = output.getvalue()

st.download\_button(label="Descarga", data=img\_bytes, file\_name="image\_art.jpg")

if \_\_name\_\_=='\_\_main\_\_':

main()

El comando “!ls” muestra los nombres de los archivos y directorios, esto se utiliza para comprobar si la codificación se guardó en un archivo app.py

!ls

Con estos comandos, estás iniciando la aplicación web de Streamlit localmente y luego utilizando LocalTunnel para hacerla accesible en línea a través de una URL pública. Esto permite compartir y mostrar la aplicación a los usuarios sin necesidad de desplegarla en un servidor web real.

!streamlit run app.py & npx localtunnel --port 8501

**Resolución de problemas:**

Preguntas frecuentes (FAQ)

: ¿Qué debo hacer si la simulación de artefactos no produce resultados realistas?

R: Verifica el tipo y la calidad de imagen médica que estás proporcionando. Asegúrate de que las imágenes médicas sean realistas y estén dentro de los límites adecuados. Si los resultados aún no son realistas, revisa la literatura científica relevante y compara tus resultados con estudios previos para identificar posibles problemas o mejoras en la simulación.

¿Qué debo hacer si encuentro errores o fallos en el software de simulación de artefactos?

R: Si encuentras errores o fallos en el software, primero verifica que estés utilizando la versión más reciente. Si el problema persiste, verifica si ya se ha informado el problema en la documentación. Si el problema es nuevo, considera informar al equipo de desarrollo o al proveedor del software, proporcionando detalles sobre el error, los pasos para reproducirlo y

cualquier mensaje de error o registro relevante. El equipo de desarrollo podrá investigar el problema y proporcionar una solución o una actualización del software.

¿Cómo puedo validar los resultados de la simulación?

R: La validación de los resultados de la simulación es crucial para garantizar la confiabilidad y la precisión del software. Puedes comparar tus resultados de simulación con datos experimentales, cuando estén disponibles, o con resultados de estudios previos publicados en la literatura científica

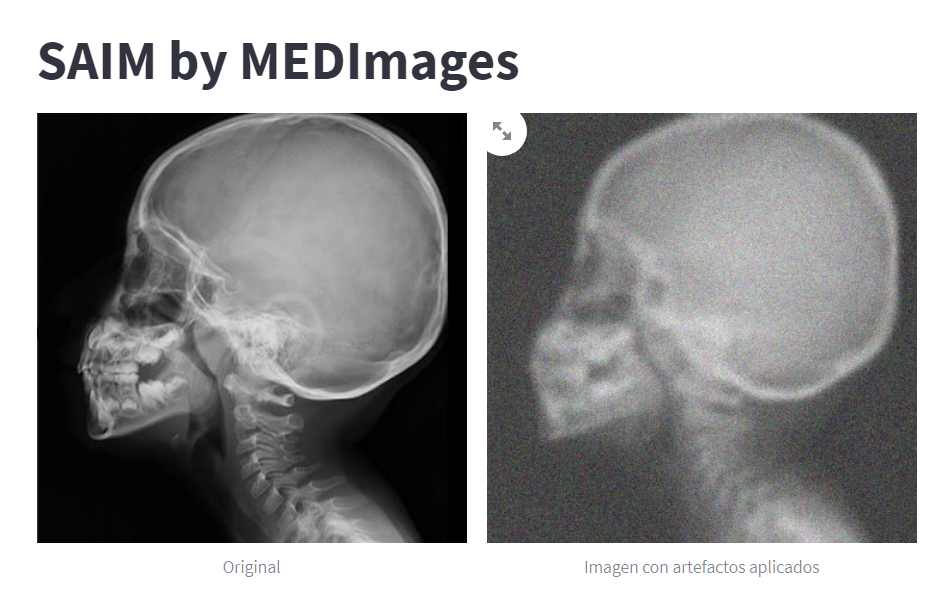
**Soporte, mantenimiento y atencion al cliente**

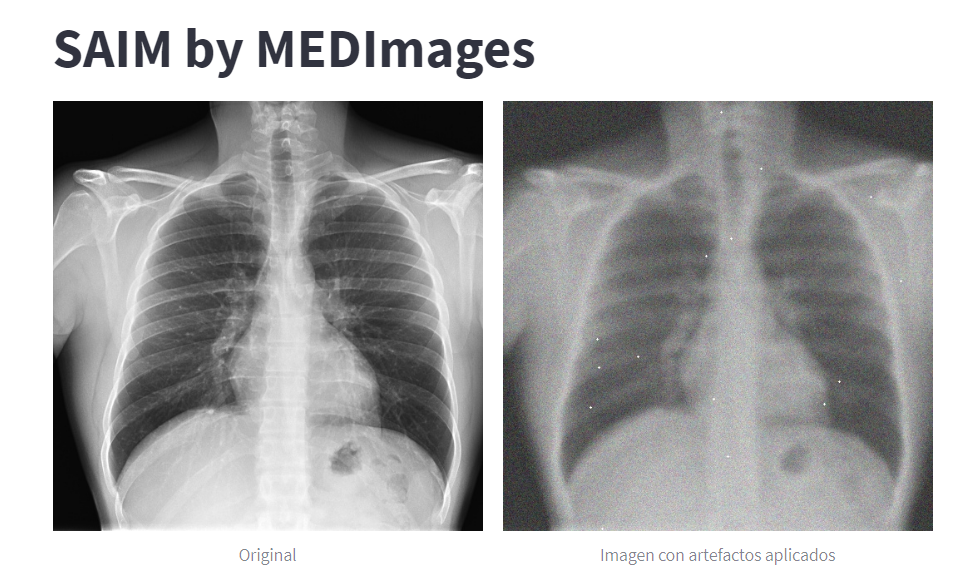
El soporte y mantenimiento de aplicaciones es imprescindible para realizar tareas de mejora continua y corregir errores, al hacerlo se asegura el funcionamiento del software, y todas las dudas que puedan surgir serán resueltas por el equipo de soporte, se introducen pequeñas mejoras en el producto, se facilitan las revisiones de seguridad y ciberseguridad, y se aportan el conocimiento y las herramientas necesarias para gestionar y abordar la complejidad de los procesos operativos de las organizaciones, tarea que se realizará con el equipo de creación del simulador.

Atención al cliente es el conjunto de acciones enfocadas en mejorar la experiencia de un cliente, con la cual se establece un proceso para resolver cualquier desafío, duda o problema del cliente de manera inmediata y efectiva. Esto puede hacerse por distintos medios de comunicación como teléfono, correo electrónico, chat en vivo, chatbots, tickets y redes sociales; este proceso sería realizado por el equipo de creadores del simulador.

**Ejemplos y casos de prueba:**

A continuación se muestran dos casos de una toma de rayos x, uno de cráneo en el plano sagital, en el cual se le aplica un artefacto de ruido y el otro de tórax en el plano frontal aplicando artefactos de ruido y de píxeles blancos.





**Definiciones de términos técnicos utilizados**.

* **Artefactos:**Los artefactos en son un aspecto de una imagen que falsifica las relaciones anatómicas y geométricas en el organismo, estas pueden degradar la calidad de imagen e interferir en el diagnóstico, en casos como los estudios de RX de coronarias, los artefactos son la causa del 100% de los falsos positivos y negativos, por lo que debemos aprender a reconocerlos.
* **Software:** El software puede definirse como todos aquellos conceptos, actividades y procedimientos que dan como resultado la generación de programas para un sistema de computación. En otras palabras, son las instrucciones que han sido predefinidas por un programador para ejecutar las tareas que se le indican.
* Simulador: Técnica numérica para conducir experimentos en una computadora digital; Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo

Referencias

* [1] Martínez, A. (2022). Software. *Concepto De - Definición De*. https://conceptodefinicion.de/software/
* [2] Team, K. (2023, March 10). ¿Qué es la documentación de software? *KeepCoding Bootcamps*. https://keepcoding.io/blog/que-es-la-documentacion-de-software/
* [3] Sartori, P., Rozowykniat, M., Siviero, L., Barba, G., Peña, A. S., Mayol, N., Acosta, D., Castro, J. M. G., & Ortiz, A. (2015). Artefactos y artificios frecuentes en tomografía computada y resonancia magnética. *Revista Argentina De Radiología*. https://doi.org/10.1016/j.rard.2015.04.005
* [4] SIMULACION Y ANALISIS DE SISTEMAS CON PROMODEL. (n.d.). Google Books. https://books.google.es/books?id=VuEfwtFr1QMC&lpg=PR6&ots=jwHEPZPmzI&dq=simulacion%20definicion&lr&hl=es&pg=PR6#v=onepage&q=simulacion%20definicion&f=false
* [5] De Enfermería, R. M. Y., & De Enfermería Ocronos, R. M. Y. (2022). Artefactos frecuentes en TAC. *Ocronos - Editorial Científico-Técnica*. https://revistamedica.com/artefactos-frecuentes-tac/
* [6] Fechter, J. (2022). How to Write Software Documentation in 7 Simple Steps. *Technical Writer HQ*. https://technicalwriterhq.com/documentation/software-documentation/how-to-write-software-documentation/
* [7] Fechter, J. (2023). 5 Types of Technical Writing. Technical Writer HQ. https://technicalwriterhq.com/writing/technical-writing/types-of-technical-writing/
* [8] Britton, W. E. (1965). What is technical writing? College Composition and Communication, 16(2), 113. https://doi.org/10.2307/354886
* [9] Morales, K. (2021, January 4). 5 real-life examples of beautiful technical documentation. *Work Life by Atlassian*. https://www.atlassian.com/blog/add-ons/5-real-life-examples-beautiful-technical-documentation
* [10] Moreno, J. (2023, abril 3). Qué es la atención al cliente, elementos clave e importancia. Hubspot.es. https://blog.hubspot.es/service/que-es-atencion-al-cliente
* [11] Conasa. (s/f). ¿Qué es el soporte y mantenimiento de aplicaciones? Grupocibernos.com. Recuperado el 25 de mayo de 2023, de https://conasa.grupocibernos.com/blog/que-es-el-soporte-y-mantenimiento-de-aplicaciones